Improving Security with Two-factor Authentication Using Image

Supakit Mahitthiburin*
Department of Information Technology, Faculty of Information Technology, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand

Sirapat Boonkrong
Department of Data Communication and Networking, Faculty of Information Technology, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand

* Corresponding author. E-mail: pz@phoenikz.com
Received: 14 November 2014; Accepted: 26 November 2014; Published online: 2 February 2015

Abstract
Most of today's authentication mechanisms only involve the use of username and password. This is known as one-factor authentication. Even though the method is the most commonly used, there have been problems with its security. This research, therefore, aims to design and develop a two-factor authentication that can improve the security of authentication mechanism. The proposed method contains an extra factor of authentication in image. Combining with the use of username and password, users are allowed to draw anything they want to make up another factor. This makes it different from other existing image authentication mechanisms which force users to choose pre-defined images. The proposed method was tested with thirty users to determine the most appriate security threshold. The performance of the system was evaluated, and the satisfactory survey was also taken.

Keywords: *Two-factor authentication, Image authentication, Authentication*

1 Introduction

Currently, it cannot be denied that Internet and computer systems have become an integral parts of many organizations, educational institutes and people's lives. In order to preserve security and privacy, authentication is required before entering any system. Unfortunately, the most commonly used method of authentication is the use of username and password [1]. With this method, there is no way of telling whether the person logging in is really the real owner of the identity. Hence, authentication cannot be done accurately [2].

Another problem with the password-only authentication is the way users choose their password. Most users tend to select passwords that are easy to guess, which only include letters and numbers. This can easily lead to such attack as password dictionary attack [3]. Moreover, with one-factor authentication like this, if anyone happens to know the password, he or she will then be able to log into the system as the owner of the password, too.

In order to enhance the security of the authentication system, another factor of authentication is needed. Such method is known as two-factor authentication. One popular additional factor is image.

Our research, therefore, has a purpose in designing and developing a two-factor authentication method, with password being the first factor and image being the second. The introduction of image as an extra factor is to reduce the risk of authentication by non-owner of the login credentials [4].

Please cite this article as: S. Mahitthiburin and S. Boonkrong, "Improving Security with Two-factor Authentication Using Image," *KMUTNB Int J Appl Sci Technol*, Vol.8, No.1, pp. 33-43, Jan.-Mar. 2015, http://dx.doi.org/10.14416/j.ijast.2014.11.003

In this research, we aim to develop a method where a password as well as a picture hand-drawn by the user are needed during the login process. This is different from other existing image authentication mechanisms [5,6], which only allow users to choose a pre-defined picture rather than letting them draw by themselves.

Once the two-factor authentication is developed, the most appropriate security and satisfactory thresholds will have to be found so that the system satisfies the users while the security is not compromised.

The rest of the paper is organized as follows. Section 2 explains the existing theories that support the proposed method as well as the related research in the subject. Section 3 provides the design of the authentication system. Section 4 explains the testing and results. The conclusion is then given in Section 5.

2 Related Work

2.1 *Background knowledge*

Authentication is a method for verifying users of computer systems. The system will verify username and password whether they are correct. The main objective of authentication is to specify who the person logging in to the computer system is One-factor authentication is the most used method today. That is, when logging into a system a user is only required to enter his or her username and password. If the entered information is correct, the user will then be allowed to access the system.

Unfortunately, one-factor authentication of this type, also known as password-only authentication, cannot withstand many of today's existing threats [7], which include password dictionary, information stealing and man-in-the-middle [8,9]. It is, therefore, necessary to introduce additional factor of authentication so that security can be increased [10]. One popular component that can be used as a second factor of authentication is image.

2.2 *Existing research*

It has been found and stated earlier that the drawback of the traditional login system or password-only authentication is the use of simple passwords that are easy too crack. Currently, image authentication is one of the best alternatives as it is one step higher than general password authentication. Image authentication is easy to remember but hard to guess randomly [11] so it is another reason why we have decided to choose image authentication as this research subject.

Figure 1: Existing image authentication methods.

Image authentication has not really been as widely used in comparison with password-only authentication. However, the idea has been taken into consideration and applied for two-factor authentication. [12] suggested that image authentication could be used easily and fast with high accuracy. It could also be memorized easily. As for numeric or alphabetical authentication, too tough password could be too tough to memorize at the same time, causing the user to write it down or record it in a paper or computer which could allow any unfaithful person to access to the account illegally. Therefore, image authentication could enhance channels for security for the account owner as well.

