Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-021-02893-8

ORIGINAL RESEARCH

Efficient secure data retrieval on cloud using multi-stage authentication and optimized blowfish algorithm

S. Immaculate Shyla[1] · **S. S. Sujatha**[2]

Received: 6 March 2020 / Accepted: 8 January 2021

Abstract
Cloud computing is currently playing an important role in the information technology industry because of its improved efficiency, wide access, low cost, and many benefits. It also provides more space for storing data and transmitting data from one location to another faster for different users on the Internet. Due to large storage, cloud customers can save huge capital investment on IT infrastructure and focus on their own core business. Therefore, many companies or organizations are moving their business to the cloud. However, many customers are reluctant to use the cloud due to security and privacy concerns. To tackle this problem, in this paper, efficient secure data retrieval is developed with the help of multi-stage authentication (MSA) and optimized blowfish algorithm (OBA). The proposed system consists of three modules namely, MSA, data security, and data retrieval. Initially, the cloud users register their information on cloud based on a multi-authentication procedure. After the registration process, the data are encrypted with the help of OBA. To increase the security of the system, the key value is optimally selected with the help of a binary crow search algorithm. After the encryption process, MSA based data retrieval process is performed. This will avoid, un-authorized person to attack the data. The performance of the proposed methodology is implemented in JAVA and performances are analyzed in terms of different metrics.

Keywords Cloud computing · Multi-stage authentication · Optimized blowfish algorithm · Registration · Binary crow search algorithm

1 Introduction

In recent years, cloud computing (CC) has made great strides in the technology industry and the scientific community (De la Prieta et al. 2019). CC is a computing model that can be used anywhere, anytime. They only pay the amount based on usage. This method is called pay-as-you-go fashion (Kumar et al. 2019). Storage is one of the most influential and needed computing resources in the current digital era. It is one of the most popular services in the CC industry (Helmi et al. 2018). Due to a large amount of storage, a lot of organizations and industries store their data on the cloud. Amazon's Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (S3) and apple icloud are well-known examples of cloud data storage. However, security is a major issue in cloud computing. To overcome the security problem, a lot of cryptography algorithms and access control mechanisms are introduced. Security goals are set at three points namely, confidentiality, integrity, and availability. Cryptography is concerned with the confidentiality of data in the cloud.

To access the cloud storage data, the access control mechanism is utilized. Access control technology can not only ensure the valid access requests of valid users but also prevent the invasion of unauthorized users, as well as address security issues caused by the misuse of valid users. Traditional access control is identity-based authentication technology and operates within the confines of a unified security domain (Vafamehr and Khodayar 2018; Li et al. 2009). For access control mechanisms, single authentication, biometric authentication, and multi-authentications methods are developed. The single-stage authentication approach may steal. Bio-metric authentication namely, fingerprints, palm prints, hand geometry, face recognition, voice recognition, iris recognition, and retina

✉ S. Immaculate Shyla
immaculateshylas1918@gmail.com

[1] Department of Computer Science, St.Alphonsa College of Arts and Science, Affiliated to Manonmaniam Sundaranar University, Tirunelveli, Nagercoil, India

[2] Department of Computer Applications, S. T. Hindu College, Manonmaniam Sundaranar University, Tirunelveli, India

Published online: 19 March 2021

authentication have been proposed in the literature. Each biometric accreditation program has its advantages and disadvantages, which are based on several factors, such as sustainability, individuality, and acceptability (Kushida and Pingali 2014; Burger 2001). One of the main drawbacks of using bio-metrics is its intrusion into the user's character. Besides, most biometric systems require a special scanning device to authenticate users, which does not apply to remote and Internet users. To avoid the problem, multi-stage authentication (MSA) is developed. The MSA consists of more than three security layers (Kang et al. 2015, Dinesha and Agrawal 2012; Rajani et al. 2016).

Similarly, recently, many cryptography-based secure data transaction is presented namely, Advanced Encryption Standard (AES) (Sachdev and Bhansali 2013), Data Encryption Standard (DES) (Ramya et al. 2016), Rivest, Shamir, & Adleman (RSA) (Somani et al. 2010), SHA-256 (Sundarakumar and Mahadevan 2019), elliptical curve cryptography (ECC) (Bai et al. 2017) and blowfish algorithm (Reddy et al. 2015) etc. multi stage authentication based security also introduced. Even though, some problems namely, maximum execution time, cost, and information loss are not reduced. The metaheuristics algorithm also developed to improve the performance of cryptography algorithms. To avoid the problem, an efficient new algorithm is needed to solve the security issues.

The main objective of the proposed methodology is to securely transmit the data on the cloud using MSA and optimized blowfish algorithm (OBA). MSA is three-stage security layers that avoid unauthorized users access the data on the cloud. The proposed works have developed based on dual security layers such as MSA based access control and cryptography algorithm. In the authentication process, MSA is developed. In MSA, if any Stage Client does not provide the information correctly, he will be rejected immediately. Thus, no unknown person can take the information from the storage center. After the authentication process, data are encrypted using OBA. Here, the key values are optimally selected using binary crow search algorithm (BCSA). Finally, the user retrieves the data, if they satisfy the multiple authentication processes. In this manner, unauthorized users are avoided. The main contribution of the paper is listed below;

- To encrypt the data blowfish algorithm is utilized. To enhance the blowfish algorithm, key values are optimally selected using BCSA. This will hide the original information from malicious.
- To avoid the malicious login process, MSA process is proposed. This will secure the provider from malicious.

The rest of the paper organized as follows; in Sect. 2, related work is presented and the attack model is given in Sect. 3. The proposed secure data transaction is presented in Sect. 5 and the experimental results are presented in Sect. 6. Finally, the conclusion is presented in Sect. 6.

2 Related work

A lot of research has been developed to secure data transactions on the cloud. Among them some of the works are analyzed here; Cheng et al. (2018) had developed an Identity-Based Encryption (IBE) based accountable privacy-preserving mechanism on CC. Initially, based on the privacy attributes accountable privacy-preserving mechanism is presented. Second, the proposed accountability for CC involves the privacy-protecting mechanism, the proposed accounting, and auditing approaches. The experimental of the proposed methodology is analyzed in terms of different metrics. In Sudhakar and Rao (2020), Sudhakar et al. had developed a secure aware data transaction on the cloud using an index based quasi–identifier approach. Here, they utilized an incremental and distributed data set for experimentation. Here, initially, input data are clustered with the help of modified fuzzy c-means clustering (MFCM). Then tuple partitioning is done. After that, important data are selected from the clustered output. To avoid sensitive data loss, data are secured with the help of Bucketization.

Brindha and Shaji (2018) had developed a conditional source trust attributes encryption mechanism with particle swarm optimization (CSTAE-PSO) based secure data transaction on the cloud. Here, initially, condition attributes are selected and then, the selected attributes are encrypted with the help of the CSTAE mechanism. In this paper, to increase the performance of the encryption process, the bilinear mapping transformation function was included in CSTAE. To attain the minimal transaction with completion time, the PSO algorithm was developed. The performance of the proposed methodology was analyzed in terms of different metrics namely, throughput level on the transaction, security rate on the data layer, and transaction completion time. Similarly, Thangavel and Varalakshmi (2018) had developed a secure data storage and retrieval on the cloud using enhanced DNA and ElGamal cryptography algorithm. Here, initially, the data are encrypted with the help of the DNA algorithm, and the encrypted information was transferred between the data owner and data client on the cloud. To address the key management issues, the ElGamal cryptosystem was utilized. The proposed method provided better user authentication and performance over security record against attacks.

