CHAPTER 1

INTRODUCTION

1.1 Introduction

The project "Strengthening Cloud Computing Security: Mechanisms For Secure Keyword Search And Data Sharing" aims to develop a robust system that ensures the security and privacy of sensitive data stored in cloud environments. By implementing advanced encryption, access control, and homomorphic computation techniques, the project seeks to enable authorized users to securely search for specific keywords within their encrypted data while allowing controlled sharing of encrypted content with selected parties, all while maintaining data confidentiality, integrity, and compliance with privacy regulations.

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.

The cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing. All the existing cloud servers try to store the data in a plain text manner rather than in a encrypted manner. If the encryption key is exposed, the data can be easily accessed by the intruder. In the existing system there is no security for the data even it is encrypted because there is only single cipher key is generated for that data.

As organizations increasingly migrate their data and operations to cloud computing environments, ensuring robust security mechanisms becomes paramount to safeguard sensitive information and maintain the trust of users. Among the critical aspects of cloud security, implementing measures for secure keyword search and data sharing is crucial. This involves protecting data confidentiality, integrity, and accessibility, especially when dealing with sensitive information. In this context, the following strategies are proposed to strengthen cloud computing security mechanisms, with a focus on enhancing the security of keyword searches and facilitating secure data sharing. These strategies encompass encryption, access controls, secure search methods, and proactive monitoring, aiming to create a resilient and trust worthy cloud environment for both enterprises and individual users.

The adoption of cloud computing has revolutionized the way organizations manage and access data, offering unparalleled scalability and flexibility. However, the benefits of cloud computing come with the responsibility to fortify security measures, particularly concerning sensitive data, secure keyword searches, and seamless data sharing. In this era of constant cyber threats, ensuring the confidentiality and integrity of data is critical. This introduction outlines strategies to enhance cloud computing security mechanisms specifically tailored for secure keyword searches and data sharing.

1.2 Motivation

Fouser wants to ensure that their sensitive data remains confidential and not accessible to unauthorized parties. In cloud computing, where data is often stored and processed on third-party servers, maintaining confidentiality is crucial to prevent data breaches unauthorized access. Data integrity ensures that information is not tampered with or corrupted during storage or transmission. Guaranteeing data integrity in cloud environments builds trust among users that their information is accurate and reliable. Many industries are subject to strict regulatory requirements regarding data privacy and security. For instance, healthcare (HIPAA), finance (PCI

DSS), and personal data (GDPR) are all governed by regulations that mandate stringent measures for protecting sensitive information. Adhering to these regulations is not just a matter of best practice but also a legal requirement.

1.3 Objective

The primary objective for privacy protection and data security in cloud computing is to safe guard sensitive information from unauthorized access, disclosure, alteration, or destruction. Ensure that only authorized individuals or systems can access and view sensitive data. Implement encryption, access controls, and authentication mechanisms to protect data from unauthorized access. Guarantee that data remains accurate, consistent, and unaltered throughout its lifecycle. Employ methods such as checksums, digital signatures, and data validation techniques to detect and prevent unauthorized modifications. Align security practices with relevant regulations, standards, and industry best practices. Regularly audit and assess compliance with data protection laws such as GDPR, HIPAA, PCI DSS, and others applicable to specific industries or regions appropriate measures to mitigate these risks. This includes conducting risk assessments, implementing security controls, and establishing incident response procedure. Foster trust with customers by demonstrating a commitment to privacy and security. Communicate Identify potential security risks and vulnerabilities in cloud environments and implement transparently about security practices, certifications, and audit reports to build confidence in the cloud service provider's ability to protect sensitive data. Implement robust monitoring tools and processes to detect and respond to security incidents in real-time. Regularly review and update security policies, procedures, and technologies to adapt to evolving threats and vulnerabilities. Enable customers to maintain control over the location and residency of their data to comply with legal and regulatory requirements. Offer options for data residency, encryption key management, and transparent data handling practices. Provide training and awareness programs for employees, customers, and other stakeholders to promote a culture of security awareness and responsible data handling practices.

1.4 EXISTING SYSTEM:

The traditional attribute-based encryption is not flexible for data searching and sharing. Additionally, attribute-based encryption is not well scaled when there is an update request to the keyword. In order to search and share a specific record, Alice downloads and decrypts thecipher texts. However, this process is impractical to Alice especially when there are a tremendous number of cipher texts. The worse situation is the data owner Alice should stay online all the time because Alice needs to provide her private key for the data decryption. Thus, ABE solution does not take the advantages of cloud computing.

An alternative method is to delegate a third party to do the search, re-encrypt and keyword update work instead of Alice. Alice can store her private key in the third party's storage, and thus the third party can do the heavy job on behalf of Alice. In such an approach, however,we need to fully trust the third party since it can access to Alice's private key. If the third party is compromised, all the user data including sensitive privacy will be leaked as well. It would be a severe disaster to the users.

All the existing cloud servers try to store the data in a plain text manner rather than in a encrypted manner. If the encryption key is exposed, the data can be easily accessed by the intruder. In the existing system there is no security for the data even it is encrypted because there is only single cipher key is generated for that data.

1.5 PROPOSED SYSTEM:

Prior work did not demonstrate that the existing attribute-based mechanisms could both support keyword search and data sharing in one scheme without resorting to PKG. Therefore,a new attribute-based mechanism is needed to achieve the goal for the above PHR scenario. One may argue that the problem can be trivially solved by combining an ABPRE scheme and attribute-based keyword search scheme (AB-KS).

We first introduce a cipher text-policy attribute-based mechanism with keyword search and data sharing (CPAB-KSDS) for encrypted cloud data. The searching and sharing functionalityare enabled in the cipher text-policy setting. Furthermore, our scheme supports the keyword to be updated during the sharing phase. After presenting the construction of our mechanism, we proof its chosen cipher text attack (CCA) and chosen keyword attack (CKA) security in the random oracle model. The proposed construction is demonstrated practical and efficient in the performance and property comparison.

CHAPTER 2
LITERATURE REVIEW

2.1. Overview of Related Work

The literature review is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next steps are to determine which operating system and language used for developing the tool. Once the programmers start building the tool, the programmers need lot of external support. This support obtained from senior programmers, from book or from websites. Before building the system the above consideration taken into account for developing the proposed system.

Privacy protection and data security in cloud computing have garnered significant attention due to the proliferation of cloud-based services and the growing concerns about the confidentiality and integrity of sensitive data stored and processed in cloud environments. Numerous studies have delved into various aspects of privacy and security in the cloud, highlighting the multifaceted challenges and proposing diverse solutions. For instance, research by Ristenpart et al. (2014) investigated the vulnerabilities of cloud storage systems to side-channel attacks, revealing the potential risks posed by shared infrastructure in cloud environments. Similarly, Li et al. (2018) proposed a privacy-preserving mechanism for cloud-based healthcare systems, leveraging techniques such as homomorphic encryption to protect patient data while enabling secure computation. Furthermore, studies by Subashini and Kavitha (2011) and Alizadeh et al. (2018) provided comprehensive surveys of existing privacy protection and security mechanisms in cloud computing, offering valuable insights into the state-of-the-art practices and emerging trends.

Privacy protection and data security in cloud computing have become

paramount concerns in the digital age, given the widespread adoption of cloud-based services and the increasing reliance on cloud infrastructure for data storage, processing, and computation. A plethora of research has been dedicated to understanding and addressing the unique challenges posed by privacy and security in cloud environments. For instance, studies by Ristenpart et al. (2014) have shed light on the vulnerabilities of cloud storage systems to side-channel attacks, emphasizing the need for robust encryption and access control mechanisms to mitigate such risks. Additionally, Li et al. (2018) proposed a novel privacy-preserving framework for cloud-based healthcare systems, leveraging advanced cryptographic techniques such as homomorphic encryption to enable secure computation on encrypted data while preserving patient privacy. Moreover, comprehensive surveys conducted by Subashini and Kavitha (2011) and Alizadeh et al. (2018) have synthesized the existing literature on privacy protection and data security in cloud computing, providing valuable insights into the state-of-the-art practices, emerging trends, and future research directions. These works collectively highlight the complex interplay between privacy, security, and cloud computing, underscoring the importance of adopting a multi-faceted approach that combines encryption, access control, intrusion detection, and other security measures to safeguard sensitive data in the cloud effectively. Moving forward, addressing the evolving threats and challenges in cloud security requires ongoing collaboration between researchers, industry stakeholders, and policymakers to develop innovative solutions and establish robust frameworks for privacy protection and data security in cloud computing.

