I. J. Computer Network and Information Security, **2025, 2, 101-114**
Published Online on April 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2025.02.07

Efficient Cloud Computing Security Using Hybrid Optimized AES-IQCP-ABE Cryptography Algorithm

Jayaprakash Jayachandran
Department of Information Technology, Dr. M. G. R Educational and Research Institute, Chennai, India
And Lecturer in Information Technology, Karaikal Polytechnic College, Varichikudy, Karaikal, India
E-mail: jpit@dhtepdy.edu.in
ORCID iD: https://orcid.org/0009-0000-3446-7715

Dahlia Sam*
Department of Information Technology, Dr. M. G. R Educational and Research Institute, Chennai, India
E-mail: dahliasam@drmgrdu.ac.in
ORCID iD: https://orcid.org/0000-0003-1460-8714

Kanya Nataraj
Department of Information Technology, Dr. M. G. R Educational and Research Institute, Chennai, India
E-mail: kanya.v@drmgrdu.ac.in
ORCID iD: http://orcid.org/0000-0001-8549-6444

Received: 09 November 2023; Revised: 14 February 2024; Accepted: 08 October 2024; Published: 08 April 2025

Abstract: Data management has been revolutionized because cloud computing technologies have increased user barriers to expensive infrastructure and storage limits. The advantages of the cloud have made it possible for significant cloud implementation in major businesses. However, the privacy of cloud-based data remains the significant and most crucial problem for data owners due to various security risks. Many researchers have proposed various methods to maintain the confidentiality of the data, including attribute-based encryption (ABE). Though, the cloud is still dogged mainly by the security issue. To protect data privacy, the new encryption model "Advanced Encryption Standard- Improved Quantum Ciphertext Policy and Attribute-based Encryption" (AES-IQCP-ABE) is introduced in the present research. The suggested method twice encrypts the data and the attributes using the ABE at first. Second, using the AES technique, the encrypted data is encrypted before being delivered to authorized users. The dynamic, chaotic map function is used in the proposed approach to protecting user attributes throughout the initialization of the key, encryption of data, and decryption of data processes. For the encryption process, the inputs used in the proposed research are both unstructured and structured extensive medical data. Regarding computational memory, time for cloud data encryption, and decryption, the proposed model outperforms the previous ABE-based encryption and decryption algorithms.

Index Terms: Attribute-based Encryption, Cipher-text, Advanced Encryption Standard (AES), Data Decryption and Data Encryption.

1. Introduction

Data networking is becoming more popular due to the development of cloud computing technologies [1]. Recently, this method has gained widespread acceptance worldwide [2]. Due to the cloud environment's capacity to manage enormous amounts of data at a low cost and its adaptable access regulations, it has undergone rapid innovation and deployment [3]. To manage their data and serve their consumers, several public and private companies have created cloud networks [4]. However, Cloud computing has severe privacy and security issues comparable to any other technology, which continue to prevent its widespread implementation [5]. There is a potential that future growth and expansion of cloud-computing application areas may be impeded if the current issues with cloud security are not adequately handled [6-7]. Naturally, data owners become more concerned when sensitive data is exchanged via the cloud, especially regarding how cloud service providers will handle their data's security and privacy [8].

Using a simple encryption technique, the sensitive data is encrypted before being stored on several cloud servers [9]. Confidential data will continue to be at risk if the cloud servers get compromised in any way [10]. To increase cloud security, several entrance control mechanisms are included, and this encrypted data may be transferred successfully across the system [11]. Open key or symmetric encryption is inaccessible in the event of flexible access control compared to other cryptographic systems [12-14]. Therefore, for exchanging business data among several cloud servers, creating a safe encryption system and a precise access control method is required.

Previously, a number of common encryption techniques can be used to protect data, but their efficacy is dependent on the encryption key being secure and unknown to prospective adversaries. But if an attacker is able to obtain the concealed key and intercept encrypted messages in the network that presents a serious problem [15-17]. In those circumstances, the attacker can decrypt the communication, view its contents, re-encrypt it, and transmit it back to the appropriate recipient quietly. The recipient and the sender would not know that their data was read or shared by an unauthorized person.

Furthermore, to protect data both in transit and at rest, conventional cryptographic methods like hash functions and public, symmetric-key encryption have been used. Although these techniques work well in traditional computer contexts, the emergence of quantum computing poses a serious challenge to them [18-20]. The security of current cryptographic protocols is seriously jeopardized by the computing capacity of quantum computers; hence it is imperative that practitioners and researchers look into novel methods that could defend against quantum attacks.

To tackle this serious problem, Quantum Key Distribution (QKD) becomes a crucial remedy which is utilized in the proposed method. It offers a high level of security by guaranteeing that any effort to steal the encryption key will be discovered. Moreover, in our proposed framework, we utilized two step encryption using Attribute-based Encryption (ABE) and Advanced Encryption Standard (AES) techniques based on novel integrity computation. This quantum-based dual encryption model's security is further enhanced using a chaotic map-based hashing. Throughout the processes like initialization of key, encoding of data, and decoding of data processes, the dynamic, chaotic map function is used in the proposed approach to protecting user attributes. For the integrity verification and encryption procedure, this approach accepts massive medical data that is both structured and unstructured as input.

The main contributions of the research are:

- Novel integrity computation-based AES-IQCP-ABE encrypts each cloud user's data before being uploaded to the remote cloud server to improve data privacy and security problems.
- For integrity value computation, input from the user's attributes is employed in the initial phase of the quantum key generation and encryption processes.
- A quantum key generation technique based on integrity value is created for the AES-IQCP-ABE model's attributes, rules, and key generation process in the second stage.
- In the third stage, setup, key creation, encryption, and decryption are performed using computed integrity values and a quantum key.

The paper is structured as follows: Section II discusses the existing work explanation and the literature. In section III, the suggested model is described and the simulation findings are discussed in section-IV. Finally, section V delivers the conclusion.

2. Related Prior Works

Utilizing hybrid cryptography on cloud computing, Chinnasamy et al. [21] provide a hybrid technique for adequate data protection. Blowfish and ECC were combined to create the methodology used in this research. Blowfish (symmetric key cryptography) and the ECC (public key cryptography) algorithm are the hybrid algorithms employed here. More effective, smaller, and faster cryptographic keys may be created using ECC (public key encryption) based on the Elliptic Curve Theory.

For maintaining data integrity and security, Rehman et al. [22] evaluated a safe and effective method for data transferring over the cloud. In this paper, the key is generated using the ECC algorithm rather than the key produced by the AES method, resulting in a minor key size. Data is encrypted using a private and public key, like symmetric/asymmetric encryption. Data encryption and decryption are performed using ECC and AES, respectively.

By choosing the best private key, Verma et al. [23] created a cloud-based data security paradigm that is QC focused. Here, ABE is carried out to ensure improved data transfer across cloud nodes. In this study, improved QKD is used for key generation in ABE. The decryption and encryption are carried out per this. The self-modified Aquila optimization (SM-AO) technique is also used for ABE's best secret essential selection.

A two-level, asymmetric (ECC) and symmetric (AES) cryptographic approach was used by Hodowu et al. [24] to construct an increased data security model. This innovative method protects data transmission across the cloud, guarding against unauthorized access and creating useful information from the transmitted data. The paper considered the duration of cryptographic processes, improved data security against unauthorized access, preventing them from seeing the actual data, enabled secrecy by preventing access, data integrity, improving the speed of the cryptography process by the use of smaller ECC keys and more people are using the Cloud Computing platform with confidence.

