Preventing Multiple Accessing Attacks via Efficient Integration of Captcha Crypto Hash Functions

Nafisah Kheshaifaty[†] **and Adnan Gutub**[††],

[†]Master of Sciences (MS) degree in Computer Science, Umm Al-Qura University, Makkah, Saudi Arabia
[††]Professor in Computer Engineering, Umm Al-Qura University, Makkah, Saudi Arabia

Summary: Authentication is the process of verifying the identity of online computer users. It can be confirmed using various methods such as captchas or encrypted hashed passwords to provide acceptable practicality. The main objective of having these authentication features, is to ensure that intended user data is protected and safe from attackers. Captchas are used as a security access measure, to check if users are real humans or robots. The lately captcha tests include, the three-dimension captcha, the no robot captcha, marketing captcha, and mathematical captcha, which all have applicability challenges to be addressed. In this paper, the text-based captcha will be recommended as part of the security system integrated with crypto hash functions for proper security preventing multiple accessing attacks. Our authentication process contains the captcha module, login system component, and SHA1 hash function, all combined for access security. Beside captcha module check of robot-automation, the SHA1 hashing and cryptography are combined to assure efficient security of the authentication process. The proposed integration of captcha crypto hash functions system has an effective improvement of about 30% over old systems. Its multiple-layers combination makes the system proposed more secure and authentic as appropriate for today's online applications.

Key words:
Access control; authentication; captcha; cryptography; encryption; hashing

1. Introduction

Multiple authentication schemes for acceptable secure access control is essential for most of the applications used today [1]. Researchers studied this authentication problem making-up the need of involving encrypted hashed captcha password as main discussion validation feature for security and human testing used in this study. The phrase captcha (Completely Automated Public Turing test to tell Computers and Humans Apart) is commonly used as a method for anti-robot testing. Normal access control authentication involves modest complexity for user data to be made safe, preventing attackers whom cannot easily make penetrations [2]. Hashing helps create systems security more precise as database owners cannot fully understand authentication functions applicability, making it securely hard to regenerate user passwords [3]. Systems using hashing for encryption normally compress inputs with random length to generate a hash of fixed size [9].

Hash functions are one-way functions reused to generate fixed-length output of sensitive data, which acts as shortened form of the original data. Hash functions can also be used with cryptography to encode and secure data [13]. It is simple to generate and match hash values, but difficult to fake hash values to hide malicious data [9]. In fact, the principle behind good privacy algorithm of data validation can be via hashing, as hash functions do table lookup to find items in the database, detect duplicate, and check sensitivity matches within the hash database. The used hash one-way functions are not invertible, i.e. it is not possible to reconstruct the input sensitive data from the hash, making it the known efficient tool for access control verification.

Online platforms today need to be checking whether the user is human or not to avoid brute force and flooding attacks as most common vulnerability due to availability of enhanced computerized power and network speed. In fact, programmed testing of all possibilities (brute force attack) as well as automatic consuming all available server resources (flooding attack), continually send accessing request causing denial-of-service (DDoS) eruption, making the most simple but overwhelming server PC machine outbreak [7]. Normally, passwords are not highly protected, since they can be directly accessed within databases [6]. This proposed system is to help solve these problems of preventing multiple accessing attacks via efficient integration of captcha crypto hash functions involve three parts, captchas (to check if the user is human), hashing (to protect the user password), and the login crypto part (to validate original privacy of user accounts). The main objective of the proposed system is to provide efficient security for online users to ensure their authentication to use accounts on various internet platforms. The remaining sections of this paper are as follows. Section 2 briefs the literature review and related work of this research. Section 3 discusses the theoretical background and component functions studied to build-up our system. Section 4 presents the proposed approach introducing the integration of captcha, encryption and hashing techniques. We show the step by step working of our proposed system to provide the requested outcomes. The section further provides the proposed work structures discussing its security developments highlighting our work efficient contribution remarks. Section 5 details the comparison and analysis of the proposed system with its features and vs. others, followed by the conclusion ending this work.

2. Literature Review

Manuscript received September 5, 2020
Manuscript revised September 20, 2020

DOI: 10.22937/IJCSNS.2020.20.09.3

In literature, there are different techniques found to play vital role in terms of access control security. These techniques include different combinations of layering to provide sufficient security needed for the systems applicability [9]. Some of those techniques are well developed and some lacks major protocol to provide the exact amount of security. Several different related techniques have been studied to build-up our proposed technique. For example, Shuster in [24] presented cognitive-based captcha system assuming sufficiency by only applying captcha, i.e. to provide security, not involving any other layer of security. This Shuster cognitive security approach provided testing showing effectiveness in stopping some attacks. However, it cannot be generalized, as considered to be unpractical for sensitive larger systems. On the other hand, Zhu in [28] proposed a technique of captcha as graphical passwords based on sophisticated AI problems. This technique combined captcha and RSA encryption techniques for the data security. However, there work had high overhead implication due to using normal (un-hashed) data overwilling the network bandwidth [3]. Furthermore, Althamary [6], proposed a captcha-based authentication in cloud environment using salted hashing without encryption. The technique was found very useful for encountering phishing attack and dictionary attack, as powerful technique for applications used to handle simple security issues, not recommended for many applications due to possible security breach from network spoofing [7].

Similarly, Zhu in [30] proposed a graphical captcha security technique pretending strong security authentication scheme without involving password databases. This Zhu technique used 2-layer security system margining captcha with public key encryption as applicable protections method. However, it is found to be suffering from computation complexity (due to asymmetric cryptography) and vulnerability of robot security breaches (due to DDoS and brute force attacks). Likewise, Luo in [16], recently introduced an authentication process by encrypted negative password as to provide security based on mixing encryption and hashing. The technique used SHA256 hashing combined with AES encryption to secure the system against different attacks but found its clear weakness of robot security breaches allowing DDoS and brute force attacks. Correspondingly, Sediyono in [25] proposed a strong and comprehensive technique to secure login by using one-time password authentication based on MD5 Hash encrypted SMS based on 2-layer of security. The technique holds strong encryption algorithm and MD5 hashing, whereas it lacks captcha for anti-robot attacks protection. Sanjeevi in [21] proposed a captcha mechanism based on improved, named DROP, security using hard AI complexity designed for accessing cloud services. Although this work solved the human authentication (antirobot) need but found suffering from its speed transmission and network delay. Its method used AES encryption as acceptable data protection but lacks applicability (due to missing hashing) making it vulnerable to DDoS attacks. Also, Shen in [23] introduced a novel security technique for online password management with sensor-based authentication adopting 3-layered security, also motivating our work. Shen work used SHA1 algorithm, applied for hashing mechanism along with captcha as well as general encryption strategy for encryption, making-up some parts of our proposal, seeking improvement possibilities for efficiency. Todorov [27] presented captcha securing passwords method, based on MD5 hash dedicated for online web applications. The work intended to provide protection against brute force attack and dictionary attack. This Todorov security technique used 3-layers of captcha, hashing and public key encryption making it slow. Similarly, Zhai in [29] presented an improved password-based identity authentication system which included 3-layers security. Zhai used captcha, hashing and encryption igniting us in our work but seeking applicability efficiency. Zhai security protocol provided privacy adopting Diffie-Hellman encryption joint with MD5 hashing in an interesting manner. The security protocol is suggested to be applied for larger complex systems. However, its Diffie-Hellman encryption algorithm is considered weak as many new and updated crypto algorithm have been recommended to replace it. Therefore, this Zhai technique is used as another seed for our improved integration proposal of captcha crypto hash functions improved to be in efficient manner.

