Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

Use of Digital Image Watermarking to Enhance the Security of Graphical Password Authentication

Saeed Sadeghi*[1], Kooroush Manochehri†[2] and Mohsen Jahanshahi‡[3]

[2]Department of computer engineering, Amirkabir university of Technology, Tehran
[1,3]Islamic Azad University (Central Branch), Tehran, Iran

ABSTRACT

There are several techniques for implement an authentication system for computers that most commonly use the clear text password. One of the security problems is the use of a text password, the lack of choosing a complicated password by users due to forgetting, and being guessable and retrieved by attackers. One of the methods for passing text passwords is the use of graphical password authentication systems that increase the amount of forgetting passwords by using images instead of text characters. In this paper, the security challenges of using a graphical password are discussed. Then, explain a method for using the watermarking digital image for the authentication process and providing an algorithm suitable for watermarking and enhance the security of graphical password authentication system, and its quantitative and qualitative security parameters will be examined.

Keyword: Digital Image, Watermarking, Graphical Password, Computer Security, Authentication

ARTICLE INFO

Article history:
Research paper
Received 17, June 2020
Received in revised form 18, April 2021
Accepted 11 May 2021
Available online 01, June 2021

AMS subject Classification: 94A62.

*Saeid.sadeghi71@gmail.com
†Corresponding author: K. Manochehri. Email: kmanochehri@aut.ac.ir
‡mjahanshahi@iauctb.ac.ir

1 Introduction

So far, various mechanisms are presented to provide security and access level to information, based on all of which provide a level of user access and the identification of the requesting person for the use of the system or the information. There is a key logic in research in the field of computer system security, and this is the authentication that determines how much the user can access the system and resources. Many authentication methods have been introduced in computer systems that regarded three factors [4]:
Information: What does the user know? The authentication by information and knowledge that is authorized by the user resources.
Possess: What does the user have in control? The authentication by the key, password, or the information that has already been possessed to you.
Individual Characteristics: Who is the user? Authentication through biological and existential attributes such as fingerprints or eye cornea patterns.
Secure authentication methods usually consist of integration and simultaneous use of two or more factors raised above. It should be noted that many of these methods or the integration of these methods according to software and hardware implementation in many computer systems such as social networks, store websites, etc. dont have the economic cost, and these systems prefer to use conventional and simple authentication methods that using the text password and username in their software.
These text passwords can be a combination of Latin letters, numbers, and symbols, but unfortunately, most of the ordinary users use simple words because they can easily remember these passwords, or when using of complicated passwords, that are usually required from the system, they write down them. One of the methods of authentication in the proposed computer system is using a graphical password instead of a text password. Scientific research has shown that the images in the human mind are more memorized than text, as well as using a graphical password has not many security vulnerabilities of text passwords such as brute-force or dictionary [4], but using this method creates new challenges and vulnerabilities that new ways to resolve these challenges are presented in various papers. In this paper, we try to provide a method for authentication of graphical passwords using digital images watermarking algorithms to solve some vulnerabilities and security challenges such as shoulder surfing, or vulnerability due to non - use of secure connections such as MITM.
In section 2 of this paper has reviewed the methods of using graphic passwords presented in papers and similar works performed in this field. In Section 3, a new method and protocol for implementing a graphical authentication system will be presented, and we will provide the new watermarking algorithm in the digital images to create the desired protocol. In section 4, the numerical and quantitative parameters of this algorithm and the proposed method will be examined.

