International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 3, Issue 3, March 2016

Captcha as Graphical Password

[1] Greeshma Pillai, [2] Radhika Nair, [3] Rudhra Menon
[1][2][3] Department of Computer Engineering,
S.I.E.S Graduate School of Technology, Mumbai
[1] pillaigreeshma@siesgst.ac.in,[2] radhikanair18@siesgst.ac.in,[3] rudhra.menon@siesgst.ac.in

***Abstract*— As a security measure, nowadays, most systems have a mechanism for authenticating the user in order to log onto the system and exploit its functionalities. This can be done in different ways like entering a password, providing a fingerprint, voice pattern sample, inserting a smart card, or using some other means to prove to the system that you are who you claim to be. The commonly used authentication schemes, which involve text based passwords, have inherent security and usability problems. This resulted in the development of different graphical passwords schemes. The existing password authentication mainly focuses on using textual passwords and a Captcha authentication which is easily susceptible to relay attacks. Thus we propose a new security primitive based on hard AI problems, which involves integrating graphical password scheme and Captcha technology, which we name as Captcha as graphical passwords (CaRP).**

***Index Terms*— authentication, captcha, security, graphical passwords**

I. INTRODUCTION

A primary task in information security, or for that matter any field of computer security, is to create cryptographic primitives that are usually based on hard mathematical problems which are computationally difficult. Using AI (Artificial Intelligence)-complete or hard AI problems for security is a new security paradigm. To call a problem AI-complete reflects that it would not be solved by a simple specific algorithm, and hence hard. Under this paradigm, the most notable primitive invented is the Captcha, which distinguishes human users from computers by presenting a test, i.e., a puzzle, beyond the capability of computers but easy for humans. Captcha, an acronym for "Completely Automated Public Turing test to tell Computers and Humans Apart", which is now becoming a standard security technique to protect online email and other services from being abused by bots. Even though powerful, Captcha can still be bypassed through relay attacks in which Captcha challenges are forwarded to human solvers using other sites, whose answers are fed back to the targeted application.

The textual passwords are the most widely used ones for authentication. There is a common tendency among the users to choose short length passwords which are quite easy to memorize. However, such passwords are very much vulnerable to various attacks like brute force attack, dictionary attacks and so on. On the other hand, to make it resistant to such attacks, textual passwords need to be long and random-appearing, which are very difficult for the users to remember. As a result, graphical passwords are used as an attempt to replace textual passwords. Many schemes like Passfaces [2] have proved to be much easier to use and remember. Dictionary attacks on graphical password schemes are infeasible because of two main reasons: large password space and nonexistence of searchable dictionaries for graphical information. It is also difficult to launch automated attacks on such schemes.

After reviewing the existing graphical passwords schemes, we introduce a new security primitive which consists of graphical password systems integrated with Captcha technology, which we call CaRP (Captcha as gRaphical Passwords). This is a click-based graphical password scheme, where a sequence of clicks on an image is used to obtain the password. The difference between other graphical passwords and CaRP is that the images used in CaRP are Captcha challenges, and a new CaRP image is instantly generated for every login attempt.

II. REVIEW OF AUTHENTICATION TECHNIQUES

A. Graphical Passwords

Graphical passwords have been proposed as a possible substitute to text-based, motivated by the fact that humans can remember pictures better than text. A

variety of graphical password schemes have been implemented and they can be crudely classified into three main categories: recognition, recall, and cued recall. This classification of graphical passwords in the aforementioned categories is based on the task involved in memorizing and entering passwords. They are briefly explained below.

A recognition-based scheme requires identifying the visual objects belonging to a password set among a large set of objects. A typical scheme is Passfaces [2] wherein a user selects a set of faces from a database in creating a password. During authentication in this scheme, a panel of candidate faces is presented for the user to select the face belonging to her password set. This process is repeated several rounds, each round with a different panel. A successful login requires the correct selection in each round. Also to be noted is the set of images in a panel remains the same between logins, but their locations are changed. The main drawback of such schemes occurs due to the larger amount of pictures stored on the server side because of which the authentication process can be slow.

A recall-based scheme requires a user to remember and regenerate the same interaction result during every login attempt. Draw-A-Secret (DAS) [3] was the first recall-based scheme proposed in which user draws her password on a 2D grid provided. The system encodes the sequence of all the grid cells along the drawing path as a user drawn password. Pass-Go [4] improves DAS's usability by directly encoding the grid intersection points rather than the grid cells. Unfortunately, it was found that this scheme is very much vulnerable to attacks such as guessing, spyware, key-logger, and shoulder surfing.

