IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 26, 2020, accepted August 16, 2020, date of publication August 21, 2020, date of current version September 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3018659

A Password-Based Authentication System Based on the CAPTCHA AI Problem

MASOUD ALAJMI[1], (Member, IEEE), IBRAHIM ELASHRY[2], HALA S. EL-SAYED[3], AND OSAMA S. FARAGALLAH[4,5]

[1]Department of Computer Engineering, Taif University, Taif 21974, Saudi Arabia
[2]Department of Electrical Engineering, Kafrelsheikh University, Kafr El-Shaikh 33511, Egypt
[3]Department of Electrical Engineering, Faculty of Engineering, Menoufia University, Shebin El-kom 32511, Egypt
[4]Department of Information Technology, College of Computers and Information Technology, Taif University, Taif 21974, Saudi Arabia
[5]Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Menouf 32952, Egypt

Corresponding author: Masoud Alajmi (ms.alajmi@tu.edu.sa)

ABSTRACT Powerful cryptographic systems based on mathematically hard problems are utilized to ensure tighter security for data communication purposes. However, these traditional cryptographic systems are bound to fail in the ensuing era of quantum computing. Thus, Artificial Intelligence (AI) inspired security methods are needed to secure communications in the era of quantum computing. This article presents a challenge-response password-based authentication system based on the Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) AI hard problem. In this system, a server sends a challenge text to a client, then the client generates a random image and blends the challenge text inside this random image using his password. Then the client sends the generated image to the server. The server extracts the challenge text from the sent image using his copy of the client's password. If the extracted challenge text is the same as the sent challenge text, then both the client's and the server's copies of the password match and the client is authenticated. The efficiency of the proposed system is analyzed and the outcomes prove that the proposed system is efficient in terms of time and space. Also, a security investigation of the proposed system is employed, and the results prove that the system is probabilistic and very sensitive to changes in its parameters. It does not leak any statistical information about the client's password and the generated images cannot be distinguished from random images. In addition, the security of the proposed system is analyzed against two possible attacks; the brute force attack and the replay attack and the results prove that the proposed system is immune to these attacks. Finally, the proposed system is ensured to be indistinguishably secure against an adaptive chosen-challenge text attack (IND-ACCTA), based on the CAPTCHA AI hard problem when the hash function H is modeled as a random oracle.

INDEX TERMS Password-based authentication, CAPTCHA, AI hard problems.

I. INTRODUCTION

The World is witnessing a new era of computing revolution. This revolution comes in the form of quantum computing. Although this revolution will renovate the way the people live, it comes with enormous information security challenges.

The security of current cryptographic systems is based on mathematical hard problems such as the integer factorization problem and the discrete log problem [1], [2]. These problems are very hard and impractical to solve with current traditional computing. But as the development of quantum computing grows each year, these problems will no longer be hard to solve. This is a major security concern as the current security systems will not be secure anymore.

The best way to overcome these security challenges is to move from depending on mathematical hard problems to AI hard problems. An AI hard problem is a problem that needs a computer to be "humanly" smart. This will be a challenge even for quantum computing [3].

An application of AI hard problems in cryptography is CAPTCHA [3]. A CAPTCHA is a program that distinguishes a human from a bot by creating a challenge that is easy for humans but hard for bots. For instance, humans can recognize a twisted text, but bots cannot. Examples of CAPTCHA challenges are shown in Figure 1.

The associate editor coordinating the review of this manuscript and approving it for publication was Yassine Maleh.

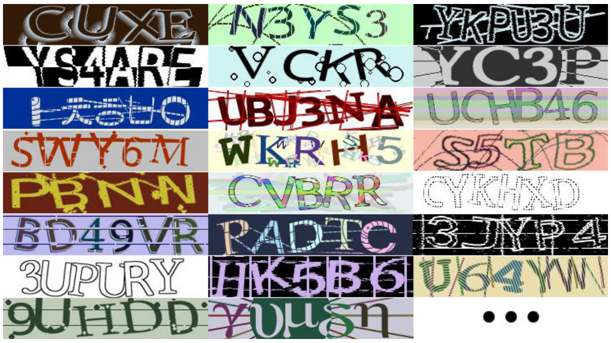


FIGURE 1. Examples of CAPTCHA challenges.

CAPTCHA is used to prevent bots from accessing websites with personal and valuable information such as emails and bank accounts.

Von Ahn *et al.* [4] were the first to use AI hard problems in information security. They proved that "any program that passes the tests generated by a CAPTCHA can be used to solve any AI hard problem" [3]–[5].

Despite the increasing number of methods to authenticate clients, password-based authentication is considered the most popular method of all [6]. Password-based authentication is a way for a client to securely access services such as emails hosted by a service provider. To prevent unauthorized persons from accessing his services, the client must provide a username and password to the service provider. The client is granted access to his requested service if the username and password pair provided by the client matches the username and password pair in the service provider's database. The major advantage in password-based authentication is that passwords can be easily used and memorized [6].

One of the most used password-based authentication systems is the Challenge Response Authentication Mechanism (CRAM) [6]. In these systems, the client requests to access a service (Ex. Email) from a service provider (Ex. Google). The service provider then challenges the client by sending a challenge to him. If the client answers the challenge correctly, the service provider grants him access to his requested service. CAPTCHA is considered a challenge that it used by CRAM to differentiate humans from bots [6]. An Example of CRAM is CRAM-MD5 [7].

A problem with these systems is that the same password is used repeatedly, and an adversary can intercept the sent password even if the password is hashed and resend it to the service provider. In this case, the service provider cannot determine if the client is legit or not. A solution to this problem is Salted Challenge Response Authentication Mechanism (SCRAM). In SCRAM, a unique salt is generated and hashed with the password to make the hash unique every time a client requests a service from the service provider [6]. An Example of CRAM is SCRAM-SHA-1 [8].

The main aim of this article is to propose a salted challenge-response password-based authentication system based on the CAPTCHA AI hard problem. The idea behind the proposed system is the same as CAPTCHA. That is, it is a hard problem for a bot to recognize a twisted text in an image. Instead of sending the challenge text in a way that is easy for humans but prohibitively difficult for bots as in CAPTCHA, the proposed system blends the challenge text and scatters it inside a random image using the client's password. This process makes the challenge text hard to spot for humans and bots. The server then receives the generated image from the client and uses his copy of the client's password to recover the challenge text. The challenge text can be recovered correctly if the client's and the server's copies of the password are the same.

The paper is structured as follows. Section II reviews the related work to the proposed system. Section III summarizes the mechanism of traditional challenge-response password-based authentication. Section IV explains in detail the proposed system. Section V investigates and evaluates the performance of the proposed system. Section VI extensively investigates the main security features of the proposed system. Section VII tests the proposed system against the brute force attack and the replay attack. A security proof for the proposed system is presented in section VIII. The paper is summarized and concluded in section IX. Finally, the future work is suggested in section X.

II. RELATED WORK

Ning *et al.* presented a hierarchical challenge-response authentication system with aggregated-proof for the Internet of Things (IoT) [9]. Two sub-protocols are made for unit and ubiquitous IoT for security protection. Their system provides confidentiality and data integrity using homomorphism-based Chebyshev chaotic maps and the directed path descriptor [9].

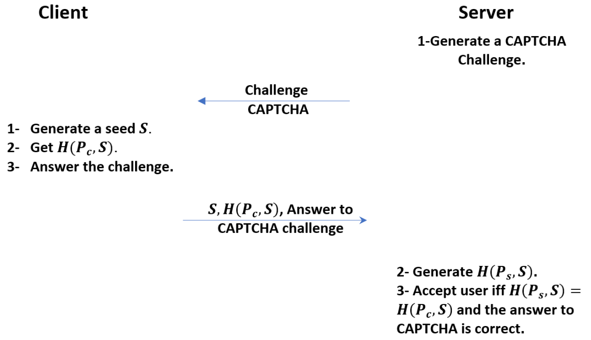
Alharbi *et al.* [10] presented a Fog Computing-based Security (FOCUS) system that is used to secure IoT. This system uses Virtual Private Networks (VPN) to connect to IoT devices, then it uses a challenge-response authentication mechanism to defend the VPN networks from distributed denial of service (DDoS) attacks. Their system is applied in the end user side to increase its speed and efficiency. The system has a low latency response without sacrificing its security.

Sluganovic *et al.* [11] presented a challenge-response authentication system based on the tracking of eye movements. The system takes advantage of the fact that eye movements are fast and contain unique biometric information for each person. The system records the images of person's eye movements and compares them to the ones associated with that person stored in the receiver's database. This system is immune against reply attacks. The system is tested practically with 30 persons. The system has achieved low latency of 5 seconds and low error rate of around 6%.

Prabhu and Shah [12] presented an authentication system combining graphics with textual password. The client enters a complex password, then it randomly generates a grid of symbols and characters and a password is selected from the grid. Although this system is improving security by combining graphics with textual password, the system can easily be compromised by taking a screenshot [12].

III. TRADITIONAL CHALLENGE-RESPONSE PASSWORD-BASED AUTHENTICATION

This section briefly explains the mechanism of traditional challenge-response password-based authentication. Assume that there is a client and an email server, and the client wants to access his email. The communication between the client and the server is illustrated in Figure 2. The symbols are defined in Table 1.



1-Generate a CAPTCHA Challenge.

1- Generate a seed S.
2- Get $H(P_c, S)$.
3- Answer the challenge.

