Protecting Cloud Service Providers Based on an Efficient Password-based Authentication System

Saja J. Mohammed*

Computer science department, College of computer science and mathematics, University of Mosul, Mosul, Iraq

Abstract

Password-based authentication systems are one of the most popular authentication systems. They are an essential way to protect many online applications and accounts. Cloud computing services also use this type of protection. Text passwords are employed in cloud computing for the authentication and authorization of access to cloud services or for the preservation of private and sensitive data kept in the cloud. For this reason, a robust password is needed, which must be safe from possible attacks. That requires the password to be as complex as possible; on the other hand, complex passwords will produce the problem of forgetting them. For these types of problems, this paper proposes a new password generator algorithm based on the 4D hyperchaotic system and a genetic algorithm. The proposed algorithm helps generate the same complex password if the same input is entered into the system. On the other hand, a genetic algorithm is also used to enhance the generated password if it loses one of the determined conditions to be considered a robust password. Then the generated password will be used to protect user data stored with any cloud provider. Testing results gave a 79.8 value of bit entropy for the generated passwords, and 93% of the generated passwords were classified as “very strong passwords."

Keywords: Cloud computing security, Authentication system, Genetic algorithm, 4D hyperchaotic system, SHA-256, Password-based authentication systems.

1. Introduction

At last era, cloud computing was and still the most trending service, however, there are many concerns surrounding its usage. Account hijacking is a major risk to cloud data security. Two prevalent instances of extremely inadequate password security are the reuse of passwords and the use of weak passwords. Because a single stolen password may be used on several accounts, this problem makes phishing schemes and data breaches more harmful [1], [2].

In other side, the cloud computing ecosystem is at risk of attacks because of the distant locations of resources and virtualization technologies. Because every client in cloud computing has access to the same resource location, there is a risk to system security. Furthermore, there is a problem with integrity when it comes to transfer, storage, and retrieval. In addition, there isn't a single standard to guarantee data integrity. Because various vendors use distinct structures for data access and storage, clients are unable to quickly switch suppliers and are stuck with only one. The requirement for a consistent standard for encryption, decryption, and client control is the other issue raised. When sensitive data is sent and stored in a cloud environment without adequate encryption and security, there is an increased risk of attacks [3], [4], [5].Therefore, the organizations must put robust security measures in place to protect user data, such as encryption and access limitations, and promptly alert clients in the case of a breach. Online, sensitive data is protected by a number of methods, including frequent data backups, authentication, and encryption [3], [6], [7]

There are three various authentication mechanisms: The password-based authentication system is a method used to verify the identity of a user by requiring them to provide a password. It is one of the most common and widely used approaches for authentication in the digital world [8]

A client can safely use password-based authentication to gain access to services like emails hosted by a service provider. In order to keep unapproved users from using his services, the client needs to give the service provider their login and password. If the username and password combination supplied by the client match the username and password combination in the service provider's database, the client is allowed access to the service he has requested. The primary benefit of password-based authentication is its ease of use and memorization [9].

Token-based authentication verifies a user's identification before allowing them to access a server, network, or other protected system. During this validation procedure, a security token that the server provides must be used. The service's duties also include enabling user inquiries and security token verification. Smart cards, USB keys, mobile devices, and Radio Frequency Identification (RFID) cards are examples of electronic devices that are often used for identification and authentication. Every time a device is used, a new password is created, making it possible to use a security token to login to a computer or virtual private network. In order to accomplish this, the user has to input the password generated by the device into the corresponding prompt [10]
Biometric-based authentication is a security procedure that is dependent on the distinct biometric attributes of an individual. This form of authentication is employed for the purpose of regulating entry to both physical and digital resources, including but not limited to buildings, rooms, and computer equipment [11], [12], [13]
The paper focused on text-based authentication system, according to it popularity and easiness. It takes in its consideration generate a complex text password, which can be regenerated, then use to protecting cloud service providers.

Traditional text-based authentication involves entering login credentials directly in the form of alphanumeric characters in the login box, by text password and username. If compared to other authentication methods, text-based passwords are less costly and require less time to create[14].

A password is a string of alphanumeric characters and symbols that is used to verify a user's identity, provide access to a resource, or authenticate a user. Online privacy may be more easily compromised by attackers and criminals if bad password practices are used. In other hand, using numerous passwords may be complex, susceptible to difficulties in remembering several passwords, and the temptation to reuse a single login credential across many accounts, among other concerns [15].

The ability of the user to develop strong password habits, such as changing passwords frequently, not using the same login information across multiple systems, and creating long, complex passwords that combine special characters, numbers, and symbols, is a major factor in how secure passwords are[16].

Later, text password-based authentication becomes risky because users fail to recall text passwords' length and strength. Users are therefore likely to choose weak passwords to improve recall. Furthermore, using techniques like dictionary cracking, guessing, shoulder surfing, and other methods, hackers may easily get the passwords [14], [17].

Longer passwords are extremely difficult to crack, which is one of their strengths. It is crucial to use strong passwords whenever using passwords. A strong secret key combines capital

letters, lowercase letters, digits, and distinctive characters. Security experts now advise using passwords with 12 characters or more. Many websites are widespread on the Internet to evaluate and check the strength of the generated password [18].

