A Hybrid Secure Cloud Platform Maintenance Based on Improved Attribute-Based Encryption Strategies

Abhishek Kumar[1]*, Swarn Avinash Kumar[2], Vishal Dutt[3], Ashutosh Kumar Dubey[1], Sushil Narang[1]

[1] Chitkara University School of Engineering and Technology, Chitkara University, Himachal Pradesh (India)
[2] Indian Institute of Information Technology, Allahabad, UP (India)
[3] Department of Computer Science, Aryabhatta College, Ajmer (India)

Received 3 April 2021 | Accepted 27 August 2021 | Published 12 November 2021

Abstract

In the modern era, Cloud Platforms are the most needed port to maintain documents remotely with proper security norms. The concept of cloud environments is similar to the network channel. Still, the Cloud is considered the refined form of network, in which the data can easily be stored into the server without any range restrictions. The data maintained into the remote server needs a high-security feature, and the processing power of data should be high to retrieve the data back from the respective server. In the past, there were several security schemes available to protect the remote cloud server reasonably. However, the attack possibilities over the cloud platform remain; only all the researchers continuously work on this platform without any delay. This paper introduces a hybrid data security scheme called the Improved Attribute-Based Encryption Scheme (IABES). This IABES combines two powerful data security algorithms: Advanced Encryption Standard (AES) and Attribute-Based Encryption (ABE) algorithm. These two algorithms are combined to provide massive support to the proposed approach of data maintenance over the remote cloud server with high-end security norms. This hybrid data security algorithm assures the data cannot be attacked over the server by the attacker or intruder in any case because of its robustness. The essential generation process generates a credential for the users. It cannot be identified or visible to anyone as well as the generated certificates cannot be extracted even if the corresponding user forgets the credentials. The only way to get back the certification is resetting the credential. The obtained results prove the accuracy level of the proposed cypher security schemes compared with the regular cloud security management scheme, and the proposed algorithm essential generation process is unique. No one can guess or acquire it. Even the person may be the service provider or server administrator. For all, the proposed system assures data maintenance over the cloud platform with a high level of security and robustness in Quality of Service.

Keywords

Advanced Encryption Standard (AES), Attribute-Based Encryption (ABE), Cloud Security, Improved Attribute-Based Encryption Scheme (IABES).

DOI: 10.9781/ijimai.2021.11.004

I. Introduction

CLOUD Computing environments are supporting users to manage their data globally over the remote server with the high end of security. The cloud server processes data remotely and provides the resulting features to the client port without any hurdles.

Many research papers illustrated that cloud computing environments are highly secure and robust in their performance and cipher policies [1],[2],[3],[4]. However, the problems in the cloud environment usually sustain until now. The issues are growing every day, and the researchers are identifying many new mechanisms day by day to tackle these security issues [5], [6].

The cloud computing environment requires a new methodology to avoid security issues and provide a high-level security measure to the proposed approach in an exemplary manner without any interventions. This paper introduces a new hybrid methodology, which integrates the two best algorithms and operates the proposed cloud server system accordingly, called Improved Attribute-Based Encryption Scheme (IABES).

This proposed algorithm combines two powerful algorithms such as Advanced Encryption Standard (AES) and the Attribute-Based Encryption (ABE) Mechanism. The concept of AES follows the Rijndael process, which is formed as a cipher-block with a 256-bit encryption technique. Each block is divided into 128-bit capacity with associated essential space. This is one of the powerful crypto algorithms usually followed over many real-time applications such as banking, mobile applications, etc. The next one is called ABE, in which the algorithm is operating based on the attributes and process the data accordingly based on cipher keys [7], [8], [9].

This is also a robust security principle, allowing users to maintain the data into the server end without any hurdles. But instead of keeping the public crypto keys, in this ABE approach, a new essential generation standard is followed based on user input attributes and based on that input attributes. The keys are generated, and the input data or document is encrypted [10], [11], [12]. So, that the encryption

* Corresponding author.

E-mail address: abhishek.kumar@chitkara.edu.in

standards are highly unique with such systems [13], [14], [15]. There are several approaches which has been covered the scenario of different ciphertext policies and other aspects for dealing the problems in different types of security system [16], [17], [18], [19].

However, the individuality of above mentioned two algorithms are working fine, but in the case of higher-end security threats, both of these algorithms struck up into a specific range [5], [8], [10]. So, that a new algorithm is designed based on the efficiency of the two separate algorithms, such as ABE and AES, and named the hybrid algorithm as IABES. It adapts the benefits of mentioned two algorithms and provides the ultimate security features over the proposed cloud server management system. The submitted paper is intended to make the new algorithm concentrate on security threats concerning different attack possibilities such as the Query-Regeneration attack, Query Modification attack, Searchable-Query attack and the Query-Removal attack. These different kinds of security threats are handled adequately over the proposed system with an advanced cipher handling algorithm. These attacks are coming under the SQL Injection attack category, which will be illustrated in detail below.

A. Query-Regeneration Attack

The attackers or intruders usually attack the server from the client end only and generate a query to regenerate multiple data over the standard table presented into the server end. For example, the table contains ten numbers of records with different unique identities; this kind of Query-Regeneration attack creates duplication over the proposed system server end, which will automatically degrade the performance of the entire server management system. The attackers usually try these kinds of attacks to copy the whole server data and place it again into the weak node presented into the network.

B. Query Modification Attack

The attackers try to modify the data available into the server using Query Modification logic. The data presented into the server must be integrity enabled and robust against multiple scenarios of attacks. But in the regular cloud server maintenance system, the usual attack is called a query modification attack. Consider the design of government organization; if the quotation is raised for some commercial contract, if the attackers modify the quoted amount, the complete reference gets spoiled. These kinds of attacks are presented based on modification attack over the server. It is considered one of the most dangerous query attacks in the information technology industry.

