ROBUST AND SCALABLE SYSTEM FOR REMOTE DATA AUTHENTICATION IN CLOUD ENVIRONMENTS

A PROJECT REPORT

Submitted by

DANUSH GUPTA V K

MOHAMED NOUFHAL ABBAS K

ANTONY GUNAL P

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER TECHNOLOGY

MADRAS INSTITUTE OF TECHNOLOGY CAMPUS

ANNA UNIVERSITY : CHENNAI 600 044

MAY 2023

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**ROBUST AND SCALABLE SYSTEM FOR REMOTE DATA AUTHENTICATION IN CLOUD ENVIRONMENTS**" is the bonafide work of "**DANUSH GUPTA V K (2019503012), MOHAMED NOUFHAL ABBAS K (2019503534)** and **ANTONY GUNAL P (2019503006)**" who carried out the project work under my supervision.

SIGNATURE

Dr. P. JAYASHREE

HEAD OF THE DEPARTMENT

Professor

Department of Computer Technology

Madras Institute of Technology

Chromepet, Chennai 600 044

SIGNATURE

Dr. S. MUTHURAJKUMAR

SUPERVISOR

Associate Professor

Department of Computer Technology

Madras Institute of Technology

Chromepet, Chennai 600 044

ABSTRACT

In recent years, cloud services have become increasingly popular because they offer a wide range of advantages, including concurrent resources, automation, scalability, and remote access. The development of the cloud computing concept has caused the appearance of cloud storage as an innovative type of storage. Using cloud storage has many benefits, like dependability and efficiency, but it also introduces new risks to the security and confidentiality of one's private data. A public cloud lacks the extensive data control, security, and network options. That lessens the appeal of using cloud services for the general public, which amplifies issues with data security and privacy. When creating secure communication in the cloud environment, the level of service quality, including protection, dependability, and efficiency is taken into account. In order to prevent unauthorized users from registering and using the data, we require a system of authorization enabling the cloud storage platform's data access. Data access therefore requires adaptable, secure, and multi-level systems. The proposed work implements a simple and power-saving mechanism for access of data and data exchange via a cloud storage environment. It provides authenticated data access, including the introduction of the password and smart based authentication method. It enables the user to be uniquely identified. A multilevel authentication based data access system is introduced to enhance the security.

ACKNOWLEDGEMENT

We are highly indebted to our respectable Dean, **Dr. J. PRAKASH** and to our reputable Head of the Department, **Dr. P. JAYASHREE**, Department of Computer Technology, MIT, Anna University, for providing us with sufficient facilities that contributed to the success of the project in this phase.

We would like to express our sincere thanks and deep sense of gratitude to our supervisor, **Dr. S. MUTHURAJKUMAR**, for his valuable guidance, suggestions and constant encouragement.

We sincerely thank our project panel members **Dr. P. JAYASHREE**, **Dr. R. KATHIROLI** and **Dr. V. P. JAYACHITRA** for their valuable suggestions and their different views on our project.

Finally, we extend our sincere thanks to all the faculty members of the Department of Computer Technology, friends, family and everyone who have rendered their valuable help in completing this project successfully.

DANUSH GUPTA V K (2019503012)

MOHAMED NOUFHAL ABBAS K (2019503534)

ANTONY GUNAL P (2019503006)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IT	Information Technology
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
CRM	Customer Relationship Management
2FA	Two Factor Authentication
CP-ABE	Ciphertext - Policy Attributes
XOR	Exclusive OR
ECC	Elliptic Curve Cryptography
ABE	Attribute - Based Encryption
ROR	Real Or Random
ESS	Encryption-box Security System
PEKS	Public Encryption Key System

CHAPTER 1
INTRODUCTION

1.1. OVERVIEW

Cloud services, due to their many benefits, such as concurrent resources, automation, scalability, and remote access, have grown in popularity over the past few years. The whole data can be easily and economically accessed from anywhere in the globe if one stores it all in the cloud. The online distribution of computing resources, including programmes, storage settings, networks, and processing power, is the basis of the cloud computing idea. A new type of storage called cloud storage has emerged as a result of the development of the cloud computing idea.

While using cloud storage has numerous advantages, such as dependability and effectiveness, it also poses new security and confidentiality concerns to one's personal information. The vast network, data management, and confidentiality options are absent from a public cloud. This system makes using cloud services less appealing to the general public, which makes problems with data privacy and confidentiality worse. The degree of services, including confidentiality, dependability, and efficiency, is taken into consideration when establishing communication that is secured in a cloud system. Additionally, private clouds enable enterprises to design their own security measures without having to rely on cloud provider security guidelines.

The cloud storage system needs to be protected from the misleading threat of an insider threat in addition to the security protocols. An authentication technique is needed for using the cloud storage system's data access interface to prevent unauthorized people from enrolling and utilising the information. For protecting

breaches of information in the cloud computing environment, only the password and username are ineffective. Data access therefore essentially needs adaptable, multi-level and secure systems.

1.2. CLOUD COMPUTING

The usage of computing resources by organizations and individuals is being revolutionized by the rapidly developing technology known as cloud computing. Rather than being hosted locally on a user's computer or server, the cloud computing paradigm of computing involves delivering resources, such as processing capacity, storage, and applications, to users over the internet as needed.

Some of the main advantages of cloud computing are:

- Scalability: Users can quickly scale up or down their processing capabilities with cloud computing as needed. As a result, organizations don't need to make expensive hardware upgrades in order to swiftly and easily change their IT infrastructure to accommodate changing demand.
- Cost savings: By removing the need for cost-intensive hardware and software purchases and maintenance, cloud computing can help businesses save money. Instead, consumers pay on a pay-per-use basis for the computer resources they actually use.
- Accessibility: With cloud computing, users can utilise their applications and information from any location in the world as long as they have an internet connection. This facilitates remote working for employees and enterprises operating from different places.

- Security: Cloud providers often have strong security measures in place to guard against dangers like hacking and data breaches and to safeguard the data and applications of their users. Cloud providers frequently have more advanced security measures than what a single company or user could put in place on their own.

Cloud computing services come in a variety of forms, including IaaS, or Infrastructure as a Service, which gives consumers on-demand access to computer infrastructure, including servers, storage, and networking. PaaS, or platform as a service, gives users a platform for creating, running, and maintaining their own apps without having to worry about the supporting infrastructure. SaaS or Software as a Service, gives consumers access to pre-built programmes like email, Customer Relationship Management (CRM), and accounting software.

1.3. AUTHENTICATION

The act of verifying a user's or system's identity is referred to as authentication. Since it ensures that only those with authorization are able to view confidential information or resources, it is an essential part of security.

There are various kinds of authentication techniques. The most popular type of authentication is password-based, which requires users to input a username and password to log into a system or programme. Users ought to be obligated to change their passwords on a regular basis, and they should be strong and complex. For access, clients must provide two separate kinds of identification resources using the more secure two-factor authentication (2FA). This can be something the user has (like a security token or mobile phone) or something they know (like a password).