Across all these cloud risk and the problems of text password authentication system, many works had been interested with cloud security, privacy, and authentication. They take in their considerations the cloud security requirements, one of these requirements is authentication which is the focus of this paper. The paper is focused on the most popular and easy to use "text-password authentication system", trying to cover the problem of unmemorable complex password. The paper provides complex text password from memorable code that entered by user. Then use the generated password to protect the user account of any cloud provider. The generated robust password is tested by pre-defined conditions. The generated algorithm is based on hyper chaotic system integrated with genetic algorithm which guarantees that the generated password is robust according to the predefined conditions. The robust password, finally, then is used to investigate cloud account.

The paper structured as follows: general introduction and fundamental of Password-Based Authentication was written in Section 1.The 4D Hyperchaotic system was explained in section 2 whereas the section 3 presented the principle of Genetic Algorithm (GA). The proposed passwords generator was comprehensively shown on section 4 and the results with the discussion were explored in section 5. Last, the paper is finished with its conclusion.

3. 4D Hyperchaotic System:

Chaotic system is a deterministic system which can produce a large number of great pseudorandom sequences because it is very sensitive to changes in the starting value. This is congruent with the keystream needed for many securities application according to the generated sequence's diffusion and scrambling [19], [20].

A state of chaotic and seemingly random behavior that results from deterministic equations is referred to as chaos in a dynamical system. While there could be several variables in a conventional chaotic system that behave in complicated and unpredictable ways, these variables are often coupled in a somewhat straightforward way [12], [21], [22].

The quantity and complexity of variables involved in a system determine whether it is chaotic or hyperchaotic. A subset of chaotic systems with four or more variables are called hyperchaotic systems. Hyperchaotic systems show much more complexity and unpredictability than chaotic systems do. Multiple positive Lyapunov exponents, which show the rate of exponential phase space divergence of nearby trajectories, are what distinguish them [23].

Hyper-chaotic systems have applications in a wide range of fields, including secure communication, encryption, and random number generation. They are of importance in these domains because of their potential to make it more difficult for an opponent to predict or comprehend the behaviour of the system due to their increased complexity. Researchers study hyper-chaotic systems for their theoretical properties and practical applications in fields that benefit from complexity and unpredictability [24].

Multiple variables in hyper-chaotic systems lead to even more complicated and chaotic behaviour, frequently with extra degrees of nonlinearity and complexity in their dynamics. Elevated positive Lyapunov exponents, a crucial sign of chaos, measure the system's susceptibility to initial conditions and characterize these systems. In a hyper-chaotic system,

the presence of more than one positive Lyapunov exponent typically suggests that many variables are developing chaotically in different directions. Equation (1) provides an explanation of the hyper-chaotic system that is utilized in this paper [25] :

$$\begin{cases} \dot{x} = -x - 4y, \\ \dot{y} = x + z^2 + aw, \\ \dot{z} = 1 + x\,, \\ \dot{w} = -by, \end{cases} \qquad \text{..... (1)}$$

in which $[x, y, z, w]^T \in R^4$ is a state vector and the two positive constants parameters are a and b. This system exhibits chaotic hidden attractors as explained in Figure. 1.

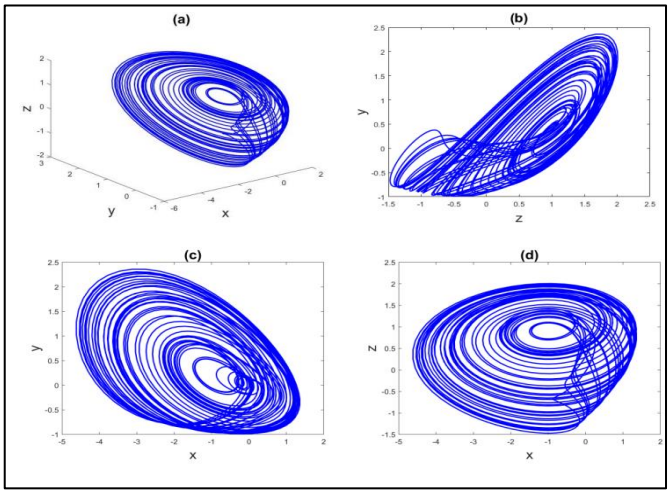

Figure-1. 4D hyperchaotic Sprott S system

4. Genetic Algorithm

In informatics and computational mathematics, the term " Genetic algorithm " (GA) refers to the broad category of evolutionary algorithms. These algorithms, which focus on bio-inspired operators like selection, convergence, or mutations, are widely employed to produce excellent solutions to optimize and search problems [26].