1. Searchable-Query Attack

The attackers not only try to attack the data or document presented over the remote server instead attempted to view the records submitted over the server without having any access control norms and proper credentials. For some of the weak cloud servers, the attacker can easily surf and get the records without the knowledge of the respective data owner. This kind of attack is usually raised to identify an individual's personal or official details and target the corresponding individual based on private information. This kind of attack is also crucial to concentrate more on it over the proposed data handling approach over the proposed system.

2. Query-Removal Attack

This kind of Query Removal attack is composed to remove the data presented into the server, which causes some severe reflections over the cloud server management scheme. Because of the removal, the entire trust over the server will lack, and this removal problem raises many legal issues in the industry. For example, if the organization maintains the employee salary records into the server means, the documents need to be robust in all ways. Suppose any intruder removes the basic pay of all employees in the server or deletes all employee records over the server means. In that case, entire operations are collapsed on the company and employees facing massive trouble. This has also happened in most server mediums; that is why all are periodically back up the server with some proper intervals.

All these issues are handled using our proposed approach of Cloud-based data maintenance concerning Improved ABE Scheme IABES. This proposed algorithm has taken care of all these mentioned injection attacks, provides the problem accessible server to the users in a suitable manner, and provides the high-level security threats elimination mechanism to real-world cloud servers.

The significant motivation behind this study is that, in the modern era, the cloud computing platform is being used at a very high level. In today's technology era, the importance of data is immense. Given the increased importance of data, its security needs to be taken very seriously. Even though people are using cloud storage in abundance, there are still many apprehensions regarding the safety of the data, which proves to be an important and significant reason for reducing the use of cloud systems. And due to this, the use of cloud storage remains limited and compressed.

The main objective of this research is to provide a high degree of security to the data stored in the cloud storage. So that the data can be protected from attackers or intruders and its use can be promoted. The processing power of the data must be high to retrieve the data from the respective server. Hence, managing data access time is also an essential part of cloud storage systems to maintain the meaningfulness of data availability and data security. To provide tight security to the data without compromising on the processing power and provide seamless access to the data is the study's main objective.

The rest of this paper have been arranged in the following manner: Section II illustrates the proposed system methodologies in detail with proper algorithm flow. Section III demonstrates the result and discussion portion of the paper, and the final section, Section IV, illustrates the concept of conclusion and future scope.

II. Proposed System

It is difficult to find out the intruders and trace them over the digital world in this modern era. The security mechanisms available nowadays provide acceptable security norms to the clients to preserve their data safely. But the consistency and stability of such security mechanisms are still raising an issue to manage the data integrity over different levels. The proposed system is intended to provide an efficient data security and integrity maintenance scheme, which will be suitable for all kind of textual data maintenance over the cloud server in an intelligent manner.

A. Components of IABES

- User: It is an entity that is going to consume the services of cloud storage.
- Authentication: This is an essential component in architecture. Through this, the identity of the participant or user is checked by the system. If a user wants to join this system and go for its services, he has to first go through the proposed authentication process. He has to prove his identity that he is an Authorized and Authentic user.
- Text Uploading: As soon as the system confirms that the user is authentic, the user gets permission to upload the data.
- Secure Data: When the data is uploaded, then the process of securing the information is started. In this process, encryption and decryption is an important addition. Some algorithms for encryption and decryption have been proposed in this study.
- Cloud server: The cloud server is included as an essential

component in this architecture. The data uploaded by the user will be stored in encrypted form on the cloud server itself so that the user can access the data anywhere and anytime via an internet connection. Just as the uploading of data has to go through authentication to access the data, the user must also follow this authentication process.

The significant contributions of the proposed algorithm IABES are as follows:

- In this proposed approach, the attackers or intruders who share their unique secret keys to others, whatever may be the purpose, needs to be traceable from our proposed logic. The present system needs to generate random access to extract some portion of the user identities and develop a new secret key so that the generated private key cannot be guessed or identified by the attackers in any case.
- The proposed approach of secure hashing allows the user to generate dynamic user credentials (refer to Fig. 1.) concerning the user's identity and the random key generation process. With these associations, a new dynamic credential is generated, and that will be forwarded to the user mail with decrypted mode. The respective user can only get to know the credential ultimately until and unless without the user knowledge. It won't be shared with anyone, and this credential cannot be breakable by anyone because we know that the secure hashing technique is a unidirectional encryption scheme using this SHA based data hashing and storing those credential values to the server so that the server administrator cannot retrieve the credentials from the server. Since each characteristic of the clients is analyzed based on the tracing values stored on the server end when the feature contained in the key satisfies the access control norms of the server, that would be able to be decoded effectively, and the particular user only can access all the features of the proposed approach. Contrasting and the related detectable ABE Scheme, it is of functional significance to present the idea of the secret key into the noticeable proposed system so that our proposed method is nearer to the genuine circumstance.
- Under the suspicion of the proposed approach, the developed hybrid algorithm of IABES is demonstrated to enhance protection from the plaintext attack in the standard model, and the trial results show that the proposed scheme of IABES is viable in the cloud condition. The proposed approach, Improved ABE Scheme (IABES), is intended to provide security and proper access control norms over the cloud server with the help of the following procedures: User Attribute Segregation Secret Key Generation Process, Encryption Process and Decryption Process. All these processes are described below. The architectural view of the proposed system is shown in Fig. 2.

B. User Attribute Segregation

This attribute segregation process as illustrated in Fig. 3., gathers the user attribute such as name, mobile number and email-id from the respective user and process the collected data with the segregation principle. For example, the User X identity is grasped and segregates the required features from that collected attribute employing data split logic over the proposed approach. The collected attributes are used to generate the dynamic credential of the user. This logic is used only to extract the attribute from the user identities, which is sufficient for the credential; instead, it generates a robust cloud network credential for further access. The process is shown in Algorithm 1.