2- Generate $H(P_s, S)$.
3- Accept user iff $H(P_s, S) = H(P_c, S)$ and the answer to CAPTCHA is correct.

FIGURE 2. Traditional challenge-response password-based authentication.

TABLE 1. Notations.

Symbols	Meaning
c	subscript for client
s	subscript for server
$CText$	Challenge Text
H	Hash Function
P	Password
S	Seed

The server generates a challenge (Ex. CAPTCHA) and sends it to the client. The client then randomly generates a seed (salt) S and the hash $H(P_c, S)$ where P_c is the client's password. Using the seed along with the password to generate the hash ensures that the hash will be different each time a different seed is used even if the password is the same. This process is essential to resist dictionary attacks [13]. The client answers the challenge and sends the answer along with S, $H(P_c, S)$ to the server.

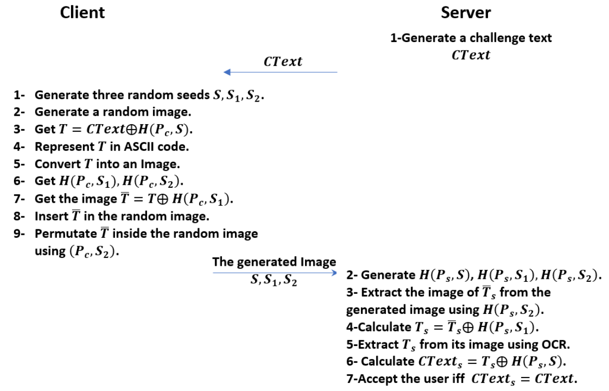
After receiving S, $H(P_c, S)$ and the answer to the challenge from the client, the server checks the answer and calculates $H(P_s, S)$ where P_s is the server's copy of the client's password. If the answer is correct and $H(P_s, S) = H(P_c, S)$, then $P_s = P_c$ and the client is authenticated.

The hardness of the traditional challenge-response password-based authentication systems is based on hash functions. Although it is very difficult to get the reverse of a hash function, it is not impossible, especially if quantum computing is taken into consideration.

Another problem is that hash functions are deterministic; the output of a hash function is the same if the input is the same [13]. This makes them vulnerable to brute force attack if the computing power requirements are met.

IV. THE PROPOSED PASSWORD-BASED AUTHENTICATION SYSTEM

The proposed system is explained in Figure 3.



1-Generate a challenge text $CText$

1- Generate three random seeds S, S_1, S_2.
2- Generate a random image.
3- Get $T = CText \oplus H(P_c, S)$.
4- Represent T in ASCII code.
5- Convert T into an Image.
6- Get $H(P_c, S_1), H(P_c, S_2)$.
7- Get the image $\overline{T} = T \oplus H(P_c, S_1)$.
8- Insert $\overline{T}$ in the random image.
9- Permutate $\overline{T}$ inside the random image using (P_c, S_2).

2- Generate $H(P_s, S), H(P_s, S_1), H(P_s, S_2)$.
3- Extract the image of $\overline{T}_s$ from the generated image using $H(P_s, S_2)$.
4-Calculate $T_s = \overline{T}_s \oplus H(P_s, S_1)$.
5-Extract T_s from its image using OCR.
6- Calculate $CText_s = T_s \oplus H(P_s, S)$.
7-Accept the user iff $CText_s = CText$.

FIGURE 3. The proposed password-based authentication.

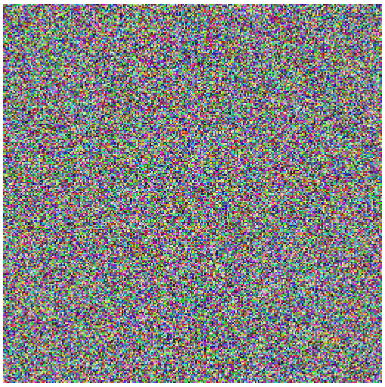
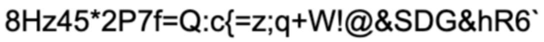


FIGURE 4. A random image.

The server sends a challenge text $CText$ to the client as a plaintext. The client then generates three random seeds S, S_1, S_2 and a random image. An example of a random image is shown in Figure 4. Next, the client calculates $T = CText \oplus H(P_c, S)$ where P_c is the password of the client and converts it into ASCII (American Standard Code for Information Interchange) characters by representing each 8 bits of T in ASCII using the standard ASCII table. Then the client converts T into an image by representing each character as a pixel as shown in Figure 5. The XORing of the challenge text with $H(P_c, S)$ conceals the challenge text and makes it random. For clarification, let $CText = HELLO$, $S = 543$ and $P_c = 4443354h$ and using SHA-3 256-bit the client calculates $T = CText \oplus H(P_c, S) = HELLO \oplus H(4443354h, 543) = 8Hz45 * 2P7f = Q : c\{= z; q + W!@\&SDG\&hR6$‘. The image T is shown in Figure 5.

The client then calculates $H(P_c, S_1)$ and $H(P_c, S_2)$ and calculates $\bar{T} = T \oplus H(P_c, S_1)$ by XORing $H(P_c, S_1)$ with the pixels of the T image. This step conceals the T image and makes it indistinguishable from a random image. The process is shown in Figure 6.



8Hz45*2P7f=Q:c{=z;q+W!@&SDG&hR6`

FIGURE 5. An image representation of T.

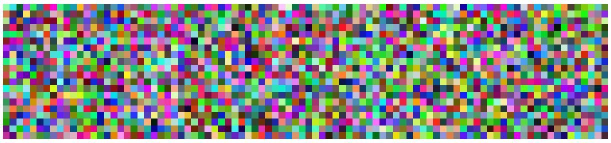


FIGURE 6. Generating the image $\bar{T}$ by concealing T using the password P.

Next, the client inserts $\bar{T}$ into the random image shown in Figure 4 such that $\bar{T}$ is scattered in the whole image using $H(P_c, S_2)$. This can be achieved using many techniques. For instance, $H(P_c, S_2)$ can be used as a seed for a random number generator (RNG) and the RNG output determines the indices of the $\bar{T}$ bits placed in the least significant bit (LSB) of the image pixels. Another way is to use permutation maps such as the baker map [14] to scramble $\bar{T}$ in the image and use $H(P_c, S_2)$ as the key of the map. The client finally sends the generated image along with S, S_1 and S_2. The generated image sent to the server is shown in Figure 7.

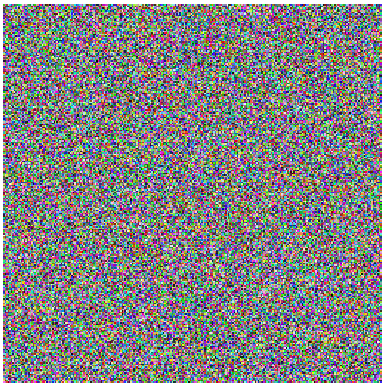


FIGURE 7. The generated image.

After receiving the image, the server uses $H(P_s, S_2)$ where P_s is the client's passwords stored in the server's database to get the image of $\bar{T_s}$ by undoing the scrambling made by the client. After that, the server gets the image of T_s by XORing $H(P_s, S_1)$ with the pixels of $\bar{T_s}$ (i.e. $T_s = \bar{T_s} \oplus H(P_s, S_1)$). Then the server extracts the text of T_s from its image using Optical Character Recognition (OCR). The server then calculates $CText_s = T_s \oplus H(P_s, S))$. If $CText_s = CText$ then $P_c = P_s$ and the client is authenticated.

A comparison between the proposed system and the traditional systems are shown in Table 2.

TABLE 2. A comparison between the proposed system and the traditional systems.

Property	The Proposed system	The Traditional systems
Hard Problem	AI	Mathematical
Security Primitives	CAPTCHA	Hash functions
CAPTCHA Definition	Hard for humans, Hard for bots	Easy for humans, Hard for bots
Purpose of CAPTCHA	Securing the challenge text	Differentiate between humans and bots
Probabilistic/ Deterministic	Probabilistic	Deterministic
Sent hash function	Blended in a random image	plaintext
Type of sent data	Images Numerical variables	Numerical variables

Although it may appear that the proposed system is not efficient compared to traditional systems because it sends images larger than the numerical variables, this is actually not the case because most password-based authentication systems already send CAPTCHA images to make sure that a human not a bot is being authenticated. Sending a CAPTCHA image specifically to differentiate between humans and bots is not required in the proposed system. In addition, the size of a 256 × 256 RGB image is 193KB. This is nothing compared to nowadays communication speed or information storage capacity.

At first glance, it may appear that the proposed system is a steganographic system since the challenge text is placed inside an image. However, this is not true for the following reasons.

- The security of steganographic systems is based on inserting a secret payload inside an innocent-looking file (container) so that the payload is sent without being detected [15]. It requires the alterations caused by the payload to be negligible. However, the security of the proposed system does not depend on sending undetected secret information; the adversary in the proposed system knows that the generated image contains the challenge text.
- The payload embedding in steganographic systems does not require any secret information such as keys or passwords. On the other hand, the proposed system embeds the challenge text inside the random image based on the client's password.
- The payload is secret in steganographic systems, while the challenge text in the proposed system is public.
- The goal of steganographic systems is securely sending the payload without being detected, while the proposed system uses the challenge text to test the client's authenticity.

