

Secure Data Sharing Mechanism in Cloud Computing With Keyword Search Base ABE Algorithm

Shaik Riyaz¹, S. Noortaj²

¹M.C.A Student, Department of M.C.A, KMMIPS, Tirupati (D.t), Andhra Pradesh, India

²Assistant Professor, Department of M.C.A, KMMIPS, Tirupati (D.t), Andhra Pradesh, India

ARTICLE INFO

Article History:

Accepted : 01 May 2025

Published: 05 May 2025

Publication Issue

Volume 11, Issue 3

May-June-2025

Page Number

161-169

ABSTRACT

Attribute-Based Proxy Re-Encryption schemes are designed to make secure data sharing possible based on user attributes, all without the need for a centralized authority to handle encryption or re-encryption. However, many of these ABPRE models fall short when it comes to supporting efficient keyword searches and dynamic keyword updates, often relying on a Private Key Generator. To address this gap, Ge et al. introduced the CPAB-KSDS scheme, which is a cipher text-policy ABPRE framework that allows for secure keyword searches and updates without needing PKG involvement. They claim that CPAB-KSDS maintains indistinguishability against chosen-cipher text attacks and chosen-keyword attacks within the random oracle model. In our review of Ge et al.'s CPAB-KSDS scheme, we've identified some major weaknesses in how they reduce security from IND-CKA to the foundational assumptions. Additionally, we've outlined a particular attack that undermines the IND-CKA security of CPAB-KSDS. Our research indicates that CPAB-KSDS falls short of the IND-CKA security it purports to offer, exposing critical vulnerabilities in encryption schemes that incorporate keyword search and update features.

Keywords: Cryptanalysis, Attribute-Based Encryption, Proxy Re-Encryption, Searchable Encryption, Keyword Update, Indistinguishability, Security Flaws

Introduction

In today's world, where data is king, the need for secure, flexible, and detailed access control systems has skyrocketed. With the rise of cloud computing and distributed systems, ensuring the safe exchange and management of sensitive information—like electronic health records, financial transactions, and

intellectual property—has become absolutely crucial.

While traditional encryption methods do a good job of keeping data confidential, they often struggle with providing dynamic and scalable access control in diverse, multi-user settings.

This delegation of re-encryption authority means that data owners can keep control over their content while

still enabling access based on attributes and policies. Even better, ABPRE schemes are designed to work without needing a constant connection to the data owner or a central authority during decryption, which greatly boosts system scalability and user-friendliness.

The advancement of Ciphertext-Policy ABPRE takes these features even further by linking access policies directly to the ciphertexts, making them align more closely with real-world access control policies in organizations and institutions. In CP-ABPRE schemes, each ciphertext is tied to a specific access policy, meaning only users whose attributes meet the criteria can decrypt the information. This makes CP-ABPRE especially well-suited for fine-grained, role-based access control systems, where different users or entities have varying levels of access. Consumer trust and satisfaction.

While the idea of these security claims sounds great in theory, most folks in the cryptographic community agree that security proofs and reductions can be pretty complicated, prone to errors, and tough to validate. A security reduction is supposed to demonstrate that breaking the proposed scheme would mean tackling a known hard problem (like Decisional Bilinear Diffie-Hellman or Discrete Logarithm), usually within a certain computational model. But the reality is, these reductions often come with complex assumptions, probabilistic arguments, and depend on idealized concepts like random oracles—models that don't always hold up in real-world applications.

Many studies have shown that even small flaws in security proofs or incorrect assumptions can leave a cryptographic scheme open to attacks, even if it looks secure on paper. That's why it's crucial to thoroughly examine security reductions and assumptions, especially in intricate systems like CPAB-KSDS that aim to provide multiple advanced features (like encryption, proxy re-encryption, keyword search, and dynamic updates) all at once. Merging these functionalities into a single framework not only

broadens the attack surface but also complicates security reasoning, which can heighten the risk of hidden vulnerabilities.

RELATED WORKS

In their paper, Liang et al. (2009) introduce an innovative cryptographic approach known as Attribute-Based Proxy Re-Encryption, which comes with delegation capabilities. This scheme is designed to meet the increasing demand for secure and adaptable data sharing in distributed systems. The authors enhance traditional proxy re-encryption methods by incorporating attribute-based encryption, which allows access and re-encryption rights to be determined by descriptive attributes instead of fixed identities. This approach facilitates a detailed access control system that can handle complex access policies and securely delegate re-encryption tasks to semi-trusted proxies, all while keeping the underlying plaintext safe. The paper outlines a formal security model for the proposed scheme and shows how it stands up against various cryptographic threats. Additionally, Liang et al. evaluate the computational demands and practicality of applying ABPRE in real-world situations, like cloud computing and secure content distribution. Their research makes a significant impact in the realm of data security by paving the way for more scalable, flexible, and privacy-conscious data sharing solutions in environments with multiple users.