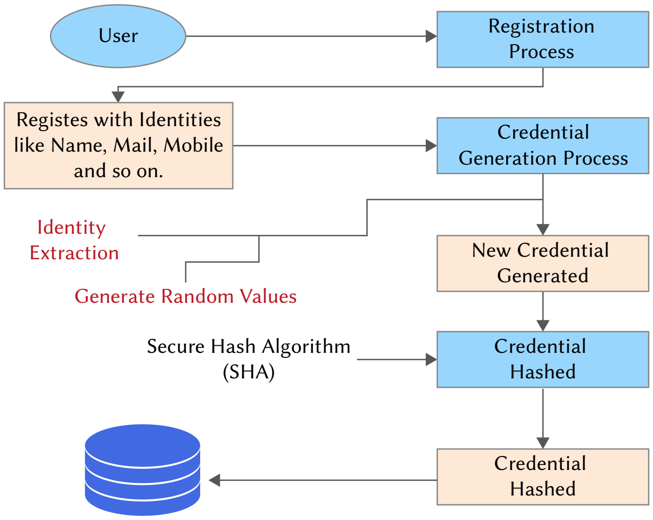

Fig. 1. Secured Credential Generation Process using SHA.

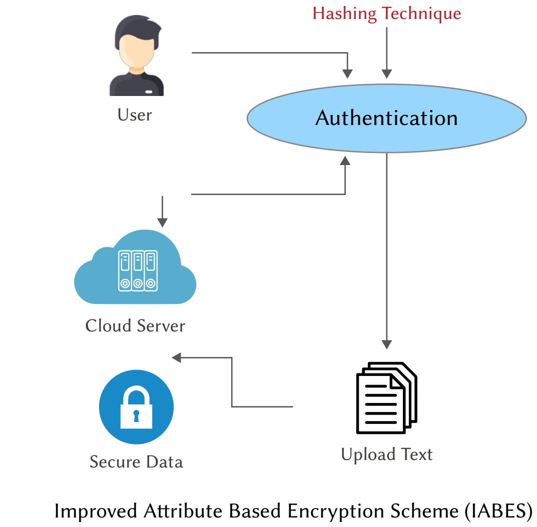

Fig. 2. IABES Architectural View.

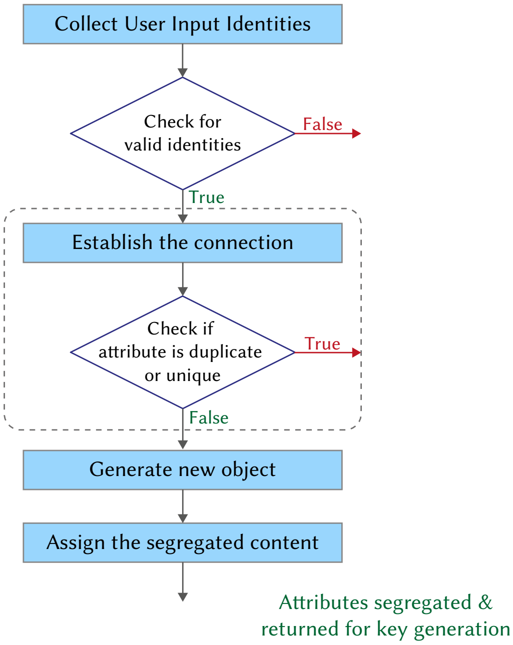
Collect User Input Identities
Check for valid identities
False
True
Establish the connection
Check if attribute is duplicate or unique
True
False
Generate new object
Assign the segregated content
Attributes segregated & returned for key generation

Fig. 3. Work flow of user segregation Process.

```
**Algorithm 1**: User Attribute Segregation
Input: User Attributes (Name, Mobile Number, Mail-ID and Contact)
Output: Segregated Portion of User Credential.
Step-1: Collect user input identities from the client end web portal.
    Input = $id_c, id_{wp}$
Step-2: Check for valid identities. In this step, the accumulated information from Step-1 is validated and allowed the user once the given identities are correct.
    $Validate(Input(id_c, id_{wp}))$
Step-3: Establish the connection between Client end and the Server end. Check whether the given attribute is duplicate or unique.
    $Conn(Client,Server))$
    $Obj_i == Input\,? \rightarrow return(True) : Obj_{i+1} = Input\,?$
                        $\rightarrow return(True) : Obj_{i+n} = Input\,?$
                        $\rightarrow return(True) : return(False)$
Step-4: Once the return statement returns false means, the identity is unique.
Step-5: Generate new object for String Segregation $Obj_{Str}$
Step-6: Assign the segregated content of the user identity to created string object.
    $\mathrm{Obj}_{\mathrm{str}} = \sum_{i=0}^{n} I(1,2, \ldots.., n)$
Step-7: User attributes segregated and returned to further process of key generation.
    $Sua \xrightarrow{Generates} P(Key_{Generation})$
Here 'S' denotes Segregated attributes, and 'P' denotes the process.
```

C. Secret Key Generation and Processing

The secret key generation process is dependent on the Attribute Segregation process over the proposed approach of IABES. The processed attributes from the user attribute segregation scheme are collected over this approach as an input and generate the random key based on the RandomClass function and merge the created random key with the already segregated user attribute. So, that the generated secret key is ultimately vital to compare to any other traditional approaches. No one can judge this kind of secret keys, or it cannot be guessable to others. As well as the created secret key is not only enough for authentication, because of providing high-level security norm, the dynamic one-time password will be generated and send to the respective users' mail-id after verifying the credentials given by the user. The system allows the user to proceed further once the given high secured one-time password is correct; otherwise, it blocks the user to proceed further. It is shown in Algorithm 2.

$$n = pq \ \varphi(n) = (p - 1)(q - 1)$$

$$e, 1 < e < \varphi(n) \ \gcd(e, \varphi(n)) = 1$$

$$d = e^{-1} \, mod\varphi(n)$$

Where,

P, Q ⟶ Prime Numbers

n ⟶ Composite Numbers

D. Encryption Process

The proposed system follows the Improved ABE Scheme as illustrated in Fig. 4., which integrates two powerful cipher algorithms: AES and ABE.

