ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Advanced GPA Based Secured Primitive Captcha Over Hard AI Problems

K.Prasad
M.Tech. Student,
Dept of CSE,
Indur Institute of Engineering and Technology,
Telangana, India.

K.Sindhura
Assistant Professor,
Dept of CSE,
Indur Institute of Engineering and Technology,
Telangana , India.

ABSTRACT:

Many security primitives are based on hard mathematical problems. Using hard AI problems for security is emerging as an exciting new paradigm, but has been under- explored. In this paper, we present a new security primitive based on hard AI problems, namely, a novel family of graph- ical password systems built on top of Captcha technology, which we call Captcha as graphical passwords (CaRP).CaRP is both a Captcha and a graphical password scheme. CaRP addresses a number of security problems altogether, such as online guessing attacks, relay attacks, and, if combined with dual-view technologies, shoulder-surfing attacks. Notably, a CaRP password can be found only probabilistically by automatic online guessing attacks even if the password is in the search set. CaRP also offers a novel approach to address the well-known image hotspot problem in popular graphical password systems, such as PassPoints, that often leads to weak password choices. CaRP is not a panacea, but it offers reasonable security and usability and appears to fit well with some practical applications for improving online security.

Index Terms:

Graphical password, password, hotspots, CaRP, Captcha, dictionary attack, password guessing attack, security primitive.

INTRODUCTION:

Password authentication is one of the most common building blocks in implementing access control. Each user has a relatively short sequence of characters commonly referred to as a password. To gain access, the user provides his/her password to the system. Access is granted if the password is correct; it is denied otherwise. A common attack against password authenticated systems is the dictionary attack.

An attacker can write a program that, imitating a legitimate user, repeatedly tries different passwords, say from a dictionary, until it finds one that works. There are several well-known ways to cope with dictionary attacks. For example, the system can deny access for the user in question after some number of tries, a technique known as account locking. However, this invites a denial of service attack: an attacker can lock anyone out of the system by submitting a sequence of incorrect passwords on behalf of the victim. Other solutions also have their own shortcomings [1]. In this paper, we present an alternative defense against dictionary attacks.The idea is to make it harder for automated programs to mount dictionary attacks by requiring the attacking programs to pass a test that is easy for humans but is hard (in terms of accuracy and/or compute time) for computer programs.

A construct with this property is called a CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) [2]. In particular, if there exists a program that can pass a CAPTCHA with high probability, then that program can be used to solve a hard AI problem. CAPTCHAs are already in use in some systems that benefit from distinguishing between humans and "bots" [3]. Recently, they have also been suggested as a means for deterring dictionary attacks in password authenticated systems [1]. One CAPTCHA suitable for password authenticated systems displays a degraded image of a word to the human, who then responds by typing the word he or she sees. A similar CAPTCHA using a sound clip instead of an image can also be used. However, these CAPTCHAs are not suitable for systems that allow remote access via consoles or dumb terminal programs. Our goal is to make it possible for such minimal systems to obtain the same benefits from CAPTCHAassisted password authentication as systems with graphical displays and/or speakers.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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 [13]. Defense against online dictionary attacks is a more subtle problem than it might appear. Intuitive countermeasures such as throttlinglogon 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 [12]) and incurs expensive helpdesk costs for account reactivation.

2) It is vulnerable to global password attacks [14] 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. CaRP also offers protection against relay attacks, an increasing threat to bypass Captchas protection, wherein Captcha challenges are relayed to humans to solve. Koobface [33] was a relay attack to bypass Facebook's Captcha in creating new accounts. CaRP is robust to shoulder-surfing attacks if combined with dual-view technologies.

Related Work:

The first mention of ideas related to "Automated Turing Tests" seems to appear in an unpublished manuscript by MoniNaor [10]. This excellent manuscript contains some of the crucial notions and intuitions, but gives no proposal for an Automated Turing Test, nor a formal definition. The first practical example of an Automated Turing Test was the system developed by Altavista [8] to prevent "bots" from automatically registering web pages. Their system was based on the difficulty of reading slightly distorted characters and worked well in practice,but was only meant to defeat off-the-shelf Optical Character Recognition (OCR) technology. (Coates et al [5], inspired by our work, and Xu et al [14] developed similar systems and provided more concrete analyses.) In 2000 [1], we introduced the notion of a captcha as well as several practical proposals for Automated Turing Tests. This paper is the first to conduct a rigorous investigation of Automated Turing Tests and to address the issue of proving that it is difficult to write a computer program that can pass the tests. This, in turn, leads to a discussion of using AI problems for security purposes, which has never appeared in the literature. We also introduce the first Automated Turing Tests not based on the difficulty of Optical Character Recognition. A related general interest paper has been accepted by Communications of the ACM. That paper reports on our work, without formalizing the notions or providing security guarantees.

