International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 20 Issue 3 – MARCH 2016.

CaRP – FOR AUTOMATIC ONLINE GUESSING ATTACKS

[1]N.Shobana, [2]V.Sowmiya, [3]R.Kanimozhi

[1,2]UG Scholar, Computer Science and Engineering, Dhanalakshmi College of Engineering
[3]Assistant Professor, Department of Computer Science and Engineering, Dhanalakshmi College of Engineering

shoba1495@gmail.com
sowmi27saran94@gmail.com
kanimozhiraja.88@gmail.com

***Abstract* - The Many security primitives relays on hard mathematical problems. The use of hard AI problems for security is as a new paradigm, but has been not explored. Here, we present a new security primitive based on hard AI problems, namely, a family of graphical password systems created on top of Captcha technology, which we say as Captcha as graphical passwords (CaRP). CaRP consists of both Captcha and a graphical password scheme. CaRP provides solution for a number of security problems, likewise relay attacks, online guessing attacks and, if added with dual-view technologies, shoulder-surfing attacks may be formed. Notably, a CaRP password can be detected only by automatic online guessing attacks and even if the password is present in the search set. CaRP also offers a novel approach to address the well-known image hotspot problem in graphical password systems, like as Pass Points that often leads to weak password choices. CaRP is not a solution for a remedy, but it offers proper security and usability and appears to fit well with some practical applications for improving online security.**

***Index Terms* – CaRP, captcha, graphical password, security primitive.**

I. INTRODUCTION

A simple task in a security is to make the primitives of crypto-graphic related on hard mathematical problems that are difficult to compute. The logarithm problem is basic to the Digital Signature Algorithm ElGamal encryption, the elliptic curve cryptography, the Diffie-Hellman key exchange, and so on. By the use of hard Artificial Intelligence problems for security, previously proposed in a new paradigm. Under this paradigm, the most identifying is that the primitive invented is Captcha, which distinguishes users from computers by presenting a challenge, i.e., a quiz, which is beyond the ability of computers but easy for humans. Captcha is now a common internet security technique to safeguard online email and other services from being attacked by bots. However, this paradigm has achieved a limited success.

We announce a new security primitives related to hard AI problems, namely, a fictional family of graphical password systems combining Captcha technology, which we call CaRP (Captcha as gRaphical Passwords). CaRP is of click-based on graphical passwords; here a sequence of clicks on an image is used to derive a password. In previous click-based graphical passwords, images used in CaRP are the challenges, and a new CaRP image is generated for every login attempt made. The idea of CaRP is simple but generic. CaRP can have multiple instances. In theory, any Captcha scheme depending on multiple-object classification can be changed to a CaRP scheme. We propose CaRPs made on both texts Captcha and image-recognition Captcha. One of them is a text CaRP is a password is a sequence of characters like a text password, are entered by clicking the right character on CaRP images. CaRP offers protection against online dictionary attacks on passwords, which is a threat for various online services and considered as a top cyber security risk , protection against online dictionary attacks is a more difficult problem than it appears. Counter measures such as throttling log on attempts do not work well for two reasons:1) It causes denial-of-service attacks. 2) It is dangerous to global password attacks.

CaRP also provides protection against relay attacks, which is an increasing threat to Captcha protection, wherein Captcha challenges are depending on humans to solve. CaRP is nice to shoulder-surfing attacks .This impact on usability can be solved by adapting the CaRP image’s difficulty level based on the login history of the users . Typical scenarios for CaRP are: 1)CaRP can be applied on touch-screen devices whereon typing passwords is easy example for secure Internet applications like e-banks. Many e-banking applications have applied Captchas in user logins. For example IDBC, the largest bank in the India, requires the users in solving a Captcha challenge for every online login attempt. 2) CaRP reduces spam emails. For an email service provider, a spam but cannot log in an account even though it has the password. Instead, human is compulsory to access an account. If CaRP is combined with a policy were by to the number of emails sent to new recipients per login session, a spam but can send only a small number of emails before leaving human for login, leading to reduced outbound spam traffic.