3. Brief Theoretical Background

Security has become a major issue due to increased amount of IT infrastructures involving sensitive data [2]. The threat of unauthorized access to any system is one of the biggest security breaches [3], because it can exploit the system to the deepest levels as a flaw that every hacker intends to make [10]. Along with the confidentiality and integrity, there is another trait of data which is authentication, where unauthenticated access is restricted. Through this restriction, sensitive information can be protected via many smart strategies. The research in this is still going on, but every research is based on the conventional methods of login using user name and password. This method possesses many vulnerabilities and have gone through number of attacks, which leads to research and develop new strategies to secure the data. This technique was strengthened by one-time password (OTP), multi-factor authentication (MFA), salted password hashing and single sign on (SSO) [6]. But hackers still adopted many ways to get into user accounts and therefore stronger actions were required [2]. Most of the current available methods are making use of usernames and passwords without incorporating captcha signs in the test [19] as clear weakness of robot security breaches allowing DDoS and brute force attacks [4].

As Internet of Things (IoT) apps are increasingly becoming omnipresent security methods are to be improved. Attackers are finding new ways to conduct and launch different outbreaks. First vulnerability of any IoT system starts from the commencement screen where we have to start access the system, known as login process. Most vulnerabilities come via the page of requesting user name and password. Therefore, this single layer security is to be improved via 2-layer and 3-layer security. Attackers can simply use different types of combinations for password cracking using different types of bots which eventually allow them to access the system, known as brute force attack. Simple password without encryption can be guessed or if the system is compromised can be obtained through the database which is another issue regarding the system. Also, the attacker can try accessing the system via distributed parallel attempts causing the system to stop, known as DDoS attack. Therefore, our proposal of anti-robot (human) verification is needed as a request for captcha vital role [6]. Some researches elaborated on the attacks of preventing multiple accessing via integration of captcha crypto hash functions, such as password-based attacks, dictionary attacks and guessing attacks [12]. These attacks compromise the confidentiality, integrity and availability of the system needing proper accessing authentication [3]. If accountability measures are not taken correctly, the accessing attacks can make very strong damage to the system, which may not be easily recoverable [2]. The three components building our captcha crypto hash authentication security is presented next.

3.1 Captcha

Captcha, is defined as Completely Automated Public Turing test to tell Computer and Human Apart, is widely adopted as a helping tool to stop the unauthorized robot access to different online resources. It is used as protection from the internet bots to different web services and web applications [8]. It is basically a cryptographic code which automatically generate test that cannot be solved by the computers or bots, but mostly human can solve easily. For the captcha, there is a default assumption that all the users are the bots, except those who can solve it. The tests are designed to get to know about the audio or visual senses through different ambiguities. The services of captcha are plausible for different issues related to security including denial of service attack, dictionary attacks and spam [17]. Captcha has been proven very useful to protect against most robot different web attacks. It provides applicable protection starting from the registration, as helping the internet in its current relation to monitory transactions and sale purchases, considered very important tool. Captcha can be found in different types which include text, audio, puzzle and graphic. Text based captcha is very simple when it is required to be implemented as it asks very simple questions. The answers to those questions are simple but need to be entered physically into the system. In the graphic based captcha, user have to view the images and match them with others based on specific visual requirements. In audio-based captcha, user have to enter the words, or characters, by listening to them [17]. In puzzle-based captcha, an image is given in chunks which user needs to join to complete all the parts of the puzzle. Every captcha type has some drawback and some advantages. For example, text-based captcha has commonly been used due to its ease of use and accommodation of low network speeds [22]. Image based captcha has been identified more easily to satisfy user culture. It can be simpler to resolve then the text and audio captcha, but requires good network as well as user clear visual observation that cannot be preferred by blurred vision people. Some can find it difficult preferring audio-based captcha, which is more specific to culture and language, as a further challenge to be considered, as the English audio is difficult for many users to understand [17]. Hence based on simplified nature of text-based captcha it is comparatively and adoptable choice.

Captcha alone cannot help to secure any system properly, as it does not protect the data but provide the system some protection. In fact, captcha can annoy the human user because it becomes hectic again and again. The biggest disadvantage of adopting captcha is it cannot stop the spammers, which include human as well. There have been many techniques that involves captcha deceiving for an attack through bot and human [15]. Therefore, secure authentication method must be followed by it. Simple user name and password can be helpful but its security becomes powerful involving it [22]. Therefore, captcha combined with user name and password along with encryption and hashing is very significant.

3.2 Encryption

With the increased popularity of the internet the web has become a fertile field for those who wants to attack and abuse it for different purposes. Simple user name and password technique cannot hold the attackers back even applied with the captcha [20]. As the password is saved into the database, once the system is accessed, all the users account are at risk of hackers. Sophisticated breaches can exploit all accounts or even publish the sensitive data online, which increases the range of real threats. Therefore, encrypted password is the approach to handle this problem, that even if a hacker has reached the database, then he cannot get access to the passwords and details of all other accounts. Encryption is proven essential to strengthen the system security [9]. There are different encryption techniques for the password access protection. The three main popular methods found are based on block cipher symmetric key approach, which include AES, DES and 3DES. DES is one of the most widely accepted standards containing 64-bit long input key. Although it is widely accepted, but there have been many attacks reported occurred on the DES making it untrusted block cipher symmetric key encryption method [2]. 3DES has extended the key size of the DES with combined key size of 168-bits. It included 3-times iterations, which increase the level of security on the price of complexity and speed. Therefore, 3DES is the slowest block cipher symmetric key encryption method to be considered. On the other hand, AES has replaced DES in 2001 and supports the key length of 128-bits. AES is claimed secure as it goes for ten rounds during the process of encryption, pretended as most secure algorithm compared to DES and 3DES [20]. Therefore, a combination of captcha along with AES encryption is the recommended secure combination to provide our intended security. This two-layer security has been used by different systems, but as the ratio and techniques of attacks are also enhancing fast, our improved security layer is suggested to involve hashing, similar in principle to the authentication research in [1].

3.3 Hashing

Hashing is the mechanism which involves converting an input of any length into a fixed size string of text, using a mathematical function. The data over web deals with important and significant information and carries all types of sensitive information, therefore a comprehensive amount of security plan is essentially needed [23]. If any unauthorized means have accessed the business or sensitive data, then the result is privacy loss. If the data is encrypted and has assisted by hashing technique, then it becomes much tough for hackers to access such data. Hash compresses any message to a fixed length message with condition of every hash chunk to be different from the other. There are different types of hashing algorithms. The most common hashing coding include SHA256, MD5 and SHA1, which are considered the most secured hash algorithm proposed for accessing authentication purposes [1]. The hash abbreviation MD5 stands for Message Digest and SHA1 stands for Secure Hash Algorithm, where SHA1 is selected as reported to be the secure fast algorithm recommended to be used [16]. SHA1 is fastest hashing function with ~587.9 ms per 1M operations for short strings, and 881.7 ms per 1M for longer strings. MD5 is 7.6% slower than SHA1 for short strings and 1.3% for longer strings [23]. The construction difference behind these hashing algorithms is within square measures accustomed as to generate novel digital fingerprint of knowledge or message, i.e. that is understood as a hash or digest [14]. In fact, SHA1 hashing and AES encryption techniques are the mostly recommended standard systems [16], but not coordinated with the anti-robot captcha. In fact, SHA1 and AES are the most preferred practical techniques recommended for practical and low RAM requirements providing high computing speeds. Therefore, efficient integration of text-captcha, AES, and SHA1 is adopted to propose our security systems making the access authentication process smooth for users.

4. Proposed System

The proposed system integrates captcha, SHA1 hashing, and AES encryption as the login access authentication scheme benefiting from the pros of different accessing control systems. The added features will make this system secure and safe for most applications, to use from any place, even if it's not on personal machines [16]. The main purpose of this proposed system is to have user data and information secured pretending that online user accounts are fully protected and validated before login. The proposed security system involves 3-layers utilizing captcha, encryption and hashing, as framework shown in Fig 1.