Figure 1 shows examples of existing image authentication systems. In order to use the system, during the registration stage, users would have to choose one or two pre-defined images. When logging into the system, they would have to select the correct images before being allowed to access the system.

The main disadvantage of this method of authentication is that it would not be difficult for an adversary to carry out shoulder-surfing attack or even to make a guess. If successful, the user could be impersonated.

Due to the above weakness, we decided to design and develop a system which would allow users to draw an image of their choice as a part of the registration and authentication processes. That means there would be no need to assign or pre-define any images beforehand. This ide ais actually supported by [9,13] in that it would make the system more secure.

According to the study in theories and researches

related to authentication, it could be seen that even though one-factor authentication system was famous but it allowed attack to happen easily [14,15]. This research therefore designed and developed Two-factor Authentication by allowing users to be able to create an image as an image code independently. This research was different from other researches which allowed users to pick only determined image code or to pick any prepared image as a passcode. For safety and risk reduction of authentication, the researcher conducted the test by using an image drawn by user for comparison to seek for an appropriate threshold for user's login. The researcher conducted research to seek for such appropriate threshold for further use and development of image authentication system.

3 Research Methodology

3.1 *Design of two-factor image authentication system*

The two-factor image authentication system has to begin with the design of a registration process, which will then be followed by the login process design.

3.1.1 Registration process

Before being able to use the authentication system, a user would need to register with the system so that necessary credentials would exist for the actual login process.

Figure 2 depicts the proposed registration process. It can be seen that the process begins with having users enter their personal information, such as name, address and citizen ID. The citizen ID is then checked and verified for any duplication with the existing ones in the database.

Next, the user will be asked to enter his or her username and password as the traditional one-factor authentication. Again, any duplication of username will need to be verified. If the database already contains the same username, the user will be asked to enter a new one. This is to ensure that there are no repetitive usernames. Once this step is done, the user will then be allowed to hand-draw an image in the provided area. The image will be used as the second factor of authentication.

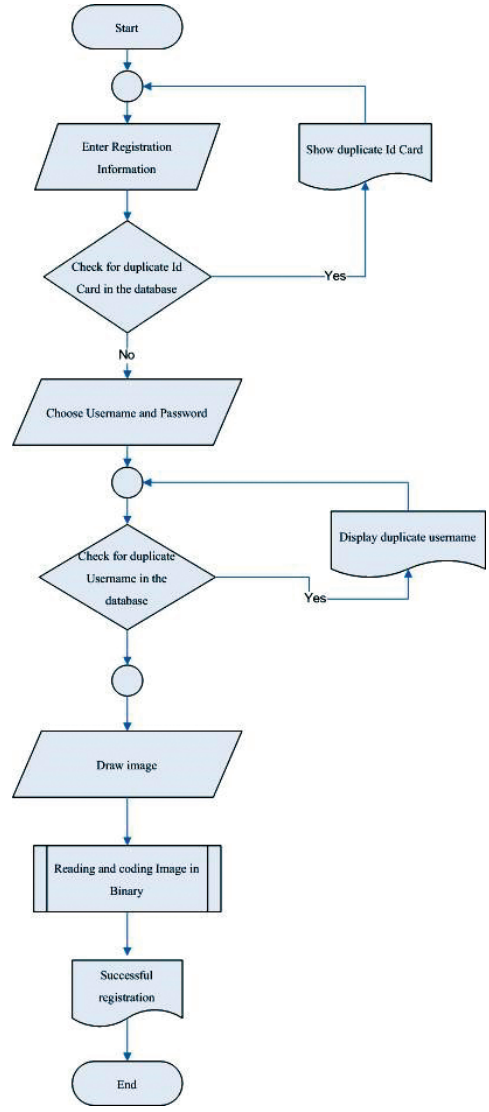

Figure 2: Registration process.

In our proposed system, we designed a drawing area to be 80 x 80 pixels. This was because we would like to restrict the area as well as to not put a lot of burden on the memory and the processor of the system.

Once the user has drawn an image inside the provided box, the system will convert the image information into the form of array of bytes. This would be then converted into the form of string to be saved in the database for security and ease of storage. This is how an image would be stored on the database. The process can be seen in Figure 3.

