MIT-WPU

।। विश्वशान्तिर्ध्रुवं ध्रुवा ।।

Project Report

on

DoubleHelix Shield: DNA-AES Encryption for Cloud Storage with Cryptographic Splitting

Submitted by

Project Members

1032201067 - Shivam Rajawat
1032201077 - Vatsal Borad
1032201085 - Sumeet B. Pavitrakar
1032201015 - Chhavi Mandowara

Under the Internal Guidance of

Prof. Vitthal Gutte

School of Computer Engineering and Technology
MIT World Peace University, Kothrud,
Pune 411 038, Maharashtra - India
2023-2024

MIT-WPU
।। विश्वशान्तिर्ध्रुवं ध्रुवा ।।

Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

SCHOOL OF COMPUTER ENGINEERING AND TECHNOLOGY

CERTIFICATE

This is to certify that, **Shivam Rajawat, Vatsal Borad, Sumeet Pavitrakar, Chhavi Mandowara** of BTech.(Computer Science & Engineering) have completed their project titled “**DoubleHelix Shield: DNA-AES Encryption for Cloud Storage with Cryptographic Splitting** ” and have submitted this Capstone Project Report towards fulfillment of the requirement for the Degree-Bachelor of Computer Science & Engineering (BTech-CSE) for the academic year 2023-2024.

[Dr Vitthal Gutte]
Project Guide
School of CET
MIT World Peace University, Pune
Pune

[Dr. Vrushali Kulkarni]
Program Head
School of CET
MIT World Peace University,

Internal Examiner:

External Examiner:

Date:

Acknowledgement

Our sincere gratitude goes to our guide, Dr. Vitthal Gutte, whose guidance, expertise, and unwavering support were instrumental throughout the development of the "DoubleHelix Shield" framework. Their invaluable insights and encouragement significantly shaped the project's trajectory.

We also extend our thanks to MIT World Peace University for providing the necessary resources, infrastructure, and conducive environment essential for the successful completion of this project. The support and facilities offered played a pivotal role in facilitating our research and experimentation.

Our heartfelt appreciation to all individuals involved for their contributions, especially Dr. Vitthal Gutte, whose mentorship and dedication were pivotal in making this project a reality.

Shivam Rajawat

Vatsal Borad

Sumeet Pavitrakar

Chhavi Mandowara

Contents

Abstract

The "DoubleHelix Shield" project presents an innovative framework designed to fortify cloud data security through a novel amalgamation of DNA-based cryptography, Cryptographic Splitting, and Advanced Encryption Standard (AES). Leveraging the unique properties of DNA sequences, the framework translates user data into a secure biological format, further divided and encrypted for distributed storage in cloud environments. The project encompasses four primary modules: DNA Encryption, Key Management System, Cryptography Splitting and AES Encryption, and Server and RSA Encryption. These modules collaboratively ensure robust encryption, secure key distribution, data segmentation, and distributed cloud storage with stringent access controls. The project's aim is to elevate data security standards within cloud infrastructures, offering a comprehensive and resilient solution to safeguard sensitive information.

List of Figures

List of Tables

Chapter 1

Introduction

This research introduces "DoubleHelix Shield," a pioneering data security framework that merges DNA cryptography, Cryptographic Splitting, and AES encryption to fortify cloud-based data protection. In response to mounting security challenges in cloud storage, this framework uniquely combines these encryption paradigms to ensure confidentiality and integrity. DNA cryptography harnesses biological structures to encode data, while Cryptographic Splitting fragments and disperses data across servers. The integration of AES adds further resilience. Four core modules constitute this framework: DNA encryption, a Key Management System, middleware for encryption and dispersion, and a server-side encryption module. This innovative system not only addresses contemporary security challenges but also elevates data security standards in cloud computing, promising comprehensive protection for sensitive information.

1.1 Problem Statement

Enhancing security in cloud environments demands innovative approaches. The integration of DNA cryptography, Cryptographic Splitting, and AES encryption data protection in distributed cloud systems.

1.2 Area

Cloud Data Security

1.3 Aim

This project aims to showcase the system's robustness in safeguarding sensitive data within distributed cloud systems, validating its effectiveness in addressing contemporary cybersecurity challenges.

Chapter 2
Literature Survey

Sr No	Publication Title with Authors (mention journal or conference)	Year of Publication	Positive Points of the Publication	Gaps in publication work
1.	A Novel Hybrid Secure Method Based on DNA Encoding Encryption and Spiral Scrambling in Chaotic OFDM-PON **Authors:** Yaoqiang Xiao, Yating Chen, Caixia Long, Jin Shi, Jie Ma, Jing He	2020	Addresses the security concern of data transmission in OFDM-PON systems, the paper proposes a novel hybrid secure method that combines DNA encoding encryption and spiral scrambling techniques.	As quantum communication technologies advance, integrating quantum key distribution (QKD) and quantum-resistant encryption techniques into OFDM-PON systems could be a promising avenue for research to ensure long-term security.
2.	Lattice-Based Key-Aggregate (Searchable) Encryption in Cloud Storage **Authors:** Yanqing Yao, Zhengde Zhai , Jainwei Liu and Zhoujun Li.I	2019	In addressing security challenges posed by quantum computing attacks in key encryption, the paper presents lattice-based key-aggregate encryption schemes. These schemes incorporate a basis delegation algorithm, ensuring aggregate key generation without expanding lattice dimension.	The current encryption algorithm operates on a single bit at a time, constraining its utility for larger data units. Moreover, unexplored complexities arise in multi-owner scenarios, presenting a gap in the existing literature.

			Additionally, the paper introduces encryption algorithms designed to enable the decryption and searchability of files.	
3.	One Time User Key: A User-Based Secret Sharing XOR-ed Model for Multiple User Cryptography in Distributed Systems **Authors:** Stefano Galantucci, Donato Impedovo and Giuseppe Pirlo.	2021	Addressing the complexity of establishing encrypted channels among more than two users, the paper presents One Time User Keys(OUTKs) to divide the secret, constituting the actual encryption key.	The system's reliance on a key management dealer introduces a security vulnerability, particularly in cases where the dealer might act maliciously. Additionally, vulnerabilities in the synchronization protocol and potential weaknesses in the storage for decryption pose notable gaps in the existing survey.
4.	Secure Outsourcing and Sharing of Cloud Data Using a User-Side Encrypted File System **Author:** Osama Ahmed Khashan	2020	OutFS is a user-side encrypted file system that is implemented based on FUSE to secure outsourced files to cloud storage systems.	Enhance the performance of OutFS using several improvement suggestions, such as parallel encryption, selective encryption, and intelligent cryptography.

Table 1. Literature Survey

Chapter 3

Problem Statement

In the dynamic landscape of cloud computing, ensuring the secure storage and transmission of sensitive data remains a pressing concern. To confront these challenges, our project presents a holistic data security solution that integrates DNA cryptography with Cryptographic Splitting using AES encryption. This hybrid approach promises robust data protection but confronts several compelling challenges that necessitate investigation. While existing literature offers valuable insights, notable gaps emerge in the research that our project seeks to bridge.

3.1 Project Scope

- Implementing the "DoubleHelix Shield" data security framework integrating DNA cryptography, Cryptographic Splitting, and AES encryption.
- Validating the functionality and interoperability of each module within the distributed cloud system.
- Assessing the framework's efficiency in securing data, ensuring confidentiality, integrity, and accessibility.
- Evaluating scalability and performance against various security threats and data types.
- Exploring real-world applications and adaptability to diverse cloud architectures.
- Conducting comprehensive testing to verify the framework's robustness in addressing contemporary cybersecurity challenges.

3.2 Project Limitations

- Scalability challenges may arise due to increased data volume and user demand.
- Implementing multiple encryption layers might introduce performance overhead.
- Dependency on the reliability and security of underlying cloud infrastructure.
- Complexity in managing encryption keys, especially in a distributed environment.
- Compatibility issues or integration challenges with existing cloud systems.

3.3 Project Objectives

- To implement a secure data encryption mechanism using DNA cryptography and AES for cloud data storage.
- To develop a distributed cloud storage system capable of storing and retrieving encrypted data efficiently.
- To evaluate the system's performance, security, and scalability through extensive testing and analysis.
- To document the entire project, including design, implementation, and testing processes for future reference.

Chapter 4

Project Requirements

4.1 Resources

- **Human Resources**
  - Security Experts: Required for the design, implementation, and testing phases to ensure robustness against potential vulnerabilities.
  - Software Engineers: Responsible for coding, integration, and testing of the "DoubleHelix Shield" framework.
  - System Administrators: Needed for configuring and maintaining the cloud infrastructure.

- **Reusable Software Components**
  - Data Preprocessing Tools: Algorithms or scripts used for standardizing and preparing data before encryption.
  - Encryption Libraries: Reusable code libraries for implementing DNA cryptography, Cryptographic Splitting, and AES encryption.

- **Software & h/w requirements**
  - Operating Systems: Compatible OS for client-side and server-side deployment.
  - Programming Languages: Languages suitable for encryption implementations (e.g., Python, Java).
  - Servers: High-capacity servers with secure storage and processing capabilities.
  - Networking Equipment: Secure networking hardware for encrypted data transmission.
  - Cloud Infrastructure: Access to distributed cloud systems for testing and deployment.

4.2 Risk Management

- Identify encryption vulnerabilities, compatibility issues, and performance overhead as potential risks.
- Assess likelihood and impact of vulnerabilities on data security and system performance.
- Develop testing and code review protocols, compatibility checks, and performance optimization measures as risk mitigation strategies.
- Continuously monitor encryption protocols, adapt to changes in cloud infrastructures, and adjust optimization strategies based on system monitoring.
- Prepare backup encryption protocols and emergency response plans for critical system failures or data breaches as contingency measures.

4.3 Functional Specifications:

4.3.1 Interfaces

- **External Interfaces Required**
  - Encryption APIs: Expose methods for initiating encryption, decryption, and key management to external systems or applications.
  - Data Retrieval Service: Interface for retrieving encrypted data blocks from distributed cloud servers.
  - Key Distribution Service: Interface for distributing encryption keys securely to authorized entities.
- **Internal Interfaces Required**
  - Encryption Module Interface: Functions/methods allowing intercommunication between DNA cryptography, Cryptographic Splitting, and AES encryption modules.
  - Key Management Interface: Internal methods managing the generation, storage, and distribution of encryption keys.
- **Communication Interfaces**
  - Secure Networking Protocols: Interfaces for secure data transmission between client-side, middleware, and server-side components.
  - API Endpoints: Defined endpoints for secure communication between different layers of the "DoubleHelix Shield" framework.
- **Graphical User Interfaces (GUI)**
  - Administrative Dashboard: GUI for system administrators to monitor encryption processes, key management, and system health.
  - User Interface (Client-Side): Interface for users to interact with encryption functionalities or initiate data encryption.

4.3.2 Interactions

- **Sustainability Goals**
  - Resource Efficiency: Minimize resource consumption in encryption processes to reduce environmental impact.
  - Longevity: Ensure the framework's adaptability to evolving encryption standards, reducing the need for frequent updates or overhauls.
  - Scalability: Design for scalability to accommodate growing data volumes without significant system restructuring.
- **Quality Management**
  - Reliability: Ensure the consistent and accurate performance of encryption and decryption processes.
  - Testing Protocols: Implement rigorous testing methodologies to verify the robustness and functionality of encryption modules.
  - Continuous Improvement: Establish feedback mechanisms for ongoing enhancements based on user experience and evolving security standards.
- **Security**Confidentiality: Guarantee the protection of sensitive data during transmission, storage, and retrieval.
  - Integrity: Ensure data remains unaltered and consistent throughout encryption and decryption processes.