CaRP: Captcha as gRaphical Passwords:

CaRP is a family of graphical password systems created with Captcha technology. Just like PassPoints, a user clicks on a CaRP image and the sequence of her clicks creates a password. However, each CaRP image is automatically generated by a Captcha generator, and thus is also a Captcha challenge. Just like a session key, a CaRP image is never reused across different sessions. Even for the same user, a new CaRP image is needed for every login attempt. To the contrary, in PassPoints a user always uses the same image to click her password, and many users use the same image for their password input, which leads to successful attacks exploiting hotspots.

Pinkas and Sander [5] introduced a protocol to protect passwords from online dictionary attack with Captchas1. Captcha and password are separate entities in this protocol, but are intrinsically combined in CaRP, which is both a Captcha and a graphical password (scheme). The notion of CaRP is simple but generic, and it can have multiple instantiations. Many Captcha schemes, regardless of whether they are text based or image recognition based, can be converted to a CaRP scheme.

ClickText:

ClickText is a CaRP scheme built on top of text Captcha. Unlike normal text Captchas, a CaRP image should contain all the alphabet to allow a user to form any allowed password.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In ClickText images, characters can be arranged randomly on 2D space. This is another major difference from traditional text Captchas in which characters are typically ordered from left to right. Using ordinary text Captcha is not suitable in this context, as it is hard to arrange all the characters one dimensionally in a reasonably small space. Also, there is no order among characters in a CaRP image whereas the order is needed for characters in a normal Captcha image so that users can type them in. Therefore, we propose a new problem, 2D text segmentation, as the underlying hard AI problem for ClickText.

ClickAnimal:

Captcha Zoo [6] is an image recognition scheme whose security relies on both object segmentation and binary object classification. It uses 3D models of two similar animals, e.g. dog and horse, to generate 2D animals with different textures, colors, lightings and poses, and then places them on a cluttered background. A user clicks all the horses in a challenge image to pass the test.

AnimalGrid:

The number of similar animals is much less than the number of available text characters. ClickAnimal has a smaller alphabet, and thus it implies a smaller password space than ClickText does. CaRP should have a sufficiently-large effective password space to resist human guessing attacks. ClickAnimal's password space can be increased by combining a grid scheme as follows, leading to a new CaRP which we call AnimalGrid.

Two Families of captchas:

We now describe two families of captchas whose security is based on the hardness of problems in P1 and P2. Notice that if P1I,T is (δ, τ)-hard then P1I,T can be used to construct a captcha trivially: the verifier simply gives the provert and asks the prover to output i. According to our definition, this would be a perfectly valid captcha. However, it would also be a very impractical one: if [I] is large, then humans would take a long time to answer. The captchas we present in this section can be quickly answered by humans. The first family of captchas, matcha, is somewhat impractical, but the second family, pix, is very practical and in fact several instantiations of it are already in use.

MATCHA A matcha instance is described by a triple M = (I, T , τ), where I is a distribution on images and T is a distribution on image transformations that can be easily computed using current computer programs. matcha is a captcha with the following property: any program that has high success over M = (I, T) can be used to solve P1I,T . The matcha verifier starts by choosing a transformation t T . It then flips a fair unbiased coin. If the result is heads, it picks k I and sets (i, j) = (k, k). If the result is tails, it sets j I and i U([I]–{j}) where U(S) is the uniform distribution on the set S. The matcha verifier sends the prover (i,t(j)) and sets a timer to expire in time τ ; the prover responds with res {0, 1}. Informally, res = 1 means that i = j, while res = 0 means that i 6= j. If the verifier's timer expires before the prover responds, the verifier rejects. Otherwise, the verifier makes a decision based on the prover's response res and whether i is equal to j:

– If i = j and res = 1, then matcha accepts.
– If i = j and res = 0, then matcha rejects.
– If i 6= j and res = 1, then matcha rejects.
– If i 6= j and res = 0, then matcha plays another round.