II. SYSTEM ARCHITECTURE

Step1:The user first receives the captcha image then he wants to select any region as a password (mainimum one, maximum

three).The details of the captcha password which the user gave will be catched in the captcha server.
Step2: Whenever the user wants to login into his account captcha image will be appeared, and he as to click on the correct region which he gave as the captcha password for the first time.
Step3: Then the captcha server will have a verify server which is used to verify whether the user has gave the password correctly or not.
Step4: Then the verify server will send acknowledgement whether the password is correct or not.
Step5: The application server will make the user to proceed further if the password is correct.

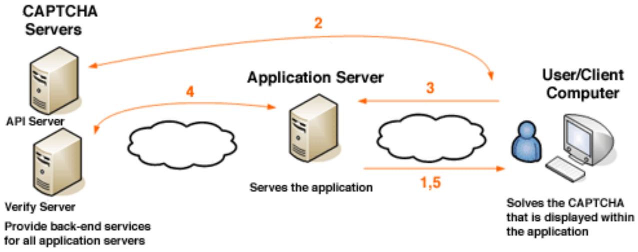
CAPTCHA Servers
2
Application Server
3
User/Client Computer
4
API Server
Verify Server
Provide back-end services for all application servers
Serves the application
1,5
Solves the CAPTCHA that is displayed within the application

Fig. 1system architecture

III. EXISTING SYSTEM

A Brute force attack and dictionary attack on password is only on remote login services were the Existing techniques and proposals made involves with the below assumption that these challenges are mainly difficult for bots and easy for people. However, users majorly dislike ATTs as these are performed as an extra step; for usability issues relatively to commonly used CAPTCHAs. Attackers can make only limited number of guessing from a single machine before locked- out, delaying, or raising challenge to answer Automated Turing Tests. Account locking is a mechanism to prevent from attempting multiple passwords for a particular user. Locking is generally temporary, the adversary can stop a DoS attack by making recently failed login attempts to lock a particular account. Delaying the server response after receiving the user credentials, whether the password is correct or incorrect, it prevents the adversary from making a large no of passwords in a small amount of time for a particular user. Traditional password-based authentication is not good for any un-trusted environment like a key logger may record all keystrokes, including passwords in a system, and forward those to a remote attacker. We do not stop those existing such attacks in un trusted environments, and assume that any machines that make users use for login are trustworthy.

A graphical password system with a supportivity sound signature is done to the existing system, Blonder-style passwords relays on cued recall. A user clicks on several previous chosen locations in a single image to log in. As done by Pass logic Corporation, the user selects certain predefined regions in an image as password. To log into the user account they have to click on the same regions cued click points (ccp) is a made that is alternative to pass points. In ccp, users click one point on each 4 images rather than on five points on single image. It offers cued-recall and introduces visual cues that readily alert correct users if they have done a mistake when they enter their latest click-point. It also makes attacks related on to a hotspot analysis more challenging. Each click results in showing a next-image.

1. A wrong click leads to an incorrect path, which will denote authentication failure only after the final click.

2. Users can select their images only to an extent and the click point indicates the next image.

A. *GRAPHICAL PASSWORDS*

Here we proposed a large number of graphical password scheme. It can be divided into three major categories, recall, and recognition. A recognition-based technique will identify decoys and the visual objects related to a password portfolio. A typical method is Pass faces wherein a user picks a portfolio of faces from a database for creating a password. During authentication, a group of candidate faces is represented to the user to select the face belonging to the portfolio. This process is repeated certain rounds, each round will have a different panel. A successful login can be done with correct selection in each round. For every login the set of images in a panel remains the same, but their locations are changed. Story is similar to pass faces but the images in the portfolio are listed and a user must identify her portfolio images in the order.

Cognitive Authentication requires a user to create a path through a collection of images as follows: starting from the top-right image, moving down if the image is in the portfolio or left otherwise. This process is made, for each time with a different set. A successful login requires that the probability that correct answers if they do not enter by chances provided it exceeds a threshold within a given no of rounds. A recall-based scheme requires a user to recreate the same communication result without cueing. Draw-A-Secret (DAS) was the initial recall-based scheme proposed. A user draws the password on a 2D grid. In a cued-recall scheme, an external cue is provided to help remember and enter a password. Pass Points is made, click-based cued-recall scheme wherein a user clicks a set of points anywhere on an image in creating a password, and re-clicks the same during authentication.