The AES scheme is a traditional scheme that encrypts the given text document or text data into cipher form based on the Rijndael encryption scheme with 256-bit operational frequency and an essential algorithm over the innovative real-world application at present. The proposed system algorithm integration is called ABE, the encryption algorithm but the difference between AES and ABE is based on key

```
**Algorithm 2**: Secret Key Generation and Processing
Input: Segregated User Attribute $Segg_{attr}$
Output: Secret Key $Sk$.
Step-1: Collect the segregated user attribute from segregation process as:
    $SegregationProcess \xrightarrow{yields} SgAttr$
Step-2: Initiate the function for generating random key, using Random_Class $Rc$.
    Create Function and assign new object $New_{Obj}$ to Random_Class $Rc$.
    Provide $int_{Min}$ and $int_{max}$ values as a parameter to the Random_Class $Rc$ as:
    Function $Integer\_Key_{Generation}\ (int_{Min}, int_{max})\ Rc \leftarrow NewObj;$
Step-3: The Rc methodology takes min value and max value parameter as an input and generates a new random key between these given parameter values.
    $int_{Min}, int_{max} \xrightarrow{Generates} Random_{Key}$
Step-4: Return the generated random value to the required function.
    $Return \rightarrow Random_{Key} \rightarrow RequiredFunction$
Step-5: Random value retained and stored that into a new variable called Rn as: $Rn = RequiredFunction(int_{Min}, int_{max})$
Step-6: Concatenate the generated random value Rn and $\mathrm{Obj}_{\mathrm{Str}}$
    to generate a new credential to user as:
    $String_{Cred} = Rn + Obj_{Str}$
    Send this $Cred$ to the respective user mail-id;
Step-7: Authenticate process required the given credential and one time password to access the system further.
Step-8: Checks the input credential with the existing server credential over encrypted form.
Step-9: Checks the one-time password with server session password.
Step-10: Allows the user to proceed further, if credential and the one-time password matched with the server credential and the server session.
Step-11: Access Control provided properly based on the generated secret key
```

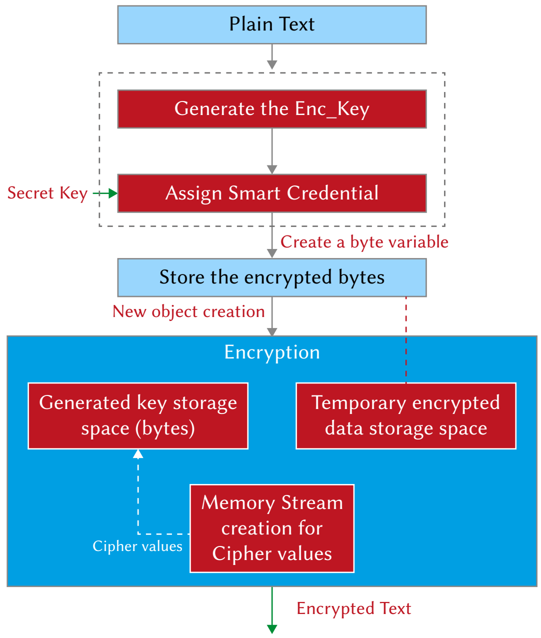

Fig. 4. Detailed work flow of encryption.

generation. In the ABE, the key generation process is based on user attributes, and the ABE generates the symmetric key for processing. All these features are integrated and make the hybrid algorithm process the entire system more securely. The Algorithm 3 explains the encryption process in detail. The following algorithm explains the encryption process in detail.

$C = E(K_E, P)$

$P = D(K_D, E(K_E, P))$

Where:

C → Cipher Text,

E → Encryption

K_E → Encryption Key

K_D → Decrypton Key

```
**Algorithm 3**: Encryption Algorithm
Input: Text Data or Document from user end as $Plain_{Text}$
Output: Encrypted Cipher Data (Cipher text) as: $Cipher_{Text}$
Step-1: Collect the Plain text or data from user end.
Step-2: Generate the string variable called $Enc\_Key$ and assign the generated smart credential from Algorithm2 as:
    $Enc_{key} \leftarrow String_{Cred}$
Step-3: Create a byte variable to store the encrypted bytes as:
    $Byte_{var[100]} \leftarrow Encoding\_Unicode\_GetBytes.GetBytes(Enc\_Text)$
Step-4: Create a new object to perform encryption based on advanced encryption procedure with respect to Rijndael process as:
    $\mathrm{AES\_Encryptor}_{\mathrm{aes\ enc}} \leftarrow \mathrm{AES}_{\mathrm{Encryptor}}.\mathrm{Create}()$
    $\mathrm{RFC\_2898\_Derive\_Bytes}_{\mathrm{Derived\_Bytes}}$
    $\leftarrow \mathrm{new}\ (\mathrm{RFC\_2898\_Derive\_Bytes}(\mathrm{Enc}_{\mathrm{key}}, \mathrm{Byte}_{\mathrm{var[100]}}))$
    $0_x49, 0_x50, 0_x51, \cdots 0_xn$
Step-5: Generate the key storage space with respect to Derived_Bytes over Step-4 as:
    $\mathrm{StorageSpace}_{\mathrm{Key}} \leftarrow \mathrm{Derived}_{\mathrm{Bytes}}$
Step-6: Generate the temporary encrypted data storage space with respect to Derived_Bytes over Step-3 and key storage space generated over Step-5 as:
    $\mathrm{Encryptor}_{\mathrm{Key}} \leftarrow \mathrm{Derived}_{\mathrm{Bytes}}.\mathrm{getBytes}(32)$
    $\mathrm{Encryptor}_{\mathrm{Data}} \leftarrow \mathrm{Derived}_{\mathrm{Bytes}}.\mathrm{getBytes}(16) + \mathrm{Enc}_{\mathrm{key}}$
Step-7: Create $\mathrm{Memory}_{\mathrm{Stream}}$ for storing the cipher values one by one to the generated encrypted data storage space over Step-6.
    $\mathrm{Memorystream}_{\mathrm{obj}} \leftarrow \mathrm{Memory}_{\mathrm{stream}}$
Step-8: Store the encrypted data to the storage space in byte format.
    $\mathrm{Crypto}_{\mathrm{stream}} \begin{Bmatrix} \mathrm{Memorystream}_{\mathrm{obj}}, \mathrm{Enc\_Create\_Encryptor}(). \\ Crypto\_Stream\_Mode[Write] \end{Bmatrix};$
Step-9: Return the encrypted text.
    $\mathrm{return} \rightarrow \mathrm{Cipher}_{\mathrm{Text}}$
```

E. Decryption Process

The proposed system decryption process illustrated in Fig. 5., is just a reverse of the encryption process, in which it is associated with the AES procedure.