Enhanced Cloud Security Model utilizing ECSM-QKDP, created by Sundar et al. [25], incorporates quantum key cryptography to provide cloud storage security and deal with data dynamics. Additionally, this approach considers the possibility of communication between three entities, including the Legitimate User (LU), Data Owner, and Cloud Server, in which two steps are taken to share the quantum keys. BB84 QKDP is utilized in the first stage. According to secure keys and distance boundaries, the Secure Authentication Protocol is defined in the second stage using Hierarchical ABE approaches. The model allows transmitting secured keys to the LU through a reliable route.

To enhance the privacy and security of cloud users' sensitive data, an innovative QH-CPABE architecture is proposed by Singamaneni et al. [26]. The trilinear Tinkerbelle chaotic function is used in the intended standard to create keys, encrypt cloud consumer text, and perform decipherment on any consumer attribute. Big cloud clinical data from both structured and unstructured sources were used to construct this model.

For consumers' vast amounts of data encryption, decryption, and key generation processes, some existing approaches require more excellent processing time and resources, and also dynamically changing access policy structures and user attribute-based access rules by requirements are not supported by traditional approaches. To overcome these issues, we propose a hybrid cryptographic approach.

3. Proposed Methodology

Traditional cloud security techniques on massive data sets have significant concerns about data security and privacy preservation because most third-party service providers are unreliable and more susceptible to clients' private information. Each cloud user's data is encrypted using a cutting-edge integrity computation based AES-IQCP-ABE before being uploaded to the distant cloud server to improve data privacy and security problems. The problems of the previous approaches, including AES, ABE, and CP-ABE, are solved by proposing a new effective algorithm in the present work. This research uses the Improved Quantum Ciphertext Policy–Attribute-Based Encryption (IQCP-ABE) and AES through the AES–IQCP–IDABE scheme for data encryption.

The proposed model uses three steps for preserving data security in the cloud: calculation of the user's integrity value, generation of quantum keys, and data encryption and decryption using integrity values. The integrity computation takes the user's attributes as inputs in the first step. Then the quantum key generation and encryption process uses these integrity values. The generation of quantum keys with integrity values is developed in the second step for the AES-IQCP-ABE model's essential generation procedures, policies, and attributes. The setup, key generation, encryption, and decryption phases all use calculated quantum keys and integrity values in the third step.

The encryption and decryption processes used in cloud computing are handled by the innovative integrity-based Improved Quantum CP-ABE algorithm. The user's input attributes list serves as the basis for the master key, public key, secret key, and quantum key in this approach. In the cipher text, the access policy structure tree encoded, as depicted in Fig. 1, is used to decrypt the cipher text.

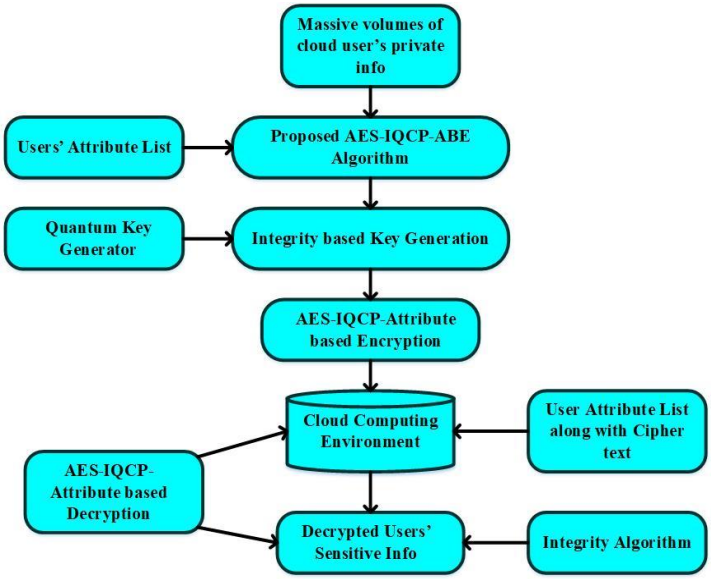

Fig.1. Block diagram of the proposed model

Massive amounts of structured and unstructured big data, which are frequently housed in cloud-based systems, raise serious issues and cause worry about the sensitive information of cloud users, especially in maintenance and management. Similarly, many security methods now used to secure that data are only sometimes sufficient. Therefore, most third-party services still have flaws that might expose or allow unauthorized access to customers' private information. This paper provides a method for encrypting massive cloud datasets before storage using a unique improved non-polynomial

integrity-centric quantum HCP-ABE prototype for directly enhancing approaches for solving the privacy and security issues of users' personal information. Tierce approaches are used in the present expected model: computation for the integrity value of the users, generation of quantum keys, and encryption and decryption of sensitive user data with a concentration on integrity.

The creation of keys is carried out using the first technique, which was established using inputs from cloud users and several parameters for digest computation and fixed random digest values. According to characteristics-based personal digest values of the users, key generation approaches, and the CPABE standard's access policies, the generated digest value is used in the second stage for constructing a quantum key for those users. The computed keys and digest metrics used in the initial arrangement, key creation, encryption, and decryption phases make up another potential technique. The proposed approach result is an Advanced Standard Encryption-Enhanced and Dynamic Non-linearNon-linear Polynomial Integrity-based Improved Quantum Hash-Cipher Policy Attribute-Based Encryption (AES-EDNPIIQHCP-ABE) algorithm, which may be held accountable for the highest level of encryption and decryption of customers' sensitive data over cloud-based databases and servers. The users' input attributes or features may serve as the foundation for the master, public, private, and quantum keys in this model. Later, the particular attributes of the users are used to decrypt the cipher text, and a formation tree is used to achieve the control access ingrained in the user's cipher text, as shown in Fig. 1. The innovative data encryption and decryption procedure in this research uses a randomized security key. For high data security, an entirely novel chaotic-based dynamic key called AES-EDNPIIQHCP-ABE is initialized in each iteration.

3.1. Iterative non-linear Polynomial Curves based Integrity Algorithm (INPIA)

In the proposed methodology, the process begins by collecting user attributes necessary for authentication and access control. These attributes might include identifiers like the user's first job. One of the following five items from the list below can constitute the user's attributes in the proposed design:

- First job;
- Name of their hometown;
- Last five digits of their credit card;
- Favorite sports;
- Favorite team.

The INPIA is then applied to these user attributes to compute an integrity value. This involves initializing a non-linear polynomial that represents the user’s attributes and iteratively calculating the polynomial’s output. This integrity value ensures that the attributes are valid and have not been tampered with. The working process of this algorithm is explained below.

The group multiplicative with the group order 'n' is represented as$Z(n^2,*)$.

$$\text{Where order } (n) \leq Order\big(Z(n^2,*)\big)$$

For increasing the security parameters, a non-linear chaotic polynomial map is employed by the proposed approach during the key generation process. The non-linear equation's fundamental relation for recursion is given as

$$T_n = kx - T_{n-2} \tag{1}$$

$$T_0 = 1 \tag{2}$$

$$T_1 = k(1 + \lambda^2) \tag{3}$$

Finally, the non-linear polynomial curves are represented $\frac{dT_n}{ds}$with more extraordinary powers while the randomized security parameter is represented by λextracted from$Z(n^2,*)$.

With absolute more significant factors, we can obtain families of non-linear polynomial curves by resolving the recurrence equation.