2 Related Works

Many papers and researches have been published in the area of creating new methods for providing secure graphical passwords in recent years, some of which are discussed below.
In the proposed method in the paper [6], the user should create a profile of images from a large number of images at the time of registration. In the time of authenticating and logging in, the user must choose an image of among the many displayed images, which he has already chosen for himself at the registration phase. In this proposed method, random art algorithms are used to create more complicated image categories at the time of logging in.
In the paper [3], a method named Pass Face has been suggested that the user must choose some human face images, and at the time of logging in, several images of the human face are displayed to the user, and the user must recognize his selected images. This process is done in steps in a few pages, and the user must recognize his images at each step. In the tests conducted in this method, each page consists of nine images, and the user must correctly identify the four steps sequentially. One of the most prominent problems of this proposed method is the time-consuming of authentication by the user and logging in.
The previous method was modified by Daviss changes in the paper [5] in order to solve the problem of time-consuming of logging time. In the proposed method, images are displayed only on a single page, but the arrangement of image selecting for logging in is essential.
In the paper [11], a method has been proposed to implementing a graphical password that partly prevents the vulnerability of seeing the selected image by the penetrative that may be near you, known as Shoulder Surf. In the proposed method, a large number of objects that are our image are presented to the user; among them, there is the number of pass-objects that the user has selected at the registration phase. The images are shown in thumbnails with a large number to the user to make it difficult to guess and distinguish by a hacker in the Shoulder surfing method.
In paper [9], another method has been proposed to eliminate the risk of Shoulder surfing. In this method, a large number of images are displayed to the user, which among them pass-objects are available for the user. Each image has a dedicated code that is displayed below. The user must type the code of each image, which is its pass-object, in the password box, and then logging in.
The Draw-a-secret (DAS) method is presented in paper [8], where the user is allowed to depict his password graphically. In this method, the user has to depict a simple shape during the registration phase in a two-dimensional space. This two-dimensional space is grid-shaped, and in the login phase and authentication, the user must remember the drawn image in the previous grid space to allow the login to be given in this technique, no help or suggestion from the server-side is given to the user to depict a password, and it is a Pure Recall Base based method.
In the paper [12], the author reviews and analyzes the method presented in paper [8]. The basis of further research has been examined the extent of remembering the graphical images drawn by users as well as to check the security of this method in implementing graphical passwords in this paper, the concept of graphical dictionaries is expressed, and

the possibility of applying brute-force attack using these dictionaries is examined. Their review showed that the use of DAS in the 5*5-grid increase the security coefficient against brute-force attacks. The current examinations in this paper also show that the security of using symmetric DASs is highly lower than those of non-symmetric DASs, but because of the high coefficient of memorizing of these schemes, using of symmetric DASs is much more than asymmetric DAS.

One of the methods presented in the paper [7] is a method named Passdoodle, which is requested to draw its password graphically on a sensitive screen, such as a smartphone screen. The accuracy of this method is high, but its cost is far higher and does not eliminate vulnerabilities such as guessing, spyware, and Shoulder surfing.

In the paper [15], a graphic password is designed where the user creates a password in the registration phase by clicking on the points of the image. The user must click on a range of boundaries specified in the image, and it is done in the login phase by clicking on the selected pixels of the login. Since the image is shown to the user at the login phase and is used to remember the graphical password, this method is placed in the category of Cue recall methods.

In the paper [14], a method known as the pass point is stated, which is an evolution of the previous method. In this method, there is no need to click on the specified ranges in the image, and the user can use any image to create a graphical password, and clicks can do anywhere on the image. In the login phase, with clicking on the image, the pixels which are clicked and the order of clicking will be compared with the step of creating the password, and the permission and authentication are performed. Creating a password in this method is simpler and more powerful than the previous method, but in terms of time, the log in the system will take longer than using text passwords.

In the paper [1] describes a method called Game Changer Password System (GCPS) in which games such as chess or Monopoly are used to create a password. In these systems, the number of games increases according to the expected level of security to increase the password selection interval and reduce system vulnerability. For example, in the game of chess, a chessboard is displayed to the user, and the user can choose between black and white (B, W) and beads (pawn, rook, knight, bishop, queen or king) and create a password.

In the paper [2], an authentication system called Association List Password is proposed to help users use complex passwords to increase system security as well as to help the user remember the password. The system will display to the user several columns of words numbered at the time of logging in, and the user will be asked to select their chosen words at the time of registration. To avoid shoulder surfing vulnerability the user can choose the number of words instead of the mouse using the keyboard.

In the paper [13], A new algorithm for using images in the authentication process is presented, It has been called Visaul Question Authentication Protocol (VQAP). This method uses a combination of image, question and answer to create a password.The steps are to show the user when registering, some images. Afer selecting the images, questions about the image are shown to the user. The user must select the question that fits the picture and then specify the answer appropriate to the image and the question.