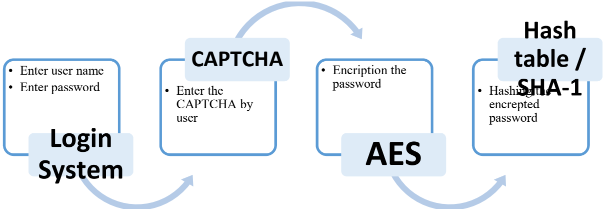
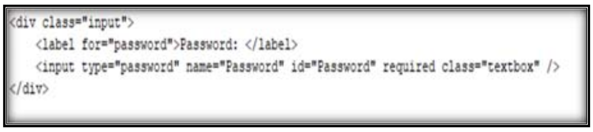


Fig 1 proposed security system framework

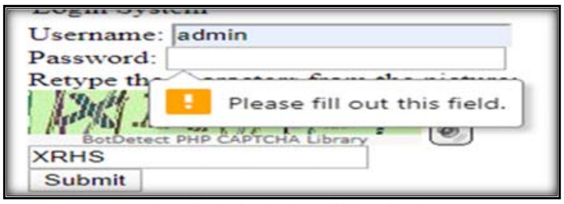
As introduced earlier, the different types of captchas used can be based on math function, word problem or text organized, which are used for the security purposes. Our proposed system chooses text captcha as recommended efficient tool used in different services over the internet, i.e. text captcha is very simple as reported solvability common for most humans handling [14]. In our proposed login system, captcha considered two properties meant to avoid understandability as well as observability clearness. The system allows to change the captcha as user requests refreshing. It also provides options to listen to the captcha words, as providing proper accessing help. Hence that the main purpose of captcha is anti-robot checking and not privacy complexity nor confidentiality. The required property in fields has been used by the code part, as shown in Fig 2.



```
<div class="input">
    <label for="password">Password: </label>
    <input type="password" name="Password" id="Password" required class="textbox" />
</div>
```

Fig 2 captcha code part within the proposed system

The required HTML attribute is Boolean function that is used to specify the input element must be filled before submitting the form. It will help enforcing accessing proper order, as shown in the Fig 3.



Username: admin
Password:
Retype th
Please fill out this field.
XRHS

Fig 3 login basic data filling field

The second layer of our system security is adopting AES (advanced encryption standard) encryption. The AES is fast and safe encryption form that preserves user data carefully [11]. The crypto AES is officially the replacement standby of old DES due to its small key size and slow speed [12]. The AES encryption is mathematically more efficient technique in comparison to DES and the bits of AES makes it exceptionally stronger than the DES. Therefore, we proposed using AES in our three-layered security as working perfect for 128-bits symmetric key block cipher. It is software implementable in most languages showing C and java full open source libraries description details. It is iterative as preferred over other Feistel ciphers [11]. The design of AES contains a series of linked processes, including substituting inputs by exact outputs involving lumbering bits around. All the operations of AES are practically performed on bytes rather than bits [12]. Therefore, AES makes the 128-bits of plaintext blocks as 16-bytes organized in four columns and four rows represented as a matrix [9]. The AES can be scalable, if more bits are to be involved, seeking more security on price of number of operation rounds. It uses 10 rounds to operate 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys, as shown in Figure 4.

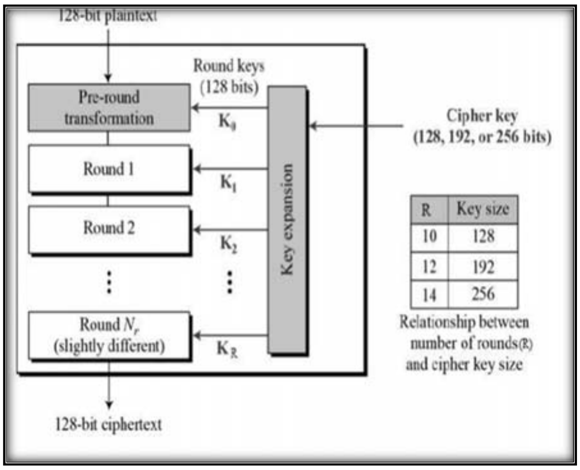


Fig 4 AES encryption rounds

Hashing is third layer of efficient security in our proposed system adopting SHA1 algorithm appropriately. The SHA1 cryptographic hash function converts any input into 160-bit hash value, termed as 40 digits long message. SHA1 gives its message-digest depending on the values differences similar in principle to those used within MD2, MD4, and MD5, but for larger number of bits, i.e. 160-bits justifying our selection of SHA1 [18]. SHA1 is commonly used as part of the numerous extensively used security applications and protocols, such as TLS and SSL. Hence, SHA1 is not appropriate for applications that is dominantly needing collision resistance. However, it is taking its publicity as mandatory by the law for use in specific requests for the safety of accurate unclassified data [14]. Furthermore, the SHA1 hash function have been proven secure for Shacal block ciphers as practically secure basic operation to convert any given value to standard 160-bits [26]. The process of SHA1 starts by appending the bits 0s and 1s as detailed in [14] before the message digest buffer to hold the intermediate and final message to access the hashing rounds, as operation shown in Figure 5. All rounds are based on individual primitive function with depiction of all the stages.

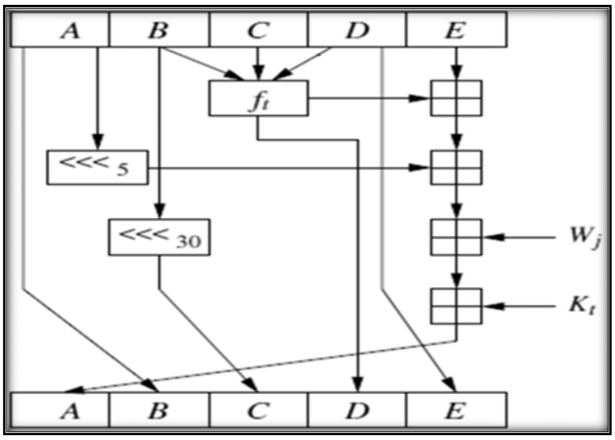


Fig 5 SHA1

Hash tables are data erections that contrivance an associative array abstract data type, a structure that can plot keys to values, as phonebook hash table shown in Figure 6. Hash tables use a hash function to calculate an index into an array of slots from which the necessary amount can be made. Hash functions will allocate each key to a perfect bucket (Figure 6). However, most hash tables are made to have a lacking hash function. It may cause hash collisions, where the hash function creates the same index for more than one key [1]. The collisions have to be put up in some way. A well-designed hash table and the regular cost of the number of commands for each lookup is independent of the number of elements kept in the table. Most hash tables plan allows arbitrary inset and removal of key-value pairs at a constant average cost per operation. In most cases, hash tables are more effective than search trees or other table lookup structures. It is for this reason, hash tables are extensively used in many types of computer software, i.e. particularly for associative arrays, database indexing, caches, and sets [26].

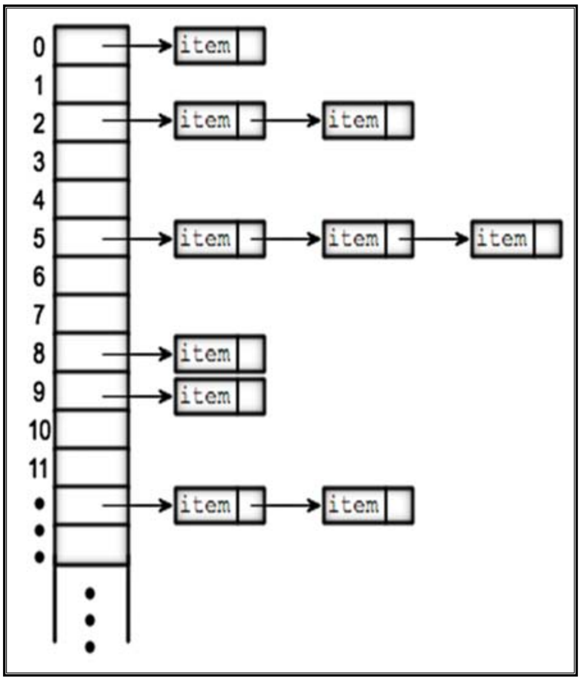


Fig 6 phonebook hash table example

The proposed system has been tested. It helps online users to keep them safe from harmful activities over the internet. In fact, integration of our three layers system ensures that it is not easy for attackers to access user's accounts without their anti-robot (human) consent assuring acceptable efficient data security at any given time. The Algorithm of our system is detailed next. The technique is based on efficient procedure which starts as user opens the starting page to login the system. After adding login details, the system requests user to enter the captcha as text, can be observed or listened. The captcha is to be written as simple text to verify reality of human interaction, i.e. to avoid robot or bot trailing. Then, the system encrypts the password entered by the user as well as calculating the hash of the password to verify it with already saved database hash list. Figure 7 shows the algorithm steps in a programmable friendly flowchart.

