National Conference on Information, Communication and Energy Systems and Technologies - 2019
© 2019 IJSRSET | Volume 5 | Issue 7 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099

# Graphical Authentication System

**Patil Pramod*, Sagar Said, Ashwini More, Ambikesh Kumar**
Computer Engineering, SKNSITS, Lonavala, India

## ABSTRACT

Image Authentication or captcha based on passwords is used largely in applications for computer and mobile security and privacy. People logs into web services and applications in public to access their personal and confidential accounts with their laptops, smartphones, tablets or public devices, like bank ATM. All these things bring great convenience but at the same time increase the risk of exposing passwords to unknowns by shoulder surfing attacks. A shoulder surfing is a kind of attack where attackers can observe directly or indirectly with the use of external recording devices to collect user's credentials. To overcome this problem of shoulder surfing attacks, we propose an image-based authentication system along with encryption. With one-time valid login indicator / token, horizontal and vertical bars covering the entire scope of an image, proposed system offers no hint for attackers to figure out or narrow down password even when they conduct multiple camera based attention. In addition to this, the login indicator is completely random and valid only for short period of time. In addition to this to protect the mobile application from theft, only one email id is allowed per application and an easy-to-remember randomly generated password required for logging into the application is also sent to the user. This password is completely encrypted and valid only for single login.

**Keywords:** Authentication, Shoulder Surfing Attack, Encryption, Decryption, Login Indicator.

## I. INTRODUCTION

Over the past few years, TEXTUAL passwords have been the most widely used authentication method for security. To mitigate the brute force attack a strong textual password comprised of numbers, uppercase and lowercase letters and special characters is required. However, a strong textual password is hard to memorize and recollect [1].

Therefore, users tend to choose passwords that are either short or from the dictionary, rather than random alphanumeric strings. Even worse, it is not a rare case that users may use only one username and password for multiple accounts [2]. Various graphical password authentication schemes [3, 4] were developed to address the problems and weaknesses associated with textual passwords. However, most of these image-based passwords were vulnerable to shoulder surfing attacks (SSAs). This type of attack either uses direct observation, such as watching over someone's shoulder or applies video capturing techniques to get passwords, PINs, or other sensitive personal information [6] An image-based authentication system named Pass-Points [4] in which the user picks up several points (3 to 5) in an image during the password creation phase and re-enters each of these pre-selected click-points in a correct order within its tolerant square during the login phase. Comparing to traditional PIN and textual passwords, the Pass- Points scheme substantially increased the password space and enhanced password memorability. Unfortunately, this graphical authentication scheme was vulnerable to shoulder surfing attacks. Hence,

based on the Pass-Points, the proposed system adds an idea of using randomly generated one-time session passwords known as login indicator in an encrypted form that is resistant to shoulder surfing attacks. The proposed system named is a secure graphical authentication system that protects users from becoming victims of shoulder surfing attacks when inputting passwords in public through the use of one-time login indicators. A user has to select a count of images, the number may vary based on confidentiality of account or the information. System discretizes / divides these images into a grid of rows and columns for e.g 7 X 11. A token / login indicator is randomly generated [9, 10] for each image and will be useless after a short period of time. The goal of this random login indicator is to provide better security against shoulder surfing attacks since users will use a dynamic pointer to point out the position of their passwords rather than typing in the password directly. In addition to this, an encryption algorithm is used to avoid login indicator from being compromised.

## II. METHODS AND MATERIAL

**Mathematical Model**

**System Description:**

**Input:** Providing input in terms of email id, number of images and image block.

**Output:** Avoids user accounts from being shoulder surfed.

Let S be the whole system which consists: S = {IP, PRO, OP, A, F}

Identify IP as the input:

IP = {u, n, v}

Where,

u→ user information.

n→ number of images required for logging in.

v→ value of the selected image for verifying graphical password or token.

Identify PRO as procedure applied to the system to process the given input:

PRO = {id, hv, crt, gp, rp}

Where,

id→ process of image discretization.

hv→ process of creating horizontal and vertical bar around image.

crt→ process of creating random token and sending it to users mobile application.

gp→ process of verifying token / graphical password after user selects a particular image block.

rp→ process of creating easy-to-remember random password, encrypting it and sending it to mobile application.

