Web - Based Authentication of Users using Image

F.M. Mohammed Farooq Abdulla [#], B. Raghu [*]

Department of IT, Sri Ramanujar Engineering College, Kolapakkam, Chennai, Tamil Nadu, India

mdfarooqabdulla@gmail.com
raghubalraj@gmail.com

***Abstract* - Access to computer systems is most often based on the use of alphanumeric passwords. However, users have difficulty remembering a password that is long and random-appearing. Instead, they create short, simple, and insecure passwords. Pictorial or Image passwords have been designed to try to make passwords more memorable and easier for people to use and, therefore, more secure. Using a pictorial or image password, users click on images rather than type alphanumeric characters. We propose an approach to improve the security of these systems relies on recognition-based, rather than recall-based authentication. We examine the requirements of a recognition-based authentication system and propose, Familiar Image Password System (FIPS) which authenticates a user through their ability to recognize previously seen images. FIPS is more reliable and easier to use than traditional recall-based schemes, which require the user to precisely recall passwords or PINs. Furthermore, it has the advantage that it prevents users from choosing weak passwords and makes it difficult to write down or share passwords with others.**

***Keywords:* Client-server, Knowledge-based systems, user authentication, user authentication through image recognition, recognition-based authentication.**

I. INTRODUCTION

User authentication is a central component of currently deployed security infrastructures. A key area in security research is authentication, the determination of whether a user should be allowed access to a given system or resource. Traditionally, alphanumeric passwords have been used for authentication, but they are known to have security and usability problems. Today other methods, including graphical passwords, are possible alternatives. This paper reports on research aimed to design a new kind of Familiar Image Password System, empirically test its usability, and compare it to alphanumeric passwords. The significance of this research is the provision of a flexible image password system with extensive human factors data to support it. We refer to the security and usability problems associated with alphanumeric passwords as "the password problem" [1]. The problem arises because passwords are expected to comply with two fundamentally conflicting requirements:

1) Passwords should be easy to remember, and the user authentication protocol should be executable quickly and easily by humans.

2) Passwords should be secure, i.e., they should look random and should be hard to guess; they should be changed frequently, and should be different on different accounts of the same user; they should not be written down or stored in plain text.

II. BACKGROUND ON PASSWORDS

A. Problems with Alphanumeric Passwords:

The password problem arises largely from limitations of humans' long-term memory (LTM). Once a password has been chosen and learned the user must be able to recall it to log in. But, people regularly forget their passwords. Decay and interference explain why people forget their passwords. Items in memory may compete with a password and prevent its accurate recall. If a password is not used frequently it will be even more susceptible to forgetting. A further complication is that users have many passwords for computers, networks, and web sites. The large number of passwords increases interference and is likely to lead to forgetting or confusing passwords.

Users typically cope with the password problem by decreasing their memory load at the expense of security. First, they write down their passwords [1]. Second, when they have multiple passwords, they use one password for all systems or trivial variations of a single password [2]. In terms of security, a password should consist of a string of 8 or more random characters, including upper and lower case alphabetic characters, digits, and special characters. As a result, users are known to ignore the recommendations on password choice. Two recent surveys have shown that users choose short, simple passwords that are easily guessable, for example, "password," personal names of

family members, names of pets, and dictionary words [3],[4]. To users the most important issue is having a password that can be remembered reliably and input quickly. They are unlikely to give priority to security over their immediate need to get on with their real work.

B. Why Graphical Passwords May Be Better

Most graphical password systems are based on either recognition or cued recall. In recognition-based systems the user must recognize previously chosen images from a larger group of distractor images. The decision is binary: either the image is known (recognized) or not known. In cued recall password systems users must click on several previously chosen areas in an image, cued by viewing the image. Both types of systems may have memory advantages over alphanumeric passwords. Alphanumeric passwords are based on pure recall (presuming the user has not written the password down). It is known that recognition memory is better than unaided recall [5]. Furthermore, psychological studies show that images are recognized with very high accuracy (up to 98 percent) after a two hour delay, which is much higher than accuracy for words and sentences [6]. In addition, it has been found that error in recognition of images is only 17 percent after viewing 10,000 pictures [7]. Other psychological research on images has shown that people can remember detailed visual information in natural scenes [8].

C. Picture Password

The visual login techniques described above face two main problems. First, due to screen size limitations, the size of the alphabet is smaller compared with traditional alphanumeric passwords, resulting in a weaker mechanism. Second, the user must select and remember a new set of images or image areas periodically whenever a password expires, which raises the level of difficulty for a user, especially if done within the context of the previous image set. Picture Password was devised to overcome both of these problems. As with textual password authentication mechanisms, Picture Password uses elements of an alphabet to form a password entry of a given length. However, instead of the user having to remember a string of random-like alphanumeric characters, the sequence of images that form a passcode must be recalled and selected. Moreover, an image sequence that has some meaning or is of interest to the individual user (e.g., images of sport team logos in order of preference) can be used. If forgotten, the sequence may be reconstructed from the inherent visual cues.