Cued Click Points (CCP) is same as to Pass Points but it makes use of one image per click, with the next image selected by a function. Persuasive Cued Click Points (PCCP) extends CCP by requiring a user to select a point inside a set of randomly placed view port when establishing a new password, resulting in more randomly distributed click-points in a password. Among the three types, recognition is the easiest for human memory but pure recall is the hardest. Recognition is typically the weakest in guessing attacks. Many proposed recognition-based schemes have a password space in the range of 214 to 217 passwords. A study reported that a significant portion of passwords of DAS and Pass-Go were successfully interrupted with guessing attacks using dictionaries of 232 to 242 entries, as compared to the full password space of 259 entries. Hotspots were exploited to make successful guessing attacks on Pass Points a significant portion of passwords were attacked with dictionaries of 227 to 236 entries, as compared to the full space of 244 passwords.

B. CAPTCHA

The Captcha depends on the capabilities between humans and bots in solving hard AI problems. There are two types of visual Captcha they are text Captcha and Image-Recognition Captcha (IRC).The former depends on character recognition while the latter depends upon the identification of non-character objects. The following principle has been established: text Captcha should rely on the difficulty of character division, which is computationally costly and hard .Machine recognition of non-character objects is less capable than character recognition.

i. ClickText

ClickText is a type of recognition-based CaRP scheme on text Captcha. Its alphabet comprises characters without any confusing characters. For example, Letter “I” and digit “1” will cause confusion in CaRP images. A ClickText password is a collection of characters in the alphabet, e.g., ρ =“MH$6FZ3”, which are same. A ClickText image is created by the underlying Captcha engine as if a Captcha image were created except that all the alphabet characters should be displayed in the image. During creation, each character’s location is tracked to produce for the location of the character in the generated image. The authentication server depends on the ground truth to identify the characters to user-clicked points. In ClickText images, characters can be placed on 2D space. This is different from text Captcha in which characters are ordered from right to left in order for users to type them continuously shows a ClickText image with an alphabet of 34 characters. While entering a password, the user clicks on this image and enters in the same order, for example “M”, “H”, “$”, “6”, “F”, “Z”, and then “8” , etc for password ρ = “MH$6FZ3”.

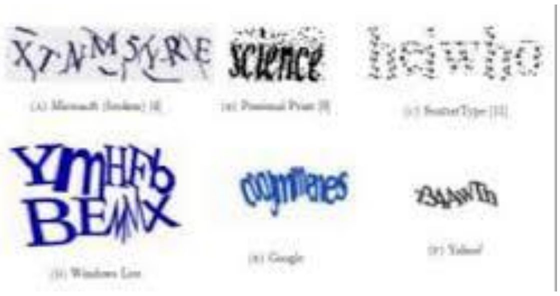

Fig. 2 clicktext image

ii. ClickAnimal

Image recognition is a Captcha scheme which makes use of 3D models of horse and dog to generate 2D animals with different colors, lightings ,textures and poses, and arranges them on the background. A user clicks all the dogs in a challenge image to clear the test. The image shows a challenge wherein all the horses are circled red. ClickAnimal is a type of recognition-based CaRP scheme on the image, with an alphabet of similar animals such as dog,elephant, horse, pig, etc. It is a set of animal names such as ρ = “Cat, Horse, Dog,….” For each animal,two or more 3D models are built. The Captcha creation process is applied to create ClickAnimal images: 3D models are used to create 2D animals by applying different views, colors, etc. The resulting 2D animals are then arranged on background such as forest. Some animals may be included by other animals in the image, but their core parts are not included in order for humans to identify them.

