Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

SECURED CLIENT CERTIFICATION SCHEME WITH STRENGTH ANALYSIS

Ms. D. Anantha Nayaki[1],Ms. V. Uma[2], Ms. K.K. Kavitha[3]
[1]M.C.A., MPhil., Assistant Professor (Computer Science) ,[2]M.C.A., MPhil, Research Scholar,[3]M.C.A., M.Phil, Assistant Professor (Computer Science)
Selvamm Arts & Science College (Autonomous), Namakkal, Tamilnadu, India

Abstract-Client server technology is mainly used to share resources and services. Access policies are designed with reference to the client privileges. Client account verification is the main task in the client server technology. Client certificates are mainly used in the account verification process. User ID, password details and key values are integrated in the client certificates. The certificates are uploaded by the client to the server for the verification process. All the user activities are allowed after the successful completion of the client certificate verification process.

Client ID and password values are validated by the authentication server. The passwords are composed with text and graphical images. The graphical password scheme uses the images and click information for the verification process. Captcha technology is used to identify the requestor is a machine or human. Captcha and graphical password schemes are integrated to construct the Captcha as Graphical Password (CaRP) scheme. Image captchas are used in the recognition and recall based password verification mechanism.

The client certificates are secured with RSA based encryption and decryption techniques. The Secure Hash Algorithm (SHA) algorithm is used for the integrity verification process. Pattern based model uses the spatial and color patterns for the password strength analysis. The transmission attacks and directory attacks are managed with security features. The system also handles the shoulder surfing attacks. Password construction process is upgraded with strength nalysis mechanism. The client certification scheme can be applied for all types of network applications.

I. INTRODUCTION

A CAPTCHA ("Completely Automated Public Turingtest to tell Computers and Humans Apart") is a type of challenge-response test used in computing to determine whether or not the user is human. The term was coined in 2000 by Luis von Ahn, Manuel Blum, Nicholas J. Hopper of Carnegie Mellon University and John Langford of IBM. The most common type of CAPTCHA was first invented by Mark D. Lillibridge, Martin Abadi, Krishna Bharat and Andrei Z. Broder. This form of CAPTCHA requires that the user type the letters of a distorted image, sometimes with the addition of an obscured sequence of letters or digits that appears on the screen. Because the test is administered by a computer, in contrast to the standard Turing test that is administered by a human, a CAPTCHA is sometimes described as a reverse Turing test. This term is ambiguous because it could also mean a Turing test in which the participants are both attempting to prove they are the computer. This user identification procedure has received many criticisms, especially from disabled people, but also from other people who feel that their everyday work is slowed down by distorted words that are illegible even for users with no disabilities at all.

CAPTCHAs are by definition fully automated, requiring little human maintenance or intervention to administer. This has obvious benefits in cost and reliability. By definition, the algorithm used to create the CAPTCHA must be made public, though it may be covered by a patent. This is done to demonstrate that breaking it requires the solution to a difficult problem in the field of artificial intelligence (AI) rather than just the discovery of the secret algorithm, which could be obtained through reverse engineering or other means. Modern text-based CAPTCHAS are designed such that they require the simultaneous use of three separate abilities—invariant recognition, segmentation and parsing—to correctly complete the task with any consistency.

1. Invariant recognition refers to the ability to recognize the large amount of variation in the shapes of letters. There are nearly an infinite number of versions for each character that a human brain can successfully identify. The same is not true for a computer and teaching it to recognize all those differing formations is an extremely challenging task.
2. Segmentation, or the ability to separate one letter from another, is also made difficult in CAPTCHAs, as characters are crowded together with no white space in between.
3. Context is also critical. The CAPTCHA must be understood holistically to correctly identify each character. For example, in one segment of a CAPTCHA, a letter might look like an "m." Only when the whole word is taken into context does it become clear that it is a "u" and an "n."