Chapter 5

high level design

5.1 Design Consideration

- Layered Encryption: DNA cryptography, Cryptographic Splitting, and AES encryption.
- Attribute-based key generation and secure distribution.
- Adaptability to diverse cloud setups and varying data volumes.
- Standardized interfaces for seamless communication and integration.
- Mechanisms for error detection and data recovery.
- Intuitive interfaces and admin dashboard for management

5.2 Assumption and Dependencies

- **Assumptions:**
    - Algorithmic Reliability: Assumes the reliability and effectiveness of DNA cryptography, Cryptographic Splitting, and AES encryption methodologies.
    - Stable Network Infrastructure: Assumes a stable and secure network environment for data transmission and system operation.
    - Compliance with Regulations: Assumes adherence to data protection regulations and industry standards throughout the framework's deployment.
    - Availability of Resources: Assumes availability and accessibility of necessary resources like servers, networking equipment, and skilled personnel.

- **Dependencies:**
    - Hardware and Software Availability: Dependent on the availability and functionality of required hardware components and compatible software tools.
    - Data Compatibility: Dependencies on standardized data formats and protocols for seamless integration with various cloud systems.
    - Security Updates: Dependencies on timely security updates and patches to address vulnerabilities in encryption methodologies.

5.3 General Constraints

Potential performance overhead due to multiple encryption layers, resource-intensive encryption processes impacting system performance, integration complexities with diverse cloud infrastructures, challenges in complying with varied data protection regulations, and the need for scalable solutions to handle increasing data volumes or user demands

5.4 Use Case Diagram

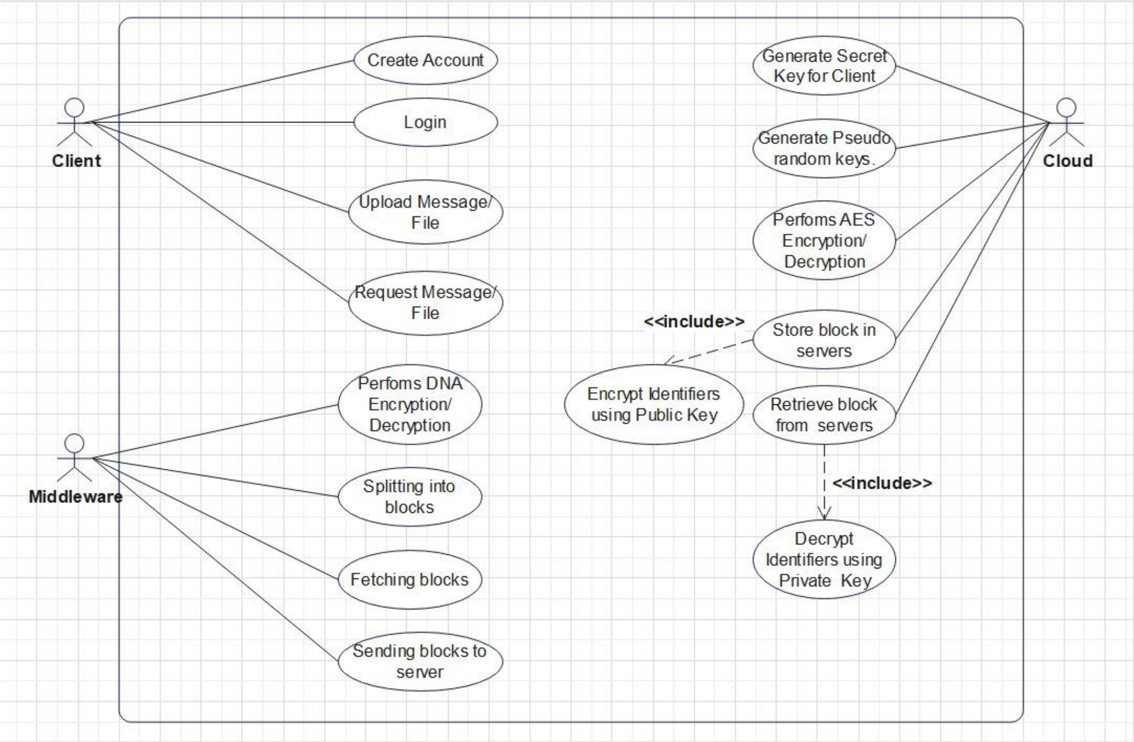

Client
Create Account
Login
Upload Message/ File
Request Message/ File
Middleware
Perfoms DNA Encryption/ Decryption
Splitting into blocks
Fetching blocks
Sending blocks to server
Generate Secret Key for Client
Generate Pseudo random keys.
Perfoms AES Encryption/ Decryption
<<include>>
Store block in servers
Encrypt Identifiers using Public Key
Retrieve block from servers
<<include>>
Decrypt Identifiers using Private Key
Cloud

Fig 1. Use Case Diagram

5.5 System Architecture

The system architecture for the "DoubleHelix Shield" framework is designed as a multi-layered and distributed model. It comprises four primary modules:

- ○ DNA Encryption: This module converts data into DNA sequences using robust encoding, employing DNA cryptography for secure transformation.
- ○ Key Management System: Responsible for generating and distributing encryption keys securely among the modules for data security.
- ○ Cryptography Splitting and AES Encryption: Divides data into sections, encrypts them using AES, and adds metadata for distributed storage across multiple servers in the cloud.
- ○ Server and RSA Encryption: Manages distributed cloud storage, utilizing RSA encryption to safeguard block addresses attached to encrypted data blocks.

The architecture ensures modularity, enabling each module to function independently while facilitating seamless communication between them. It emphasizes security, scalability, and adaptability to different cloud environments, providing robust data protection mechanisms within distributed systems.

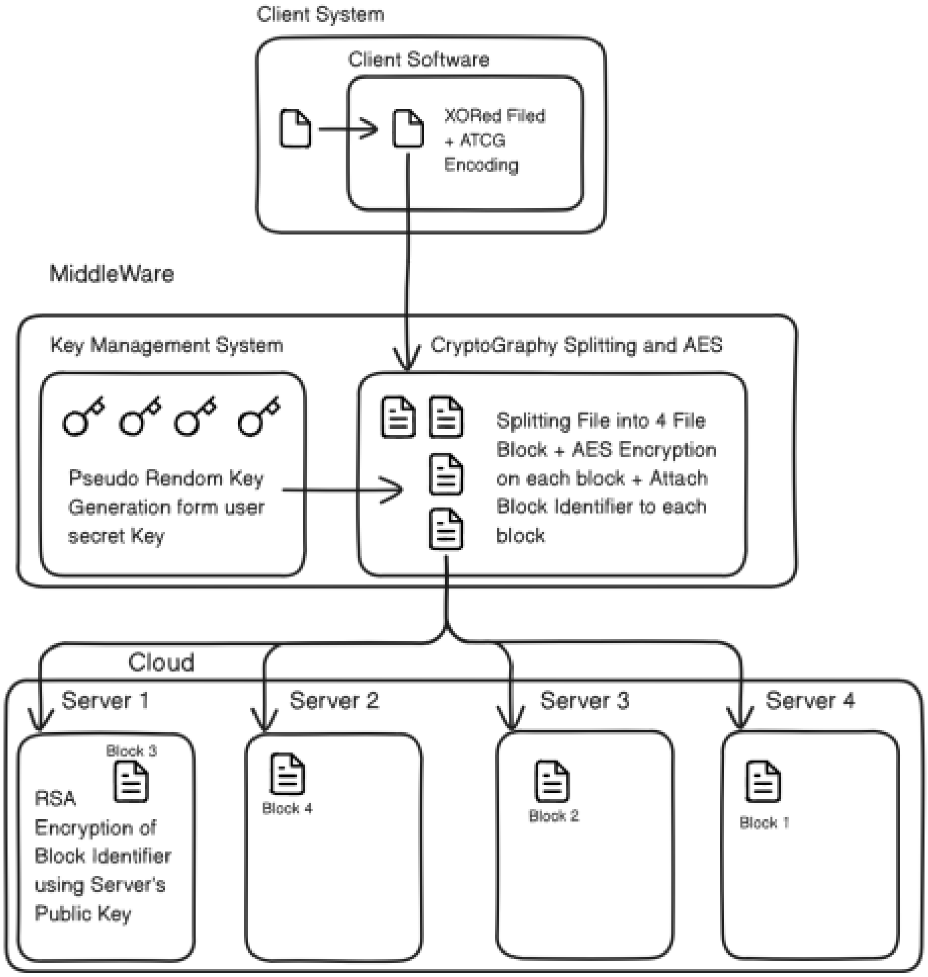


Fig 2. System Architecture

5.6 Modules of the Project

- DNA Encryption Module: Responsible for translating user data into DNA sequences using encoding schemes for secure transformation and storage.
- Key Management System: Generates and manages encryption keys based on user-defined attributes, ensuring secure distribution among modules.
- Cryptography Splitting and AES Encryption: Divides data into sections for distributed storage, applies AES encryption to each section, and adds metadata for retrieval.
- Server and RSA Encryption Module: Manages distributed cloud storage, utilizing RSA encryption for safeguarding block addresses attached to encrypted data blocks.

5.7 Low level Design

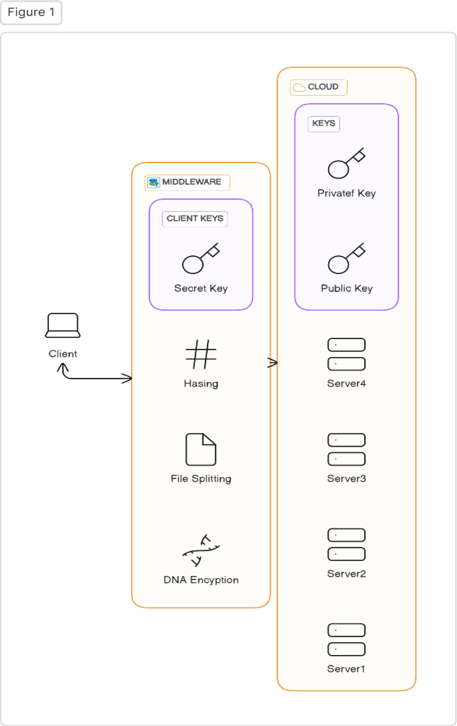


Fig 3. Low Level Design

5.8 UML Diagrams

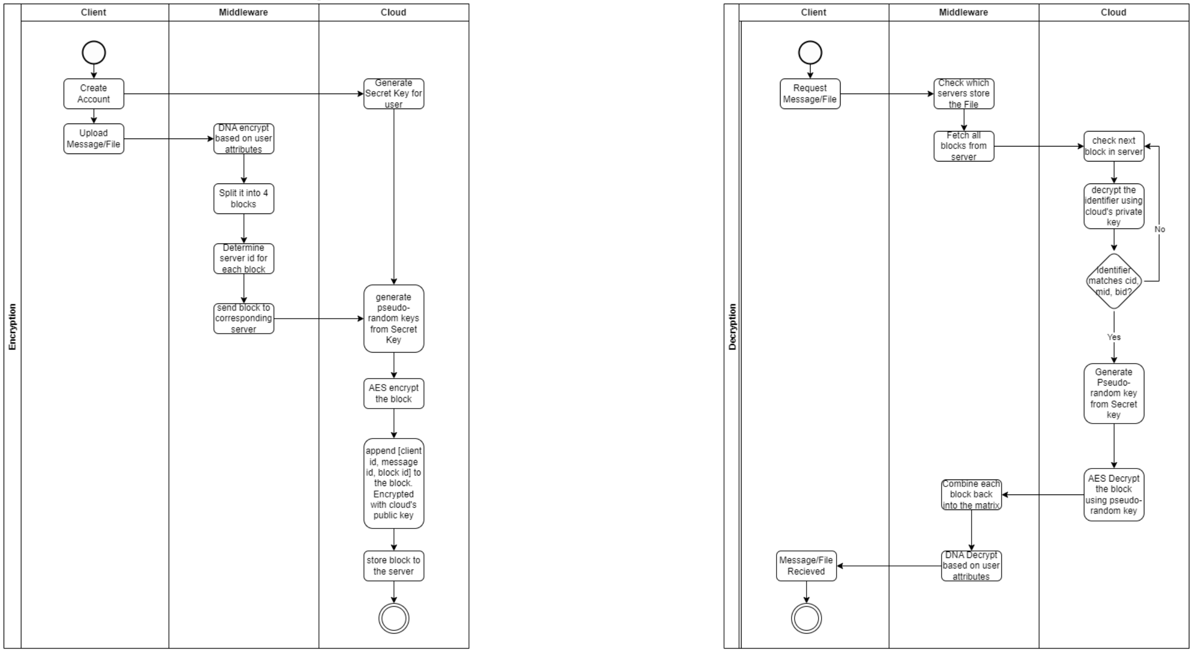

Encryption
Client
Middleware
Cloud
Create Account
Upload Message/File
Generate Secret Key for user
DNA encrypt based on user attributes
Split it into 4 blocks
Determine server id for each block
send block to corresponding server
generate pseudo-random keys from Secret Key
AES encrypt the block
append [client id, message id, block id] to the block. Encrypted with cloud's public key
store block to the server
Decryption
Client
Middleware
Cloud
Request Message/File
Check which servers store the File
Fetch all blocks from server
check next block in server
decrypt the identifier using cloud's private key
Identifier matches cid, mid, bid?
No
Yes
Generate Pseudo-random key from Secret key
AES Decrypt the block using pseudo-random key
Combine each block back into the matrix
DNA Decrypt based on user attributes
Message/File Recieved