In their 2014 paper, Zheng, Xu, and Ateniese introduce an innovative approach called VABKS, which stands for Verifiable Attribute-Based Keyword Search. This scheme is designed to tackle the challenges of securely and efficiently retrieving data from the cloud, especially when it comes to encrypted information. The authors point out the shortcomings of traditional keyword search methods in encrypted settings, particularly the issues with fine-grained access control and the ability to verify results. To overcome these hurdles, VABKS merges attribute-based encryption with searchable encryption,

allowing users to search for keywords in encrypted data while ensuring that only those who meet specific attribute-based access criteria can access the results. Additionally, the scheme includes a verification feature, enabling users to check the accuracy and completeness of the search results provided by the untrusted cloud server. Zheng and his colleagues also lay out a formal security model for VABKS and demonstrate its resilience against adaptive chosen-keyword attacks. Their experiments show that the scheme is practical, balancing search efficiency with verification costs. This research pushes the boundaries of privacy-preserving cloud data services by offering secure, controlled, and verifiable search options.

EXISTING METHOD

The current methods for transferring data are grappling with a host of serious challenges that impact both security and efficiency. Traditional approaches, like physically moving storage devices or using basic file-sharing over networks, often fall short when it comes to protecting sensitive information. These shortcomings leave data vulnerable to breaches during transmission, particularly when encryption protocols are either weak or outdated. Risks such as unauthorized access, data tampering, and interception by malicious actors are significant concerns, especially in fields where data confidentiality is crucial, like healthcare, finance, and defences.

While physical data transfers might seem more secure because they're offline, they can be incredibly time-consuming and labour-intensive. They involve manual handling, logistical coordination, and often require people to be physically present to send and receive data. This makes the whole process slow, inefficient, and expensive. Plus, physical transfer methods come with their own set of risks, such as

hardware damage, loss, or theft, which can further jeopardize data integrity and continuity.

Even in the digital realm, moving large amounts of data between different locations or systems can lead to hefty costs. These expenses can include network infrastructure, high bandwidth usage, and licensing fees for specialized data transfer tools or platforms. Additionally, ensuring that only authorized personnel can access and initiate data transfers adds an administrative burden, often requiring multiple layers of verification and manual oversight. Keeping data intact during transit also becomes a tricky task, especially when it has to navigate through various systems or networks.

Disadvantages

- Data Breaches Due to Inadequate Security
- Time-Consuming Physical Transfers
- High Transfer Costs
- Complex Authorization Processes

PROPOSED METHOD

The proposed system takes a big leap forward in the world of Attribute-Based Proxy Re-Encryption (ABPRE) by addressing the security flaws found in the CPAB-KSDS framework that Ge et al. introduced. While the original CPAB-KSDS scheme was groundbreaking for merging Ciphertext-Policy Attribute-Based Encryption (CP-ABE) with keyword search capabilities, it has some serious vulnerabilities. These include being vulnerable to chosen-ciphertext attacks (IND-CCA) and chosen-keyword attacks (IND-CKA), along with relying on a centralized Private Key Generator (PKG). Such weaknesses put both privacy and the practicality of secure data sharing and retrieval at risk, especially in sensitive areas like cloud computing and secure information systems.