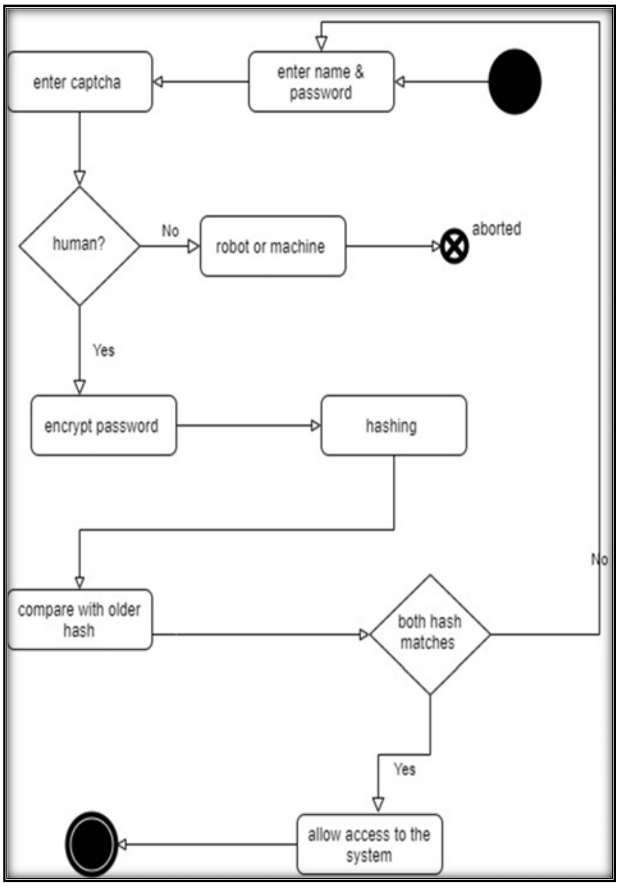
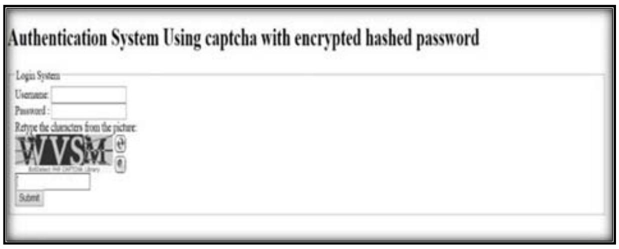


Fig 7 proposed system algorithm flow diagram

As Figure 8, our proposed authentication is asking for username, passwords, and captcha. The system waits for captcha confirmation to apply AES and hash function to the password. The system is tested as implemented using PHP, SHA1 function, and bot detect lib similar in software principles of the work presented in [12]. For example, the user enters the login inputs aiming access, as shown in Figure 9. The results of hashing passwords can be seen (for testing purposes), as the results of using the SHA1 function, as shown in Figure 10. If any mistakes or illegal access attempts are made during the login process, the system will not be accessed, showing Figure 11.



Authentication System Using captcha with encrypted hashed password

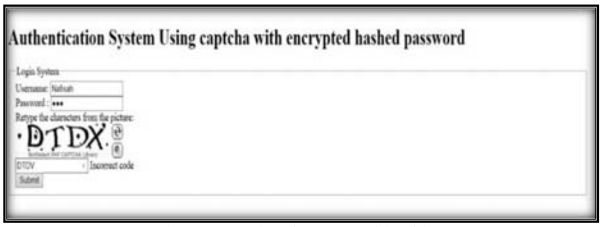
Login System

Username:

Password :

Retype the characters from the picture:

Fig 8 proposed system login process



Authentication System Using captcha with encrypted hashed password

Login System

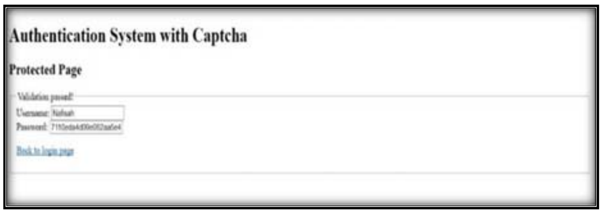
Username: Nafisah

Password : •••

Retype the characters from the picture:

DTDV Incorrect code

Fig 9 example of password hashing



Authentication System with Captcha

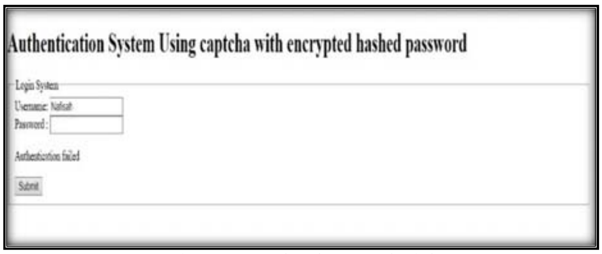
Protected Page

Validation passed!

Username: Nafisah

Password: 7150eda4d00e052aa6e4

Fig 10 example of password hashing depiction



Authentication System Using captcha with encrypted hashed password

Login System

Username: Nafisah

Password:

Authentication failed

Fig 11 example of access denial

Interestingly, our system process coordinates its integration authentication algorithm in sequence to avoid brute force, flooding DDoS attacks. This organization strengthens the system and allow the process to work in a smoother way, i.e. by avoiding unauthorized access to the system.

4.1 Practicality Security

The proposed work has improved login security features as efficient integration of captcha, encryption and hashing functions to secure the system from hackers. Recall from our review that most of the available systems apply encryption without captcha or some apply captcha without encryption. Other systems include captcha and encryption together without hashing causing delay in processing, igniting our proposal of efficient combination to increase level of security to the maximum preserving efficient applicability. In the following subsections, we analyze our proposal to show its optimization features to be remarked. The proposed system is based on different attributes that formulate the complete improved security technique. In web security logging in (or logging on, signing in, or signing on) is the process in which an individual gains access to a computer system by identifying and authenticating himself. The logging system is based on our three inputs, as example shown in Figure 9. As proof, we will later compare this implementation with online websites presented in [10]. When entering the password, the system encrypts it in "alpha-numeric" format, which will be saved into the database privately, i.e. even database admin won't be able to see the actual values.

Figure 12 shows hashed password process of verification into the different steps.

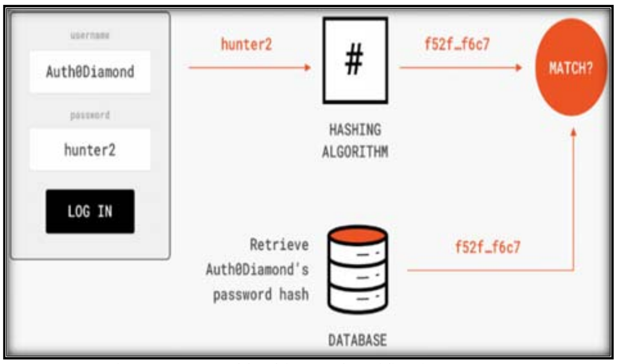


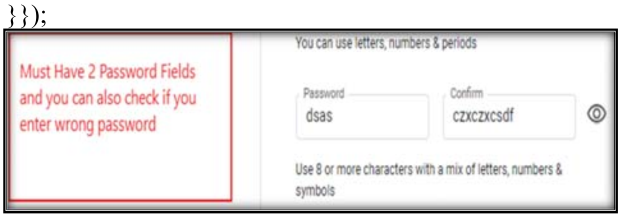
Fig 12 password hashing and verification process