Fig 4. Activity Diagram

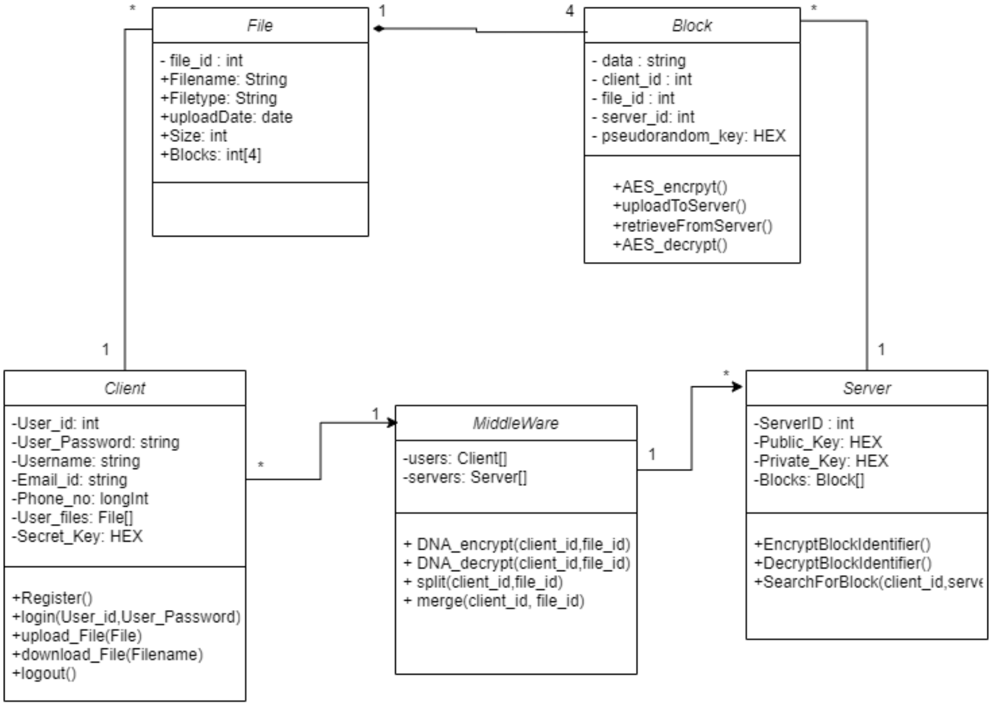


Fig 5. Class Diagram

Chapter 6

Project Plan

- Define project scope and objectives clearly.
- Identify and sequence project tasks logically.
- Allocate human and material resources efficiently.
- Create a timeline with milestones for progress tracking.
- Assess and mitigate project risks effectively.
- Analyze project budget and allocate resources accordingly.
- Establish communication channels and reporting structures.
- Develop a comprehensive testing and quality assurance plan.
- Document project components and devise training materials.
- Schedule regular reviews for adaptation and improvement

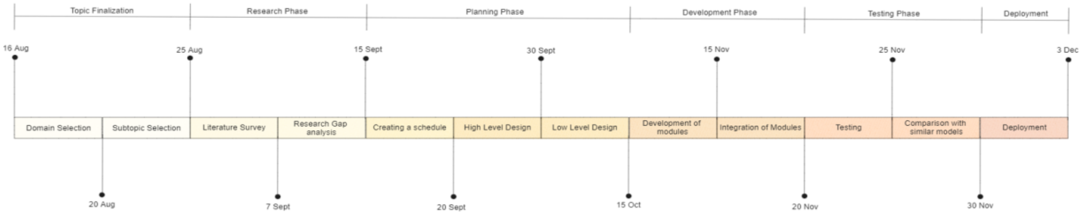


Fig 6. Timeline Diagram

Chapter 7
Implementation

The development process of the "DoubleHelix Shield" framework involved a methodical series of steps to ensure robust data security in cloud environments. Firstly, extensive research was conducted to understand the nuances of data protection and identify existing challenges. This was followed by a thorough analysis to determine the precise requirements our system needed to meet to ensure data security within cloud platforms.

7.1 Methodology

- **Research and Requirement Analysis**
    - This initial phase involved extensive research into existing encryption methodologies, prevalent data security vulnerabilities in cloud environments, and the evolving landscape of technology. We delved into understanding the intricacies of data protection, studying industry-specific needs, and identifying regulatory compliance standards.

    - This phase was instrumental in comprehending the challenges faced in securing data within cloud infrastructures. The research insights were further utilized to precisely define the requirements and objectives of the "DoubleHelix Shield" framework.

- **Framework Design**
    - The "DoubleHelix Shield" architecture operates through a cohesive integration of distinct modules. The system initiates by encoding user data into DNA sequences using sophisticated encoding methods. These DNA sequences represent the digital information in a unique biological format, leveraging the inherent complexity and variability of DNA for secure storage.

    - The encoded data then undergoes a process known as Cryptographic Splitting, where it gets divided into smaller sections or blocks. Each block undergoes Advanced Encryption Standard (AES) encryption using unique keys generated by the Key Management System. This encryption ensures that the data within each block remains secure and inaccessible without the corresponding decryption keys.

    - These encrypted blocks, along with their block addresses containing ownership details and essential metadata, are distributed across various servers within the cloud. Each server employs RSA encryption to further secure the block addresses, ensuring that unauthorized access to the distributed data is prevented.

  - During data retrieval, the server uses its private key to decrypt the block addresses and locate the specific encrypted block requested. This block is then retrieved and sent back to the system for decryption using the corresponding AES decryption keys obtained from the Key Management System. Finally, the decrypted blocks are reassembled to reconstruct the original user data.

  - This architecture's strength lies in its multi-layered encryption approach, leveraging DNA-based encoding, robust encryption methods, and distributed storage across multiple servers to ensure comprehensive data security within cloud environments.

- **Validation and Testing**

  - In the Validation and Testing phase, rigorous assessments were conducted to ensure the reliability and functionality of the "DoubleHelix Shield" framework. Various tests, simulations, and trials were performed to verify the system's performance, security, and adherence to established standards. This phase aimed to identify and rectify any weaknesses or vulnerabilities within the system before deployment, ensuring a robust and effective solution for securing data in cloud environments.

7.2 Algorithm

- **DNA-based Encryption Algorithm**
  - The DNA-based encryption algorithm employed in the "DoubleHelix Shield" system translates user data into sequences of DNA bases (A, T, C, G). It utilizes an encoding scheme designed to convert binary data into DNA sequences and vice versa, ensuring reversible conversion.
  - This process involves mapping specific DNA sequences to represent distinct binary elements, ensuring accuracy and error-checking mechanisms to maintain fidelity during encoding and decoding operations.
  - This algorithm also involves the XORing of the binary bits data with the user encrypted password.
  - The algorithm's robust mapping process efficiently translates binary data into corresponding DNA sequences, ensuring accurate representation and data integrity while leveraging the unique properties of DNA for secure storage and encoding.

- **Cryptography Splitting Algorithm**
  - The Cryptography Splitting algorithm divides the encoded data into independent file blocks for secure distribution. This process involves segmenting the DNA-encoded data into four distinct sections or blocks.
  - Each block undergoes encryption using the Advanced Encryption Standard (AES) with unique keys generated by the Key Management System. Post-encryption, the system generates encrypted file blocks, each embedded with a block address containing ownership details, file numbers, and block numbers essential for retrieval.
  - This algorithmic process ensures data segmentation, encryption, and attachment of metadata for secure storage within a distributed cloud system.

- **Server-side RSA Encryption Algorithm**
  - The Server-side RSA Encryption algorithm manages the distributed cloud system's storage, retrieval, and access mechanisms.
  - It employs RSA encryption to secure the block addresses attached to the encrypted file blocks. During retrieval requests, servers utilize their private keys to decrypt the block addresses and locate the specific file blocks requested.
  - This decryption process ensures secure data retrieval while maintaining the integrity and confidentiality of the distributed data.

7.3 Technologies and Libraries Used

- **Technologies:**
  - **Java:** Java served as the primary programming language for developing the "DoubleHelix Shield" framework. Its versatility and robustness facilitated the implementation of various encryption algorithms and system functionalities.

  - **Servlet:** Servlet technology was employed for handling requests and responses within the web application, enabling seamless interaction between the user and the system.

  - **JSP (JavaServer Pages):** JSP provided a dynamic web page creation environment, allowing for the integration of Java code within HTML pages, enhancing the system's user interface.

    - **MySQL:** MySQL, a widely-used relational database management system, was utilized for data storage, retrieval, and management, ensuring efficient and organized data handling within the framework.

- **Libraries:**
    - **JavaX.crypt for AES:** The JavaX.crypt library was utilized for Advanced Encryption Standard (AES) encryption within the framework. This library provided essential functionalities for securing data blocks during the encryption process.

    - **java.security for RSA:** The java.security library played a pivotal role in implementing the RSA encryption algorithm within the framework. It offered essential tools and functionalities for managing keys, encryption, and decryption processes, ensuring robust security measures.

