Volume 5, Issue 4, 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A New Security Primitive Based on Carp Using Hard Ai Problems

P. Kalaivizhi
M.Tech (CSE),
Prist University, Puducherry, India

V. Udhayakumar
Professor & Head, Dept. of CSE,
Prist University, Puducherry, India

Abstract*— Many security primitives are based on hard mathematical problems. In this project, we present a new security primitive based on hard AI problems, namely, a novel family of graphical 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. 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. 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.**

I. INTRODUCTION

FUNDAMENTAL task in security is to create cryptographic primitives based on hard mathematical problems that are computationally intractable. For example, the problem of integer factorization is fundamental to the RSA public-key cryptosystem and the Rabin encryption. The discrete logarithm problem is fundamental to the ElGamal encryption, the Diffie-Hellman key exchange, the Digital Signature Algorithm, the elliptic curve cryptography and so on. Using hard AI (Artificial Intelligence) problems for security, initially proposed, 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. 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 [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 requires solving a Captcha challenge in every login. This impact on usability can be mitigated by adapting the CaRP image's difficulty level based on the login history of the account and the machine used to log in. Typical application scenarios for CaRP include:

1) CaRP can be applied on touch-screen devices whereon typing passwords is cumbersome, esp. for secure Internet applications such as e-banks. Many e-banking systems have applied Captchas in user logins [39]. For example, ICBC (www.icbc.com.cn), the largest bank in the world, requires solving a Captcha challenge for every online login attempt.

2) CaRP increases spammer's operating cost and thus helps reduce spam emails. For an email service provider that deploys CaRP, a spam bot cannot log into an email account even if it knows the password. Instead, human involvement

is compulsory to access an account. If CaRP is combined with a policy to throttle the number of emails sent to new recipients per login session, a spam bot can send only a limited number of emails before asking human assistance for login, leading to reduced outbound spam traffic.

II. RELATED WORK

A. Graphical Passwords

A large number of graphical password schemes have been proposed. They can be classified into three categories according to the task involved in memorizing and entering passwords: recognition, recall, and cued recall. Each type will be briefly described here. More can be found in a recent review of graphical passwords [1].

A *recognition-based* scheme requires identifying among decoys the visual objects belonging to a password portfolio. A typical scheme is Pass faces [2] wherein a user selects a portfolio of faces from a database in creating a password. During authentication, a panel of candidate faces is presented for the user to select the face belonging to her portfolio. This process is repeated several rounds, each round with a different panel. A successful login requires correct selection in each round. The set of images in a panel remains the same between logins, but their locations are permuted.

Story [20] is similar to Pass faces but the images in the portfolio are ordered, and a user must identify her portfolio images in the correct order. Déjà Vu [21] is also similar but uses a large set of computer generated "random-art" images. Cognitive Authentication [22] requires a user to generate a path through a panel of images as follows: starting from the top-left image, moving down if the image is in her portfolio, or right otherwise. The user identifies among decoys the row or column label that the path ends.

B. Captcha

Captcha relies on the gap of capabilities between humans and bots in solving certain hard AI problems. There are two types of visual Captcha: text Captcha and Image-Recognition
Captcha (IRC). The former relies on character recognition while the latter relies on recognition of non-character objects. Security of text Captchas has been extensively studied [26]–[30]. The following principle has been established: text Captcha should rely on the difficulty of character segmentation, which is computationally expensive and combinatorially hard [30].

Machine recognition of non-character objects is far less capable than character recognition. IRCs rely on the difficulty of object identification or classification, possibly combined with the difficulty of object segmentation. Asirra [31] relies on binary object classification: a user is asked to identify all the cats from a panel of 12 images of cats and dogs. Security of IRCs has also been studied. Asirra was found to be susceptible to machine-learning attacks [24]. IRCs based on binary object classification or identification of one concrete type of objects are likely insecure [25]. Multi-label classification problems are considered much harder than binary classification problems.

Captcha can be circumvented through relay attacks whereby Captcha challenges are relayed to human solvers, whose answers are fed back to the targeted application.