Still, the variation over here is the key used to decrypt the data is unique compared to the traditional approach, which is extracted from the user attributes, and the dynamically generated key is mailed to the receiver. The receiver needs to provide the correct dynamic secret access to the system to decrypt the data. In this case, the given key is the valid means. The data is decrypted and allows the user to download the same otherwise, and the system blocks the user to proceed further. The algorithm 4 clearly illustrates the process of decryption straightforwardly.

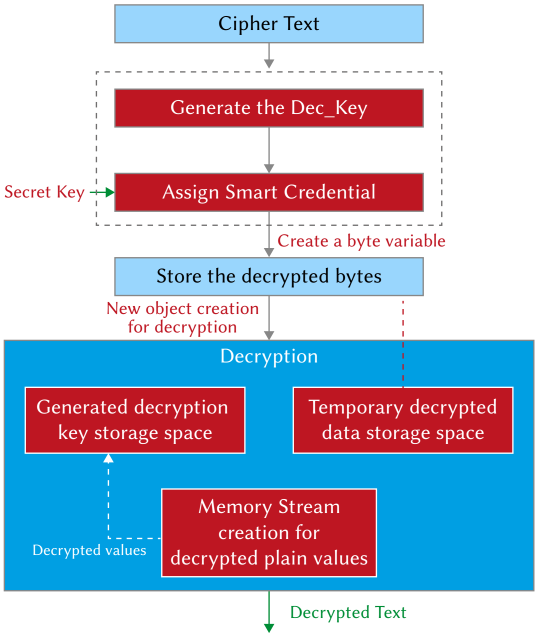
Cipher Text
Generate the Dec_Key
Secret Key
Assign Smart Credential
Create a byte variable
Store the decrypted bytes
New object creation for decryption
Decryption
Generated decryption key storage space
Temporary decrypted data storage space
Memory Stream creation for decrypted plain values
Decrypted values
Decrypted Text

Fig. 5. Detailed work flow of decryption.

P = D(C)

Where:

C → Plain Text,

D → Decryption Function

P → Plain Text

```
**Algorithm 4**: Decryption Algorithm
Input: Encrypted Cipher Data $Cipher_{Text}$.
Output: Decrypted Text Data or Document $Dec_{Text}$
Step-1: Collect the Cipher text from the server end.
Step-2: Generate the string variable called $Dec_{key}$ and assign the generated smart credential from Algorithm2.
    $Dec_{key} \leftarrow \mathrm{String}_{\mathrm{Cred}}$
Step-3: Create a byte variable to store the decrypted bytes as:
    $Byte_{var[100]} \leftarrow Encoding\_Unicode\_GetBytes.GetBytes(Dec\_Text)$
Step-4: Create a new object to perform decryption based on advanced encryption procedure with respect to Rijndael process.
    $AES\_Decryptor_{aes\_dec} \leftarrow AES\_Dncryptor.Create()$
    $\mathrm{RFC\_2898\_Derive\_Bytes}_{\mathrm{Derived\_Bytes}}$
    $\leftarrow new\ (RFC\_2898\_Derive\_Bytes(Dec_{key}, Byte_{var[100]}))$
    $0_x49, 0_x50, 0_x51, \cdots 0_xn$
Step-5: Generate the key storage space with respect to Derived_Bytes over Step-3.
    $\mathrm{StorageSpace}_{\mathrm{Key}} \leftarrow \mathrm{DerivedBytes}$
Step-6: Generate the temporary decrypted data storage space with respect to Derived_Bytes over Step-3 and decryption key storage space generated over Step-5.
    $Decryptor_{Key} \leftarrow Derived\_Bytes.\mathrm{getBytes}(32)$
    $\mathrm{Encryptor}_{\mathrm{Data}} \leftarrow \mathrm{Derived\_}Bytes.\mathrm{getBytes}(16) + Dec_{key}$
Step-7: Create a memory stream for storing the decrypted plain values one by one to the generated decrypted data storage space over Step-6.
    $\mathrm{Memorystream}_{\mathrm{obj}} \leftarrow \mathrm{Memory}_{\mathrm{stream}}$
Step-8: Store the decrypted data to the storage space in byte format.
    $Crypto_{stream} \begin{Bmatrix} Memorystream_{obj}, Dec\_Create\_Encryptor(), \\ Crypto\_Stream\_Mode[Write] \end{Bmatrix}$
Step-9: Return the decrypted text.
    $return \rightarrow Dec_{Text}$
```

III. Results and Discussion

In this summary, the experimental analysis of the proposed algorithm Improved ABE Scheme is to be discussed transparently with a practical graphical outcome. The entire process estimation proves the performance ratio of the proposed system with IABES is high compared to the classical cloud service structure. The proposed system performance and accuracy measures are estimated in terms of cost and time required to process the entire system over a real-time working environment. The whole programming and analysis are composed by using Microsoft supported tool platform, and the resulting units are properly accumulated pleasingly. The graphical estimations prove the resulting summary of the proposed approach and the proposed encryption and decryption accuracy levels in detail. Table I illustrates the performance measures of the proposed method, and that has been compared with many existing algorithms.

