

Click-Based Graphical Password Schemes to Prevent Access against Spyware



Velugoti.Nadiya

Dept of Software Engineering,
SKR College of Engineering and Technology,
Nh-5, Kondurusatram, Manubolu, Spsr Nellore,Ap.



Syed Baji, Ph.D

Associate Professor,
Dept of Software Engineering,
SKR College of Engineering and Technology,
Nh-5, Kondurusatram, Manubolu, Spsr Nellore,Ap.

ABSTRACT :

Click-based graphical password schemes require a user to click on a set of points on one or more presented background images. With Pass Points, users create a password by clicking five ordered points anywhere on the given image. To log in, users must correctly repeat the sequence of clicks, with each click falling within an acceptable tolerance of the original point. To implement this aspect, along with a scheme converting the user-entered graphical password into a cryptographic verification key, a “robust discretization” scheme. It consisted of three overlapping grids (invisible to the user) used to determine whether the click-points of a login attempt were close enough to the original points to be accepted.

1.INTRODUCTION:

AFUNDAMENTAL 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 El-Gamal 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 in [17], is an exciting new paradigm. Under this paradigm, the most notable primitive invented is Captcha, which distinguishes human users from computers by presenting a challenge, i.e., a puzzle, beyond the capability of computers but easy for humans. Captcha is now a standard Internet security technique to protect online email and other services from being abused by bots.

However, this new paradigm has achieved just a limited success as compared with the cryptographic primitives based on hard math problems and their wide applications. Is it possible to create any new security primitive based on hard AI problems? This is a challenging and interesting open problem. In this paper, we introduce a new security primitive based on hard AI problems, namely, a novel family of graphical password systems integrating Captcha technology, which we call CaRP (Captcha as gRaphical Passwords). CaRP is click-based graphical passwords, where a sequence of clicks on an image is used to derive a password. Unlike other click-based graphical passwords, images used in CaRP are Captcha challenges, and a new CaRP image is generated for every login attempt. The notion of CaRP is simple but generic. CaRP can have multiple instantiations. In theory, any Captcha scheme relying on multiple-object classification can be converted to a CaRP scheme. We present exemplary CaRPs built on both text Captcha and image recognition Captcha. One of them is a text CaRP wherein a password is a sequence of characters like a text password, but entered by clicking the right character sequence on CaRP images. CaRP offers protection against online dictionary attacks on passwords, which have been for long time a major security threat for various online services. This threat is widespread and considered as a top cyber security risk [13]. Defense against online dictionary attacks is a more subtle problem than it might appear. Intuitive countermeasures such as throttling logon attempts do not work well for two reasons: 1) It causes denial-of-service attacks (which were exploited to lock highest bidders out in final minutes of eBay auctions [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,



International Journal of Research in Advanced Computer Science Engineering

A Peer Reviewed Open Access International Journal

www.ijracse.com

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. Koob-face [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. 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. The remaining paper is organized as follows: Background and related work are presented in Section II. We outline CaRP in Section III, and present a variety of CaRP schemes in Sections IV and V. Security analysis is provided in Section VI. A usability study on two CaRP schemes that we have implemented is reported in Section VII. Balance of security and usability is discussed in Section VIII. We conclude the paper with Section IX.

II. BACKGROUND AND 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. This process is repeated, each time with a different panel. A successful login requires that the cumulative probability that correct answers were not entered by chance exceeds a threshold within a given number of rounds. A recall-based scheme requires a user to regenerate the same interaction result without cueing. Draw-A-Secret (DAS) [3] was the first recall-based scheme proposed. A user draws her password on a 2D grid. The system encodes the sequence of grid cells along the drawing path as a userdrawn password. Pass-Go [4] improves DAS's usability by encoding the grid intersection points rather than the grid cells. BDAS [23] adds background images to DAS to encourage users to create more complex passwords.