D. Web-Based Authentication Method

The methods to be discussed in this section are Internet based client-server model as shown in fig.1. The web-based authentication system consists of two parts: the authentication system and other one is allowing user to access. The image is selected by the users from list of image available in server system. Any authorized user, who has access to server, can generate image. The distribution can use any kind of network transmission such as FTP, e-mail etc. Once image is distributed to externally, client can access to authentication web page to get verification of image.

III. PROPOSED WORK

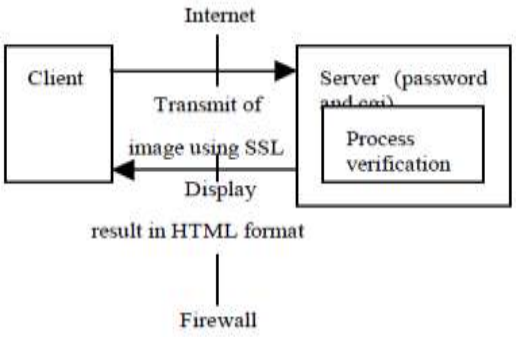

Fig.1. Client – Server Model

The secure web server system has its own security policy to protect against unauthorized use of it, and also the firewall to add network security. The server assumed to be located in a safe location for physical security. The server should be secure enough to install authentication system inside of it. The process of verification would be started with transferring image to the server. Once the image is uploaded, server uses its private key to detect image. There could be an attack while transmitting image file to the server, so the extra encryption can be used to hide image. There is a concept called SSL (Secure Socket Layer), which is the transport layer over TCP/IP network to provide authentication of server, client and encryption of message. Programming API like JAVA provides its own interface of SSL that will be used in the development of the system. This fulfils four major security aspects, which are the confidentiality using encryption, the integrity using watermark, the access control using pass word and the physical security of using server [9]. The detection process is an inverse function to check each pixel's LSB. If there is a difference in any pixel, the server will generate warning message that the image may have been modified or damaged. The block containing false pixel will be displayed instead of specific pixel since attacker may use this information to find out the binary function. All these information will be generated in to HTML format. The diagram of figure 2 displays the overall procedure and the structure of the authentication and access of data.

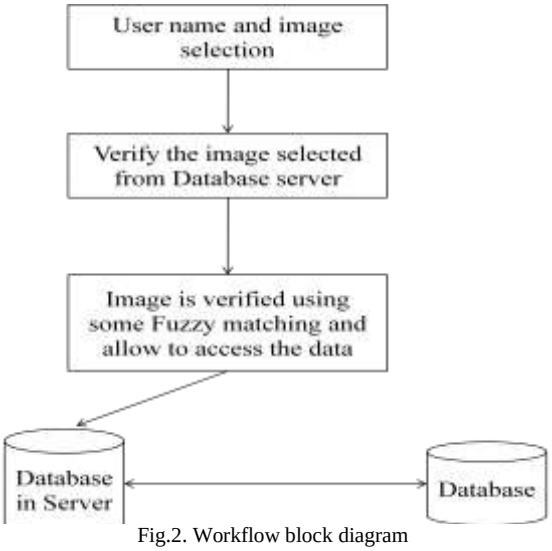

Fig.2. Workflow block diagram

For each image, the database stores the location and size of the dominant colors for an area in a quad-tree representation. The reason for using this type of index is primarily for optimization purposes: it turns out that with our quad-tree segmentation algorithm, a 100 K-pixel image can be approximated by only a few kilobytes worth of database tables. The image index itself is stored in Monet, a main-memory database engine [10]. All image structures are laid out in main memory in such a manner that we do not have to worry about disk I/O performance. We run the database on a 32-processor, 64 Gb main-memory machine, which is capable of storing a color index for more than 1 M images. Note that at the moment (by far) we do not use the full capacity of our server's memory. The quad-tree representation of the image is not used for the actual search process itself. Instead, when the database is accessed for searching and a number of fore- and background colors are presented, the quad trees are converted into structures through the help of a 128x128 image matrix: relevant colors are copied from the quad tree into the image matrix. Afterwards, the image matrix is processed to extract the relevant image structures or shapes, which are stored in Monet temporary tables.

Once a multi-spot query is formulated, the query is sent to the search engine for processing. In the database, a query processor parses the query; it derives the search parameters and drives the search operation by sending search requests to the actual database. Database searching happens in several phases. First, our database index is combined with the user's selected colors, and the structures are synthesized from the quad trees. Based on these structures, gravity points are calculated, which are used for the background embedding (or enclosure) tests. Finally, all remaining structures are used for a multi-spot match and all foreground structures are discarded that do not satisfy the user selected multi-spot. The reason for splitting up the search process in several phases in a pipeline is for optimization purposes. We try to prune away as many structures as possible in simple and cheap tests before applying expensive operations on the intermediate results. Also, caching techniques may help to reduce processing time by keeping intermediate search results available.