Biometric identity verification uses physical characteristics consisting of fingerprints, recognition of facial features, or recognition of voice to confirm a user's identity. Although this is a very secure method of authorisation, it can be costly to install and cause privacy issues. Digital certificates are used in certificate-based authentication to confirm the identity of individuals or systems. Online banking and e-commerce applications frequently employ this safe method of authentication.

To prevent unauthorized access and safeguard sensitive information, systems and applications must employ strong authentication techniques and frequently review and update their security procedures. The best practices for creating and managing passwords should also be explained to users, and they should be urged to utilise 2FA or other stronger forms of authentication wherever available.

1.4. ENCRYPTION

To prevent unauthorized access or theft, plain text or data must be encrypted and transformed into ciphertext, a coded or jumbled form. Modern data security must include encryption, which is used to safeguard sensitive information like financial information, personal information, and trade secrets. Using a mathematical method, encryption transforms plaintext into ciphertext. An algorithm is used to scramble the data using a key, which is a string of characters. The key must be used to undo the encryption process and turn the ciphertext back into plain text in order to decrypt the data.

1.4.1 BCRYPT SALT ENCRYPTION

A popular cryptographic algorithm for hashing and storing passwords is bcrypt. The algorithm produces a secure hash of the password using a "salt" value, rendering it difficult for intruders to determine the password using a rainbow table attack. Before the password is hashed, a random string of characters is added as a salt. Even if an identical password is used by two people, this random salt value guarantees that their password hashes will differ, rendering it difficult for intruders to determine the password. When a user tries to log in, the salt value can be used to verify the password because it is stored in the database with the password hash.

A popular and safe password hashing method, bcrypt has undergone rigorous testing and been shown to be impervious to attacks. Password hashing is just one part of password security, so organizations should also take other steps to protect the safety of their systems and user data, including robust password guidelines, multiple-factor authorization, and frequent security audits.

1.4.2 IMAGE - BASED INFORMATION HIDING ENCRYPTION

Image - based information hiding is a method for concealing data inside of images without changing how they appear to be. Image steganography tries to make it more difficult for unauthorised users to access the secret information by concealing its existence. The fundamental idea underlying it is to hide information in the least significant bits of an image's pixel values. This is so that the least important elements of a pixel value can be changed without affecting how the image looks as a whole. The procedure can insert concealed data into the image by altering the least significant bits.

The capacity to conceal a significant quantity of the data as part of an image and the recognition that the presence of a concealed message is not readily apparent are two benefits of this technique.

CHAPTER 2
LITERATURE SURVEY

2.1 AUTHENTICATION SCHEMES

Sultan *et al.*, [16] offered a system that showed a safe inter-cloud authorisation system that used encryption with ciphertext-policy attributes (CP-ABE). Using web apps that perhaps registered with a different service provider, the proposed approach enabled the ability for data owners to access files stored on service provider-managed cloud storage systems. Users of a web application can access the stored files by using one-time access tokens that the data owner can issue. The protocol did not provide authentic data access that enables strong verification of the legitimate clients prior to doing the different data activities, such as storing, updating, and sharing data.

Tiwari *et al.*, [4] have proposed a scheme with an authorization strategy based on biometrics for secure access to data exchange and storage. The use of a proxy re-encryption key by the cloud server and the production of an authentication token by the data master during decryption to restrict user accessibility and enable adaptive sharing of information under the management of a data administrator. It has been noted that it was created utilising intricate cryptographic processes, which have a high cost for computing and communication and make them unfeasible for use in environments with limited resources.

Roy *et al.*, [20] proposed a method that offered mobile user authorization, safe exchange of keys, confidentiality of users, and untraceability features in a dispersed portable cloud computing environment. The protocol used fuzzy extractor operations, bitwise XOR operations, and efficient one-way cryptographic hash functions. The cost of energy consumed using their protocol was found out to be very high and not reliable for use in the pacing environment.

Zhou *et al.*, [13] have elucidated an authentication method for cloud servers and IoT-based systems. In order to achieve the highest efficiency, the authentication strategy used crypto-modules like the one-way hash function. The major limitation of the proposed method was the computation cost and the storage cost required to implement the protocol. Even though the energy consumed was comparatively less, the storage and computation costs were relatively high.

Mo *et al.*, [11] have proposed a model that suggested an efficient secure two-factor anonymized authorization for users mechanism. In addition to offering bidirectional authorization between cloud computing and mobile devices, the suggested system also satisfied established security evaluation standards. The research offered a robust and effective key agreement for MCC and ID-based ECC with anonymised two-factor authorization without pairing. The suggested approach had a drawback as the energy consumed was very high compared to the other proposals. Also the computation cost on the user side was very high.

Junejo *et al.*, [1] proposed a scheme where any entity that needed to be a member of a Fog-IoT system should initially register with the organization based on a set of qualities in the proposed secure integrated framework. In an Attribute Based Encryption (ABE) scheme, the attributes were also used to produce the secret keys that are used to encrypt the data. The Server Client (SC) implemented an ABE scheme that is based on Elliptical Curve Cryptography (ECC) in order to enforce strong authentication and access control in Fog-IoT systems. The suggested strategy had certain drawbacks because it only employed a single factor authentication technique. Moreover, registration was required for each person, which was an unsuitable manual process.

Sharaf *et al.*, [21] presented a reliable cloud-based Electronic Health Record (EHR) system that ensured the privacy and security of health information stored in the cloud, depending on tiered multi-authority CP-ABE to implement access control regulations. The system allowed healthcare professionals, patients, and patients to share EHRs at a high level of integration, interoperability, and sharing. The suggested plan avoided using the conventional encryption method because it was inappropriate for the cloud environment.

Srinivas *et al.*, [12] had devised a method which used a novel user identification method for secured medical information authentication in the cloud. Both the user and wearable sensor node established a secret session key that was used for subsequent secure communications after successful mutual authentication between the two parties. The proposed approach offered the session-key security and shielded active attacks, according to formal security analysis based on the Real

or Random (ROR) model and formal security verification based on the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool. Although the overall computation cost was lower than any other current schemes' costs, it was still required to provide superior performance and security.

Zhong *et al.*, [10] had presented a mutual authentication and key agreement technique based on elliptic curve certificate - free cryptography for peer-to-peer cloud, along with an effective transfer of information model between cloud service providers. In order to preserve the privacy of service providers and customers, the method also leveraged server anonymity. To track rogue cloud servers, the technique offered identity traceability. The proposed work blocked multiple users from sharing data across different cloud servers.

Chaudhary *et al.*, [19] had developed a brand-new, lattice-based strong encryption system for smart healthcare that, at various points, utilised both a light-weight key transfer and a method of authentication. A mutual authentication system based on lattices was created to validate requests made to multiple end users and cloud storage. For the purpose of transferring data between numerous users and cloud storage, a data encryption system was developed. For the purpose of issuing permissions to end users, an access rights verification system was created. Most of the system-reported solutions rely heavily on computing to secure smart healthcare data, which may not be appropriate given the limited resources of contemporary smart devices.