V. PERFORMANCE ANALYSIS

This section analyzes the performance of the proposed system by inspecting the computation time and the communication complexity of the proposed system and comparing these results with the traditional system presented in Section III.

In this analysis, random colored red, green, and blue (RGB) images with a size of 256×256 pixels are used. The size of the T and $\bar{T}$ images are 20 × 90 pixels. The hash function used in this analysis is SHA-3 256-bit [16] and the baker map is used for the scrambling process [14]. The parameters used in these tests are shown in Table 3.

It is stressed here that the results of this analysis do not rely on the type of the hash function nor the type of scrambling algorithm. Any strong one-way hash function and any secure

TABLE 3. Parameters values.

Parameter	Value
$CText$	HELLO
P	4443354h
S	453
S_1	451472693
S_2	483678982

scrambling algorithm will give the same results. In addition, the values of *CText*, P, S, S_1 and S_2 can be chosen arbitrary and the results will still be the same.

A. COMPUTATION TIME

This section measures the time required for the proposed system to 1) generate the image in the client side and 2) verify the password at the server side. These results are obtained using a PC running windows 10 and having a 10th generation intel® core i7-1065g7 processor with 16 Gigabytes of RAM. The results are shown in Table 4.

TABLE 4. Computation time results.

System	Time (client)	Time (server)	Total Time
$Proposed$	0.033 Seconds	0.017 Seconds	0.05 Seconds
$Traditional$	0.015 Seconds	0.008 Seconds	0.023 Seconds

From these results, it is concluded that the proposed system is time-efficient and the difference between the proposed system and the traditional system with respect to computation time is negligible.

B. COMMUNICATION COMPLEXITY

This section evaluates the communication complexity of the proposed system by examining the size of the data sent from the client to the server and vice versa and compares these results with the traditional system. The results are shown in Table 5.

TABLE 5. Communication complexity results.

System	Data sent to client	Data sent to server
$Proposed$	$CText$	The generated image, S, S_1, S_2
$Traditional$	CAPTCHA Image	The Answer to CAPTCHA, $S, H(P, S)$

From these results, it is concluded that the data traffic between the server and the client in both systems is dominated by images; the generated image in the proposed system and the CAPTCHA image in the traditional system and the difference between the two systems in terms of communication complexity is negligible. In addition, the image size in these systems is negligible. For example, a colored red, green, and blue (RGB) image with a size of 256 × 256 pixels is 193 KB represented in Portable Network Graphics (PNG) format. This means that the proposed system is efficient with respect to the speed of today's communication networks and the storage spaces of today's storage media.

To conclude, the proposed system is efficient in terms of computation time and communication complexity.

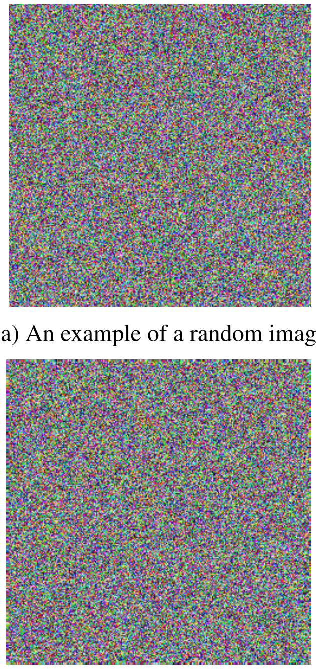
VI. SECURITY FEATURES

This section examines and verifies the main security features of the proposed system by inspecting 1) the indistinguishability of the generated images from random images using visual inspection, histogram analysis and entropy. 2) the probabilistic property of the proposed system and 3) the sensitivity tests (the diffusion property) for changes in *CText*, P, S, S_1 and S_2. The same parameters used in the performance analysis are utilized in these tests.

A. THE INDISTINGUISHABILITY TESTS

1) VISUAL INSPECTION

A comparison between a generated image sent from a client to a server and a random image is shown in Figure 8.



(a) An example of a random image

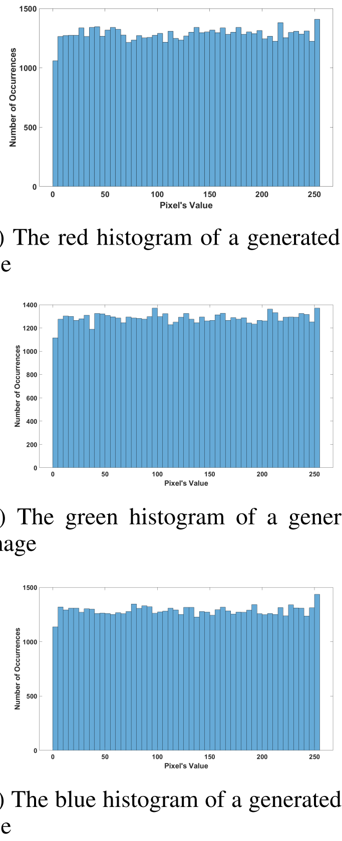
(b) An example of a generated image

FIGURE 8. A Comparison between a random image and an image generated by the proposed system.

Based on Figure 8, the image generated by the proposed system is indistinguishable from any random image.

2) THE HISTOGRAM ANALYSIS

A histogram is a measure of the distribution of pixels' values in an image [17]. It is shown as a graph with the values of the pixels versus the number of times these values repeated in the image. A good random image should have a uniform distribution. The more uniform the histogram, the better the randomness of the generated image. Since these images are colored RGB images, the histograms of the red, green, and blue sub-images of a generated image and a



(a) The red histogram of a generated image

(b) The green histogram of a generated image

(c) The blue histogram of a generated image

FIGURE 9. **The red, green, and blue histograms of an image generated by the proposed system.**

random image are compared and these histograms are shown in Figures 9 and 10.

Based on Figures 9 and 10, the histograms are uniform implying that the generated image is random.

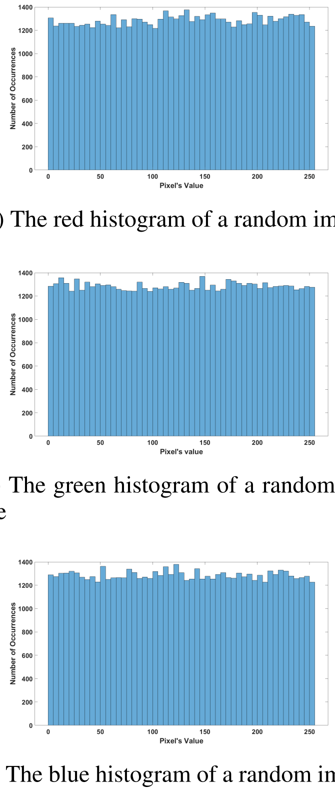
3) THE ENTROPY

The entropy measures how much information is in an image. It is function of the probabilities of the existence of the image pixels' values. The closer the probabilities of the existence of these pixels' values to each other, the more the entropy and hence, the better the randomness of the generated images [18]. The information entropy can be calculated as [18]:

$$I(x) = \sum_{i=0}^{2^N-1} P(x_i) \log_2 \frac{1}{P(x_i)} \tag{1}$$

where I is the entropy, $P(x_i)$ is the probability of the existence of pixel x_i and N is the pixel size in bits for each red, green and blue subimages. The maximum value of I is 8 [18]. I is calculated for each red, green and blue sub images for a generated image and a random image. The results are illustrated in Table 6.

Based on Table 6, it is deduced that the entropy is similar for both images and it is close to the maximum value.



(a) The red histogram of a random image

(b) The green histogram of a random image

(c) The blue histogram of a random image

FIGURE 10. **The red, green and blue histograms of a random image.**

TABLE 6. **Entropy of the generated and random images.**

Image	blue	Green	Blue
Generated Image	7.9957	7.9949	7.9958
Random Image	7.9956	7.9947	7.9953

Based on these results, it is concluded that the generated images are indistinguishable from random images and there is no statistical information leaked about the client's password.

B. PROBABILISTIC PROPERTY

This section tests the probabilistic property of the proposed system by measuring the correlation coefficients between two different images generated using the same $CText, P, S, S_1$ and S_2. Since each image is initially generated randomly, these images should be uncorrelated.

The correlation coefficient between two images x, y is calculated as follows [19]:

$$CC(x, y) = \frac{E[x - E(x)] \times E[y - E(y)]}{\sqrt{E[x - E(x)]^2 \times E[y - E(y)]^2}} \tag{2}$$

where E is the average intensity of the image pixels. The lower the correlation coefficients between the generated images, the better the probabilistic property of the proposed system.

This test is repeated by generating five images and measuring the correlation coefficient among the first image and

TABLE 7. The probabilistic property test results.

Test	Correlation Coefficient
Image 1 & Image 2	0.0456
Image 1 & Image 3	0.0572
Image 1 & Image 4	0.048
Image 1 & Image 5	0.047

the other four. The same $CText$, P, S, S_1 and S_2 are used in generating these images.

The results are illustrated in Table 7. The images are shown in Figures 11.

Although these images are generated using the same parameters, they are highly uncorrelated. Consequently, the proposed system is probabilistic. This process makes the system immune to dictionary and brute force attacks.