Kanna and Vasudevan (2019) had developed a hybrid crypto mechanism-based privacy preservation on the cloud. The crypto mechanism was designed based on a fully homomorphic–elliptic curve cryptography (FH-ECC) algorithm. Initially, DO encrypt the information using the ECC

algorithm. To improve the security of the data, again data was encrypted with the help of a fully homomorphic (FH) algorithm. After encryption process data was stored on the cloud. After the storage process, the access control policy was developed to avoid the unauthorized person login. The performance of the proposed methodology was analyzed in terms of different measures namely, execution time, encryption time, and decryption time. Moreover, Mohiuddin et al. (2019) had developed adaptive bin packing algorithm based secure data storage on the cloud. Here, they introduced an end to end security framework for data at rest in cloud storage to eliminate insider threats. The security threads were identified and performance was analyzed.

Pournaghi et al. (2020) had explained a block chain and attribute based encryption is developed. In this paper, they securely store the medical data on cloud. To avoid the unauthorized person login process, fine-grain access control mechanism is utilized. Moreover, Sumathi and Sangeetha (2020) had developed a data security on cloud. In this paper, the security issues are reduced by Group Key Based Attribute Encryption using Modified Random Fibonacci Cryptographic (MRFC).initially, the input data was separated into sensitive and non-sensitive data using attribute segregation method. Then, the sensitive data are encrypted using MRFC. The performance of this approach is analyzed in different metrics. Pragaladan and Sathappan (2018) had developed Combining DNA Structure and Multi-aspect Time-Integrated Cut-off Potential based secure data storage on cloud. This method was reducing the time complexity for establishing confidentially of the data. This framework improves the authentication level of security by using confidentiality and authentication techniques from unauthorized user change. The DNA-MACP framework with comprehensive security analysis and implementation reduces time and space issues in business data transactions in a cloud environment. Suresha and Karthick (2020) had explained a data security using Threshold Cryptography Technique. This approach addresses data security issues in the cloud system more efficiently. In this threshold cryptography technique, the data owner creates users in groups based on location or location, project and department, the data owner assigns a single secret key to encrypt data to each group, and each user in the group shares parts or pieces of the secret key. This method also uses the Data Access Control List to control data access. The proposed approach provides data protection more efficiently, which increases the efficiency of the system and reduces the number of secret keys.

Resende et al. (2015) had explained a Physical Unclonable Function (PUF)-based mutual multifactor entity and transaction authentication for secure banking. Here, PUF with Password-based Authenticated Key Exchange (PAKE). Using this method, only parties authenticated in the current session can valid bank transactions. Tsai and Su (2020) had explained the authentication of online banking customers and transactions through use of a hash-based multi-server authentication scheme in conjunction with a smart card. The proposed system provides strong security features and low maintenance costs for financial institutions' Internet banking platforms. The proposed mechanism was associated with a customized interface and thus easily integrated into existing banking systems for use in Internet banking applications. Moreover, Guo et al. (2019) had explained a block chain algorithm based security for online education classes. The introduced block chain-enabled digital rights management system has been demonstrated as a promising candidate solution to the block chain-based multimedia data protection in an online education environment. Similarly, Mubarakali et al. (2019) had developed health record transaction using block chain algorithm. In healthcare, this work enables you to securely control and share a patient's health record in cloud storage without infringing on any privacy. This provides a great way to personalize patient data in intelligent health systems.

3 Attack model on cloud

The cloud data are loosed due to attacks. In this, the unauthorized persons are attack the sensitive data. Lot of arrack models are available in cloud. In this paper, five important attacks are analyzed namely, Denial of service (DOS), Distributed denial of service (DDOS), Hijacking Account, malware injection attack, man- in- the- middle- attack and authentication attack. In the DoS attack, the attacker overloads the target cloud system with service requests, thereby stopping responding to any new requests, so its users are unavailable. A DDoS is a DoS that uses a high number of has to make the attack indeed more disruptive. The number of hosts can reach hundreds of thousands. Cloud account hijacking is the process by which an individual or company's cloud account is stolen or stolen by an attacker. Cloud accounting is a common tactic in identity theft schemes, in which an attacker uses stolen account information to perform malicious or unauthorized activity. In Cloud Malware Injection Attack an attacker tries to inject malicious service or virtual machine into the cloud. Authentication is a weak point in cloud computing services which is frequently targeted by an attacker. The above mentioned attacks are most popular in cloud. To avoid the attack, in this paper MSA with cryptography algorithm is developed.

4 Proposed secure data transaction on cloud

Cloud computing is a service that has been rapidly increasing its growth in the information technology industry in recent years. Privacy and security are challenging issues

for cloud users and providers. In a public cloud environment, users cannot control its remote data as it transfers its data to a public cloud server. Therefore, information security is a critical issue in public cloud storage, such as confidentiality, integrity, availability, and reliability of data. To resolve this issue, in this research, an OBA as well as MSA is proposed. OBA aims to strengthen sensitive data confidentiality in public cloud storage. The proposed method consists of three stages namely, MSA, data security using OBA, and query-based data retrieval. In this method SaaS is used. This method prevents the data into insider attacker and outsider attackers. The overall diagram of the proposed methodology is given in Fig. 1.

The proposed methodology consists of three phases namely, registration phase, security phase and retrieval phase. In the registration phase, users are registered their information on cloud. In this phase, to avoid the unauthorized person login process, MSA is developed. In the security phase, the data are encrypted using Blowfish algorithm. To enhance the blowfish, the encryption keys are optimally selected using BCSO algorithm. In the retrieval phase, authorized persons are given the request to the server. The user is registered means, they will receive the data otherwise the request will neglect.

4.1 MSA process

The authentication process is crucial to avoid data loss, data theft, and malicious attack. Especially in a centralized environment, unauthorized clients can easily transfer data without the owner's knowledge, meaning that a security breach is inevitable. To avoid this problem, in this paper, MSA is proposed to securely access the data on the cloud. It will protect cloud assets against unauthorized access by enforcing access control mechanisms. The MSA process contains two stages such as registration and login. The detail explanation is given below;

A. Registration process

In the registration stage, clients have entered their information on data centre. At first, the client generates the user id U^{id}, password P^{id}, and entering all the information about the user. After receiving the U^{id} and P^{id}, the server shows N several images to the client. Among the N number of images, the user selects one image I_1. Then, the selected image is sent to the server and stored. Then, to further enhance the authentication, the selected image I_1 is cropped and the cropped area is sent to the server. This also stored on the server. The registration process is given in Fig. 2.