Akram *et al.*, [15] had proposed an ECC-based three-factor user authorised key agreement system that was cost-effective to implement and withstood the main security concerns like internal threat, impersonator attack, and credential alteration assaults. Additionally, it used a random oracle model to formally analyse the proposed scheme's security. The suggested scheme had somewhat more communication bits than the related methods.

Amin *et al.*, [6] had presented an architecture suitable for dispersed cloud environments, and on its foundation, a smartcard-based verification protocol had been suggested, allowing the registered user to safely access all private data from all the private cloud servers. The system made use of the BAN logic model and AVISPA tool to improve the suggested protocol. Furthermore, an informal cryptanalysis verified that the protocol is safe from all security risks. Although the protocol was designed to be lightweight, it may still introduce additional overhead and complexity that can negatively impact the efficiency of the system.

Fang *et al.*, [14] had proposed an evolutionary game-based approach for collaborative authorization. Based on the evolutionary game model, they conducted game theoretical analyses. Also they designed an ESS-based algorithm and conducted experiments both on simulated as well as real datasets. The results of the proposed strategy were efficient at an acceptable level of resource consumption. The drawback of the proposed system was that only homogeneous neighbouring nodes were discussed. Unquestionably, much more organised explorations are necessary to study heterogeneous neighbouring nodes.

Yoon *et al.*, [25] had proposed a mobile network improved lightweight authorization with key-agreement protocol based on smart cards and elliptic curve encryption. Because it completed a straightforward one-way hash function, authentication of the message code, and exclusive-OR operation, the suggested protocol was small and appropriate for real-time applications. Concatenation and an exclusive-OR function required relatively little work, whereas elliptic curve point multiplication, symmetrical operations, and asymmetrical operations do.

Ghaffar *et al.*, [15] had presented an architecture where symmetric encryption was used in the proposed protocol to control and define the accessibility of authorized users as well as to securely distribute data to users. The symmetric key encryption technique also included the file's contents into the metadata to ensure data protection. In order to provide safe encrypted data access, the suggested protocol implements a symmetric decryption method. It has been noted that they were created utilising complicated cryptographic processes, which have a high cost for computing and communication and make them unfeasible for use in environments with limited resources. Furthermore, some protocols prevented them from providing common security features.

2.2 ENCRYPTION TECHNIQUES

Harn *et al.*, [18] had proposed Hilbert Curve Encoding which was used to encrypt the user's message and the locations for processing privately by the dependable authority. The system also suggested hybrid H N G M N encoding, which merged Grey encoding and the Hilbert curve encoding. According to the

experiment's findings, in terms of user reaction time, token remaining percentage, and execution time, the encoding approaches were superior than Hierarchical Encoding and comparable to Grey Encoding. The major drawbacks of the work was it required High memory management and High Complexity. The work does not support address grid cells and several spatial methods are used, such as 2D range trees.

Shen *et al.*, [22] had presented Alperin-Sheriff and Peikert's straightforward and tight noise analysis method which was used to create the first identity-based fully homomorphic encryption scheme from identity-based encryption and lattice-based cryptography. Additionally, it created a successful multi-identity completely homomorphic encryption technique by expanding a brand-new ciphertext under a single identity key to an expanded one under a combination key, allowing ciphertexts under many identities to be homomorphically evaluated. The fact that the suggested schemes were only levelled homomorphic systems like the GSW-IBFHE scheme was a negative.

Amalarethinam *et al.*, [3] had proposed a brand-new Magic Rectangle-based picture encryption technique. According to the procedure, the simple image was divided into single-byte blocks, and the block was then substituted for the value of MR. Additionally, the user chose at random the Magic Rectangle control parameters. After that, the image was encrypted using public key cryptography techniques like RSA, etc. The model improved memory efficiency and added a layer of security to the public key technique. The work's main flaw was that it took more time to construct the magic rectangle and compress the photographs.

Zhang *et al.*, [23] had proposed SPADE, a method of encrypted data deduplication that did not rely on vulnerable key servers and released clients from the key administration issue. The paper proposed a password - based stacked encoding system and a password - based authorization system and incorporated them into SPADE to allow users to gain access to their information only using the passwords. The protocol was designed to prevent dictionary guessing attacks. Without independent key servers, the suggested system cannot provide the same functionality as SPADE with the same security assurance. The key management issue was incompatible with the suggested password - hardening approach.

Deng *et al.*, [7] had proposed a model by combining two encryption techniques, namely identity - based encryption and identity - based broadcast encryption, an identity - based encryption transformation model is created. The scheme provided the following attractive features such as Identity-Based Data Storage, Cross - Domain Encryption Transformation and Strong Security Guarantee. The drawback of this proposed approach was it introduced additional complexity to the encryption and decryption process, which can increase the likelihood of errors and make the system more difficult to implement and manage.

Munir *et al.*, [17] had carried out a few cryptographic attacks to extract the key from the auxiliary cryptosystem. Using a chosen - plaintext attack and a single known plaintext ciphertext combination, the key was quickly and efficiently recovered. The diffusion-based encryption algorithm's weakness was shown by the attacks' minimum execution times. The only flaw in the proposed strategy was the

diffusion process. The initial system that was put out included conditional shift, hamming distance, and chaos implementation. However, the bitwise XOR and hamming distance operations had the same behaviour, ending the effects of one another after diffusion.

Chang *et al.*, [2] had proposed a method of cancellable multi-biometric authentication. Utilising a secret key produced from another biometric model, a revolutionary bitwise encryption method converted the biometric template into a secure copy. Also the paper has introduced algorithms for bitwise encryption which were defined over keyed-hash function and block cipher based encryption. The method reduced recognition accuracy compared to traditional biometric authentication systems, as the biometric data was transformed in a way that made it less distinct and more prone to errors.

Li *et al.*, [8] had presented using a public key tree (PKTree) and a PEKS scheme, a PEKS version known as Hierarchical Public Key Encryption with Keyword Search (HPEKS) that enabled a semi-generic design. The approach was based on a more advanced HPEKS method with improved security that combined symmetric and public key encryptions. The security of the work was analysed using the Random Oracle model. The major drawback was that it was not faster than the previous schemes like BSS-PKE/PEKS scheme and HL-PEKS scheme. The computational cost of the DHPEKS scheme's AES encryption technique was not taken into account in the trial results because it was unpredictable and influenced by the size of the plaintext.

Xiong *et al.*, [9] had proposed a comprehensive analysis of three representative Public - key encryption methods that included search capabilities were plaintext - checkable encryption, public - key encryption with equality testing, and public - key encryption with keyword search. In order to aid both novice and experienced researchers in understanding the schemes, the paper presented it from a variety of angles. The drawbacks of the method was that it was computationally intensive, especially when dealing with large amounts of data which can result in increased latency and reduced system performance.