Chapter 8
Performance Evaluation and Testing

- We conducted the test on the files of different sizes and calculated the encryption and decryption time. we performed the step by step demonstration of the project

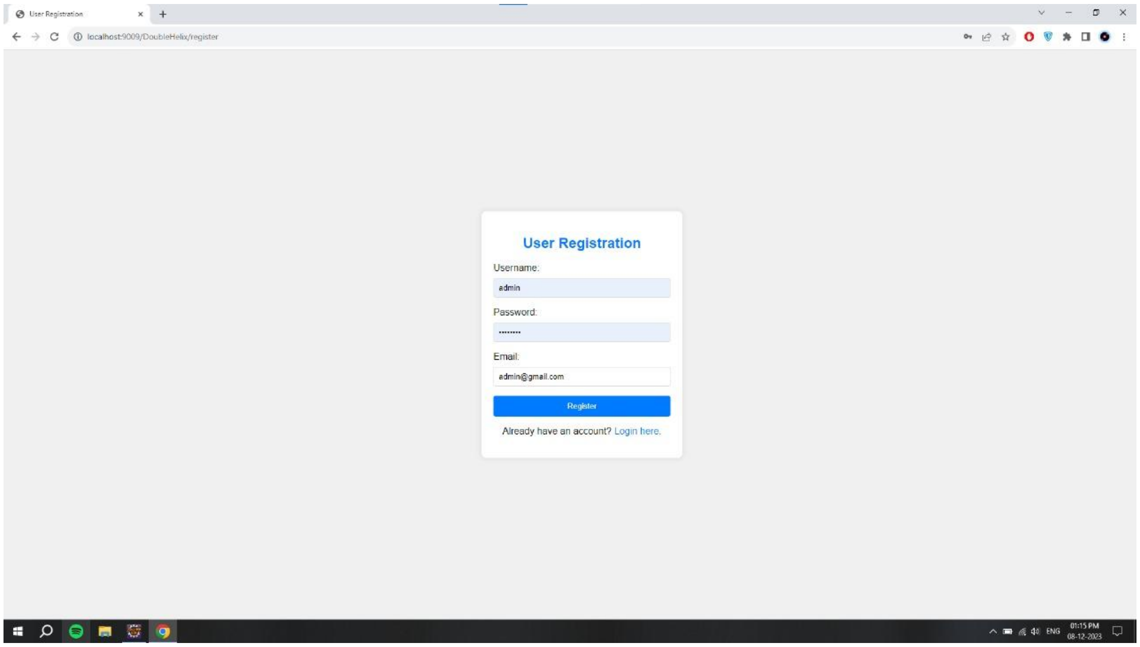
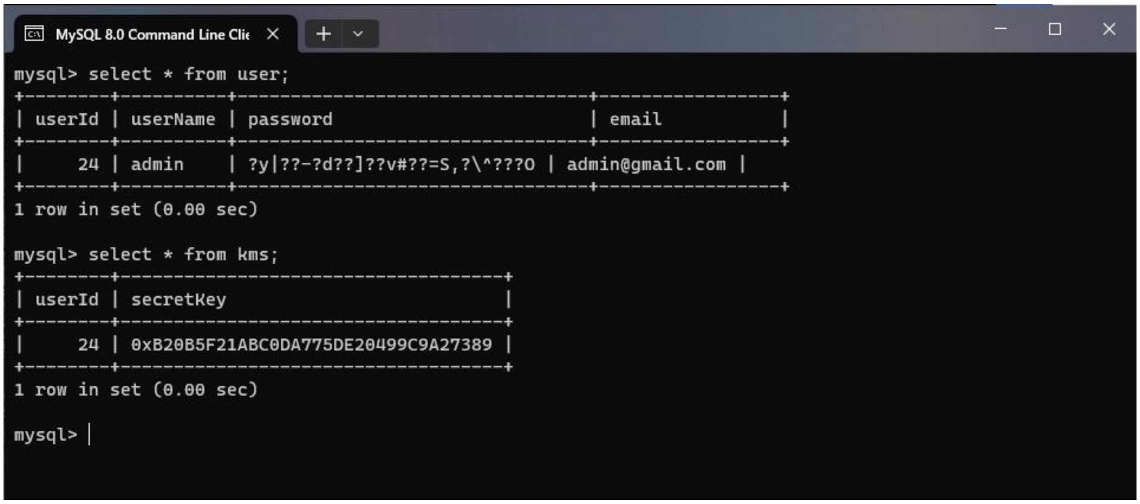


Fig 7. User Registration Page

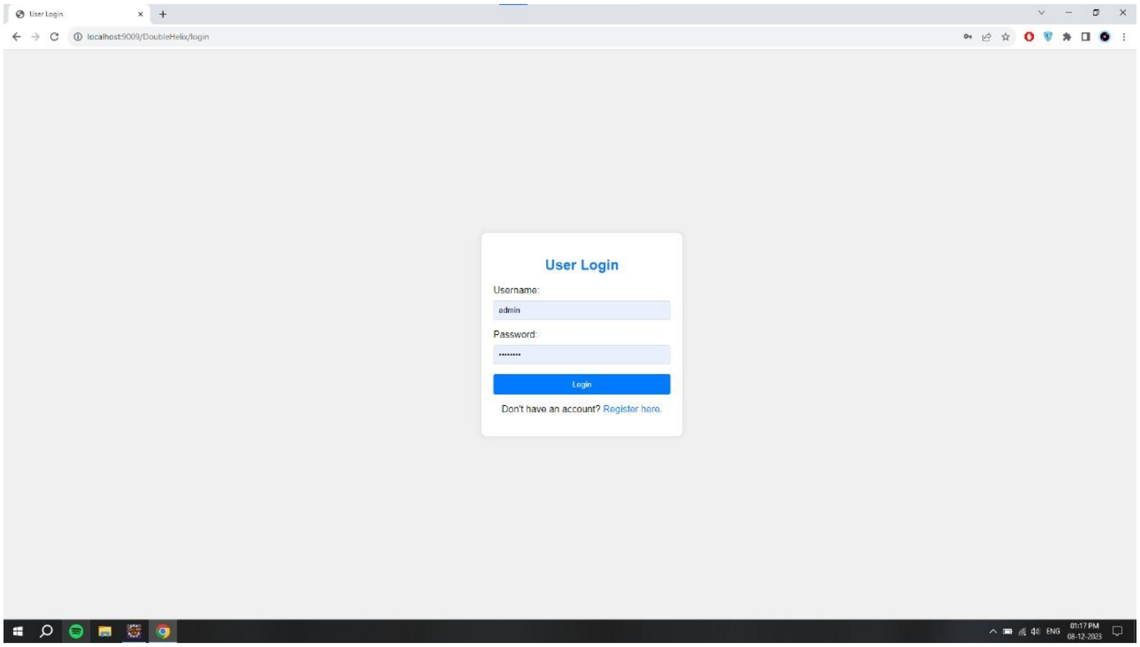


```
mysql> select * from user;
+--------+----------+---------------------------------+-----------------+
| userId | userName | password                        | email           |
+--------+----------+---------------------------------+-----------------+
|     24 | admin    | ?y|??-?d??]??v#??=S,?\^???O | admin@gmail.com |
+--------+----------+---------------------------------+-----------------+
1 row in set (0.00 sec)

mysql> select * from kms;
+--------+------------------------------------+
| userId | secretKey                          |
+--------+------------------------------------+
|     24 | 0xB20B5F21ABC0DA775DE20499C9A27389 |
+--------+------------------------------------+
1 row in set (0.00 sec)

mysql> |
```

Fig 8. User Database



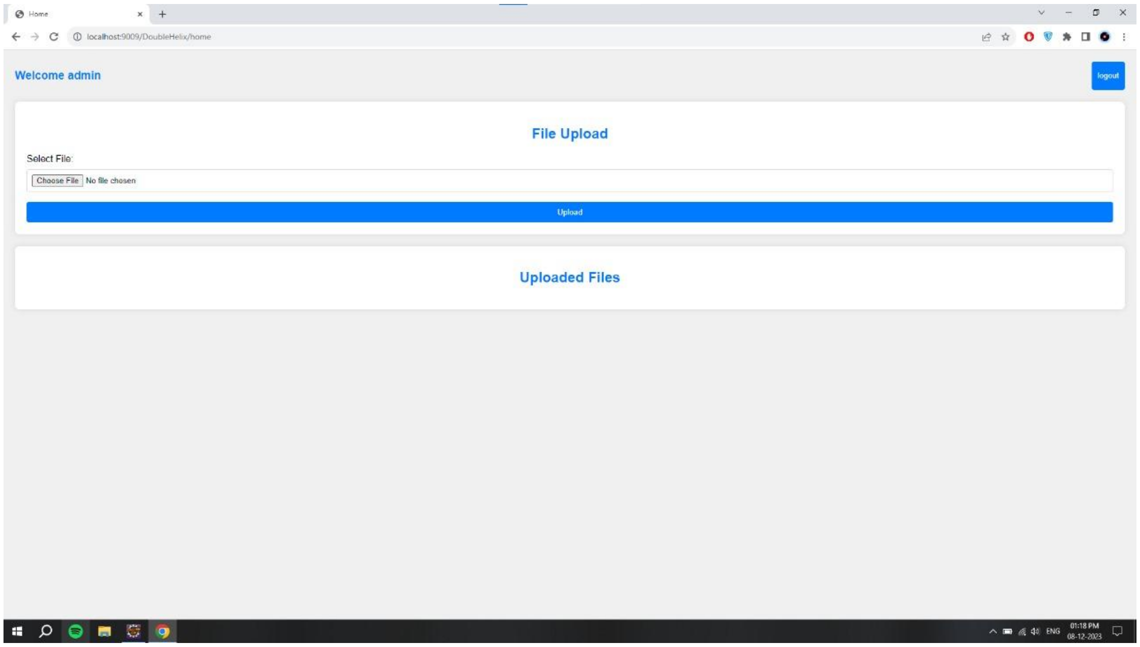
User Login

Username: admin

Password: ••••••••

Don't have an account? Register here.

Fig 9. User Login Page



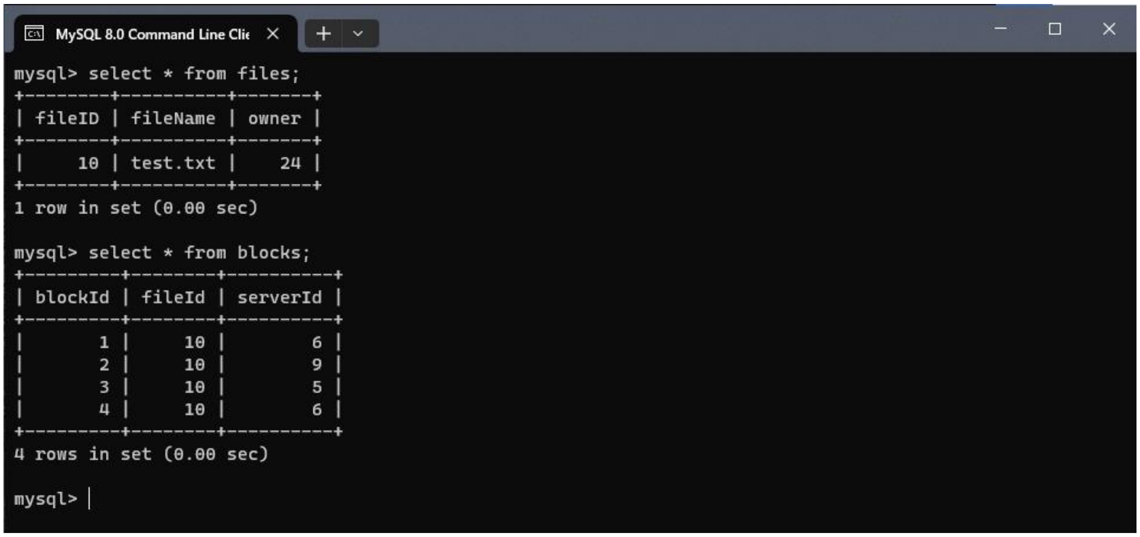
Welcome admin

File Upload

Select File: No file chosen

Uploaded Files

Fig 10. Home Page

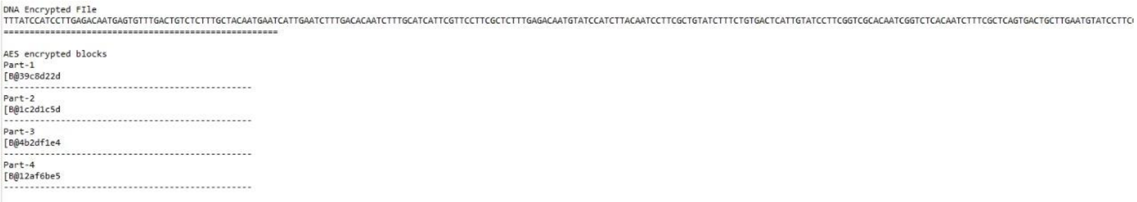


```
mysql> select * from files;
+--------+----------+-------+
| fileID | fileName | owner |
+--------+----------+-------+
|     10 | test.txt |    24 |
+--------+----------+-------+
1 row in set (0.00 sec)

mysql> select * from blocks;
+---------+--------+----------+
| blockId | fileId | serverId |
+---------+--------+----------+
|       1 |     10 |        6 |
|       2 |     10 |        9 |
|       3 |     10 |        5 |
|       4 |     10 |        6 |
+---------+--------+----------+
4 rows in set (0.00 sec)

mysql> |
```