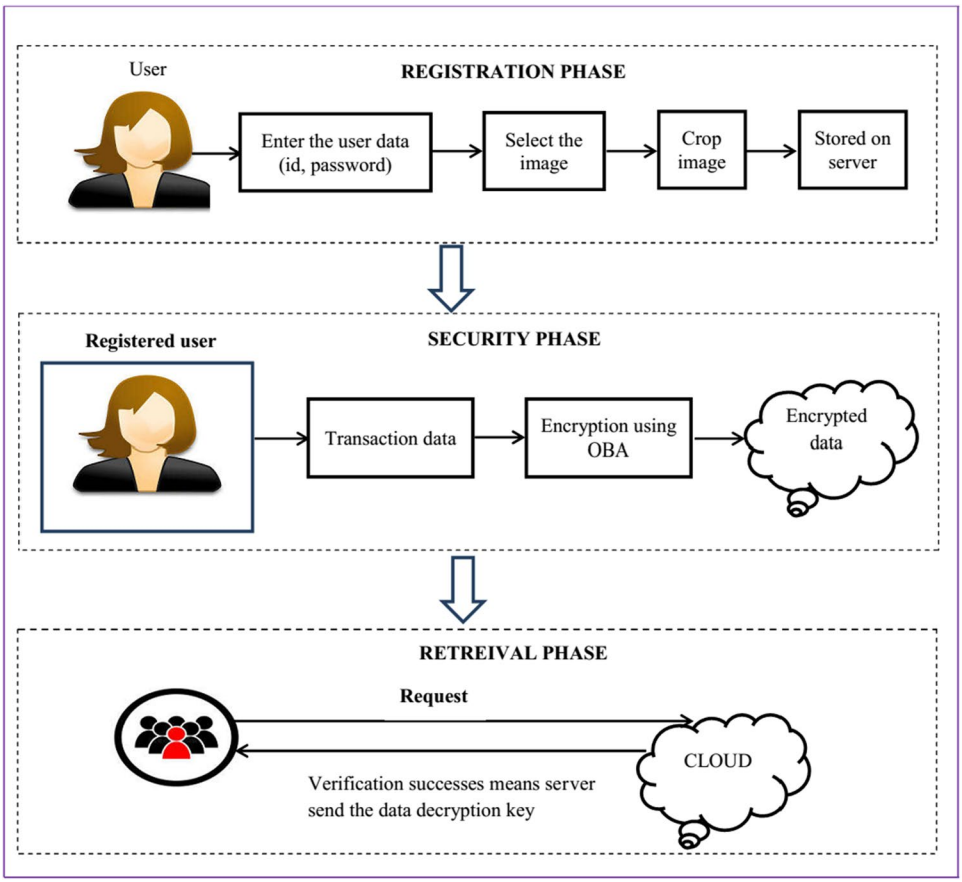

Fig. 1 Overall diagram of the proposed secure data transaction system

B. Login process

In this section, the login process is explained. Once the registration is done successfully, the client can upload or download data to the cloud. Without the registration process, no one client retrieves any information to the cloud. Using this process, we can avoid data loss. For login, at first, clients enter their information such as user id U^{id} and password P^{id}. After receiving the information, the server checks if the given information is correct or not. If it is correct means, the server immediately displays N number of images. The registered image also included in the displayed images. From the images, the client selects one image. This image is the same as the registered image means, the process is continued. Otherwise user request it neglected. After the image selection process, the client crops the same image and sends it to the server. Then, the server checks the similarity between the cropped image and registered crop image. If it is matched means, the server allows the client to access the data, otherwise, they neglect the request.

4.2 Data security using the optimized blowfish algorithm

After the registration process, the input data is encrypted by using an OBA. The Blowfish Algorithm (BA) is the symmetric key cryptography algorithm. The key length of the 64-bit block is 32–448 pieces (Meyers and Desoky 2008). Here, P-array and four 32-bit S-boxes are available. The S-boxes recognize 8-bit information with convey 32-bit yield. BA has two main stages, namely the key expansion and encryption process. For the encryption process, a 16 round fistel network is used. Each round has a main dependency permutation and a key-dependent substitution. All functionality is to add 32-bit words in XOR and BA. The BA structure is given in Fig. 3.

Consider the plaintext value is **123456abcd132536.** The step by step procedure of blowfish algorithm is given as,

- Generate key size
- Initialize sub situation box
- Encryption
- Decryption

Step 1: key size generation

The encryption and decryption process requires 18 sub-keys and the same keys are used for both processes. These 18 sub-keys are stored in an S-array, each element being 32-bit input. It is initialized with the digits of $S_i[x]$. Some examples of hexagonal representation of sub keys are given as,

S [0]: 243f6a88	S [9]: 38d01877
S [1]: 85a368d3	S [10]: be5466cf
S [2]: 13198a2e	S [11]: 34e90c6c
S [3]: 03707244	S [12]: c0ac24b7
S [4]: a4093822	S [13]: c97c50dd
S [5]: 279f31d0	S [14]: 3f87d5b5
S [6]: 082efa98	S [15]: b5470517
S [7]: ec4e6c89	S [16]: 9296d5d9
S [8]: 452821e6	S [17]: 8879fb1b

Each of the sub keys is changed with respect to the input keys.

Step 2: initialize the substitution boxes

Four substitution boxes $\{S^b[1], S^b[2], S^b[3], S^b[4]\}$ are needed in both the encryption and the decryption process

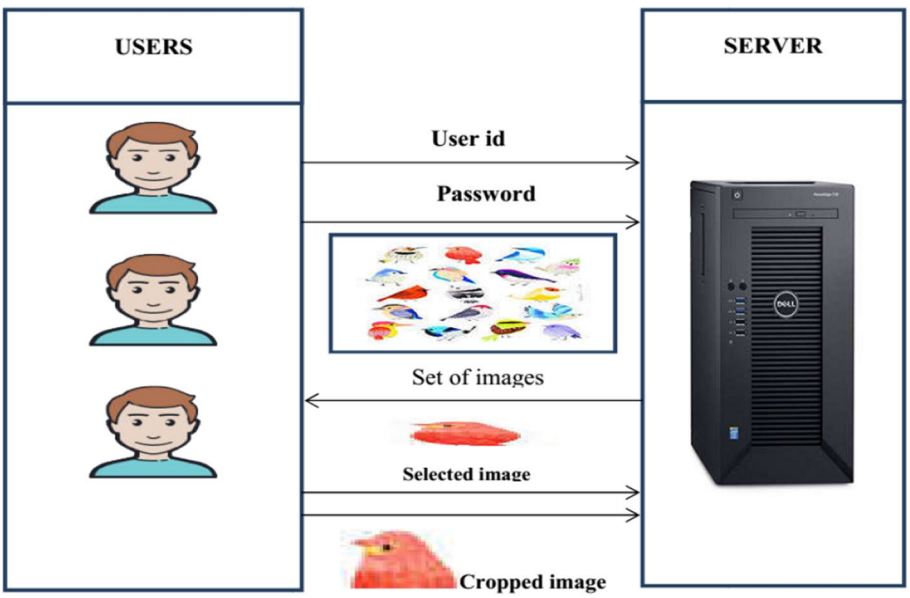

Fig. 2 Registration process

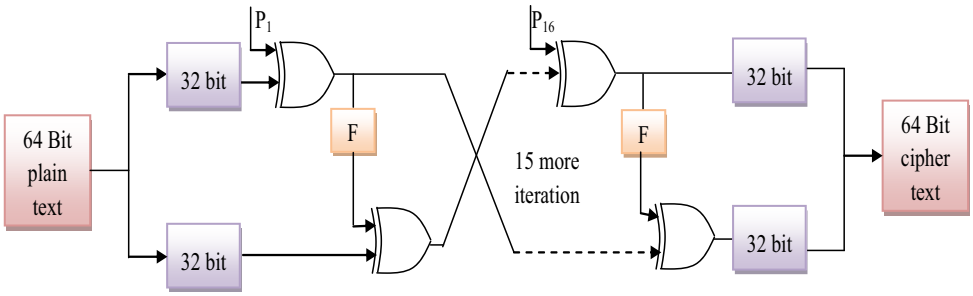

Fig. 3 Blowfish structure

which having 255 entities $\{S^b[i][0], S^b[i][1], \ldots, S^b[i][255]\}$ with 32 bits.