An Application:
Robust Image-Based Steganography:

We detail a useful application of (δ, τ)-solutions to instantiations of P1 and P2 (other than reading slightly distorted text, which was mentioned before). We hope to convey by this application that our problems were not chosen just because they can create captchas but because they in fact have applications related to security. Our problems also serve to illustrate that there is a need for better AI in security as well. Areas such a Digital Rights Management, for instance, could benefit from better AI: a program that can find slightly distorted versions of original songs or images on the world wide web would be a very useful tool for copyright owners. There are many applications of solutions to P1 and P2 that we don't mention here. P1, for instance, is interesting in its own right and a solution for the instantiation when I is a distribution on images of works of art would benefit museum curators, who often have to answer questions such as "what painting is this a photograph of Robust Image-Based Steganography. Robust Steganography is concerned with the problem of covertly communicating messages on a public channel which is subject to modification by a restricted adversary.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

For example, Alice may have some distribution on images which she is allowed to draw from and send to Bob; she may wish to communicate additional information with these pictures, in such a way that anyone observing her communications can not detect this additional information. The situation may be complicated by an adversary who transforms all transmitted images in an effort to remove any hidden information. In this section we will show how to use (δ, τ)-solutions to instantiations of P1I,T or P2I,T ,λ to implement a secure robust steganographic protocol for image channels with distribution I, when the adversary chooses transformations from T . Note that if we require security for arbitrary I, T , we will require a (δ, τ)-solution to P1 for arbitrary I, T ; if no solution works for arbitrary (I, T) this implies the existence of specific I, T for which P1 is still hard.

Thus either our stegosystem can be implemented by-computers for arbitrary image channels or their is a (non-constructive) hard AI problem that can be used to construct a captcha. The results of this subsection can be seen as providing an implementation of the "supraliminal channel" postulated by Craver [6]. Indeed, Craver's observation that the adversary's transformations should be restricted to those which do not significantly impact human interpretation of the images (because the adversary should not unduly burden "innocent" correspondents) is what leads to the applicability of our hard AI problems.

Security Analysis:

The computational intractability of hard AI problems such as object recognition is fundamental to the security of CaRP. Existing analyses on Captcha security were mostly case by case or used an approximation approach. No theoretic security model has been established yet. Segmenting similar objects (e.g. characters) is considered as a computationally-expensive and combinatorially-hard problem [7], which modern text Captchaschemes rely on. According to [7], the complexity of object segmentation is exponentially dependent of the number of objects contained in a challenge, and polynomially dependent of the size of the Captcha alphabet. A Captcha challenge typically contains 6 to 10 characters, whereas a CaRP image typically contains 30 or more characters. Therefore, ClickText is much more secure than normal text Captcha.

Furthermore, characters in a CaRP scheme are arranged two-dimensionally, which further increases segmentation difficulty due to an additional dimension to segment. ClickAnimal relies on both object segmentation and multiple-label classification. Its security remains an open question. As a framework of graphical passwords, CaRP does not rely on the security of any specific Captcha scheme. If one Captcha scheme gets broken, a new and more robust Captcha scheme may appear and be used to construct a new CaRP scheme. CaRP offers protection against online dictionary attacks on passwords, which have been for long time a major security threat for various online services.CaRP makes it much harder for bad guys to perform automated guess attacks. Even when a human is involved, the attack is still expensive and slowed down.

CaRP also offers protection against relay attacks, which have been an increasing threat to online applications protected by Captchas. In a relay attack, Captcha challenges are relayed to humans to solve, with their answers returned. CaRP is robust to shoulder-surfing attacks, if combined with Microsoft's dualview technologies [9] that show two sets of completely different images simultaneously on the same LCD screen: one for private, and the other for public. When a CaRP image is displayed as private, attackers can capture a user's click-points but not the private image, but these points are useless for a next login session (where a new CaRP image will be used).

CaRP is robust to cross-site scripting attacks targeting at stealing users' graphical passwords, although other click-based graphical passwords such as PassPoints are vulnerable to such attacks. However, a longitudinal evaluation is needed to establish the effective password space for each CaRP instantiation. CaRP is vulnerable if a client is compromised, and the image and user-clicked points can both be captured.

RECOGNITION-RECALL CaRP:

In recognition-recall CaRP, a password is a sequence of some invariant points of objects. An invariant point of an object (e.g. letter "A") is a point that has a fixed relative position in different incarnations (e.g., fonts) of the object, and thus can be uniquely identified by humans no matter how the object appears in CaRP images.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

To enter a password, a user must identify the objects in a CaRP image, and thenuse the identified objects as cues to locate and click the invariant points matching her password. Each password point has a tolerance range that a click within the tolerance range is acceptable as the password point. Most people have a click variation of 3 pixels or less [18].TextPoint, a recognition recall CaRP scheme with an alphabet of characters,is presented next, followed by a variation for challenger-esponseauthentication.

TextPoints:

Charactrrs contain invariant points some invariant points of letter "A", which offers a strong cue to memorize and locate its invariant points. A point is said to be an internal point of an object if its distance to the closest boundary of the object exceeds a threshold. A set of internal invariant points of characters is selected to form a set of clickable points for TextPoints. The internality ensures that a clickable point is unlikely occluded by a neighboringcharacter and that its tolerance region unlikely overlaps with any tolerance region of a neighboring character's clickable points on the image generated by the underlying Captchaengine. In determining clickable points, the distance between any pair of clickable points in a character must exceed a threshold so that they are perceptually distinguishable and their tolerance regions do not overlap on CaRP images.