Input: Initialization parameters, the permutation matrices like x and K, input data D, h[]round hash vector, block bits, Number of rounds NR, block size BS, and Data size DS.

Output: Biometric Integrity value.

Step 1: Hash vector and input parameter initialization. $h[blockbits/8] \leftarrow 0//initialize$Hash vector to zeros.

Step 2: Choose a polynomial equation using the secret key$s(k)$. A unique chaotic hash-based approach is created to increase the stability of the security. In general, it combines chaotic systems and Bernoulli Logistic. More complicated behaviors are provided by the traditional chaotic mapping methods like logistic mapping from intervals [0, 1] into [0, 1]. According to the below equation, the generation of state happens.

$$s(n+1) = \alpha(H_i)\big(1 + x(H_i)\big) \tag{4}$$

In this case, αit falls in the range of [3, 7.4]. The chaotic discrete-time dynamic system in the above map creates a pseudorandom state sequence. The discrete-time stochastic process is defined by using Bernoulli's procedure.

$$P(x) = s(k) * p(x) \tag{5}$$

From the curve family, the$p(x)$is derived. The element selected at random from group Z is$s(k)$.
Step 3: Compute

$$\theta = lcm(r_1 - 1, r_2 - 1) \text{and} n = r_1 \times r_2. \tag{6}$$

Where elements are denoted asr_1 andr_2, and cyclic group Z is used for taking these elements.
Step 4: Choose an integer that is close to being prime $\theta$and use it asp_1 computep_2d as

$$p_2 = \frac{\theta}{(p_1+1)} \tag{7}$$

Step 5: a, b, c, d from $Z(n,*)$are used for random value selection
Step 6: Calculate k_1, k_2, k_3, k_4 as

$$K_1 = 1 + b * (p_1 \times P_2) \tag{8}$$

$$K_2 = K_1^a \, mod \quad (n^2) \tag{9}$$

$$K_3 = c^{m_1}.k_1^{a_2} \, mod \quad (n^2) \tag{10}$$

$$K_4 = c^{m_1}.d^{p_1.(r \times r_{2_1})} \, mod \quad (n^2) \tag{11}$$

Step 7: Round key permute $R_1 = \{k_1, k_2, k_3, n_1, g_1.g_2, P(x)\}$
Step 8: Round key permute$R_2 = \{m, r_1, r_2, \theta, s(k)\}$s.
Step 9: While $(DS > BlocBit/8)$
Do
$P \leftarrow$ Subblock's first 4 bytes
For each PBlock (partition block)
Do
Choose a random permutation box (K) and $R_1 \, R_2$ keys as$x$.

$$x_1 = ((K^T.x) \, mod \quad (max \quad \{x.eigenvalues()\}) \tag{12}$$

$$y = K.Kromproduct(x_1).scale(256) \tag{13}$$

$$C_i = P[i] + c[max \quad (0, i-1)] \tag{14}$$

$$C_i = Max\{R_1, R_2\} \oplus C_i \oplus y_i \tag{15}$$

Done

$$P[i] \leftarrow Left \, Re \, v \, erse(P[i])$$

$$P[i] \leftarrow RightShift(P[i], 3)$$

If $(r + 1 < NR)$ Then

$$P[i] \leftarrow RightShift(P[i], 3)$$

$$P[i] \leftarrow LeftShift(P[i], 6)$$

$$P[i] \leftarrow RightShift(P[i])$$

End if
Done

3.2. Bilinear Pairing

Once the integrity value is computed, bilinear pairing is employed to facilitate cryptographic operations. This allows for secure communication between parties by enabling operations like attribute verification. Bilinear pairing ensures that only users with valid attributes can access certain data or perform specific actions.

For prime order Q, the two separate multiplicative cyclic groups are represented as G_0andG_1. Here $< g >$is the producer ofG_0. e is used for representing the bilinear map. This implies$G_0 \times G_1 \rightarrow G_1$.

The occurrence of bilinear pairing is used for achieving two essential requirements.

For every individual $a, b \in Z_q^+ < g >$and inG_0.

Here,$e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$

$$e(g, g) \neq 1 \text{ (in case of non-degenerated).}$$

Only the bilinear map can be examined if the first two requirements are satisfied. Bernoulli's mapping is combined with this method to increase speed and total unpredictability.

3.3. Key Setup and Generation Using Quantum Key Distribution

The system then performs key setup and shared key generation using quantum key distribution (QKD). Our suggested paradigm incorporates quantum key distribution (QKD) as a fundamental element for secure key exchange, mitigating the threat posed by quantum computing. QKD offers a strong defense against attacks from quantum computers by utilizing the concepts of quantum physics to identify any effort at eavesdropping. Because of this, it is intrinsically more secure than conventional cryptography techniques, which are susceptible to quantum algorithms.

In this approach, the quantum signals are used for sending bits from source to destination to create an essential K without a shared key value, which is the primary objective of QKD. Fig. 2 depicts the QKD procedure. Regular data and quantum channels are needed for the proposed QKD. A set of primary and polarizing cubits and cyclic group random generators are needed for both sender and receiver. During communication, the man-in-the-middle assaults are prevented from attacking the quantum channel using the BB84 authentication protocol in this model. For the QKD-based AES-CPABE architecture, the authorized users get the shared key created by the QKD. The KeyGen, Decrypt, Setup, and Encrypt algorithms, also part of the QKD-based AES-CP-ABE system, are explained here.

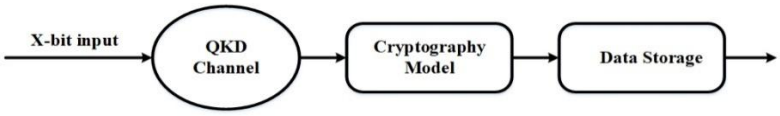

Fig.2. QKD on cryptography model

The QKD comes with its own set of challenges, including issues related to quantum key distribution network infrastructure, susceptibility to quantum attacks, and the need for specialized hardware. However, our suggested method solves this issue by integrating dual encryption techniques (ABE and AES) that defend against both quantum and classical assaults, facilitating secure communication while preserving the confidentiality and integrity of the data. Moreover, the strong key management procedures that lessen the demand for specialized hardware.

Also, the suggested method makes use of a decentralized QKD network, which allows several nodes to cooperate together on key distribution. This cooperative method can get over distance restrictions by relaying quantum keys through several nodes, increasing the effective range of quantum key distribution. This decentralized network can be quickly expanded to handle new nodes or users, making it flexible for real cloud settings where the number of users and devices may change. Furthermore, the decentralized architecture lowers the possibility of a single point of failure by allocating cryptographic duties throughout the network.

3.4. System Setup

After the shared key generation with QKD, the master key and public key generation process is conducted. It is explained in the following equations.

$$PublicKey(P_k) = \begin{Bmatrix} Pr\,o\,poseHash(Quantumkey|attributes), (g \in G_1(H[i]), \\ g_p \in G_2(H[i])/G_1, G_2 \in \text{Integritvalues}(H[i]), h \in Z_r, g = e(g, g_p) \end{Bmatrix} \quad (16)$$

$$\text{MasterKey}(M_k) = \begin{Bmatrix} \alpha \in P_k(g_p), \beta \in Hash(sec\,r\,etkey(\text{HAtr1}[0]\text{^Hatr2}[1]\text{^}\ldots\text{HAtr1}[n])), \\ Z_r, e(\alpha, \beta)^{\theta_2} \end{Bmatrix} \quad (17)$$

In this, the bilinear group is represented as G with generator k and prime order p, which $\theta_1, \theta_2 \in G_p$satisfies the bilinear criterion and the lack of degeneracy property. After this process, we obtained public key(P_k), QKD shared key and master key(M_k).