Identify OP as the output of system

OP = {di, hv, rtp}

Where,

di→ discretized image generated by system.

hv→ horizontal and vertical bar generated by system.

rtp→ random token and encrypted easy-to-remember random password generated by system.

Identify A as case of success

A = {accept} Where, accept→ only registered user get access to the system in an environment of shoulder surfing attack.

Identify F as case of failure F = {ip} Where, ip→ poor network connection.

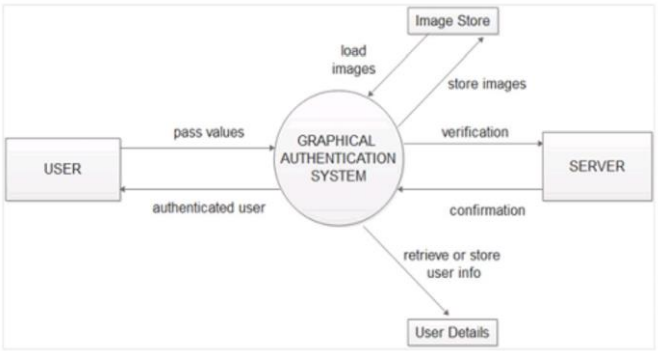


**Figure 1.** Data flow diagram.

**Module Details:**

**1. Image Discretization** Module: This module takes the image as its input and draws the horizontal and the vertical line over it forming a grid of blocks. Each image block thus formed comprises a value which will then be verified when the user clicks on the block.

**2. Token / Login Indicator Generator Module:** This module generates a token consisting of a single

alphabet and a single number. For example, characters from A to G and numbers from 1 to 11 can be used to form 77 different tokens. Each time when a call is made to this module a different token gets generated. The generated login indicator is delivered to user's mobile android application.

**3. Horizontal and Vertical Bars Module:** There are two bars: a horizontal bar with a sequence of letters and a vertical bar with a sequence of numbers or vice versa. The bars are used to implicitly point out the location of the block.

**4. Communication Module:** This module is in charge of all the information transmitted between the client devices and the authentication server. Any communication is protected by SSL (Secure Socket Layer) protocol and AES encryption algorithm thus, is safe from being eavesdropped and intercepted.

**5. Password Verification Module:** This module verifies the transmitted token with a token generated on click of the image block. The process is repeated for a particular count of the images. If both match, then user is authenticated. On the android application, this module restricts the limit of the email id to only one and hence only one email id is allowed per application. At the same time, this module is also responsible to verify easy-to remember password with user entered password.

**6. Database:** The database server contains several tables that store each user information and including the count of the image. It also contain dynamic tables which contains information such as which user is currently logged in. On the android application, it stores only one email id and an initial password.

**7. Encryption:** This module uses AES encryption algorithm to encrypt the easy-to-remember password. On the android application, this module decrypts, the received password and display it to the user. 3.8. Android application: This module consists of two phases one is registration and other is login phase. In registration phase user need enter the email id and an initial password. This email id is then stored in the database (SQLite). In login phase user need to enter the email id and an initial password. The user will request for then token through this application. Upon receiving the token from server application will display it. Next, it will request for easy-to-remember password. Upon receiving this encrypted easy-to-remember password it will decrypt it and display it to the user.

**Algorithms**

Following two algorithms are used to generate a random login indicator / token.

**1. Fisher-Yates Algorithm:** This algorithm is used to randomly shuffle an array of data. Based on the requirement the first input to algorithm is an array consisting of eleven or more alphabets. The algorithm will shuffle this input array and generate a new array consisting same elements but at a new randomly shuffled indices. This newly generated array is than given as input to the next algorithm for further processing. Similarly, a second input to the algorithm is an array consisting of seven or more numbers. The rest of the processing of this array is similar to that of the first one.

**2. Reservoir Sampling Algorithm:** The output of the above algorithm is given as input to this algorithm. This algorithm is used for choosing a sample of item / items from a large set of items. As, the randomly shuffled array of eleven or more alphabets is given as input to this algorithm it will randomly select only one element from the subsequent set. Similarly, as the second input to the algorithm is a randomly shuffled array of seven or more alphabets it will randomly select only one element from the subsequent set. Than this chosen elements are that is one alphabet and one number are clubbed together and sent to android application. For example, if A is randomly choose

alphabet and 7 is randomly chosen number than they are clubbed together as A7 and sent to the application.