Step 3: encryption

Blowfish has 16 round and the input has a 64-bit data element contains x. Divide x into the 64 halves are xL and xR. Then for, i = 1 to 16.

$$xL = xL\,XOR\,S_i$$

$$x\mathrm{R} = F(xL)\,XOR\,xR$$

Swap xL and xR. After the sixteenth round, swap xL and xR again to undo the last swap. Then $xR = xR\,XOR\,S_{17}$ and $xL = xL\,XOR\,S_{18}$. Finally, recombine xL and xR to get the cipher text. The examples of encryption process is given as,

Round 0: 77b3ba639cb0353b	Round 8: 6af47a4b230745ef
Round 1: 0cc7d63fd7267e6d	Round 9: 9fb82cc57312a5e1
Round 2: c799728ab5655509	Round 10: 1106c1ab8b574312
Round 3: 69612395e3dfcd13	Round 11: 7d7a616509d9011a
Round 4: f3f5b79b67d312af	Round 12: 81e9ce71176d41ca
Round 5: 52023a4efd5c4a46	Round 13: 9727e50g6fa35271
Round 6: 5b785180f097cece	Round 14: eb761e34021839a7
Round 7: cc946d119000f1d4	Round 15: 0599d9367907dbfe

After encryption we get the Cipher text like as "***d748ec383d3405f7***". The encryption process is given in Fig. 4.

Step 4: decryption

The process of decryption using the blowfish algorithm is shown in Fig. 2. In this process, an already encrypted image is decrypted using the same key that was used at the time of encryption. The decryption process is similar to encryption except that in decryption,$\{S[1], S[2], \ldots, S[18]\}$ are used in reverse order. The Decryption values are given below;

Round 17: 3ab5e5667907dbfe	Round 9: 1f04e6309000f1d4
Round 16: fdd297bb021839a7	Round 8: 3624ea12f097cece
Round 15: 82529d676fa35271	Round 7: c546e12ffd5c4a46
Round 14: ec939d1a176d41ca	Round 6: ed76301e67d312af
Round 13: e14063bd02d9011a	Round 5: bbd76433e3dfcd13
Round 12: 66cd65508b574312	Round 4: f160c1f4b5655509
Round 11: 37e82a387512a5e1	Round 3: 251260dd5267e6d
Round 10: 8fe62e7e230745ef	Round 2: 6f86e1389cb0353b

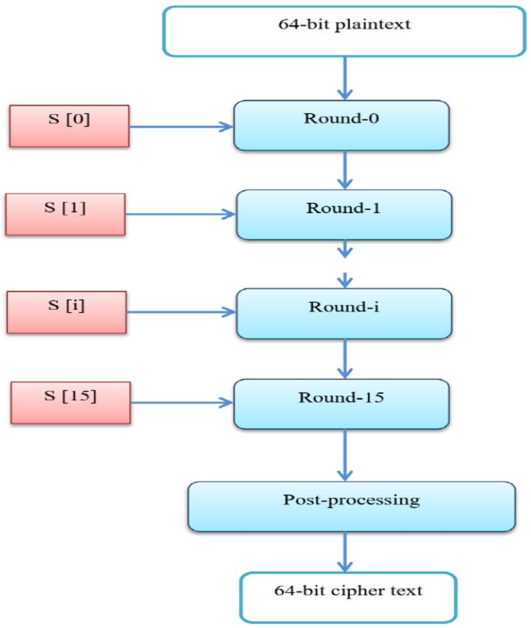

Fig. 4 Blowfish encryption process

Finally, we obtained the Plain text of "***123456abcd132536***". This plain text is same as the original input text. To enhance the BA, the keys values are optimally selected with the help of the BCSO algorithm. The step by step process of key generation is explained below;

4.2.1 Key generation using BCSO

The optimal key selection process is enhancing the blowfish algorithm. For a key generation in this paper BCSO algorithm is utilized. The crow search algorithm (Burger 2001) is one of the late-developed novel meta-heuristic calculations that rely on the behavior and knowledge of crows. The Crow Search algorithm is mainly based on the following principles.

- Crows live in the flock form.
- Crows remember the position of their concealing spots.
- Crows pursue others to do the robbery.

Step 1: solution encoding and key initialization

Initially, the key values are randomly selected and the selected keys are given to the input of the solution encoding process. Here, the solution is represented as crows.

Step 2: fitness calculation

After the solution generation process, the fitness of each solution is calculated. The maximum throughput is considered as fitness. The fitness is given in the below equation.

$$Condition = MAX(Throughput) \tag{2}$$

Step 3: updating using BCSO

In this procedure, we update the solution using the BCSO algorithm after the fitness value is evaluated. In this updating process, two cases are available.

Case 1: Consider the possessor crow *a* and crow b. if, the possessor crow a, without any idea of crow b, moves to the food source B_n^t; in this situation, crow b, follow the possessor crow *a* and reach the food source B_n^t. Based on the food source, crow b updates its position. The Updating function is given in Eq. (3)

$$S_m^{t+1} = S_m^t + R_m \times FL_m^t \times \left(B_n^t - S_m^t\right) \tag{3}$$

where, R_m is an arbitrary number in the range [0, 1], FL_m^t represents flight length of the crow *m* at *t* iteration.

Case 2: If the possessor crow *a*, understand that the crow *b* is following, in this situation, it will change the way. For this situation, the position of crow *a* is refreshed utilizing condition (4).

$$S_m^{t+1} = \begin{cases} if\ R_m \geq P_m^k\ \ update\ position\ using\ (3) \\ else\ \ \ \ \ \ update\ to\ random\ position \end{cases} \tag{4}$$

where P_m^k represent the probability.

In the BCSA, the decision variables are specified in the binary paths (either 1 or 0), and to enable it, the following changes are required on the CSA. A sigmoid transformation $'sig\ t'$ has been introduced to convert the crow's position into a binary form. Its formation is given in the following equation:

$$sig\ t\left(s_m^{t+1}\right) = \frac{1}{1 + \exp\left(-s_m^t\right)}$$

After the sigmoid transformation, the crow position s_m^{t+1} is updated with the help of Eq. (6).

$$s_m^{t+1} = \begin{cases} 1, & if\ R_{sig\ t} < sig\ t\left(c_m^{t+1}\right) \\ 0, & otherwise \end{cases}$$

where $R_{sig\ t}$ is a random number generated between [0, 1].

Step 4: termination criteria

Once we obtain the best fitness function, the iteration will stop. The attained optimal solution is given as the BA.

4.3 Query-based file retrieval phase

Once the encryption process is completed, the encrypted data is stored in the cloud. After the storage process, if any user wants to access the data, initially, they must send a request to the CSP. In the CSP, the MSA is performed, in which user information will be checked by the processor. If the given information is correct, the user is allowed access to the data. Otherwise, the request is ignored.