In recent years, metaheuristic algorithms have been used to address intricate real-world issues that emerge from many domains, including economics, politics, management, and engineering. The algorithms may be roughly categorized into two groups: single-solution algorithms and population-based metaheuristic algorithms [27], [28]

The genetic algorithm (GA) is a well-recognized algorithm based on the biological phenomenon of evolution. Genetic algorithm follows the Darwinian concept of natural selection, favouring individuals most well-suited to their environment. J. H. Holland introduced the concept of Genetic Algorithms in 1992. The Genetic algorithm have three essential components: chromosomal representation, fitness selection, and operators that emulate biological processes [28]. The operators inspired by biological processes include selection, mutation, and crossover. During selection, chromosomes are chosen based on their fitness value to undergo further processing. The crossover operator selects a random locus and modifies the sub-sequences across chromosomes to generate progeny. During the mutation process, some segments of the chromosomes undergo random flipping, influenced by

probability [27], [28]. The classical Genetic algorithm is explained in the following algorithm [26]. Where, Figure 2 explains the steps of Genetic algorithm [28].

- **Genetic algorithm algorithm steps to obtain fitness:**
  Input: Population p
  Output: Fitness value

1. Randomly consider populations p.
2. Find out how fit the population is.
3. Continue doing steps 4 through 7 until convergence.
4. Select a parent at random from the population.
5. Creates a new population by use of crossover operation.
6. To perform mutation operation, introduce arbitrary chromosome into a new population.
7. Determine the fitness of recently created populations.
8. Out put fitness value

These days, Genetic algorithms are used in a lot of places. Examples include the Internet of Things, Smart Traffic Signal System, Intelligent Routing in MANET of Smart Devices, Blockchain Technology, Cloud Computing's load balancing and work scheduling, and Engineering Pedagogy [26].

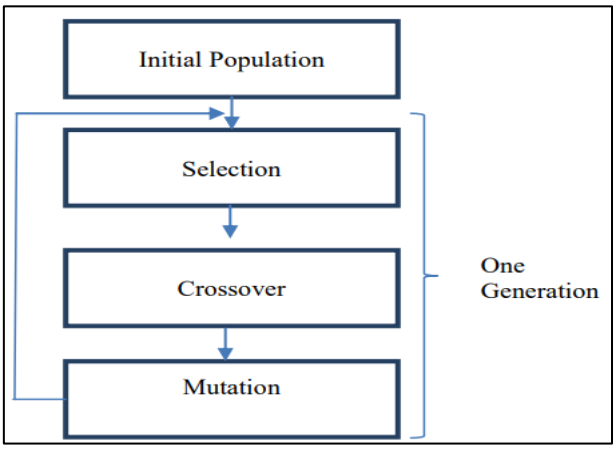

Figure - 2. Algorithm of the classical genetic Algorithm

5. Secure hash algorithm (SHA):

The Secure Hash Algorithm (SHA) has been the most often utilized hash function in recent years. The fact that nearly all other commonly used hash functions had significant cryptanalytic flaws by 2005 meant that SHA was essentially the only standardized hash algorithm still in use. Since SHA's architecture closely resembles that of Message Digest 4 (MD4) , it is based on the MD4 hash algorithm [29].
Preprocessing and hash computation are the two processes that make up each SHA algorithm. Preprocessing includes initializing data to be utilized in the hash calculation, padding a message, and parsing the padded message into m-bit blocks. From the padded message, the hash computation creates a message schedule. It then iteratively constructs a sequence of hash values using the schedule, functions, constants, and word operations. The message digest is calculated using the final hash value that is produced by the hash computation [30].

Nowadays, SHA-256 is the most often used SHA function because it offers a high level of protection given the capabilities of modern computers. A 256-bit hash value is computed by SHA-256 for a 512-bit input message. The hash value for a lengthy message might need to be calculated by the actual program. The message is split up into several 512-bit data blocks in these situations. Padding is added if the final block has less than 512 bits. Fig. 3 displays the hash computation for a lengthy message. One data block at a time, the SHA-256 algorithm calculates intermediate hash values. The hash value from the previous block is used as the starting hash value for the hash computation of the subsequent block. The hash value of the entire message is regarded as the outcome of the last data block. Fig. 4 shows an overview to SHA 256 operations [31].