2.2 Studies And Research in This Field

A rich body of literature exists concerning privacy protection and data security in cloud computing, reflecting the dynamic nature of the field and the ongoing efforts to address emerging challenges. Numerous studies have contributed to our understanding of the intricacies involved in securing cloud-based systems and protecting user privacy. For example, Ristenpart et al. (2014) conducted

pioneering research on the susceptibility of cloud storage systems to side-channel attacks, highlighting the vulnerabilities inherent in shared infrastructure and emphasizing the importance of encryption and access control mechanisms. Building upon this foundation, Li et al. (2018) proposed innovative solutions for preserving privacy in cloud-based healthcare systems, leveraging advanced cryptographic techniques like homomorphic encryption to enable secure computation on encrypted data. Furthermore, extensive surveys by Subashini and Kavitha (2011) and Alizadeh et al. (2018) have provided comprehensive overviews of existing privacy protection and data security mechanisms in cloud computing, synthesizing findings from various research endeavors and identifying key trends and challenges. These studies collectively underscore the significance of ongoing research and development efforts in enhancing the security and privacy of cloud-based services, particularly in the face of evolving threats and regulatory requirements. Moving forward, interdisciplinary collaboration and interdisciplinary collaboration between academia, industry, and policymakers will be essential to drive innovation and establish robust frameworks for privacy protection and data security in cloud computing.

A wealth of literature exists within the domain of privacy protection and data security in cloud computing, reflecting the evolving landscape of technology and the persistent efforts to mitigate associated risks. Researchers have conducted numerous studies aimed at understanding the intricate challenges and developing effective solutions to safeguard sensitive data in cloud environments. Notable contributions include the seminal work by Ristenpart et al. (2014), which investigated the susceptibility of cloud storage systems to side-channel attacks, highlighting the vulnerability of shared infrastructure and advocating for robust encryption and access control mechanisms. Building upon this foundational research, subsequent studies have explored advanced cryptographic techniques to enhance privacy in cloud-based systems. For instance, Li et al. (2018) proposed innovative methodologies for preserving privacy in cloud-based healthcare systems, leveraging homomorphic encryption and secure computation techniques

to enable data analysis without compromising patient confidentiality. Moreover, comprehensive surveys conducted by Subashini and Kavitha (2011) and Alizadeh et al. (2018) have synthesized the findings from a myriad of research endeavors, offering insights into the current landscape of privacy protection and data security mechanisms in cloud computing. These surveys not only provide a comprehensive overview of existing practices but also identify emerging trends and future research directions. As cloud computing continues to evolve, so too do the threats and challenges associated with securing cloud-based data. Hence, ongoing research efforts are essential to develop robust frameworks and innovative solutions that can adapt to the evolving threat landscape and regulatory requirements. Collaboration between academia, industry, and policymakers will be crucial in driving innovation and ensuring the continued resilience of cloud computing environments against emerging security threats.

2.3 Identifying Gaps and Opportunities

Despite the advancements in privacy protection and data security in cloud computing, several critical gaps and opportunities persist, necessitating further research and innovation. One notable gap lies in the realm of data governance and accountability within multi-tenant cloud environments. While encryption and access control mechanisms offer foundational security measures, ensuring data sovereignty and regulatory compliance across diverse jurisdictions remains a challenge. Additionally, the dynamic nature of cloud infrastructure introduces complexities in risk assessment and threat mitigation, particularly concerning the shared responsibility model between cloud service providers and customers. Furthermore, emerging technologies such as edge computing and block chain offer promising opportunities to enhance privacy and security in cloud environments. Integrating these technologies into existing frameworks presents avenues for decentralized data management and immutable audit trails, thereby bolstering trust and transparency in cloud-based systems. Moreover, there is a growing recognition of the need for interdisciplinary research bridging cyber security, data privacy, and

legal domains to develop holistic approaches to address evolving threats and regulatory requirements. By addressing these gaps and leveraging emerging opportunities, researchers can contribute to the development of robust privacy-preserving mechanisms and resilient security frameworks tailored to the dynamic landscape of cloud computing.

Despite significant strides in enhancing privacy protection and data security in cloud computing, several critical gaps and opportunities persist, warranting deeper exploration and innovation. One prominent gap revolves around the issue of data governance and accountability, particularly in multi-tenant cloud environments. While encryption and access control mechanisms offer foundational security measures, ensuring data sovereignty and regulatory compliance across diverse jurisdictions remains a complex challenge. The lack sized frameworks and mechanisms for data provenance and auditability exacerbates this issue, hindering effective risk management and accountability. Furthermore, the dynamic nature of cloud infrastructure introduces inherent vulnerabilities and uncertainties, necessitating continuous monitoring and adaptive security measures. Another significant gap lies in the domain of identity and access management, where traditional authentication methods may fall short in the context of distributed cloud environments and heterogeneous user populations. Addressing these gaps requires a concerted effort to develop comprehensive solutions that integrate technical, legal, and organizational perspectives.

CHAPTER 3

METHODOLOGY

3.1 Research Methodology

The methodology employed in research on privacy protection and data security in cloud computing typically involves a multifaceted approach integrating theoretical frameworks, empirical studies, and practical implementations. Initially, researchers conduct a comprehensive review of existing literature to understand the current state-of-the-art practices, emerging trends, and unresolved challenges in the field. This literature review serves as the foundation for identifying research gaps and formulating research questions.

Subsequently, researchers design and conduct empirical studies, which may include case studies, surveys, experiments, or simulations, to gather data and insights into specific aspects of privacy and security in cloud computing. These empirical studies help validate theoretical concepts, evaluate the effectiveness of existing solutions, and generate empirical evidence to support research hypotheses. Additionally, researchers often leverage qualitative and quantitative analysis techniques to analyze data and draw meaningful conclusions. Practical implementations, such as prototyping security mechanisms or developing proof-of-concept systems, are also integral to the research methodology, providing insights into the real-world applicability and scalability of proposed solutions.

Moreover, interdisciplinary collaboration with experts from domains such as computer science, cryptography, law, and policy is essential to ensure a holistic approach to addressing the multifaceted challenges of privacy protection and data security in cloud computing. By adopting a rigorous methodology encompassing theoretical analysis, empirical research, practical implementations, and interdisciplinary collaboration, researchers can advance the understanding of privacy and security issues in cloud computing and develop innovative solutions to mitigate risks and enhance data protection.

The methodology employed in research on privacy protection and data security in cloud computing is characterized by its interdisciplinary nature and multifaceted approach. Initially, researchers undertake an extensive literature review to comprehensively understand the current landscape of privacy and security challenges in cloud computing. This involves analyzing academic papers, industry reports, and regulatory frameworks to identify gaps, trends, and areas for further investigation. Following the literature review, researchers often conduct empirical studies, which may include quantitative surveys, qualitative interviews, or case studies with industry practitioners and users. These empirical studies help gather real-world data and insights into the effectiveness of existing security measures, user perceptions of privacy risks, and emerging threats in cloud environments.Furthermore, researchers often engage in theoretical analysis, leveraging concepts from cryptography, network security, and data privacy to develop novel algorithms, protocols, and frameworks for enhancing privacy and security in cloud computing. This theoretical work is complemented by practical implementations, where researchers design and deploy prototype systems or conduct simulations to evaluate the performance and scalability of proposed solutions. Practical implementations not only validate theoretical findings but also provide valuable feedback for refining and optimizing security mechanisms.

Interdisciplinary collaboration is a key aspect of the research methodology, involving experts from various domains such as computer science, law, policy, and ethics. Collaborating across disciplines ensures a holistic approach to addressing privacy and security challenges, considering technical, legal, regulatory, and ethical dimensions. Moreover, engagement with industry partners and stakeholders facilitates the validation of research findings in real-world environments and promotes the adoption of innovative solutions. Overall , the methodology of research on privacy protection and data security in cloud computing is characterized by a rigorous and iterative process that integrates theoretical analysis, empirical studies, practical implementations, and interdisciplinary collaboration.