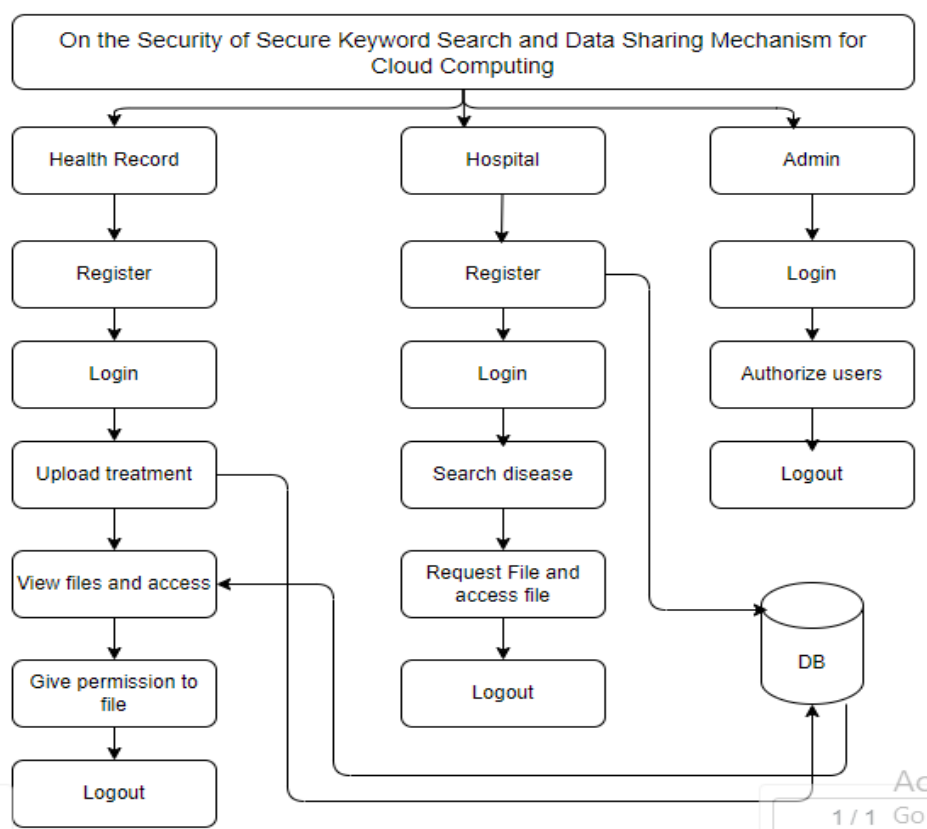


Figure 1: Flow of Proposed Method

To tackle these issues, the proposed system rolls out a fresh and decentralized ABPRE architecture that seamlessly incorporates secure keyword search and allows for dynamic keyword updates—all without needing a PKG. By removing the PKG, it reduces the chances of a single point of failure and key escrow

problems, which boosts the system's reliability and decentralization. This approach gives data owners and users more control over their encryption keys and access policies, paving the way for a more secure and transparent data-sharing environment.

