ISSN 2347-3983

International Journal of Emerging Trends in Engineering Research, Vol.3. No.10, Pages : 234-240 (2015)
Special Issue of ICACSSE 2015 - Held on October 30, 2015 in St. Ann's College of Engineering & Technology, Chirala, AP, India
http://www.warse.org/IJETER/static/pdf/Issue/icacsse2015sp41.pdf

CAPTCHA AS GRAPHICAL PASSWORDS—A NEW SECURITY PRIMITIVE BASED ON HARD AI PROBLEMS

Ms. T.RANI[1], Dr. P. Harini[2]

[1]*II M.Tech. - II Sem., Dept. of CSE, St. Ann's College of Engineering. & Technology. Chirala,*
Andhra Pradesh -,523 187 INDIA,
raniteetla@gmail.com
[2]*Professor &Head , Dept. of CSE, St. Ann's College of Engg. & Tech., Chirala, A. P, INDIA*
drharinicse@gmail.com

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.

INTRODUCTION

AFundamental task in security is to create cryptographic primitives based on hard mathematical problems that are computationally intractable. For example, the problemof 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 in [17], is an exciting new par-adigm. Under this paradigm, the most notable primitive invented is Captcha, which distinguishes human users from computers by presenting a challenge, i.e., a puzzle, beyondthe 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.

CAPTCHA AS GRAPHICAL PASSWORDS

A. A New Way to Thwart Guessing Attacks

In a guessing attack, a password guess tested in an unsuccessful trial is determined wrong and excluded from subsequent trials. The number of undetermined password guesses decreases with more trials, leading to a better chance of finding the password. Mathematically, let *S* be the set of password guesses before any trial, ρ be the password to find, *T* denotea trial whereas T_n denote the *n*-th trial, and $p(T = \rho)$ be the probability that ρ is tested in trial *T* . Let E_n be the set of password guesses tested in trials up to (including) T_n . The password guess to be tested in *n*-th trial T_n is from set $S \backslash E_{n-1}$, i.e., the relative complement of E_{n-1} in *S*. If $\rho \in S$, then we have

$$p\ (T = \rho | T_{1}\_= \rho, \ldots , T_{n}-_{1}\_= \rho\ ) > p(T = \rho), \qquad (1)$$

and

$$E_n \rightarrow S \qquad \text{with } n \qquad S\ , \quad (2)$$

$$p(T = \rho | T_{1}\_= \rho, \ldots , T_{n-1}\_= \rho) \rightarrow 1 \qquad \rightarrow |\ \ |$$

where $|S\ |$ denotes the cardinality of *S*. From Eq. (2), the password is always found within $|S|$ trials if it is in *S*; otherwise *S* is exhausted after $|S|$ trials. Each trial determines if the tested password guess is the actual password or not, and the trial's result is deterministic.

CaRP adopts a completely different approach to counter automatic guessing attacks. It aims at realizing the following equation:

$$p(T = \rho | T_1, \ldots , T_{n-1}) = p(T = \rho), \quad \forall n \qquad (3)$$

in an automatic guessing attack. Eq. (3) means that each trial is computationally independent of other trials. Specifically, no matter how many trials executed previously, the chance of finding the password in the current trial always remains the same.

B. CaRP: An Overview

In CaRP, a new image is generated for every login attempt, even for the same user. CaRP uses an *alphabet* of visual objects (e.g., alphanumerical characters, similar animals) to generate a CaRP image, which is also a Captcha challenge. A major difference between CaRP images and Captcha images is that all the visual objects in the alphabet should appear in a CaRP image to allow a user to input any password but not necessarily in a Captcha image. Many Captcha schemes can be converted to CaRP schemes, as described in the next subsection.

C. Converting Captcha to CaRP

In principle, any visual Captcha scheme relying on recogniz-ing two or more predefined types of objects can be converted to a CaRP. All text Captcha schemes and most IRCs meet this requirement. Those IRCs that rely on recognizing a single predefined type of objects can also be converted to CaRPs in general by adding more types of objects. In practice, conversion of a specific Captcha scheme to a CaRP scheme typically requires a case by case study, in

D. User Authentication WithCaRP Schemes

Like other graphical passwords, we assume that CaRP schemes are used with additional protection such as secure channels between clients and the authentication server through Transport Layer Security (TLS). A typical way to apply CaRP schemes in user authentication is as follows. The authentica-tion server *AS* stores a salt *s* and a hash value $H(\rho,s)$ for each user ID, where ρ is the password of the account and not stored. A CaRP password is a sequence of visual object IDs or clickable-points of visual objects that the user selects. Upon receiving a login request, *AS* generates a CaRP image, records the locations of the objects in the image, and sends the image to the user to click her password. The coordinates of the clicked points are recorded and sent to *AS* along with the user ID. *AS* maps the received coordinates onto the CaRP image, and recovers a sequence of visual object IDs or clickable points of visual objects, ρ^{-}, that the user clicked on the image. Then *AS* retrieves salt *s* of the account, calculates the hash value of ρ^{-} with the salt, and compares the result with the hash value stored for the account. Authentication succeeds only if the two hash values match. This process is called the *basic CaRP authentication* and shown in Fig. 1.