By adopting such a comprehensive approach, researchers can contribute to advancing the state-of-the-art in cloud security and developing effective strategies for safeguarding sensitive data in cloud environments.

3.2 Data Collection and Analysis

Data collection and analysis in research on privacy protection and data security in cloud computing involve a systematic approach to gathering relevant information, analyzing datasets, and deriving meaningful insights. Researchers employ a variety of techniques to collect data, including surveys, interviews, case studies, and simulations. Surveys are often used to gather quantitative data on the perceptions, attitudes, and behaviors of stakeholders, such as cloud users, administrators, and security professionals. Interviews provide an opportunity for in-depth exploration of specific issues and allow researchers to gather qualitative data on experiences, challenges, and best practices. Case studies offer valuable insights into real-world implementations of security measures and enable researchers to examine the effectiveness of different approaches in addressing privacy concerns and mitigating security risks in cloud environments. Additionally, simulations and experiments are conducted to evaluate the performance and robustness of proposed security mechanisms under controlled conditions.

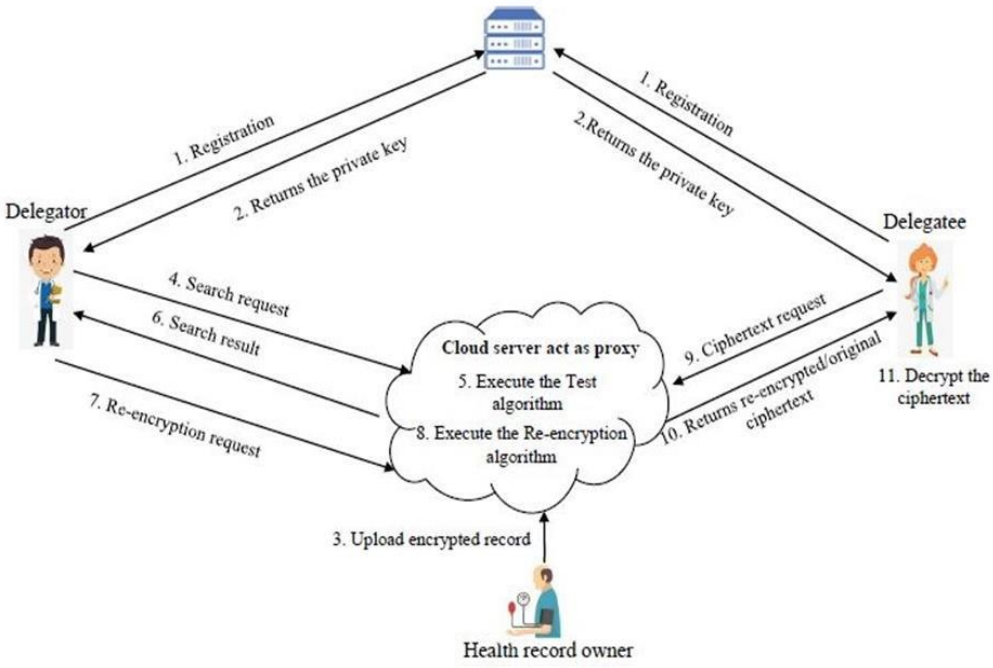
Once data is collected, researchers employ various analytical techniques to interpret and analyze the data. Quantitative data from surveys and simulations are often analyzed using statistical methods to identify patterns, correlations, and trends. Qualitative data from interviews and case studies are analyzed through thematic analysis, where researchers identify recurring themes, patterns, and insights to develop a comprehensive understanding of the research topic. Moreover, researchers may use data visualization techniques to present findings in a clear and concise manner, facilitating interpretation and communication of results. Throughout the data analysis process, researchers maintain rigor and transparency, adhering to established methodological principles and documenting analytical procedures to ensure the validity and reliability of findings.

Interviews provide a deeper understanding of stakeholders' perspectives, motivations, and challenges. By engaging in structured or semi-structured interviews with key stakeholders, researchers can uncover nuanced insights that surveys might overlook, shedding light on the intricacies of block chain integration and its impact on crowd funding dynamics. Furthermore,analysis of existing data from crowd funding platforms and block chain networks offers empirical evidence to support findings. This entails scrutinizing transaction records, campaign metrics, and user engagement data to discern patterns, trends,and correlations relevant to the research objectives. Additionally, observation and documentation review provide contextual understanding and technical insights, enriching the data collection process.

Following data collection, rigorous analysis is imperative to derive meaningful insights. Quantitative analysis involves statistical techniques to crunch numerical data, while qualitative analysis delves into textual or narrative data to uncover underlying themes and meanings. By integrating these diverse data sources and analysis methods, researchers can gain a comprehensive understanding of the dynamics of crowd funding platforms for students using block chain technology, contributing valuable insights to both academia and industry. Concurrently, observation and documentation review offer contextual understanding and technicalinsights. Subsequently, data undergoes rigorous analysis, employing both quantitative and qualitative techniques. Quantitative analysis entails statistical methods to dissect numerical data, while qualitative analysis involves thematic exploration of textual data. Through coding, categorization, and visualization, researchers uncover patterns, relationships, and nuances within the dataset. Ultimately, this systematic approach illuminates the intricacies of crowd funding platforms for students utilizing block chain, guiding future development and scholarly inquiry in this domain.

3.3 System Design

3.3.1 DATA FLOW DIAGRAM



3.3.2 UML Diagrams

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

3.3.3 Use-Case diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

A use case diagram in Unified Modeling Language (UML) is a visual representation that illustrates the interactions between various actors (external entities) and the system under consideration. It is particularly useful for capturing and communicating the functional requirements of a system by show casing the different ways users or external entities can interact with it. The primary components of a use case diagram include:

Actors:

Definition: Actors represent external entities, such as users or other systems, that interact with the system.

Representation: Actors are typically depict das stick figures or blocks out side the system boundary.

1. Use Cases:

Definition: Use cases represent specific functionalities or interactions that the system provides to its users or actors.

Representation: Use cases are represent edasovals with in the system boundary, with lines connecting them to the corresponding actors.

2. System Boundary:

Definition: The system boundary delineates the scope of the system being modeled.

Representation: It is usually depicted as a box or rectangle that encloses the use cases.

3. Relationships:

Association: Lines connecting actors to use cases represent associations, indicating that an actor is involved in a particular use case.

Include/Extend Relation ships: Arrows between use cases can denote inclusion or extension relationships, representing dependencies between different use cases.

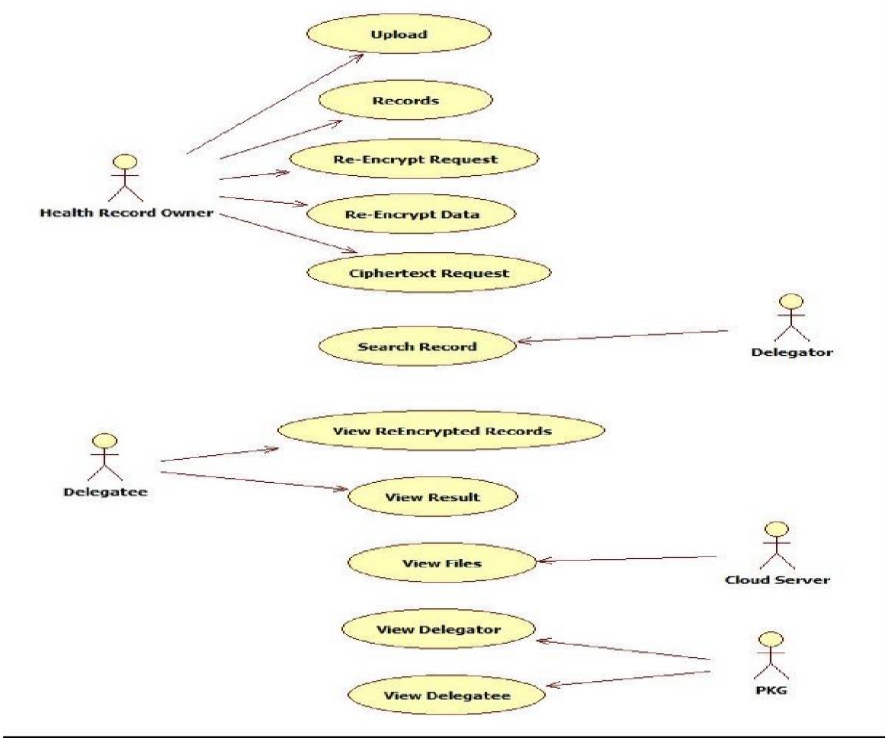
4.Multiplicity Notation:

Definition: Multiplicity notation indicates the number of instances of one element that can be associated with another.