Zhou et al., [24] created an accessible through search public-key encryption with cryptographic inverse firewalls and was implemented using the JPBC library. Lacking protected channels, the protocol can withstand a selected keyword assault and an algorithmic replacement attack. The protocol has a large cost and transmission benefit and is immune to malevolent insider attacks in cloud environments using the KGA and ASA. In order to test whether the IBE-based/CL-PKE-based PEKS method can be effective against a randomly selected ciphertext attack, the scheme did not employ the CRF to resist exfiltration attempts.

CHAPTER 3
PROPOSED SYSTEM OF REMOTE DATA AUTHENTICATION IN CLOUD ENVIRONMENTS

3.1. OBJECTIVE

The main goal of the following work is to implement a simple and power-saving mechanism for utilising a cloud storage environment to access and share data. To provide authenticated data access, the work has the introduction of the password and smart based authentication method. It enables the client to be uniquely identified. A system is developed to enhance the security by providing a multilevel authentication based data access system.

3.2. PROPOSED SYSTEM

The proposed system can be mainly divided into 3 modules:

- Registration
- Data Access Control
- Cloud Data Storage

Registration

The data owner has the lone access to the registration panel. User ID, Password and Biometric Authentication is required for a user to register. Using the

credentials, they are encrypted by concatenating and using hash functions. Multiple hashes are performed and each value is stored in a database used for authentication by the cloud server.

Data Access Control

To access data from the cloud server, primarily login and authentication needs to be done. Using concatenation, XOR operations and Hash functions on the credentials, numerous encryptions are performed and communicated to the Cloud Server. The cloud authenticates the user and if valid decrypts the image stored in the cloud server else the session will be discarded.

Cloud Data Storage

In this module, verification is done to check if the entered user is the data administrator by entering the credentials. The credentials are concatenated and hashed to be stored and later used for validation by decrypting the encryption used. The file to be encrypted and the image to be stored are communicated to the cloud server where the encryption occurs.

3.3. ARCHITECTURE DIAGRAM

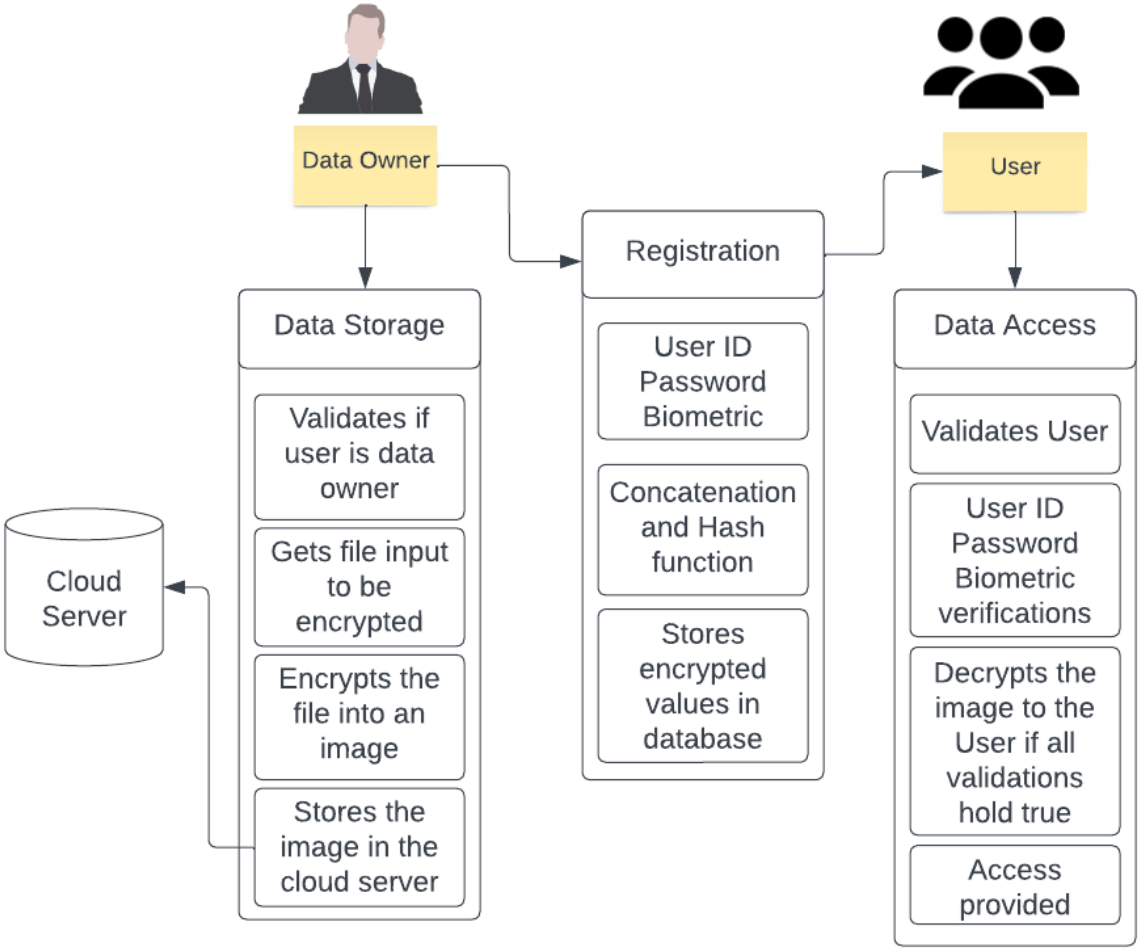

Figure. 3.1 Proposed Architecture Diagram

The two entities in the architecture design depicted in Figure 3.1 are the data administrator and the data users. The proposed system has three modules where the Data Storage model is handled by the data administrator and the user interacts with the Data Access module.

Primarily, the users need to get registered with the cloud server. The users need to get the approval from the data owner to create a login authentication for themselves. A user needs to have an username, password and fingerprint credentials for sign up. These data are encrypted using low cost schemes and stored into the database for authentication by the server.

For the data access module, the flow starts from the verification of the user, i.e. if the client is authenticated to utilize the information stored. The authentication takes place with two credentials, username and password, and an additional layer consisting of the fingerprint verification of the user. If the authentication of the user checks all the verification, the user is allowed access to get the decrypted version of the data needed.

Moving on to the data storage module, it undergoes the same set of authentication mechanisms for the validation of the data owner. After the access is granted, the data owner feeds the input of the file that needs to be stored which gets encrypted and further used in the cloud server for storage.

3.4. TOOLS REQUIRED

The major tools that are used in the development of the system are:

- Amazon Web Services (AWS)
    - The extensive platform for cloud computing offered by Amazon is called Amazon Web Services (AWS). Organisational tools like computing capacity, storage of databases, as well as content distribution services are available through AWS services.

- Google Colab (Python IDE)
    - Colaboratory, or "Colab" for short, is a product of the research arm of Google. Data analysis, instructing, and machine learning are three areas where Colab excels. Using a web browser, anyone may write and run any code written in Python. Colab is an online service for Jupyter notebooks that enables no-cost use of computer features, which includes GPUs, and doesn't require any installation.

