Ann. For. Res. 65(1): 1335-1348, 2022
ISSN: 18448135, 20652445

ANNALS OF FOREST RESEARCH
https://www.e-afr.org/

AUTOMATED TURING TEST FOR A SAFETY ANALYSIS BASED ARTIFICIAL INTELLIGENCE TECHNIQUE

[1]Dr.Y.David Solomon Raju, [2]Dr.Kande Srinivas, [3]Dr.D.Nagaraju [4]Dr.MVSS Sastri,
[1]Associate Professor, Department of Electronics and Communication Engineering, Holy Mary Institute of Technology and Sciences, Hyderabad
[2]Associate Professor and Head, Department of Computer Science and Engineering, Balaji Institute of Technology and Sciences, Warangal, Telangana.
[3]Professor, Department of Computer Science and Engineering, Sri Venkatesa Perumal College of Engineering and Technology, Puttur, Andhra Pradesh.
[4]Associate Professor, Department of Civil Engineering, Vasavi College of Engineering, Hyderabad
Email:davidsolomonraju.4@gmail.com,raj2dasari@gmail.com,sri.kande3697@gmail.com,mvss.sastri@staff.vce.ac.in

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 underexplored. In this paper, 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. 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, which 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.

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

I.INTRODUCTION

A 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 in , 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. Defense against online dictionary attacks is a more subtle problem thanit 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) and incurs expensive helpdesk costs for account reactivation. 2) It is vulnerable to global password attacks 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 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. 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.LITRATURE SURVEY

ON PREDICTIVE MODELS AND USERDRAWN GRAPHICAL PASSWORDS

AUTHOR: P. C. van Oorschot and J. Thorpe*ACM Trans.*

PUBLISH: *Inf. Syst. Security*, vol. 10, no. 4, pp. 1–33, 2008

In commonplace text-based password schemes, users typically choose passwords that are easy to recall, exhibit patterns, and are thus vulnerable to brute-force dictionary attacks. This leads us to ask whether other types of passwords (e.g., graphical) are also vulnerable to dictionary attack because of users tending to choose memorable passwords. We suggest a method to predict and model a number of such classes for systems where passwords are created solely from a user's memory. We hypothesize that these classes define weak password subspaces suitable for an attack dictionary. For user-drawn graphical passwords, we apply this method with cognitive studies on visual recall. These cognitive studies motivate us to define a set of *password complexity factors* (e.g., reflective symmetry and stroke count), which define a set of classes. To better understand the size of these classes and, thus, how weak the password subspaces they define might be, we use the "Draw-A-Secret" (DAS) graphical password scheme of Jermyn et al. [1999] as an example. We analyze the size of these classes for DAS under convenient parameter choices and show that they can be combined to define apparently popular subspaces that have bit sizes ranging from 31 to 41—a surprisingly small proportion of the full password space (58 bits).

PURELY AUTOMATED ATTACKS ON PASSPOINTS-STYLE GRAPHICAL PASSWORDS

AUTHOR: P. C. van Oorschot, A. Salehi-Abari, and J. Thorpe

PUBLISH: *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 393–405, Sep. 2010.

We introduce and evaluate various methods for purely automated attacks against PassPoints-style graphical passwords. For generating these attacks, we introduce a graph-based algorithm to efficiently create dictionaries based on heuristics such as click-order patterns (e.g., five points all along a line). Some of our methods combine click-order heuristics with focus-of-attention scan-paths generated from a computational model of visual attention, yielding significantly better automated attacks than previous work. One resulting automated attack finds 7%-16% of passwords for two representative images using dictionaries of approximately 2^{26} entries (where the full password space is 2^{43}). Relaxing click-order patterns substantially increased the attack efficacy albeit with larger dictionaries of approximately 2^{35}entries, allowing attacks that guessed 48%-54% of passwords (compared to previous results of 1% and 9% on the

same dataset for two images with 2^{35} guesses). These latter attacks are independent of focus-of-attention models, and are based on image-independent guessing patterns. Our results show that automated attacks, which are easier to arrange than human-seeded attacks and are more scalable to systems that use multiple images, require serious consideration when deploying basic PassPoints-style graphical passwords.

EXPLOITING PREDICTABILITY IN CLICKBASED GRAPHICAL PASSWORDS

AUTHOR: P. C. van Oorschot and J. Thorpe, *J. Comput. Security*,

PUBLISH: vol. 19, no. 4, pp. 669–702, 2011.