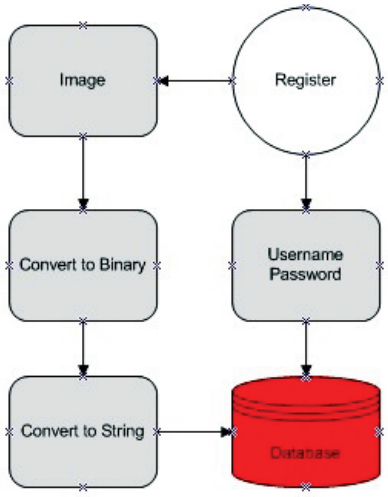

Figure 3: System information storage.

According to Figure 3, the user had to complete personal information, including name, family name, username, password, ID card number, and image code. After completing those details, the system would then store information into the system first. As it was required to save many images into the database, saving files in the form of image would cause delay of recalling and excessive space used. Due to the fact that for the next operation the system would compare image by using bytes to fasten the processing of information, we converted image code into the form of bytes of array. As database was limited for information storage due to inability to determine length of binary which could cause slow processing, binary was also converted to save in the database in the form of string which could be recalled for the next operation, the login process. This is shown in Figure 4.

3.1.2 Login proess

For the login process, the design was divided into two steps. The first step would be done by the traditional login process of username and password. The second step would, of course, be done with the image, hand-drawn by the user. This is considered the second factor of our proposed authentication system.

Figure 4 illustrates an overview of how the login process works. It can be seen that a user has to enter his or her username and password into the system. He or she will then have to draw an image.

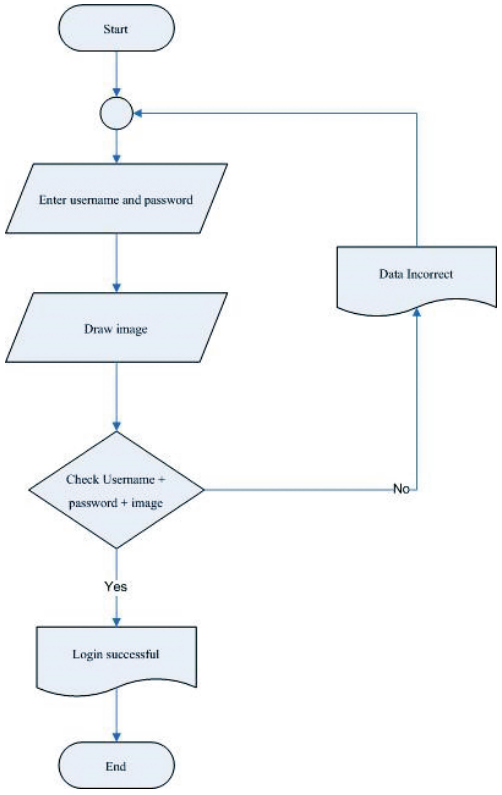

Figure 4: Login process.

Once all the information has been filled in, the system will verify whether or not the username, password and the image match the ones in the login database. If so, the user will be allowed to access the system. If not, the user will have to try to login again.

It needs to be mentioned here that it has been decided that the login process does all the verification in one single step. In other words, instead of entering username and password, and verifying them before allowing users to draw an image, users will enter all the information at once before the verification. This is because checking the credential step-by-step would make the authentication system less secure. That is, in the step-by-step verification, if the username and password were incorrect, the user would know that the problem of logging in would be here. If the image were incorrect, the user would also know that the problem lied here. However, if everything were verified at once, and the login attempt failed, there would be no way the user would know which part of the credential was incorrect.

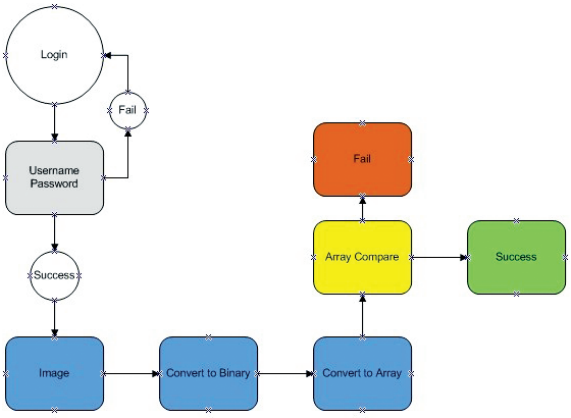

Figure 5: Steps of system use.

Figure 5 provides a little more in-depth detail of the verification process of our proposed authentication mechanism.

The login begins with the entering of username and password before an image is drawn. We have followed the usual process [10,16] of checking the username and password. Therefore, the focus here will be the image verification only.