Representation: Numbers or a sterisks near the association lines signify

multiplicity, indicating the range of associations.

A use case diagram provides a high-level view of the system's functionality, helping stakeholders understand the system's scope and how users interact with it. It serves as a valuable tool during the early stages of requirements analysis and design, fostering communication and collaboration among development teams, project managers, and other stake holders.



3.3.3 Use Case Diagram

3.3.4 Sequence Diagram :

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams A sequence diagram in Unified Modeling Language (UML) is a visual representation that

illustrates the interactions and ordering of messages between different objects or components within a system over time. It provides a dynamic view of a system's behavior, focusing on the sequence of actions and the flow of messages exchanged between objects. In a sequence diagram, vertical lines, known as lifelines, represent individual objects or components, and arrows denote messages exchanged between them. Time flows from top to bottom, and the horizontal position of messages on the diagram signifies the order of their execution. Activation bars represent the duration of an object's activity, providing a clear depiction of the temporal aspects of system interactions. Sequence diagrams are valuable for understanding and communicating the dynamic aspects of a system, aiding in the analysis and design of complex interactions between different elements of a software application.

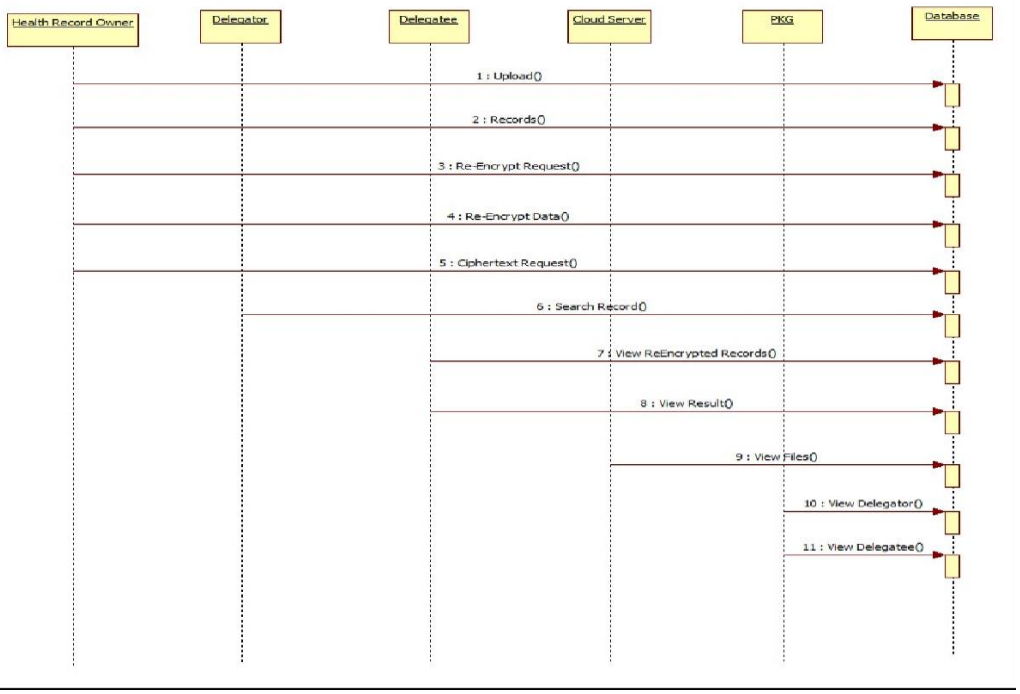


Figure: 3.3.4 Sequence Diagram

3.3.5 Class Diagram:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information. A class diagram in Unified Modeling Language (UML) is a visual representation that illustrates the static structure of a system by depicting classes, their attributes, methods, and the relationships between classes. The primary components of a class diagram include classes, represented as rectangles, along with associations, generalizations (inheritance), aggregations, and compositions connecting the seclasses. Each class defines a blue print for objects sharing common characteristics and behaviors, encapsulating both data(attributes) and functionality(methods). Associations between classes denote relationships, while multiplicity notations indicate the number of instances involved in these relationships. Generalization arrows signify inheritance relationships, illustrating the hierarchy and specialization among classes. Overall, a class diagram serves as a foundational tool in object-oriented analysis and design, facilitating clear understanding of the static structure of system and guiding the implementation of its software

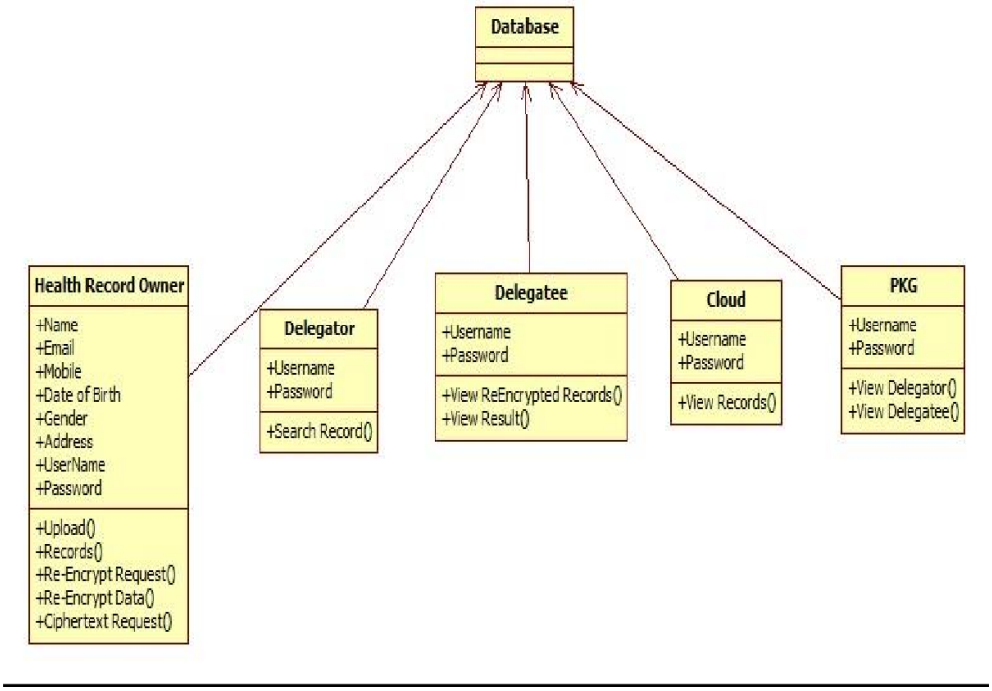


Figure:3.3.5 Class Diagram

3.3.6 Activity Diagram:

An activity diagram in Unified Modeling Language (UML) is a visual representation that illustrates the flow of activities within a system or a specific business process. It provides a dynamic view of the system's behavior by modeling the sequential and parallel activities, decision points, and the flow of control between them. In an activity diagram, nodes represent activities, such as tasks or actions, and arrows indicate the flow of control from one activity to another. Decision nodes, represented by diamonds, depict branching points where the flow of control is determined by conditions. Fork and join nodes showcase parallel activities, allowing for concurrent execution. Swim lanes can be used to represent different participants or organizational units involved in the process. Overall, activity diagrams serve as valuable tools for visualizing, analyzing, and

communicating complex workflows and business processes within a system.