Table I shows the proposed IABES approach for evaluating the performance of key generation process. On the other hand, the ARMAX [20], took 29.61 milliseconds as response time with 80% robustness and 92% of accuracy. Whereas, the proposed approach took very less response time of 10.26 milliseconds, and achieved 92% of robustness with 98% of accuracy.

TABLE I. Performance Measures of Proposed Approach

Algorithm	Response Duration (ms)	Robustness (%)	Accuracy (%)
[20]	29.61	80%	92%
IABES	10.26	92%	98%

Fig. 6. illustrates the evaluation of the Secret Key generation process and its time requirement of the proposed system, which is explained in terms of several taken user attributes and key generation duration.

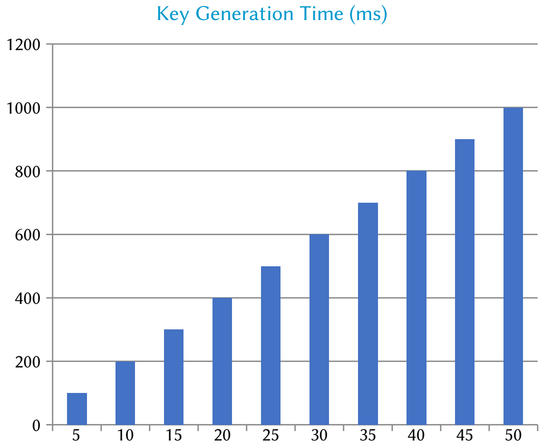

Fig. 6. Proposed algorithm key generation time evaluation.

Fig. 7. illustrates the proposed algorithm encryption time evaluation concerning the evaluation of processing time in milliseconds versus the number of user attributes taken for processing. Fig. 8. illustrates the proposed algorithm decryption time evaluation concerning the evaluation of processing time in milliseconds versus several users' attributes taken for processing. Fig. 9. illustrates the impact of the number of attributes used by users on the time cost. To analyze the effect of the number of attributes used by users on time cost, we set the total number of attributes to 21 and change the number of attributes used by users, the number of nodes in the access tree, and the access tree's depth shift together. This can affect KeyGen(R) and test as previously analyzed. The change in the number of attributes used by users will also affect the time consumption of the proposed algorithm. The impact is observed that the time cost of the proposed algorithm has a positive linear correlation with the number of attributes used by users. The proposed approach needs to bind attributes to the ciphertext, and computing the corresponding cost and time for each attribute results in a longer encryption time [21]. Fig. 10. illustrates that the impact of the number of keywords in the cipher-text on the time cost [21].

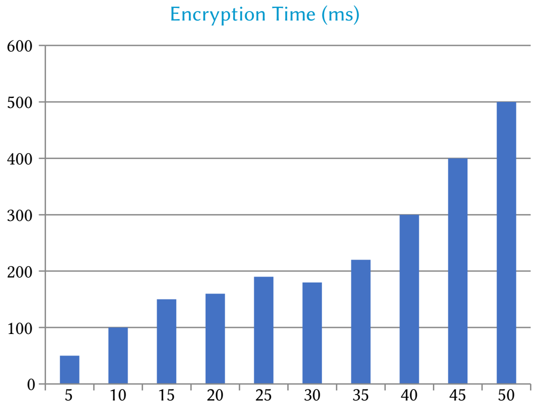

Fig. 7. Proposed algorithm encryption time evaluation.

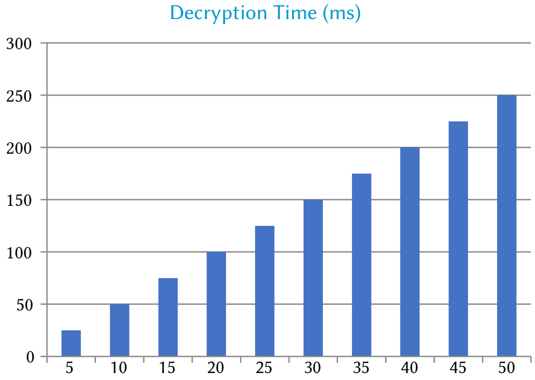

Fig. 8. Proposed algorithm decryption time evaluation.

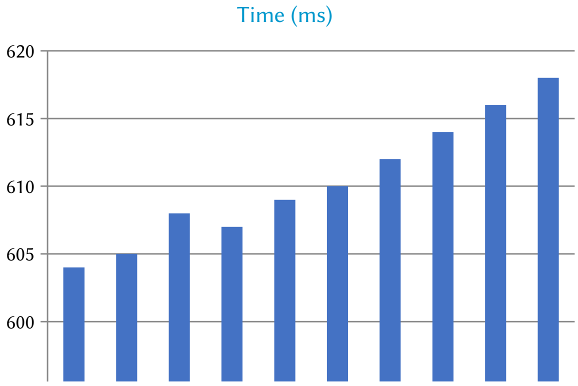

Fig. 9. Impact of the number of attributes used by users on the time cost.

IV. Conclusion and Future Scope

This paper demonstrates the performance and security features of the proposed algorithm Improved ABE Scheme (IABES). It shows the accuracy levels as high over the result and discussion section. This paper provides the secure hashing principle to prove the access control security in a detailed manner over the proposed system summary section. The hashing algorithm provides deep security to the users during authentication into the system with complete

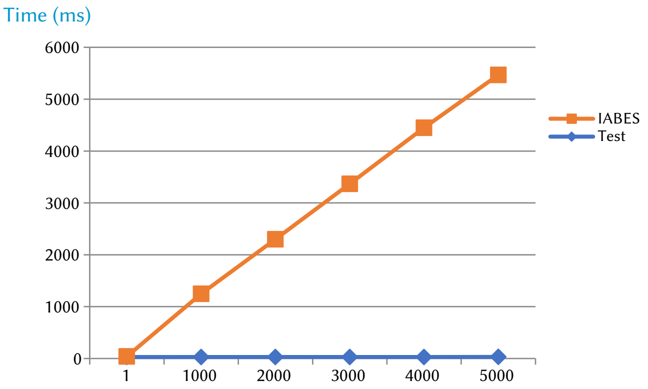

Fig. 10. Impact of the number of keywords in the cipher text on the time cost.