**3. Advanced Encryption Standard Algorithm:** AES algorithm is used to encrypt the easy-to-remember random password being sent to user's mobile application. It can be 128 / 192 / 256 bit.

It has following steps:

STEP 1: Derive the set of round keys from the cipher key.

STEP 2: Initialize the state array with the block data (plaintext).

STEP 3: Add the initial round key to the starting state array.

STEP 4: Perform nine rounds of state manipulation.

STEP 5: Perform the tenth and final round of state manipulation.

STEP 6: Copy the final state array out as the encrypted data (cipher text).

## III. RESULTS AND DISCUSSION

- Introducing a Graphical authentication system, based on graphical passwords to resist shoulder surfing attacks.
- With a one-time valid login indicator and circulative horizontal and vertical bars covering the entire scope of pass-images, system offers no hint for attackers to figure out or narrow down the password even they conduct multiple camera-based attacks.

## SYSTEM ARCHITECTURE

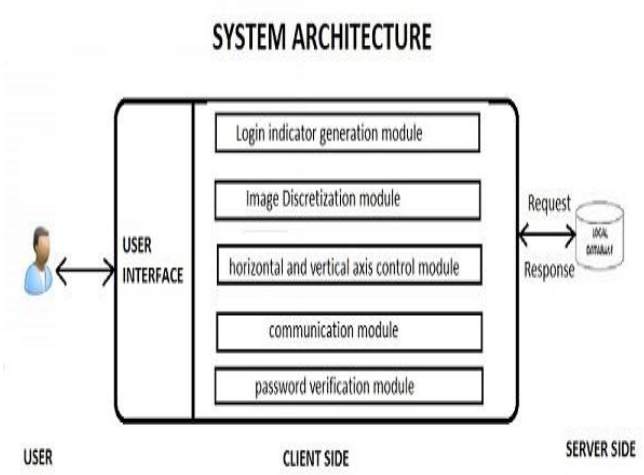


**Figure 2**

## IV. CONCLUSION

- The proposed system is a novel and easy-to-use graphical password authentication system, which can effectively alleviate shoulder-surfing attacks.
- In addition, system can be applied to any authentication scenario and device with simple input and output capabilities.
- The survey data in the user study also showed that system is practical in the real world.
- The proposed system is an image-based authentication system which is resistant to shoulder surfing attack. It generates a one-time random login indicator per image which eliminates the need to memorize the complex passwords. An android application is created where the user receives their random password for logging in. In addition to this, the random password is encrypted due to which it is unable to compromise the password when it is being sent to the user. Hence, with the use of proposed system user can log in to their confidential, personal accounts in public places without exposing their passwords to shoulder surfing attackers. The proposed system will provide a greater freedom for users to

authenticate themselves in a vulnerable environment.

## V. REFERENCES


[1]. S. Sood, A. Sarje, and K. Singh, "Cryptanalysis of password authentication schemes: Current status and key issues ", in Method and Models in Computer Science, 2009. ICM2CS 2009. Proceeding of International Conference on, Dec 2009

[2]. S. Gurav, L. Gawade, P. Rane, and N. Khochare, "Graphical password authentication: Cloud securing scheme," in Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on, Jan 2014

[3]. I. Jermyn, A. Mayer, F. Monrose, M. Reiter, and A. Rubin, "The design and analysis of graphical passwords," in Proceedings of the 8th conference on USENIX Security Symposium-Volume 8. USENIX Association, 1999

[4]. S. Wiedenbeck, J. Waters, J. Birget, A. Brodskiy, and N. Memon, "Passpoints: Design and longitudinal evaluation of a graphical password system," International Journal of Human-Computer Studies,vol.63

[5]. S. Brostoff and M. Sasse, "Are passfaces more usable than passwords? a field trial investigation," PEOPLE AND COMPUTERS

[6]. T. Kwon, S. Shin, and S. Na, "Covert attentional shoulder surfing: Human adversaries are more powerful than expected," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 44

[7]. M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd, "Reducing shoulder-surfing by using gaze-based password entry,' in Proceedings of the 3rd symposium on Usable privacy and security. ACM, 2007

[8]. L. Wang, X. Chang, Z. Ren, H. Gao, X. Liu ,and U. Aickelin, "Against spyware using captcha in graphical password scheme," in 24th IEEE International Conference on Advanced Information Networking and Applications. IEEE, 2010