5 Results and discussions

The test results obtained from the proposed method are analyzed in this section. The proposed methodology is implemented in Java on a machine with the Intel Core processor running at 3.00 GHz and Windows 7 Ultimate running on 4 GB RAM. Tests were performed on a cluster of 5 engines on the Amazon EC2. All virtual machines used in the cluster are M3 General Purpose Extra Large with 15 GB of memory, 20 EC2 compute units (4 virtual cores with 3.25 units each), (2 × 40 GB) local storage. All 5 engines in the cluster are connected to each other by a gigabit lane. M3 General Purpose Extra large cost event-hour 50 0.50. An EC2 compute unit provides the CPU capacity of a 1.0–1.2 GHz 2007 Optron or 2007 Zion processor. The transaction load and the number of users are simulated using multiple threaded requests. The performance of the proposed method is analyzed based on different metrics, namely memory usage, activation time, encryption, and encryption time shows in Table 1.

5.1 Performance analysis

The performance of the proposed methodology is analyzed in terms of retrieval time, file access time, encryption time, decryption time, and memory. The main purpose of the proposed method is to transmit data to the cloud securely and without losing any information. To attain the objective, a novel MSA process and cryptography algorithm is utilized. The performance of the proposed methodology is given in the below figures.

The performance of the proposed methodology is analyzed in terms of file access time is given in Fig. 5. Here, the x-axis represents the file size and the y-axis represents the time. When analyzing Fig. 5, the proposed method takes 75 ms as the minimum time to access the 10 kb file. Also, as file size increases, image access time increases

Table 1 Experimental setting

Number of machines	Environment	Description
5	CPU	M3 general purpose extra large, (4 core × 3.25 unit)
	Memory	15 GB
	Storage	(2×40 GB)SSD
	OS	Windows 8
	.NET Framework	4.0
	NO SQL Database	Amazon Simple DB

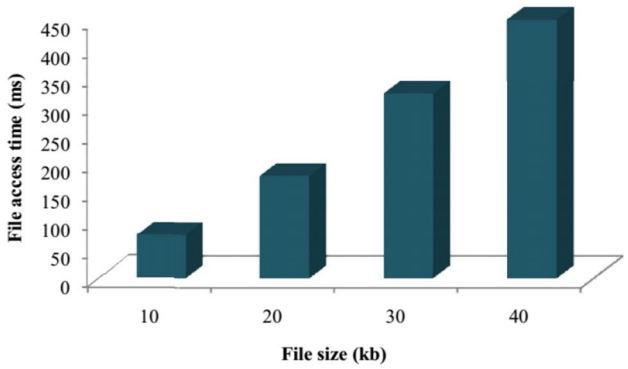

Fig. 5 Performance of the proposed method based on file access time

gradually, which is clearly understood from the image. Similarly, in Fig. 6, the performance of the proposed methodology is analyzed in terms of encryption time, and Fig. 7 performance of the proposed methodology is analyzed in terms of decryption time. Here also proposed method took minimum for encryption and decryption process. In Fig. 8, the performance of the proposed method is analyzed in terms of memory. In this case, as the file size increases, memory usage gradually increases. When analyzing Fig. 8, it takes 15,359,766 bits of memory to send 40 kb of data. It is only a very small amount. In Fig. 9, the file size extracted for the proposed method is analyzed. Note that the original size is 5 kb, 10 kb, 15 kb, and 20 kb. After encrypting the file, the file size is originally 5 kb, but after encryption, the size is changed to 8 kb. After encoding 13 kb, 19 kb, and 24 kb, the file size is replaced by 10 kb, 15 kb, and 20 kb. Then, the encrypted file is stored in the cloud. To view the document, you must encrypt the document after downloading it. After the decryption process, the size of the file is converted into the original file size.

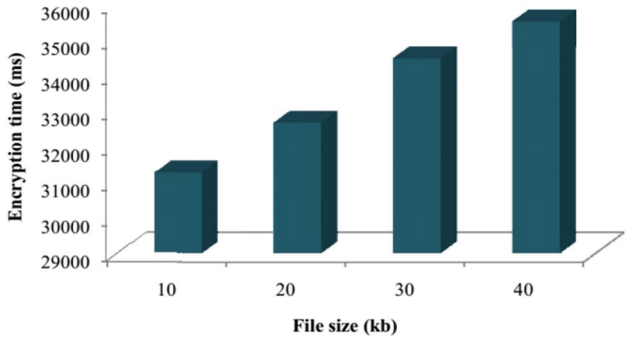

Fig. 6 Performance of the proposed method based on encryption time

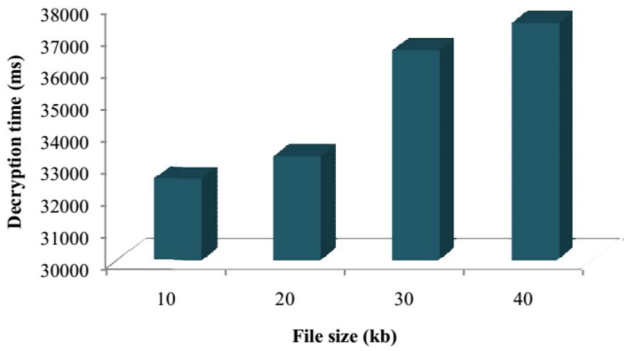

Fig. 7 Performance of the proposed method based on decryption time

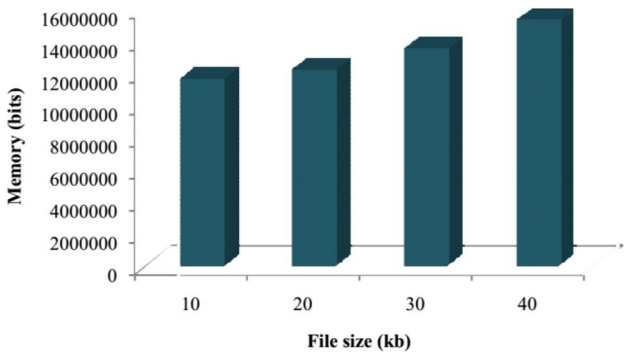

Fig. 8 Performance of the proposed method based on memory

5.2 Comparative analysis

To prove the effectiveness of the proposed methodology, the proposed algorithm compared with different state of art methods namely, (Raghul et al. 2015; Rohini and Shinde 2015; Kanna and Vasudevan 2019), DES based