Our proposed system has involved eye-sign login feature to facilitate the user to enter the password correctly; these features consists of an eye button and caps letter button options, which allow the user to view typing right password [13]. This feature will facilitate the user to see the accuracy of the typed password and allow him not to add wrong password, which will help to avoid blocking of the account due to several wrong password attempts, as shown in Fig 13. Our system also considered caps-lock on/off feature, as shown in Fig 14. This caps-lock notification is very helpful for typing the correct password too, which is available in the compared works as built-in features. The code of the caps-lock on/off feature algorithm within our system is shown below:

```
var input = document.getElementById("myInput");
// Get the warning text
var text = document.getElementById("text");

// When the user presses any key on the keyboard, run the
function
input.addEventListener("keyup", function(event) {

  // If "caps lock" is pressed, display the warning text
  if (event.getModifierState("CapsLock")) {
    text.style.display = "block";
  } else {
    text.style.display = "none"
  }});
```


Must Have 2 Password Fields and you can also check if you enter wrong password

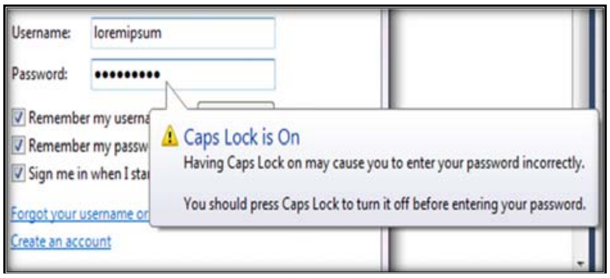
You can use letters, numbers & periods

Password: dsas

Confirm: czxczxcsdf

Use 8 or more characters with a mix of letters, numbers & symbols

Fig 13 eye-sign feature in our proposed systems


Username: loremipsum

Password: •••••••••

☑ Remember my userna

☑ Remember my passw

☑ Sign me in when I sta

Forgot your username or

Create an account

Caps Lock is On

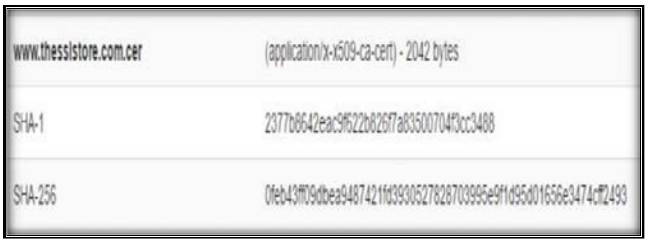
Having Caps Lock on may cause you to enter your password incorrectly.

You should press Caps Lock to turn it off before entering your password.

Fig 14 caps-lock notification

4.2 Hashing Applicability

Although our proposal system testing used SHA1, it can involve any improved hashing algorithm, as shown in Figure 15. In fact, scientifically SHA1 has been improved to SHA2 (224, 256, 384 and 512 bits), and more recently to SHA3. However, SHA1 is selected for this research scope as unilateral function that decrypt the plaintext behind hash as confront it to an online database. This research website allows comparison for research purposes of SHA1 hashes to be decrypted as needed in our study. The online database contains 15, 183, 605, 161 words, coming from all the wordlists, i.e. found online suitable for testing. The database is explored uniquely for SHA1 decryption to fulfill our system exploration but can applied in any updated work. The reader is referred to reference [18] for more elaboration on SHA improvements.



www.thesslstore.com.cer	(application/x-x509-ca-cert) - 2042 bytes
SHA-1	2377b8642eac9f622b826f7a83500704f3cc3488
SHA-256	0feb43ff09dbea9487421fd3930527828703995e9f1d95d01656e3474cff2493

Fig 15 SHA improvement

In our system, SHA1 produces a 160-bit hash from any input data. Given that there is an infinite number of messages that hash to each possible value, there is an infinite number of possible collisions [18]. But because the number of possible hashes is so large, the odds of finding one by chance is negligibly small (one in 280, to be exact) [5]. If you hashed 280 random messages, you'd find one pair that hashed to the same value. That's the "brute force" way of finding collisions, and it depends solely on the length of the hash value. Therefore, "Breaking" the hash function means being able to find collisions faster than that and that's what is reported by Chinese attempts [18]. They can find collisions in SHA1 in 2^{69} calculations, about 2,000 times faster than brute force. Right now, that is just on the far edge of feasibility with current technology. The pseudocode expressing our hashing work can be expressed as below:

Append the bit '1' to the message i.e. by adding 0x80 if characters are 8 bits.

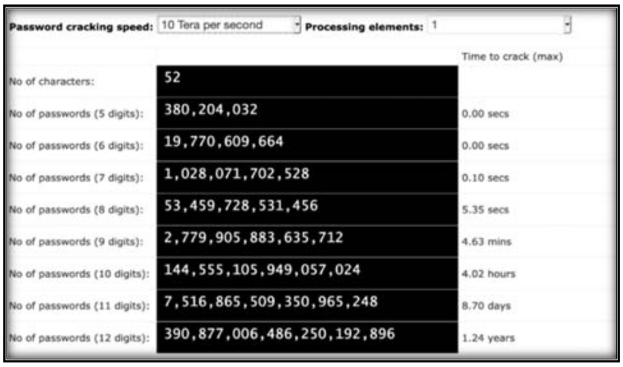
```
Append 0 ≤ k < 512 bits '0', thus the resulting message length (in bits)
Is congruent to 448 (mod 512)
Append ml, in a 64-bit big-endian integer. So now the message length is a multiple of 512 bits.
Process the message in successive 512-bit chunks:
break the message into 512-bit chunks
for each chunk
break the chunk into sixteen 32-bit big-endian words w[i], 0 ≤ i ≤ 15
Extend the sixteen 32-bit words into eighty 32-bit words:
  for i from 16 to 79
      w[i] = (w[i-3] xor w[i-8] xor w[i-14] xor w[i-16]) leftrotate 1
```

4.3 Performance Delay

Our system is considered acceptably efficient as running faster than others although adopting complex structure of programming, as performance listed in Table 1. For example, SHA1 hashing give us extra fast speed as compare to other systems using direct links of the online encrypted algorithm that take a lot of loading time, as describe in details in [13]. In fact, to generate the final output, the SHA1 core block occupies 80 clock cycles, as better than all others value as variable created considering critical path including the delay of additions and non-linear functions. To improve this delay, to enhance the hashing function speeding up the system, the comprehensive performance of addition operation needs reconsideration, as known critical path influence maker for delay within the SHA1 algorithm [18]. Through the numerical analysis of brute force attack to break SHA1 (Figure 16), it can be observed faster than old techniques as holding secure functionality.

Table 1 SHA1 performance

Throughput	824Mbps	1456Mbps
Efficiency	0.83%	2.83%
Maximum Frequency	103Mhz	183Mhz



Password cracking speed: 10 Tera per second Processing elements: 1

		Time to crack (max)
No of characters:	52	
No of passwords (5 digits):	380,204,032	0.00 secs
No of passwords (6 digits):	19,770,609,664	0.00 secs
No of passwords (7 digits):	1,028,071,702,528	0.10 secs
No of passwords (8 digits):	53,459,728,531,456	5.35 secs
No of passwords (9 digits):	2,779,905,883,635,712	4.63 mins
No of passwords (10 digits):	144,555,105,949,057,024	4.02 hours
No of passwords (11 digits):	7,516,865,509,350,965,248	8.70 days
No of passwords (12 digits):	390,877,006,486,250,192,896	1.24 years

Fig 16 brute force numerical analysis

5. Comparisons and Analysis

The difference between the proposed system and most of the used systems is mainly the efficient integration security functions as main control access applicability measure [15]. The proposed system uses hash functions to secure user data from attackers as hashing is not used in many applications, which makes them vulnerable to confidentiality attacks. In fact, some applications involved hashing but found without captcha function as vulnerable to brute force or DDoS attacks, as discussed earlier. This efficiency and integration need enforced our collective coordination of both to benefit best protection situations against different attacks. This section is going through comparison and analysis of our system proofing its principle effectiveness compared to others. We start by applicability comparison of our proposed development to the basic common access control systems.