When logging in, using machine learning algorithms images, questions and answers are categorized intelligently and displayed to the user. This can greatly reduce of guessing a combination password. It also does not resolve some security issues such as Brute force, Shoulder surfing and MITN, and is a bit difficult for users to understand easily.
In the paper [16], A new method called the Evolutionary Authentication System is being introduced and Its main purpose is to eliminate Shoulder surfing vulnerability. In this way the graphical password is uploaded by the user and extracted from that particular graphic design. When logged in, this drawing will be displayed along with other designs which can only be recognized by knowing the original image.
All of the methods and categories mentioned above to create a graphical password have been investigated based on security and effectiveness in these papers. Many techniques have been able to remove a large number of security vulnerabilities, but they have not a good rank in terms of efficiency and usefulness. In the discussion of the usefulness of the items are considered, such as the difficulty of creating user password by the user and the average time of creating the password and the time of the login process. Some of the proposed methods have a high-performance rating, but they are more vulnerabilities in terms of security. Vulnerabilities such as brute-force attacks using existing image dictionaries and as well as Man-in-middle attacks are of these vulnerabilities which they are not fully articulated in the proposed methods. A new method is proposed in this paper using the encryption methods between client and server as well as the usage of marking algorithms in digital images. This new method is introduced to implement a graphical password-based authentication system that can significantly resolve the two mentioned vulnerabilities. We will examine the extent of these vulnerabilities in the new method.

3 Digital Image Watermarking for Authentication

The proposed authentication system is based on the use of a graphical password for authentication and login system. In the first part, a method to utilize these images and design in the information registration and authentication phase is presented which can prevent of the vulnerability of Shoulder surfing, which is one of the most critical vulnerabilities in graphic authentication systems. In the second part, a protocol will be provided to secure the process of selecting images and authenticating the selected images, preventing vulnerabilities such as MITM or network-hacking and obtaining images using markup in digital images. In the third part, a method will be provided for marking images with considering a proposed protocol.

3.1 Registration and Authentication Process

The registration process and authenticating (login) are considered as follows:
Registration:

1. Choosing the desired username.
2. Display of 9 images in 3 rows and three columns.

3. Choose of 6 images as a password in three pairs of images (three binary image categories).

Login:

1. Enter username
2. Display of 9 random pictures in which there are six selective images at the registration process
3. Choosing the images that intersect along its length and width end in of the selected image pair at the registration phase.

In this method of using images since the user does not directly select images that are related to his password in the system, the hacker cannot use the Shoulder surfing vulnerability, because the probable range of the images become very wide-spread in which the next section will investigate the numerical evaluation and current probabilities presented in this method.

3.2 Watermarking Image

The proposed system is based on watermarking images based on a specific server-side algorithm and dewatermarking the image created on the client-side. The watermarked data in the images are random characters on the server-side, which are created once every time of the authentication process for each image and is stored on the server-side. The basis of the work and the whole of the process involves the following steps:

1. Create a string of random characters for each image on the server-side.
2. Store string created on the server-side.
3. Do watermark the images with the proposed algorithm by the data created for each image.
4. Send the watermarked image by the server to the client.
5. Dewatermarking the image on the client-side with the same algorithm.
6. Send data obtained from Dewatermarking images from client-side to the server.
7. Make a comparison of the sent data from the client with the original data on the server-side.

In the proposed method and protocol, two security vulnerabilities can be considered. One of them is disclosing of the dewatermark algorithm due to its implementation on the client-side, which can address the total security of the system. The other is audibility and obtain the string after the dewatermark from the client-side to the server for review data transferring by the penetrator on the network or MITM attack on the network and obtain the desired string that can generally bypass the proposed authentication system. In the proposed method, data transfer is based on the RSA protocol, causing a lack of

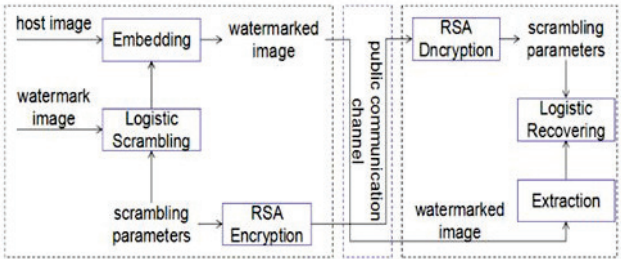


Figure 1: The proposed protocol

audible and extraction of a string derived from dewatermarking of images, and virtually MITM vulnerability will be resolved. In the process, the watermarking algorithm based on a parameter is done as a key, that our client, based on this key, using the watermark on server-side images, can Dewatermark selected images and obtain the desired sting. As shown in figure1, this key will be transported using the RSA protocol between the client and the server. Practically even with knowing the Dewatermark algorithm on the client-side, one cannot use watermark images without having a unique key on the server-side and using those watermarked images to extract the string needed for authentication from the images.