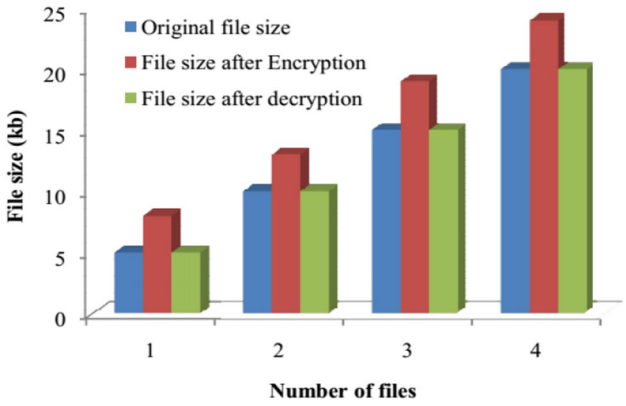

Fig. 9 File size taken for the proposed method

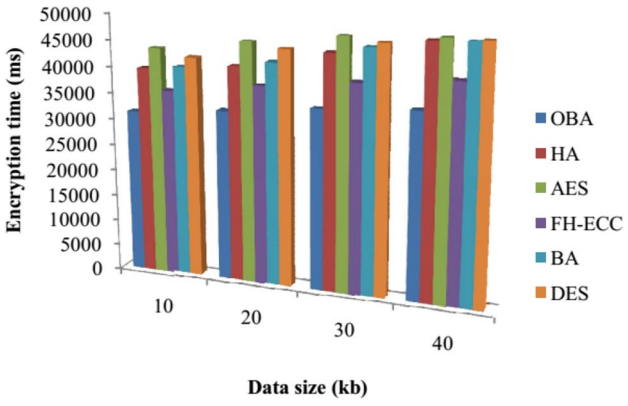

Fig. 11 Comparative analysis based on encryption time

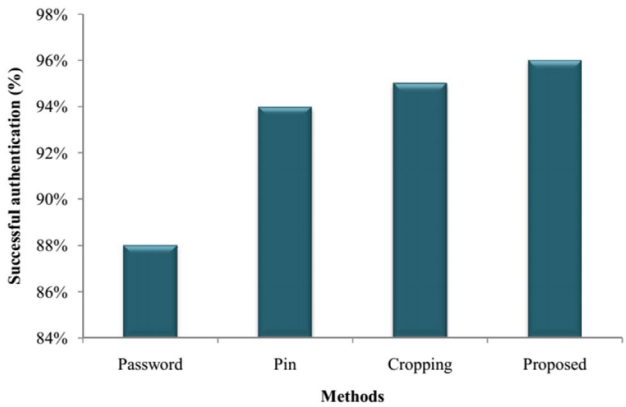

Fig. 10 Performance-based on successful authentication

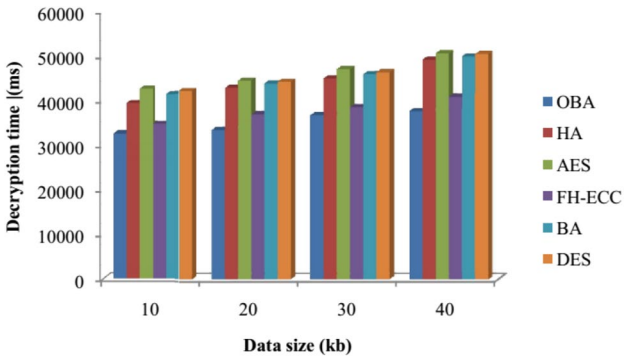

Fig. 12 Comparative analysis based on decryption time

data security and BA based data security. In (Rohini and Shinde 2015) the data are encrypted using AES algorithm. In Raghul et al. (2015), the homomorphic algorithm (HA) is used for the security process. Similarly, in Kanna and Vasudevan (2019), hybridization of fully homomorphic and electrical curve cryptography (FH-ECC) is introduced for security process.

To prove the effectiveness of the proposed authentication process, we compare our method with a different authentication process. The performance of the proposed methodology is analyzed in terms of successful authentication is given in Fig. 10. When analyzing Fig. 10, our proposed method attains a better authentication rate. This because, our proposed method has four mandatory fields namely, user id, password, image selection, and cropping. Moreover, comparative analysis of proposed against existing method in terms of encryption time by varying data size is given in Fig. 11. When analyzing Fig. 11, our proposed method attains the minimum time of 31254 ms for encrypting 10 kb file which is 39645 ms for using HA based encryption (Raghul et al. 2015), 35674 ms for using FH-ECC (Kanna and Vasudevan 2019), 40,435 for BA based encryption, 43675 ms for using AES based data encryption (Rohini and Shinde 2015), and 42312 ms for DES based data encryption. Similarly, the performance of proposed against the existing method is analyzed in terms of decryption time is given in Fig. 12. Here, also our proposed method takes minimum time to retrieve the encrypted data compared to the existing method.

Analysis of the performance of encryption time by different transactions is given in Fig. 13. As we can see in Fig. 13, it becomes clear that the encryption time also increases as the number of transactions increases. Furthermore, when analyzing Fig. 13, the proposed method achieves a minimum encryption time compared to other methods. The performance analysis in terms of decryption time by a varying number of transactions is given in Fig. 14. To prove

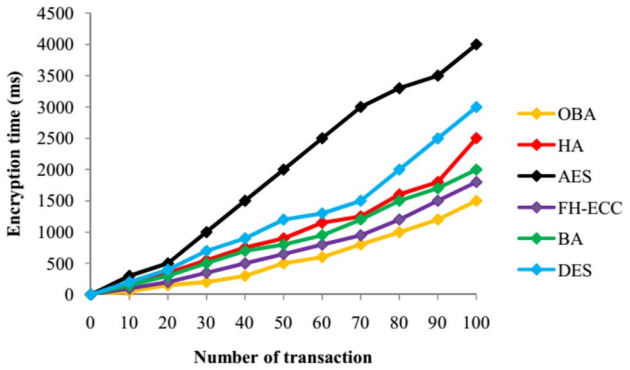

Fig. 13 Number of transaction vs. encryption time

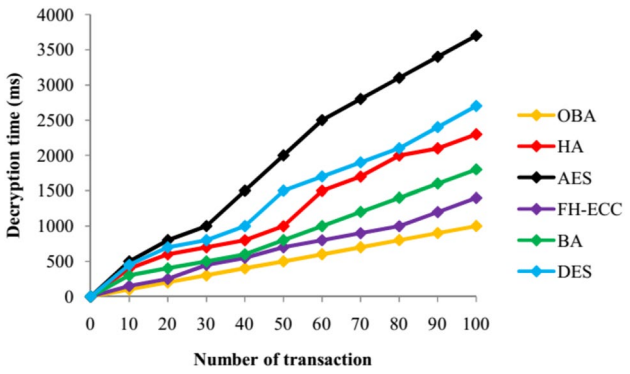

Fig. 14 Number of transaction vs. decryption time

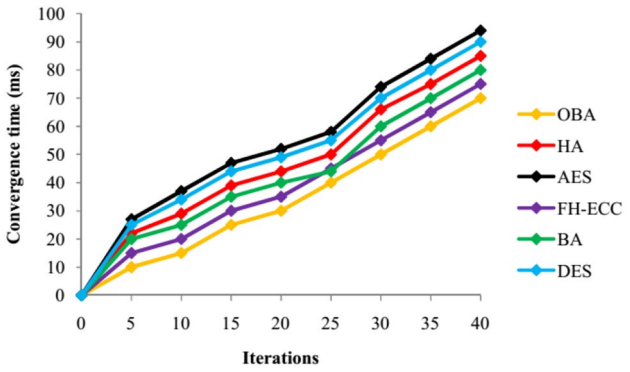

Fig. 15 Convergence time comparison for proposed vs. existing methods

the effectiveness of the proposed method, the existing two algorithm performances are compared with the proposed method. When analyzing Fig. 14, our proposed algorithm