From Figure 5, it can be seen that the image verification works as follows. First of all, after an image is hand-drawn by the user, it is converted into binary, which is then put into the form of array. Secondly, the existing image information from the authentication database will go through the similar process, which is binary conversion and array conversion.

The two pieces of image information, namely the one done at the login process and the one from the authentication database, will be compared to one another. If the differences between the two are below the specified threshold, the image will be accepted by the authentication system. On the other hand, if the differences between them are over the threshold, the image will be deemed as incorrect and will be rejected.

Note that how the threshold is determined will be discussed later on in the paper.

3.2 *Detail of the system processes*

In order to understand the processes of image conversion, information storing and image verification better, this section takes us through the detail of those processes.

```
string basestring = Convert.ToBase64String(ImageToBinary(fileName));
basestring | "/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRofHh0aHBw
```

Figure 6: Information Storage in the form of Basestring.

3.2.1 Image storing process

Table 1 shows how information is stored in the database during the registration process. It is divided into two parts. They are storing username and password, and storing image. We will look at Part 2, image storing, in more detail here.

Table 1: Information storage command of the system

1.	`aGetData = new GetData();` `aGetData = GetDataManagement.GetUser (this._txtUserName.Text.Trim(), this._txtUserPassword.Text.Trim());`

2.	`string fileName = "D:\\Authen\\" + DateTime.Now. ToString ("yyyyMMddHHmmss") + ".jpg";`
	`pictureBox1.Image.Save(fileName, System.Drawing.Imaging.ImageFormat.Jpeg);`
	`string basestring = Convert.ToBase64String(ImageToBinary (fileName));`

From Table 1, it can be seen that Part 2 provides the steps taken for the storage of image drawn by user in the space provided during the registration. For this particular example, we stored them in the D:\\Authen\\ directory. Here, the images were stored in the order that they were generated, i.e., by year, month, day and time.

After that, the system would save image into the database by converting image into string. This was done in order to save space and to facilitate information storage of the database. By using the function, `ToBase64String`, an image can be converted into string as seen in an example in Figure 6.

Apart from the reason of saving storage space mentioned above, there is one other reason why we have decided to convert image to string for our two-factor authentication mechanism. That reason is, of course, security. By storing an image in the string format, even if the database is compromised, the adversary will not be able to see an original image at all. Moreover, a string encoding process could be use to further protect the information.

Our chosen process of Basestring conversion works as follows.

Text content	M							a							n						
ASCII	77							97							110						
Bit pattern	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Index	19				22				5				46								
Base64-encoded	T				W				F				u								

Figure 7: Information Conversion in the form of Basestring.

Figure 8: Number of byte gained from binary conversion.

According to figure 7, an image drawn by user would be comparable with text content section. An image file would be divided into sub-files, three bits each, by using a command, known as Filestream, to convert it into binary with non-assessable length in the part of the bit pattern.

The ASCII value could then be obtained from the calculation of the binary converted value by dividing the binary value into eight bits, and finding an index from it which had to be reduced to six bits. After getting such index, the last step would be to find Base64(String) by taking index for comparison by arranging such index from information as follows "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefgh ijklmnopqrstuvwxyz0123456789+/".

The illustration in Figure 7 shows that the first index obtained was 19. This means that we had to arrange the aforementioned information nineteen times from left to right by counting the first one from B to get the value at T exactly. This is the Base64-coded value.

In this part, it was the explanation of only one Basestring. From Figure 8, there were so many values which were uncountable because the more binary length, the longer the Basestring. In order to achieve this, we wrote a function that would automatically process such objective [16,17]. Once an image has been stored in the database, it is the end of the registration process.

3.2.2 Image verification process

The next process that needs to be explained in detail is the verification of images during the login process.

The process begins with the process shown in Table 2.

Table 2: Command for recalling string of image stored in database

1.	`public static byte[] ImageToBinary(string _path) FileStream fS = new FileStream(_path, FileMode.Open, FileAccess.Read);`
2.	`byte[] b = new byte[fS.Length]; fS.Read(b, 0, (int)fS.Length); fS.Close(); return b;`

According to Table 2, once a user entered his or her login credentials together with his or her hand-drawn image into the system, the system would recall image associated to the user in the form of String from the database. This would then be converted to binary by using the FileStream command, as shown in Part 1 of Table 2.

For Part 2, the FileStream command would convert binary into byte for the image comparison and verification processes (explained later). The number of bytes used for comparison would be as shown in Figure 8.