In a cued-recall scheme, an external cue is provided to help memorize and enter a password. Pass Points [5] is a widely studied click-based cued-recall scheme wherein a user clicks a sequence of points anywhere on an image in creating a password, and re-clicks the same sequence during authentication. Cued Click Points (CCP) [18] is similar to Pass Points but uses one image per click, with the next image selected by a deterministic function. Persuasive Cued Click Points (PCCP) [19] extends CCP by requiring a user to select a point inside a randomly positioned viewport when creating a password, resulting in more randomly distributed click-points in a password.



International Journal of Research in Advanced Computer Science Engineering

A Peer Reviewed Open Access International Journal

www.ijracse.com

Among the three types, recognition is considered the easiest for human memory whereas pure recall is the hardest [1]. Recognition is typically the weakest in resisting guessing attacks. Many proposed recognition-based schemes practically have a password space in the range of 2^{13} to 2^{16} passwords [1]. A study [6] reported that a significant portion of passwords of DAS and Pass-Go [4] were successfully broken with guessing attacks using dictionaries of 231 to 241 entries, as compared to the full password space of 258 entries. Images contain hotspots [7], [8], i.e., spots likely selected in creating passwords. Hotspots were exploited to mount successful guessing attacks on PassPoints [8]–[11]: a significant portion of passwords were broken with dictionaries of 226 to 235 entries, as compared to the full space of 243 passwords.

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 on-line 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.

Captcha was also used with recognition-based graphical passwords to address spyware [40], [41], wherein a text Captcha is displayed below each image; a user locates her own pass-images from decoy images, and enters the characters at specific locations of the Captcha below each pass-image as her password during authentication. These specific locations were selected for each pass-image during password creation as a part of the password. In the above schemes, Captcha is an independent entity, used together with a text or graphical password. On the contrary, a CaRP is both a Captcha and a graphical password scheme, which are intrinsically combined into a single entity.

D. Other Related Work:

Captcha is used to protect sensitive user inputs on an untrusted client [35]. This scheme protects the communication channel between user and Web server from keyloggers and spyware, while CaRP is a family of graphical password schemes for user authentication. The paper [35] did not introduce the notion of CaRP or explore its rich properties and the design space of a variety of CaRP instantiations.

i) EXISTING SYSTEM:

Previously used RSA algorithm which distinguishes human users from computers by presenting a challenge, i.e., a puzzle, beyond the capability of computers but easy for humans. Captcha address the well-known image hotspot problem in popular graphical password systems, such as PassPoints, that often leads to weak password choices.

DISADVANTAGES OF EXISTING SYSTEM:

This RSA PUBLIC KEY CRYPTOSYSTEM AND RABIN ENCRYPTION paradigm has achieved just a limited success as compared with the cryptographic primitives based on hard math problems and their wide applications. RSA PUBLIC key Cryptosystem providing less security and also somewhat complicated.

ii) PROPOSED SYSTEM:

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.

ADVANTAGES OF PROPOSED SYSTEM:

CaRP offers protection against online dictionary attacks on passwords, which have been for long time a major security threat for various online services. CaRP also offers protection against relay attacks, an increasing threat to bypass Captchas protection.

IV. 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., $\rho = "AB\#9CD87"$, 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. In ClickText images, characters can be arranged randomly on 2D space. This is different from text Captcha challenges in which characters are typically ordered from left to right in order for users to type them sequentially. Fig. 2 shows a ClickText image with an alphabet of 33 characters. In entering a password, the user clicks on this image the characters in her password, in the same order, for example "A", "B", "#", "9", "C", "D", "8", and then "7" for password $\rho = "AB\#9CD87"$.



Fig. 2. A ClickText image with 33 characters.

B. Click Animal 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 $\rho = "Turkey, Cat, Horse, Dog, \dots"$



Fig. 3. Captcha Zoo with horses circled red.