C. THE SENSITIVITY

An essential property for the proposed system is the diffusion property. It measures the sensitivity of the proposed system for changes in its parameters ($CText$, P, S, S_1 and S_2). A small change in any parameter must result in tremendous changes in the generated image. This property is essential for the proposed system to withstand cryptanalysis attacks [20], [21]. The sensitivity of the proposed system is tested by changing 1 bit in each of the challenge text $CText$, the password P, and the seeds S, S_1 and S_2 one at a time while keeping the other parameters constant. The modifications among the original image and the modified images are calculated using the number of pixels change rate (NPCR) and the unified average changing intensity (UACI) [20], [21]. The NPCR calculates the percentage of the total number of different pixels between two images to the total amount of pixels in these two images, while the UACI calculates the average intensity of the differences between the pixels of these two images. The more the values of NPCR and UACI, the better the sensitivity and the diffusion property of the proposed system.

Let two images x, y whose $CText$ or P or S or S_1 or S_2 be different in only one bit. Let i, j be the horizontal and vertical indices of the pixels in both images. Define an array $D_{i,j} = 0$ if $x_{i,j} = y_{i,j}$, otherwise $D_{i,j} = 1$. The NPCR and the UACI are calculated as [20], [21]:

$$NPCR = \frac{1}{H \times W} \sum_{i,j} D_{i,j} \times 100\% \tag{3}$$

$$UACI = \frac{1}{H \times W} \sum_{i,j} \frac{|x_{i,j} - y_{i,j}|}{255} \times 100\% \tag{4}$$

where W, H represent the dimensions of the image [20], [21]. The original image is compared with respect to five images different in 1-bit in $CText$, P, S, S_1 and S_2 respectively. These images are shown in Figures 12. The NPCR and the UACI results are shown in Table 8.

Based on Table 8, 1) the proposed system is sensitive to any changes to its parameters, 2) 96% of the pixels in the original and the changed images are different from each other and 3) the average intensity measurements between the original and the changed images are more than 32%.

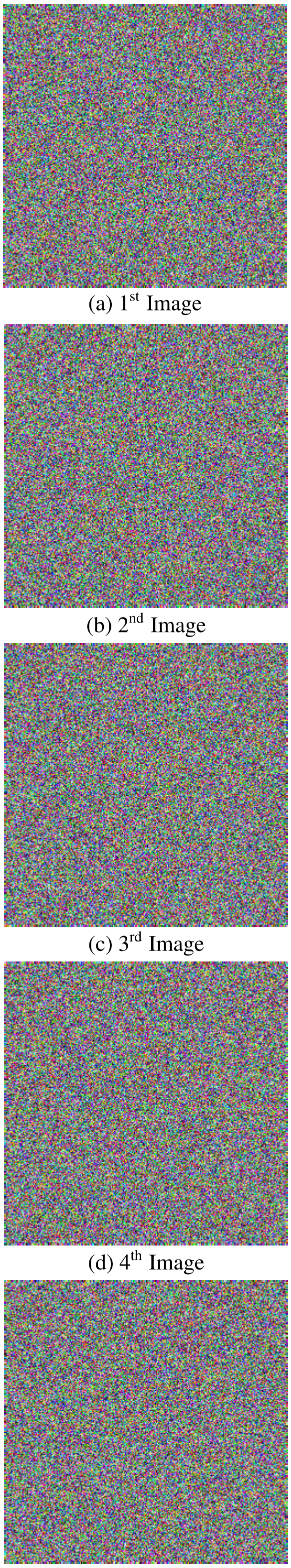
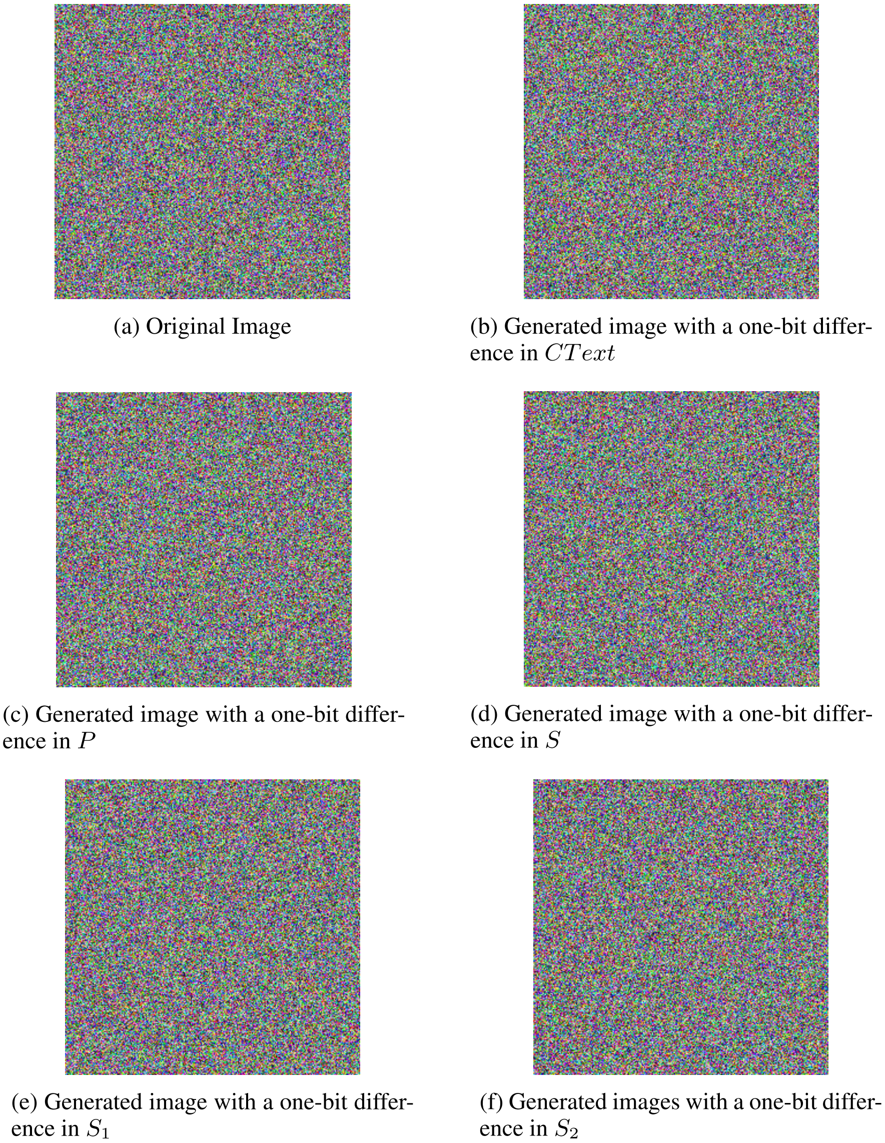

(a) 1st Image
(b) 2nd Image
(c) 3rd Image
(d) 4th Image
(e) 5th Image

FIGURE 11. The generated images using the same $CText$, P, S, S_1 and S_2 parameters.



(a) Original Image

(b) Generated image with a one-bit difference in $CText$

(c) Generated image with a one-bit difference in P

(d) Generated image with a one-bit difference in S

(e) Generated image with a one-bit difference in S_1

(f) Generated images with a one-bit difference in S_2

FIGURE 12. The generated images with a one-bit difference in *CText*, *P*, *S*, S_1 and S_2.

TABLE 8. Sensitivity test results.

Changed Value	NPCR	UACI
P	96.8664%	32.3503%
$CText$	97.0139%	32.3307%
S	99.6134%	32.4087%
S_1	99.6277%	32.4117%
S_2	99.5997%	33.3050%

Based on these tests, the proposed system has a very strong diffusion property and is highly sensitive to changes in its parameters.

VII. IMMUNITY TO KNOWN ATTACKS

This section compares between the proposed system and the traditional system in terms of their resistance against known attacks such as the brute force attack and the replay attack.

A. IMMUNITY AGAINST THE BRUTE FORCE ATTACK

The traditional system protects the password using the hash function and utilizes the seed to randomize the output of the hash function for the same password. To brute forcefully attack the hash, an adversary needs to try 2^n different combinations where n is the hash size. Assuming a hash size of 256 bits, an adversary needs to try $2^{256} = 1.15 \times 10^{77}$ different combinations to get the password P. Assume that this adversary uses a computer that can try $2^{50} = 1.12 \times 10^{15}$ combinations per second, then the adversary will be able to break the hash and obtain the password in 2^{206} seconds or 3.26×10^{54} years. This makes the traditional system secure for the current computing capabilities. But this is not true for quantum computing, because quantum computing can break the hash in few seconds [22].