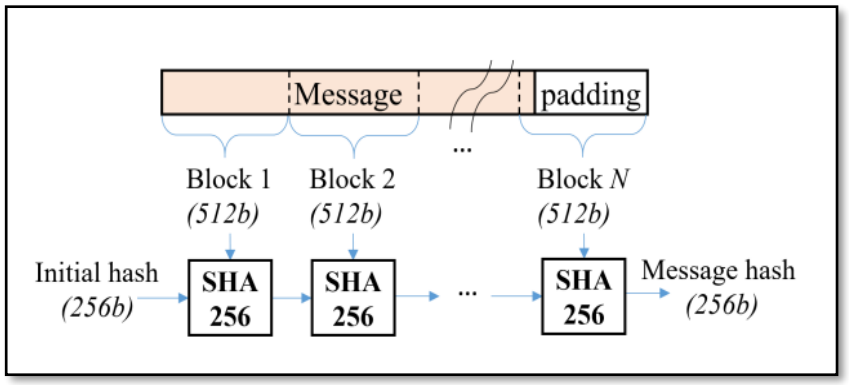

Figure .3 The hash computation for a lengthy message

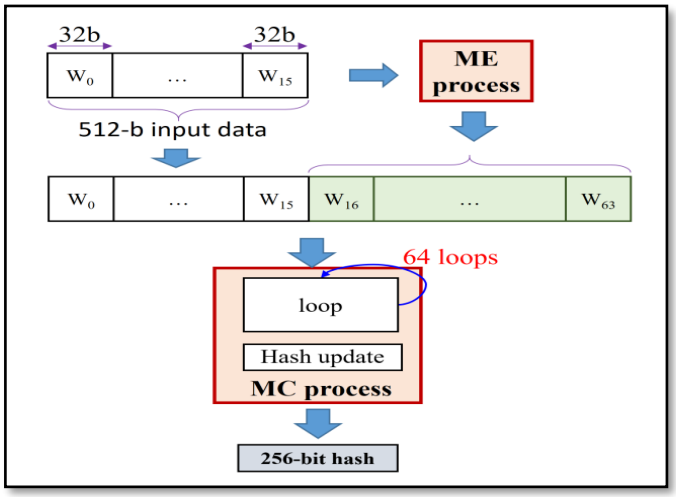

Figure. 4 An overview to SHA 256 operation

6. The proposed passwords -based authentication system:

The proposed system is divided in to two major phases:

Phase a. Generating a strong password.

Phase b. Designing a text based authenticated model to protect cloud account.

Passwords are generated using a 4D hyperchaotic system integrated with genetic algorithm. Each one of the previous principles plays a great role in password generation stages. The chaotic system provides the required randomization that helps to create the initial passwords. It depends on the user input value to begin its job. By no. of steps, the password is created.

Where the genetic algorithm is used to test and increase the robustness of the generated password, its stages are applied when the created password has some weak points.

It is worth noting that the restrictions imposed in the proposed algorithm to consider the password a strong are:

- At least the password length must be 10 characters.
- At least 50% of it includes (capital letter (A - Z), small letter (a - z)).
- The other ratio is divided between numbers characters (0-9), the special characters (!"#$%&()*+,- ./:;<=>?@[\\]^_{|}~'}).
- Dose not contain respective similar characters.
- Every two consecutive alphabetical characters in the generated password must not be consecutive in the alphabet.

The second phase include using the generated password as text authentication system to protect cloud accounts from the unauthorized access. Note that the two phases are overlapped with each other during the real execution of the proposed system. The following subsection explains these phases in detail

5.1 Phase A, Generating a strong password

The following is an explanation of the steps in the suggested password generating algorithm (phase a):

The proposed algorithm to generate robust password:

Input: user initial data (D), password length
Output: Robust password used in cloud services provider.

1- Accept input data from user (say UD), with password length (from 8 to 16). UD must be at least 3 characters.
2- Convert UD to ASCII code format (say UD_{asci}).
3- Using UD_{asci}; compute 4D hyperchaotic system initial value (X).
4- Generate chaos sequence using the 4D hyperchaotic machine, then generate an initial password (PW_i) from resulted sequence.
5- Using the specific robustness conditions to check the robustness of the PW_i :
   if PW_i reaches all condition exit with it,
   else continue.
6- To begin with genetic algorithm: PW_i will be the first chromosome (Y1).
7- Changing parameters of 4D hyperchaotic system and generating new password it will be second chromosome (Y2).
8- Begin with GA operator by applying the following operations:
   a) Crossover operation between Y1 and Y2 (using single-point crossover (char to char)).
   b) Mutation operation.
9- If the result dose not satisfy the conditions of robust password:
   Take the most robust chromosome between Y1 and Y2; consider it as Y1; go to 7.
   Else exit with the robust password (PW_{robust}).

5.2 Phase B, designing a text based authenticated model to protect cloud account

To protect the cloud account from any un authorized access, a proposed model was suggested based on text password authentication system (phase b). The proposed model used the generated password in phase (a). phase (b) also has two options: *Registration,* and

Login/Authentication phases. The two phases need to input the email address of the user. But the operation direction takes another path in each one of them.

A. Registration Phase: a new user always needs to register (or sign up) on any specific account. Normally, any registration needs a user name, user email address, and password as inputs to the authentication system. So, the proposed registration phase can be detailed as follows:

1. The first step to complete registration of cloud account is to input the user's name and email address.
2. The second step is to enter a robust password (PW). The robustness of the input PW is checked (according the previous explained conditions), if it passes all conditions, go to 5; else, open a new window to generate robust PW (PW_{robust}),
3. Ask the user to input a memorable code. This operation is the beginning to generate the robust strong password (PW_{robust}). The user can use the output PW then.
4. Input the generated password (PW_{robust}).
5. Hashing all input data (using SHA-2 (SHA-256) hash algorithm).
6. Create an encrypted user's record (user name, e-mail, and PW), then save it in cloud secure database. Figure 5 shows the steps of the proposed registration (sign up) algorithm.