3.3 Watermark Algorithm Using a Unique Key

The purpose of our proposed algorithm is watermarking a string of information using a key in the desired images. According to the fact that a string of characters has a little bit of information, the proposed algorithm is based on the LSB algorithm. The algorithms using rely on the frequency domain to watermarking the images, such as DCT are more efficient for watermarking the image in the other image. However, using frequency-based algorithms for watermarking the characters in the images will increase the processing elements and reduce the performance of the algorithm.

The LSB algorithm stands for the Least Significant Bit, and it is one of the easiest watermarking methods based on changing the least significant bit in every eight bits of the image with the bit of the data set to be the watermark. In digital images, the data can be placed directly in each bit of the image information, and the bits are better placed in the crowded areas of the image so that hidden data be less considered. The basic principle of this method is changing the number of bits of each pixel of the image [10].

In this method, adding the required data bits to be hidden is very simple and effective. If we want to do hidden in a black and white bitmap image, it is enough to change its priceless bit to our desired bit for every 8 bits per pixel. In black and white images, each pixel consists of one byte of data or 8 bits, which can create 256 black and white colours ranging from 0 to 255 which can become hidden in each of these data bytes or a bit of the desired data. Figure 2 shows this method.

Now, if we want to apply this algorithm on color images, according to that each pixel of RGB color images has 3 bytes or 24 bits of data, it can be changed in 3-bit low-value

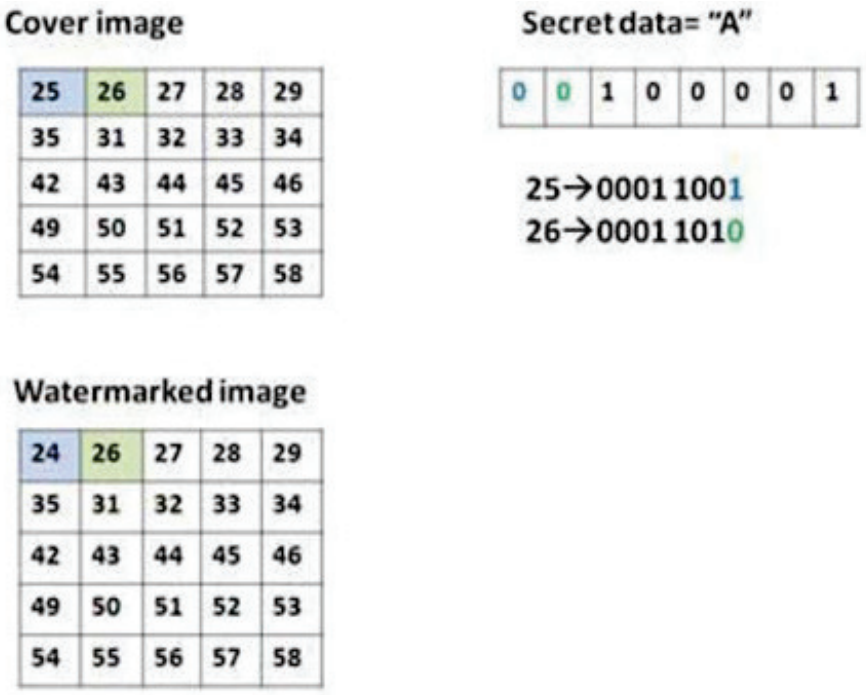
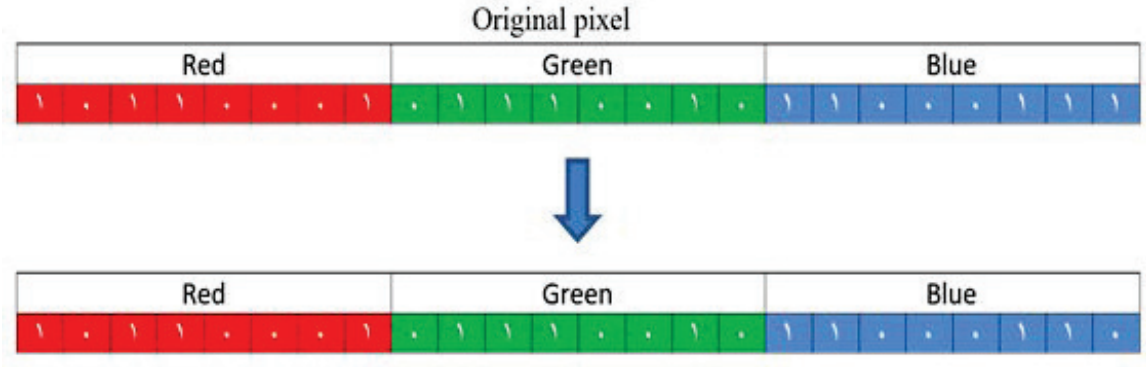