CHAPTER 4
ALGORITHM AND IMPLEMENTATION

4.1. MODULE 1 : REGISTRATION

The registration segment is totally under the control of the admin. It is not available to the public. For the registration, the admin has to manually ride the system with the credentials of username, password and fingerprint of the approved user as shown in Figure 4.1. These credentials will be encrypted and stored in the database so that it is secure and cannot be easily hacked by trespassers as shown in Figure 4.2.

The encryption used in this phase is the BCRYPT algorithm. For user login, Bcrypt works nicely. It uses salt to encrypt byte code. The password can be made more secure by adding a certain amount of characters, known as salt, at random. A password string is encoded for encryption into machine-level bytes during the procedure. Raw strings cannot be read by Bcrypt; only bytes can. Typically, byte code is encoded using the 'utf-8' encoding. It is one of the greatest hashing algorithms since it is deliberately slick, which also serves as a security measure. A password string is less vulnerable to brute force assaults when salt is added. The genuine password can't simply be discovered and extracted from the salt by intruders.

Algorithm 4.1: BCRYPT Salt Encryption Algorithm
Input: Plaintext password, Randomly generated salt value **Output:** Encrypted password

1: Create a random salt
2: Append the salt to the password
3: Salted password hash
4: Store salted password in database

```
Pwd1 = Pwd1.encode('utf-8')
Pwd1 = bcrypt.hashpw(Pwd1, bcrypt.gensalt())
```

```
Welcome, please select an option
Login | Signup:Signup
Enter a username:danush
Create password:danush21042
Confirm Password:danush21042
Enter fingerprint:danush_left.BMP
User created successfully!
```

Figure. 4.1 Registration of User

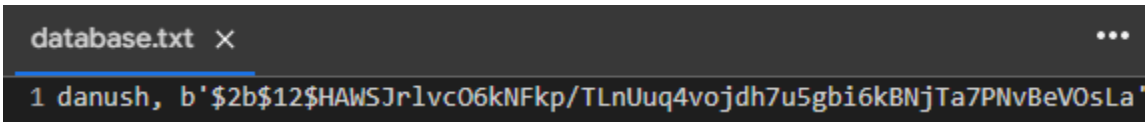

Figure. 4.2 Encrypted User Credentials

4.2. MODULE 2 : DATA ACCESS CONTROL

The second module is the data access module. This is the phase where the user interacts with the cloud server. It is more like a client - server interaction. To begin with, the user has to verify his identity to check the integrity of the client. The client has to input the respective username and password credentials as shown in Figure 4.3. The system obtains the record using the client's provided username.

Since the password is stored in an encrypted manner within the database, it needs to be decrypted and mapped with the input password. Hence, the input password would be converted to byte language and the hashed password would be passed as parameters to the checkpw function of the bcrypto module which validates and returns the boolean value.

Algorithm 4.2: BCRYPT checkpw( ) Algorithm
Input: Plaintext password to be checked, stored BCRYPT-encrypted password **Output:** Boolean value
1: Extract salt from hashed password **2:** Hash input password with extracted salt **3:** Compare the generated hash with the stored hash
`h = h.encode('utf-8')` `bcrypt.checkpw(pwd.encode(), h)`

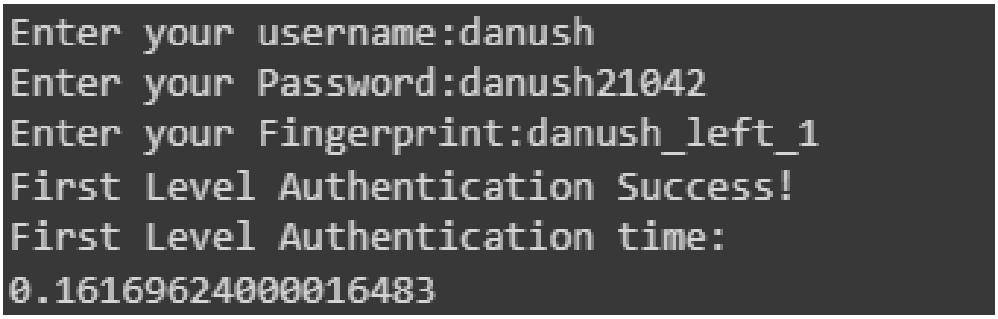

Figure. 4.3 First Level of Authentication

Moving on to the second level of authentication, biometric verification is introduced. Fingerprint authentication is the prime mode of verification which cannot be bypassed that easily. The flow diagram of the fingerprint authentication

is shown in Figure 4.4. In this phase, openCV and kNN match are being used to calculate the score of match between the stored fingerprint and the input fingerprint obtained from the user. If the score of the match is on the higher slide, access will be granted as shown in Figure 4.5.