C. Captcha in Authentication

It was introduced in [14] to use both Captcha and password in a user authentication protocol, which we call

Captcha-based Password Authentication (CbPA) protocol, to counter online dictionary attacks. The CbPA-protocol in [14] requires solving a Captcha challenge after inputting a valid pair of user ID and password unless a valid browser cookie is received. For an invalid pair of user ID and password, the user has a certain probability to solve a Captcha challenge before being denied access. An improved CbPA-protocol is proposed in [15] by storing cookies only on user-trusted machines and applying a Captcha challenge only when the number of failed login attempts for the account has exceeded a threshold. It is further improved in [16] by applying a small threshold for failed login attempts from unknown machines but a large threshold for failed attempts from known machines with a previous successful login within a given time frame.

III. RECOGNITION-BASED CaRP

For this type of CaRP, a password is a sequence of visual objects in the alphabet. Per view of traditional recognition based graphical passwords, recognition-based CaRP seems to have access to an infinite number of different visual objects. We present two recognition-based CaRP schemes and a variation next.

A. ClickText

ClickText is a recognition-based CaRP scheme built on top of text Captcha. Its alphabet comprises characters without any visually-confusing characters. For example, Letter "O" and digit "0" may cause confusion in CaRP images, and thus one character should be excluded from the alphabet. A ClickText password is a sequence of characters in the alphabet, e.g., ρ ="AB#9CD87", which is similar to a text password. A ClickText image is generated by the underlying Captcha engine as if a Captcha image were generated except that all the alphabet characters should appear in the image. During generation, each character's location is tracked to produce ground truth for the location of the character in the generated image. The authentication server relies on the ground truth to identify the characters corresponding to user-clicked points.

B. ClickAnimal

Captcha Zoo [32] is a Captcha scheme which uses 3D models of horse and dog to generate 2D animals with different textures, colors, lightings and poses, and arranges them on a cluttered background. A user clicks all the horses in a challenge image to pass the test. Fig. 3 shows a sample challenge wherein all the horses are circled red.

ClickAnimal is a recognition-based CaRP scheme built on top of Captcha Zoo [32], with an alphabet of similar animals such as dog, horse, pig, etc. Its password is a sequence of animal names such as ρ = "Turkey, Cat, Horse, Dog" For each animal, one or more 3D models are built. The Captcha generation process is applied to generate ClickAnimal images: 3D models are used to generate 2D animals by applying different views, textures, colors, lightning effects, and optionally distortions. The resulting 2D animals are then arranged on a cluttered background such as grassland. Some

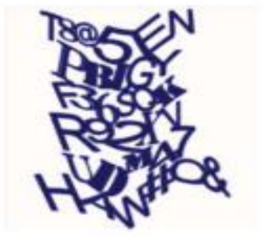

Fig. 2. A ClickText image with 33 characters.

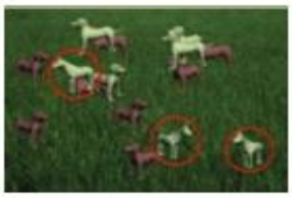

Fig. 3. Captcha Zoo with horses circled red.

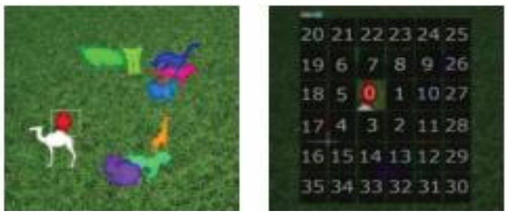

Fig.4. A Click Animal image (left) and 6 × 6 grid (right) determined by red turkey's bounding rectangle.

animals may be occluded by other animals in the image, but their core parts are not occluded in order for humans to identify each of them. Fig. 4 shows a ClickAnimal image with an alphabet of 10 animals.

C. AnimalGrid

The number of similar animals is much less than the number of available characters. ClickAnimal has a smaller alphabet, and thus a smaller password space, than ClickText. CaRP should have a sufficiently-large effective password space to resist human guessing attacks. AnimalGrid's password space can be increased by combining it with a grid-based graphical password, with the grid depending on the size of the selected animal.