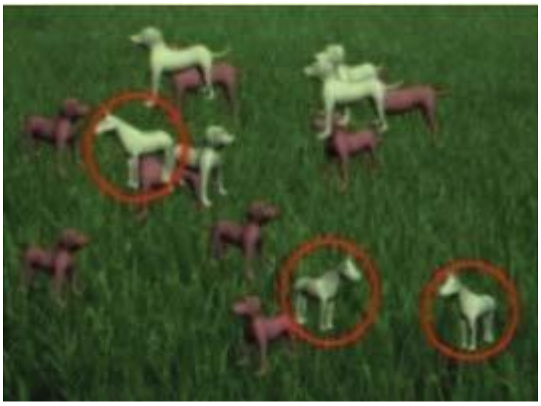

Fig. 3 A clickanimal image

IV. PROPOSED SYSTEM

In recognition-recall CaRP, a password is a set of some invariant points of an objects. An invariant point of an object (e.g. letter “A”) is a point that has a fixed position in different formation (e.g., fonts) of the object, and thus can be uniquely found by humans no matter how the object appears in CaRP images. A user must identify the objects by entering password in a CaRP image, and then use the identified objects as cues to position and click the points matching her password. Each password point has a range that a click within the range is acceptable as the password. People have a click changes of 3 pixels or less. TextPoint, a recall CaRP scheme with an alphabets, is presented , followed by a variation for challenge authentication.

A. TEXTPOINT

Characters have invariant points. In the image it shows some invariant points of letter “A”, which gives a strong cue to memorize and situate its invariant points. A point is said to be an point of an object if its distance to the nearest boundary of the object is more than a threshold. A set of internal invariant points of characters is made to form a set of clickable points for Text Points. They makes that a clickable point is included by a neighboring character and that its region overlaps with any region of a neighboring character’s clickable points on the image created by the below Captcha engine.

In finding clickable points, the distance between any pair of clickable points in a character must be more than a threshold so that they are easy to distinguishable and regions do not overlap on CaRP images. In addition, changes should also be taken into account. For example, if the center of a stroke part in one character is selected, we should omit selecting the center of a same stroke part in another character.

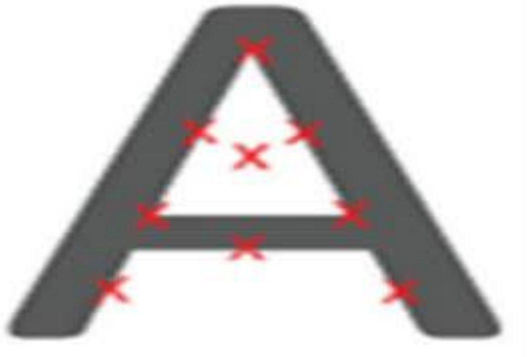

Fig. 4 Some invariant points of “A”

Some invariant points (red crosses) of “A”. a varied point from the stroke segment, e.g., a point at one-third length of the stroke part to an end. This variation in finding clickable points makes that a clickable point is context-dependent: a same structured point may or may not be a clickable point, depends on the character that the point lies in it. Character identification is required in situating clickable points on a TextPoints image although the clickable points are for each character. This is a test above a bot’s capability. A password is a subsequent of clickable points. A character can typically be multiple clickable points. Therefore TextPoints has a much big password space than ClickText.

TextPoints images look same to ClickText images and are created in the same way except that the regions of all the clickable points are checked so that none of them is included or its region overlaps other clickable point’s. We generate another image if the check expires. As such failures occur occasionally due to the fact that clickable points are all given as internal points; the restriction due to the check has a negotiation impact on the security of created images.

When generating a password, all clickable points are sited on characters in a CaRP image for a user to find. During confirmation, the user first choses characters, and clicks the password points on the characters. The authentication server places each user-clicked point on the image to obtain the closest clickable point. If distance exceeds a range, login fails or else a set of clickable points is found, and its hash value is calculated to compare with the value. It is worth differentiating potential password points between TextPoints and traditional click-based passwords such as PassPoints.

In PassPoints, points should be removed since they are always picked up by adversaries to be dictionary attacks, but points would increase the burden to remember a password. This fight does not exist in TextPoints. Clickable points in TextPoints are points of their choice and thus make a password, but cannot be found by bots since they are dynamic and contextual.

- Dynamic: regions of clickable points and their contexts (i.e., characters) are different from one image to another. The clickable points in one image are calculated independent of the clickable points in other image.

- Contextual: Whether a same structured point is a clickable point or not relaying on its context. It is only if inside the right context, i.e., at the right region of a right character given . These two features needs recognizing the correct details, i.e., characters, first. By the varying nature of Captcha, recognizing characters in a Captcha image is a task above computer’s capability. Therefore, these points of characters cannot be found mounted in dictionary attacks on TextPoints.