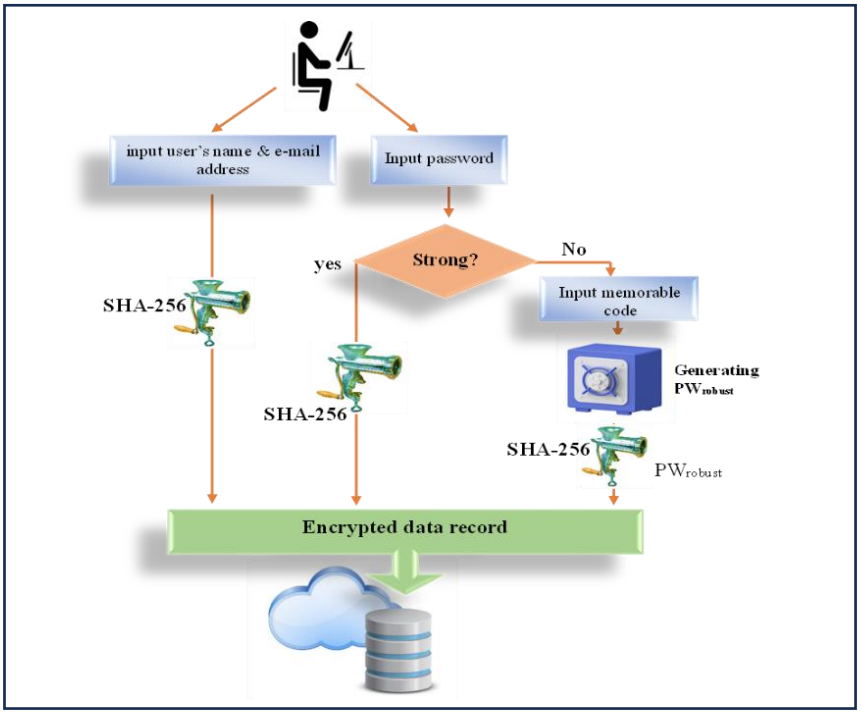

Figure – 5. The proposed registration algorithm

B. Login/Authentication phase: when a trusted user tries to sign in to his/ her account in the cloud provider, the following steps are applied:

1. Receiving the user's name, encrypting them by SHA 256.
2. Search in cloud password's database for the entire encrypted user name; if there is an identical encrypted one: continue login operations; else the user can try to input new user name 3 times; if after that he/ she fail, exit with an error message and deny access.

3. If the user is authorized, continue input user email's address, and his/ her password.
4. For input generated password (PW $_{robust}$): open the dialog window to input the user memorable code.
5. Generate strong password (PW $_{robust}$) using user code, the generated password will be entered directly with user information.
6. After send this information, they hashed and continue to compare with the stored one which place in the same row of the identical user's name.
7. If they are identical, the user is authorized, else give the user three times to reinput his information; if fail: deny user access.

Figure 6 explain proposed sign in algorithm.

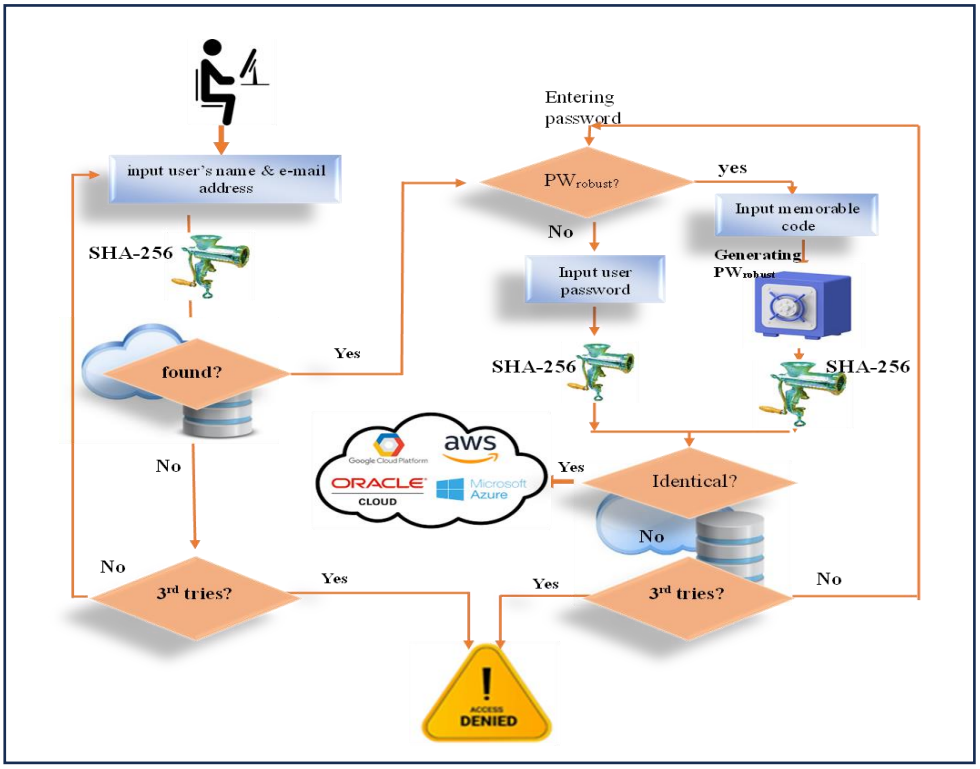

Figure – 6. The proposed log-in algorithm

6. Results and Discussion

Many practical experiments were applied based on the proposed algorithm. All thses experiments were applied on a PC (CPU=Core i5, RAM 16 G) with Windows 11 64-bit operating system and Visual C# 2022 programming language. Table 1. shows some examples of generated passwords, in various length, using the proposed algorithm.