We provide an in-depth study of the security of click-based graphical password schemes like PassPoints (Weidenbeck et al., 2005), by exploring popular points (hot-spots), and examining strategies to predict and exploit them in guessing attacks. We report on both short- and long-term user studies: one labcontrolled, involving 43 users and 17 diverse images, the other a field test of 223 user accounts. We provide empirical evidence that hot-spots do exist for many images, some more so than others. explore the use of "human-computation " (in this context, harvesting click-points from a small set of users) to predict these hot-spots. We generate two "human-seeded " attacks based on this method: one based on a first-order Markov model, another based on an independent probability model. Within 100 guesses, our first-order Markov model-based attack finds 4 % of passwords in one image's data set, and 10 % of passwords in a second image's data set. Our independent model-based attack finds 20 % within 2 33 guesses in one image's data set and 36 % within 2 31 guesses in a second image's data set. These are all for a system whose full password space has cardinality 2 43. We also evaluate our first-order Markov model-based attack with cross-validation of the field study data, which finds an average of 7-10 % of user passwords within 3 guesses. We also begin to explore some click-order pattern attacks, which we found improve on our independent model-based attacks.

A NEW CAPTCHA INTERFACE DESIGN FOR MOBILE DEVICES

AUTHOR: R. Lin, S.-Y. Huang, G. B. Bell, and Y.-K. Lee,

PUBLISH: *Proc. 12th Austral. User Inter. Conf.*, 2011, pp. 3–8.

This paper discusses and demonstrates the interplay between system security and user interface convenience in CAPTCHA design, and in particular, mobile device CAPTCHA design. A CAPTCHA is a computer-based security test used to distinguish human users from artificial users, preventing automated abuse of networked resources. As mobile network services improve, we can anticipate that future mobile network services will come under attack from automated programs. Importantly, while CAPTCHA techniques have existed for Internet services for some time, only limited work has been carried out to establish CAPTCHAs suitable for mobile device interfaces. The Drawing CAPTCHA (2006) is one of the most well known systems of this type. Unfortunately, though it is straightforward, it is not secure. To demonstrate this, an image-processing technique is newly proposed that breaks the Drawing CAPTCHA.

III.SYSTEM ANALYSIS

EXISTING SYSTEM:

Security primitives are based on hard mathematical problems. Using hard AI problems for security is emerging as an exciting new paradigm, but has been underexplored. **A** FUNDAMENTAL task in security is to create cryptographic primitives based on hard mathematical problems that are computationally intractable.

DISADVANTAGES:

This paradigm has achieved just a limited success as compared with the cryptographic primitives based on hard math problems and their wide applications.

Using hard AI (Artificial Intelligence) problems for security, initially proposed in, 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.

PROPOSED SYSTEM:

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. 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**.** 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. Defense against online dictionary attacks is a more subtle problem than it might appear.

ADVANTAGES:

It offers reasonable security and usability and appears to fit well with some practical applications for improving online security.

This threat is widespread and considered as a top cyber security risk. Defense against online dictionary attacks is a more subtle problem than it might appear.

Ann. For. Res. 65(1): 1335-1348, 2022
ISSN: 18448135, 20652445

IV.SYSTEM DESIGN

ARCHITECTURE DIAGRAM / UML DIAGRAMS / DAT FLOW DIAGRAM

- The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system
- The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
- DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
- DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

NOTATION:

SOURCE OR DESTINATION OF DATA:

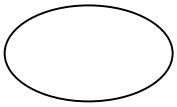
External sources or destinations, which may be people or organizations or other entities

DATA SOURCE:

Here the data referenced by a process is stored and retrieved.

PROCESS:

People, procedures or devices that produce data. The physical component is not identified.



DATA FLOW:

Data moves in a specific direction from an origin to a destination. The data flow is a "packet" of data.

MODELING RULES:

There are several common modelling rules when creating DFDs:

1. All processes must have at least one data flow in and one data flow out.
2. All processes should modify the incoming data, producing new forms of outgoing data.
3. Each data store must be involved with at least one data flow.
4. Each external entity must be involved with at least one data flow.