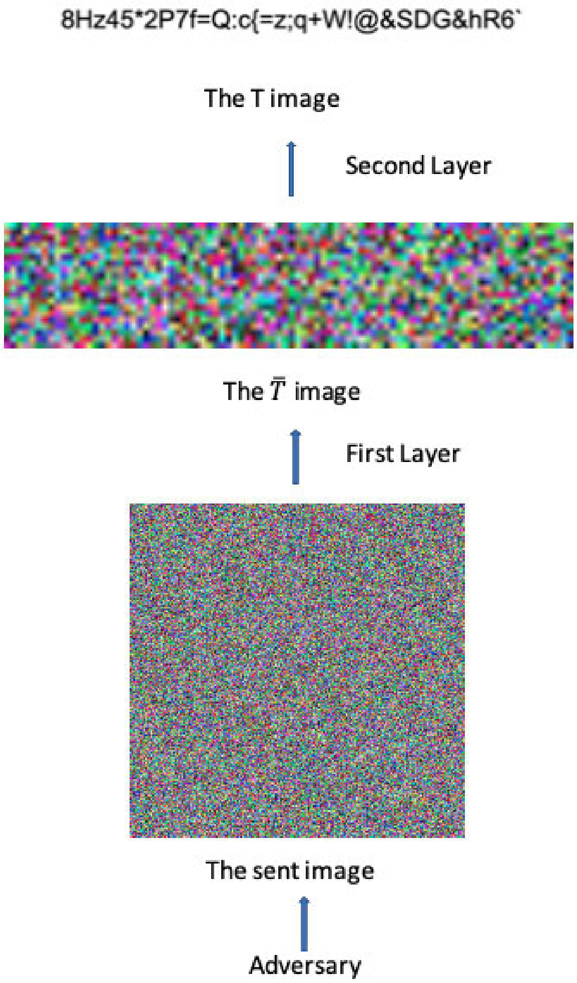


FIGURE 13. **The security layers of the proposed system.**

The proposed system protects the hash by concealing it inside the image. To successfully extract the hash from the image, the adversary needs to bypass two layers of security. The first layer is extracting $\bar{T}$ from the random image, while the second layer is extracting T from $\bar{T}$. These layers of security are shown in Figure 13. After obtaining T, the adversary needs to extract the characters of T using OCR, calculate $H(P, S) = CText \oplus T$ and then break the hash.

The first layer depends on the relation between the sizes of the random and the $\bar{T}$ images, while the second layer depends on the relation between the sizes of the $\bar{T}$ and T images.

1) THE FIRST SECURITY LAYER

There are two approaches to extract $\bar{T}$ from the random image; the first is brute forcefully attacking the random image by trying every possible combination of $\bar{T}$. The second approach is achieved through extracting the pixels of $\bar{T}$ scrambled, then breaking the scrambling algorithm to get $\bar{T}$. The second approach is more effective than the first one.

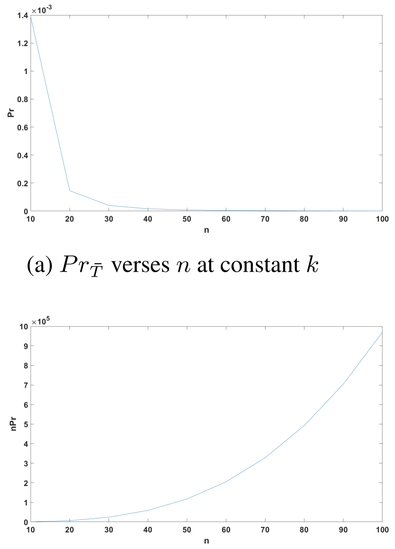
Let the number of pixels in $\bar{T}$ be k and the number of pixels in a random image be n. The number of possible ways to store $\bar{T}$ inside the random image is:

$$nPr_{\bar{T}}(n, k) = \frac{n!}{(n-k)!} \tag{5}$$

where $nPr_{\bar{T}}$ is the permutation function. The probability of finding the specific $\bar{T}$ inside the random image is:

$$Pr_{\bar{T}}(n, k) = \frac{1}{nPr_{\bar{T}}(n, k)} = \frac{(n-k)!}{n!} \tag{6}$$

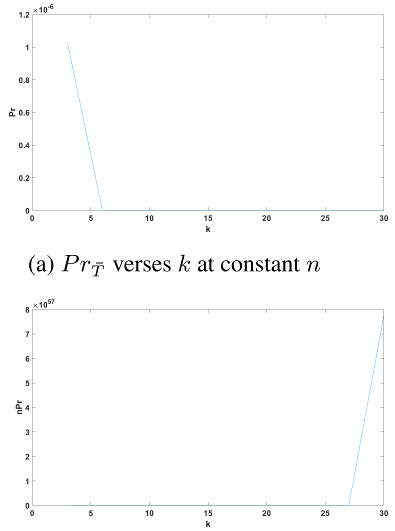
The relations between $Pr_{\bar{T}}$ and $nPr_{\bar{T}}$ with respect to n at constant k are shown in Figure 14, while the relations between $Pr_{\bar{T}}$ and $nPr_{\bar{T}}$ with respect to k at constant n are shown in Figure 15.



(a) $Pr_{\bar{T}}$ verses n at constant k

(b) $nPr_{\bar{T}}$ verses n at constant k

FIGURE 14. **$Pr_{\bar{T}}$ and $nPr_{\bar{T}}$ verses n at constant k.**



(a) $Pr_{\bar{T}}$ verses k at constant n

(b) $nPr_{\bar{T}}$ verses k at constant n

FIGURE 15. **$Pr_{\bar{T}}$ and $nPr_{\bar{T}}$ verses k at constant n.**

From these figures, it is deduced that $nPr_{\bar{T}}$ increases with n, k while $Pr_{\bar{T}}$ decreases with n, k. Using sufficiently large values of n, k will give very large possible combinations $nPr_{\bar{T}}$ and the probability of finding $\bar{T}$ image $Pr_{\bar{T}}$ will be very small, making a brute force attack of finding $\bar{T}$ an extremely difficult task. For demonstration, if an image $\bar{T}$ with 1800 pixels and a random image with 65536 pixels are used. Then $nPr_{\bar{T}}$ and $Pr_{\bar{T}}$ can be calculated as:

$$nPr_{\bar{T}} = \frac{65536!}{(65536-1800)!} = 6.83 \times 10^{8658} \tag{7}$$

$$Pr_{\bar{T}} = \frac{(65536-1800)!}{65536!} = 1.46 \times 10^{-8659} \tag{8}$$

This means that a computer requires 6.83×10^{8658} different combinations to find the correct $\bar{T}$ and the probability to find such $\bar{T}$ is 1.46×10^{-8659}. If a computer tries 1×10^{100} combinations per second, it will require approximately 6.83×10^{8558} seconds or 2.16×10^{8551} years to try all the combinations.

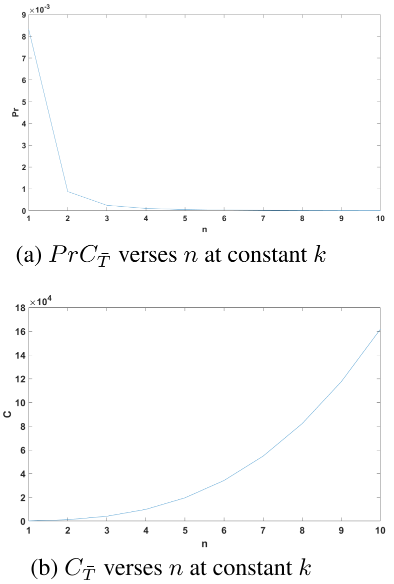
The second approach to extract $\bar{T}$ is to get the pixels of $\bar{T}$ out of order then break the scrambling algorithm to get $\bar{T}$.

The number of possible ways to store the pixels of $\bar{T}$ out of order inside the random image can be calculated as:

$$C_{\bar{T}}(n, k) = \frac{n!}{(n-k)! \times k!} \tag{9}$$

where $C_{\bar{T}}$ is the combination function. The probability of finding the pixels of $\bar{T}$ out of order inside the random image is:

$$PrC_{\bar{T}}(n, k) = \frac{1}{C_{\bar{T}}(n, k)} = \frac{(n-k)! \times k!}{n!} \tag{10}$$

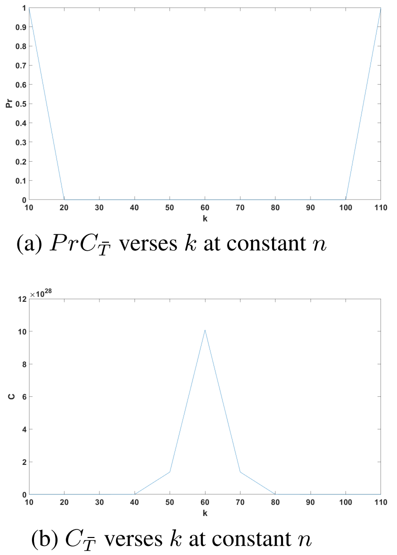


(a) $PrC_{\bar{T}}$ verses n at constant k

(b) $C_{\bar{T}}$ verses n at constant k

FIGURE 16. *PrC*$_{\bar{T}}$ and *C*$_{\bar{T}}$ verses *n* at constant *k*.

The relations between $PrC_{\bar{T}}$ and $C_{\bar{T}}$ with respect to n at constant k are shown in Figure 16, while the relations between $PrC_{\bar{T}}$ and $C_{\bar{T}}$ with respect to k at constant n are shown in Figure 17.



(a) $PrC_{\bar{T}}$ verses k at constant n

(b) $C_{\bar{T}}$ verses k at constant n

FIGURE 17. *PrC*$_{\bar{T}}$ and *C*$_{\bar{T}}$ verses *k* at constant *n*.

Based on Figures 16 and 17, $C_{\bar{T}}$ increases with n for constant k. For constant n, $C_{\bar{T}}$ increases when $k < \frac{n}{2}$ and decreases when $k > \frac{n}{2}$. This is because at small k values, $(n-k)! \gg k!$ and $C_{\bar{T}}$ is increasing, while at large k values, $(n-k)! \ll k!$ and $C_{\bar{T}}$ is decreasing. $PrC_{\bar{T}}$ decreases with n for constant k. For constant n, $PrC_{\bar{T}}$ decreases when $k < \frac{n}{2}$ and increases when $k > \frac{n}{2}$ for the same reasons mentioned above. Based on this analysis, it is recommended to have $k < \frac{n}{2}$.