5.1 Applicability Comparison with Commonly used Systems

The basic difference between the proposed system and other systems that are already playing their contribution in the field of access control security is that it offers multi-level authentication against many attacks. The proposed system encrypts the database tables among all. Even database administrator won’t be able to view any passwords as well as all users’ details. Hence if, in any case, the data of the IT control security database is compromised, our proposal system allows proper recovery, as to be able to control security retrieval safely. The research found that all authentication methods without captcha are not properly securely useful. As any robot can get into the system and create account flooding its database with false information, as shown in Figure 17. This low level of security encourages multiple attempts that can be easily made through the bot simple coding. This bot flooding issue can be fully stopped via our captcha involvement in a practical manner. Figure 18 briefs the involvement of our proposed system with already working systems in this access control authentication domain. This is the clear depiction that more layers of security provide great resistance, as less layers of security will lower the resistance of the system against attacks.

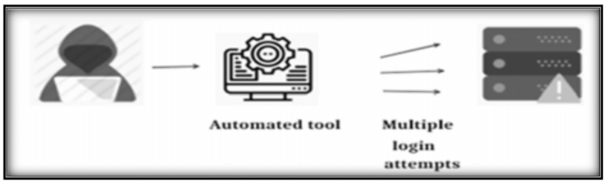


Fig 17 multiple login registration attempts

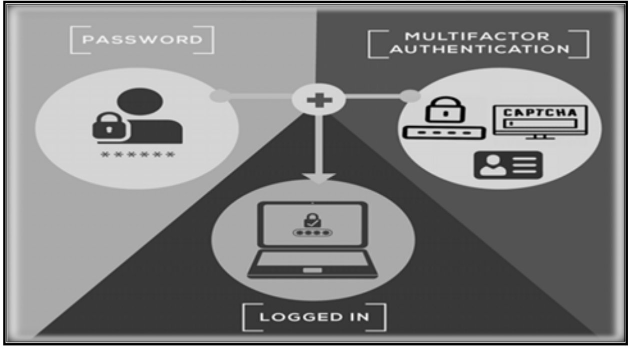
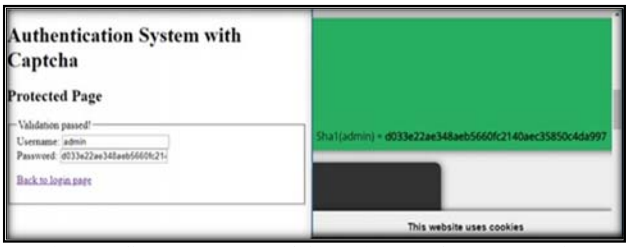


Fig 18 comparison between proposed and basic systems

When comparison is performed between different passwords decryption procedures and ours, within our proposed system, it is that it cannot show hashing clearly to the user, i.e. as it saves it to the database, for us to avoid users from any type of complexity inconvenience. A representation of both proposed and already present standard system is shown in Figure 19. Note that most system depend on cookies to make the system accessible but providing full ambiguity removing the user observation from the real control.



Authentication System with Captcha

Protected Page

Validation passed!

Username: admin

Password: d033e22ae348aeb5660fc21

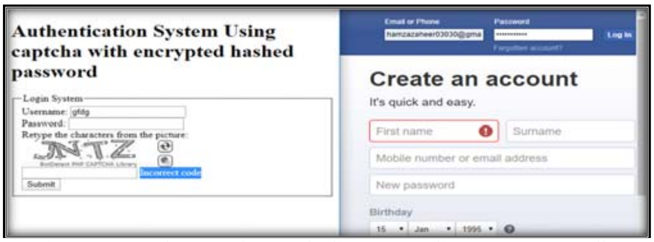
Back to login page

Sha1(admin) = d033e22ae348aeb5660fc2140aec35850c4da997

This website uses cookies

Fig 19 Online SHA1 password decryption vs. other systems using ambiguous cookies

It is believed that authentication is less secure as if no anti-robot is placed in first security checking step. As in Figure 20, our system is observed vs. vs. most online messengers' toll website approached missing anti-robot verification. It has been tested to access the site many times pretending millions of people holding personal and professional information about them via running a bot code. Therefore, not having captcha is an alarming situation, where user's database can be easily damaged through different spam attacks as well as DDoS problem that occurred hanging the platform.



Authentication System Using captcha with encrypted hashed password

Login System

Username: gfdg

Password:

Retype the characters from the picture:

Incorrect code

Submit

Email or Phone: hamzazaheer03030@gma

Password: ••••••••••

Create an account

It's quick and easy.

First name

Surname

Mobile number or email address

New password

Birthday: 15 Jan 1995

Fig 20 proposed approach stops fooling most online messengers' toll application

5.2 Technical Comparison to Others

This work has been compared to different authentication systems claimed to keep the security to its maximum level. The comparison study involved ten different login authentication reported techniques used within years from 2010 to 2020, as found detailed in a research manner. The comparison works are collected such that all ten are featuring the same relative attributes providing same intention of defense against authentication different attacks, as mostly briefed earlier in the literature review. This evaluation elaboration considered exploring features of the approaches commenting on passwords cryptography, hashing adoptability, anti-robot captcha usages and login encryption technique utilization, as well as integration complexity of the system (as security levels) and vulnerabilities attacks prevention. Our study comparison notes are presented in Table 2.

Table 2 comparison of the proposed system and other systems

Techniques	Password Encryption	Hashing technique is applied	Captcha is used	Encryption technique for data	Security integration complexity	Attack Prevention
Captcha as Graphical Passwords—A New Security Primitive Based on Hard AI Problems [28]	No	No	Yes	RSA	0	Relay attack Online guessing attack
A More Secure Scheme for CAPTCHA-Based Authentication in Cloud Environment [6]	No	Salted hashing	Yes	None	1	Phishing attack Dictionary attack Guessing password attack
The improved DROP security based on hard AI problem in cloud [21]	No	No	Yes	AES algorithm	0	Password guessing attack
Design and implementation of password-based identity authentication system [29]	No	MD5	Yes	Diffie-Hellman	2	Dictionary attack
Usable security of online password management with sensor-based authentication [23]	No	SHA1	Yes	General encryption	2	Offline dictionary attack
Cognitive-based CAPTCHA system [24]	No	_	Yes	_	0	Dictionary attack
Enhanced Graphical Captcha Framework and Applications to Strong Security Authenticated Scheme without Password Table [30]	No	No	Yes	Public key encryption	0	DOS attack Impersonating attack
Authentication by Encrypted Negative Password [16]	Yes	SHA256	No	AES	1	lookup table attack rainbow table attack
Secure login by using One-time Password authentication based on MD5 Hash encrypted SMS [25]	Yes	MD5	No	Public key encryption	1	Password attack
Securing passwords with CAPTCHA based hash when used over the web [27]	Yes	MD5	Yes	Public key encryption	2	Brute force attack Dictionary attack

Improved the Security Using Captcha with Encrypted Hashed Password to Access System [proposed paper]	**Yes**	**SHA1**	**Yes**	**AES**	**3**	**Relay attack** **Online guessing attack** **Brute force attack** **Dictionary attack** **DOS attack** **Password attack**

As Table 2, our proposed system is benefitting from all others using the best efficient strategies to counter most multiple types of attacks. Notes that most systems compared used hashing method of either SHA1 or MD5 in such practical evaluation cases assuming SHA1 is comparatively efficient to be tested as proofing the efficiency functionality. Also, the proposed strategy implements a very comprehensive encryption technique, i.e. AES cryptography, which gives an effective smart accessible edge to the authentication process.