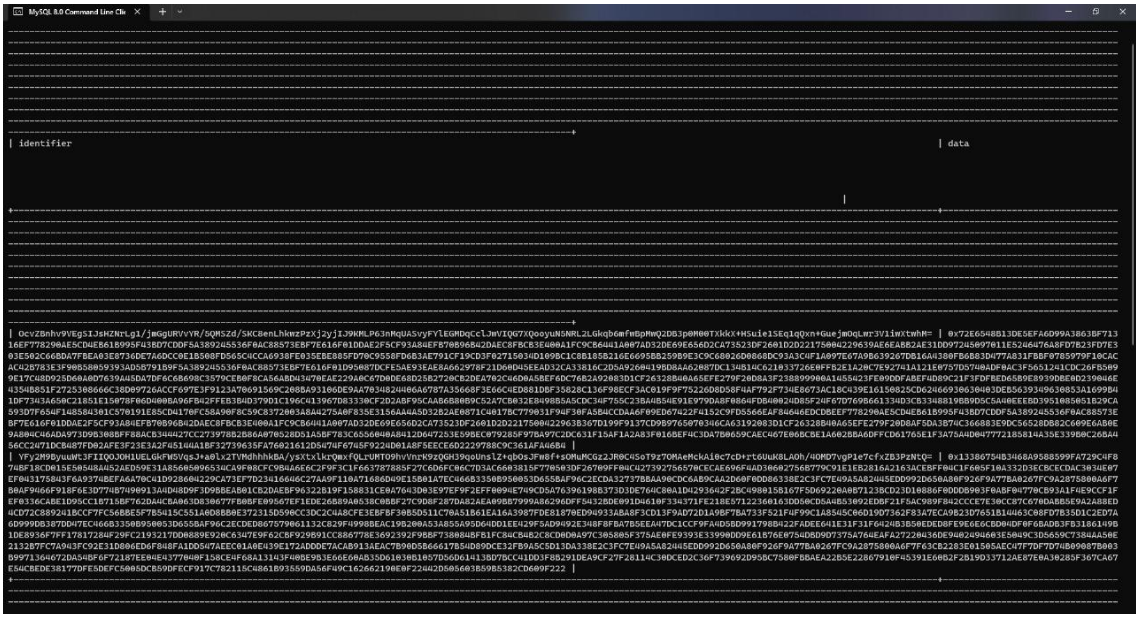
Fig 11. File Database of user



```
DNA Encrypted FIle
TTTATCCATCCTTGAGACAATGAGTGTTTGACTGTCTCTTTGCTACAATGAATCATTGAATCTTTGACACAATCTTTGCATCATTCGTTCCTTCGCTCTTTGAGACAATGTATCCATCTTACAATCCTTCGCTGTATCTTTCTGTGACTCATTGTATCCTTCGGTCGCACAATCGGTCTCACAATCTTTCGCTCAGTGACTGCTTGAATGTATCCTTC
=====================================================

AES encrypted blocks
Part-1
[B@39c8d22d
------------------------------------------------
Part-2
[B@1c2d1c5d
------------------------------------------------
Part-3
[B@4b2df1e4
------------------------------------------------
Part-4
[B@12af6be5
------------------------------------------------
```

Fig 12. DNA Encrypted and Splitted Files



```
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------+
| identifier                                                                                                  | data

                                                                                              |
+-----------------------------------------------------------------------------------------------------------+------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------+
| OcvZBnhv9VEgSIJsHZNrLg1/jmGgURVvYR/5QMSZd/SKC8enLhkwzPzXj2yjIJ9KMLP63nMqUASvyFYlEGMDqCclJwVIQG7XQooyuN5NRL2LGkqb6mfwBpMwQ2DB3p0M00TXkkX+HSuie1SEq1qQxn+GuejmOqLwr3V1iwXtwhM= | 0x72E6548B13DE5EFA6D99A3863BF713
16EF778290AE5CD4EB61B995F43BD7CDDF5A389245536F0AC88573EBF7E616F01DDAE2F5CF93A84EFB70B96B42DAEC8FBCB3E400A1FC9CB6441A007AD32DE69E656D2CA73523DF2601D2D22175004229639AE6EABB2AE31DD97245097011E5246476A8FD7B23FD7E3
03E502C66BDA7FBEA03E8736DE7A6DCC0E1B508FD565C4CCA6938FE035EBE885FD70C9558FD6B3AE791CF19CD3F02715034D109BC1C8B185B216E6695BB259B9E3C9C68026D0868DC93A3C4F1A097E67A9B639267DB16A4380FB6B83D477A831FBBF0785979F10CAC
AC42B783E3F90B58059393AD5B791B9F5A389245536F0AC88573EBF7E616F01D95087DCFE5AE93EAE8A662978F21D60D45EEAD32CA33816C2D5A9260419BD8AA62087DC134B14C621033726E0FFB2E1A20C7E92741A121E0757D5740ADF0AC3F5651241CDC26FB509
9E17C48D925D60A0D7639A45DA7DF6C6B698C3579CEB0F8CA56ABD43470EAE229A0C67D0DE68D25B2720CB2DEA702C46D0A5BEF6DC76B2A92083D1CF26328B40A65EFE279F20D8A3F238899900A1455423FE09DDFABEF4D89C21F3FDFBED65B9E8939DBE0D239046E
4354B851F2725308666C38D09726ACCF697E3F9123A70691569C208BA93106DE9AA703482406A6787A35668F3E66C4ED881DBF35828C136F98ECF3AC019F9F75226D8D58F4AF792F734E8673AC18C439E16150825CD62466930630403DEB5639349630853A1699B4
1DF7343A650C21851E15078F06D400BA96FB42FFEB3B4D379D1C196C413967D83330CF2D2ABF95CAAB6B80B9C52A7CB032E8498B5A5CDC34F755C23BA4B54E91E979DA8F0864FDB40024D85F24F67D769B661334D3CB3348819BB9D5C5A40EEEBD3951085051B29CA
593D7F654F148584301C570191E85CD4170FC58A90F8C59C8372003A8A4275A0F835E3156AA4A5D32B2AE0871C4017BC779031F94F30FA5B4CCDAA6F09ED67422F4152C9FD5566EAF84646EDCDBEEF778290AE5CD4EB61B995F43BD7CDDF5A389245536F0AC88573E
BF7E616F01DDAE2F5CF93A84EFB70B96B42DAEC8FBCB3E400A1FC9CB6441A007AD32DE69E656D2CA73523DF2601D2D2217500422963B367D199F9137CD9B9765070346CA63192083D1CF26328B40A65EFE279F20D8AF5DA3B74C366883E9DC56528DB82C609E6AB0E
9A804C46ADA973D9B308BFF88ACB344427CC273978B2B86A070528D51A5BF783C6556040A8412D647253E59BEC079285F97BA97C2DC631F15AF1A2A83F016BEF4C3DA7B0659CAEC467E06BCBE1A602BBA6DFFCD61765E1F3A75A4D047772185814A35E339B0C26BA4
56CC2471DCB487FD02AFE3F23E3A2F45144A1BF32739635FA76021612D5474F6745F9224D01A8F5EECE6D2229788C9C361AFA46B4 |
| YFy2M9ByuuWt3FIIQOJOH1UELGkFW5VqsJ+a0lx2TVMdhhhkBA/ysXtxlkrQmxfQLrUMTO9hvVnrK9zQGH39qoUnslZ+qbOsJFw8f+sOMuMCGz2JR0C4SoT9z7OMAeMckAi0c7cD+rt6UuK8LAOh/4OMD7vgP1e7cfxZB3PzNtQ= | 0x13386754B3468A9588599FA729C4F8
74BF18CD015E50548A452AED59E31A85605096534CA9F08CFC9B4A6E6C2F9F3C1F663787885F27C6D6FC06C7D3AC6603815F770503DF26709FF04C427392756570CECAE696F4AD30602756B779C91E1EB2816A2163ACEBFF04C1F605F10A332D3ECBCECDAC3034E07
EF043175843F6A9374BEFA6A70C41D928604229CA73EF7D23416646C27AA9F110A71686D49E15B01A7EC466B3350B950053D655BAF96C2ECDA32737BBAA90CDC6AB9CAA2D60F0DD86338E2C3FC7E49A5A82445EDD992D650A80F926F9A77BA0267FC9A2875800A6F7
B0AF9466F918F6E3D774B7490913A4D48D9F3D9BBEAB01CB2DAEBF96322B19F158831CE0A7643D03E97EF9F2EFF0094E749CD5A76396198B373D3DE764C80A1D4293642F2BC498015B167F5D69220A0B7123BCD23D10886F0DDDB903F0ABF04770CB93A1F4E9CCF1F
EF0336CABE1D95CC1B715BF762DA4CBA063D830677FB0BFE09567EF1EDE26B89A0538C0BBF27C9D8F287DA82AEA09BB7999A86296DFF5432BDE091D4610F334371FE218E57122360163DD50CD5A4B53092EDBF21F5AC989F842CCCE7E30CC87C670DABB5E9A2A88ED
4CD72C889241BCCF7FC56BBE5F7B5415C551A0D8BB0E372315D590CC3DC2C4A8CFE3EBFBF30B5D511C70A51B61EA16A3987FDE81870ED94933ABA8F3CD13F9AD72D1A9BF7BA733F521F4F99C1A8545C06D19D7362F83A7ECA9B23D7651B14463C08FD7B35D1C2ED7A
6D999DB387DD47EC466B3350B950053D655BAF96C2ECDED867579061132C829F4998BEAC19B200A53A855A95D64DD1EE429F5AD9492E348F8FBA7B5EEA47DC1CCF9FA4D5BD991798B422FADEE641E31F31F6424B3B50EDED8FE9E6E6CBD04DF0F6BADB3FB3186149B
1DE8936F7FF17817284F29FC2193217DD0889E920C6347E9F62CBF929B91CC886778E3692392F9BBF738084BFB1FC84CB4B2C8CD0D0A97C305805F375AE0FE9393E33990DD9E61B76E0754DBD9D7375A764EAFA27220436DE9402494603E5049C3D5659C7384AA50E
2132B7FC7A943FC92E31D806ED6F848FA1DD547AEEC01A0E439E172ADDDE7ACAB913AEAC7B90D5B66617B54D89DCE32FB9A5C5D13DA338E2C3FC7E49A5A82445EDD992D650A80F926F9A77BA0267FC9A2875800A6F7F63CB2283E01505AEC47F7DF7D74B09087B003
B9971364672DA54BF6F72187EE04E4377040F158CE4F68A13143F40BE9B3E66E60AB35D61030B1057D56D61413BD7BCC41DD3F8B291DEA9CF27F28114C30DCED2C36F739692D95BC7580FBBAEA22B5E22867910F45391E60B2F2B19D33712AE87E0A30285F367CA67
E54CBEDE38177DFE5DEFC5005DCB59DFECF917C782115C4861B93559DA56F49C162662190E0F22442D505603B59B5382CD609F222 |
+-----------------------------------------------------------------------------------------------------------+------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
```

Fig 13. Saved File on server

File Size (Kb)	**Encryption Time**	**Decryption Time**	**Throughput(Kb/Sec)**
5	**0.471**	**0.213**	**14.619**
10	**0.677**	**0.340**	**19.860**
15	**0.864**	**0.401**	**23.715**
20	**1.141**	**0.689**	**21.857**
25	**1.602**	**0.911**	**19.89**
30	**1.920**	**1.027**	**20.689**

Table 2. Performance Evaluation in time

Chapter 9
Deployment Strategies

- **Client-Side DNA Encryption:**
    - The process of encoding user data into DNA sequences occurs at the client-side. This step involves converting binary data into DNA sequences using sophisticated encoding methods. The transformed DNA-encoded data is securely transmitted to the middleware for further processing.