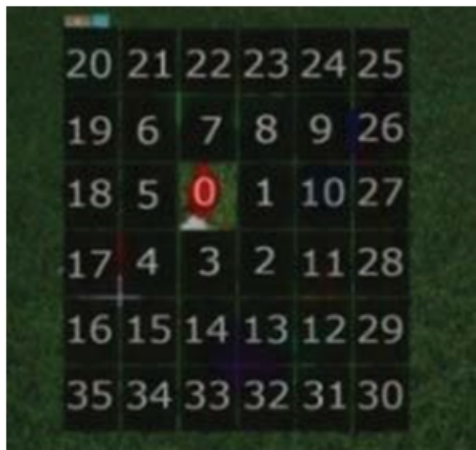


Fig. 4. A ClickAnimal image (left) and 6 × 6 grid (right) determined by red turkey’s bounding rectangle.

For each animal, one or more 3D models are built. The Captcha generation process is applied to generate Click-Animal images: 3D models are used to generate 2D animals by applying different views, textures, colors, lighting effects, and optionally distortions. The resulting 2D animals are then arranged on a cluttered background such as grassland. Some 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. Note that different views applied in mapping 3D models to 2D animals, together with occlusion in the following step, produce many different shapes for the same animal’s instantiations in the generated images. Combined with the additional anti-recognition mechanisms applied in the mapping step, these make it hard for computers to recognize animals in the generated image, yet humans can easily identify different instantiations of 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. To enter a password, a ClickAnimal image is displayed first. After an animal is selected, an image of $n \times n$ grid appears, with the grid-cell size equaling the bounding rectangle of the selected animal. Each grid-cell is labeled to help users identify. Fig. 4 shows a 6×6 grid when the red turkey in the left image of Fig. 4 was selected. A user can select zero to multiple grid-cells matching her password. Therefore a password is a sequence of animals interleaving with grid cells, e.g., $\rho = \text{“Dog, Grid_2_, Grid_1_}; \text{Cat, Horse, Grid_3_”}$, where Grid_1_ means the grid-cell indexed as 1, and grid-cells after an animal means that the grid is determined by the bounding rectangle of the animal. A password must begin with an animal. When a ClickAnimal image appears, the user clicks the animal on the image that matches the first animal in her password. The coordinates of the clicked point are recorded. The bounding rectangle of the clicked animal is then found interactively as follows: a bounding rectangle is calculated and displayed, e.g., the white rectangle shown in Fig. 4. The user checks the displayed rectangle and corrects inaccurate edges by dragging if needed. This process is repeated until the user is satisfied with the accuracy of the bounding rectangle. In most cases, the calculated bounding rectangle is accurate enough without needing manual correction. Once the bounding rectangle of the selected animal is identified, an image of $n \times n$ grid with the identified bounding rectangle as its grid-cell size is generated and displayed.

International Journal of Research in Advanced Computer Science Engineering

A Peer Reviewed Open Access International Journal

www.ijracse.com

If the grid image is too large or too small for a user to view, the grid image is scaled to a fitting size. The user then clicks a sequence of zero to multiple grid-cells that match the grid896 cells following the first animals in her password, and then gets back to the ClickAnimal image. For the example password ρ given previously, she clicks a point inside grid-cell₂, and then a point inside grid-cell₁ to select the two grid-cells. The coordinates of user-clicked points on the grid image (the original one before scaling if the grid image is scaled) are recorded. The above process is repeated until the user has finished entering her password. The resulting sequence of coordinates of user-clicked points, e.g., “AP_{150,50}, GP_{30,66}, GP_{89,160}, AP_{135,97},...” where “AP_{x,y}” denotes the point with coordinates _{x,y} on a ClickAnimal image, and “GP_{x,y}” denotes the point with coordinates _{x,y} on a grid image, is sent to the authentication server. Using the ground truth, the server recovers the first animal from the received sequence, regenerates the grid image from the animal’s bounding rectangle, and recovers the clicked grid-cells. This process is repeated to recover the password the user clicked. Its hash is then calculated and compared with the stored hash.

V. 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. 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 select



Fig. 5. Some invariant points (red crosses) of “A”.

a different point from the stroke segment, e.g., a point at one-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 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.