In the comparison Table 2, all techniques are evaluated for protection integration complexity as marked between low level valued 0, moderate level as 1, standard level as 2 and high level as 3. The levels of the security complexity depend upon the number of independent security scheme algorithms involved. e.g. the technique with only captcha is not much safety complex as it is just checking for anti-robot security, therefore it is termed as low level security approach, i.e. as can be found for works known as captcha as graphical passwords-a new security primitive based on hard AI problems [28], improved DROP security based on hard AI problem in cloud [21], cognitive-based captcha system [24], and enhanced graphical captcha framework and applications to strong security authenticated scheme without password table [30]. The techniques with captcha and encryption of data is marked as moderate level 1 security complex schemes such as the works entitled as more secure scheme for captcha-based authentication in cloud environment [6], authentication by encrypted negative password [16] and secure login by using one-time password authentication based on MD5 hash encrypted SMS [25]. A more higher security level of integration complexity techniques mixing captcha, encryption and password hashing within basic algorithm as considered standard level 2, such as design and implementation of password-based identity authentication system [29], usable security of online password management with sensor-based authentication [23], and securing passwords with captcha based hash when used over the web [27]. Our proposed system is considered the best secure complex integration efficient system as having level 3 indication (Table 2) of security, i.e. integrating captcha crypto hashing well-organized authentication algorithm. Our complex level-3 techniques is found applicable to protect most authentication expected breaches, i.e. protection against relay attack, online guessing attack, brute force attack, dictionary attack, DDoS attack and password attack. The focused comparison of security complexities of all techniques has been demonstrated through the graph shown in Fig 21.

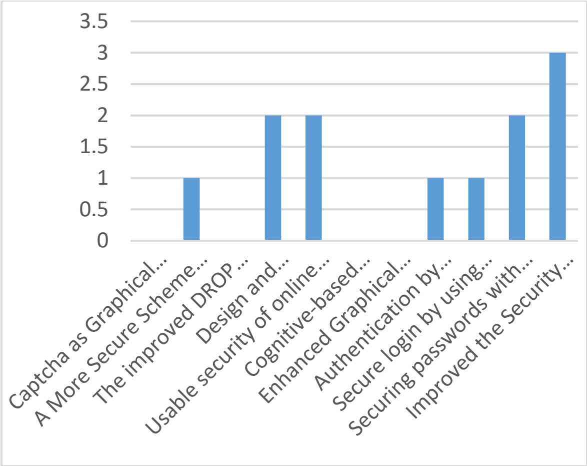


Fig 21 comparison of security complexities graph

Our technical comparison investigation covered the speed of running the hashing verification. It demonstrated that our proposed application is efficient considering the complete system and optimize the security plan based on speed, performance, security and availability, i.e. in comparison to other techniques as shown in Table 3.

Table 3 hash comparison table of proposed and other techniques

Techniques	Speed	Performance	Size
Captcha as Graphical Passwords—A New Security Primitive Based on Hard AI Problems [28]	N/A	N/A	N/A
A More Secure Scheme for CAPTCHA-Based Authentication in Cloud Environment [6]	N/A	N/A	N/A
The improved DROP security based on hard AI problem in cloud [21]	N/A	N/A	N/A
Design and implementation of password-based identity authentication system [29]	881.7 ms	7.6% slower than SHA1	32 chars hash
Usable security of online password management with sensor-based authentication [23]	587.9 ms	15.5% slower than SHA1	40 chars hash
Cognitive-based CAPTCHA system [24]	N/A	N/A	N/A
Enhanced Graphical Captcha Framework and Applications to Strong Security Authenticated Scheme without Password Table [30]	N/A	N/A	N/A

Authentication by Encrypted Negative Password [16]	587.9 ms	15.5% slower than SHA1	64 chars hash
Secure login by using One-time Password authentication based on MD5 Hash encrypted SMS [25]	881.7 ms	7.6% slower than SHA1	32 chars hash
Securing passwords with CAPTCHA based hash when used over the web [27]	881.7 ms	7.6% slower than SHA1	32 chars hash
Improved the Security Using Captcha with Encrypted Hashed Password to Access System [proposed paper]	**587.9 ms**	**Faster than others**	**40 chars hash**

5.4 Discussion Remarks

Analyzing our work compared to other systems, it has been reported that our system have unique applicable characteristics of security and speed. The proposed technique is considered preferred as users can save the password safely, i.e. the system assures the confidentiality of the password protected even from the system administrator. In normal systems, hackers attack the application and its database to retrieve all details, hacked to be used for the wrong purposes. This password database is found clean for reading due to the absence of captcha and encryption techniques [10]. On the other hand, our system has fulfilled this protection by saving the passwords encrypted securely that can only be decrypted tested via large frameworks of PHP like larval code ignitor technology. Furthermore, our usage of captcha verification completes the protection making it winning scheme among other applied systems [5]. Note that robot coding try to login benefitting from the absence of captcha, which is highly secured in our attempt as not allowing the webpage to send empty login information, pretending the visitor is a robot (not human), which is found big saver of the network bandwidth against DDoS problems. Many other systems used crypto accessing algorithms without using captcha verification which is clear weakness solved in our work accepting text-captcha as observed or listened as found convenience. Sound captcha alphabets provide us opportunity to confirm text writing that should be typed as captcha verification text, as captcha can be requested to be refreshed if completely unclear as complete friendly optimized system to provide convenient access.

Our system can be further used by any developer using PHP framework, i.e. he can use our system as authentication adopting the privacy of login passwords adopting encrypted keys [18]. Our new developed system guarantees more safety allowing the user to operate in a more secure manner as tough to penetrate by eavesdroppers. Therefore, it can be trusted as more secure approach better than others because of their low level of security, as clarified earlier via Table 2.

6. Conclusion

Today, there is increased use of computers and computer systems needing efficient security access control schemes. In fact, the advancement of hacking is forcing the need to have more secure systems that have strong authentication and security features. Hackers and attackers come up with new ideas daily, trying to bypass proper accessing systems causing huge amount of data manipulation. This work proposed an efficient authentication system having high-security features to login a system safely. We use hash functions to protect users' passwords. We use captchas to confirm anti-robot situation, i.e. to detect non-human users where all communications between the system and users are encrypted via the secure browser. All these captcha hash crypto control features are integrated to ensure that user’s data are safe and their information is free form possible attacks. Our proposed system is considered the best secure complex integration efficient system to protect most authentication expected breaches, i.e. protection against relay attack, online guessing attack, brute force attack, dictionary attack, DDoS attack and password attack. We have presented and implemented an alternate and secure approach for entering user login details. The method eliminates threats arising, shoulder surfing and cursor tracking, thus securing login procedures on unknown or public computers. The higher the correlation of user-clicked points between different login attempts is, the less effective the protection the dual-view technology would provide to thwart shoulder- surfing attacks.

The work is considered in beginning stage of this authentication process. The research future work can be done for further refinements such as trying different hashing functions. The suggestion is to test the differences of replacing SHA1 by SHA2 (224, 256, 384 and 512 bits), and more advanced to SHA3. Also, the hashing can be improved to adopt MD5 or more advanced techniques seeking for possible benefits in speed, security, and applicability. The research can improve its captcha method involving images of puzzle and graphic or video-based schemes. In fact, new captcha methods can help in providing more convenience to the user that can depend on the personal culture with its aim to verify anti-robot situation. The work can test involving light-weight cryptography replacing AES encryption providing acceptable security in limited computational power situations. Future systems can incorporate more security through the use of machine learning or artificial intelligence hoping to get more improved protection systems.

Acknowledgment

Thanks to Umm Al-Qura University (UQU) for supporting this research. Special thanks to the Computer Engineering Department and the Master graduate studies program for motivating this contribution.

References

[1] Almazrooie, M., Samsudin, A., Gutub, A., Salleh, M.S., Omar, M.A., Hassan, A.A. (2020). Integrity verification for digital Holy Quran verses using cryptographic hash function and compression. Journal of King Saud University - Computer and Information Sciences 32(1):24-34.

[2] Alanazi, N., Khan, E., Gutub, A. (2020). Inclusion of Unicode Standard Seamless Characters to Expand Arabic Text Steganography for Secure Individual Uses. Journal of

King Saud University - Computer and Information Sciences, (in press).