- **Middleware for Key Management and Cryptography Splitting:**
    - The middleware layer is responsible for managing key generation and Cryptography Splitting.
    - Key Management System (KMS): The generation of unique encryption keys for data blocks takes place within the middleware. These keys are attribute-based and tailored to specific user-defined attributes, ensuring secure encryption.
    - Cryptography Splitting: The middleware divides the DNA-encoded data into distinct file blocks. Each block undergoes AES encryption using the generated keys from the KMS. The encrypted blocks, along with metadata like block addresses, are prepared for storage in the cloud servers.

- **Individual Cloud Servers with RSA Encryption:**
    - Each individual cloud server hosts encrypted file blocks and implements the RSA encryption algorithm.
    - RSA Encryption Module: Servers utilize their private keys to encrypt the block addresses attached to the encrypted file blocks. This additional layer of encryption ensures the security of metadata associated with the stored data.
    - During data retrieval, servers use their private keys to decrypt block addresses, allowing them to locate and retrieve the specific encrypted blocks requested by the middleware.

- **Data Storage and Retrieval:**
    - The distributed cloud servers securely store the encrypted file blocks with their associated metadata.
    - Upon retrieval requests, servers decrypt block addresses and provide the encrypted blocks to the middleware for decryption and reassembly into the original user data.

Chapter 10

Result and Analysis

- Experiments in the "DoubleHelix Shield" framework followed a structured approach: clear objectives were set, test environments mirrored real-world scenarios, and the system's components were rigorously tested. From DNA encryption accuracy to RSA encryption effectiveness, key performance indicators were measured and compared against industry benchmarks. Iterative testing allowed for enhancements, with comprehensive documentation capturing protocols and outcomes for further analysis and future improvements. This systematic process ensured robust validation of the framework's functionality, security, and performance in cloud environments.

File Size(kb)	**Encryption Times (s)**				
	BlowFish	**AES**	**DES**	**DNA**	**DoubleHelix**
5	**0.054**	**0.053**	**0.152**	**0.200**	**0.471**
10	**0.074**	**0.081**	**0.283**	**0.483**	**0.677**
15	**0.097**	**0.106**	**0.405**	**0.625**	**0.864**
20	**0.130**	**0.172**	**0.571**	**0.972**	**1.141**
25	**0.133**	**0.198**	**0.785**	**1.250**	**1.602**
30	**0.170**	**0.218**	**0.955**	**1.487**	**1.920**

Table 3. Encryption Time Comparison with other Algorithms

Algorithm	**ThroughPut (Kb/s)**
BlowFish	**180.4**
AES	**159.6**
DES	**126.8**
DNA	**33.32**
DoubleHelix	**20.92**

Table 4. ThroughPut Comparison with other Algorithms

Chapter 11

Applications

- Cloud Security Enhancement: Securing sensitive data stored in distributed cloud systems, ensuring confidentiality and integrity.
- Healthcare Data Protection: Safeguarding electronic health records (EHRs) and patient information with advanced encryption.
- Financial Sector Security: Ensuring secure storage and transmission of financial data, protecting transactions and client details.
- Government and Defense: Protecting classified information and sensitive government data from unauthorized access.
- IoT Security: Securing data transmitted and stored in IoT devices, ensuring privacy and integrity of information.
- Research and Academia: Protecting intellectual property and sensitive research data stored in cloud systems.
- Corporate Data Protection: Safeguarding proprietary information, trade secrets, and customer data stored in the cloud.

Chapter 12

Conclusion

The "DoubleHelix Shield" framework epitomizes a revolutionary stride in fortifying cloud data security through a sophisticated integration of avant-garde technologies. By harmonizing DNA-based cryptography, Cryptographic Splitting, and potent encryption algorithms, this framework forms a robust bulwark against contemporary cyber threats.

Systematic design and rigorous experimentation unveiled the framework's prowess in securing data within cloud environments. DNA-based encryption exhibited exceptional precision and reversibility, establishing a pioneering methodology for securely encoding data. Simultaneously, the Cryptography Splitting algorithm adeptly fragmented and encrypted data, facilitating secure dispersal across diverse cloud servers. Complementing these layers, the application of RSA encryption on individual servers fortified metadata security, enhancing comprehensive data protection.

Conducting systematic experiments across diverse layers validated the framework's performance and resilience. Benchmarking against industry standards underscored its competitive edge and readiness for real-world deployment.

In summation, the "DoubleHelix Shield" framework represents an innovative breakthrough in fortifying cloud data security. Its intricate encryption mechanisms, meticulous design, and demonstrated capabilities position it as a promising vanguard in safeguarding sensitive information within the dynamic landscape of cloud

Future prospects of the project

- Advancements in Data Security: Continual improvements in encryption methodologies and data protection within cloud systems.
- Innovation in DNA Cryptography: Further exploration and development of DNA-based encryption for enhanced security measures.
- Integration with Emerging Technologies: Integration with AI, blockchain, or quantum computing for more robust security protocols.
- Industry-Specific Applications: Tailoring the framework for specific industry needs like healthcare, finance, or IoT for heightened security.
- Standardization and Compliance: Potential contributions to shaping industry standards and compliance frameworks for cloud security.
- Research and Development: Opportunities for further R&D, exploring new encryption techniques or optimizing existing methodologies.
- Global Adoption: Expansion into global markets and industries seeking advanced data protection solutions.

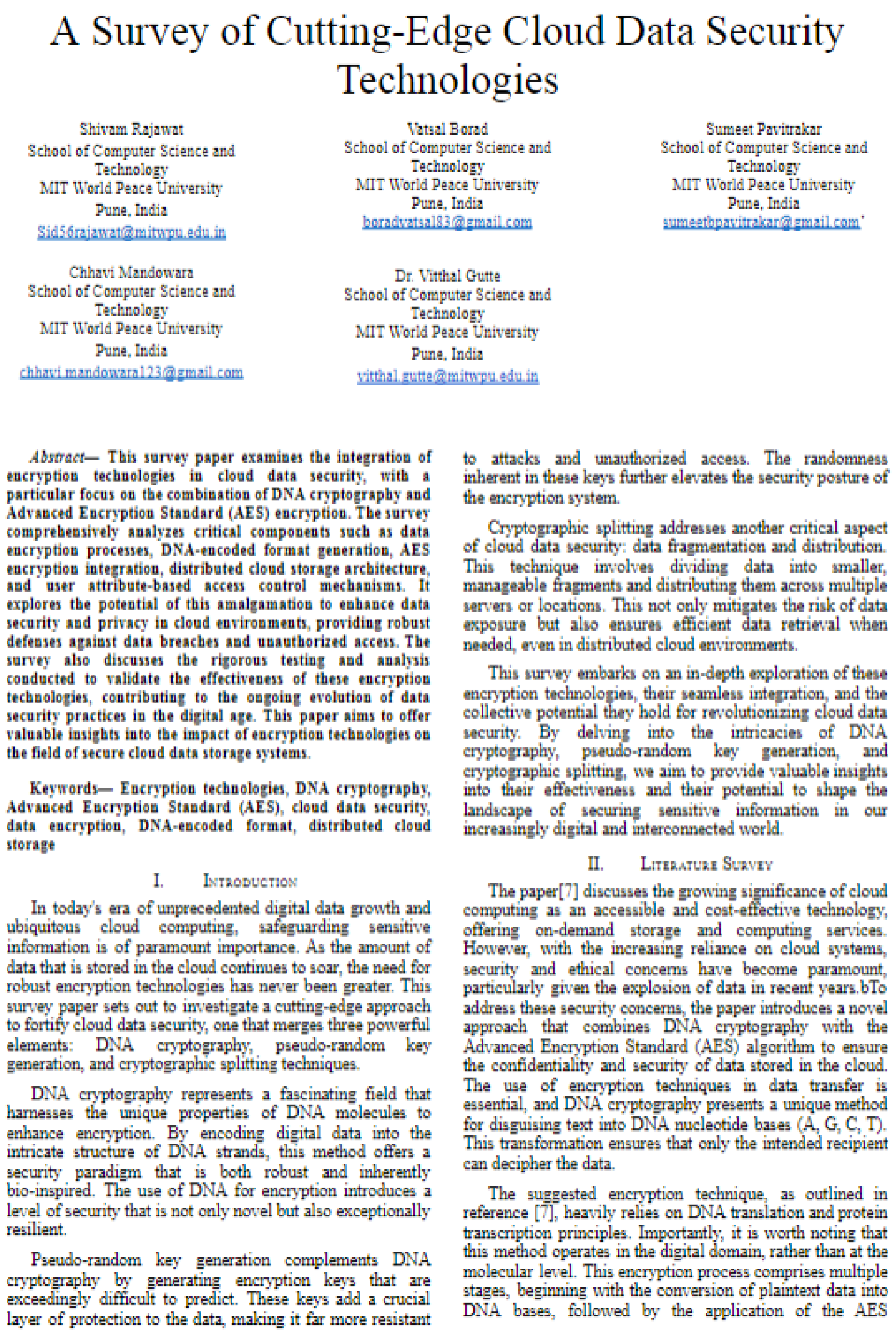
References

01. Y. Xiao, Y. Chen, C. Long, J. Shi, J. Ma and J. He, "A Novel Hybrid Secure Method Based on DNA Encoding Encryption and Spiral Scrambling in Chaotic OFDM-PON," in IEEE Photonics Journal, vol. 12, no. 3, pp. 1-15, June 2020, Art no. 7201215, doi: 10.1109/JPHOT.2020.2987317
02. J. S. Khan et al., "DNA and Plaintext Dependent Chaotic Visual Selective Image Encryption," in IEEE Access, vol. 8, pp. 159732-159744, 2020, doi: 10.1109/ACCESS.2020.3020917.
03. J. P. G. Perez et al., "A Modified Key Generation Scheme of Vigenère Cipher Algorithm using Pseudo-Random Number and Alphabet Extension," 2021 7th International Conference on Computer and Communications (ICCC), Chengdu, China, 2021, pp. 565-569, doi: 10.1109/ICCC54389.2021.9674565.
04. E Ravi Kumar et al, "A Multi-Stage Cloud Security for Cloud Datausing Amalgamate Data Security" 2023 International Conference for Advancement in Technology (ICONAT) | 978-1-6654-7517-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICONAT57137.2023.10080583
05. S. E. El-Khamy, N. O. Korany and A. G. Mohamed, "A New Fuzzy-DNA Image Encryption and Steganography Technique," in IEEE Access, vol. 8, pp. 148935-148951, 2020, doi: 10.1109/ACCESS.2020.3015687.
06. X. Wang and Y. Su, "An Audio Encryption Algorithm Based on DNA Coding and Chaotic System," in IEEE Access, vol. 8, pp. 9260-9270, 2020, doi: 10.1109/ACCESS.2019.2963329.
07. A. Kumar, "Data Security and Privacy using DNA Cryptography and AES Method in Cloud Computing," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2021, pp. 1529-1535, doi: 10.1109/I-SMAC52330.2021.9640708.
08. K. S. Charan, H. V. Nakkina and B. R. Chandavarkar, "Generation of Symmetric Key Using Randomness of Hash Function," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-7, doi: 10.1109/ICCCNT49239.2020.9225280.
09. Y. Yao, Z. Zhai, J. Liu and Z. Li, "Lattice-Based Key-Aggregate (Searchable) Encryption in Cloud Storage," in IEEE Access, vol. 7, pp. 164544-164555, 2019, doi: 10.1109/ACCESS.2019.2952163.
10. S. Galantucci, D. Impedovo and G. Pirlo, "One Time User Key: A User-Based Secret Sharing XOR-ed Model for Multiple User Cryptography in Distributed Systems," in IEEE Access, vol. 9, pp. 148521-148534, 2021, doi: 10.1109/ACCESS.2021.3124637.
11. W. Susilo, F. Guo, Z. Zhao and G. Wu, "PKE-MET: Public-Key Encryption With Multi-Ciphertext Equality Test in Cloud Computing," in IEEE Transactions on Cloud Computing, vol. 10, no. 2, pp. 1476-1488, 1 April-June 2022, doi: 10.1109/TCC.2020.2990201.
12. Prof. Mrs. D S Zingade et al, "SECURE DATA STORAGE ON MULTI-CLOUD USING DNA BASED CRYPTOGRAPHY" Department of Computer Engineering, AISSMS IOIT. International Journal of Advance Engineering and Research Development Volume 2, Issue 3, March -2015