Each of these problems poses a significant challenge for a computer, even in isolation. The presence of all three at the same time is what makes CAPTCHAs difficult to solve. Unlike computers, humans excel at this type of task. While segmentation and recognition are two separate processes necessary for understanding an image for a computer, they are part of the same process for a person. For example, when an individual understands that the first letter of a CAPTCHA is an "a", that individual also understands where the contours of that "a" are and also where it melds with the contours of the next letter. Additionally, the human brain is capable of dynamic thinking based upon context. It is able to keep multiple explanations alive and then pick the one that is the best explanation for the whole input based upon contextual clues. This also means it will not be fooled by variations in letters.

II. RELATED WORK

The fact that shoulder-surfing attacks are directed at the human user makes their prevention through cryptographic techniques quite infeasible. Since the seminal work of Matsumoto and Imai and the work by Wang et al. that discussed its security, a large number of studies have considered alternatives that are within the limitations of humans. The central theme has been to incorporate an indirect method for secret transfer, that is, to separate the visible key entry procedure from the secret itself. Some of the attempts have focused on textual passwords, graphical passwords and PINs [5]. Some have leveraged the use of haptic channels [4], with the work even taking into account the possibility of the haptic channel leaking information visually. Even the approach of physically occluding the leakage from at least some component of the visual channel [10], [3] can be found. The very existence of these diverse schemes testifies as to how challenging it is to design an authentication scheme that is both secured and usable [14]. Measures that strengthen security are likely to result in highly complex, error-prone, and tedious user procedures, while putting more emphasis on usability can lead to insecure schemes [2].

The difficulty of the task is also evident from the following non-exhaustive list of related works: Golle and Wagner revealed the insecurity of the cognitive authentication scheme. Li et al. represented a brute force attack against the so-called PAS scheme. Dunphy et al. showed a replay-based shoulder-surfing attack against the recognition-based graphical password system [7]. Asghar et al. revealed the insecurity of the Convex Hull Click (CHC) and its related methods [1]. Yan et al. reported on general attacks against leakage-resilient password systems and discussed the security-usability tradeoff [9].

Asghar et al. revisited Yan et al.'s work and showed how to theoretically estimate a lower bound on the number of authentication sessions that are safe against passive observers [2]. Kwon et al. showed that the basic IOC version of the BW method was vulnerable to human shoulder-surfing attacks if those attackers were trained and prepared [11]. Bianchi et al. discussed how a directional microphone or similar device was a realistic threat to vibrotactile signaling schemes [4]. Perkovic et al. disclosed the insecurity of Undercover by exploiting the user's behavioral (timing) characteristics or the systematic intersection of multiple random challenges [8].

Let us very briefly present the Undercover scheme, created by Sasamoto et al. that uses graphical passwords. As with almost any other schemes, they made use of graphical challenges, but made further use of separate tactile challenges, delivered through a specially designed haptic device. The user placed one hand on a trackball to sense its direction of spin or vibration. This tactile challenge was mentally combined with the graphical challenge to create a temporary identifier for the user's secret image, which was submitted to the system with the other hand. Our TictocPIN scheme, which uses short vibration signals, relies on the innovative idea set forth by Undercover in that we also exploit the security enhancements made possible by a hidden haptic channel. Needless to say, TictocPIN, which provides sufficient security against even the camera-based recording attackers, is clearly also a security-strengthened extension to the creation.

Our work may also be seen as extending the work of Kwon et al. [11], which dealt with the human shoulder surfer attacking a IOC BW system. By exercising selective cognitive attention, a trained attacker was able to conduct a perceptual grouping of colored patterns to single out a PIN digit. Furthermore, through parallel motor operations, each of his finding could be written down without impeding his ability to identify the next PIN digit, thus completely breaking the IOC BW method. As noted previously, some of the analysis results given in our work provides insights as to why such an attack was possible. Furthermore, our work extends the work of Kwon et al. by providing security shortcoming of all versions of the BW method, through both theoretical and experimental means. Finally, we remark that the lessons learned from the timing attacks of [8] has allowed us to specify for the display of the input color pad to be delayed and randomized with TictocPIN.