DAS [3] is a candidate but requires drawing on the grid. To be consistent with ClickAnimal, we change from drawing to clicking: *Click-A-Secret (CAS)* wherein a user clicks the grid cells in her password. *AnimalGrid* is a combination of ClickAnimal and CAS. The number of grid-cells in a grid should be much larger than the alphabet size. Unlike DAS, grids in our CAS are object-dependent, as we will see next. It has the advantage that a correct animal should be clicked in order for the clicked grid-cell(s) on the follow-up grid to be correct. If a wrong animal is clicked, the follow-up grid is wrong. A click on the correctly labeled grid-cell of the wrong grid would likely produce a wrong grid-cell at the authentication server side when the correct grid is used.

IV. 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. To enter a password, a user must identify the objects in a CaRP image, and then use 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 challenge response authentication.

A. TextPoints

Characters contain invariant points. Fig. 5 shows 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 neighboring character 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 Captcha engine. 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.

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 to ensure that none of them is occluded or its tolerance region overlaps another clickable point's. We simply generate another image 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 CaRP image 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.

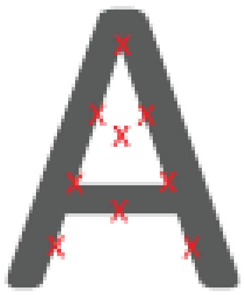

Fig.5. some invariant points (red crosses) of "A".

Clickable points in TextPoints are 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:

- **Dynamic:** locations of clickable points and their contexts (i.e., characters) vary from one image to another. The clickable points in one image are computationally independent of the clickable points in another image, as we will see in Section VI-B.
- **Contextual:** Whether a similarly structured point is a clickable point or not depends on its context. It is only if within the right context, i.e., at the right location of a right character.

B. 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 challenge-response authentication. This variation is called TextPoints for Challenge-Response or TextPoints4CR.

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 challenge response authentication protocol.

Image Generation To generate a TextPoints4CR image, the same procedure to generate a TextPoints image is applied. Then the following procedure is applied to make every clickable point at least τ distance from the edges of the grid-cell it lies in. All the clickable points, denoted as set _, are located on the image. For every point in _, we calculate its distance along x-axis or y-axis to the center of the grid-cell it lies in. A point is said to be an *internal point* if the distance is less than $0.5\mu-\tau$ along both directions; otherwise a boundary point.

Authentication In entering a password, a user-clicked point is replaced by the grid-cell it lies in. If click errors are within τ , each user-clicked point falls into the same grid-cell as the original password point. Therefore the sequence of grid-cells generated from user-clicked points is identical to the one that the authentication server generates from the stored password of the account. This sequence is used as if the shared secret between the two parties in a challenge-response authentication protocol.

V. PASS POINTS MODULE

A. Pass Points Module:

Based on Blonder's original idea, Pass Points (PP) is a click-based graphical password system where a password consists of an ordered sequence of five click-points on a pixel-based image shown in fig 6. To log in, a user must click within some system-defined tolerance region for each click-point. The image acts as a cue to help users remember their password click-points.
The images could be provided by the system or chosen by the user. The only practical requirement is that the image be intricate and rich enough so that many possible click points are available. Another source of flexibility is that we do not need artificial predefined click regions with well-marked boundaries. A user's password consists of any arbitrarily chosen sequence of points in the image. Since an intricate image easily has hundreds of memorable points, not many click points are needed to make a password hard to guess. For example, with five or six click points one can make more passwords than 8-character Unix-style alphanumeric passwords over a standard 64-character alphabet. In order to log in, the user has to click close to the chosen click points, within some (adjustable) tolerance distance, e.g. within .25 cm from the actual click point.