access control. The key generation process is handled through ABE logic, in which it accumulates the secret processing key from user attributes instead of using a general symmetric key principle. So, the proposed algorithm's security and processing nature are high as well as the proposed algorithm time efficiency is proved over the result and discussion section. The time accuracy and consumption scenario will diversely prove the cost efficiency of the proposed approach. The entire work is more suitable to provide cloud storage security to the data maintenance scheme with proper access control norms. In future, the work is further extended by adding some deep learning or machine learning algorithms to train the machine based on security threats. That kind of artificial intelligence approaches improves the efficiency of the overall system in terms of robustness and accuracy.

References

[1] N. A. Wigati, A. Wibisono, and A. N. Hidayanto, "Challenges of Infrastructure in Cloud Computing for Education Field: A Systematic Literature Review," Insight, vol. 43, no. 23, pp. 351-358, 2021.

[2] G. Ramachandra, M. Iftikhar, and F. A Khan, "A comprehensive survey on security in cloud computing," Procedia Computer Science, vol. 110, pp. 465-72, 2017.

[3] M. Samvatsar and P. Kanungo, "An Analytical Review and Analysis for The Data Control and Security in Cloud Computing," International Journal of Advanced Technology and Engineering Exploration, vol. 7, no. 73, pp. 241-246, 2020.

[4] S. M. Sasubilli, A. K. Dubey, and A. Kumar, "A Computational and Analytical Approach for Cloud Computing Security with User Data Management," In International Conference on Advances in Computing and Communication Engineering, IEEE, pp. 1-5, 2020.

[5] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User Collusion Avoidance CP-ABE with Efficient Attribute Revocation for Cloud Storage," IEEE Systems Journal, vol. 12, no. 2, pp. 1767-1777, 2018.

[6] J. Li, N. Chen and Y. Zhang, "Extended File Hierarchy Access Control Scheme with Attribute Based Encryption in Cloud Computing," IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 2, pp. 983-993, 2019.

[7] J. Li, Q. Yu, and Y. Zhang, "Hierarchical Attribute-Based Encryption with Continuous Leakage-Resilience," Information Sciences, vol. 484, pp. 113-134, 2019.

[8] J. Li, Q. Yu, Y. Zhang, and J. Shen, "Key-policy attribute-based encryption against continual auxiliary input leakage," Information Sciences, vol. 470, pp. 175-188, 2019.

[9] A. K. Dubey, A. K. Dubey, M. Namdev, and S.S. Shrivastava, "Cloud-user Security Based on RSA And MD5 Algorithm for Resource Attestation and Sharing in Java Environment," In Sixth International Conference on Software Engineering, IEEE, pp. 1-8, 2012.

[10] J. Li, Y. Wang, Y. Zhang, and J. Han, "Full Verifiability for Outsourced Decryption in Attribute Based Encryption," IEEE Transactions on Services Computing, vol. 13, no. 3, pp. 478-487, 2017.

[11] X. Liu, J. Ma, J. Xiong, Q. Li, and T. Zhang, "Ciphertext Policy Weighted Attribute-Based Encryption Scheme," Journal of Xi'an Jiaotong University, vol. 47, no. 8, pp. 4448, 2013.

[12] Y. T. Wang, K. F. Chen, and J. H. Chen, "Attribute-based traitor tracing," Journal of Information Science and Engineering, vol. 27, no. 1, pp. 181195, 2011.

[13] W. J. Chung and T.H. Cho, "A Security Scheme Based on Blockchain and A Hybrid Cryptosystem to Reduce Packet Loss in IoV," International Journal of Advanced Technology and Engineering Exploration, vol. 8, no. 81, pp. 945-956, 2021.

[14] X. Liu, J. Ma, J. Xiong, Q. Li, and J. Ma, "Ciphertext-Policy Weighted Attribute-Based Encryption for One-Grained Access Control," In International Conference on Intelligent Networking and Collaborative Systems, IEEE, pp. 51-57, 2013.

[15] Z. Liu, Z. Cao, and D. S. Wong, "White-Box Traceable Ciphertext-Policy Attribute-Based Encryption Supporting any Monotone Access structures," IEEE Transactions on Information Forensics and Security, vol. 8, no. 1, pp. 76-88, 2013.

[16] J. Li, Q. Huang, X. Chen, S. S. M. Chow, D. S. Wong, and D. Xie, "Multi-Authority Ciphertext-Policy Attribute-Based Encryption with Accountability,'' In Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, IEEE, pp. 386-390, 2011.

[17] Y. Jiang, W. Susilo, Y. Mu, and F. Guo, "Ciphertext-Policy Attribute-Based Encryption Against Key-Delegation Abuse in Fog Computing," Future Generation Computer Systems, vol. 78, pp. 720-729, 2018.

[18] Q. Li, H. Zhu, Z. Ying, and T. Zhang, "Traceable Ciphertext-Policy Attribute-Based Encryption with Variable Outsourced Decryption in Ehealth Cloud," Wireless Communications and Mobile Computing, vol. 2018, pp. 112, 2018.

[19] J. Zhou, Z. Cao, X. Dong, and X. Lin, "TR-MABE: White-Box Traceable and Revocable Multi-Authority Attribute-Based Encryption and Its Applications to Multi-Level Privacy-Preserving E-Healthcare Cloud Computing Systems,'' In proceedings of IEEE Conference on Computer Communication, IEEE, pp. 2398-2406, 2015.

[20] F. Piltan, S. TayebiHaghighi, and N. B. Sulaiman, "Comparative Study Between ARX and ARMAX System Identification," International Journal of Intelligent Systems and Applications, vol. 9, no. 2, pp. 25-34, 2017.