13. O. A. Khashan, "Secure Outsourcing and Sharing of Cloud Data Using a User-Side Encrypted File System," in IEEE Access, vol. 8, pp. 210855-210867, 2020, doi: 10.1109/ACCESS.2020.3039163.
14. Z. Tang, "A Preliminary Study on Data Security Technology in Big Data Cloud Computing Environment," 2020 International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), Bangkok, Thailand, 2020, pp. 27-30, doi: 10.1109/ICBASE51474.2020.00013.
15. Yuan Huihua, "Research on Data Security in Big Data Cloud Computing Environment[J]", Information Technology and Informatization, 2019.
16. B. S. V. Krishna, S. S. Rao and M. K. Prasad, "Security on Data Auditing Protocols for Cloud Storage Data," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2019, pp. 6-10, doi: 10.1109/I-SMAC47947.2019.9032518.
17. J. Zhang, "Research on the Application of Computer Big Data Technology in Cloud Storage Security," 2021 IEEE International Conference on Data Science and Computer Application (ICDSCA), Dalian, China, 2021, pp. 405-409, doi: 10.1109/ICDSCA53499.2021.9650284.
18. K. Hossain, M. Rahman and S. Roy, "Iot data compression and optimization techniques in cloud storage: current prospects and future directions", International Journal of Cloud Applications and Computing (IJCAC), vol. 9, pp. 43-59, February 2019.
19. Sohal, Manreet & Sharma, Sandeep. (2018). BDNA-A DNA Inspired Symmetric Key Cryptographic Technique to Secure Cloud Computing. Journal of King Saud University - Computer and Information Sciences. 34. 10.1016/j.jksuci.2018.09.024.

Publication Details

Survey Paper



A Survey of Cutting-Edge Cloud Data Security Technologies

Shivam Rajawat
School of Computer Science and Technology
MIT World Peace University
Pune, India
Sid56rajawat@mitwpu.edu.in

Vatsal Borad
School of Computer Science and Technology
MIT World Peace University
Pune, India
boradvatsal83@gmail.com

Sumeet Pavitrakar
School of Computer Science and Technology
MIT World Peace University
Pune, India
sumeetbpavitrakar@gmail.com'

Chhavi Mandowara
School of Computer Science and Technology
MIT World Peace University
Pune, India
chhavi.mandowara123@gmail.com

Dr. Vitthal Gutte
School of Computer Science and Technology
MIT World Peace University
Pune, India
vitthal.gutte@mitwpu.edu.in

***Abstract*— This survey paper examines the integration of encryption technologies in cloud data security, with a particular focus on the combination of DNA cryptography and Advanced Encryption Standard (AES) encryption. The survey comprehensively analyzes critical components such as data encryption processes, DNA-encoded format generation, AES encryption integration, distributed cloud storage architecture, and user attribute-based access control mechanisms. It explores the potential of this amalgamation to enhance data security and privacy in cloud environments, providing robust defenses against data breaches and unauthorized access. The survey also discusses the rigorous testing and analysis conducted to validate the effectiveness of these encryption technologies, contributing to the ongoing evolution of data security practices in the digital age. This paper aims to offer valuable insights into the impact of encryption technologies on the field of secure cloud data storage systems.**

Keywords— Encryption technologies, DNA cryptography, Advanced Encryption Standard (AES), cloud data security, data encryption, DNA-encoded format, distributed cloud storage

I. Introduction

In today's era of unprecedented digital data growth and ubiquitous cloud computing, safeguarding sensitive information is of paramount importance. As the amount of data that is stored in the cloud continues to soar, the need for robust encryption technologies has never been greater. This survey paper sets out to investigate a cutting-edge approach to fortify cloud data security, one that merges three powerful elements: DNA cryptography, pseudo-random key generation, and cryptographic splitting techniques.

DNA cryptography represents a fascinating field that harnesses the unique properties of DNA molecules to enhance encryption. By encoding digital data into the intricate structure of DNA strands, this method offers a security paradigm that is both robust and inherently bio-inspired. The use of DNA for encryption introduces a level of security that is not only novel but also exceptionally resilient.

Pseudo-random key generation complements DNA cryptography by generating encryption keys that are exceedingly difficult to predict. These keys add a crucial layer of protection to the data, making it far more resistant to attacks and unauthorized access. The randomness inherent in these keys further elevates the security posture of the encryption system.

Cryptographic splitting addresses another critical aspect of cloud data security: data fragmentation and distribution. This technique involves dividing data into smaller, manageable fragments and distributing them across multiple servers or locations. This not only mitigates the risk of data exposure but also ensures efficient data retrieval when needed, even in distributed cloud environments.

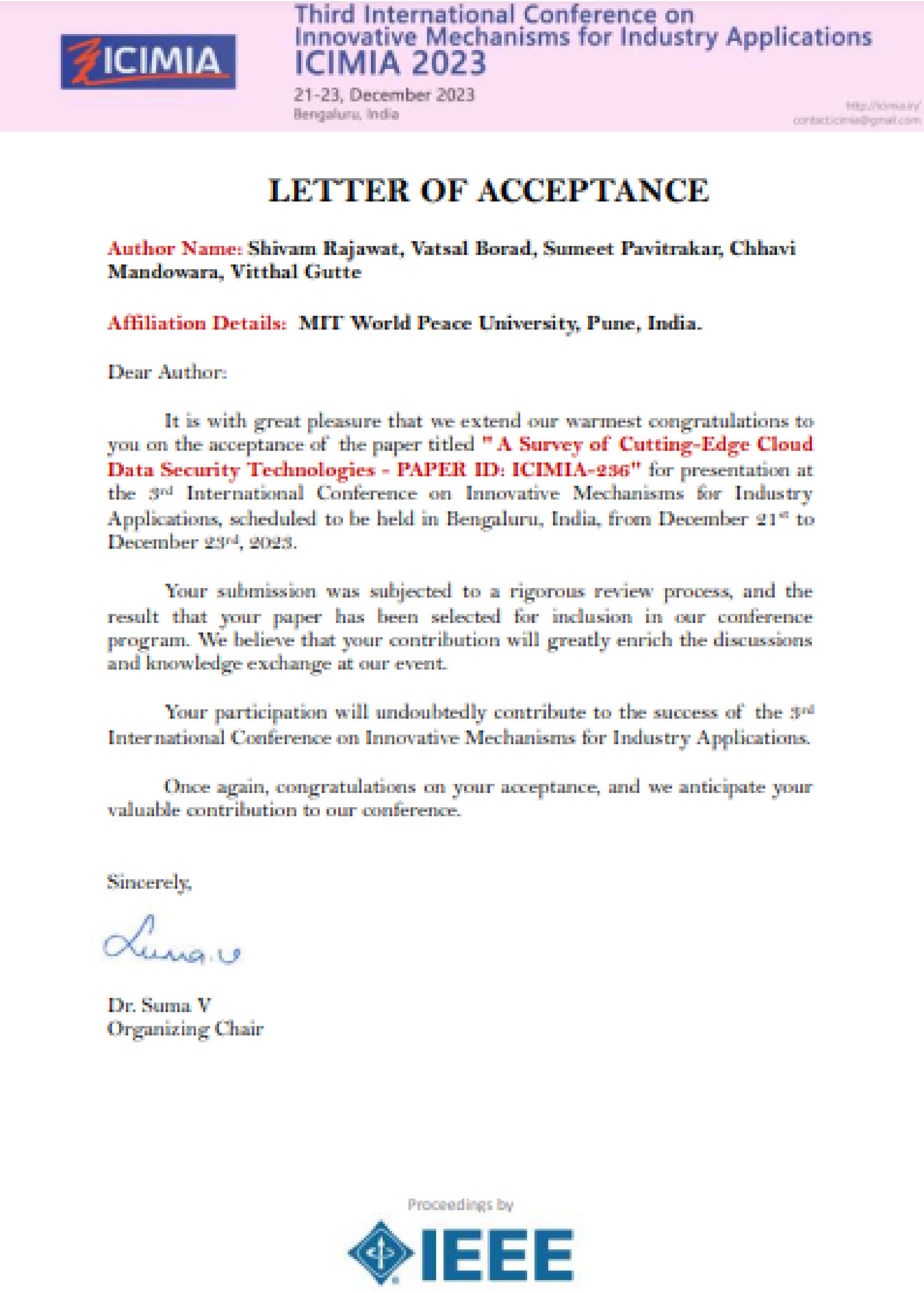
This survey embarks on an in-depth exploration of these encryption technologies, their seamless integration, and the collective potential they hold for revolutionizing cloud data security. By delving into the intricacies of DNA cryptography, pseudo-random key generation, and cryptographic splitting, we aim to provide valuable insights into their effectiveness and their potential to shape the landscape of securing sensitive information in our increasingly digital and interconnected world.

II. Literature Survey

The paper[7] discusses the growing significance of cloud computing as an accessible and cost-effective technology, offering on-demand storage and computing services. However, with the increasing reliance on cloud systems, security and ethical concerns have become paramount, particularly given the explosion of data in recent years.bTo address these security concerns, the paper introduces a novel approach that combines DNA cryptography with the Advanced Encryption Standard (AES) algorithm to ensure the confidentiality and security of data stored in the cloud. The use of encryption techniques in data transfer is essential, and DNA cryptography presents a unique method for disguising text into DNA nucleotide bases (A, G, C, T). This transformation ensures that only the intended recipient can decipher the data.

The suggested encryption technique, as outlined in reference [7], heavily relies on DNA translation and protein transcription principles. Importantly, it is worth noting that this method operates in the digital domain, rather than at the molecular level. This encryption process comprises multiple stages, beginning with the conversion of plaintext data into DNA bases, followed by the application of the AES

Acceptance Letter for Survey Paper



Third International Conference on Innovative Mechanisms for Industry Applications
ICIMIA 2023
21-23, December 2023
Bengaluru, India

http://icimia.in/
contact.icimia@gmail.com

LETTER OF ACCEPTANCE

Author Name: Shivam Rajawat, Vatsal Borad, Sumeet Pavitrakar, Chhavi Mandowara, Vitthal Gutte

Affiliation Details: MIT World Peace University, Pune, India.

Dear Author:

It is with great pleasure that we extend our warmest congratulations to you on the acceptance of the paper titled **" A Survey of Cutting-Edge Cloud Data Security Technologies - PAPER ID: ICIMIA-236"** for presentation at the 3rd International Conference on Innovative Mechanisms for Industry Applications, scheduled to be held in Bengaluru, India, from December 21st to December 23rd, 2023.

Your submission was subjected to a rigorous review process, and the result that your paper has been selected for inclusion in our conference program. We believe that your contribution will greatly enrich the discussions and knowledge exchange at our event.

Your participation will undoubtedly contribute to the success of the 3rd International Conference on Innovative Mechanisms for Industry Applications.

Once again, congratulations on your acceptance, and we anticipate your valuable contribution to our conference.

Sincerely,

Suma.V

Dr. Suma V
Organizing Chair

Proceedings by
IEEE

Implementation Paper

Enhanced Cloud Security: DNA Encryption and AES Integration with Cryptographic Splitting

Shivam Rajawat, Vatsal Borad, Chhavi
Mandowara, Sameer Pavitrakar, Dr
Vitthal Gutte
School of Computer Science and
Technology
MIT World Peace University
Pune, India
Sid58rajawat@mitwpu.edu.in
, boradvatsal83@gmail.com,
sameerpavitrakar@gmail.com,
chhavi.mandowara123@gmail.com,
vitthal.gutte@mitwpu.edu.in

Abstract— In today's cloud-driven landscape, safeguarding sensitive data stands as a pivotal challenge. This research presents "DoubleHelix Shield," an innovative data security framework integrating DNA cryptography with Cryptographic Splitting through AES encryption for robust cloud storage protection. The study delineates four primary modules that collectively ensure a comprehensive encryption mechanism.