The next process is the image comparison and verification process. Table 3 shows in Part 1 that the two images would be compared with one another. One would be from the authentication database and the other would be from the login process.

Table 3: Command for image byte comparison

1.	`static bool ByteArrayCompare(byte[] a1, byte[] a2)`
2.	`double div = a1.Length - a2.Length; if (div > 0 && div <= 50) { val = true; }`

Part 2 of Table 3 illustrates how the comparison and verification of the two images is to be carried out. We proposed that the image verification should be done by finding the difference between the entered image and the image in the database.

One particular value would have to be preset before the verfication was done. This value is known as the threshold value. If the difference between the two images is within the specified threshold, the user will be allowed to enter the system. If not, the image drawn by the user will be deemed as an incorrect image. Hence, the authentication fails. Figure 9 sums up the image verification process.

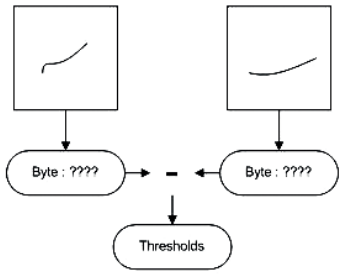

Figure 9: Image verification process.

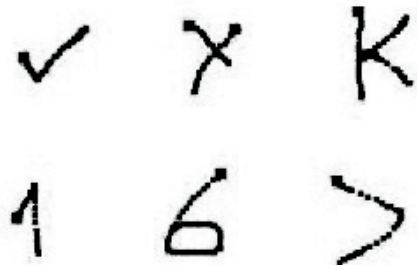

Figure 10: Example of Users' Images.

In the example shown in Table 3, the threshold value was set to have the value of fifty. In truth, the threshold value must be configured system-by-system. That is, we recommend that the threshold value must not be set to any number. It must be the number that satisfies users and still holds the security of the authentication system.

We show how an appropriate threshold of the proposed two-factor authentication system can be found in the next section.

4 Results

This section provides the results of our experiments. We divided our expermiments and tests into several parts. The first part was the preliminary satisfactory survey of the system. The second part was the one that would help us determine the appropriate threshold for the system.

For the experiments, user satisfaction was evaluated for the different values of threshold. There were thirty users participating in the experiments. During the test, different values of the system threshold were used, starting from threshold 0 – 1, which was then increased by the step of five units until the threshold was 0 -50. For each threshold, user satisfaction was recorded.

Moreover, after the appropriate threshold was found, we re-evaluated the user satisfaction in order to confirm that this value could really be used. The speed and performance of the proposed two-factor authentication was also tested.

4.1 *Finding appropriate threshold*

First of all, before the system could be used, an appropriate threshold needed to to be determined. In order to do so, the word "appropriate" should be defined.

From our point of view, an appropriate threshold is a value that is used in such a way that it satisfies the user when carrying out the two-factor authentication as well as maintains the satisfied security level.

Therefore, two experiments were carried out in order to find the appropriate value. The first was the user satisfaction [18]. The second was the correctness of image verification. Thirty participants took part in the experiments. The results were as follows.

4.1.1 User satisfactory evaluation

The evaluation of user satisfaction for the use of the image authentication part of the proposed two-factor authentication system had thirty participants. Each user designed image code by themselves with various levels of difficulty. Examples of images used during the test are shown in Figure 10.

As for the test of satisfaction level of image authentication system, during the test, the threshold value was adjusted starting from 0-1, then 0-5, and had been increased 5 levels each time until the value reached the 0 – 50 level. The reason that we stopped at the value of 50 was that we did not want to compromise the security of the system. In other words, if the threshold were over 50, it would be too easy for an adversary to fake an images. That is, images that were not alike to the existing one in the database would also be accepted by the system, i.e., the security would be easily compromised.

For the experiment, each user had to carry out the image authentication twenty times for each theshold value. Each time, the satisfaction level was recorded. The results of user's satisfaction for the different threshold values are shown in Figure 11.

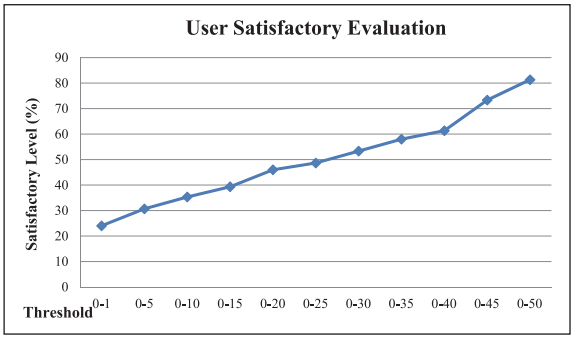

Figure 11: Level of user's satisfaction.