Figure 2: LSB algorithm

pixels to fit the desired character ASCII code. For example, the ASCII code of letter A becomes 65, and the first three bits of this number can be placed in a pixel of the image as follow:

ASCII code of A= 65, binary: 01000001



As you can see, the first three bits of capital A watermark in a single-pixel with changing only one bit of 24 bits per pixel. Typical characters for creating a security string in order to watermark the images are shown in figure 3 [10].
There are two phases in the proposed algorithm for watermarking using the LSB algorithm, before the bits of ASCII code mapping on bits of each pixel of the image. The first

Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char
32	space	52	4	72	H	92	\	112	p
33	!	53	5	73	I	93	]	113	q
34	"	54	6	74	J	94	^	114	r
35	#	55	7	75	K	95	_	115	s
36	$	56	8	76	L	96	`	116	t
37	%	57	9	77	M	97	a	117	u
38	&	58	:	78	N	98	b	118	v
39	'	59	;	79	O	99	c	119	w
40	(	60	<	80	P	100	d	120	x
41	)	61	=	81	Q	101	e	121	y
42	*	62	>	82	R	102	f	122	z
43	+	63	?	83	S	103	g	123	{
44	,	64	@	84	T	104	h	124	|
45	-	65	A	85	U	105	i	125	}
46	.	66	B	86	V	106	j	126	~
47	/	67	C	87	W	107	k		
48	0	68	D	88	X	108	l		
49	1	69	E	89	Y	109	m		
50	2	70	F	90	Z	110	n		
51	3	71	G	91	[	111	o		

Figure 3: The typical characters with ASCII code

phase is data compression, and the next phase is coding the information according to the secret key that will be transferred between the client and the server using RSA on the network. The purpose of this operation and adding these two phases to the algorithm is the following:

1. Reduce the amount of watermarked information in the image.
2. Make the least change in the bits of the original image and create the most similarity between the original image and the watermarked image.
3. The need for knowing the compression algorithm and the sent secret key, besides the need for knowing the watermarking algorithm will increase the security of the protocol and the authentication system on the client-side.

In the first phase, which is the compression of the characters that we intended to use for watermarking, we consider the ASCII code of the first character then we will consider the numerical differences of the ASCII codes in order to the next characters, for example:
Secret string: ga\Upp

According to figure 3, the bits of this field are as follows:

Char	g	a	\	U	p	p
ASCII	103	97	92	85	80	80

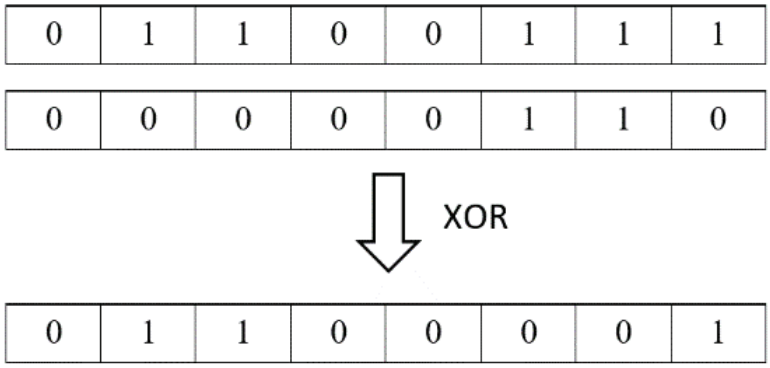
According to the definition for each character, we will calculate the difference between its ASCII code with the previous character, and thus, the values will be changed as follows:

Char	g	a	\	U	p	p
ASCII	103	6	5	7	5	0

Considering that the differences may be negative, one bit is considered for the sign of this difference number, which 0 means positive, and one means negative.
The second phase in the proposed algorithm is using a secret key for watermarking. According to the above description, in order to dewatermarking of the image and obtain the string, the length of the string should be specified by the dewatermark algorithm, so that it can dewatermark of image pixels according to the length of the string. In addition to required data for dewatermarking, the length of the string is used as a bit string to encrypt the data by the XOR operator. For this purpose, all the bit strings that are going to be watermarked in the pixels of the image using the LSB, at first become XOR with the length of the string specified on the server-side and then they will be placed in the image. The length of the string, which is required for both decrypting and dewatermarking image on the client-side, will be encoded and transmitted from the server-side to the client using RSA. Therefore, in order to penetrate to the proposed authentication system, the penetrator, in addition to detecting the watermarking algorithm, should be aware of

the type of compression data and the length of the string which is sent as the secret key in the RSA. In the mentioned example, the length of our string is 6. The compression and encryption of the desired string will be as follows:

Watermark string: ga\Upp
Secret key = Length of string = 6
g = 103
secret key = 6



In the above example, all numbers are positive. The difference between the ASCII codes of the selected characters must be a maximum of 7, which ultimately can be stored in 4 bits of data. Since the difference is calculated based on the preceding letter, the range of selected characters is constraints from 7 to - 7, will still be suitable. Flowchart figure 4, showing the proposed authentication system and how to use the watermarking algorithm in this system. In the next section, we will state the effectiveness of the proposed algorithms and the proposed method for implementing the authentication system.

4 Analysis and Evaluation

The analysis and evaluation of the proposed protocol and the proposed algorithm for hiding the data in images can be examined in several phases as follows:

1. The number of selected images in the selection method of the paired images to avoid the vulnerability of shoulder surfing.
2. The number of changes in the original image compared to the watermarked image in the proposed algorithm.
3. The number of selected characters, and their relationship with the amount of image variation and the length of the secret key, which is needed to decode the watermarked images.
4. Evaluate the possible vulnerabilities of the proposed protocol and protocol strength to resist these vulnerabilities.

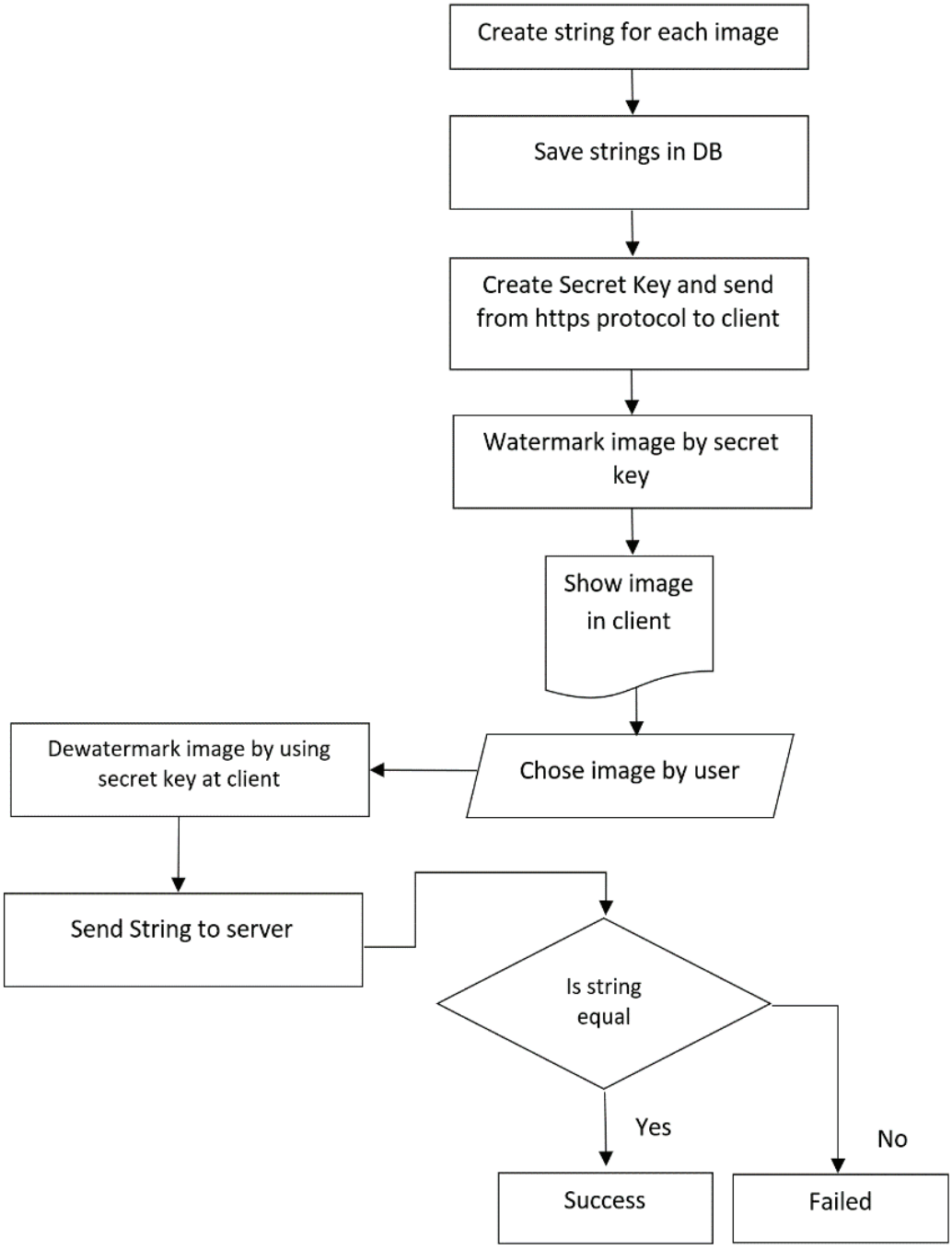


Figure 4: Flowchart of the proposed authentication

Each of these phases will be analysed and investigated.

4.1 Analysing the Selective Range of the Image Pair

Selecting the pair of images at the time of registration and selecting the intersection of this pair of images at the time of login solves the shoulder surfing vulnerability. The number of built-in passwords in a 5- in-5 ranges of images and the choice of two images will be calculated to create a password with three clicks at login time as follows:

$$33 \times (25c2) = \frac{3 \times 25!}{2! \times 23!} = 900password \tag{1}$$

This number of passwords is calculated regardless of the random position of images in the 5-in-5 grid. With considering the random representation of these images in 25 positions of this grid, the number of passwords can be calculated as follows:

$$P = 900 \times 25! = 1.39 \times 10^{28} > 10^{28} \tag{2}$$

4.2 The Number of Image Changes and the Range of Selected Watermarked Characters

According to the proposed algorithm, the first character can be used to create a character string for watermarking in the image is between the 32-126 ASCII code. The number of ASCII code can store the first character in at maximum 8 bits. So, if the base is watermarking in an RGB image, we will need 3 pixels to store the first character, which will be used in the last pixel of the last 8 bits of R, G.