The initial module orchestrates DNA encryption at the user end, managing file encoding and encryption keys. This interface links with subsequent modules, notably a Key Management System (KMS) generating and distributing secure encryption keys. The subsequent middleware splits files, applies AES encryption to discrete blocks with unique keys from the KMS, and disperses them across servers. The final module manages block storage and encrypts block addresses for secure data handling. By collaboratively orchestrating these modules, the system enables secure data transmission, storage, and retrieval. During retrieval, the modules collectively decrypt and reassemble the file for user access. This research not only proposes an innovative encryption system but also tackles inherent challenges in cloud data security, offering a holistic approach to protect sensitive data in cloud environments.

Keywords— Encryption technologies, DNA cryptography, Advanced Encryption Standard (AES), cloud data security, data encryption, Cryptographic splitting

I. Introduction

The exponential growth of cloud-based infrastructures has revolutionized data storage and accessibility, yet it introduces significant security challenges, particularly concerning the confidentiality and integrity of sensitive information. In response to these challenges, this research presents an innovative data security framework named "DoubleHelix Shield." This framework amalgamates two potent encryption paradigms, DNA cryptography and Cryptographic Splitting, seamlessly integrating them with the Advanced Encryption Standard (AES) for robust and comprehensive cloud data protection.

The emergence of novel encryption methodologies has sparked innovative approaches to fortify data security in cloud environments. This research endeavors to contribute to this evolving landscape by introducing "DoubleHelix Shield," an integrated data security framework designed to offer robust protection leveraging DNA cryptography, Cryptographic Splitting, and AES encryption within cloud storage systems.

Cloud computing's fundamental principle of ubiquitous data access and storage, while transformative, has spurred a corresponding rise in cybersecurity threats. Unauthorized access, data breaches, and privacy violations have become pressing concerns for both users and service providers. The inherent vulnerabilities in traditional encryption methods have underscored the need for innovative and multi-layered security mechanisms capable of withstanding sophisticated cyber threats.

The premise of DoubleHelix Shield stems from the fusion of advanced encryption techniques, each contributing distinct layers of security. DNA cryptography represents a pioneering field harnessing the inherent properties of DNA molecules for data encryption. This bio-inspired approach capitalizes on the complex and unique structure of DNA strands, encoding digital data into this biological framework to create a novel and resilient security paradigm.

Complementing DNA cryptography, the framework incorporates Cryptographic Splitting, a technique addressing data fragmentation and distribution. By dividing data into smaller fragments and dispersing them across diverse servers, this methodology mitigates the risk of data exposure and enables efficient data retrieval even within distributed cloud architectures. Moreover, the integration of the Advanced Encryption Standard (AES) further fortifies the encryption process, utilizing pseudo-random key generation to augment data security and resilience against unauthorized access.

The research delineates four primary modules constituting the DoubleHelix Shield framework, each contributing crucial functionalities to ensure comprehensive data encryption and storage security. The initial module focuses on DNA encryption at the user's end, managing the encoding of files and encryption keys based on user-defined attributes.

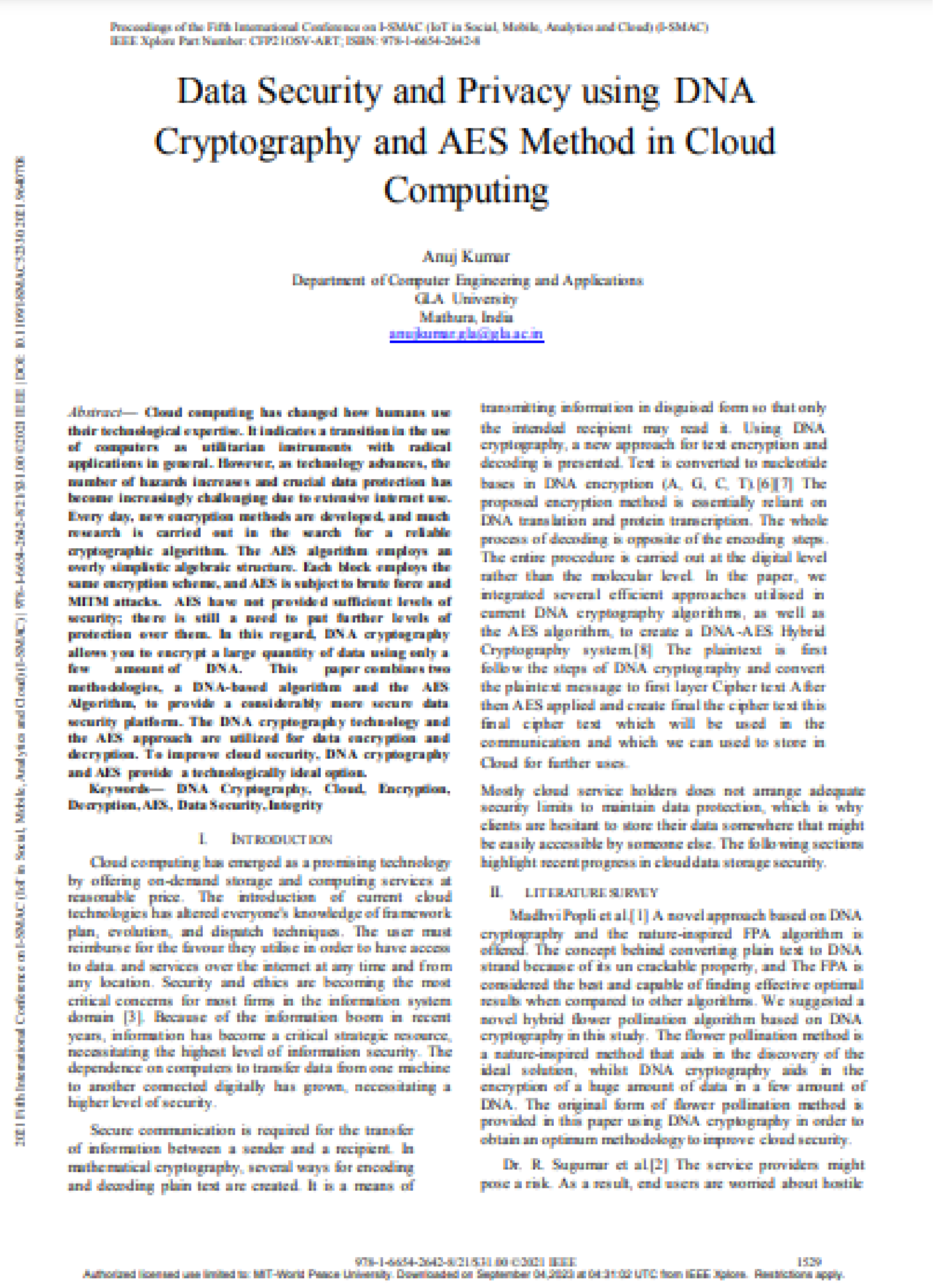
Subsequently, a Key Management System (KMS) serves as the central middleware responsible for generating and distributing secure encryption keys. This module interacts intricately with other components, especially the middleware handling Cryptographic Splitting and AES encryption, which further disperse encrypted file blocks across diverse servers.

The final module oversees block storage and encrypts block addresses using its own public key, ensuring secure data handling and accessibility. Collaboration among these

Submitted To

Appendices

A. Base Paper



Proceedings of the Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)
IEEE Xplore Part Number: CFP21OSV-ART; ISBN: 978-1-6654-2642-8

Data Security and Privacy using DNA Cryptography and AES Method in Cloud Computing

Anuj Kumar
Department of Computer Engineering and Applications
GLA University
Mathura, India
anujkumar.gla@gla.ac.in

2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) | 978-1-6654-2642-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/I-SMAC52330.2021.9640706

***Abstract*— Cloud computing has changed how humans use their technological expertise. It indicates a transition in the use of computers as utilitarian instruments with radical applications in general. However, as technology advances, the number of hazards increases and crucial data protection has become increasingly challenging due to extensive internet use. Every day, new encryption methods are developed, and much research is carried out in the search for a reliable cryptographic algorithm. The AES algorithm employs an overly simplistic algebraic structure. Each block employs the same encryption scheme, and AES is subject to brute force and MITM attacks. AES have not provided sufficient levels of security; there is still a need to put further levels of protection over them. In this regard, DNA cryptography allows you to encrypt a large quantity of data using only a few amount of DNA. This paper combines two methodologies, a DNA-based algorithm and the AES Algorithm, to provide a considerably more secure data security platform. The DNA cryptography technology and the AES approach are utilized for data encryption and decryption. To improve cloud security, DNA cryptography and AES provide a technologically ideal option.**

Keywords— DNA Cryptography, Cloud, Encryption, Decryption, AES, Data Security, Integrity

I. INTRODUCTION

Cloud computing has emerged as a promising technology by offering on-demand storage and computing services at reasonable price. The introduction of current cloud technologies has altered everyone's knowledge of framework plan, evolution, and dispatch techniques. The user must reimburse for the favour they utilise in order to have access to data and services over the internet at any time and from any location. Security and ethics are becoming the most critical concerns for most firms in the information system domain [3]. Because of the information boom in recent years, information has become a critical strategic resource, necessitating the highest level of information security. The dependence on computers to transfer data from one machine to another connected digitally has grown, necessitating a higher level of security.

Secure communication is required for the transfer of information between a sender and a recipient. In mathematical cryptography, several ways for encoding and decoding plain text are created. It is a means of transmitting information in disguised form so that only the intended recipient may read it. Using DNA cryptography, a new approach for text encryption and decoding is presented. Text is converted to nucleotide bases in DNA encryption (A, G, C, T).[6][7] The proposed encryption method is essentially reliant on DNA translation and protein transcription. The whole process of decoding is opposite of the encoding steps. The entire procedure is carried out at the digital level rather than the molecular level. In the paper, we integrated several efficient approaches utilised in current DNA cryptography algorithms, as well as the AES algorithm, to create a DNA-AES Hybrid Cryptography system.[8] The plaintext is first follow the steps of DNA cryptography and convert the plaintext message to first layer Cipher text After then AES applied and create final the cipher text this final cipher text which will be used in the communication and which we can used to store in Cloud for further uses.

Mostly cloud service holders does not arrange adequate security limits to maintain data protection, which is why clients are hesitant to store their data somewhere that might be easily accessible by someone else. The following sections highlight recent progress in cloud data storage security.

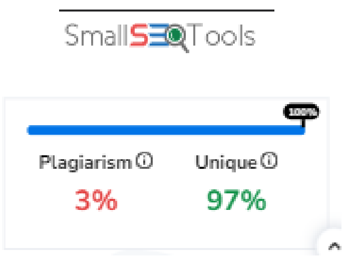
II. LITERATURE SURVEY

Madhvi Popli et al.[1] A novel approach based on DNA cryptography and the nature-inspired FPA algorithm is offered. The concept behind converting plain text to DNA strand because of its un crackable property, and The FPA is considered the best and capable of finding effective optimal results when compared to other algorithms. We suggested a novel hybrid flower pollination algorithm based on DNA cryptography in this study. The flower pollination method is a nature-inspired method that aids in the discovery of the ideal solution, whilst DNA cryptography aids in the encryption of a huge amount of data in a few amount of DNA. The original form of flower pollination method is provided in this paper using DNA cryptography in order to obtain an optimum methodology to improve cloud security.

Dr. R. Sugumar et al.[2] The service providers might pose a risk. As a result, end users are worried about hostile

978-1-6654-2642-8/21/$31.00 ©2021 IEEE 1529

B. Plagiarism Report from any open source



SmallSEOTools

100%

Plagiarism 3%

Unique 97%