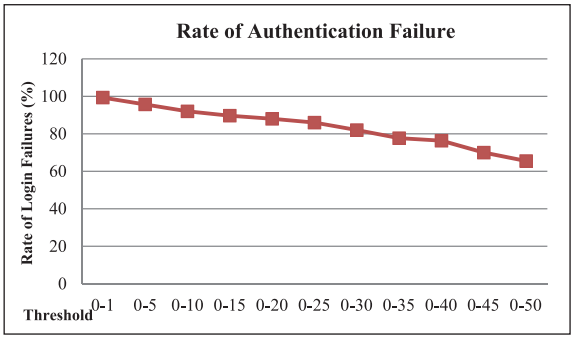

Figure 12: Rate of authentication failure.

The graph in Figure 11 showed that the level of satisfaction at the lower threshold value was low. That is, the level of satisfaction was only approximately 22% when the threshold was 0 – 1. This was because it was not easy to pass the image authentication process when the threshold value was low. In other words, the image would have to be almost an exact match to pass the image authentication. However, the satisfaction level had increased when threshold value increased. At the threshold of 0-50, the trial users had up to 81.33% satisfaction level.

4.1.2 Success rate evaluation

This is an important step for determining an appropriate threshold as we would not want a system where an exact match of an image was needed. This would be too difficult for users to login. On the other hand, we would not want a system that would match any image. This would be too easy for an adversary to impersonate.

Again, the threshold value was adjusted starting from 0-1, then 0-5, and had been increased 5 levels each time until the value reached 0 – 50. Also, thirty people participated in the experiment. For each threshold level, the user had to try to log in twenty times. The results of successes and failures were then recorded.

One point to be stated here is that in order to reduce the familiarity in continued access to the system of the trial users, we decided to stop the test for at least one day when the threshold reached the value of 0 – 25, and continued testing until reaching 0 - 50 on the next day. Figure 13 shows the results of the failure rate for the different thresholds.

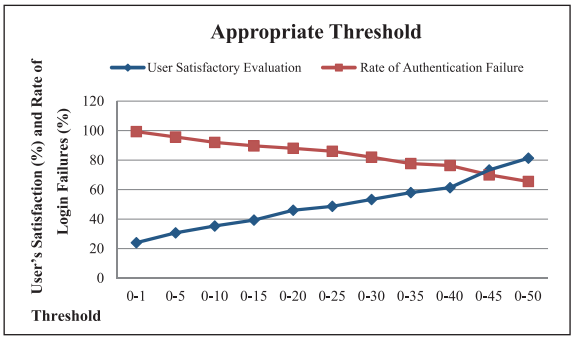

Figure 13: Appropriate threshold value.

The graph in Figure 12 shows that on average, the rate of login failure when the threshold had the value of 0 – 1 was approximately 99.33%. That means it was almost not possible for users to pass the image verification at the level of threshold. However, the rate of failure decreased as the values of threshold increased.

When the value of threshold reached 0-50, the failure had reduced to approximately 65.50%. This was because it was not as difficult to pass the verification as when the lower values of thresholds were used.

4.1.3 Appropriate threshold

At this stage, we had enough information to determine an appropriate threshold for the proposed two-factor authentication system, especially in the image authentication part.

In order to determine such value, the results of the user satisfactory evaluation and the rate of login failures were plotted in the same graph. At the point where the graphs crossed, the appropriate threshold

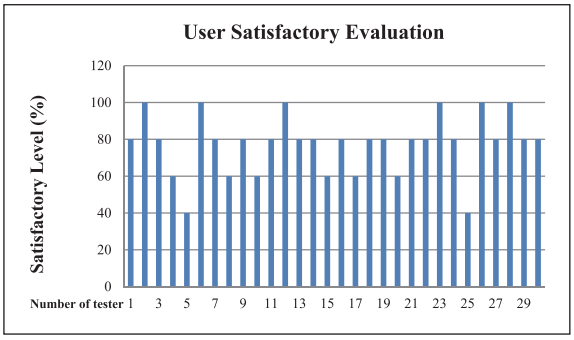

Figure 14: Satisfactory evaluation of threshold valued 0-45.