B. MODULES

- *Graphical Password*

In this module, The users are have authentication and security to connect the detail which is given in the Image system. Before entering or finding the details user should have the account in that or else they should register initially.

- *Captcha in Authentication*

In this module we use both Captcha and password in a user authentication protocol, which we call Captcha-based Password Authentication (CbPA) protocol, to alternate online dictionary attacks. The CbPA-protocol in requires solving a Captcha challenge after inputting a valid pair of user ID. For an invalid pair of user ID and password, the user has a probability to resolve a Captcha challenge before being denied access.

- *Overcoming Thwart Guessing Attacks*

In a guessing attack, a password guess proved in an unsuccessful trial is stated wrong and removed from subsequent trials. The number of undetermined password guesses reduces with more trials. To alternate guessing attacks, traditional approaches in creating graphical passwords plans at increasing the strong password space to create passwords harder to guess and thus needs more trials.

- *Security of Underlying Captcha*

Computational provocation is done in recognizing objects in CaRP which is basic to CaRP. Existing study on Captcha security were mainly case by case or used an near by process. No theoretic security model has been created yet. Object division is investigated as a computationally expensive and saved as CSV (Comma Separated Values) format.

V. CONCLUSION

Here we have proposed CaRP, a new security primitive based on unsolved hard AI problems. CaRP is made up of both a Captcha and a graphical password method. The main idea of CaRP is that it introduces a new graphical password, which adopts a new method to stop online guessing attacks. When one Captcha scheme is acquired, a new and more secured scheme will be created, by taking everything into account; our work is one step ahead in this paradigm of using hard AI problems for security with security, usability and practical applications. More importantly, we believe that CaRP to inspire new creations 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]The Science Behind Passfaces [Online]. Available: http://www.realuser.com/published/ScienceBehindPassfaces.pdf

[3] Jermyn, A. Mayer, F. Monrose, M. Reiter, and A. Rubin, “The design and analysis of graphical passwords,” in Proc. 8th USENIX Security Symp., 1999.

[4]H. Tao and C. Adams, “Pass-Go: A proposal toimprove the usability of graphical passwords,” Int. J. Netw. Security, vol. 7, no. 2, 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, Jul. 2005.

[6]P. C. van Oorschot and J. Thorpe, “On predictive models and user-drawn graphical passwords,” ACM Trans. Inf. Syst. Security, vol. 10, no. 4, 2008.

[7]K. Golofit, “Click passwords under investigation,” in Proc. ESORICS, 2007.

[8]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.

[9]J. Thorpe and P. C. van Oorschot, “Human-seeded attacks and exploiting hot spots in graphical passwords,” in Proc. USENIX Security, 2007.

[10]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, Sep. 2010.

[11]P. C. van Oorschot and J. Thorpe, “Exploiting predictability in click-based graphical passwords,” J. Comput. Security, vol. 19, no. 4, 2011.

[12]T. Wolverton. (2002, Mar. 26). Hackers Attack eBay Accounts [Online].Available:http://www.zdnet.co.uk/news/networking/2002/03/ 26/hackers-attack-ebay-accounts-2107350/

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

[14]B. Pinkas and T. Sander, “Securing passwords against dictionary attacks,” in Proc. ACM CCS, 2002.

[15]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, 2006.

[16]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, Jan./Feb. 2012.

[17]L. von Ahn, M. Blum, N. J. Hopper, and J. Langford, “CAPTCHA: Using hard AI problems for security,” in Proc. Eurocrypt, 2003.

[18]S. Chiasson, P. C. van Oorschot, and R. Biddle, “Graphical password authentication using cued click points,” in Proc. ESORICS, 2007.

[19]S. Chiasson, A. Forget, R. Biddle, and P. C. van Oorschot, “Influencing users towards better passwords: Persuasive cued click-points,” in Proc. Brit. HCI Group Annu. Conf. People Comput., Culture, Creativity, Interaction, vol. 1. 2008.

[20]D. Davis, F. Monrose, and M. Reiter, “On user choice in graphical password schemes,” in Proc. USENIX Security, 2004.