3.5. First Level of Encryption using ABE Algorithm

After the key generation process, the first layer of encryption is performed using Attribute-Based Encryption (ABE) to encrypt both the sensitive data and the associated user attributes. ABE ensures that data is encrypted in such a way that only users with specific attributes can decrypt it, enabling fine-grained access control.

For input of the initial plain text message (M), the desired cipher text is produced by the encryption method. The access tree structure T is used by the encryption method for message (M) encryption. A p-integer random number r modulo Z is selected by this approach and set $q(R,0) = r$ before moving on from the root node. It sets $q(x,0) = q\big(parentnode(x, index)\big)$ for the intermediate nodes x. The collection of leaf nodes is represented as L, and T is the access tree structure provided to construct the cipher text.

$$CipherText(C) = \{C^1 = M.e(k,k)^{\Theta_1.r}\} \tag{18}$$

$$T, C^2 = m^r, \forall x \in X: C_x = k^{q(x,0)} \tag{19}$$

$$C_x^1 = H\big(A(x)\big)^{q(x,0)} \tag{20}$$

Regarding data encryption, examining the homomorphic property:
Homomorphism encryption and decryption use $\Psi(i)_0, \Psi_0^{'}$as input.
Additive Homomorphic Encryption

$$\text{EncD}(I_1 + I_2) = EncD(I_1) + EncD(I_2) \tag{21}$$

Multiplicative Homomorphic Encryption

$$\text{EncD}(I_1.I_2) = EncD(I_1) + EncD(I_2) \tag{22}$$

$$\Psi(i)_0 = EncDc(I_1).Enc(I_2) \tag{23}$$

$$(I_1 + \gamma * \beta)\, mod\, n \tag{24}$$

Where$n = \alpha * \beta$.

$$EncD(I_1 + I_2) := Enc(\psi(i)_0) + Enc(\psi(i)_0^{'}) \tag{25}$$

$$EncD(I_1 + I_2) := (I_1 + \gamma * \beta)\, mod(n) + (I_2 + \gamma * \beta)\, mod\, n \tag{26}$$

$$EncD(I_1.I_2) := EncD(\psi(i)_0).EncD(\psi(i)_0^{'}) \tag{27}$$

$$:=(I_1 + \gamma * \beta)\, mod\, n + (I_2 + \gamma * \beta)\, mod\, n \tag{28}$$

After this process, the first layer of encrypted text is obtained. Alongside the attributes, the sensitive data itself is also encrypted using this method.

3.6. Key Generation

Then, to produce private key (prK), the attributes set (A) is used by the Key generation method. A private or secret key is produced as an output by the KeyGen algorithm from a collection of characteristics A and QKD (shared key). This approach assigns each attribute a random integer between r and$rand_j$. As the QKD (shared key) factor, these random integers were chosen.

$$SecretKey(S_k) = \Big\{D = k^{{((\Theta_1 + ranmd))}/{\Theta_2}} \tag{29}$$

$$D(j) = k^{rand * H(j).rand_j}, D(j) = krand_j\Big\} \tag{30}$$

3.7. Second Level of Encryption using AES Algorithm

After the first layer of encryption using ABE, we performed second layer of encryption process using AES algorithm The AES is a symmetric cypher family that includes the three block ciphers AES-128, AES-192, and AES-256. The block size of AES is always 128 bits, and the key size can be either 128 or 192 or 256 bits. The AES algorithm consists of four rounds. They are Add round key, mix columns, Shift rows, and Substitute bytes. For 128-bit keys, these stages are repeated

ten times; for 192-bit keys, they are repeated twelve times; and for 256-bit keys, they are repeated fourteen times. In this work, we are utilized AES-256. Therefore, the steps are iterated for fourteen times. The working procedure of this algorithm is given as follows,

Step 1: In this step, the first level encrypted cipher text is given as the input x. Then, assign State= x and execute the Add Round Key operation, which combines the XOR operation with the state to execute Add Round Key.
Step 2: Use an S-box to execute the SubBytes substitution process on State for each of the first Nr - 1 rounds;
Step 3: Permute ShiftRow, implement MixColumns and Add Round Key on State.
Step 4: Implement a SubBytes, ShiftRows, and Add Round Key operations.
Step 5: obtained cipher text.

After this process, we obtained dual encrypted cipher text. The dual encrypted data is then securely stored in the cloud environment. This dual approach ensures that data remains secure at multiple levels, protecting against unauthorized access and ensuring that only users with the correct attributes can access the information.

3.8. Data Decryption

The decryption process involves two main steps: decrypting the AES layer first and then the ABE layer. It receives the following inputs: ciphertext (C), public key (PK) for the private key (Sk) for a set of attributes, and access structure (T). Decryption is the inverse process of encryption. The following parameters are used in this procedure: attributes set A, secret key, cipher text, and the access tree T's node x. To convert the first layer of ciphertext, the AES-256 decryption procedure entails reversing the encryption processes. In this, a 256-bit key is first divided into many round keys. In order to reverse the final key addition from the encryption phase, the ciphertext is first XORed with the final round key. The data then goes through 14 rounds of transformations: each round starts with the InvShiftRows operation, which moves the state array's rows to the right, and ends with InvSubBytes, which replaces each byte with its matching value from the inverse S-box. In the AddRoundKey step, the state array and the current round key are XORed. The InvMixColumns transformation adds intricacy to the data by combining the state's columns for the first 13 rounds. The last round skips the mixing phase and ends with another use of AddRoundKey, InvShiftRows, and InvSubBytes. These actions essentially reverse the AES encryption process, storing the decrypted text in the state array.

After the first layer of decryption using AES, the second layer of decryption using ABE is carried out to obtain the plain text. This decryption process is performed by examining the homomorphic property and the added pattern policy is extracted by considering$\big(EncD(I_1 + I_2)\big), Enc(I_1.I_2).e(C(i)_1, K_1, i).e\big(C(j)_2 K_{1,j}\big).e\big(C(k)_3, K_{1,k}\big).EncD(I_1 + I_2).$

$$= Enc(\psi(i)_0 + \psi(i)_0^{'}) \tag{31}$$

$$= Enc(\Psi(i)_0) + Enc(\psi(i)_0^{'}); \tag{32}$$

$$:= (\Psi(i)_0^{'} + \gamma * \beta)\ mod\ n + (\Psi(i)_0^{'} + \gamma * \beta)\ mod\ n\ EncD(I_1.I_2) \tag{33}$$

$$:= Enc(\psi(i)_0.\psi(i)_0^{'}) \tag{34}$$

$$:= Enc(\psi(i)_0).Enc(\psi(i)_0^{'}) \tag{35}$$

$$:= (\psi(i_0) + \gamma * \beta)\ mod\ n\ .+(\psi(i_0) + \gamma * \beta)\ mod\ n\ ; Dec\big(EncD(I_1 + I_2)\big) \tag{36}$$

$$:= \Big((\psi(i)_0 + \gamma * \beta)\ mod\ n + (\psi(i)_0^{'} + \gamma * \beta)\ mod\ n\Big)\ mod\ \alpha \tag{37}$$

$$:= I_1 + I_2 \rightarrow (1) \tag{38}$$

$$Dec\big(EncD(I_1.I_2)\big) := \Big(EncD(\psi(i)_0.\psi(i)_0^{'})\Big) \tag{39}$$

$$:= EncD(\psi(i)_0).Enc(\psi(i)_0^{'})\ mod\ \alpha; \tag{40}$$

$$:= (\psi(i)_0 + \gamma * \beta)\ mod\ \ \ n. + \Big((\psi(i)_0^{'} + \gamma * \beta)\ mod\ n\Big)\ mod\ \ \ \alpha; \tag{41}$$

$$:= I_1.I_2 \tag{42}$$

At the end of these procedures, we obtained the decrypted plaintext. This multi-layered strategy assures both the integrity and confidentiality of critical data, shielding it from illegal access while permitting secure distribution to

authorized users.