III. 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. Text Point, a recognition recall CaRP scheme with an alphabet of characters, is presented next, followed by a variation for challenge response authentication.

3.1. TextPoints

Characters contain invariant points. Fig. 3.1. 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, the system should avoid selecting the center of a similar stroke segment in another character. Instead, the should select 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.

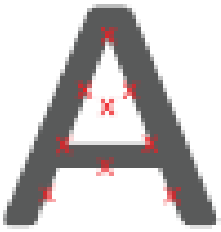

Fig. 3.1. Some Invariant Points (Red Crosses) of "A".

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. The system 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. It is worth comparing potential password points between TextPoints and traditional click-based graphical passwords such as PassPoints. 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 and contextual:

- **Dynamic**: locations of clickable points and their contexts vary from one image to another. The clickable points in one image are computationally independent of the clickable points in another image.
- **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.

3.2. 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 , the system requires 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 $\acute{\Gamma}$, are located on the image. For every point in $\acute{\Gamma}$, its distance is calculated as 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 $\acute{\Gamma}$ 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 $\acute{\Gamma}$, a new set is obtained by $\acute{\Gamma}$ comprising internal points; these are either internal clickable points or target points of boundary clickable points. Mesh warping, widely used in generating text Captcha challenges, is then used to warp the image so that $\acute{\Gamma}$ maps to Γ. The result is a TextPoint4CR 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 $\acute{\Gamma}$ to Γ.

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. Unlike other CaRP schemes, 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.

IV. PROBLEM STATEMENT

Captcha as graphical passwords (CaRP) is a graphical password scheme used for user authentication. Online guessing attacks, relay attacks and shoulder surfing attacks are handled in CaRP. CaRP is click-based graphical passwords where a sequence of clicks on an image is used to derive a password. Dynamic captcha challenge image is used for each login attempt in CaRP. Text Captcha and image-recognition Captcha are used in CaRP scheme. Text CaRP scheme constructs the password by clicking the right character sequence on CaRP images. CaRP schemes can be classified into two categories recognition based CaRP and *recognition-recall based CaRP*. Recognition-based CaRP seems to have access to an infinite number of different visual objects. Recognition-recall based CaRP *r*equires recognizing an image and using the recognized objects as cues to enter a password. Recognition-recall combines the tasks of both recognition and cued-recall. Password information is transferred and verified

using hash codes. Secure channels between clients and the authentication server through Transport Layer Security (TLS). The following drawbacks are identified from the existing system. Click point relationship are not analyzed, Directory attacks are not handled, Device dependant shoulder surfing attack handling mechanism and Hash code security is not considered.

V. SECURED CLIENT CERTIFICATION SCHEME WITH STRENGTH ANALYSIS

The CaRP scheme is enhanced with strength analysis and security features. Pattern based attacks are handled with Color and Spatial patterns. Pixel colors in click points are considered in the color pattern analysis model. Pixel location patterns are considered in the spatial pattern analysis model. Dictionary attacks and transmission attacks handling process is also improved with high security. Password security level assessment mechanism is used in the graphical password construction process. Cryptography (RSA) and data integrity (SHA) schemes are also integrated with the system to improve the security level in online applications. CAPTCHA and graphical password schemes are used for the user authentication process. Pixel physical and spatial properties are used in the strength analysis process. Transmission security is improved with integrity verification mechanisms. The system is divided into six major modules. They are CaRP with Text CAPTCHA, authentication server, CaRP with image Recognition CAPTCHA, pattern analysis, attack handler and enhanced CaRP scheme.

Character sequence selection is used in CaRP with Text CAPTCHA scheme. The authentication server is designed to manage and verify the user accounts. CaRP with Image Recognition CAPTCHA scheme uses the recognition and recall mechanism with image objects. The color and spatial patterns are analyzed under the pattern analysis module. The directory and shoulder surfing attacks are handled under attack handler module. Enhanced CaRP Scheme integrates the security and attack control mechanism for user authentication process. Textual characters based CAPTCHA is used in Text CaRP scheme. Password is constructed by selecting character sequences in the text CAPTCHA collection. The textual CAPCHA characters are dynamically rearranged at the time of recognition process. Password details are converted into hash codes and applied in verification process. The authentication server application is used to authenticate the users. User registration and password management operations are carried out under the server. Password verification is carried out under the server. Key and signature values are maintained under the server.