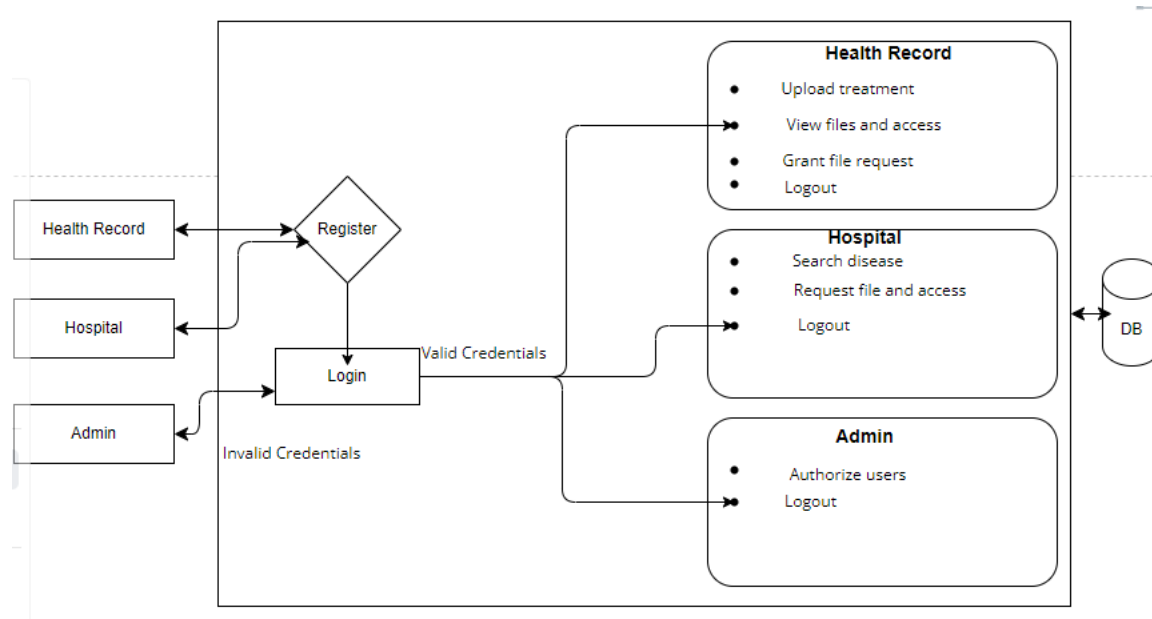


Figure 2: Architecture of Proposed Method

A key feature of this new system is its robust security model, which effectively protects against chosen-ciphertext attacks (IND-CCA) and chosen-keyword attacks (IND-CKA)—two common and powerful threats in the world of cryptography. The system is built on solid cryptographic proofs that ensure adversaries can't extract any useful information from encrypted data or related keywords, even if they get to interact with the decryption or search oracles under specific conditions. This means that both the data and the keywords used for searching stay confidential, even when faced with active and adaptive attackers.

Advantages:

- Enhanced Security
- No PKG Dependency
- Secure Keyword Search.
- Dynamic Keyword Updates

IMPLEMENTATION

IND-CKA Algorithm

Step 1: Key Generation (Setup)

- First off, we generate a secure symmetric key using a reliable cryptographic key generation algorithm.
- This key plays a crucial role, as it's used for both encrypting the keyword and creating the trapdoor, ensuring that only those with the right permissions can access the keyword data.

Step 2: Keyword Encryption

- Next, the keyword gets encrypted based on a specific attribute-based access policy, all thanks to that symmetric key.
- This access policy outlines the conditions under which the keyword can be decrypted, adding an extra layer of control.
- The result? The keyword transforms into a secure ciphertext, keeping it confidential.

Step 3: Trapdoor Generation

- Now, we generate a trapdoor for the keyword, which allows for keyword searches without exposing the actual keyword.

- Think of the trapdoor as an encrypted version of the keyword, created using the same symmetric key. This enables the system to match keywords without revealing the original text.

Step 4: Search and Matching

- The trapdoor is then utilized to search through the encrypted keywords.
- The system can match the trapdoor with the stored encrypted keywords, facilitating keyword searches over encrypted data without needing to decrypt the keywords themselves.

Step 5: Decryption

- If a match is found, the corresponding encrypted keyword is decrypted using the symmetric key.
- The decryption process will only reveal the original keyword if the search request meets the criteria set by the attribute-based access policy.

Step 6: Security Guarantees

- This algorithm achieves IND-CKA security by ensuring that an attacker can't tell the difference between various encrypted keywords, even if they have a trapdoor.
- It effectively prevents keyword guessing attacks and maintains the confidentiality of the keyword data, as unauthorized individuals can't decrypt or deduce the keyword content without the correct decryption key.

Modules

Health Record Module:

Register: To create a health record, you need to register by providing essential information like your name, email, and password.

Login: You can log in to your health record after getting authorization from the admin user.

Upload Treatment: You can add treatment information by uploading a treatment file, which will be automatically encrypted for security.

View and Access: You can view and download the files you've uploaded in your health record.

Give File Access: You can see the file requests from other users and choose to accept or reject them.

Logout: Don't forget to log out of your health record when you're done.

Hospital Module:

Register: To register a hospital, you need to provide necessary details like the name, email, and password.

Login: The hospital can log in after receiving authorization from the admin user.

Search Disease: Hospitals can search for medical data by entering keywords in the search bar and find available medicines from the health record.

Request File: Hospitals can request the necessary medicine files.

Access Files: Once the health record accepts the request, hospital users can access the files.

Logout: The hospital should log out after completing their tasks.

Admin Module:

Login: The admin can log in directly using default credentials.

Authorize Users: The admin is responsible for authorizing users or rejecting any proxy users.

Logout: The admin should log out once all actions are completed.

RESULTS

OUTPUT SCREENS:

1) HOME PAGE:

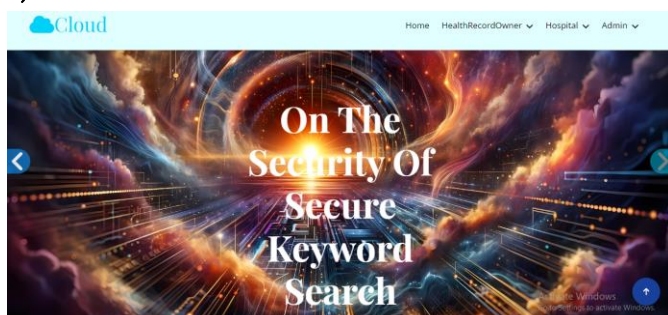


Figure 3: This is the project home, with having Health Record Owner, and Hospital, Admin.