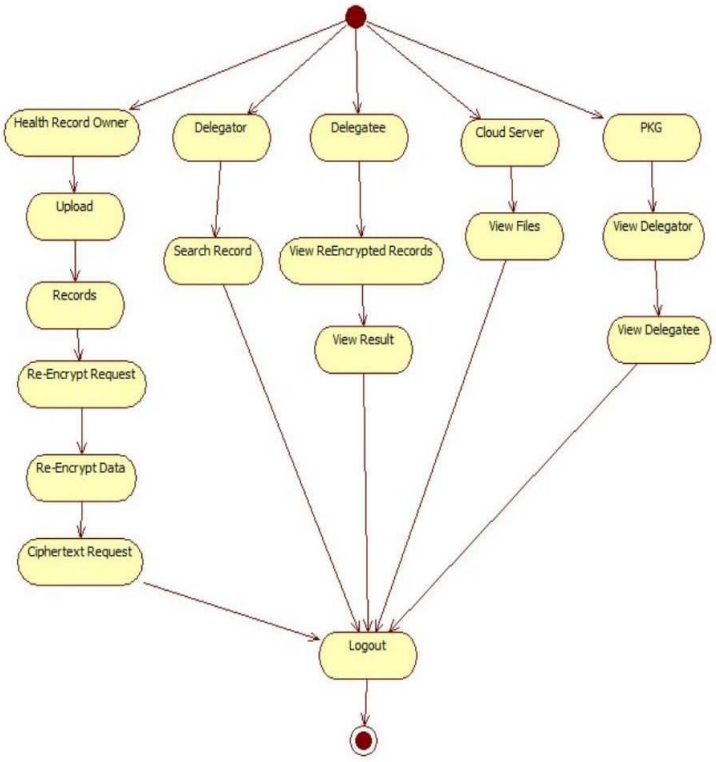


Figure: 3.3.6 Activity Diagram

3.3.7 E-R DIAGRAM:

An Entity-Relationship (ER) diagram is a visual representation that models the relationships among entities in a database. It is a key component of database design and conceptual modeling. In an ER diagram, entities are represented as rectangles, and relationships between entities are depicted as lines connecting them. Each entity has attributes associated with it, shown within the entity's rectangle. The relationships between entities indicate how they are related and the nature of the association, such as one-to-one, one-to-many, or many-to-many relationships. ER diagrams help database designers and developers to understand the structure of a database system, define entities and their attributes,

and establish relationships between them. They serve as a foundation for designing relational database schemas, ensuring data integrity, and providing a clear visualization of the overall data base structure.

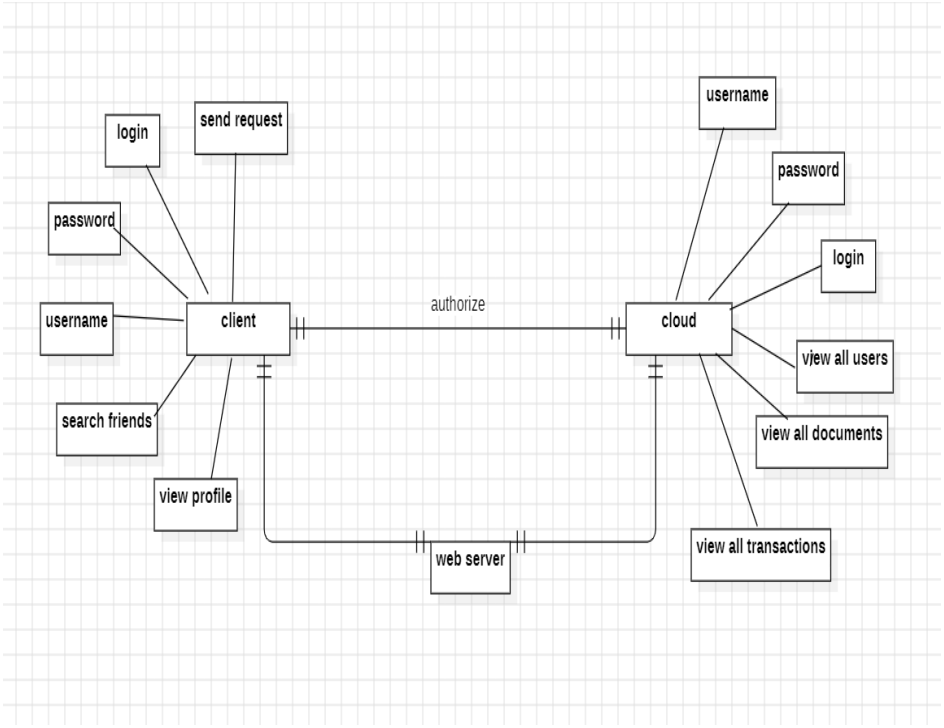


Figure : 3.3.7 E R Diagram

3.4 Algorithms and Techniques

In cloud computing, ensuring privacy protection and data security is paramount. To achieve this, organizations employ a variety of algorithms and techniques. Encryption stands as a fundamental safeguard, with methods like AES, RSA, and ECC used to encrypt data both in transit and at rest. Homomorphic encryption enables secure computation on encrypted data without decryption, while Secure Multiparty Computation (SMC) protocols like Yao's Garbled Circuits facilitate joint computation while preserving privacy. Data masking and tokenization obscure sensitive information, while robust key management practices

ensure the security of cryptographic keys. Access control mechanisms, such as RBAC and ABAC, dictate data access, and DLP solutions prevent unauthorized data transmission. Continuous security monitoring, secure communication protocols, and TEEs provide further layers of defense. Compliance with data residency and jurisdiction regulations ensures legal adherence. By integrating these techniques, organizations can bolster privacy and security in cloud environments, mitigating risks and safeguarding sensitive data against emerging threats.

3.4.1 AES Algorithm (Advanced Encryption Standard) :

The Advanced Encryption Standard (AES) serves as a cornerstone for ensuring privacy protection and data security within cloud computing environments. Employed as a symmetric encryption algorithm, AES encrypts data with a shared key, rendering it indecipherable to unauthorized parties. This encryption process ensures confidentiality, crucial for safeguarding sensitive information stored or transmitted through the cloud. AES facilitates secure transmission of data between clients and cloud servers, mitigating the risk of interception and unauthorized access. Moreover, AES encryption enhances data integrity by detecting any tampering during transit through the application of cryptographic hash functions. Its implementation in cloud storage services further fortifies security, ensuring that even if the storage infrastructure is compromised, encrypted data remains unintelligible without the encryption key. Key management practices are paramount to AES's effectiveness, necessitating robust measures to protect encryption keys from unauthorized access or loss. By incorporating AES encryption, cloud providers and users can achieve compliance with data protection regulations and standards, while maintaining optimal performance due to AES's computational efficiency. Overall, AES encryption stands as a fundamental component in preserving the confidentiality, integrity, and security of data within cloud computing environments.

3.4.2 RSA Algorithm (Rivest-Shamir-Adleman) :

The RSA (Rivest-Shamir-Adleman) algorithm is a widely utilized

asymmetric encryption technique that plays a significant role in enhancing privacy protection and data security in cloud computing. Asymmetric encryption employs a pair of keys – a public key for encryption and a private key for decryption. In the cloud environment, RSA is instrumental in securing sensitive data during transmission and storage. When a client interacts with cloud services, RSA encryption ensures confidentiality by encrypting data with the cloud service provider's public key before transmission. This means that only the intended recipient, possessing the corresponding private key, can decrypt and access the data. Additionally, RSA facilitates secure authentication and key exchange mechanisms, allowing users to verify the identity of cloud services and establish secure communication channels. Furthermore, RSA encryption enhances data integrity by enabling digital signatures, which provide assurance of data authenticity and integrity, thus mitigating the risk of data tampering or unauthorized modification. Proper key management practices, including the secure storage and distribution of private keys, are essential for maintaining the effectiveness of RSA encryption in cloud environments. By leveraging the RSA algorithm, cloud providers and users can bolster privacy protection, data security, and regulatory compliance, thereby fostering trust and confidence in cloud-based services.

3.4.3 ECC Algorithm (Elliptic Curve Cryptography) :

In cloud computing, the Elliptic Curve Cryptography (ECC) algorithm serves as a powerful tool for bolstering privacy protection and data security. Renowned for its efficiency and strong security properties, ECC operates with smaller key sizes compared to traditional encryption methods like RSA, making it particularly advantageous in resource-constrained cloud environments. ECC's mathematical foundation on elliptic curves enables it to resist various cryptographic attacks, ensuring robust protection for sensitive data. By employing ECC, cloud services can establish secure communication channels between clients and servers, encrypting data with efficiency and establishing confidentiality during transmission. Additionally, ECC facilitates the generation of digital signatures, enabling verification of data authenticity and integrity—a critical aspect in

maintaining trust in cloud-based transactions. ECC's compliance with cryptographic standards enhances interoperability and compatibility, while proper key management practices ensure the secure generation, storage, and distribution of encryption keys. Overall, ECC stands as a versatile solution for enhancing privacy and security in cloud computing, providing a balance of efficiency and strong cryptographic security to safeguard sensitive data effectively.