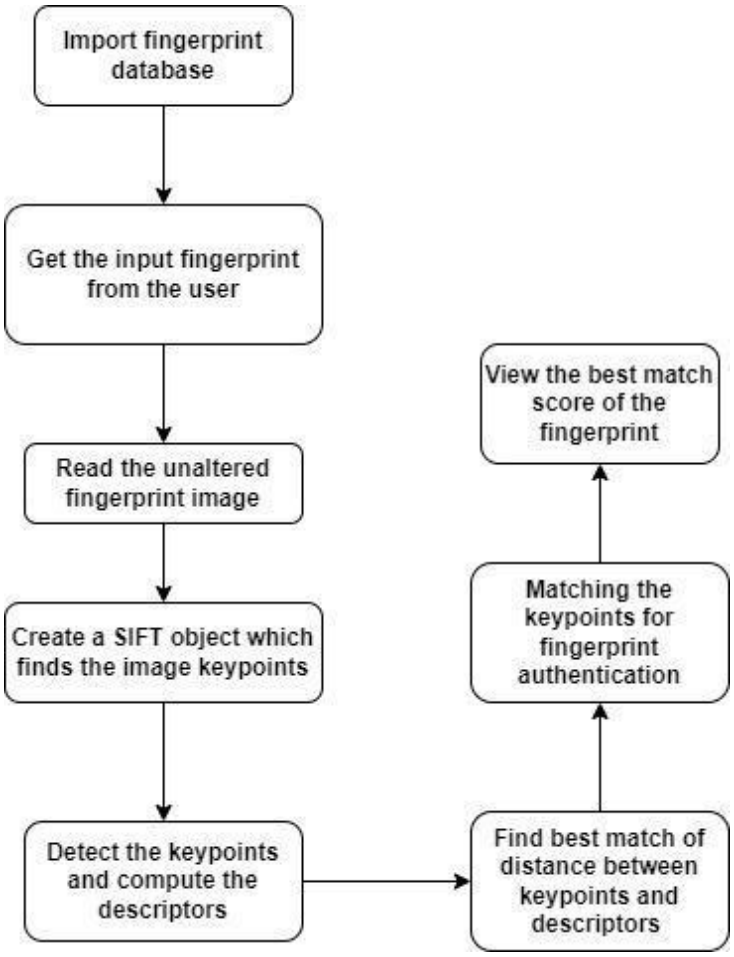

Figure. 4.4 Flow Diagram of Fingerprint Authentication

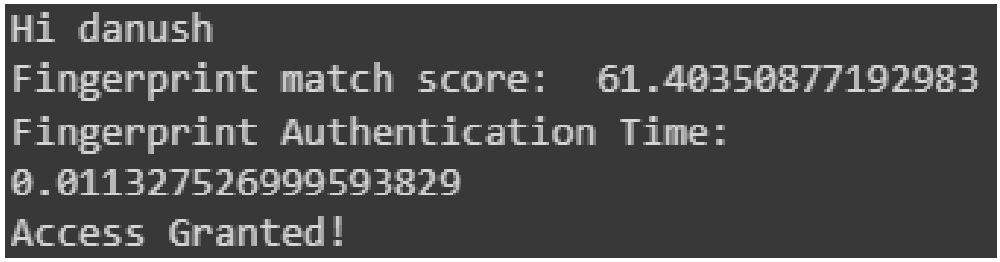

Figure. 4.5 Second Level of Authentication

Once all the user validations hold true, the user is given permission to utilise the cloud server's information. But direct access is not provided further inorder to tighten the security feature. The user will receive the data in an encrypted format so that nose pokers will not be able to get access to the files with ease.

4.3. MODULE 3: CLOUD DATA STORAGE

The third module is the cloud data storage module. This is the phase where the data owner solely interacts with the cloud server. Only the data owner has privileged access to this module. To begin with, the data owner has to verify his identity to check the integrity of the incoming request. The owner needs to enter the respective username and password credentials. An additional level of security of the fingerprint authentication is also done. Both these algorithms use the same encryption and validation algorithms as the user validation shown in section 4.2.

After all the validations hold true, the data owner is granted access to the cloud server to use. The owner can upload the data file which will be stored in an unique encrypted format in the cloud server as shown in Figure 4.7.

Image-based information hiding entails secret information being concealed in digital images in an approach that is undetectable to the naked eye. Information concealing using images aims to hide the existence of the concealed information in the image, so that it can be transmitted or stored without detection. It is commonly used for various purposes, such as digital watermarking, copyright protection and hiding malware or conducting cyber attacks. The flowchart for the image-based information hiding is shown in Figure 4.6.

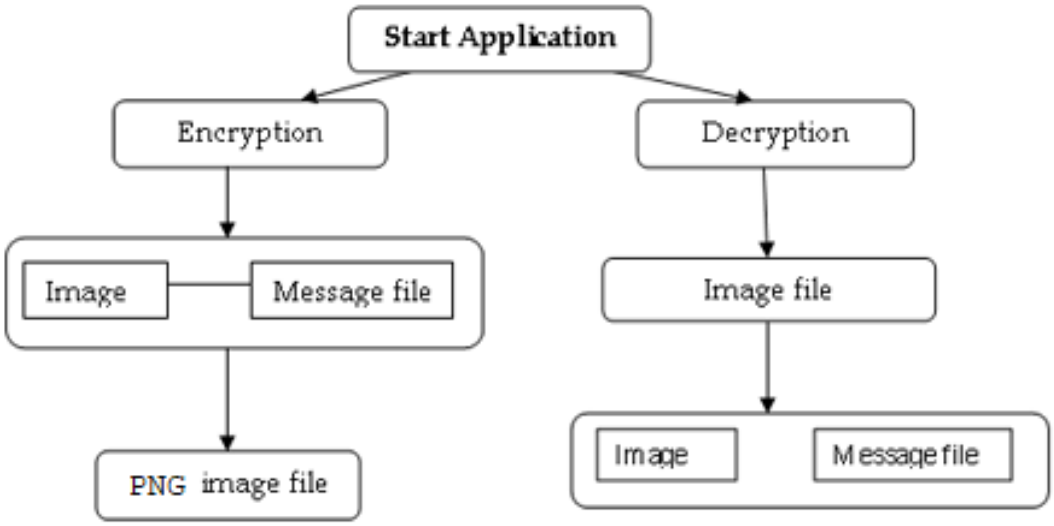

Figure. 4.6 Flowchart for Image-based Information Hiding Encryption Algorithm

Algorithm 4.3: Image-based Information Hiding Encryption Algorithm
Input: Data file to be encrypted, PNG Image on which data is to be hidden **Output:** PNG Image encoded with encrypted data
1: ASCII value of every character present in the information file is used. **2:** Value is transformed to a binary of 8 bits. **3:** Read three pixels from the image, a total of 9 RGB values, i.e. 3 X 3 values

are used.
4: A single character is stored as an 8-bit binary using eight RGB values.
5: Binary data and the associated RGB value are contrasted.
    If the binary digit is equal to 1
        RGB value is translated as odd
    Else if the binary digit is equal to 0
        RGB value is translated into even
6: Ninth value specifies whether or not further pixels are to be parsed.
    If additional information needs to be embedded
        Ninth value becomes even
    Else
        Ninth value becomes odd
7: Continue doing this until the image has encoded all of the data.

```
for i when in the range of data length:
p = [v for v in id.__next__()[:3] + id.__next__()[:3]
+ id.__next__()[:3]]
     for j when in the range of 0 to 8:
     if dl[i][j] is equal to '0' and p[j] is odd:
          p[j] = p[j] - 1

     else if dl[i][j] is equal to '1' and p[j] is
even:
          if p[j] is not equal to 0:
               p[j] = p[j] - 1
          else:
               p[j] = p[j] + 1
```

```
Access Granted!
Enter file to be encrypted: excel.csv
Enter image file: sample.png
Encryption time:
0.04626935999840498
```

Figure. 4.7 Time Taken for Image-based Information Hiding Encryption Algorithm

The user, after all verification validations hold true, needs to get the decrypted version of this encrypted image-based information hiding data as shown in Figure 4.8.

Algorithm 4.4: Image-based Information Hiding Decryption Algorithm

Input: PNG Image encoded with encrypted data
Output: Decrypted data file

1: Three pixels are read from the image, a total of 9 RGB values, i.e. 3 X 3 values are used.
2: The secret data is revealed by the initial eight RGB values, and the final value tells us whether we should move further or not.
3: For the first eight numbers
    If value is odd
        Binary bit is set to 1
    Else
        Binary bit is set to 0
4: If final value is even
    Continue parsing three pixels at once
  Else
    Decryption is done

```
while True:
    p = [v for v in id.__next__()[:3] +
    id.__next__()[:3] + id.__next__()[:3]]
    b = ''
    for i = p[:8]:
         if i is even:
              b = b + '0'
         else if i is odd:
              b = b + '1'
    d = d + c(int(b, 2))
    if p[-1] is odd:
        return d
```