Image objects are used in recognition-recall based CaRP Recognition CAPTCHA. Object recognition and click cue identification mechanism are used in the system. Rectangular regions are used in the cued recall process. CAPTCHA-Zoo image object collection is used for the password construction process. Color and spatial patterns are analyzed in the system. Pixel color for click points are used in the color pattern analysis. Spatial patterns are extracted from location information. Password complexity is assessed with pattern information. Directory and shoulder surfing attacks are managed by the system. RSA algorithm is used to perform password encryption/decryption tasks. Image dimming mechanism is used to control shoulder surfing attacks. Mouse cursor size and location are automatically adjusted for attack handling process. CaRP scheme and attack handling mechanism are integrated in the Enhanced CaRP scheme. Distribution, strength and pattern analysis schemes are integrated with CaRP scheme. The Secure hashing algorithm (SHA) is used to generate password signatures. Reusability level is analyzed.

VI. PERFORMANCE ANALYSIS

The user authentication system is designed with the graphical passwords and Captcha schemes. The Captcha as Graphical Passwords (CARP) scheme is constructed with the integration of the Captcha scheme and Graphical password techniques. The Captcha is applied to verify the request is initiated by the user or not. The graphical passwords are used to verify the correct user registered in the server.

Recognition based scheme and Recognition and recall based schemes are used in the CARP scheme. Captcha text and images are used in the verification process. The password details are transferred as hash codes. The CARP scheme did not perform the strength analysis.

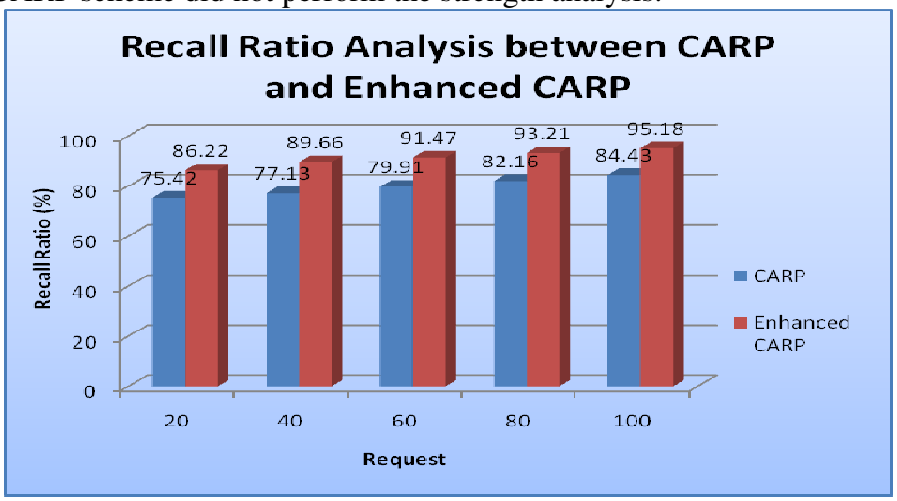

Figure No.6.1: Recall Ratio Analysis between CARP and Enhanced CARP

The Enhanced CARP scheme is constructed with the CARP scheme with strength and attack analysis mechanism. The pattern analysis mechanism is used to check the password strength levels. Three types of password strength analysis mechanism are used in the system. They are distribution analysis, color pattern analysis and spatial pattern analysis. The system also handles the directory attacks and transmission attacks. The RSA algorithm is used to secure the password transmission process. The digital signature scheme is used to verify the transmission errors. The Secure Hash Algorithm (SHA) is used in the system. The system also handles the shoulder surfing attacks. Image dimming, mouse cursor thinning and dynamic mouse curse movement techniques are used to control the shoulder surfing attacks.