Table 1- Some examples of the generated passwords with various length

Case	User Input	Password length	Generated password
1	config	8	}fA:#4h&
2	Compute	9	Bs{3XzV1}
3	myname	10	k^`&4d-W?9
4	Net2024	11	(4p-$;>~A,S
5	Mobile	12	m)A,gB5^9/~X
6	mobile	13	}fhA:#4h&B50%
7	pass	14	9BS*8#zg=X:af\
8	Word	15	fZ@?4W<f%zrHnuA
9	Iris	16	BkS{j8;zg=j4]A#2

As Table 1 showed, the generated passwords (explained in Table.1) underwent many tests to check their robustness. Some of most popular websites are used to test their robustness [24]-[28]. The results of testing explained in Figure 7 shows that among various generated passwords with various lengths, 95% of the passwords of length 16 were classified as “very strong.”. Only 5% of them were classified as weak passwords (those of length 8).

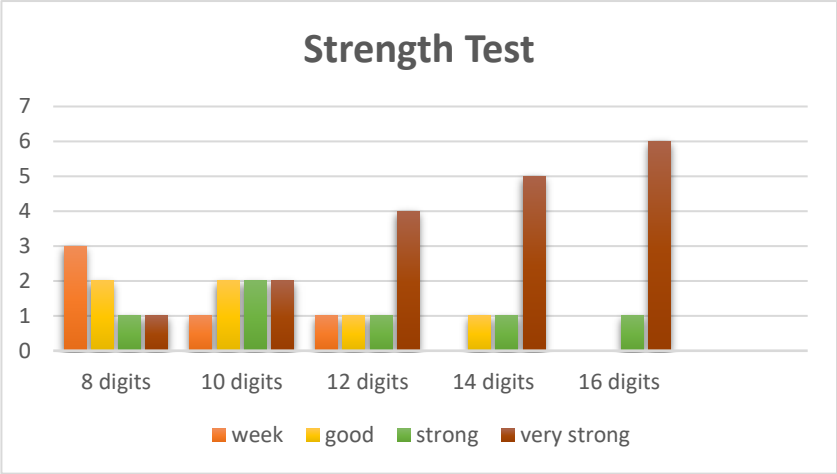

Figure- 7 The results of testing various length of the generated passwords

The elapsed time for generating passwords using the proposed algorithm is calculated. Table 2 explains the result of calculating the time; it shows that the time is linearly proportional with the length of the password. The proposed algorithm does not affect the length of user input compared with the required length of generated password.

Table 2- The elapsed time of generating passwords

Password length	Average Time (Milliseconds)
8	0.0178267
9	0.0444572
10	0.1284238
11	0.1927973
12	0.2358594
13	0.5745209
14	0.7710640
15	0.8702945
16	1.0109572

The crackability of the generated passwords was also tested using the online service "Password Checker Online." [29]. This service depends on "the brute force attack" to check the crackability of the generated passwords using various machines. Table 3 explains the results of some testing samples of the generated passwords using "Password Checker Online." According to the results of this test, the proposed algorithm is not easily penetrable to brute force attack.

Table 3- The estimated time of brute force attack using Password Checker Online using various machines

Password length	machines					
	Standard Desktop PC	Fast Desktop PC	GPU	Fast GPU	Parallel GPUs	Medium size botnet
8	2 years	6 month	2 month	1 month	4 days	1 minute
9	$2x10^3$ years	46 years	18 years	9 years	11 month	2 hour
10	$208x10^3$ years	$52x10^3$ years	$21x10^3$ years	$10x10^3$ years	78 years	6 days
11	20 $x10^6$ years	5 $x10^6$ years	2 $x10^6$ years	277 x 10^3 years	$78x10^3$ years	20 years
12	$2x10^9$ year	459 $x10^6$ year	$184x10^6$ year	92 $x10^6$ year	9 $x10^6$ year	$2x10^3$ years
13	$173x10^9$ year	$43x10^9$ year	$17x10^9$ year	$9x10^9$ year	$863x10^6$ year	$173x10^3$ year

The entropy of passwords is also calculated using the "password strength" online service [30]. The entropy measures the unpredictability of the generated passwords. The bigger the entropy value, the harder password cracks. When a password's entropy falls between 28 and 35 bits, it is considered extremely weak. It is reasonable (i.e., reasonably secured for network and enterprise usage) if it falls between 36 and 59. A strong password is indicated by entropy values

greater than sixty [32] The proposed system-generated passwords in this checker have entropy values ranging from 39 (minimum) to 79.8 (highest). That proved the difficulty of cracking the generated passwords using the proposed algorithm, as explained in Figure. 8.