In addition, variation should also be taken into consideration. For example, if the center of a stroke segment in one character is selected, we should avoid selecting the center of a similar stroke segment in another character. Instead, we should selecta different point from the stroke segment, e.g., a point atone-third length of the stroke segment to an end. This variation in selecting clickable points ensures that a clickable point is context-dependent: a similarly structured point may or may not be a clickable point, depending on the character that the point lies in.

Character recognition is required in locating clickable points on a TextPoints image although the clickable points are known for each character. This is a task beyond a bot's capability. A password is a sequence of clickable points. A character can typically contribute multiple clickable points. Therefore TextPoints has a much larger password space than ClickText.

Image Generation:

TextPoints images look identical to ClickText images and are generated in the same way except that the locations of all the clickable points are checked tom ensure that none of them is occluded or its tolerance region overlaps another clickable point's. We simply generate anotherimage if the check fails. As such failures occur rarely due to the fact that clickable points are all internal points, the restriction due to the check has a negligible impact on the security of generated images.

Authentication:

When creating a password, all clickable points are marked on corresponding characters in a CaRPimage for a user to select. During authentication, the user first identifies her chosen characters, and clicks the password points on the right characters. The authentication server maps each user-clicked point on the image to find the closest clickable point. If their distance exceeds a tolerable range, login fails. Otherwise a sequence of clickable points is recovered, and its hash value is computed to compare with the stored value. It is worth comparing potential password points between TextPoints and traditional click-based graphical passwords such as PassPoints [5]. In PassPoints, salient points should be avoided since they are readily picked up by adversaries to mount dictionary attacks, but avoiding salient points would increase the burden to remember a password. This conflict does not exist in TextPoints. Clickable points in TextPointsare salient points of their characters and thus help remember a password, but cannot be exploited by bots since they are both dynamic (as compared to static points in traditional graphical password schemes) and contextual.

TextPoints4CR:

For the CaRP schemes presented up to now, the coordinates of user-clicked points are sent directly to the authentication server during authentication. For more complex protocols, say a challenge-response authentication protocol, a response is sent to the authentication server instead. TextPoints can be modified to fit challengeresponse authentication. This variation is called TextPoints for Challenge-Response or TextPoints4CR.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Unlike TextPoints wherein the authentication server stores a salt and a password hash value for each account, the server in TextPoints4CR stores the password for each account. Another difference is that each character appears only once in a TextPoints4CR image but may appear multiple times in a TextPoints image. This is because both server and client in TextPoints4CR should generate the same sequence of discretized grid-cells independently. That requires a unique way to generate the sequence from the shared secret, i.e., password. Repeated characters would lead to several possible sequences for the same password. This unique sequence is used as if the shared secret in a conventional challengeresponse authentication protocol.

Advanced Mechanisms:

The CbPA-protocols described in Section II-C require a user to solve a Captcha challenge in addition to inputting a password under certain conditions. For example, the scheme described in [16] 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. This technique can be integrated into CaRP to enhance usability:

1. A regular CaRP image is applied when an account has reached a threshold of failed login attempts. As in [16], different thresholds are applied for logins from known and unknown machines.

2. Otherwise an "easy" CaRP image is applied. An "easy" CaRP image may take several forms depending on the application requirements. It can be an image generated by the underlying Captcha generator with less distortion or overlapping, a permuted "keypad" wherein undistorted visual objects (e.g. characters) are permuted, or even a regular "keypad" wherein each visual object (e.g., character) is always located at a fixed position. These different forms of "easy"CaRP images allow a system to adjust the level of difficulty to fit its needs. With such a modified CaRP, a user would always enter a password on an image for both cases listed above. No extra task is required. The only difference between the two cases is that a hard image is used in the first case whereas an easy image is used in the second case.

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 AnimalGrid and ClickText easier to use than PassPoints and a combination of text password and Captcha. Both AnimalGrid and ClickText had better password memorability 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.

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

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[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. 273–292, 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] 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.

[7] K. Golofit, "Click passwords under investigation," in Proc. ESORICS, 2007, pp. 343–358.

.

[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, pp. 20–28.

[9] J. Thorpe and P. C. van Oorschot, "Human-seeded attacks and exploiting hot spots in graphical passwords," in Proc. USENIX Security, 2007, pp. 103–118.

[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, pp. 393–405, Sep. 2010.

[11] P. C. van Oorschot and J. Thorpe, "Exploiting predictability in clickbased graphical passwords," J. Comput. Security, vol. 19, no. 4, pp. 669–702, 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/