2) HEALTH OWNER REGISTRATION

Figure 4: The Health need to be register to this site to access their account to share the medicines to the required users.

3) HEALTH LOGIN

Figure 5: The health record owner need to be login, using their valid credentials.

4) HEALTH RECORD HOME

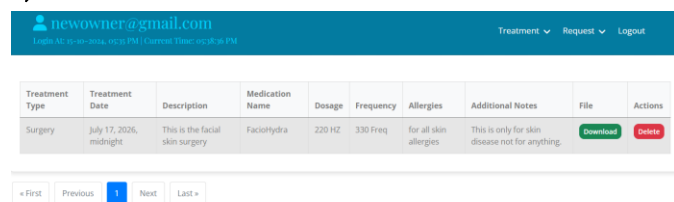


Figure 6: The health record can view their account after successful login, without any issue.

5) Upload Treatment:

Figure 7: The health owner need to be upload the treatment to share the available medicines in the cloud environment.

6) VIEW AND DELETE:

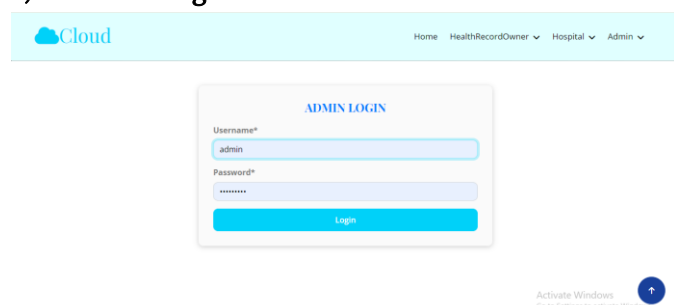


The screenshot shows a web application interface for a treatment record. At the top, there is a header with the user's email 'newowner@gmail.com' and a login time 'Login At: 01:40:55, 05/06/2024'. Below the header, there is a table with columns: Treatment Type, Treatment Date, Description, Medication Name, Dosage, Frequency, Allergies, Additional Notes, File, and Actions. The table contains one row for a surgery treatment. Below the table, there are navigation buttons: '< First', 'Previous', '1', 'Next', and 'Last >'. The '1' button is highlighted, indicating the current page.

Treatment Type	Treatment Date	Description	Medication Name	Dosage	Frequency	Allergies	Additional Notes	File	Actions
Surgery	July 17, 2026, midnight	This is the facial skin surgery	Faciohydra	220 HZ	330 Freq	for all skin allergies	This is only for skin disease not for anything.	Download	Delete

Figure 8: The owner can be delete and view their uploaded treatment data.

7) Admin Login:



The screenshot shows the 'ADMIN LOGIN' form. It has a header with the user's email 'newowner@gmail.com' and a login time 'Login At: 01:40:55, 05/06/2024'. Below the header, there is a form with fields for 'Username*' and 'Password*'. The 'Username*' field contains the text 'admin'. Below the form, there is a 'Login' button. At the bottom right, there is a link to 'Activate Windows'.

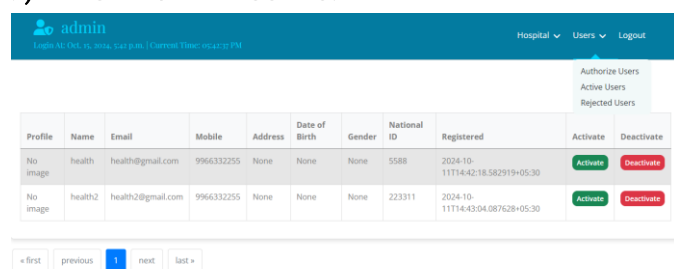
Figure 9: The admin can login using their valid credentials.