In a cued-recall scheme, an external cue is provided to help recall and enter a password. PassPoints [5] is a widely studied click-based cued-recall scheme. In this scheme the user has to click a sequence of points anywhere on an image for creating the password, and then has to re-click the same sequence during authentication. The major problem with this scheme is related to the memorable password space. Users cannot just randomly click the background of the image provided since it will make the created password difficult to remember because of the simple background of the image.

Among the three types, recognition is considered the easiest for human memory whereas pure recall is the hardest [6]. However, recognition is typically the weakest in resisting guessing attack.

B. Captcha

A CAPTCHA is a type of challenge-response test which is used in computing to determine whether or not the user is human. There are basically two types of visual Captcha: text Captcha and Image Recognition Captcha (IRC). The first captcha relies on character recognition while the second completely relies on recognition of non-character objects, requiring users to identify simple objects in the images presented. Security of text Captchas has been extensively studied and it has been found out that character recognition CAPTCHAs are susceptible to computer vision based attacks simply due to the limited number of characters and digits within the English alphabet domain. The study conducted thus led to the establishment of the following principle: "Textual Captcha should certainly rely on the difficulty of their character segmentation which has to be strictly computationally expensive and combinatorically hard."

C. Authentication using Captcha

Captcha is used along with textual passwords in a user authentication protocol, which we call Captcha-based Password Authentication (CbPA) protocol,is used to counter online dictionary attacks. The CbPA-protocol in [1] requires solving a Captcha challenge after inputting a valid pair of login credentials unless a valid browser cookie is received. For an invalid pair of credentials, the user has a certain probability to solve a Captcha challenge before being denied access. An improved CbPA-protocol is proposed in [7] by storing cookies only on user-trusted machines and applying a Captcha challenge only when the total number of failed login attempts for the account has exceeded a threshold. It is further improved in [8] by applying a small threshold for failed login attempts from unknown machines but a large threshold for failed attempts coming from known machines with a previous successful login within a given time frame.

III. CAPTCHA AS GRAPHICAL PASSWORD

A. Need for CaRP

Conventional password authentication

techniques are susceptible to many kinds of attack. Hence we need such a technique that can be resistant to a large number of such cryptographic attacks. CaRP offers protection against online dictionary attacks on passwords, which have been a major security threat for various online services. This threat is extensive and is considered as a top cyber security risk [9]. Defense against online dictionary attacks is a more delicate problem than it might appear. Instinctive countermeasures such as throttling logon attempts do not work well for two reasons:

1. It causes denial-of-service attacks
2. It is vulnerable to global password attacks [1] whereby adversaries intend to break into any account rather than a specific one, exploiting the general tendency of the user to have the same password for different accounts, and thus try each password candidate on multiple accounts and ensure that the number of trials on each account is certainly below the threshold to avoid triggering account lockout.

CaRP also offers protection against relay attacks, an increasing threat to Captcha protection, wherein Captcha challenges are forwarded to humans to solve. Koobface [10] was a relay attack to bypass Facebook's Captcha in creating new accounts. Also, CaRP is robust to shoulder-surfing attacks if combined with dual-view technologies.

B. Key Idea

The proposed system combines Captcha and password into a single entity. A new image is generated for every login attempt, even for the same user. The CaRP is a Captcha challenge. A major difference between CaRP images and Captcha images is that all the visual objects in the password range (all alphabets and/or digits) should appear in a CaRP image, but only a subset of such objects is present in a normal Captcha image. The user clicks on the CaRP image corresponding to the objects of his password and is then provided access.

CaRP is a recognition-based graphical scheme built on top of textual Captcha. Currently it has been considered that the password is a sequence of characters in the number system, e.g., ρ ="6987", which is similar to a text password. The image is generated by the underlying Captcha engine as if a Captcha image were generated except that all characters (i.e. numbers 0 to 9) should appear in the image. During generation, each character's location is recorded in the generated image. The system relies on the recorded locations to identify the characters corresponding to user-clicked points. In these images, characters can be arranged randomly on 2D space. An example of the CaRP image as described above is shown in Figure1.

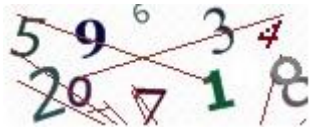

Figure 1.CaRP image consisting of only digits

C. User Authentication

User authentication in a CaRP scheme can be explained as follows. The system stores a salt s and a hash value h(ρ, s) for each user ID, where ρ is the password and is not stored directly in the database. A CaRP password is a sequence of the x and y coordinates of the 2D CaRP image that determines the exact location where the user has clicked on the image. Upon receiving a login request, system generates a CaRP image, records the locations of the objects in the image, and sends the clickable image to the user. The coordinates of the clicked points are recorded and sent to system along with the user ID. The system then attempts to extract those objects the user has intended to select by comparing the user's clicked points with the recorded click points and thus obtains the clicked password ρ'. Then, the system retrieves salt s of the account and calculates the hash value of ρ' with the salt. Both hash values-the generated and the stored- are compared and access is granted only if both values match. This process is shown in Fig. 2.