4. Results and Discussion

Using a client with 8 GB of RAM and an Intel (R) CPU set at 2.13 GHz, each proposed experiment runs in real-time on Amazon AWS cloud storage. Third-party libraries such as Apache Math, Amazon Python SDK, and Apache Commons are necessary for this architecture.

4.1. Cloud Platform Base

For the evaluation analysis of the proposed approach, cloud platform AWS cloud servers are used on biomedical user-sensitive private info security. The experimental results are generated by using Amazon EC2 instances and Simple Storage Service (Amazon S3) in this cloud server and secure sensitive data using integrity and encryption techniques. In the cloud, EC2 provides a dynamically adaptable calculating facility. For innovators, EC2 offers quick and convenient web-based scalar computing. The arrangement of kit control projected to the best aspect is available with no effort countability and an easy technique to understand the edge. The overhead edge has the ability to consider and exclude certain situations. By allowing customers to conduct an experiment on which circumstances should be published to the WWW, EC2 instances are put up in Virtual Personal Cloud (VPC). System ACLs, privacy marches, and Inbound and outbound systems will all be adopted as part of the final pact. This model may be used with many 2D or 3D image types. Cloud Watch online support from Amazon is available for monitoring and testing EC2 instances. Cloud Watch is in charge of functional implementation, demand layouts, and dynamic resource management. Amazon's elite registration service handles the complicated computing demands of its customers. With regard to Amazon EC2, customers of the cloud are allowed to select a working structure, load pre-existing standard programs, set up system access rights, and replicate that scenario with preprocessing power.

4.2. Experimental Results

According to the hash digest value's bit range and its arbitration, Table 1 provides the performance analysis of the proposed hybrid data security technique. The highest bit change hash digest value is used for the proposed approach. According to the table, the proposed non-linear polynomial integrity approach stands out because of its large amount of susceptible user data, constant/fixed key, or chaotic dynamic key generation process. Additionally, the recommended solution employs minimal communication/transmission overhead/cost and highly randomized arbitrary/ changeable key sizes compared to typical methods.

Table 1. Comparative evaluation of the proposed algorithm concerning various computing parameters

Properties	**Proposed AES-IQCP-ABE**
Big data	Yes
Dynamic Key	Yes
Transmission cost	Very low
Key size	Highly randomized
Static key	Yes

Table 2. The proposed model performance is compared to established hash methods

Medical cloud data	Whirlpool	SHA-1024	MD5	Proposed Integrity Algorithm
Efficiency	$o(n\,log(nk))$	$O(nk\,log(n))$	$o(n^{2k})$	$o(n\,log(k))$

Regarding hash computation and sensitivity, the comparative analysis of the proposed integrity method and the conventional methodologies is given in Table 2. It can be shown from Table 2 that the present integrity technique to hash creation is superior to the traditional approach. Here, the bit size is represented by n, and the key size is represented by k.

Encryption and Decryption Processing Time Costs

Tables 3 and 4 provide information on the costs and time associated with data encrypting and data decrypting of varying sizes with increasing ciphertext properties. When attributes are less, the suggested method AES-IQCP-ABE was shown to have higher encryption and decryption times than AES-CP-IDABE and Improved-CP-ABE (I-CP-ABE), according to the observation. However, encryption time is increased when the number of attributes are increased. Furthermore, when the characteristics are greater than 40 KB, it was discovered that the suggested AES-IQCP-ABE works better. The proposed method achieved 17.44% and 24.30% and 23.44% better results than AES–CP–IDABE and I-CP–ABE methods for 10KB, 50KB and 100KB data size in encryption. This result is achieved due to the advantages of both ABE and AES encryption methods. In our proposed framework, the ABE rapidly encrypt smaller data pieces and AES effectively handle the larger payloads. This results in faster total encryption speeds. Moreover, the decryption time also reduced compared to other algorithms. It takes only 245ms time to decrypt the 100KB data due to the optimizing

key management through QKD. It minimizes the waiting times involved in retrieving keys throughout the decryption process by offering quick and safe key access. Thus, the suggested approach is more suitable for real-world applications due to its fewer complexity compared to other algorithms. Figs. 3 and 4 provide a comparison graph for the various data sizes.

Table 3. Time for data encryption with different data sizes

Data size	Method	Number of attributes (ms)				
		10	**20**	**30**	**40**	**50**
10KB	AES–CP–IDABE	210	275	390	470	550
	I-CP–ABE	150	250	400	500	600
	AES–IQCP–ABE	190	220	300	385	420
50KB	AES–CP–IDABE	285	430	555	580	690
	I-CP–ABE	190	400	550	600	780
	AES–IQCP–ABE	150	330	425	450	500
100KB	AES–CP–IDABE	360	480	595	640	850
	I-CP–ABE	270	440	600	700	1000
	AES–IQCP–ABE	250	360	455	500	650