[21] Y. Yu, J. Shi, H. Li, Y. Li, X. Du, and M. Guizani, "Key-Policy Attribute-Based Encryption with Keyword Search in Virtualized Environments," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1242-1251, 2020.

Abhishek Kumar

Dr. Abhishek Kumar is Doctorate in computer science from University of Madras and done M.tech in Computer Sci. & Engineering from Government engineering college Ajmer, Rajasthan Technical University, Kota India. He has total Academic teaching experience of more than 7 years with more than 80 publications in reputed, peer reviewed National and International Journals, books & Conferences. He has guided more than 20 M.Tech Projects and Thesis and guiding 2 PhD Scholar. His research area includes- Artificial intelligence, Image processing, Computer Vision, Data Mining, Machine Learning. He has been Session chair and keynote Speaker of many International conferences, webinars in India and Abroad. He has been the reviewer for IEEE and Inderscience Journal. He has authored/Co-Authored 6 books published internationally and edited 16 books (Published & ongoing with Elsevier, Wiley, IGI GLOBAL Springer, Apple Academic Press, De-Grueter and CRC etc. He has been member of various National and International professional societies in the field of engineering & research like Senior Member of IEEE, IAENG (International Association of Engineers), Associate Member of IRED (Institute of Research Engineers and Doctors), He has got Sir CV Raman National award for 2018 in young researcher and faculty Category from IJRP Group. He is Editor of Special issue in the Journal Computer materials and continua [SCI and SCOPUS.IF- 4.98] and Intelligent Automation and Soft Computing [SCI, SCOPUS, IF-1.276] Cognitive Neuro dynamics, Springer [SCI, SCOPUS, IF-3.925].

Swarn Avinash Kumar

Mr. Swarn Avinash Kumar is a Research Engineer at the self-driving division of Lyft. He has previously worked at AI divisions of Google and Amazon as well. He has a total professional research experience of 6 years with more than 10 publications in reputed, peer-reviewed national and international journals, books & conferences. His research area includes: Artificial intelligence, computer vision, robotics, data mining, machine learning. He has filed multiple patents in the field of AI. He has been the reviewer for IEEE and IET conferences. He has co-authored 2 books (ongoing with Institution of Engineering and Technology & Eureka publications) and edited 2 books(Published & ongoing with Institution of Engineering and Technology & Eureka publications).

Vishal Dutt

Dr. Vishal Dutt is Doctorate in computer science from University of Madras, Chennai and has done MCA (Gold Medalist) from MDS University, Ajmer, Rajasthan, India. He has been working as the Assistant Professor of Computer Science at Aryabhatta College, Ajmer and also visiting faculty in Maharshi Dayanand Saraswati University (State Govt. University) Ajmer. He has total Academic teaching experience of more than 4 years. He has more than 35 publications in reputed, peer reviewed National and International, Scopus Journals & Conferences and Book Chapters. He has edited 2 books with Wiley, Eureka publications. He has been keynote Speaker and resource person of many workshops and webinars in India. He has been the reviewer for Elsevier, Springer, and IEEE Access. He has been Program Committee Member and Reviewer in the International Conference on Computational Intelligence and Emerging Power System ICCIPS 2021. He has recently presented 2 articles in Sixth International Conference on Advances in Computing & Communication Engineering Las Vegas USA ICACCE 2020 (22-24 June) IEEE EXPLORE Digital Library [SCOPUS] and 2 articles in the Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 2021 (4-6 Feb. 2021) IEEE EXPLORE Digital Library [SCOPUS]. His research area includes- Data Science, Data Mining, Machine Learning and Deep Learning. He also has Data Analytics Experience in Rapid Miner, Tableau, and WEKA. He has been working for more than 4 years in the field of data analytics, Java & Assembly Programming, Desktop Designing and Android Development.

Ashutosh Kumar Dubey

Dr. Ashutosh Kumar Dubey is currently in the department of Computer Science and Engineering, Chitkara University School of Engineering and Technology, Chitkara University, Himachal Pradesh, India. He received his PhD degree in Computer Science and Engineering from JK Lakshmipat University, Jaipur, Rajasthan, India. He is the Senior Member of IEEE and ACM. He has more than 14 years of teaching experience. He has authored a book name Database Management Concepts. He has been associated with many international and national conferences as the Technical Program Committee member. He is also associated as the Editor/Editorial Board Member/ Reviewer of many peer-reviewed journals. His research areas are Data Mining, Health Informatics, Optimization, Machine Learning, Cloud Computing, Artificial Intelligence and Object-Oriented Programming.

Sushil Narang

Dr. Sushil Narang is an Associate Professor in the Department of Computer Science and Engineering at Chitkara University, Rajpura, Punjab (India) since 2019. From 2006-2019, He was head of IT department at SAS Institute of IT & Research, Mohali, Punjab (India). From 1996-2006, He was Assistant Professor at Department of Computer Science &amp; Applications, MLN College, Yamunanagar, Haryana (India). He Completed his Ph.D. at Panjab University, Chandigarh (India). His Research on “Feature Extraction and Neural Network Classifiers for Optical Character Recognition for Good quality handwritten Gurmukhi and Devnagari Characters” focused on various image processing, machine as well as deep learning algorithms. His research interests lie in the area of programming languages, ranging from theory to design to implementation, Image Processing, Data Analytics and Machine Learning. He has collaborated actively with researchers in several other disciplines of computer science, particularly Machine Learning on real world use cases. He is a certified Deep Learning Engineer from Edureka. He possesses expertise in Object-Oriented Analysis; Design and Development using Java and Python programming using OpenCV in Image Processing and Neural Network construction. He has strong knowledge of C++ and Java with experience in component architecture of product interface. With Solid training and management skills, He has demonstrated proficiency in leading and mentoring individuals to maximize levels of productivity, while forming cohesive team environments.