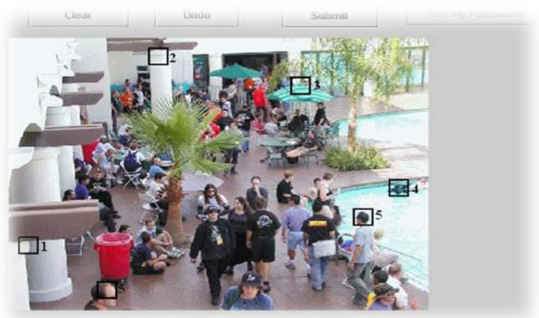

Fig. 6 Pass Point

B. Cued Click Points:

Cued Click Points (CCP) was developed as an alternative click based graphical password scheme where users select one point per image for five images. The interface displays only one image at a time; the image is replaced by the next image as soon as a user selects a click point. The system determines the next image to display based on the user's click-point on the current image. The next image displayed to users is based on a deterministic function of the point which is currently selected.
It now presents a one to-one cued recall scenario where each image triggers the user's memory of the one click-point on that image. Secondly, if a user enters an incorrect click-point during login, the next image displayed will also be incorrect.
Legitimate users who see an unrecognized image know that they made an error with their previous click-point. Conversely, this implicit feedback is not helpful to an attacker who does not know the expected sequence of images.

C. Persuasive Cued Click- Points Module:

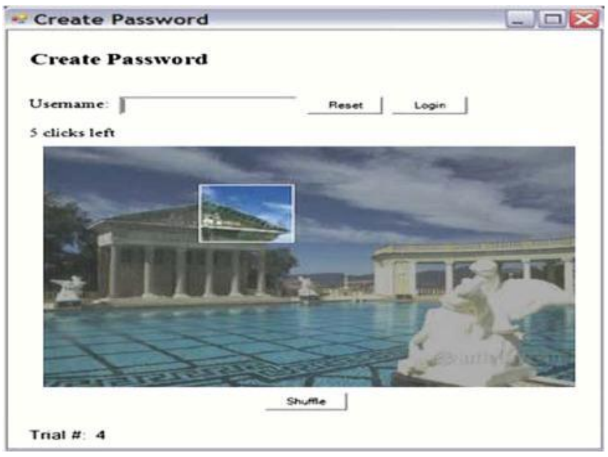

Fig.7 PCCP

To address the issue of hotspots, Persuasive Cued Click Points (PCCP) was proposed. As with CCP, a password consists of five click points, one on each of five images. During password creation, most of the image is dimmed except for a small view port area that is randomly positioned on the image.
Users must select a click-point within the view port. If they are unable or unwilling to select a point in the current view port, they may press the Shuffle button to randomly reposition the view port. The view port guides users to select more random passwords that are less likely to include hotspots. A user who is determined to reach a certain click-point may still shuffle until the view port moves to the specific location, but this is a time consuming and more tedious process.

VI. DISCUSSION

A. Security of Underlying Captcha

As a framework of graphical passwords, CaRP does not rely on 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. In the remaining security analysis, we assume that it is intractable for computers to recognize any objects in any challenge image generated by the underlying Captcha of CaRP. More accurately, the Captcha is assumed to be *chosen-pixel attack (CPA)*-secure defined with the following experiment: an adversary *A* first learns from an arbitrary number of challenge images by querying a groundtruth oracle *O* as follows: *A* selects an arbitrary number of internal object-points and sends to *O*, which responds with the object that each point lies in. Then *A* receives a new challenge image and selects an internal object-point to query *O* again.

B. Automatic Online Guessing Attacks

In automatic online guessing attacks, the trial and error process is executed automatically whereas dictionaries can be constructed manually. If we ignore negligible probabilities,
CaRP with underlying CPA-secure Captcha has the following properties:
1. Internal object-points on one CaRP image are
Computationally - independent of internal object-points on another CaRP image. Particularly, clickable points on one image are computationally-independent of clickable points on another image.

C. Human Guessing Attacks

In human guessing attacks, humans are used to enter passwords in the trial and error process. Humans are much slower than computers in mounting guessing attacks. For 8-character passwords, the theoretical password space is 338 ≈ 240 for ClickText with an alphabet of 33 characters, 108 ≈ 226 for ClickAnimal with an alphabet of 10 animals, and 10 × 467 ≈ 242 for AnimalGrid with the setting as ClickAnimal plus 6×6 grids. If we assume that 1000 people are employed to work 8 hours per day without any stop in a human guessing attack, and that each person takes 30 seconds to finish one trial. It would take them on average 0.5 · 338 · 30/ (3600 · 8 · 1000 · 365) ≈ 2007 years to break a ClickText password, 0.5 · 108 · 30*/(*3600 · 8 · 1000*)* ≈ 52 days to break a ClickAnimal password, or 0.5 · 10 · 467 · 30*/(*3600 · 8 · 1000 · 365*)* ≈ 6219 years to break an AnimalGrid password. Human guessing attacks on TextPoints require a much longer time than those on ClickText since TextPoints has a much larger password space.