To recover a password successfully, each user-clicked point should be belonging to a single object. Since objects in a CaRP image may slightly overlap with neighboring objects in order to defend against segmentation, users should not click inside an overlapping region to avoid ambiguity in identifying the clicked object.

IV. SECURITY ANALYSIS

A. Automatic Online Guessing Attack

Online guessing attacks are the most common attacks on browser login and other user authentication systems. These attacks include brute force and

dictionary attacks. Brute force attack is trying every possible combination of password until you find the correct one while in dictionary attack a set of most likely words is formed and compared with the guessed password. Automatic online guessing attacks on existing graphical passwords are deterministic ,i.e, each trial in a guessing attack can always determine if the guessed password is the actual password or not, and all the password guesses can be determined by a limited number of trials.

However, in a CaRP image clickable points on one image are computationally-independent of clickable points on another image since every login attempt creates a unique image. Also, trials in guessing attacks are mutually independent. So guessing based on random clicking on the image cannot be carried out by a machine. It is computationally difficult for a machine to recognize the objects in every CaRP image and then test a password guess. The bots will be successful in finding a password only probabilistically no matter how many trials are executed.

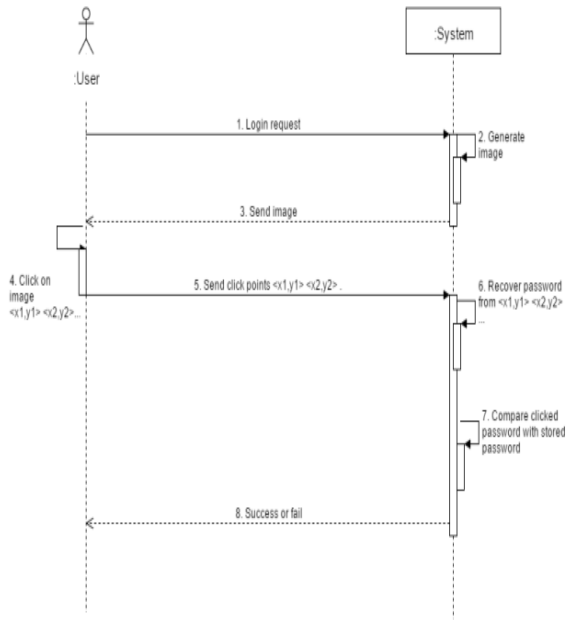

Figure 2. Flowchart of basic CaRP authentication

B. Human Guessing Attack

In human guessing attacks, humans enter the password by using trial and error method. Humans are much slower as compared to computers or bots in mounting guessing attacks. For 8-character passwords, the theoretical password space is 108 for CaRP with an alphabet range of 10 characters. If we assume that 1000 people are employed to work 8 hours per day without any stopping, and that each person takes 30 seconds to finish one trial. It would take them on average 0.5*108*30/ (3600*8*1000) ≈ 52 days to break a CaRP password. The human guessing attack can be slowed down further and the process can be made very expensive if we incorporate all English alphabets (i.e 26 characters) into the CaRP image and keep the password length variable (like 6 to 10 characters).

C. Relay Attacks

Captchas used on the internet are usually susceptible to relay attacks. These attacks either involve copying the Captcha image to a user somewhere else, making them unknowingly solve the Captcha, and then copying their response back to the original website; or employ large temporary teams of staff to solve Captchas in return of small payments. In CaRP scheme, the image displayed is not any usual captcha image. It is different from solving a normal captcha challenge. The person should have the knowledge of the password. Hence relaying the image to some unwitting user doesn't work in this case. In addition, human input obtained by performing a Captcha task on a CaRP image is useless for testing a password guess.