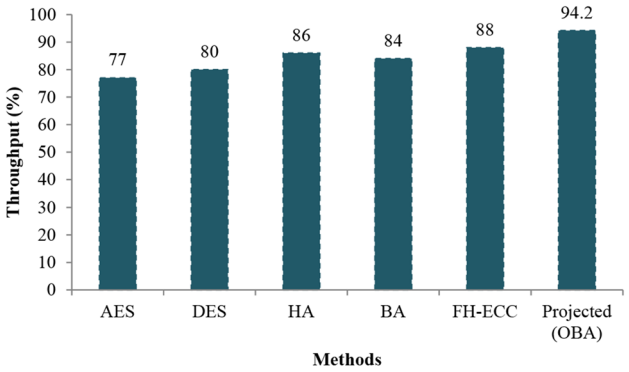

Fig. 16 Throughput comparison for proposed vs. existing methods

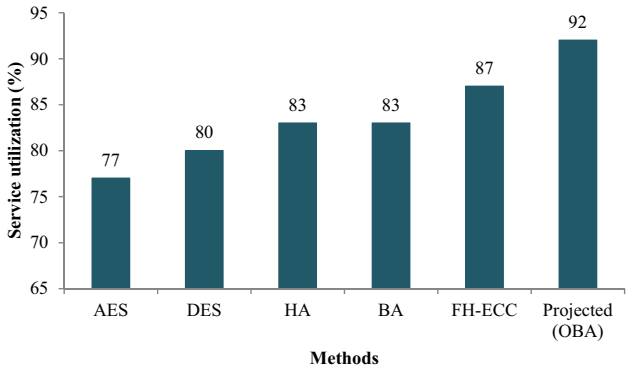

Fig. 17 Service utilization comparison for proposed vs. existing methods

takes minimum time to decrypt the file. In this method, the BCSA algorithm enhances the performance of the proposed method. In Fig. 15, the convergence time is analyzed. Moreover, in Fig. 16 the performance of proposed approach is analyzed using throughput. The throughput is measured based on percentage of packets successfully delivered to receiver or destination. It is measured in data packets per second or bits per second (bps). From the graph, the projected method obtained the maximum throughput of 94.2% which is 88%, 84%, 86%80% and 77% for FH-EC based secure data transaction, BA based secure data transaction, HA based secure data transaction, DES based secure data transaction and AES based secure data transaction respectively. Due to BCSA based optimal key selection process, the projected method attained the maximum throughput. Similarly, in Fig. 17 the performance of proposed approach is analyzed using service utilization. Service utilization refers to the use of the service used for a given number of

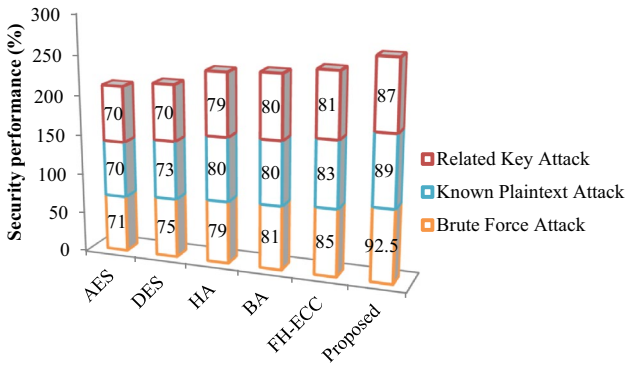

Fig. 18 Comparative analysis based on security level

requests. From the table, projected approach utilized the services 92%.

The security level of the proposed algorithm is measured are given in Fig. 18. Security level is based on the number of packets sent by the source side throughout the simulation and the total number of packets successfully received by the target side. When analyzing table 5.5, proposed method attained the maximum security level of 92.5% for Brute force attack, 89% for known plaintext attack and 87% for related key attack. After applying attack also, proposed method has the high security. This is due to MSA and key selection process. The successful authentication and authentication fail screenshot are given in Figs. 19 and 20 respectively. It is obvious from all of the above findings that the technique suggested is better than existing approach.

6 Conclusion

In this paper, a secure aware data transaction on the cloud has been explained. Here, user authentication verification is an important task that has been explained. Here, the data has been encrypted using the OBA algorithm. The BA has been enhanced using the BCSO algorithm. The mathematical expression of both the algorithm has been explained. Using this method, we can avoid the unauthorized user login process. The performance of the proposed methodology has been analyzed in terms of different metrics namely, encryption time, decryption time, file access time and memory. Our suggested technique has less encryption and decryption time than the current technique, as shown by experimental outcomes. Hence our proposed method is highly preferable than the existing methods. The user cannot retrieve the file without authentication verification hence this method is highly secured. In future efficient algorithms could be used for increasing the speed of the overall process.