5. A data flow must be attached to at least one process.

DATAFLOW DIAGRAM

USER:

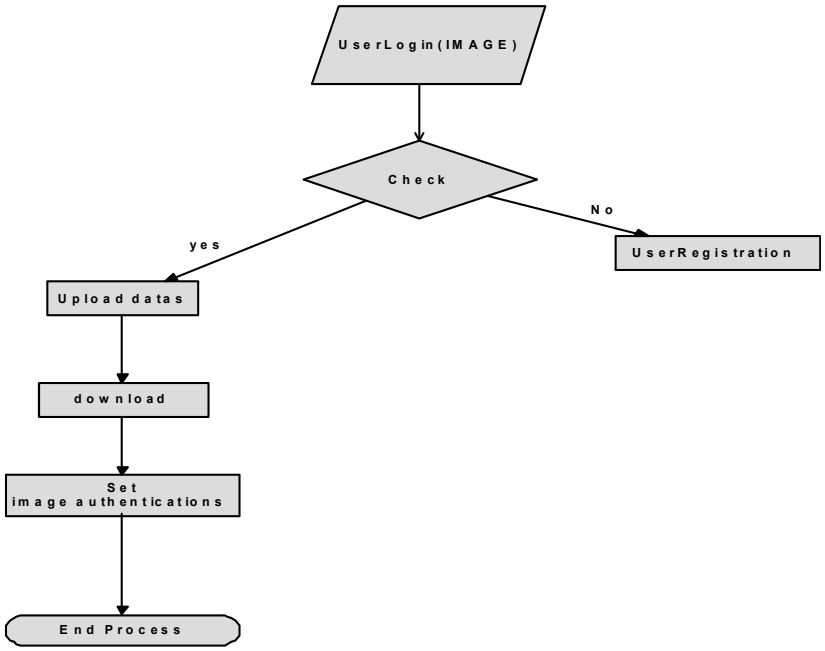


Figure 1: Flow Diagram

UML DIAGRAMS:

USE CASE DIAGRAM:

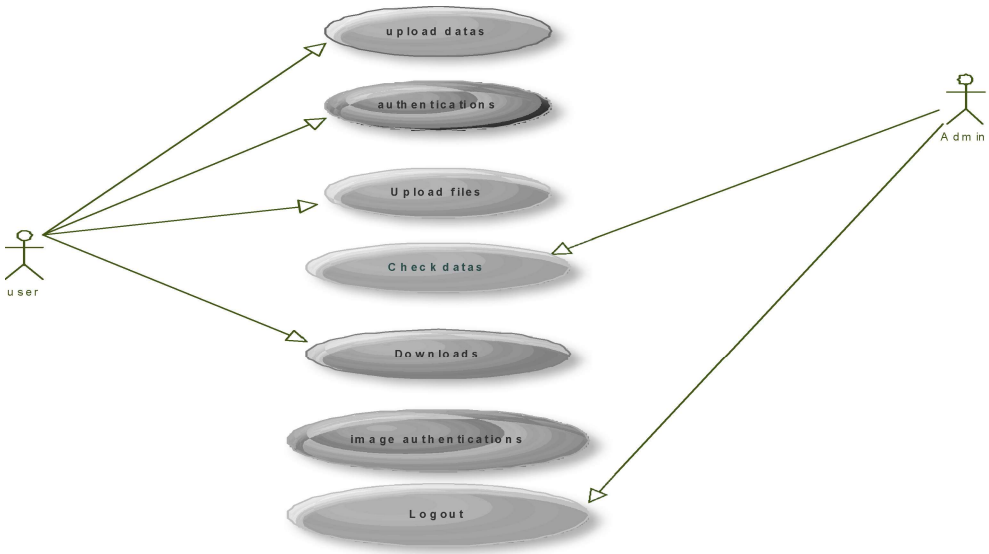


Figure 2: Use Case Diagram

CLASS DIAGRAM:

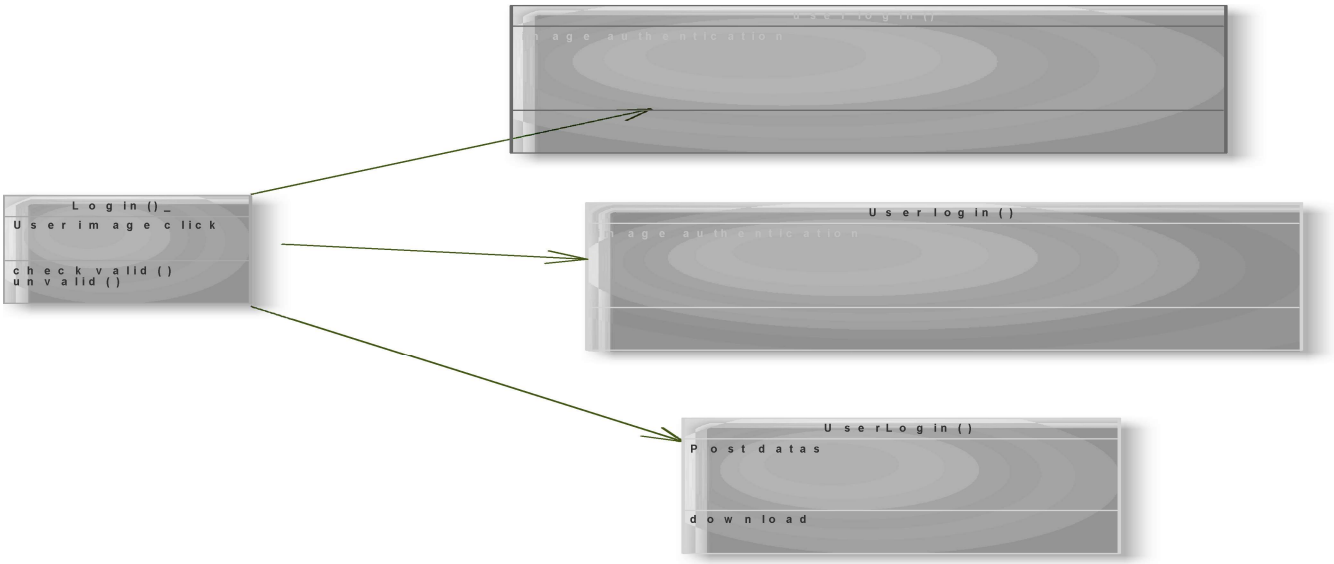


Figure 3: Class Diagram

SEQUENCE DIAGRAM:

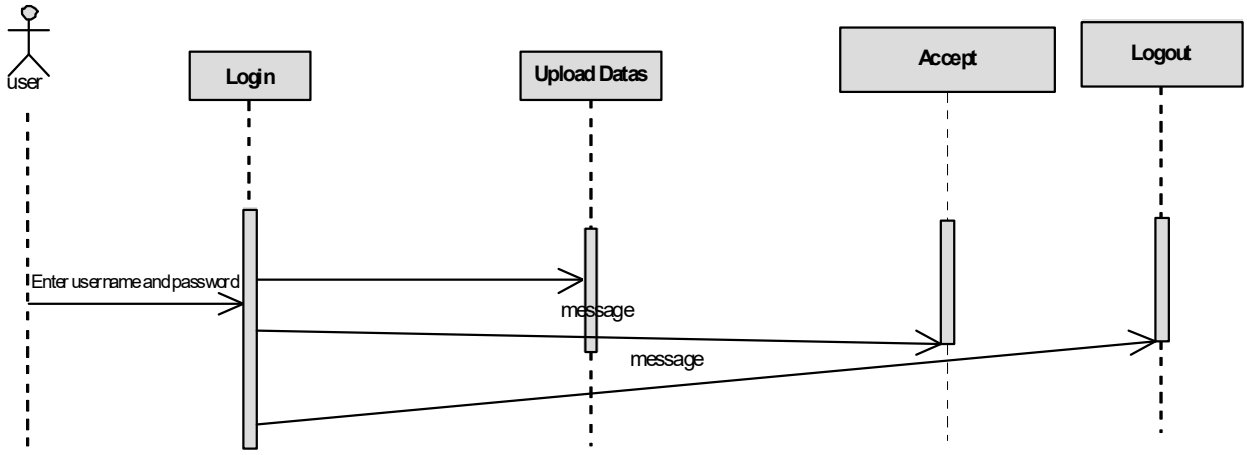


Figure 4: Sequence Diagram

ACTIVITY DIAGRAM:

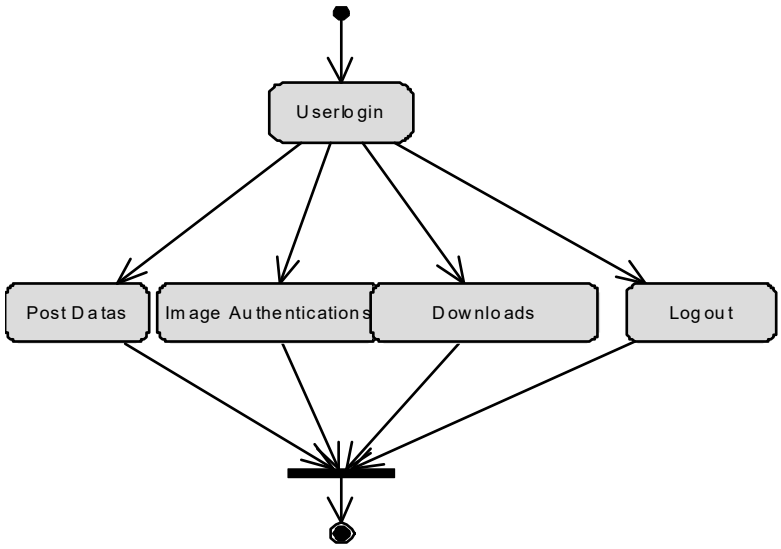


Figure 5: Activity Diagram

V.IMPLEMENTATION

COMPONENT DIAGRAM:

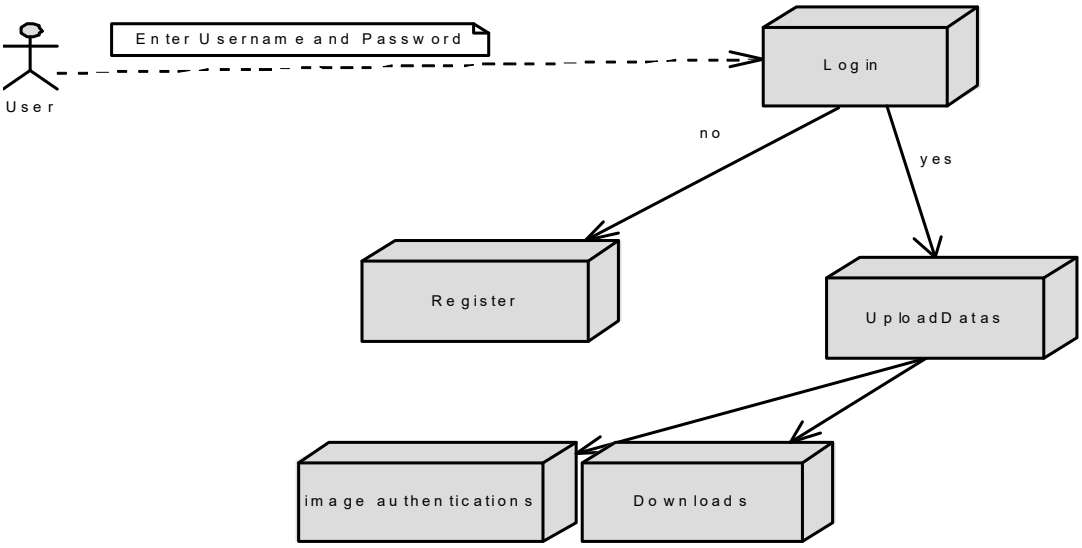


Figure 6: Implementation of Automated Turing Test

IMPLEMENTATION:

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

MODULES:

a. Graphical Password
b. Captcha in Authentication
c. Thwart Guessing Attacks
d. Security Of Underlying Captcha

MODULE DESCRIPTION:

GRAPHICAL PASSWORD:

In this module, Users are having authentication and security to access the detail which is presented in the Image system. Before accessing or searching the details user should have the account in that otherwise they should register first.

CAPTICA IN AUTHENTICATION:

It was introduced in 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 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.

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. To counter guessing attacks, traditional approaches in designing graphical passwords aim at increasing the effective password space to make passwords harder to guess and thus require more trials. No matter how secure a graphical password scheme is, the password can always be found by a brute force attack. In this paper, we distinguish two types of guessing attacks: *automatic guessing attacks* apply an automatic trial and error process but *S* can be manually constructed whereas *human guessing attacks* apply a manual trial and error process.

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, which modern text Captcha schemes rely on.