D. Relay Attacks

Relay attacks may be executed in several ways. Captcha challenges can be relayed to a high-volume Website hacked or controlled by adversaries to have human surfers solve the challenges in order to continue surfing the Website, or relayed to sweatshops where humans are hired to solve Captcha challenges for small payments. Is CaRP vulnerable to relay attacks? We make the same assumption as Van Oorschot and Stubblebine [15] in discussing CbPA-protocol's robustness to relay attacks: a person will not deliberately participate in relay attacks unless paid for the task. The task to perform and the image used in CaRP are very different from those used to solve a Captcha challenge. This noticeable difference makes it hard for a person to mistakenly help test a password guess by attempting to solve a Captcha challenge. Therefore it would be unlikely to get a large number of unwitting people to mount human guessing attacks on CaRP. In addition, human input obtained by performing a Captcha task on a CaRP image is useless for testing a password guess.

VII. CONCLUSION

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. 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 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. 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] G. Wolberg, “2-pass mesh warping,” in *Digital Image Warping*. Hoboken, NJ, USA: Wiley, 1990.

[7] HP TippingPoint DVLabs, New York, NY, USA. (2011). *The Mid-Year Top Cyber Security Risks Report* [Online]. Available:http :// h20195. www2. hp. Com /v2 /GetPDF .aspx/4AA3-7045ENW.pdf

[8] 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.

[9] S. Li, S. A. H. Shah, M. A. U. Khan, S. A. Khayam, A.-R. Sadeghi, and R. Schmitz, “Breaking e-banking CAPTCHAs,” in *Proc. ACSAC*, 2010, pp. 1–10.

[10] H. Gao, X. Liu, S.Wang, and R. Dai, “A new graphical password scheme against spyware by using CAPTCHA,” in *Proc. Symp. Usable Privacy Security*, 2009, pp. 760–767.

[11] Sonia Chiasson, P.C. van Oorschot, and Robert Biddle, “Graphical Password Authentication Using Cued Click Points” ESORICS , LNCS 4734, pp.359- 74,Springer- Verlag Berlin Heidelberg 2007.

[12] Manu Kumar, Tal Garfinkel, Dan Boneh and Terry Winograd, “Reducing Shoulder-surfing by Using Gazebased Password Entry”, Symposium On Usable Privacy and Security (SOUPS) , July 18-20, 2007, Pittsburgh,PA, USA.

[13] Zhi Li, Qibin Sun, Yong Lian, and D. D. Giusto, „An association-based graphical password design resistant to shoulder surfing attack‟, International Conference on Multimedia and Expo (ICME), IEEE.2005

[14] R. Dhamija and A. Perrig, "Deja Vu: A User Study Using Images for Authentication," in *Proceedings of9th USENIX Security Symposium*, 2000.

[15] S. Akula and V. Devisetty, "Image Based Registration and Authentication System," in *Proceedings ofMidwes Instruction and Computing Symposium*, 2004.

[16] L. Sobrado and J.-C. Birget, "Graphical passwords," *The Rutgers Scholar, An Electronic Bulletin forUndergraduate Research*, vol. 4, 2002.

[17] Sonia Chiasson, Alain Forget , Robert Biddle, P. C. van Oorschot, “User interface design affects security: patterns in click-based graphical passwords”, Springer-Verlag 2009.

[18] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A.D. Rubin, "The Design and Analysis of Graphical Passwords," in *Proceedings of the 8th USENIXSecurity Symposium*, 1999.

[19] S. Man, D. Hong, and M. Mathews, "A shoulder surfing resistant graphical password scheme," in *Proceedingsof International conference on security andmanagement*. Las Vegas, NV, 2003.

[20] A. Adams and M. A. Sasse, "Users are not the enemy: why users compromise computer security mechanisms and how to take remedial measures," *Communicationsof the ACM*, vol. 42, pp. 41-46, 1999.