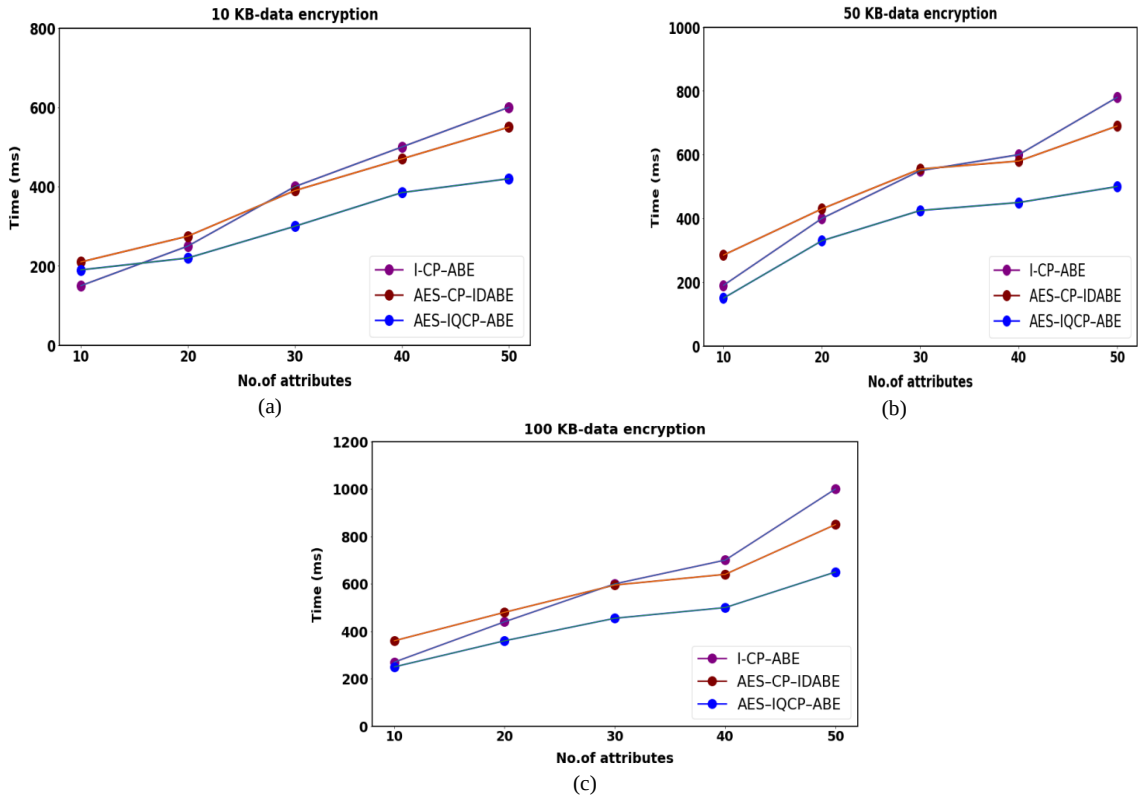

Fig.3. (a) Performance comparison for encrypting 10 KB of data. (b) Performance comparison for encrypting 50 KB data. (c) Performance comparison for encrypting 100 KB data

4.3. Performance Comparison

For runtime computation, a performance comparison of a proposed model with an existing model is shown in Fig. 5. For establishing the average computing time, different data sizes are used in experimental analysis. The proposed approach for cloud security in the figure offers high computational efficiency. For cutting-edge models, the average computing time is provided in Table 5, depending on the size of the data and the number of attributes. From the table, we identified that the proposed achieved minimal computing time compared to other methods. It takes 214.742ms execution with 50 number of attributes. Moreover, it achieved 67.24%, 52.12%, and 29.24% better results than existing CPABE, FH-ABE, and QKD CPABE methods respectively. Because the proposed approach minimized the computational effort due to the iterative non-linear polynomial curves based integrity algorithm. With this method, the proposed performed rapid user attribute verification and speeds up the execution process.

Table 4. Time for data decryption with different data sizes

Data size	Method	Number of attributes (ms)				
		10	**20**	**30**	**40**	**50**
10KB	AES–CP–IDABE	110	115	125	145	175
	I-CP–ABE	80	90	110	150	190
	AES–IQCP–ABE	120	100	115	150	160
50KB	AES–CP–IDABE	130	140	155	180	250
	I-CP–ABE	110	130	150	200	300
	AES–IQCP–ABE	120	150	150	175	220
100KB	AES–CP–IDABE	300	375	410	525	610
	I-CP–ABE	250	350	400	550	700
	AES–IQCP–ABE	245	300	400	520	600

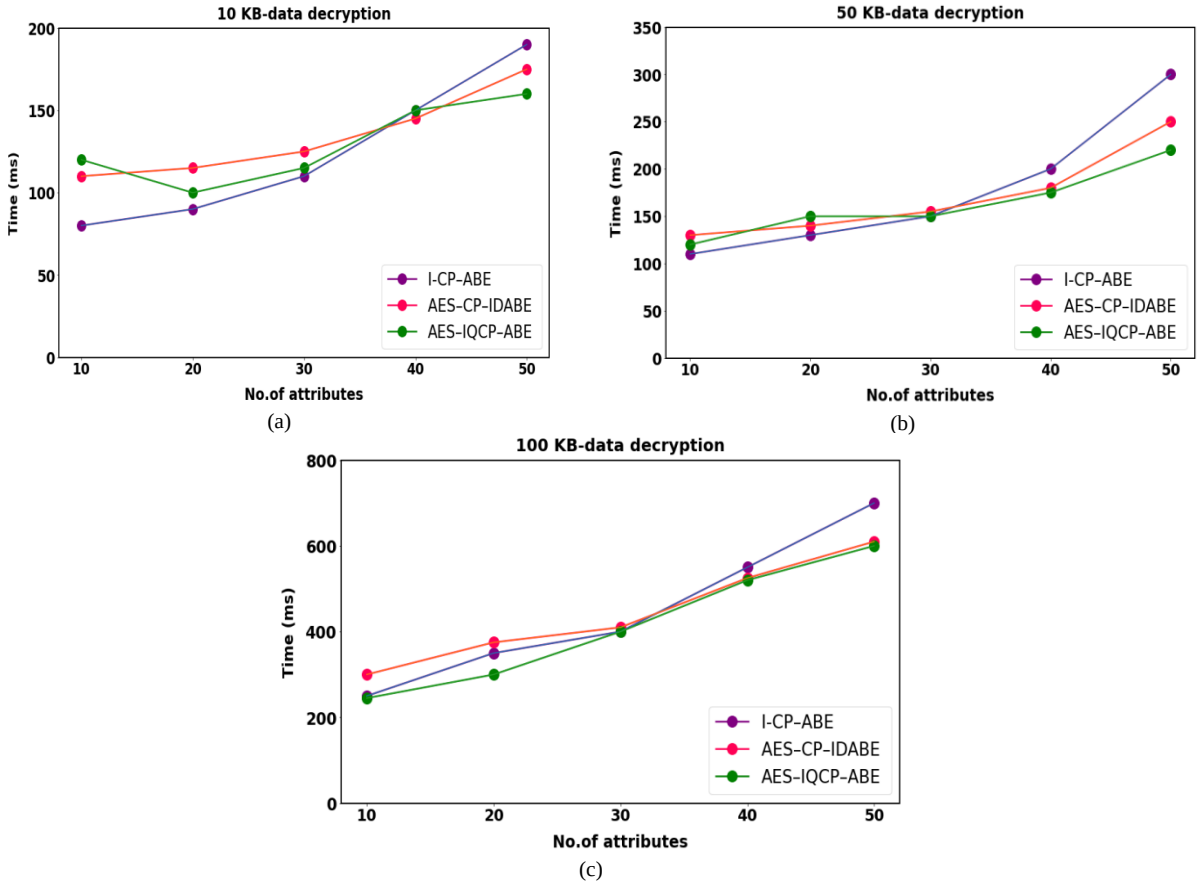

Fig.4. (a) Performance comparison for decrypting 10 KB data. (b) Performance comparison for decrypting 50 KB data. (c) Performance comparison for decrypting 100 KB data

Table 5. Comparative analysis of the intended prototype's mean computing time (ms) in comparison to more established methods

Method	Number of attributes (ms)				
	10	**20**	**30**	**40**	**50**
CPABE	692.356	768.307	727.958	751.237	829.709
FH-ABE	591.051	633.099	559.974	423.071	425.777
QKD CPABE	345.545	337.354	372.332	359.441	336.597
AES-IQCP-ABE	223.568	286.425	267.743	245.456	214.742

Table 6 shows the Computational time comparison analysis of proposed and prior cloud security models for key generation. The performance analysis for the process of key generation for computational time is shown in Fig. 6. For dynamic key generation, the computational time of the proposed model is more effective than the previous cloud data security approaches. It takes 1987.9ms to generate key for 50 attributes. It improved 38.57%, 33.67%, 27.99%, and 10.14% better performance than ABE, CP-ABE, KP-ABE, and QKD-CPABE methods, respectively. By utilizing the

benefits of Quantum Key Distribution, the suggested AES-IQCP-ABE technique produces keys at more quickly. In order to ensure that keys are generated and distributed on time, delays related to key distribution can be minimized through the use of QKD and carefully thought out communication protocols. Additionally, the approach supports key generation scalability, making it more effective in handling fluctuating loads, particularly when the number of attributes increases. These elements help to create keys in a simpler and more efficient manner, which leads to quicker key availability and better overall performance.