D. Shoulder-Surfing Attacks

A potential drawback in graphical password schemes is that they are more vulnerable to shoulder surfing than conventional alphanumeric text passwords. When the users input their passwords in a public place, they may be at risk of attackers capturing their password by means of direct observation or by recording the individual's authentication session. CaRP is not resistant to shoulder-surfing attacks by itself. However, it can be made resistant if combined with the dual-view technology. A new technology called dual-view support can display two images on a LCD screen concurrently, one public image viewable which is at most view-angles, and the other private image viewable only at a specific view-angle [12]. This technology exploits the technical limitation showed by commonly-used LCDs that varies brightness and color depending on the viewing angle. A CaRP image will be displayed as the private image by the dual-view system. So, a shoulder-surfing attacker will capture user clicked points on the screen, but will not capture the private CaRP image that only the user can see. Also the obtained user-clicked points are thus useless for another

login attempt, where a new, computationally-independent image will be used and the captured points will not be representing the correct password on the new image anymore.

E. Security of Underlying Captcha

Modern text Captcha schemes rely on Object segmentation being considered as a computationally expensive and combinatorically-hard problem [11]. According to [11], the complexity of object segmentation, C, is exponentially dependent of the number M of objects contained in a challenge, and polynomial dependent of the size N of the Captcha alphabet: $C = \alpha^M P(N)$, where $\alpha > 1$ is a parameter, and P() a polynomial function. A Captcha challenge typically contains 6 or more characters, whereas a CaRP image typically contains 10 or more characters. The complexity to break a CaRP image is about $\alpha^{10}P(N)/(\alpha^6P(N)) = \alpha^4$times the complexity to break a Captcha challenge generated by its underlying Captcha scheme. Therefore CaRP is much harder to break than its underlying Captcha scheme. Furthermore, characters in a CaRP scheme are arranged two dimensionally, thus further increasing the segmentation difficulty due to one more dimension to segment. As a result, we can reduce distortions in CaRP images for improved usability yet maintain the same security level as the underlying text Captcha.

V. CONCLUSION

The proposed system, CaRP, is a new security scheme used for authentication. It basically combines existing graphical passwords and captcha into a single scheme. The image used, called CaRP image, is a captcha challenge which is clickable as well. Here the password is entered into the system by means of clicking on appropriate characters on the image. This scheme attempts to counter online guessing attacks by using different CaRP images for every login attempt so that the trials of an online guessing attack are independent of each other. In addition to this, CaRP provides resistance to Captcha relay attacks, since a person cannot mistakenly attempt to solve a CaRP challenge. The complexity to break a CaRP image is multiple times of that to break a Captcha challenge generated by its underlying Captcha scheme. CaRP can also help reduce the problem of spam emails. Thus the CaRP scheme provides reasonable security and can be used for practical applications or related future work on hard AI problems.

ACKNOWLEDGEMENT

We would like to express our deepest appreciation to all those who provided us with the opportunity to complete this paper. A special gratitude to our final year project guide Prof. Aparna Bannore, whose contribution in stimulating suggestions and encouragement helped us to coordinate our project and to write this paper. We would also like to thank Head of Department Prof. Rizwana Shaikh for giving us the opportunity to take up this field as our BE project.

REFERENCES

[1] B. Pinkas and T. Sander, "Securing passwords against dictionary attacks," in Proc. ACM CCS, 2002, pp. 161–170.

[2] (2012, Feb.). The Science Behind Passfaces [Online].Available:http://www.realuser.com/published/ScienceBehindPassfaces.pdf

[3] 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.

[4] 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. 73–292, 2008.

[5] 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.

[6] 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.

[7] P. C. van Oorschot and S. Stubblebine, "On countering online dictionary attacks with login histories and humans-in-the-loop," ACM Trans. Inf. Syst. Security, vol. 9, no. 3, pp. 235–258, 2006.

[8] M. Alsaleh, M. Mannan, and P. C. van Oorschot, "Revisiting defenses against large-scale online password guessing attacks," IEEE Trans. Dependable Secure Comput., vol. 9, no. 1, pp. 128–141, Jan./Feb. 2012.

[9] HP TippingPoint DVLabs, Vienna, Austria. (2010). Top Cyber Security Risks Report, SANS Institute and Qualys Research Labs [Online]. Available: http://dvlabs.tippingpoint.com/toprisks2010

[10] N. Joshi. (2009, Nov. 29). Koobface Worm Asks for CAPTCHA [Online]. Available: http://blogs.mcafee.com/mcafee-labs/koobface-worm-asksfor- CAPTCHA

[11] K. Chellapilla, K. Larson, P. Simard, and M. Czerwinski, "Building segmentation based human-friendly human interaction proofs," in Proc. 2nd Int. Workshop Human Interaction Proofs, 2005, pp. 1–10.

[12] S. Kim, X. Cao, H. Zhang, and D. Tan, "Enabling concurrent dual views on common LCD screens," in Proc. ACM Annu. Conf. Human Factors Comput. Syst., 2012, pp. 2175–2184.