The CARP scheme and Enhanced CARP scheme are analyzed with different verification instances. The recall ratio analysis is used to verify password recall levels in user authentication process. The recall ratio is estimated with different user verification entries. The recall ratio analysis between the CARP and Enhanced CARP scheme are shown in figure 6.1. The analysis results show that the Enhanced CARP scheme produces 15% better recall ratio than the CARP scheme.

VII. CONCLUSION AND FUTURE WORK

The graphical passwords are used to ensure the high level security for the remote logins. CAPTCHA techniques are used to verify the source type of request. Captcha as Graphical Passwords (CaRP) scheme integrates the text and image captchas to construct graphical password scheme. CaRP scheme is enhanced with strength based password construction and attack resistant user authentication model. Password complexity prediction system is integrated to improve password construction process. The system increases the success and recall rates. User interface is upgraded to avoid capture attacks in password recall process. Efficient shoulder surfing attack controlling models are used to protect the system from attackers. The Enhanced CaRP scheme can be adapted to perform the user verification under mobile network environment. The system can be improved with multi framed image based verification mechanism.

REFERENCES

[1] H. J. Asghar, S. Li, J. Pieprzyk and H. Wang, "Crypto Analysis Of The Convex Hull Click Human Identification Protocol," In Proc. 13th Int. Conf. Inf. Secur., 2011, pp. 24–30.

[2] H. J. Asghar, S. Li, R. Steinfeld and J. Pieprzyk, "Does Counting Still Count? Revisiting The Security Of Counting Based User Authentication Protocols Against Statistical Attacks," In Proc. 20th Symp. Internet Soc. Netw. Distrib. Syst. Secur. (Ndss), Apr. 2013, pp. 1–18.
[3] Q. Yan, J. Han, Y. Li, J. Zhou and R. H. Deng, "Designing Leakage Resilient Password Entry On Touchscreen Mobile Devices," In Proc. Asia CCS, 2013, pp. 37–48.
[4] A. Bianchi, I. Oakley and D. S. Kwon, "Counting Clicks And Beeps: Exploring Numerosity Based Haptic And Audio Pin Entry," Interact. Comput., Vol. 24, No. 5, pp. 409–422, Sep. 2012.
[5] A. De Luca, K. Hertzschuch and H. Hussmann, "Colorpin—Securing Pin Entry Through Indirect Input," In Proc. ACM Chi Conf. Human Factors Comput. Syst., 2010, pp. 1103–1106.
[6] Y. Michalevsky, D. Boneh and G. Nakibly, "Gyrophone: Recognizing Speech From Gyroscope Signals," In Proc. Usenix Secur. Symp., Aug. 2014, pp. 1053–1067.
[7] P. Dunphy, A. P. Heiner and N. Asokan, "A Closer Look At Recognition Based Graphical Passwords On Mobile Devices," In Proc. 6th Symp. Usable Privacy Secur., 2010, pp. 1–12.
[8] T. Perkovi´C, S. Li, A. Mumtaz, S. A. Khayam, Y. Javed and M. Cagalj, "Breaking Undercover: Exploiting Design Flaws And Nonuniform Human Behavior," In Proc. 7th Symp. Usable Privacy Secur., 2011, pp. 1–15, Art. Id 5.
[9] Q. Yan, J. Han, Y. Li and R. H. Deng, "On Limitations Of Designing Leakage-Resilient Password Systems: Attacks, Principals And Usability," In Proc. 19th Symp. Internet Soc. Netw. Distrib. Syst. Secur. (Ndss), Feb. 2012.
[10] D. Kim Et Al., "Multi-Touch Authentication On Tabletops," In Proc. ACM Sigchi Conf. Human Factors Comput. Syst. (Chi), 2010, pp. 1093–1102.
[11] T. Kwon, S. Shin and S. Na, "Covert Attentional Shoulder Surfing: Human Adversaries Are More Powerful Than Expected," IEEE Trans. Syst., Man, Cybern., Syst., Vol. 44, No. 6, pp. 716–727, Jan. 2014.