International Journal of Research in Advanced Computer Science Engineering

A Peer Reviewed Open Access International Journal

www.ijracse.com

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. 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 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.

These two features require recognizing the correct contexts, i.e., characters, first. By the very nature of Captcha, recognizing characters in a Captcha image is a task beyond computer's capability. Therefore, these salient points of characters cannot be exploited to mount dictionary attacks on TextPoints.

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. In TextPoints4CR, an image is partitioned into a fixed grid with the discretization grid-cell of size μ along both directions. The minimal distance between any pair of clickable points should be larger than μ by a margin exceeding a threshold to prevent two clickable points from falling into a single grid-cell in an image. Suppose that a guaranteed tolerance of click errors along both x-axis and y-axis is τ , we require that $\mu \geq 4\tau$.

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. For each boundary point in $_$, a nearby internal point in the same grid-cell is selected. The selected point is called a target point of the boundary point. After processing all the points in $_$, we obtain a new set $___$ comprising internal points; these are either internal clickable points or target points of boundary clickable points. Mesh warping [36], widely used in generating text Captcha challenges, is then used to warp the image so that $_$ maps to $___$. The result is a TextPoint4-CR image wherein every clickable point would tolerate at least τ of click errors. Selection of target points should try to reduce warping distortion caused by mapping $_$ to $___$. Authentication. In entering a password, a user-clicked point is replaced by the grid-cell it lies in. If click errors are within τ ,



International Journal of Research in Advanced Computer Science Engineering

A Peer Reviewed Open Access International Journal

www.ijracse.com

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. Unlike other CaRP schemes presented in this paper, Text-Points4CR requires the authentication server to store passwords instead of their hash values. Stored passwords must be protected from insider attacks; for example, they are encrypted with a master key that only the authentication server knows. A password is decrypted only when its associated account attempts to log in.

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

REFERENCES:

- [1] R. Biddle, S. Chiasson, and P. C. van Oorschot, "Graphical passwords: Learning from the first twelve years," *ACM Comput. Surveys*, vol. 44, no. 4, 2012.
- [2] (2012, Feb.). The Science Behind Passfaces [Online]. Available: <http://www.realuser.com/published/Science-BehindPassfaces.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. ZHU et al.: NEW SECURITY PRIMITIVE BASED ON HARD AI PROBLEMS 903
- [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.



International Journal of Research in Advanced Computer Science Engineering

A Peer Reviewed Open Access International Journal

www.ijracse.com

[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/> [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, pp. 161–170.

[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, pp. 235–258, 2006.

About Authors:

1. Miss Velugoti. Nadiya She Was Born In Vinjamur [V] [M], S.P.S.R.Nellore [Dt], Andhra Pradesh, India. She Received The B.Tech Degree In Information Technology From Jnt University, Anantapur In 2012 And Pursuing M.Tech Degree In Software Engineering From Jnt University, Anantapur. She Completed Her B.Tech Degree In Sri Raghavendra Institute Of Science & Technology, Vinjamur [V] [M], S.P.S.R.Nellore [Dt], Andhra Pradesh And M.Tech Degree In Skr College Of Engineering & Technology, Konduru Satram [V], Manubolu [M], S.P.S.R.Nellore [Dt], Andhra Pradesh, India.

2. Mr. Syed Baji He Was Born In Andhra Pradesh, India. He Received The Bachelor Of Computer Applications Degree From Sri Krishnadevaraya University, Anantapur In 1998-2001 And Master Of Science In Information Technology From Bharath Institute Of Science & Technology, Madras University, Chennai In 2001-2003 And Master Of Technology In Information Technology From Bharath Institute Of Higher Education & Research From Bharath University, Chennai In 2003-2005. He Has 10 Years Experience In The Field Of Associate Professor And Hod In Dept .Of Cse & It. He Had Working As Associate Professor And P.G Co-Ordinator In Dept. Of Software Engineering In Skr College Of Engineering & Technology, Konduru Satram [V], Manubolu [M], S.P.S.R Nellore [Dt], Andhra Pradesh, India.