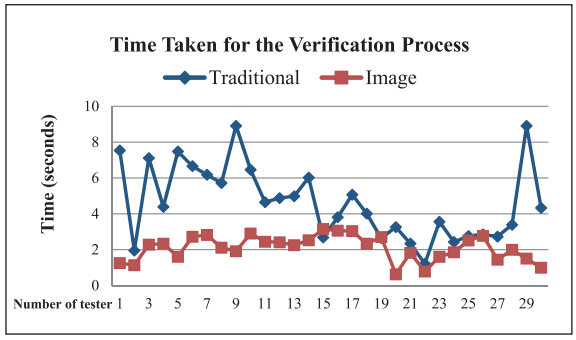

Figure 15: Time taken to complete the authentication.

value for this particular system would be determined.

We chose the crossing point to be the value of appropriate threshold because, at this point, the user's satisfaction and the success or failure rate of logins were both acceptable. In other words, users would still be satisfied while the authentication process would not be too easy to compromise.

After plotting the satisfactory graph and the login failure rate graph, Figure 13 is obtained.

From Figure 13, it can be seen that the thresholds from 0-1 to 0-50, the participants could access the system more as threshold increased. At the same time, level of satisfaction also increased.

We then took both graphs for comparison to seek for an appropriate threshold to be used. Consequently, both graphs were crossed at 0-45. It, therefore, could be concluded that the threshold between 0-45 was the most appropriate one for the use of the image authentication part for the proposed two-factor authentication mechanism.

4.2 *The satisfactory confirmation on the appropriate threshold*

Having obtained what would be thought of as the appropriate threshold for the proposed two-factor authentication system, we thought it would be necessary to evaluate the user's satisfaction on this threshold again. This was done to ensure that users would be happy enough when working with this threshold.

Again, thirty people participated in the evaluation. Each user was asked to carry out the image authentication process twenty times. The satisfactory levels were then recorded. The results are shown in Figure 14.

The graph in Figure 14 shows that the fifth and twenty-fifth users gave low satisfaction. Our observations suggested that the users appeared to use images that were more complex than other users.

Moreover, the average score of the user satisfactory level here was approximately 77.4% which would be acceptable [19]. Therefore, it can be claimed that at the previously determined threshold, users were satisfied and still happy with the image authentication process.

4.3 *System performance evaluation*

This section provides the result of system efficiency interms of speed. Here, we compared the speed of the traditional authentication system that used username and password with the proposed image authentication system [20].

It has to be noted that only the image authentication part of the proposed two-factor authetication was tested at first. After that the time taken for the whole of two-factor authentication will also be shown.

In order to obtain the results, thirty participants were asked to log into the system using the traditional method. The time was measured as soon as the first letter was entered into the system. The same group of people then carried out the image authentication where the timer started as soon as the participants began drawing an image. Figure 15 shows the time taken for the traditional authentication method and the image authentication method to complete the verification process when carried out by each participant.

The average time taken to complete the authentication process for both the traditional and image authentication methods can be seen in Table 4.

Figure 15 and Table 4 show that the time taken to complete the traditional authentication method that required username and password was approximately 4.63 seconds. This, on average, was longer than the time taken to complete the image authentication process, which took approximately 2.09 seconds.

Another thing that can be pointed out in Figure 15 is the fluctuation of the time taken to complete the authentication process. It can be seen that the time taken to complete the traditional authentication process varied more than that of the image authentiation from user to user.

This implies that the traditional one-factor authentication system was also subject to user's ability in typing which was varied, resulting in the longer duration of logging in than that of image authentication.

Table 4: Average time taken to complete authentication

	Traditional Method	**Image Method**
Time to Process (Seconds)	4.630	2.093

Let us now compare the time taken to complete the traditional one-factor authentication with the proposed two-factor authentication method that also required an image.

A similar experiment to the previous one was run and the times were recorded. The average time taken to complete the two methods can be seen in Table 5.

Table 5: Comparison of one-factor and two-factor authentication

	One-factor Authentication	**Two-factor Authentication**
Time to Process (Seconds)	4.630	6.724

It can be seen in Table 5 that with an additional authentication factor being added to the traditional system, approximate two seconds were also added to the time taken to complete the process. The time it took to complete the proposed two-factor authentication was longer than that of the traditional method because users were required to draw an image during the login process as well.

5 Conclusions

Authentication can be thought of the first line of defence for any computer systems and networks. It was, therefore, realised in this paper that a more secure authentication method would be needed.

This paper began with the analysis of the existing authentication methods, which included the traditional username and password method, and two-factor authentication methods, focusing on image authentication. It was pointed out the existing methods were not secure enough to prevent such attacks as credential stealing and impersonation.