For demonstration, if $n = 65536$ and $k = 1800$ then $C_{\bar{T}}$ and $PrC_{\bar{T}}$ can be calculated as follows:

$$C_{\bar{T}} = \frac{65536!}{(65536-1800)! \times 1800!} = 1.11 \times 10^{3579} \tag{11}$$

$$PrC_{\bar{T}} = \frac{(65536-1800)! \times 1800!}{65536!} = 8.96 \times 10^{-3580} \tag{12}$$

This means that a computer requires 1.11×10^{3579} different combinations to find the pixels of $\bar{T}$ out of order and the probability to find such pixels is 8.96×10^{-3580}. If a computer tries 1×10^{100} combinations per second, it will require approximately 1.11×10^{3479} seconds or 3.54×10^{3471} years to get the pixels of $\bar{T}$.

After finding the pixels of $\bar{T}$, it will be an easy task for a quantum computer to break the scrambling algorithm. That is why in this analysis the probability of a quantum computer to break a scrambling algorithm is ignored i.e. P (breaking the scrambling algorithm) ≈ 1.

2) THE SECOND SECURITY LAYER

The same analysis can be made for extracting T from $\bar{T}$ as follows. For an image $\bar{T}$ with a size of k bits and an image T with a size of l bits, nPr_T and Pr_T can be calculated as:

$$nPr_T = \frac{k!}{(k-l)!} \tag{13}$$

$$Pr_T = \frac{(k-l)!}{k!} \tag{14}$$

The second approach that is used to extract $\bar{T}$ from the image is not applicable here. That is because $T = CText \oplus H(P_c, S)$. Since the hash function has a random distribution, T will have a random distribution because of the XORing operation.

Assuming that both images have a size of 1800 pixels (i.e. $l = k = 1800$), nPr_T and Pr_T can be calculated as:.

$$nPr_T = \frac{k!}{(k-l)!} = \frac{(1800)!}{(1800-1800)!} = 6.12 \times 10^{5079} \quad (15)$$

$$Pr_T = \frac{(k-l)!}{k!} = \frac{(1800-1800)!}{1800!} = 1.63 \times 10^{-5080} \quad (16)$$

This means that a computer requires 6.13×10^{5079} different combinations to find the pixels of T and the probability of finding such pixels is 1.63×10^{-5080}. If a computer tries 1×10^{100} combinations per second, it will require approximately 6.12×10^{4979} seconds or 1.98×10^{4972} years to get the pixels of T.

After successfully extracting T, the adversary needs to calculate $H(P, S) = CText \oplus T$ and break the hash to obtain the password. The probability of breaking a 256-bit hash can be calculated as:

$$Pr_{Hash} = \frac{1}{2^{256}} = 8.63 \times 10^{-78} \quad (17)$$

To summarize, the probability for the adversary to break the proposed system and obtain the password is equal to the probability to get $\bar{T}$ multiplied by the probability to get T multiplied by the probability to break the hash.

$$\begin{aligned} Pr_P &= PrC_{\bar{T}} \times Pr_T \times Pr_H \\ &= 8.96 \times 10^{-3580} \times 1.63 \times 10^{-5080} \times 8.63 \times 10^{-78} \\ &= 1.26 \times 10^{-8736} \end{aligned} \quad (18)$$

The number of different combinations the adversary has to try for breaking the system is $\frac{1}{Pr_P} = 7.93 \times 10^{8735}$. If a computer tries 1×10^{100} combinations per second, it will require approximately 7.93×10^{8635} seconds or 2.58×10^{8628} years to get the password.

This section concludes that brute forcefully breaking the proposed system requires enormous amount of computing power and time and breaking the proposed system will be a challenge even for quantum computing.

A comparison between traditional systems and the proposed system in terms of resisting a brute force attack is shown in Table 9

TABLE 9. A comparison between traditional systems and the proposed system in terms of resisting a brute force attack.

System	Number of operations	The probability
Proposed	7.93×10^{8735}	1.26×10^{-8736}
Traditional	1.15×10^{77}	8.63×10^{-78}

This means that the proposed system is more powerful than traditional systems by a factor of 6.89×10^{8658}. If a quantum computer can break a hash function in just one second, then this computer will be able to break the proposed system in 2.17×10^{8650} years.

This can be achieved without sacrificing the space-efficiency of the proposed system; a 256 × 256 random RGB image has a size of 193KB when stored in Portable Network Graphics (PNG) file format.

B. IMMUNITY AGAINST THE REPLAY ATTACK

In traditional systems, if an adversary obtains $S, H(P, S)$, a replay attack can be launched. The adversary can resend $S, H(P, S)$ to the server pretending to be the legitimate client. The adversary can hack the system without knowing the password P. This attack is not possible with the proposed system because the server sends random $CText$ to the client. If an adversary obtains a generated image along with S, S_1, S_2, the authentication will fail because the server sends a different $CText$.

VIII. THE SECURITY PROOF

This section proves that the proposed system is secure by proving the following theorem.

Theorem 1: Suppose the CAPTCHA AI hard problem holds and the scrambling algorithm is secure. Then the proposed system is indistinguishably secure against an adaptive chosen-challenge text attack (IND-ACCTA) based on the CAPTCHA AI hard problem when the hash function H is modeled as a random oracle [23]. Let $\mathcal{A}$ be an efficient IND-ACCTA adversary whose running time is at most τ, then there are two efficient algorithms B_1 and B_2 whose running time is the same as the running time of $\mathcal{A}$ such that:

$$Adv_{\mathcal{A},pr}(n,k) \leq Adv_{B_1,C} + \frac{(n-k)! \times k!}{n!} \times Adv_{B_2,S} \quad (19)$$

where $Adv_{\mathcal{A},pr}$ is the advantage of $\mathcal{A}$ to break the proposed system, $Adv_{B_1,C}$ is the advantage of algorithm B_1 to break CAPTCHA and $Adv_{B_2,S}$ is the advantage of algorithm B_2 to break the scrambling algorithm.

Proof: To prove Theorem 1, the CAPTCHA AI hard problem and the security notion of challenge text confidentiality are defined. After that, a game played between an adversary and a challenger who challenges the adversary to break the proposed system is presented.

1) THE CAPTCHA AI HARD PROBLEM

CAPTCHA is an information security algorithm that creates challenges that are easy for most humans but hard and unsolvable for bots [3]. Its security is based on the hardness assumption of AI hard problems. A bot that solves a challenge C generated by CAPTCHA can be used to solve these AI hard problems [3].

The definition of CAPTCHA is modified here by elevating the restriction of being easy for humans. The CAPTCHA AI hard problem in this article should be hard for both humans and bots who do not have the secret key required to solve the challenge C. This will harden the CAPTCHA AI hard problem and will make it more difficult for an adversary to solve.

Definition 1: An AI problem is defined as a tripartite $P = (Q, R, b)$, such that Q is a set of hard problem cases, R is the probability distribution over the set Q, and $b : Q \rightarrow \{0, 1\}^*$ is the answer to the set Q. Assume an adversary $\mathcal{A}$ whose running time is at most τ for any input from the set Q. $\mathcal{A}$ receives a problem P as input and outputs $\bar{b} : Q \rightarrow \{0, 1\}^*$. $\mathcal{A}$ solves the hard problem P if $b = \bar{b}$. The advantage of $\mathcal{A}$ to solve P is:

$$Adv_{\mathcal{A}}(Q, D) = |\Pr[\bar{b} = b] - \frac{1}{2}| \tag{20}$$

Definition 2: An η-CAPTCHA is a challenge C that is defined as follows. Assume that there is an AI hard problem P and an adversary $\mathcal{A}$. $\mathcal{A}$ can solve the challenge C if the adversary $\mathcal{A}$ runs a program B whose running time is the same as $\mathcal{A}$ and B has success greater than η over P.

Von Ahn *et al.* [4] introduced a set of hard problems that can be used to construct CAPTCHA images as shown in Figure 1. If a computer can solve these hard problems, then the twisted texts can be extracted from the CAPTCHA images and the CAPTCHA AI hard problem can be solved.

Assume that there is a colored RGB image with a height H and width W. Define an image transformation that takes an image as input and produces another image as output (not essentially of the similar width and height). Instances of image transformations are altering the size of an image, transforming an image into its negative version, etc.

Choose I from a set of colored RGB images and D from a set of image transformations. Assume for simplicity that D is a one-to-one transformation. More formally, $D(I) \neq D(\bar{I})$ if $I \neq \bar{I}$.

Definition 3: Problem Family (P). Consider writing a program that takes $D(I)$ as input and outputs I (i.e. find the reverse of the transformation D). More formally, let $Q_{D,I}$ be the set of all possible transformations $\{D, I\} \rightarrow D(I)$, $R_{D,I}$ be the probability distribution on $Q_{D,I}$ that comes from executing the program and $b_{D,I}(D(I)) = I$. Then $P_{D,I} = (Q_{D,I}, R_{D,I}, b_{D,I})$ [3].

Von Ahn *et al.* [4] proved that an adversary $\mathcal{A}$ can break the CAPTCHA AI hard problem if $\mathcal{A}$ is running a program B whose running time is close to that of $\mathcal{A}$ and the program B can solve the problem family P.

2) THE CHALLENGE TEXT CONFIDENTIALITY NOTION

This section presents a new security notion for the proposed system called the challenge text confidentiality notion. The proposed system must maintain this notion that models an adversary who tries to distinguish the challenge text from a random text under an adaptive chosen-challenge text attack. The challenge text confidentiality is illustrated as a game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. This game is as follows.