In the first phase the quad-tree color index is traversed and, with the user selected colors, organized in structures. For each image considered, our search engine creates a data structure for all *closed* regions in either the fore- or background colors. The created structures are made available in our database for further processing. Once the structures have been created, our search engine calculates the horizontal and vertical gravity points of each structure. These gravity points are the initial main point of interest for the structure and are used for testing the embedding of the foreground in the background structures. Note that the gravity point here does not have to be the real gravity point of the object: if the desired object is occluded or overlapped by another object in almost same color (*e.g.* a yellow and green balloon on green fields), the calculated gravity may be located at a different location due to the influence of the overlapping object. Although this may seem wrong, the effects are limited: the gravity points are, in this phase, only used for testing the embedding which does not change. Next, the foreground gravity points are tested for their embedding in the background structures. For each image, all foreground gravity points are analyzed by scanning over the background structures once. Structures that do not satisfy the embedding constraint are discarded. At the final step, all foreground structures are interrelated with each-other. Currently two types of shapes can be detected: gravity points and ellipsoids. The computed gravity points are used to correlate gravity points of multi-spots. The search engine adjusts the distances and angles of a multi-spot match in the original image to what is available in the considered image. For example if the distance between two spots in the original image is do, while the distance in the considered image is dc, a multiplication of m = dc /do is used to match the other points in the multi-spot. Likewise, the search engine also calculates a constellation of the multi-spot m and α. This constellation represents the rotation of the multi-spot in other images. To allow for some degree of *fuzzy* matching, the user can supply a number of parameters that determine maximum offsets to both m and α. Recall that the calculated gravity points are not correct. When an object is occluded by another similarly colored object, the calculated gravity point is shifted from the desired gravity point. This implies that candidate structures may be dropped while they would have

matched the criteria. This problem is reduced by applying gravity point matching only to distinctive objects in the query. We will work on this problem for future versions of our search engine. Other shapes, like ellipsoids, are matched differently. Since there can be many places where to position an ellipsoid even when m and α, are restricted by the user, only a single instance of an ellipsoid is considered. First, a center point for the ellipsoid is calculated based on earlier matched gravity points. Based on the center point m and α, the constellation of the ellipsoid, an internal representation of the desired ellipsoid is calculated. This internal representation of the ellipsoid is matched with the foreground structures. When a sufficiently large portion of the ellipsoid matches, the selected area is approved as a match.

IV. DISCUSSION

The proposed system can be used where the content of image is valuable, which requires it to be ensured in the distribution that the copy is identical to original. For example, medical image requires great integrity of content since any change in image might affect the diagnosis even it is small amount. This could be the further research area to provide efficient image authentication tool for medical images. The thread hold of diagnostically acceptable distortion level is the key issue in medical images. Using web-based authentication tool like the developed system may be used without affecting diagnosis.

V. CONCLUSION AND FUTURE WORK

In this paper, web based image authentication was investigated. There are advantages of using server-based authenticator since it can cope with distribution of public key and using extra security provided within server. The Familiar Image Password System (FIPS) is efficient for authentication of content whether it is altered or not. The result showed that this system could detect most of modification to content of image. The more secure and accurate system can be achieved by using some degrees of fuzzy and multi-spot image recognition.

Previous research recognized the weaknesses of knowledge-based authentication schemes (in particular password-based computer logins). So far, however, most of the proposed solutions have been based on technical fixes or on educating users. Neither of these addresses the fundamental problem of knowledge-based authentication systems, which is that the authentication task is based on precise recall of the secret knowledge.

REFERENCE

[1] Adams, A. and Sasse, M.A. (1999). Users are not the enemy. Communications of the ACM 42, 12, 41-46.

[2] Birget, J.C., Hong, D., and Memon, N. (2003). Robust discretization, with an application to graphical passwords. Cryptology ePrint Archive. http://eprint.iacr.org/2003/168 accessed January 17, 2005.

[3] Brostoff, S. and Sasse, M.A. (2000). Are Passfaces more usable than passwords: A field trial investigation. In McDonald S., et al. (Eds.), People and Computers XIV - Usability or Else, Proceedings of HCI 2000, Springer, pp. 405-424.

[4] Brown, A.S., Bracken, E., Zoccoli, S. and Douglas, K. (2004). Generating and remembering passwords. Applied Cognitive Psychology, 18, 641-651.

[5] Norman, D.A. The Design of Everyday Things. Basic Books,New York, NY, 1988.

[6] Shepard, R.N. Recognition memory for words, sentences, and pictures. Journal of Verbal Learning and Verbal Behavior 6, 156-163.

[7] Standing, L.P. Learning 10,000 pictures. Quarterly Journal of Experimental Psychology 25, 207-222.

[8] Hollingsworth, A. and Henderson, J.S. Accurate visual memory for previously attended objects in natural scenes. Journal of Experimental Psychology – Human Percpetion and Performance 28 (2002), 113-136.

[9] William Stallings (1999): Cryptography and Network Security 2nd edition, p 23-24, Prentice-Hall, Inc. ISBN 0-13-869017-0.

[10] P. A. Boncz and M. L. Kersten. MIL Primitives for Querying a Fragmented World. VLDB Journal, **8**(2):101-19, 1999.