SCREEN SHOTS OF RESULTS

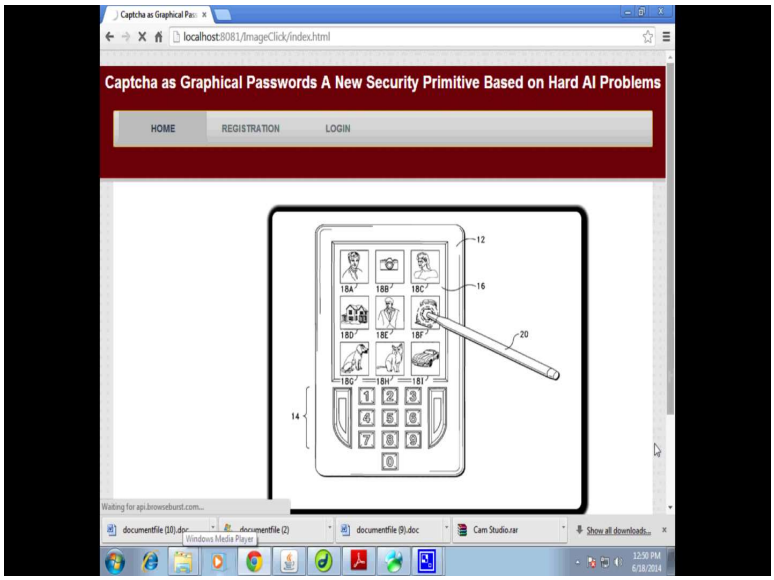


Figure 7: Captcha as Graphical Passwords a Security Based AI

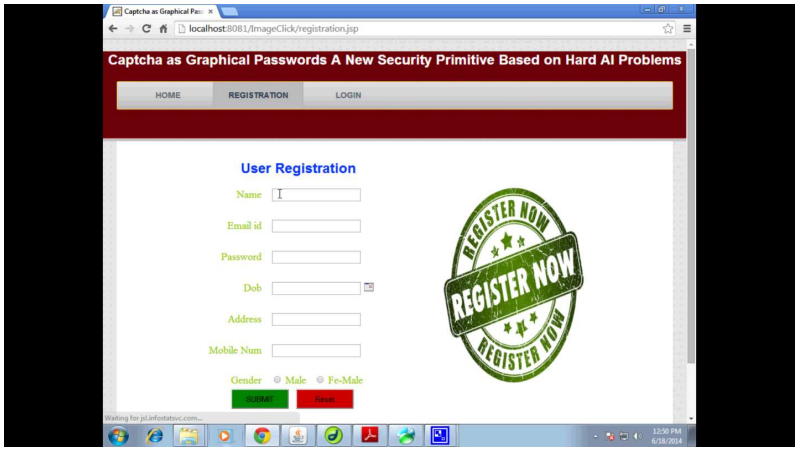


Figure 8: User Registration

Figure 9: Selection of Image Captcha for Click

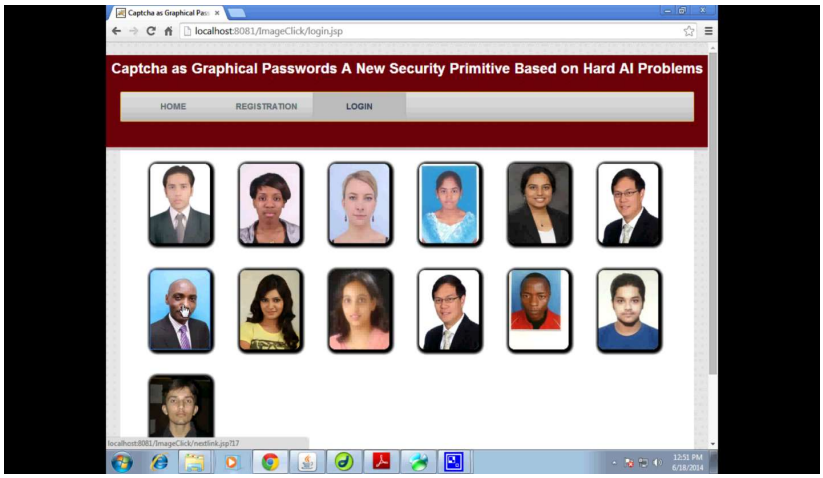


Figure 10: User Login after Successful Registration

VI.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]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

[2]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.

[3] P. C. van Oorschot and J. Thorpe, "Exploiting predictability in clickbased graphical passwords," *J. Comput. Security*, vol. 19, no. 4, pp. 669–702, 2011.

[4]R. Lin, S.-Y. Huang, G. B. Bell, and Y.-K. Lee, "A new CAPTCHA interface design for mobile devices," in *Proc. 12th Austral. User Inter. Conf.*, 2011, pp. 3–8.

[5]N. Joshi. (2009, Nov. 29). *Koobface Worm Asks for CAPTCHA* [Online]. Available: http://blogs.mcafee.com/mcafee-labs/koobface-worm-asksfor- CAPTCHA

[6] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G. M. Voelker, and S. Savage, "Re: CAPTCHAs—Understanding CAPTCHA-Solving Services in an Economic Context," in *Proc. USENIX Security*, 2010

[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 click- based 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.