- Setup (n, k): $\mathcal{C}$ produces the public parameters n, k and transfers them to $\mathcal{A}$ and hides the password (P).
- Query Phase 1: In this phase, $\mathcal{A}$ can query $H(P, S)$, $H(P, S_1)$ and $H(P, S_2)$ based on S, S_1 and S_2 of his choosing. Upon receiving the queried S, S_1 and S_2 from $\mathcal{A}$, $\mathcal{C}$ calculates $H(P, S), H(P, S_1)$ and $H(P, S_2)$ and sends the results to $\mathcal{A}$. In addition, $\mathcal{A}$ can query a generated image based on $CText, S, S_1$ and S_2 of his choice. To answer his query, the challenger calculates $T = CText \oplus H(P, S)$, generates a random image, converts T into an image, gets $\bar{T} = T \oplus H(P, S_1)$ and scrambles $\bar{T}$ inside the random image using $H(P, S_2)$ as shown in Figure 3. Then the challenger sends the generated image to the adversary $\mathcal{A}$.
- Challenge Phase: $\mathcal{A}$ sends to $\mathcal{C}$ two challenge texts $CText_1$ and $CText_2$ and three seeds S^*, S_1^* and S_2^* of his choosing. $\mathcal{C}$ tosses a fair coin $b \in [0, 1]$ and uses $CText_b$ as the challenge text. Using $CText_b, S^*, S_1^*$ and S_2^*, the challenger $\mathcal{C}$ generates the challenge image and sends it to the adversary $\mathcal{A}$. It is noted here that the challenged $CText_b, S^*, S_1^*$ and S_2^* must not be queried before in the query phase.
- Query Phase 2: The adversary can make adaptive queries as in phase 1 but the challenged parameters.
- Guess: $\mathcal{A}$ outputs $\bar{b} \in [0, 1]$. $\mathcal{A}$ wins the game if $b = \bar{b}$.

The advantage of $\mathcal{A}$ to break a system z and win this game is:

$$Adv_{\mathcal{A},z}(n, k) = |\Pr[\bar{b} = b] - \frac{1}{2}| \tag{21}$$

Definition 4: The proposed system is indistinguishably secure against an adaptive chosen-challenge text attack (IND-ACCTA) if the advantage of any polynomially bounded adversary $\mathcal{A}$ in the above game is negligible.

3) THE PROOF

To prove Theorem 1, this section first proves the following Lemma.

Lemma 1: Any value x that is XORed with a uniformly distributed value y results in a uniformly distributed value z regardless of the distribution of x.

Proof: Assume that the probability of x to be one is P. Then the probability of x to be zero is $1 - P$. The truth table of x and a random value y is shown in Table 10.

TABLE 10. Truth table for $z = x \oplus y$.

x	y	$z = x \oplus y$	P_x	P_y	$P_{z=x\oplus y}$
1	0	1	P	0.5	$0.5P$
1	1	0	P	0.5	$0.5P$
0	0	0	$1 - P$	0.5	$0.5(1 - P)$
0	1	1	$1 - P$	0.5	$0.5(1 - P)$

The probability that $z = 0$ is $0.5P + 0.5(1 - P) = 0.5$. The same goes for $z = 1$.

Now, the security proof of the proposed system is introduced.

- This segment describes an arrangement of games. Let W_i be the winning of the i_{th} game by the adversary $\mathcal{A}$. These games are described as follows.
  - **Game-0**. This game is the usual adversarial game.
  - **Game-1**. This game illustrates how to reply to a generated image query from $\mathcal{A}$.

  - **Game-2**. This game replaces the hash function H with a truly random function.
  - **Game-3**. This game replaces T with a randomly uniform distributed value.
  - **Game-4**. This game proves that extracting T from $\bar{T}$ is a CAPTCHA AI hard problem.
  - **Game-5**. This game scrambles $\bar{T}$ inside the random image.
  - **Game-6** replaces the challenge text T with a random text Z.
- Game-0. This is the typical adversarial game for defining the IND-ACCTA security of the proposed system. The challenger $\mathcal{C}$ picks the public parameters (n, k) and sends them to $\mathcal{A}$. The challenger also picks a random oracle $H : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N}$ at random from the group of all similar functions in the Setup algorithm and permits $\mathcal{A}$ to query H at random points. Thus.

$$|\Pr[W_0] - \frac{1}{2}| = Adv_{\mathcal{A},pr}(n, k) \tag{22}$$

- Game-1. This game explains how to respond to a generated image query from $\mathcal{A}$. Upon receiving a challenge text *CText* and three seeds S, S_1 and S_2 from $\mathcal{A}$, the challenger $\mathcal{C}$ calculates $T = CText \oplus H(P, S)$, generates a random image, converts T into an image, gets $\bar{T} = T \oplus H(P, S_1)$ and scrambles $\bar{T}$ inside the random image using $H(P, S_2)$ as shown in Figure 3. Then the challenger sends the generated image to the adversary $\mathcal{A}$. Since this is similar to Game-0, thus.

$$|\Pr[W_0] = \Pr[W_1]| \tag{23}$$

- Game-2. The hash function H is replaced in Game-1 with a truly random function. To answer a hash query from $\mathcal{A}$ for S, S_1 and S_2, the challenger $\mathcal{C}$ builds a hashing table with two columns; the first column is for the queried values of S, S_1 and S_2 and the other column is for randomly generated values that correspond to S, S_1 and S_2. When $\mathcal{C}$ receives queries of $H(P, S)$, $H(P, S_1)$ and $H(P, S_2)$ from $\mathcal{A}$ for S, S_1 and S_2 of his choice, $\mathcal{C}$ looks up the hashing table for S, S_1 and S_2. If they exist, then $\mathcal{C}$ answers the queries with the random values associated with S, S_1 and S_2. If S, S_1 and S_2 do not exist in the table, $\mathcal{C}$ generates three random values, sends them to $\mathcal{A}$ and then puts these random values in the hashing table alongside the queried values of S, S_1 and S_2. Building this table ensures that the generated values are collision-free (each generated random value is unique and matches with a unique queried value of S, S_1 and S_2).
  This replaces the hashes $H(P, S)$, $H(P, S_1)$ and $H(P, S_2)$ with three randomly uniform distributed variables x, y and z. Then $T = CText \oplus z$, $\bar{T} = T \oplus x$ and $\bar{T}$ is scrambled using y. Since in the random oracle model (ROM), the hash is viewed as a truly random function generating random values with uniform distribution, the adversary $\mathcal{A}$ will not be capable of distinguishing between Game-1 and Game-2. Thus.

$$|\Pr[W_1] = \Pr[W_2]| \tag{24}$$

- Game-3. $T = CText \oplus z$ is replaced with a random value. Based on Lemma 1, T will be random and uniformly distributed because z is random and uniformly distributed regardless of *CText*. Based on that, the adversary $\mathcal{A}$ will not distinguish between Game-3 and Game-2. Thus.

$$|\Pr[W_2] = \Pr[W_3]| \tag{25}$$

- Game-4. This game proves that extracting T from $\bar{T}$ is a CAPTCHA AI hard problem. A CAPTCHA AI hard problem is defined based on Definition 3 as follows. Let $I = T$ and $D(I) = \bar{T}$. Let $Q_{D,I}$ be the transformation $\bar{T} = T \oplus x$. Since T is an unknown random value (because x is an unknown random value), the transformation $Q_{D,I}$ is irreversible. In addition, $R_{D,I}$ is uniform based on Lemma 1. Then $b_{D,I}(\bar{T}) = T$ and extracting T from $\bar{T}$ is a CAPTCHA AI hard problem $P_{D,I} = (Q_{D,I}, R_{D,I}, b_{D,I})$. To extract T from $\bar{T}$, the adversary $\mathcal{A}$ needs an algorithm B_1 whose running time is close to the running time of $\mathcal{A}$ and can break CAPTCHA. Therefore.

$$|\Pr[W_4] - \Pr[W_3]| = Adv_{B_1,C} \tag{26}$$

- Game-5. The $\bar{T}$ image is scrambled inside a random image using the secret, randomly distributed variable y. This requires the adversary to extract the CAPTCHA image from the random image. Using the second approach presented in section VII-A1 (since it is more efficient than the first one) and assuming that there is an algorithm B_2 that breaks the scrambling algorithm (whose running time is close to the running time of $\mathcal{A}$), thus.

$$|\Pr[W_5] - \Pr[W_4]| = \frac{(n-k)! \times k!}{n!} \times Adv_{B_2,S}(n, k) \tag{27}$$

- Game-6: This game replaces the challenge text *CText* in CAPTCHA with a random text Z. Since the challenge text *CText* in CAPTCHA is unrecognizable from random text, $\mathcal{A}$ will not be able to differentiate between Game-6 and Game-5. Therefore.

$$|\Pr[W_6] = \Pr[W_5]| \tag{28}$$

- Clearly in Game-6.

$$|\Pr[W_6] = \frac{1}{2}| \tag{29}$$

  Combining all the previous equations proves Theorem 1.

IX. CONCLUSION

This article presented a novel password-based authentication system. Unlike traditional authentication systems based on mathematical hard problems, the proposed system is based on the CAPTCHA AI hard problem. This requires bots to be humanly smart. This is a challenge even for quantum computing.