[3] Alotaibi, M., Al-hendi, D., Alroithy, B., AlGhamdi, M., Gutub, A. (2019). Secure Mobile Computing Authentication Utilizing Hash, Cryptography and Steganography Combination. Journal of Information Security and Cybercrimes Research (JISCR) 2(1):9-20.

[4] Alanizy, N., Alanizy, A., Baghoza, N., AlGhamdi, M., Gutub, A. (2018). 3-Layer PC Text Security via Combining Compression, AES Cryptography 2LSB Image Steganography. Journal of Research in Engineering and Applied Sciences (JREAS) 3(4):118-124.

[5] Adhatrao, K., Gaykar, A., Jha, R., & Honrao, V. (2013). *A secure method for signing in using quick response codes with mobile authentication.* arXiv preprint arXiv:1310.4000.

[6] Althamary, I.A., & El-Alfy, E.S.M. (2017, May). *A more secure scheme for CAPTCHA-based authentication in cloud environment.* IEEE 8th International Conference on Information Technology (ICIT) pp. 405-411.

[7] Al-Juaid, N., Gutub, A. (2019). Combining RSA and audio steganography on personal computers for enhancing security", SN Applied Sciences 1:830.

[8] Babu Rajeev Maddipati. (2018). *Implementation of Captcha as Graphical Passwords for Multi Security.*

[9] Chatterjee, S., & Sarkar, P. (2011*). Identity-based encryption*. Springer Science & Business Media.

[10] Dang, Q., & National Institute of Standards and Technology (U.S.). (2009). *Randomized hashing for digital signatures*. Gaithersburg, MD: U.S. Dept. of Commerce, National Institute of Standards and Technology.

[11] Daemen, J., &Rijmen, V. (2011). *The design of Rijndael: AES - the advanced encryption standard*. Berlin: Springer.

[12] Easttom, C. (2016). *Modern cryptography: Applied mathematics for encryption and information security*.

[13] *Fast Software Encryption.* (2010). Berlin: International Association for Cryptologic Research.

[14] Hidalgo, J. M. G., & Alvarez, G. (2011). *Captchas: An artificial intelligence application to web security*. In Advances in Computers 83:109-181

[15] Konheim, A. G. (2010). *Hashing in computer science: Fifty years of slicing and dicing*. John Wiley & Sons.

[16] Luo, W., Hu, Y., Jiang, H., & Wang, J. (2018). *Authentication by encrypted negative password*. IEEE Transactions on Information Forensics and Security, 14(1), 114-128.

[17] Moradi, M., & Keyvanpour, M. (2014). *CAPTCHA and its Alternatives: A Review.* Security and Communication Networks, 8(12), 2135–2156.

[18] Ratna, A. A. P., Purnamasari, P. D., Shaugi, A., & Salman, M. (2013, June*). Analysis and comparison of MD5 and SHA-1 algorithm implementation in Simple-O authentication based security system.* IEEE International Conference on QiR, pp. 99-104.

[19] Rubinstein-Salzedo, S., & Springer International Publishing. (2018). *Cryptography*.

[20] Singh, G. (2013). *A study of encryption algorithms (RSA, DES, 3DES and AES) for information security.* International Journal of Computer Applications, 67(19).

[21] Sanjeevi, P., Balamurugan, G., & Viswanathan, P. (2016). *The improved DROP security based on hard AI problem in cloud.* International Journal of Internet Protocol Technology, 9(4):207-217.

[22] Singh, V. P., & Pal, P. (2014). *Survey of different types of CAPTCHA*. International Journal of Computer Science and Information Technologies, 5(2), 2242-2245.

[23] Shen, G., Yang, F., & Zhou, L. (2015). *"Usable security of online password management with sensor-based authentication."* U.S. Patent No. 9,141,779. Washington, DC: U.S. Patent and Trademark Office.

[24] Shuster, G. S. (2015). *"Cognitive-based CAPTCHA system."* U.S. Patent No. 8,978,121. Washington, DC: U.S. Patent and Trademark Office.

[25] Sediyono, E., & Santoso, K.I. (2013, August). *Secure login by using One-time Password authentication based on MD5 Hash encrypted SMS.* IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI) pp.1604-1608.

[26] Thomas, G.N. (2015). *Optimum ordered hash tables*.

[27] Todorov, A. (2014). *"Securing passwords with CAPTCHA based hash when used over the web."* U.S. Patent No. 8,640,212. Washington, DC: U.S. Patent and Trademark Office.

[28] Zhu, B.B., Yan, J., Bao, G., Yang, M., & Xu, N. (2014). *Captcha as graphical passwords—a new security primitive based on hard AI problems.* IEEE transactions on information forensics and security, 9(6), 891-904.

[29] Zhai, S., & He, T. (2010, October*). Design and implementation of password-based identity authentication system.* IEEE International Conference on Computer Application and System Modeling (ICCASM) 9(V9):253.

[30] Zhu, H., Zhang, Y., & Xia, Y. (2015). *Enhanced graphical captcha framework and applications to strong security authenticated scheme without password table.* Journal of Information Hiding and Multimedia Signal Processing, 6(6), 1295-1309.

Nafisah Matouq Kheshaifaty is a graduate student pursuing Master of Sciences (MS) degree in Computer Science and Engineering from UQU Makkah main campus. Her studies are fully sponsored by the Ministry of Education in Saudi Arabia. Currently her MS Thesis is specialized in “Engineering Systems Accessibility via Multi-Level Password Authentication” under supervision of Prof. Adnan Gutub within computer security track offered, hoping to complete her MS degree requirements soon during 2021.

Prof. Adnan Abdul-Aziz Gutub is ranked as Full Professor in Computer Engineering specialized in Information and Computer Security within College of Computers and Information Systems at Umm Al-Qura University (UQU). He has been working as the general supervisor of UQU Scientific Council following his previous assignment as the Vice Dean of the Custodian of the Two Holy Mosques Institute of the Hajj & Omrah Research, Known publicly as Hajj Research Institute (HRI), within (UQU), Makkah Al-Mukarramah, all Muslims religious Holy City located within the Kingdom of Saudi Arabia.

Adnan’s academic experience in Computer Engineering was gained from his previous long-time work as Associate Professor, Assistant Professor, Lecturer, and Graduate Assistant, all in Computer Engineering at King Fahd University of Petroleum and Minerals (KFUPM) in Dhahran, Saudi Arabia. He received his Ph.D. degree (2002) in Electrical and Computer Engineering from Oregon State University, USA. He had his BS in Electrical Engineering and MS in Computer Engineering both from KFUPM, Saudi Arabia.

Adnan's research work can be observed through his 110+ publications (international journals and conferences) as well as his 5 US patents registered officially by USPTO with his main research interests involved optimizing, modeling, simulating, and synthesizing VLSI hardware for crypto and security computer arithmetic operations. He worked on designing efficient integrated circuits for the Montgomery inverse computation in different finite fields. He has some work in modeling architectures for RSA and elliptic curve crypto operations. His interest in computer security also involved steganography and secret-sharing focusing on image-based steganography and Arabic text steganography as well as counting based secret sharing. Adnan’s research interest in computing and information technology have been broaden to also relate to smart crowd management and intelligent transportation engineering systems as smart systems and internet of things (IoT) research because of the involvement in Hajj and Omrah Research at UQU - Makkah.

Administratively, Adnan Gutub filled many executive and managerial academic positions at KFUPM as well as UQU. At KFUPM - Dhahran, he had the experience of chairing the Computer Engineering department (COE) for five years until moving to Makkah in 2010. Then, at UQU - Makkah, Adnan Chaired the Information Systems Department at the College of Computers & Information Systems followed by his leadership of the Center of Research Excellence in Hajj and Omrah (HajjCoRE) serving as HajjCoRE director for around 3-years until the end of 2013. Then, he was assigned his position (until March 2016) as the Vice Dean of HRI, i.e. the Custodian of the Two Holy Mosques Institute of the Hajj & Omrah Research, followed by his last 2-year administrative position as general supervisor of UQU Scientific Council, which ended September 2020.