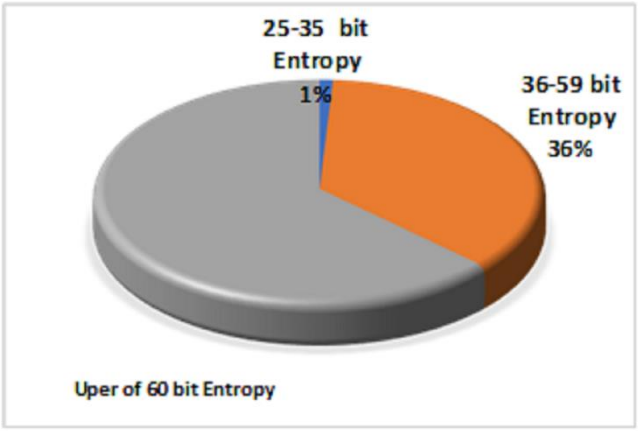

Figure -8 The resulted bit entropy

7. Conclusion

In this paper, an authentication system for cloud provider account protection is proposed. The system was based on the text password, robust and strong one only, to investigate a user authenticated access. The paper proposes authentication system with a strong password generator based on the 4D hyperchaotic system and the genetic algorithm. The 4D hyperchaotic system generates the initial password using user input through ordered steps. If the generated password does not achieve all the considered conditions of a strong password, the genetic algorithm is used to enhance the resulting password. The proposed algorithm proved that the generated passwords are safe against brute-force attacks. That fact was concluded when the attack took a long time to discover the password. 63 % of the generated passwords have a bit entropy greater than 60 (that means robust password), whereas 36 % have acceptable strength. Also, the practice proved that elapsed time is linearly proportional to the length of the password and affected by user input. The maximum elapsed time does not exceed 0.19 milliseconds, which makes the algorithm acceptable to be used in the day life.

8. Acknowledgment

The authors are very grateful to the University of Mosul/ College of Computer Science and Mathematics for their provided facilities, which helped to improve the quality of this work.

References

[1] S. Achar, H. Patel, and S. Hussain, “DATA SECURITY IN CLOUD: A REVIEW,” 2022.

[2] S. Nalajala, B. Moukthika, M. Kaivalya, K. Samyuktha, and N. L. Pratap, “Data Security in Cloud Computing Using Three-Factor Authentication,” in *Lecture Notes in Electrical Engineering*, 2020. doi: 10.1007/978-981-15-2612-1_33.

[3] M. H. Barkadehi, M. Nilashi, O. Ibrahim, A. Zakeri Fardi, and S. Samad, "Authentication systems: A literature review and classification," *Telematics and Informatics*, vol. 35, no. 5. 2018. doi: 10.1016/j.tele.2018.03.018.

[4] S. J. Mohammed and D. B. Taha, "From Cloud Computing Security towards Homomorphic Encryption: A Comprehensive Review," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 9, no. 4, 2021.

[5] S. Dey, S. Sampalli, and Q. Ye, "MDA: message digest-based authentication for mobile cloud computing," *Journal of Cloud Computing*, vol. 5, no. 1, 2016, doi: 10.1186/s13677-016-0068-6.

[6] E. Younis and S. J. Mohammed, "THE LANDSCAPE OF AUTHENTICATION SYSTEMS: A COMPREHENSIVE SURVEY," *MINAR International Journal of Applied Sciences and Technology*, vol. 5, no. 4, pp. 1–16, 2023, doi: 10.47832/2717-8234.17.1.

[7] S. J. Mohammed and D. B. Taha, "Paillier cryptosystem enhancement for Homomorphic Encryption technique," *Multimed Tools Appl*, 2023, doi: 10.1007/s11042-023-16301-0.

[8] D. Singla and N. Verma, "Theoretical analysis for the fluctuation in the electric parameters of the electroporated cells before and during the electrofusion," 2023, doi: 10.21203/rs.3.rs-2520547/v1.

[9] M. Alajmi, I. Elashry, H. S. El-Sayed, and O. S. Faragallah, "A Password-Based Authentication System Based on the CAPTCHA AI Problem," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3018659.

[10] Z. Xu, J. Xu, and L. D. Kuang, "A Token-based Authentication and Key Agreement Protocol for Cloud Computing," in *Proceedings - 2021 IEEE 6th International Conference on Smart Cloud, SmartCloud 2021*, 2021. doi: 10.1109/SmartCloud52277.2021.00014.

[11] P. Padma and S. Srinivasan, "A survey on biometric based authentication in cloud computing," in *Proceedings of the International Conference on Inventive Computation Technologies, ICICT 2016*, 2016. doi: 10.1109/INVENTIVE.2016.7823273.

[12] S. J. Mohammed, "Using biometric watermarking for video file protection based on chaotic principle," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 15, no. 12, pp. 201–206, Dec. 2017, [Online]. Available: https://sites.google.com/site/ijcsis/

[13] Y. Lu and D. Zhao, "Providing impersonation resistance for biometric-based authentication scheme in mobile cloud computing service," *Comput Commun*, vol. 182, 2022, doi: 10.1016/j.comcom.2021.10.029.