8) ADMIN HOME PAGE:



Figure 10: The admin home page

9) AUTHORIZE USERS:

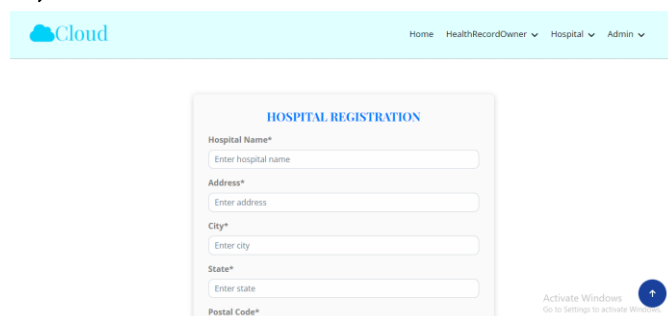


The screenshot shows the 'AUTHORIZE USERS' interface. It has a header with the user's email 'admin' and a login time 'Login At: 01:40:55, 05/06/2024'. Below the header, there is a navigation bar with 'Hospital' and 'Users' dropdown menus, and a 'Logout' button. The main content area shows a table with columns: Profile, Name, Email, Mobile, Address, Date of Birth, Gender, National ID, Registered, Activate, and Deactivate. The table contains two rows for users. Below the table, there are navigation buttons: '< first', 'previous', '1', 'next', and 'last >'. The '1' button is highlighted, indicating the current page.

Profile	Name	Email	Mobile	Address	Date of Birth	Gender	National ID	Registered	Activate	Deactivate
No image	health	health@gmail.com	9966332255	None	None	None	5588	2024-10-11T14:42:18.582919+05:30	Activate	Deactivate
No image	health2	health2@gmail.com	9966332255	None	None	None	223311	2024-10-11T14:43:04.087628+05:30	Activate	Deactivate

Figure 11: The admin need to be authorize the users.

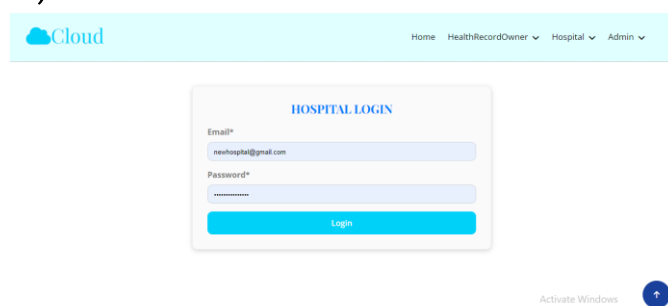
10) HOSPITAL REGISTRATION:



The screenshot shows the 'HOSPITAL REGISTRATION' form. It has a header with the user's email 'newowner@gmail.com' and a login time 'Login At: 01:40:55, 05/06/2024'. Below the header, there is a form with fields for 'Hospital Name*', 'Address*', 'City*', 'State*', and 'Postal Code*'. Below the form, there is a 'Login' button. At the bottom right, there is a link to 'Activate Windows'.

Figure 12: The hospital need to be register with using the required information

11) HOSPITAL LOGIN:



The screenshot shows the 'HOSPITAL LOGIN' form. It has a header with the user's email 'newowner@gmail.com' and a login time 'Login At: 01:40:55, 05/06/2024'. Below the header, there is a form with fields for 'Email*' and 'Password*'. The 'Email*' field contains the text 'newhospital@gmail.com'. Below the form, there is a 'Login' button. At the bottom right, there is a link to 'Activate Windows'.

Figure 13: The hospital need to be login with using their valid credentials.

12) HOSPITAL HOME:

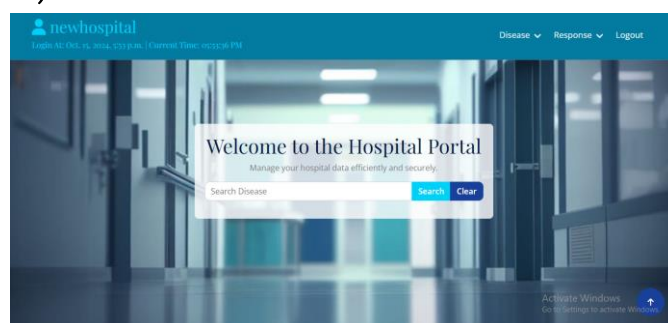


Figure 14: This is the hospital home page. The hospital need to be search the required organ from the health record owner.

13) SEARCH TREATMENT DISEASE:



The screenshot shows the 'SEARCH TREATMENT DISEASE' interface. It has a header with the user's email 'newhospital' and a login time 'Login At: 01:40:55, 05/06/2024'. Below the header, there is a navigation bar with 'Disease' and 'Response' dropdown menus, and a 'Logout' button. The main content area shows a table with columns: Treatment Type, Suitable Disease, Treatment ExpDate, Description, Medication Name, Dosage, Frequency, Allergies, Additional Notes, Original File Name, and Request. The table contains one row for a surgery treatment. Below the table, there are navigation buttons: '< first', 'previous', '1', 'next', and 'last >'. The '1' button is highlighted, indicating the current page.