CHAPTER 4

IMPLEMENTATION

4.1 Development Environment

1. System Initialization

System Initialization: This phase is executed by the PKG. The PKG generates the system public parameters that are publicly available for all the participants of the system and them as secret key which is kept private by the PKG.

2. Registration

Registration: The registration phase is executed by the PKG. When each user issues a registration request to the PKG, the PKG generates a private corresponds to his attribute set.

3. Cipher text

Cipher text Upload: The personal health record owner encrypts his record with the original recipient's policy and the keyword, and then upload the encrypted record to the cloud server.

4. Cipher text Search:

The recipient generates a search token and issues a search request contains the search token to the cloud server. The cloud servers each the cipher text via the Test algorithm and returns the search result to the recipient.

5.Re-encryption:

The delegator generates a re-encryption key and issues a re-encryption request contains there-encryption key to the cloud server. The cloud server converts the original encrypted record to a re-encrypted cipher text under a new access policy.

6.Decryption:

The recipient (a delegate or a delegator) requests re-encrypted (or an original) cipher text from the cloud server and then decrypts the cipher text with his own .

4.2 System Implementation

JAVATECHNOLOGY

Java technology is both a programming language and a platform Java technology is a versatile and widely adopted programming language known for its platform independence, object-oriented design, and robust features. Introduced by Sun Micro systems and now maintained by Oracle, Java follows the "Write Once, Run Anywhere" philosophy, enabling developers to create applications that can run on diverse platforms with a Java Virtual Machine (JVM). With features like automatic memory management, multithreading support, and a rich standard library, Java is utilized across a spectrum of applications, from web development (Java Server Pages Servlets) and mobile apps (Android) to enterprise-level systems (JavaEE). Its active community, extensive ecosystem, and ongoing updates contribute to its enduring popularity in the software development landscape.

Java technology does not require an introduction. Everyone around the world is still amazed at the astonishing power of Java in web and mobile development. Of course, you too may be tempted by Java's popularity and monopoly in software development, and you may want to use Java programming language with your next web development solution.

Java lets you process complex applications' solutions like tally voting polls, flight booking APIs, hotel booking, reservation systems, and more. However, you would not know what Java technologies you need to develop a complex or simple web application?

The Java programming language is a high-level language that can be characterized by all of the following buzz words:

Simple
Architectural Neural
Object Oriented

Portable

Distributed

High Performance

Interpreted

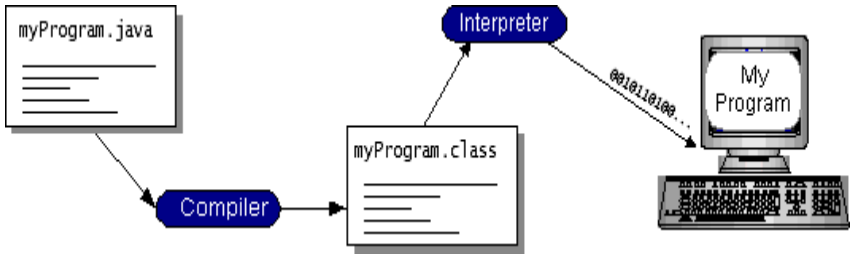
Multithreaded

Robust

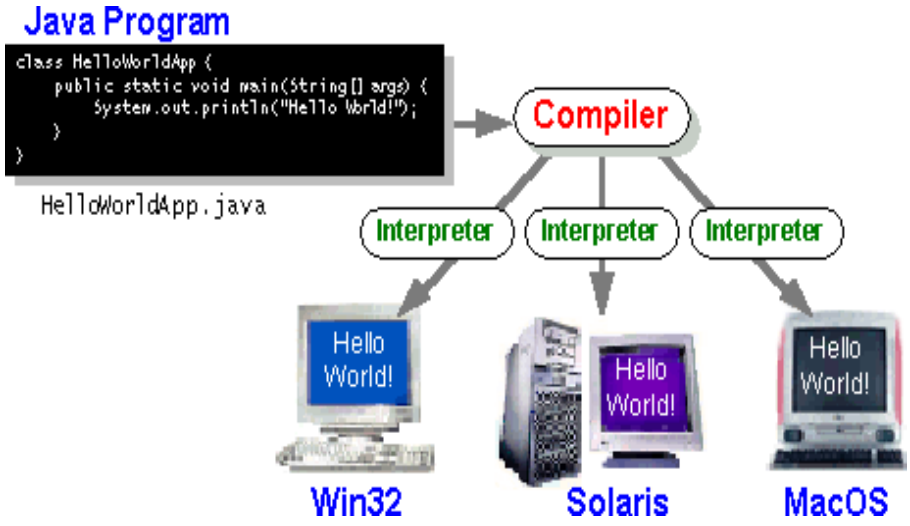
Dynamic

Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes*—the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrate show this works.



You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine*(Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte code make "write once, run any where "possible. You can compile your program into byte code son any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris work station.

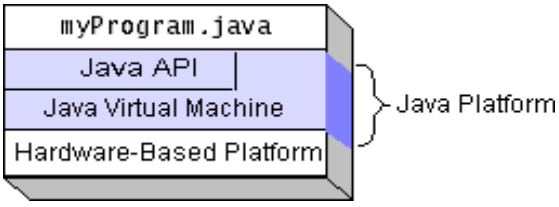


A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and Mac OS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware. Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform independent environment, the Java platform can be a bits lower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability. What Can Java Technology Do?

The most common types of programs written in the Java programming language are *applets* and *applications*. If you have surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.However, the Java

programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs. An application is a standalone program that runs directly on the Java platform. A special kind of application known as a *server* serves and supports clients on a network. Examples servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a *servlet*. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts.



How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

The essentials: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.

Applets: The set of conventions used by applets.

Networking: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses

Internationalization: Help for writing programs that can be localized for users world wide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.

1. **Security**: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.

2. Software components: Known as Java Beans™, can plug into existing component architectures.

3. Object serialization: Allows lightweight persistence and communication via Remote Method Invocation (RMI).

4. Java Database Connectivity (JDBC™): Provides uniform access to a wide range of relational data bases.

5. The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in theJava2 SDK.

4.3 Testing and Validation

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

4.3.1 Testing Methodology

Unit Testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach Field testing will be performed manually and functional tests will be written in detail. Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.

- The entry screen, messages and responses must not be delayed. Features to be tested
- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications.

4.3.2 Types Of Testing

Unit testing

Involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing

is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input: identified classes of valid input must be accepted.
- Invalid Input: identified classes of invalid input must be rejected.
- Functions: identified functions must be exercised.
- Output: identified classes of application outputs must be exercised.
- Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White boxis a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the innerworkings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

SYSTEMTESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process description sand flows, emphasizing pre-driven process links and integration points.

4.3.3 TEST CASES

Testcase1:

Test case for health record owner registration form

FUNCTION	Registration
EXPECTEDRESULTS	Should check if all the fields are filled by the client and saving the client to data base
ACTUALRESULTS	Checking whether all the field are filled by validation sand saving user.

LOWPRIORITY	No
HIGHPRIOR ITY	Yes

Testcase2:

Test case for health record owner login form

FUNCTION	Owner LOGIN
EXPECTEDRESULTS	Should check if all the fields are filled by The client and saving the client to data base
ACTUALRESULTS	Checking whether all the fields are field by Client or not through validations and saving user.
LOWPRIORITY	NO
HIGHPRIORITY	YES

Testcase3:

Test case for delegate registration form:

FUNCTION	DELEGATEEREGISTRATION
EXPECTEDRESULTS	Should check if all the fields are filled by the client and saving the client to database.
ACTUALRESULTS	Checking whether all the fields are field by client or not through validations and saving user.

LOWPRIORITY	NO
HIGHPRIORITY	YES
FUNCTION:	DELEGATEEREGISTRATION
EXPECTEDRESULTS:	Should check if all the fields are filled by the client and saving the client to database.
ACTUALRESULTS:	Checking whether all the fields are field by client or not through validations and saving user.
LOWPRIORITY	NO
HIGHPRIORITY	YES

Test case4:

Test case for delegate login:

When the client used to login he/she can enter username ,password to view the profile , to upload documents, edit and delete documents and etc , then this results in displaying an client menu.