[14] P. C. Golar and R. Sharma, "An Advanced Knowledge Based Graphical Authentication Framework with Guaranteed Confidentiality and Integrity," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, 2023, doi: 10.17762/ijritcc.v11i8s.7676.

[15] A. Ezugwu *et al.*, "Password-based authentication and the experiences of end users," *Sci Afr*, vol. 21, 2023, doi: 10.1016/j.sciaf.2023.e01743.

[16] A. Ezugwu *et al.*, “Password-based authentication and the experiences of end users,” *Sci Afr*, vol. 21, 2023, doi: 10.1016/j.sciaf.2023.e01743.

[17] S. Kaur, G. Kaur, and M. Shabaz, “A Secure Two-Factor Authentication Framework in Cloud Computing,” *Security and Communication Networks*, vol. 2022, 2022, doi: 10.1155/2022/7540891.

[18] C. Luevanos, J. Elizarraras, K. Hirschi, and J. H. Yeh, “Analysis on the security and use of password managers,” in *Parallel and Distributed Computing, Applications and Technologies, PDCAT Proceedings*, 2017. doi: 10.1109/PDCAT.2017.00013.

[19] Y. Hu, H. Wu, and L. Zhou, “A Novel Hyperchaotic 2D-SFCF with Simple Structure and Its Application in Image Encryption,” *Entropy*, vol. 24, no. 9, 2022, doi: 10.3390/e24091266.

[20] S. J. Mohammed and D. B. Taha, “Privacy Preserving Algorithm using Chao-Scattering of Partial Homomorphic Encryption,” in *Journal of Physics: Conference Series*, 2021. doi: 10.1088/1742-6596/1963/1/012154.

[21] A. Mouafak, S. Jasem, and M. Jader, “Apply new algorithm for chaotic Encryption using CBC&CFB,” 2013. [Online]. Available: www.ajbasweb.com

[22] G. TH.Talee, M. J. Jelmeran, and S. J. Mohammad, “A New Approach For Chaotic Encrypted Data Hiding In Color Image,” *Int J Comput Appl*, vol. 86, no. 8, 2014, doi: 10.5120/15006-3233.

[23] S. John and S. N. Kumar, “6D Hyperchaotic Encryption Model for Ensuring Security to 3D Printed Models and Medical Images,” *Journal of Image and Graphics*, vol. 12, no. 2, pp. 117–126, 2024, doi: 10.18178/joig.12.2.117-126.

[24] S. F. AL-AZZAWI and M. A. AL-HAYALI, “Coexisting of self-excited and hidden attractors in a new 4D hyperchaotic Sprott-S system with a single equilibrium point,” *Archives of Control Sciences*, vol. 32, no. 1, pp. 37–56, 2022, doi: 10.24425/acs.2022.140863.

[25] M. A. Al-Hayali and F. S. Al-Azzawi, “A 4D hyperchaotic Sprott S system with multistability and hidden attractors,” in *Journal of Physics: Conference Series*, 2021. doi: 10.1088/1742-6596/1879/3/032031.

[26] T. Alam, S. Qamar, A. Dixit, and M. Benaida, “Genetic algorithm: Reviews, implementations and applications,” *International Journal of Engineering Pedagogy*, vol. 10, no. 6. 2021. doi: 10.3991/IJEP.V10I6.14567.

[27] M. Kumar, M. Husain, N. Upreti, and D. Gupta, “Genetic Algorithm: Review and Application,” *SSRN Electronic Journal*, 2020, doi: 10.2139/ssrn.3529843.

[28] S. Katoch, S. S. Chauhan, and V. Kumar, “A review on genetic algorithm: past, present, and future,” *Multimed Tools Appl*, vol. 80, no. 5, 2021, doi: 10.1007/s11042-020-10139-6.

[29] W. Stalling, *Cryptography and Network Security, Principles and Practices*. 2017.

[30] U. S. Commerce and N. I. and Technology, “Secure Hash Standard (SHS),” *Federal Information Processing Standards Publication 180-4*, no. October, 2012.

[31] T. H. Tran, H. L. Pham, and Y. Nakashima, “A High-Performance Multimem SHA-256 Accelerator for Society 5.0,” *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3063485.

[32] F. Z. Glory, A. Ul Aftab, O. Tremblay-Savard, and N. Mohammed, “Strong Password Generation Based on User Inputs,” in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2019*, 2019. doi: 10.1109/IEMCON.2019.8936178.

[33] Nordpass, https://nordpass.com/secure-password/

[34] Passwordmonster, https://www.passwordmonster.com/

[35] UIC Academic, https://www.uic.edu/apps/strong-password/

[36] Bitwarden, https://bitwarden.com/password-strength

[37] LastPass, https://lastpass.com/howsecure.php

[38] Password Checker Online, http://password-checker.online-domain-tools.com/#google_vignette

[39] Strength test, https://rumkin.com/tools/password/