The authentication process starts by sending a challenge text from the server to the client. The client sends back the challenge text concealed by his password inside a random image. The detailed analysis proved that the proposed system is time- and space efficient, probabilistic, sensitive to changes in its parameters, immune to leakage of any statistical information about the client's password and the generated images are indistinguishable from random images. Also, the security analysis proved that the proposed system is immune to brute force and replay attacks. In addition, the proposed system is proven to be indistinguishably secure against an adaptive chosen-challenge text attack (IND-ACCTA) based on the CAPTCHA AI hard problem when the hash function H is modeled as a random oracle. The intent behind this article is to address the challenges presented by the impending advent of powerful quantum computing. These findings represent an important step towards that future.

X. FUTURE WORK

In the future, it is intended to investigate the following questions in depth.

- How will AI hard problems in general and the CAPTCHA AI hard problem in particular stand against an actual quantum computer?
- Can the world depend on AI hard problems to protect the information in the quantum computing era?
- Can the proposed system withstand an attack from an actual quantum computer?
- Can other cryptographic systems, such as encryption, digital signature, and key exchange systems be built using AI hard problems?

REFERENCES

[1] C. Meshram and M. S. Obaidat, "An efficient provably secure ibs technique using integer factorization problem," in *Proc. 1st Int. Conf. Comput., Commun., Cyber-Secur.*, 2020, pp. 427–439.

[2] Y. Huang, Z. Su, F. Zhang, Y. Ding, and R. Cheng, "Quantum algorithm for solving hyperelliptic curve discrete logarithm problem," *Quantum Inf. Process.*, vol. 19, no. 2, p. 62, Feb. 2020.

[3] F. H. Alqahtani and F. A. Alsulaiman, "Is image-based CAPTCHA secure against attacks based on machine learning? An experimental study," *Comput. Secur.*, vol. 88, Jan. 2020, Art. no. 101635.

[4] L. Von Ahn, M. Blum, N. J. Hopper, and J. Langford, "Captcha: Using hard ai problems for security," in *Proc. Int. Conf. theory Appl. Cryptograph. Techn.*, 2003, pp. 294–311.

[5] M. M. S. Sonwalkar, "Captcha: Novel approach to secure user," *Pramana Res. J.*, vol. 10, no. 1, pp. 106–114, 2020.

[6] M. H. Barkadehi, M. Nilashi, O. Ibrahim, A. Zakeri Fardi, and S. Samad, "Authentication systems: A literature review and classification," *Telematics Informat.*, vol. 35, no. 5, pp. 1491–1511, Aug. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0736585318301400

[7] Y. Wenlong, "Authentication method, apparatus, and system," U.S. Patent 10 516 666, Dec. 24, 2019.

[8] W. Zugaj and A. S. Beichler, "Analysis of standard security features for selected nosql systems," *Amer. J. Inf. Sci. Technol.*, vol. 3, no. 2, pp. 41–49, 2019.

[9] H. Ning, H. Liu, and L. T. Yang, "Aggregated-proof based hierarchical authentication scheme for the Internet of Things," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 3, pp. 657–667, Mar. 2015.

[10] S. Alharbi, P. Rodriguez, R. Maharaja, P. Iyer, N. Subaschandrabose, and Z. Ye, "Secure the Internet of Things with challenge response authentication in fog computing," in *Proc. IEEE 36th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2017, pp. 1–2.

[11] I. Sluganovic, M. Roeschlin, K. B. Rasmussen, and I. Martinovic, "Using reflexive eye movements for fast challenge-response authentication," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 1056–1067.

[12] S. Prabhu and V. Shah, "Authentication using session based passwords," *Procedia Comput. Sci.*, vol. 45, pp. 460–464, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915003154

[13] H.-P. Ren, C.-F. Zhao, and C. Grebogi, "One-way hash function based on delay-induced hyperchaos," *Int. J. Bifurcation Chaos*, vol. 30, no. 02, Feb. 2020, Art. no. 2050020.

[14] H. Alhumyani, "Efficient image cipher based on baker map in the discrete cosine transform," *Cybern. Inf. Technol.*, vol. 20, no. 1, pp. 68–81, 2020.

[15] X. Liao, J. Yin, S. Guo, X. Li, and A. K. Sangaiah, "Medical JPEG image steganography based on preserving inter-block dependencies," *Comput. Electr. Eng.*, vol. 67, pp. 320–329, Apr. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790617302756

[16] Y. Yang, D. He, N. Kumar, and S. Zeadally, "Compact hardware implementation of a sha-3 core for wireless body sensor networks," *IEEE Access*, vol. 6, pp. 40128–40136, 2018.

[17] G. Hua, Y. Xiang, and L. Y. Zhang, "Informed histogram-based watermarking," *IEEE Signal Process. Lett.*, vol. 27, pp. 236–240, 2020.

[18] P. Zhen, G. Zhao, L. Min, and X. Jin, "Chaos-based image encryption scheme combining DNA coding and entropy," *Multimedia Tools Appl.*, vol. 75, no. 11, pp. 6303–6319, Apr. 2015.

[19] X. Hu, A. Jung, and G. Qin, "Interval estimation for the correlation coefficient," *Amer. Statistician*, vol. 74, no. 1, pp. 29–36, Jan. 2020.

[20] M. Zhou and C. Wang, "A novel image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks," *Signal Process.*, vol. 171, Jun. 2020, Art. no. 107484.

[21] W. S. Sayed, A. G. Radwan, H. A. H. Fahmy, and A. El-Sedeek, "Software and hardware implementation sensitivity of chaotic systems and impact on encryption applications," *Circuits, Syst., Signal Process.*, vol. 39, no. 5, May 2020. [Online]. Available: https://doi-org.sdl.idm.oclc.org/10.1007/s00034-020-01424-8

[22] L. Gyongyosi and S. Imre, "A survey on quantum computing technology," *Comput. Sci. Rev.*, vol. 31, pp. 51–71, Feb. 2019.

[23] K. Hövelmanns, E. Kiltz, S. Schäge, and D. Unruh, "Generic authenticated key exchange in the quantum random oracle model," in *Proc. IACR*, 2020, pp. 389–422.

MASOUD ALAJMI (Member, IEEE) received the B.S. degree in electrical engineering from the King Fahd University of Petroleum and Minerals, in 2004, and the M.S. degree in electrical engineering and the Ph.D. degree in electrical and computer engineering from Western Michigan University, Kalamazoo, MI, USA, in 2010 and 2016, respectively. He has four years of experience in industry. He was an Electrical Engineer with Zamel and Turbag Consulting Engineers, Al-Khobar, Saudi Arabia, for three months. He was a Pre-Commissioning Engineer with Saudi Electricity Company (SEC), Abha, Saudi Arabia, from 2004 to 2008. During that period, he completed many training programs in the technical and administrative fields with well-known institutes. He was assigned to be a Commissioning Leader for many projects in Saudi Arabia. He was assigned to be the SEC Representative to supervise factory acceptance tests for Siemens Company, Berlin, Germany, in 2007, and then for Hyundai Heavy Industries Company Ltd., Ulsan, South Korea, in 2008. From 2012 to 2015, he was a Teaching Assistant with the Electrical and Computer Engineering Department, Western Michigan University, where he received the 2014–2015 Graduate Teaching Effectiveness Award for excellent teaching skills. He is currently an Associate Professor with the Computer Engineering Department, Taif University, Taif, Saudi Arabia. He is a coauthor of about 20 papers in international journals and conference proceedings. His research interests include wireless communications, signal processing, image encryption, watermarking, steganography, data hiding, biomedical image processing, machine learning, and smart grids. He is involved in various technical committees.

IBRAHIM ELASHRY received the B.Sc. degree (Hons.) from the Faculty of Engineering, Kafrelshiekh University, Egypt, in 2007, the M.Sc. degree from the Faculty of Electronic Engineering, Menoufia University, in 2010, and the Ph.D. degree in cryptography from The University of Wollongong (UOW), Australia, in 2015. He currently works as a Lecturer with the Faculty of Engineering, Kafrelshiekh University. His research interests include cryptography and information security over wired and wireless networks.

HALA S. EL-SAYED received the B.Sc. (Hons.), M.Sc., and Ph.D. degrees in electrical engineering from Menoufia University, Shebin El-Kom, Egypt, in 2000, 2004, and 2010, respectively. She is currently an Assistant Professor with the Department of Electrical Engineering, Faculty of Engineering, Menoufia University, where she was a Demonstrator, from 2002 to 2004, and has been an Assistant Lecturer, from 2004 to 2010. Since 2010, she has been a Teaching Staff Member with the Department of Electrical Engineering, Faculty of Engineering, Menoufia University. Her research interests include database security, network security, data hiding, image encryption, wireless sensor networks, secure building automation systems, medical image processing, and biometrics.

OSAMA S. FARAGALLAH received the B.Sc. (Hons.), M.Sc., and Ph.D. degrees in computer science and engineering from Menoufia University, Menouf, Egypt, in 1997, 2002, and 2007, respectively. He is currently a Professor with the Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, where he was a Demonstrator, from 1997 to 2002, and has been an Assistant Lecturer, from 2002 to 2007. Since 2007, he has been a Teaching Staff Member with the Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University. His current research interests include network security, cryptography, Internet security, multimedia security, image encryption, watermarking, steganography, data hiding, medical image processing, and chaos theory.