We, therefore, designed and developed a two-factor authentication mechanism which allowed users to enter username, password and a hand-drawn image of their choice. This was what made our system different from the existing ones.

Next, in order to both satisfy users and to preserve the security of the proposed system, an appropriate value of threshold needed to be determined. Experiments were carried out, graphs of user's satisfaction and rate of login failures were plotted to find out that the most suitable value for this particular system was forty-five.

The efficiency of the proposed system was also tested against that of the traditional method. It was found that the proposed system took approximately two seconds longer. Even though this was the case, we believe that additional security always comes with the speed trade-off.

References

[1] A. P. Sabzevar and A. Stavrou, "Universal Multi-Factor Authentication Using Graphical Passwords," *IEEE International Conference on Signal Image Technology and Internet Based Systems*, 2008, pp. 625-632.

[2] H. AI-Assam, H. Sellahewa, and S. Jassim, "On Security of Multi-Factor Biometric Authentication," *Internet Technology and Secured Transactions (ICITST) International Conference*, 2010, pp. 1-6.

[3] G. Agarwal, S. Singh, and R.S. Shukla, "Security Analysis of Graphical Passwords over the Alphanumeric Passwords," *International Journal of Pure and Applied Sciences and Technology*,

2010, pp. 60-66.

[4] H. Gao, X. Liu, S. Wang, and H. Liu, "Design and Analysis of a Graphical Password Scheme," *Fourth International Conference on Innovative Computing, Information and Control. Taiwan*, 2009, pp. 675-678.

[5] F. Monrose and M. Reiter. (2005, Aug.). Graphical Passwords [Online]. Available: http://hornbeam.cs.ucl.ac.uk/hcs/teaching/GA10

[6] H. Gao, Z. Ren, X. Chang, and X. Liu, "A New Graphical Password Scheme Resistant to Shoulder-Surfing," *Cyberworlds (CW) International Conference*, 2010, pp. 194-199.

[7] S. U. Shah, F. Hadi, and A. A. Minhas, "New Factor of Authentication: Something You Process," *International Conference on Future Computer and Communication*, *Malaysia*, 2009, pp. 102-106.

[8] OWASP Testing Guide. (2008, Oct.). Testing Multiple Factors Authentication (OWASP-AT-009) [Online]. Available: https://www.owasp.org/images/5/56/OWASP_Testing_Guide_v3

[9] B. Schneier. (2013, Oct.). Schneier on Security [Online]. Available: https://www.schneier.com/news/type/written-interviews/

[10] S. Boonkrong, "Security of Passwords," *Information Technology Journal*, vol. 8(2), pp. 112-117, Jul - Dec 2012.

[11] P. Prapakittikul. (2012). The dangers that come with a password. [Online]. Available: http://www.thaicert.or.th

[12] 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*, pp. 102-127, 2005.

[13] P. Hom-anek. (2004, May). Compare Two-Factor Authentication Between using One Time Password (OTP) and Smart Card in conjunction with Public Key Infrastructure (PKI). [Online]. Available: http://www.acisonline.net/

[14] H. Kim, K. Lee, and Y. Jung, "A Graphical Password Based System for Small Mobile Devices," *International Conference on Convergence and it's Application*, Korea, 2013, pp. 156-746.

[15] W. Z. Khan, M. Y. Aalsalem, and Y. Xiang, "A Graphical Password Based System for Small Mobile Devices," *International Journal of Computer Science Issues*, Australia, pp. 145-154, 2011.

[16] F. Towhidi and M. Masrom, "A Survey on Recognition-Based Graphical User Authentication Algorithms," *International Journal of Computer Science and Information Security*, vol. 6(2), 2009.

[17] L. Wei, C. Yuanyuan, W. Boxiong, Y. Chunyu, and L. Jiannan, "Fast Method to Detect Particle Sizes of Objects in Binary Image," *Seventh International Conference on Image and Graphics*, China, 2013, pp. 265-268.

[18] A. D. Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann, "Touch me once and I know it's you! Implicit Authentication based on Touch Screen Patterns," *Media Informatics Group*, Austin, Texas, 2012.

[19] Y. Gong, "Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval," *Pattern Analysis and Machine Intelligence*, pp. 2916-2929, 2013.

[20] M. Rouse. (2013, Jan.). Single-factor authentication (SFA) is the traditional security process that requires a user name and password before granting access to the user. [Online]. Available: http://security.techtarget.com