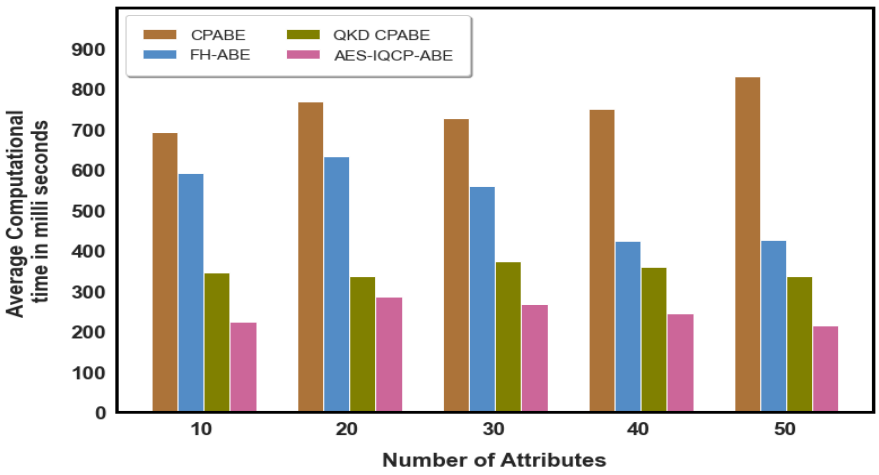

Fig.5. Comparison analysis of proposed and prior cloud security models for average computational time

Table 6. Performance comparison of the proposed and prior cloud security techniques for the total number of user attributes and chaotic-centered dynamic key production time (ms).

Method	Number of attributes (ms)				
	10	20	30	40	50
ABE	4435.54	4043.24	3731.25	3475.93	3307.47
CP-ABE	3937.9	3767.17	3339.3	3108.91	3170.75
KP-ABE	3766.17	3336.36	3222.08	2927.51	2776.32
QKD-CPABE	3217.73	3067.63	2568.58	2292.16	1997.24
AES-IQCP-ABE	3022.44	2362.65	24221.54	1995.79	1987.9

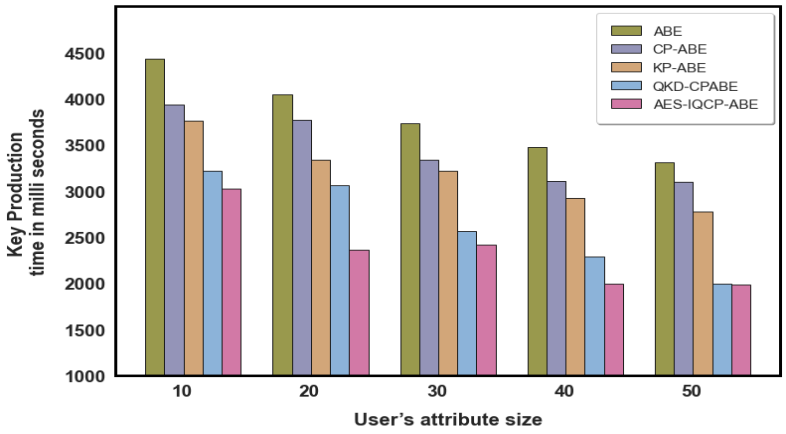

Fig.6. Comparison analysis of proposed and prior cloud security models for the number of attributes and dynamic key generation time

Additionally, the recommended solution employs minimal communication/transmission overhead/cost and has highly randomized arbitrary/ changeable key sizes compared to typical methods. The stated standards have greater exactness, as shown by real-time simulated results. In terms of bit change, it attained over 90% accuracy. For dynamic key generation, encryption time, and decryption time, it was achieving over 95% accuracy compared to state-of-the-art models.

4.4. Advantages of the Proposed Model

- Media- or text-based high-dimensional datasets are handled by this approach.
- On massive databases, the time required for encryption and decryption is effectively reduced by this model.

- 3D unstructured medical image formats like NII and HDR are supported by this model.

4.5. Limitations of the Proposed Model

- Large computational memory is needed if the data is more than 1 GB.
- This approach's integrity computation and encryption stages require the initialization of static parameters.

5. Conclusions

In the context of cloud computing, this study emphasizes the benefits of using AES-IQCP-ABE dual stage cryptography to secure files. In order to enhance the security of this framework, the suggested method incorporates QKD and the dynamic chaotic map function to safeguarding user attributes throughout the key initialization, data encryption, and data decryption stages. Several investigations are employed to assess the efficiency of the suggested methodology, and the results of the experiments demonstrate the effectiveness and dependability of the proposed framework. The proposed approach outperformed the current ones in terms of encryption time, decryption time, computational time, and key generation time, with improvements of 21.72%, 23.56%, 49.53%, and 27.59%, respectively. In addition, this suggested solution uses less transmission cost and has strong randomly assigned key sizes in comparison to conventional methods. This shows how the recommended method creates a robust and adaptable security system that aligns with the evolving requirements of cloud computing and effectively addresses security issues in cloud environments. In the future, this work is extended to establish a role-based advanced design for a multi-user cloud environment. Furthermore, to handle both threat detection and data security, we intend to combine cryptographic techniques with an intrusion detection system.

References

[1] S. Kumaresan and V. Shanmugam, “Time-variant attribute-based multitype encryption algorithm for improved cloud data security using a user profile,” The Journal of Supercomputing, vol. 76, pp.6094-6112, 2020. https://doi.org/10.1007/s11227-019-03118-8

[2] F. Thabit, S. Alhomdy, A.H. Al-Ahdal and S. Jagtap, A new lightweight cryptographic algorithm for enhancing data security in cloud computing,” Global Transitions Proceedings, vol. 2, no. 1, pp.91-99, 2021. https://doi.org/10.1016/j.gltp.2021.01.013

[3] P. Sharma, R. Jindal and M.D. Borah, “Blockchain-based cloud storage system with CP-ABE-based access control and revocation process,” The Journal of Supercomputing, pp.1-29, 2022. https://doi.org/10.1007/s11227-021-04179-4

[4] K.K. Singamaneni, K. Ramana, G. Dhiman, S. Singh, and B. Yoon, “A novel blockchain and Bi-linear polynomial-based QCP-ABE framework for privacy and security over the complex cloud data,” Sensors, vol. 21, no. 21, pp.7300, 2021. https://doi.org/10.3390/s21217300

[5] K.R. Sajay, S.S. Babu and Y. Vijayalakshmi, “Enhancing the security of cloud data using a hybrid encryption algorithm,” Journal of Ambient Intelligence and Humanized Computing, pp.1-10, 2019. https://doi.org/10.1080/08839514.2021.1991661

[6] B. Seth, S. Dalal, D.N. Le, V. Jaglan, N. Dahiya, A. Agrawal, M.M. Sharma, D. Prakash and K.D. Verma, “Secure Cloud Data Storage System Using Hybrid Paillier–Blowfish Algorithm,” Computers, Materials & Continua, vol. 67, no. 1, 2021. DOI:10.32604/cmc.2021.014466