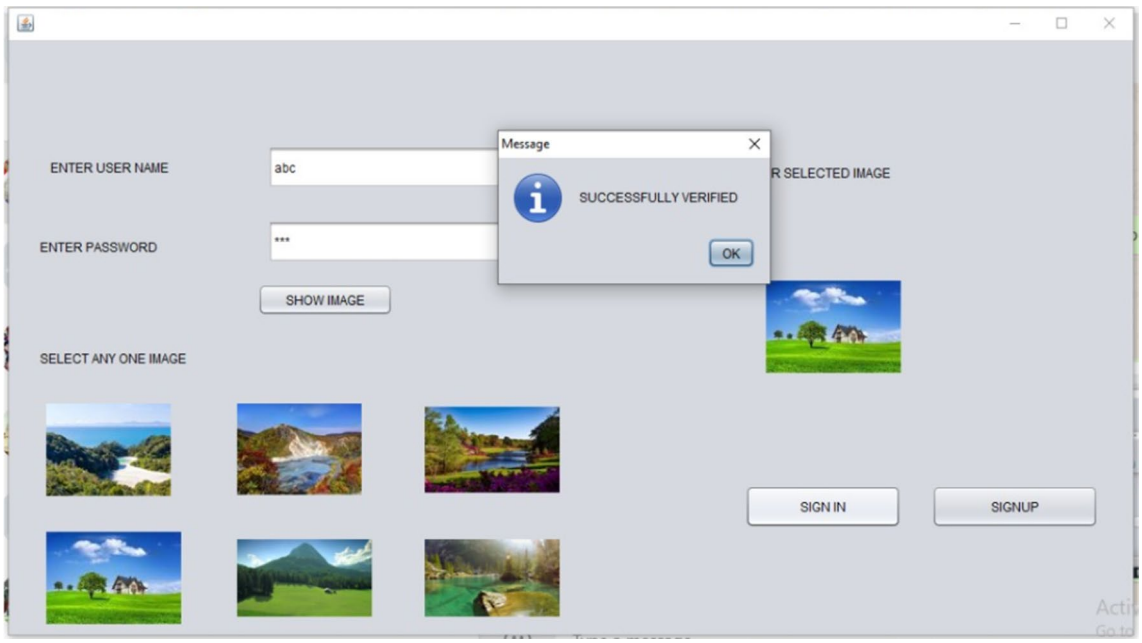

Fig. 19 Screenshot for successful verification

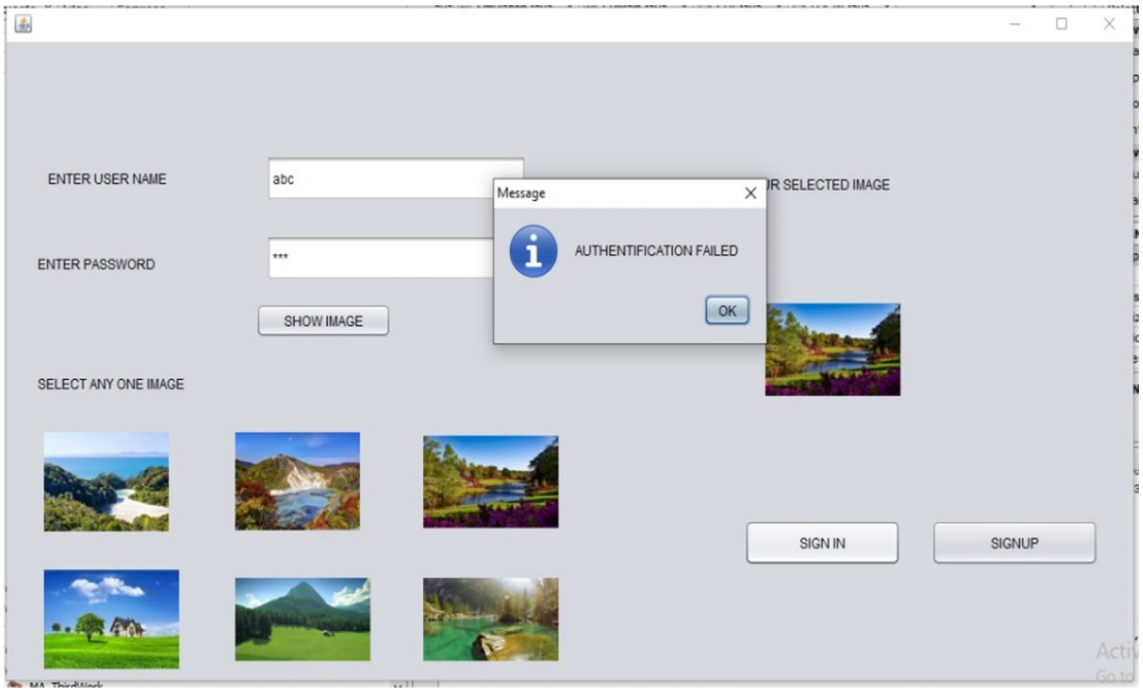

Fig. 20 Screenshot for authentication fail

References

Bai TDP, Raj KM, Rabara SA (2017) Elliptic curve cryptography based security framework for internet of things (IoT) enabled smart card. In: 2017 World congress on computing and communication technologies (WCCCT) IEEE, pp 43–46

Brindha T, Shaji RS (2018) A secure transaction of cloud data using conditional source trust attributes encryption mechanism. Soft Comput 22(3):1013–1022

Burger PM (2001) Biometric authentication system. U.S. Patent 6(219): 439

Cheng H, Rong C, Qian M, Wang W (2018) Accountable Privacy-preserving mechanism for cloud computing based on identity-based encryption. IEEE Access 6:37869–37882

De la Prieta F, Rodríguez-González S, Chamoso P, Corchado JM, Bajo J (2019) Survey of agent-based cloud computing applications. Future Gener Comput Syst 100:223–236

Delfs H, Knebl H, Knebl H (2002) Introduction to cryptography, vol 2. Springer, Heidelberg

Dinesha HA, Agrawal VK (2012) Multi-level authentication technique for accessing cloud services. In: 2012 international conference on computing, communication and applications IEEE, pp 1–4

Guo J, Li C, Zhang G, Sun Y, Bie R (2019) Blockchain-enabled digital rights management for multimedia resources of online education. Multim Tools Appl 1–21

Helmi AM, Farhan MS, Nasr MM (2018) A framework for integrating geospatial information systems and hybrid cloud computing. Comput Electr Eng 67:145–158

Kang HS, Son JH, Hong CS (2015) Defense technique against spoofing attacks using reliable ARP table in cloud computing environment. In: 2015 17th Asia–Pacific network operations and management symposium (APNOMS) IEEE, pp 592–595

Kanna GP, Vasudevan V (2019) A fully homomorphic–elliptic curve cryptography based encryption algorithm for ensuring the privacy preservation of the cloud data. Cluster Computing 22(4):9561–9569

Kumar M, Sharma SC, Goel A, Singh SP (2019) A comprehensive survey for scheduling techniques in cloud computing. J Netw Comput Appl 143:1–33

Kushida T, Pingali GS (2014) Industry cloud-effective adoption of cloud computing for industry solutions. In: 2014 IEEE 7th international conference on cloud computing IEEE, pp 753–760

Li H, Dai Y, Tian L, Yang H (2009) Identity-based authentication for cloud computing. In: IEEE international conference on cloud computing Springer, Berlin, pp 157–166

Meyers RK, Desoky AH (2008) An implementation of the Blowfish cryptosystem. In: 2008 IEEE international symposium on signal processing and information technology IEEE, pp 346–351

Mohiuddin I, Almogren A, Al Qurishi M, Hassan MM, Al Rassan I, Fortino G (2019) Secure distributed adaptive bin packing algorithm for cloud storage. Future Gener Comput Syst 90:307–316

Mubarakali A, Bose SC, Srinivasan K, Elsir A, Elsier O (2019) Design a secure and efficient health record transaction utilizing block chain (SEHRTB) algorithm for health record transaction in block chain. J Ambient Intell Humaniz Comput 1–9

Pournaghi SM, Bayat M, Farjami Y (2020) MedSBA: a novel and secure scheme to share medical data based on blockchain technology and attribute-based encryption. J Ambient Intell Humaniz Comput 1–29

Pragaladan R, Sathappan S (2018) A secure cloud data storage combining DNA structure and multi-aspect time-integrated cut-off potential. In: Advances in big data and cloud computing. Springer, Singapore, pp 361–374

Raghul H, Ramagopal RN, Saravanan B, Guhapriya T, Anitha R (2015) Data security in federated cloud environment using homomorphic encryption technique. Int J Emerg Technol Adv Eng 5(4):137–141

Rajani S, Ghorpade V, Dhange M (2016) Multi-factor authentication as a service for cloud data security. Int J Comput Sci Eng 4:43–46

Ramya B, Kokila MS, Hemalath S (2016) Secure public cloud using DES algorithm. Int J Adv Res Comput Eng Technol 5(11)

Reddy BT, Chowdappa KB, Reddy SR (2015) Cloud security using Blowfish and key management encryption algorithm. Int J Eng Appl Sci 2(6)

Resende ACD, Mochetti K, Aranha DF (2015) PUF-based mutual multifactor entity and transaction authentication for secure banking. In: Lightweight cryptography for security and privacy, pp 77–96

Rohini V, Shinde VD (2015) Secure role-based access control on encrypted data in cloud storage using raspberry PI. Int J Multidiscip Res Dev 2(7):20–27

Sachdev A, Bhansali M (2013) Enhancing cloud computing security using AES algorithm. Int J Comput Appl 67(9)

Somani U, Lakhani K, Mundra M (2010) Implementing digital signature with RSA encryption algorithm to enhance the data security of cloud in cloud computing. In: 2010 first international conference on parallel, distributed and grid computing (PDGC 2010) IEEE, pp 211–216

Sudhakar RV, Rao TCM (2020) Security aware index based quasi–identifier approach for privacy preservation of data sets for cloud applications. Clust Comput 1–11

Sumathi M, Sangeetha S (2020) A group-key-based sensitive attribute protection in cloud storage using modified random Fibonacci cryptography. Complex Intell Syst 1–15

Sundarakumar MR, Mahadevan G (2019) Authorization for secured cloud storage through SHA-256

Suresha K, Karthick PV (2020) Enhancing data security in cloud computing using threshold cryptography technique. In: Advances in cybernetics, cognition, and machine learning for communication technologies. Springer, Singapore, pp 231–242

Thangavel M, Varalakshmi P (2018) Enhanced DNA and ElGamal cryptosystem for secure data storage and retrieval in cloud. Cluster Comput 21(2):1411–1437

Tsai CH, Su PC (2020) The application of multi-server authentication scheme in internet banking transaction environments. Inf Syst e-Bus Manag 1–29

Vafamehr A, Khodayar ME (2018) Energy-aware cloud computing. Electric J 31(2):40–49