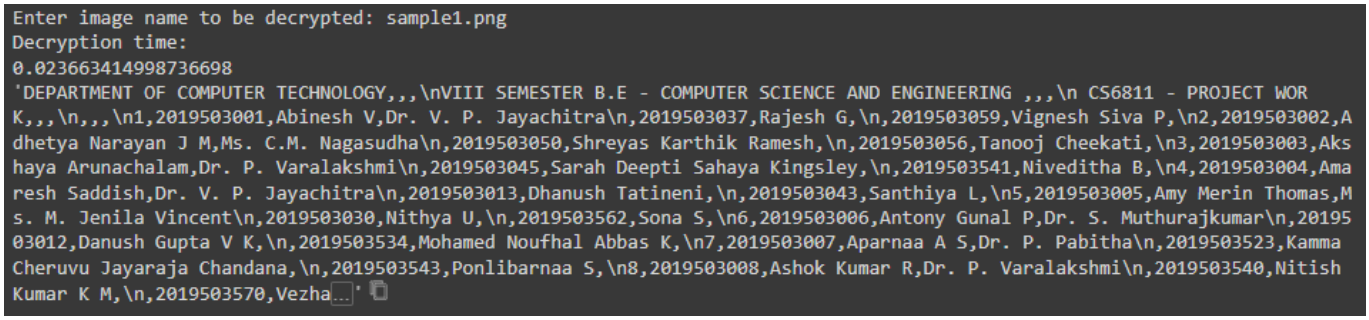

Figure. 4.8 Image-based Information Hiding Decryption Algorithm

CHAPTER 5
RESULTS

The implementation of the system has resulted in novel results that have significant implications for data security and management in the cloud.

Firstly, the system has demonstrated significant time savings compared to existing systems. This is due to the system's efficient performance and streamlined processes. The system has been designed to minimize computing time and storage resources, which has resulted in a faster, more responsive system that can handle large amounts of data.

Secondly, the system has significantly improved security aspects by providing a multilevel authentication scheme. This means that data is protected by multiple layers of security, reducing the ease with which attackers can get sensitive information. The authentication scheme is designed to confirm users' identities before granting access to the system, ensuring that the data is only accessible to authorised users.

Finally, the system protects the data by encrypting the stored data in the cloud server. This ensures that even if attackers gain access, they won't be able to view or utilise the info if they have access to it. The encryption process uses advanced cryptographic algorithms, making it virtually impossible for attackers to break the encryption.

To ensure that the implemented modules function as intended, various test cases and scenarios have been designed and executed. The test results have been averaged and documented, and the system has been found to perform well under

various conditions. This ensures that the system is reliable and can be trusted to handle sensitive data.

Table 5.1 shows the time taken for each level of authentication, i.e the username and password validation including the encryption and decryption time and the fingerprint verification.

Table 5.1 Time Taken for Authentication in Each Level

ASPECT	TIME TAKEN (SECONDS)
First Level (User Credential Validation)	0.16169
Second Level (Fingerprint Verification)	0.01132
Total (Authentication)	0.17301

The first level of authentication, i.e. the username and password verification encompassed with the encryption and decryption of the same from the database takes approximately 0.16169 seconds as shown in Figure 4.3.

Similarly, the time taken for the second level of authentication, i.e. the fingerprint verification is relatively less as compared with the other systems. This second level of authentication takes approximately 0.01132 seconds as shown in Figure 4.5.

In total, the complete time taken for the validation of the user is around 0.17301 seconds which is proven to be less than the other system models shown below in Figure 5.1.

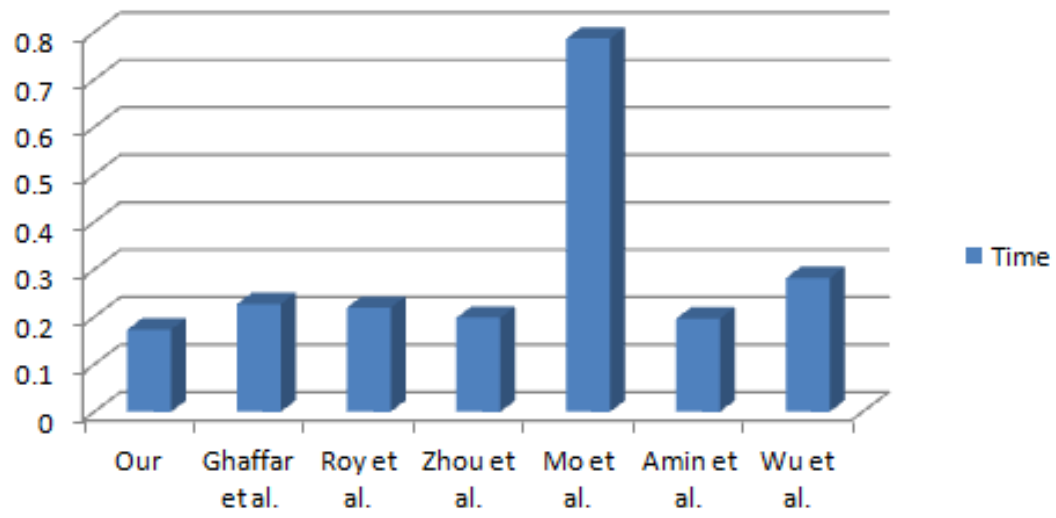

Figure. 5.1 Comparison of Computation Costs

Table 6.2 shows the period of time it takes to encrypt and decrypt data before it can be safely stored on cloud servers. On average, the time taken for encrypting the data file into the image takes approximately 0.04626 seconds as shown in Figure 4.7. Similarly, the period of time it takes for decrypting the image file to retrieve the information encoded takes approximately 0.02366 seconds. In total, the encryption and decryption process takes 0.06992 seconds on the whole.

Table 5.2 Time Taken for Data Encryption and Decryption

ASPECT	TIME TAKEN (SECONDS)
Data Encryption	0.04626
Data Decryption	0.02366
Total	0.06992

CHAPTER 6
CONCLUSION

The suggested system model, which offers secure credential encryption and satisfies most of the criteria for safe data exchange and storage in the cloud, has several benefits.

Firstly, it offers a significant degree of protection by encrypting credentials and ensuring that data is exchanged and stored safely. This is particularly important in cloud computing, where data is stored on servers that are accessed via the internet, making them vulnerable to security threats. By using secure encryption techniques and meeting the criteria for safe data exchange and storage, the suggested system model provides a strong defense against cyberattacks and unauthorized access.

Secondly, the system's performance study shows it to be more effective than comparable regimens. This means that it is faster and requires less storage resources, making it a more practical and cost-effective solution for organizations that need to exchange and store data securely in the cloud. By minimizing computing time and storage resources, the system can reduce operational costs and improve overall efficiency.

In conclusion, the suggested system model is an effective, reliable, and secure solution for exchanging and storing data in the cloud. Its secure credential encryption, compliance with safe data exchange and storage criteria, and efficient performance make it an ideal choice for businesses seeking to protect their data while enjoying the benefits of cloud computing. This means that businesses can enjoy the cloud computing advantages like scalability and flexibility, without compromising on security.