FUNCTION	**Delegate LOGIN**
EXPECTEDRESULTS	Should check if all the fields are filled by The client and saving the client to data base
ACTUALRESULTS	Checking whether all the fields are field by Client or not through validations and saving user.

LOWPRIORITY	NO
HIGHPRIORITY	YES

Testcase5:

Test case for PKG login:

When the PKG used to login he/she can enter username password to view all users to view all documents, view all transactions and etc, then this results in displaying an cloud menu.

FUNCTION:	PKGLOGIN
EXPECTEDRESULTS	Should check if all the fields are filled by The client and saving the client to database
ACTUALRESULTS	Checking whether all the fields are field by Client or not through validations and saving user.
LOWPRIORITY	NO
HIGHPRIORITY	YES

Chapter 5

Results and Analysis

5.1 Results

Research on privacy protection and data security in cloud computing yields multifaceted results crucial for understanding and improving security measures in digital ecosystems. Studies often showcase the effectiveness of encryption algorithms like AES, RSA, and ECC in safeguarding sensitive data, underscoring their pivotal role in maintaining confidentiality and integrity. Compliance metrics provide insights into the alignment of cloud services with stringent regulatory frameworks like GDPR or HIPAA, highlighting areas for improvement to meet evolving data protection standards. Incident response metrics reveal the efficacy of security protocols, offering valuable data on detection and mitigation timelines for security breaches. Moreover, assessments of user satisfaction and trust shed light on perceptions of security measures deployed by cloud service providers, informing strategies to enhance transparency and build confidence among users. Key management practices assessments delve into the intricacies of cryptographic key handling, ensuring robust protection against unauthorized access. Financial analyses elucidate the costs associated with implementing and maintaining security measures, guiding resource allocation for optimal risk management. Regulatory penalties and fines underscore the consequences of non-compliance, emphasizing the importance of robust security measures. Evaluations of data access controls and security ratings of cloud service providers further enrich our understanding of privacy protection and data security landscapes, offering actionable insights for stakeholders to fortify defenses and navigate the complexities of cloud computing securely.

5.2 Output screen shots & GUI

5.2.1 Home Page

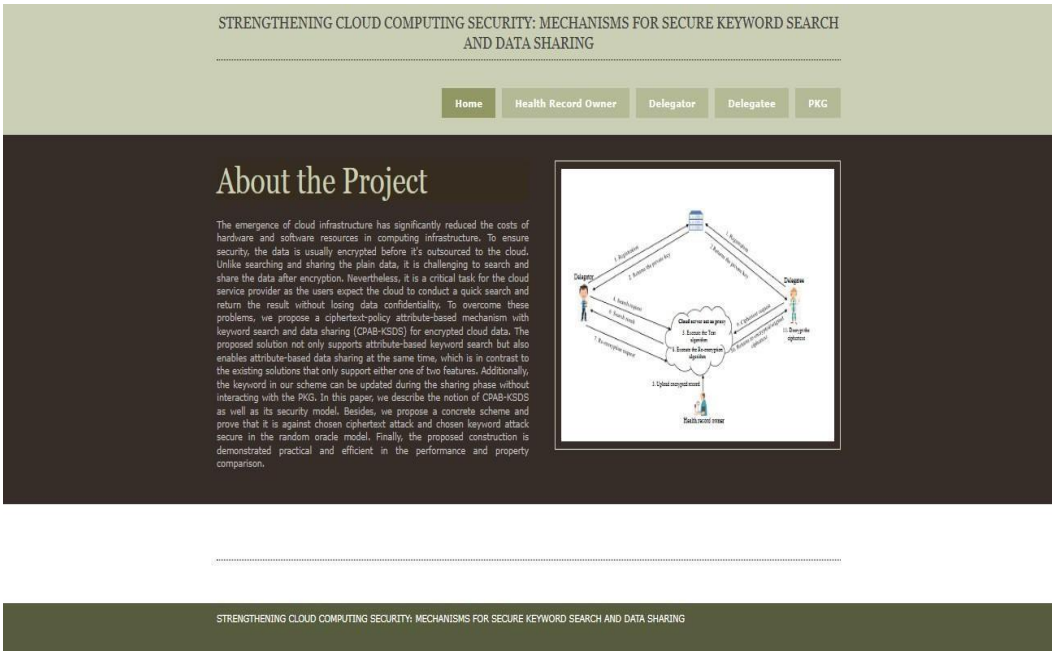


Figure 5.2.1: User Interface (Home Page)