Treatment Type	Suitable Disease	Treatment ExpDate	Description	Medication Name	Dosage	Frequency	Allergies	Additional Notes	Original File Name	Request
Surgery	Skin disease	July 17, 2026, midnight	This is the facial skin surgery	Faciohydra	220 HZ	330 Freq	for all skin allergies	This is only for skin disease not for anything.	Requirements.txt	Request

Figure 15: The hospital need to be search the disease

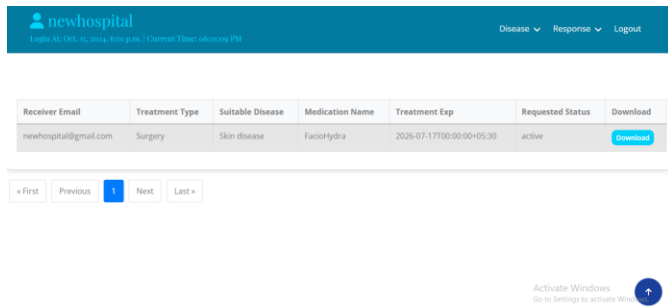
14) VIEW REQUEST RESPONSE:

Figure 15: The hospital user can view the response of the request.

Test cases Model building:

A functional test case is a type of software test case designed to verify that specific features or functions of a system work according to the defined requirements. It focuses on what the system does, rather than how it does it.

A functional test case checks the behavior of a software system against the functional specifications or user requirements. It ensures that each function of the software application operates as expected under specified conditions.

S.NO	Test cases	I/O	Expected O/T	Actual O/T	P/F
1	Registration	Enter name, email, password, address etc.	User data added successfully	User registration successfully	P
2	Registration	Enter name, email, password, address etc.	User data not valid.	Please enter valid information.	F
3	Login	Enter email and password	Password matched	Login successful	F
4	Login	Enter email and password	Password not matched	Login fail	F
5	Upload Treatment	Enter treatment information, with medical file	Treatment file uploaded successfully	Treatment file uploaded successfully	P
6	Upload Treatment	Enter treatment information, with medical file	Upload a supported formats.	File format not supported	F

Table 1: The table presented is a Test Case Summary for validating the core functionalities of a machine learning-based application, possibly one involving dataset handling, user authentication, and classification results.

CONCLUSION

Our in-depth security analysis of Ge et al.'s CPAB-KSDS scheme uncovers some serious flaws in its security reductions, which really puts a dent in the claimed IND-CKA and IND-CCA guarantees. The attack we demonstrated effectively compromises the scheme's integrity, revealing some critical weaknesses in how keyword search and update functionalities are integrated within ABPRE frameworks. These findings highlight the urgent need for stronger cryptographic constructions and more thorough security validations in future ABPRE schemes. It's crucial to ensure secure and efficient data sharing without relying on

centralized authorities, and our work emphasizes the importance of tackling these vulnerabilities to push the field of attribute-based encryption and proxy re-encryption forward.

References

- [1]. X. Liang, Z. Cao, H. Lin, and J. Shao – "Attribute-based proxy re-encryption with delegating capabilities" (2009)
- [2]. Q. Zheng, S. Xu, and G. Ateniese – "VABKS: Verifiable attribute-based keyword search over outsourced encrypted data" (2014)

- [3]. K. Liang and W. Susilo – "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage" (2015)
- [4]. C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski, and L. Fang– "Secure keyword search and data sharing mechanism for cloud computing" (2021)
- [5]. Cong Li, Xinyu Feng, Qingni Shen, and Zhonghai Wu – "On the Security of Secure Keyword Search and Data Sharing Mechanism for Cloud Computing" (2024)
- [6]. Ge et al. – "Ciphertext-policy ABPRE scheme with keyword search (CPAB-KSDS)" (2020)
- [7]. IEEE Transactions on Dependable and Secure Computing – "Secure Data Sharing Mechanism for Cloud Computing" (2023)
- [8]. National Key R&D Program of China – Funding support for cryptographic research (2022)
- [9]. IEEE Transactions on Dependable and Secure Computing – "Flaws in CPAB-KSDS Scheme" (2024)
- [10]. PKU-OCTA Laboratory for Blockchain and Privacy Computing – Research on cryptography (2024)