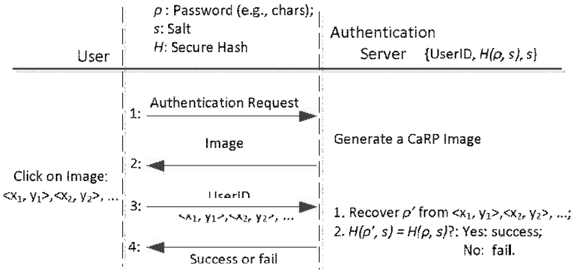

User
ρ : Password (e.g., chars);
s: Salt
H: Secure Hash
Authentication Server {UserID, H(ρ, s), s}
1: Authentication Request
Image
Generate a CaRP Image
2:
Click on Image: <x1, y1>,<x2, y2>, ...
3: UserID <x1, y1>,<x2, y2>, ...
1. Recover ρ′ from <x1, y1>,<x2, y2>, ...;
2. H(ρ′, s) = H(ρ, s)?: Yes: success; No: fail.
4: Success or fail

Fig. 1. Flowchart of basic CaRP authentication.

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 topof 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", 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

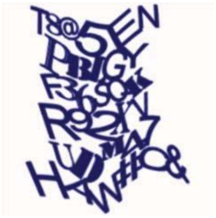


Fig. 2. A ClickText image with 33 characters.

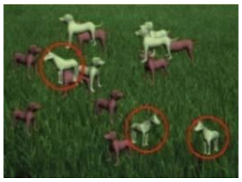


Fig. 3. Captcha Zoo with horses circled red.

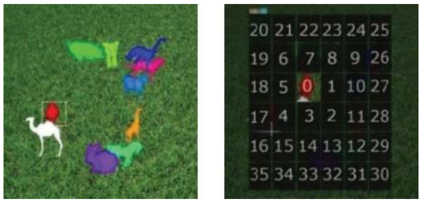


Fig. 4. A ClickAnimal image (left) and 6 × 6 grid (right) determined by red turkey's bounding rectangle.on 2D space. This is different from text Captcha challenges in which characters are typically ordered from left to right in order for users to type them sequentially. Fig. 2 shows a ClickText image 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 ρ= “AB#9CD87”.

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.

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.

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.

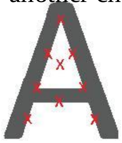
Once the bounding rectangle of the selected animal is identified, an image of $n \times n$ grid with the identified bounding rectangle as its grid-cell size is generated and displayed. If the grid image is too large or too small for a user to view, the grid image is scaled to a fitting size. The user then clicks a sequence of zero to multiple grid-cells that match the grid-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-cell_2_, and then a point inside grid-cell_1_ 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

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. 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. 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



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

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 click-able 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

Authentication. In entering a password, a user-clicked pointis 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.

. SECURITY ANALYSIS

A. Security of Underlying Captcha

Computational intractability in recognizing objects in CaRP images is fundamental to CaRP. Existing analyses on Captcha security were mostly case by case or used an approximate process. No theoretic security model has been established yet. Object segmentation is considered as a computationally-expensive, combinatorially-hard problem [30], which modern text Captcha schemes rely on. According to [30], the com-plexity of object segmentation, C, is exponentially dependent of the number M of objects contained in a challenge, and polynomially dependent of the size N of the Captcha alphabet: $C = \alpha^{M}P(N)$, where $\alpha > 1$ is a parameter, and $P()$ is apolynomial function. A Captcha challenge typically contains 6 to 10 characters, whereas a CaRP image typically contains

image is about $\alpha^{30}P(N)/(\alpha^{10}P(N)) = \alpha^{20}$ times the

B. Automatic Online Guessing Attacks

If we ignore negligible probabilities, CaRP with underlying CPA-secure Captcha has the following properties:

The first property can be proved by contradiction. Assume that the property does not hold, i.e., there exists an inter-nal object-point α on one image A that is non-negligibly dependent of an internal object-point β on another image B. An adversary can exploit this dependency to launch thefollowing chosen-pixel attack. In the learning phase, image A is used to learn the object that contains point α.