First character = 3 pixel
Regarding the algorithm for subsequent characters, instead of storing the entire ASCII code, its difference with the previous ASCII code will be watermarked. If we consider the maximum difference between two characters in the creation of a string, X, and the length of the entire string L, the number of bits required to store each character and the number of pixels required to store the entire string in the image will be as follows:
x $\rightarrow$ binary $\rightarrow$ n bit $\rightarrow$ n+1 bit
The number of required bits to store the number X is stored in n. One bit is considered to store the sign of the desired number. Finally, we will need (n + 1) bits to store ASCII codes of the second character. The number of required pixels to store the secondary characters in an RGB image will be as follows:

$$M(pixel) = \frac{L \times (n+1)}{3} \tag{3}$$

Also, to store the entire field of this value-add to four pixels to store the first character:

$$Total(pixel) = M(pixel) + 4 \tag{4}$$

Regarding constraint established for choosing the second character and so on, if we consider L as the chosen range and n as the number of characters, the chosen range of these characters will be as follows:

$$P = 95 \times (12)^n \tag{5}$$

Indeed, the creation of a vast range for selecting images and character strings to create a secret key will increase the security coefficient of the proposed method against brute force attacks.

4.3 Evaluation of Protocol Security Vulnerabilities

In a graphical authentication system, there may be security vulnerabilities that in this section, we state these vulnerabilities and the resistance of the proposed system against such attacks:

- Shoulder surfing: The common vulnerability that exists in systems with graphical passwords. This vulnerability is considered in the part of image selection and was greatly resolved.

- MITM (men in the middle): If there is a penetrator between the server and the client. This vulnerability will be examined in such a way that the penetrator receives images from the server and introduces itself as a server for the client, and after receiving the correct image from the client will give it to the server and then log in itself instead of the user in the system. In the proposed protocol, the client, in addition to knowing the correct pair image, must be aware of the watermark algorithm and can extract the hidden string in the correct images from the images and transfer it to the server. However, the correct images by the penetrator will not be virtually sufficient conditions to log into the system instead of the main user.