REFERENCES

[1] A. K. Junejo, N. Komninos and J. A. McCann, "A Secure Integrated Framework for Fog-Assisted Internet-of-Things Systems," in IEEE Internet of Things Journal, vol. 8, no. 8, pp. 6840-6852, doi: 10.1109/JIOT.2020.3035474, April 15, 2021.

[2] D. Chang, S. Garg, M. Hasan and S. Mishra, "Cancelable Multi-Biometric Approach Using Fuzzy Extractor and Novel Bit-Wise Encryption," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3152-3167, 2020, doi: 10.1109/TIFS.2020.2983250.

[3] D. I. G. Amalarethinam and J. S. Geetha, "Image encryption and decryption in public key cryptography based on MR," 2015 International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, 2015, pp. 133-138, doi: 10.1109/ICCCT2.2015.7292733.

[4] D. Tiwari, G. K. Chaturvedi, and G. Gangadharan, “ACDAS: Authenticated controlled data access and sharing scheme for cloud storage,” Int. J. Commun. Syst., vol. 32, no. 15, 2019.

[5] F. Wu and L. Xu, “A chaotic map-based authentication and key agreement scheme with user anonymity for cloud computing,” in Proc. Int. Conf. Cloud Comput. Secur., 2017, pp. 189–200.

[6] G.P. Biswas, R. Iqbal, Ruhul Amin, Neeraj Kumar, Victor Chang,"A lightweight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment", Future Generation Computer Systems, Volume 78, Part 3, 2018.

[7] H. Deng et al., "Identity-Based Encryption Transformation for Flexible Sharing of Encrypted Data in Public Cloud," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3168-3180, 2020, doi: 10.1109/TIFS.2020.2985532.

[8] H. Li, Q. Huang and W. Susilo, "A Secure Cloud Data Sharing Protocol for Enterprise Supporting Hierarchical Keyword Search," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 3, pp. 1532-1543, 1 May-June 2022, doi: 10.1109/TDSC.2020.3027611.

[9] H. Xiong, T. Yao, H. Wang, J. Feng and S. Yu, "A Survey of Public-Key Encryption With Search Functionality for Cloud-Assisted IoT," in IEEE Internet of Things Journal, vol. 9, no. 1, pp. 401-418, 1 Jan.1, 2022, doi: 10.1109/JIOT.2021.3109440.

[10] H. Zhong, C. Zhang, J. Cui, Y. Xu and L. Liu, "Authentication and Key Agreement Based on Anonymous Identity for Peer to Peer Cloud," in IEEE Transactions on Cloud Computing, vol. 10, no. 3, pp. 1592-1603, doi: 10.1109/TCC.2020. 1 July-Sept. 2022.

[11] J. Mo, Z. Hu, H. Chen, and W. Shen, "An efficient and provably secure anonymous user authentication and key agreement for mobile cloud computing," Wireless Commun. Mobile Comput., vol. 2019, 2019, Art. no. 4520685.

[12] J. Srinivas, A. K. Das, N. Kumar and J. J. P. C. Rodrigues, "Cloud Centric Authentication for Wearable Healthcare Monitoring System," in IEEE Transactions, vol. 17, no. 5, pp. 942-956, 1 Sept.-Oct. 2020, doi:10.1109/TDSC.2018.2828306.

[13] L. Zhou, X. Li, K.-H. Yeh, C. Su, and W. Chiu, “Lightweight IoT-based authentication scheme in cloud computing circumstance,” Future Gener. Comput. Syst., vol. 91, pp. 244–251, 2019.

[14] Liang Fang, Guozhen Shi, Lianhai Wang, Yongjun Li, Shujiang Xu, Yunchuan Guo, “Incentive mechanism for cooperative authentication: An evolutionary game approach”, Information volume 527,2020, pages 369-381.

[15] M.A. Akram, Z. Ghaffar, K.Mahmood, et al. “An anonymous authenticated key-agreement scheme for multi-server infrastructure”. Hum. Cent. Comput. Inf. Sci. 10, 22 (2020).

[16] N. H. Sultan, F. A. Barbhuiya, and M. Laurent, “ICAuth: A secure and scalable owner delegated inter-cloud authorization”, Future Gener. Comput. Syst., vol. 88, pp. 319–332, 2018.

[17] N. Munir et al., "Cryptanalysis of Internet of Health Things Encryption Scheme Based on Chaotic Maps," in IEEE Access, vol. 9, pp. 105678-105685, 2021, doi:10.1109/ACCESS.2021.3099004.

[18] P. -w. Harn, S. D. Yeddula, L. Sun, M. -T. Sun and W. -S. Ku, "Location-based Alert System Using Searchable Encryption with Hilbert Curve Encoding," 2022

IEEE International Conference on Big Data (Big Data), Osaka, Japan, 2022, pp. 1445-1454, doi: 10.1109/BigData55660.2022.10020428.

[19] R. Chaudhary, A. Jindal, G. S. Aujla, N. Kumar, A. K. Das and N. Saxena, "LSCSH: Lattice-Based Secure Cryptosystem for Smart Healthcare in Smart Cities Environment," in IEEECommunications Magazine, vol. 56, no. 4, pp. 24-32, April 2018, doi: 10.1109/MCOM.2018.1700787.

[20] S Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, “On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services,” IEEE Access, vol. 5, pp. 25808–25825, 2017.

[21] S. Sharaf and N. F. Shilbayeh, "A Secure G-Cloud-Based Framework for Government Healthcare Services," in IEEE Access, vol. 7, pp. 37876-37882, 2019, doi: 10.1109/ACCESS.2019.2906131.

[22] T. Shen, F. Wang, K. Chen, K. Wang and B. Li, "Efficient Leveled (Multi) Identity-Based Fully Homomorphic Encryption Schemes," in IEEE Access, vol. 7, pp. 79299-79310, 2019, doi:10.1109/ACCESS.2019.2922685.

[23] Y. Zhang, C. Xu, N. Cheng and X. Shen, "Secure Password-Protected Encryption Key for Deduplicated Cloud Storage Systems," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 4, pp. 2789-2806, 1 July-Aug. 2022, doi: 10.1109/TDSC.2021.3074146.

[24] Y. Zhou, Z. Hu and F. Li,"Searchable Public-Key Encryption With Cryptographic Reverse Firewalls for Cloud Storage," in IEEE Transactions on Cloud Computing, vol. 11, no. 1, pp. 383-396, 1 Jan.-March 2023, doi: 10.1109/TCC.2021.3095498.

[25] Yoon, Eun-Jun & Das, Ashok Kumar & Kee-Young, Yoo & Reddy, Alavalapati. ”Lightweight authentication with key agreement protocol for mobile network environment using smart cards”, IET Information Security, 2016.

[26] Z. Ghaffar, S. Shamshad, K. Mahmood, S. Kumari and M. K. Khan, "A Lightweight and Efficient Remote Data Authentication Protocol Over Cloud Storage Environment," IEEE Transactions on Network Science and Engineering, vol. 10, no. 1, pp. 103-112, 1 Jan.-Feb. 2023, doi: 10.1109/TNSE.2022