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 $33^8 \approx 2^{40}$ for ClickText with an alphabet of 33 characters, $10^8 \approx 2^{26}$ for ClickAnimal with an alphabet of 10 animals, and $10 \times 46^7 \approx 2^{42}$ for AnimalGrid with the setting as ClickAnimal plus 6 × 6 grids.

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?

E. Shoulder-Surfing Attacks

Shoulder-surfing attacks are a threat when graphical passwords are entered in a public place such as bank ATM machines. CaRP is not robust to shoulder-surfing attacks by itself. However, combined with the following dual-view technology, CaRP can thwart shoulder-surfing attacks.

. EMPIRICAL EVALUATIONS

A. Implementations

ClickText and AnimalGrid were implemented using ASP.NET. ClickText was implemented by calling a config-urable text Captcha engine commercially used by Microsoft.

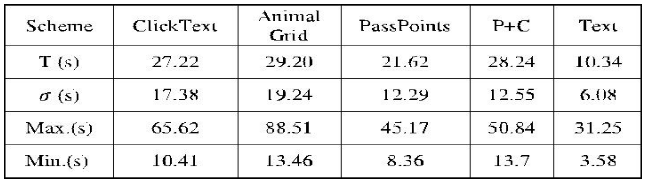
This Captcha engine accepts only capital letters. As a result, we chose the following 33 characters in our usability studies: capital letters except I, J, O, and Z, digits except 0 and 1, and three special characters "#", "@", and "&".Each image was set to 400 by 400 pixels. Fig. 2 in Section IV-A shows an image generated with the above setting.

Experimental Results

Usability. Table I shows the login time averaged over the 40 participants' successful login attempts and the sample standard deviation as well as the maximum and minimum login times for each scheme. ClickText, AnimalGrid and P + C had similar average login time whereas PassPoints had a little shorter average login time.

TABLE I

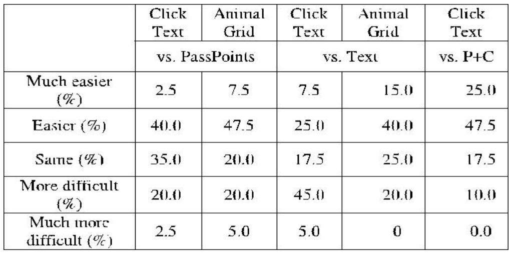
LOGIN TIME FOR DIFFERENT SCHEMES: AVERAGE (*T*), SAMPLESTANDARD DEVIATION (σ), MAX. AND MIN.



Scheme	ClickText	Animal Grid	PassPoints	P+C	Text
T (s)	27.22	29.20	21.62	28.24	10.34
σ (s)	17.38	19.24	12.29	12.55	6.08
Max.(s)	65.62	88.51	45.17	50.84	31.25
Min.(s)	10.41	13.46	8.36	13.7	3.58

TABLE II

COMPARING DIFFERENT SCHEMES FOR EASE OF USE



	Click Text	Animal Grid	Click Text	Animal Grid	Click Text
	vs. PassPoints		vs. Text		vs. P+C
Much easier (%)	2.5	7.5	7.5	15.0	25.0
Easier (%)	40.0	47.5	25.0	40.0	47.5
Same (%)	35.0	20.0	17.5	25.0	17.5
More difficult (%)	20.0	20.0	45.0	20.0	10.0
Much more difficult (%)	2.5	5.0	5.0	0	0.0

CONCLUSION

CaRP is a combination of both a CAPTCHA and a graphical password scheme. CaRP schemes are classified as Recognition-Based CaRP and Recognition-Recall CaRP. We have discussed RecognitionBasedCaRP 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. Due to reasonable security and usability and practical applications, CaRP has good potential for refinements. 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.

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 SecuritySymp.*, 1999, pp. 1–15.

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.

[4] 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.

AUTHORS :

Ms. T.Rani Studying II M.Tech (SE) in St. Ann's College of Engineering & Technology, Chirala, She completed B.Tech.(CSE) in 2012 in Sri Mittapalli institute of Technology for Womens, Guntur

Dr. P.Harini is presently working as Professor & Head, Department of Computer science & Engineering in St. Ann's College of Engineering and Technology, Chirala. She Completed Ph.D. in Distributed and Mobile Computing from JNTUA. She guided many U.G. & P.G projects. She has more than 19 Years of Teaching and 2 Years of Industry Experience. She published more than 20 International Journals and 25 research Oriented Papers in various areas. She was awarded Certificate of Merit by JNTUK., Kakinada on the University Formation day, 21st August 2012.