- The existence of a watermarking algorithm on the client-side: regarding the dewatermark algorithm will be carried out on the client-side; the vulnerability will be considered if the dewatermark algorithm is extracted, the effectiveness of the algorithm in the face of MITM attacks will be eliminated. According to the fact that the proposed algorithm for dewatermarking images requires a secret key corresponding to those watermark images and the server sends this secret key in each request to the user, the dewatermark algorithm actually will not be sufficient condition for extracting the desired string on the client-side. The client will need a secret key created by the server for dewatermarking. This secret key will be represented the length of the range for choosing the ASCII code used in watermarking, and without it, the dewatermark algorithm cannot extract the hidden string from the images.

The need for security in sending secret key and string obtain of dewatermarking of sent images: according to all the security mechanisms described in order to deal with security attacks, there is an essential point that the whole system will be unsafe if the network has been intercepted, and require the secret key or the watermark string. Using of RSA

protocol for Secret key encryption and watermarked string based on the public key and private key between the server and the client ensures that compliance is not possible, even if penetrator obtains compliance parameters, it is impossible decoding and using it in the system.

5 Conclusions

One of the critical concepts that can be used to design an authentication system based on the using images instead of text is the use of data encryption concepts and watermark and steganography algorithms in the design of these systems. In line with these concepts, design for implementation of graphical passwords proposed by integration and combining watermarking in the detection and authentication of users in the system try to increase the security coefficient and reduce the current vulnerability of users and can overcome the requirements of a graphical authentication system significantly. Another aspect of the design of this system is provided watermarking algorithms and the use of data hiding to verify the sent images between the client and the server that has managed to remove some vulnerabilities due to the absence of an SSL secure connection between the client and the server. As well as using watermarking to involve some of the unique features of the user in the images and creation of unique graphical modules, it can significantly increase the security coefficient of the system, and this area of authentication systems can be significantly developed.

References

[1] Bosnjak, L., Brumen, B., 2019. Shoulder surfing: From an experimental study to a comparative framework. International Journal of Human-Computer Studies. 130, 1–20.

[2] Brumen, B., 2019. Security analysis of game changer password system. International Journal of Human-Computer Studies. 126, 44–52.

[3] Corporation, P., . The science behind passfaces. white paper. `http://www.passfaces.com/enterprise/resources/whitepapers.htm`.

[4] Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T., 2008. Digital watermarking and steganography.

[5] Davis, D., Monrose, F., Reiter, M., 2004. On user choice in graphical password schemes, in: In proceedings of the 13th USENIX Security Symposium.

[6] Dhamija, R., Perrig., A., 2000. Deja vu-a user study using images for authentication, in: In proceedings of the 9th USENIX Security Symposium.

[7] Goldberg, J., Hagman, J., Sazawal, V., 2002. Doodling our way to better authentication, in: CHI '02 Extended Abstracts on Human Factors in Computing Systems.

[8] Jermyn, I., Mayer, A., Monrose, F., Reiter, M., Rubin, A., 1999. The design and analysis of graphical passwords, in: In proceedings of the 8th USENIX Security Symposium.

[9] Man, S., Hong, D., Mathews, M., 2003. A shoulder-surfing resistant graphical password scheme, in: In proceedings of international conference on security and management.

[10] Parashar, P., Singh, R., 2014. A survey: digital image watermarking techniques. International Journal of Signal Processing, Image Processing and Pattern Recognition. 7, 111–124.

[11] Sobrado, L., Birget, J., 2003. Graphical passwords. The Rutgers Scholar, An Electronic Bulletin for Undergraduate Research 5.

[12] Thorpe, J., Oorschot, P., 2004. Graphical dictionaries and the memorable space of graphical passwords, in: In roceedings of the 13th USENIX Security Symposium.

[13] Toor, A., Wechsler, H., Nappi, M., Raymond, K., 2019. Visual question authentication protocol (vqap). Computers & Security 76, 285–294.

[14] Wiedenbeck, S., Waters, J., Birget, J., Brodskiy, A., Memon, N., 2005a. Authentication using graphical passwords: Basic results. Human-Computer Interaction International .

[15] Wiedenbeck, S., Waters, J., Birget, J., Brodskiy, A., Memon, N., 2005b. Authentication using graphical passwords: Effects of tolerance and image choice, in: In ymposium on Usable Privacy and Security (SOUPS).

[16] Yu, X., Wang, Z., 2017. Evopass: Evolvable graphical password against shoulder-surfing attacks. Computers & Security 70, 179–198.