[7] D. Rayappan and M. Pandiyan, “Lightweight Feistel structure-based hybrid-crypto model for multimedia data security over uncertain cloud environment,” Wireless Networks, vol. 27, pp.981-999, 2021. https://doi.org/10.1007/s11276-020-02486-x

[8] I.G. Amalarethinam and H.M. Leena, “Enhanced RSA algorithm with varying key sizes for data security in the cloud,” In 2017 World Congress on Computing and Communication Technologies (WCCCT) (pp. 172-175). IEEE, 2017, February. DOI: 10.1109/WCCCT.2016.50

[9] S. Hussaini, “Cyber security in the cloud using blowfish encryption,” Int. J. Inf. Technol. (IJIT), 6(5), 2020. DOI: 10.1002/9781119488149

[10] D. Commey, S. Griffith and J. Dzisi, “Performance comparison of 3DES AES, blowfish, and RSA for dataset classification and encryption in cloud data storage. International Journal of Computer Applications, vol. 177, no. 40, pp.17-22, 2020. Doi: 10.5120/ijca2020919897

[11] K.K. Singamaneni and P.S. Naidu, “An efficient quantum hash-based CP-ABE framework on cloud storage data,” International Journal of Advanced Intelligence Paradigms, vol. 22, no. 3-4, pp.336-347, 2022. https://doi.org/10.1504/IJAIP.2022.124317

[12] K.K. Singamaneni, A. Juneja, M. Abd-Elnaby, K. Gulati, K. Kotecha and A.S. Kumar, “An Enhanced Dynamic Non-linearNon-linear Polynomial Integrity-Based QHCP-ABE Framework for Big Data Privacy and Security,” Security and Communication Networks, pp. 1-20, 2022. https://doi.org/10.1155/2022/4206000

[13] H. Wang, J. Liang, Y. Ding, S. Tang and Y. Wang, “Ciphertext-policy attribute-based encryption supporting policy-hiding and cloud auditing in smart health. Computer Standards & Interfaces, vol. 84, pp.103696, 2023. https://doi.org/10.1016/j.csi.2022.103696

[14] Y. Jiang, X. Xu and F. Xiao, “Attribute-based encryption with blockchain protection scheme for electronic health records,” IEEE Transactions on Network and Service Management, pp. 1-20, 2022. DOI: 10.1109/TNSM.2022.3193707

[15] Y. Zuo, Z. Kang, J. Xu and Z. Chen, “BCAS: a blockchain-based ciphertext-policy attribute-based encryption scheme for cloud data security sharing,” International Journal of Distributed Sensor Networks, vol. 17, no. 3, p.1550147721999616, 2021. https://doi.org/10.1177/1550147721999616

[16] K. K. Singamaneni and P.S. Naidu, “IBLIND Quantum Computing and HASBE for Secure Cloud Data Storage and Accessing,” Rev. d'Intelligence Artif., vol. 33, no. 1, pp.33-37, 2019. https://doi.org/10.18280/ria.330106

[17] C. Huang, S. Wei and A. Fu, "An efficient privacy-preserving attribute-based encryption with hidden policy for cloud storage," Journal of Circuits, Systems and Computers, vol. 28(11), pp.1950186, 2019. https://doi.org/10.1142/S021812661950186X
[18] S. Chandel, G. Yang and S. Chakravarty, "RSA-CP-IDABE: a secure framework for multi-user and multi-owner cloud environments. Information, vol. 11, no. 8, pp.382, 2020. https://doi.org/10.3390/info11080382
[19] J. Wei and L. Zhang, "Ciphertext policy attribute-based encryption with hierarchical domain authorities and users in the cloud," International Journal of High-Performance Computing and Networking, 10(6), pp.524-533, 2017. https://doi.org/10.1504/IJHPCN.2017.087470
[20] M. Bakro, S.K. Bisoy, A.K. Patel, and M.A. Naal, "Hybrid blockchain-enabled security in cloud storage infrastructure using ECC and AES algorithms," In Blockchain-based Internet of Things 1, pp. 139-170, 2022. Singapore: Springer Singapore. https://doi.org/10.1007/978-981-16-9260-4_6
[21] P. Chinnasamy, S. Padmavathi, R. Swathy and S. Rakesh, "Efficient data security using hybrid cryptography on cloud computing," In Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020 1, pp. 537-547, 2021. Springer Singapore. https://doi.org/10.1007/978-981-15-7345-3_46
[22] S. Rehman, N. Talat Bajwa, M.A. Shah, A.O. Aseeri and A. Anjum, "Hybrid AES-ECC model for the security of data over cloud storage," Electronics, 10(21), p.2673, 2021. https://doi.org/10.3390/electronics10212673
[23] G. Verma and A. Kumar, "Novel quantum key distribution and attribute-based encryption for cloud data security," Concurrency and Computation: Practice and Experience, pp.e7700, 2023. https://doi.org/10.1002/cpe.7700
[24] D. K. M. Hodowu, D. R. Korda and E. D. Ansong, "An enhancement of data security in cloud computing with an implementation of a two-level cryptographic technique using AES and ECC algorithm," Int. J. Eng. Res. Technol, vol. 9, pp. 639-650, 2020.
[25] K. Sundar, S. Sasikumar and C. Jayakumar, "Enhanced cloud security model using QKDP (ECSM-QKDP) for advanced data security over cloud," Quantum Information Processing, vol. 21, no. 3, pp.115, 2022. https://doi.org/10.1007/s11128-022-03452-6
[26] K.K. Singamaneni, A. Nauman, S. Juneja, G. Dhiman, W. Viriyasitavat, Y. Hamid and J.H. Anajemba, "An Efficient Hybrid QHCP-ABE Model to Improve Cloud Data Integrity and Confidentiality," Electronics, vol. 11, no. 21, pp.3510, 2022. https://doi.org/10.3390/electronics11213510

Authors' Profiles

Jayaprakash Jayachandran completed his B. Tech in Information Technology in the year 2003 from Sri Manakula Vinayagar Engineering College (Pondicherry University), Pondicherry and completed M. Tech in Information Technology in the year 2012. He has 17 Years of teaching experience and working in Karaikal Government Polytechnic College, Karaikal. His area of interest includes Cloud Computing, Networking, Cyber Security and Multimedia. He received various funds from Central sponsored Schemes and State Government Projects. He is a certified Microsoft Certified Profession and active Life member of ISTE. He is a certified trainer for Entrepreneurship Development under NEISBUD. His current area of research interest includes Cryptography and Cyber Security.

Dr. Dahlia Sam is a Professor of Department of Information Technology at the Dr. MGR Educational and Research Institute, Chennai, India. She completed her PhD. in Computer Science Information Systems from Dr. MGR Educational and Research Institute, her areas of interest are Artificial Intelligence, Machine Learning, Image Processing, Deep Learning.

Dr. Kanya Nataraj is a Professor of Department of Information Technology at the Dr. MGR Educational and Research Institute, Chennai, India. She completed her Ph. D. in Computer Science Information Systems from Manonmaniam Sundaranar University, her areas of interest are Machine Learning, Data Science, and Network Security.

How to cite this paper: Jayaprakash Jayachandran, Dahlia Sam, Kanya Nataraj, "Efficient Cloud Computing Security Using Hybrid Optimized AES-IQCP-ABE Cryptography Algorithm", International Journal of Computer Network and Information Security(IJCNIS), Vol.17, No.2, pp.101-114, 2025. DOI:10.5815/ijcnis.2025.02.07