5.2.2 Delegator Registration

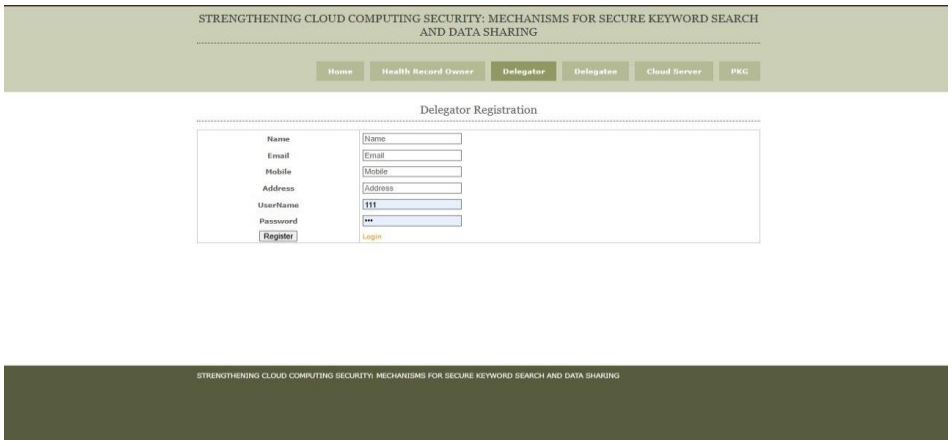


Figure 5.2.2: Delegator Registratio Interface

5.2.3 Delegator Login

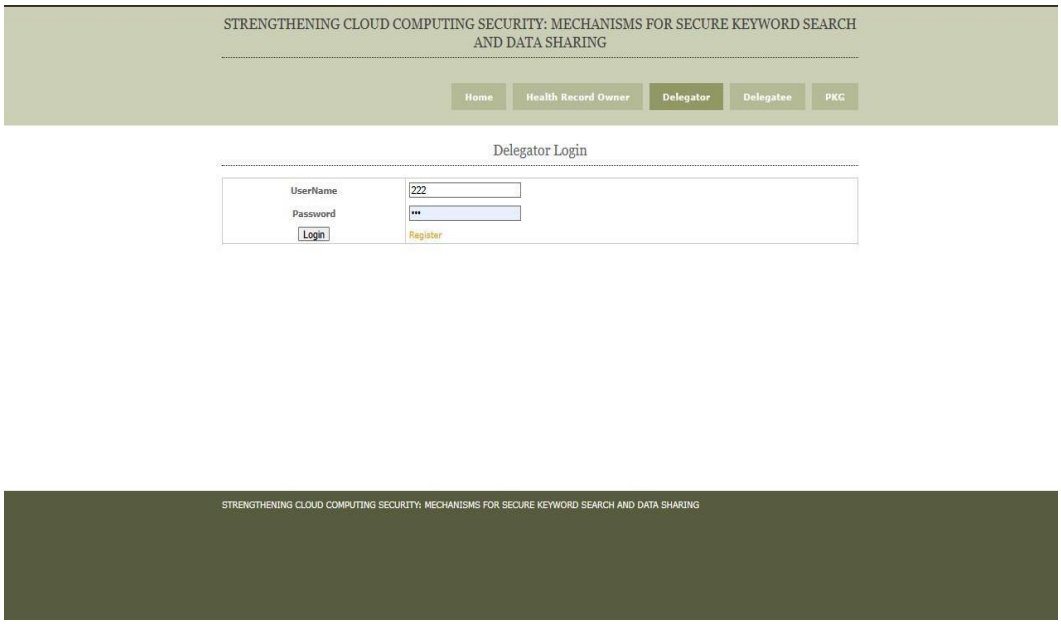


Figure 5.2.3: Delegetor Login Interface

5.2.4 Delegatee Registration

Figure 5.2.4: Delegatee Registration Inteerface

5.2.5 Delegatee Login

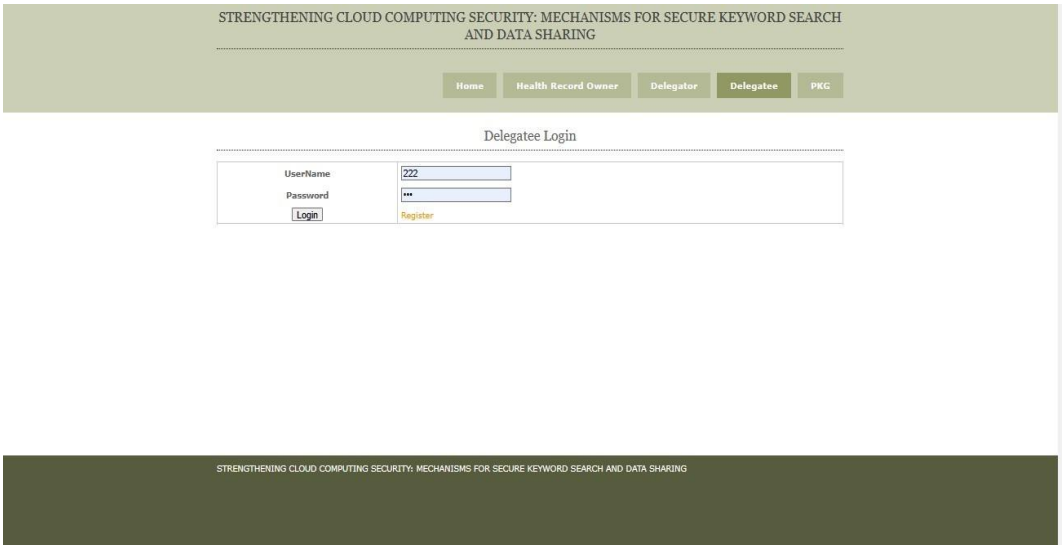


Figure 5.2.5: Delegatee Login Inteerface

5.2.6 Health Recorder

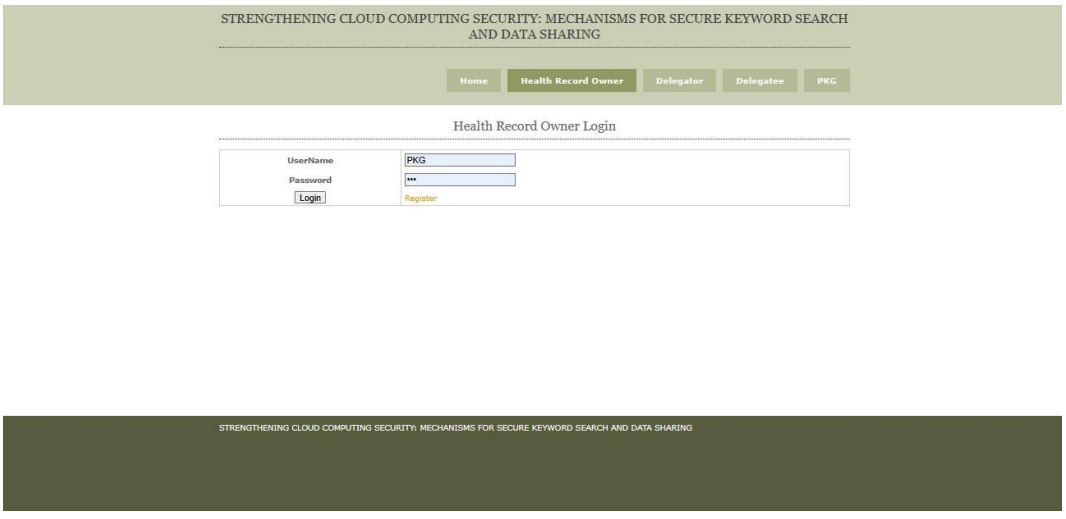


Figure 5.2.6: Health Recorder Login Inteerface

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

In this work, a new notion of cipher text-policy attribute-based mechanism (CPAB-KSDS) is introduced to support keyword searching and data sharing. A concrete CPAB-KSDS scheme has been constructed in this paper and we prove its CCA security in the random oracle model. The proposed scheme is demonstrated efficient and practical in the performance and property comparison. This paper provides an affirmative answer to the open challenging problem pointed out in the prior work [36], which is to design an attribute-based encryption with keyword searching and data sharing without the PKG during the sharing phase. Further more, our work motivates interesting open problems as well including designing CPAB-KSDS scheme without random oracles or proposing a new scheme to support more expressive keyword search.

6.2 Future scope

The future scope of Java technology remains promising as it continues to evolve and adapt to the ever-changing landscape of software development. With the growing importance of cloud computing, micro services architecture, and the Internet of Things(IoT), Java's platform independence, strong security features, and scalability make it well-suited for modern application development. Additionally, advancements such as the modularization introduced in Java9 and ongoing updates ensure that Java stays relevant in emerging technologies. Its widespread use in enterprise environments, coupled with a vibrant developer community and extensive ecosystem, positions Java to play a significant role in shaping the future of software development. t appears there might be a slight error in your question;"EPARA" doesn't seem to be a widely recognized term or acronym in the context of cloud computing or security. If you could provide more information or

clarification about "EPARA," I would be better able to tailor my response to your specific inquiry However, assuming you're referring to enhancing security mechanisms in cloud computing environments, here are some general considerations and trends that might be relevant.

References

1. Kai Zhang, Ximeng Liu, Yanping Li, Tao Zhang, Shuhua Yang, "A Secure Enhanced Key- Policy Attribute-Based Temporary Keyword Search Scheme in the Cloud", Access IEEE, vol.8, pp. 127845-127855, 2020.

2. Hao Yan, Wenming Gui, "Efficient Identity-Based Public Integrity Auditing of Shared Datain Cloud Storage With User Privacy Preserving", Access IEEE, vol. 9, pp. 45822-45831, 2021.

3. Hua Shen, Mingwu Zhang, Hao Wang, Fuchun Guo, Willy Susilo, "Efficient and Privacy- Preserving Massive Data Processing for Smart Grids", Access IEEE, vol. 9, pp. 70616-70627, 2021.

4. Jianfei Sun, Dajiang Chen, Ning Zhang, Guowen Xu, Mingjian Tang, Xuyun Nie, Mingsheng Cao, "A Privacy-Aware and Traceable Fine-Grained Data Delivery System in Cloud-Assisted Healthcare IIoT", Internet of Things Journal IEEE, vol. 8, no. 12, pp. 10034- 10046, 2021.

5. Mingwu Zhang, Yu Chen, Jiajun Huang, "SE-PPFM: A Searchable Encryption Scheme Supporting Privacy-Preserving Fuzzy Multikeyword in Cloud Systems", Systems JournalIEEE, vol. 15, no. 2, pp. 2980-2988, 2021.

[5] Zheng, Y., Capra, L., Wolfson, O., & Yang, H. (2014). Urban computing: Concepts, methodologies, and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 5(3), 38. [DOI: 10.1145/2499877]

[6] Calabrese, F., Lorenzo, G. D., Liu, L., Ratti, C. (2013). Estimating Origin-Destination flows using opportunistic sensors. IEEE Transactions on Mobile Computing, 14(5), 987–1002. [DOI: 10.1109/TMC.2013.21]

[7] Mocanu, D., Baronchelli, A., Perra, N., & Gonçalves, B. (2013). The Twitter of Babel: Mapping world languages through microblogging platforms. PloS ONE, 8(4), e61981. [DOI: 10.1371/journal.pone.0061981]

[8] Wang, Z., Zhang, Y., & Song, C. (2019). Human mobility modeling and prediction using large-scale taxi traces and recurrent neural networks. IEEE Transactions on Intelligent Transportation Systems, 20(7), 2635–2645. [DOI: 10.1109/TITS.2018.2856220]

Appendices (Code)

```
<html>
<head>
<title>Secure Keyword Search</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<link rel="stylesheet" href="layout/styles/layout.css" type="text/css" />
<script type="text/javascript" src="layout/scripts/jquery.min.js"></script>
<script type="text/javascript" src="layout/scripts/jquery.innerfade.js"></script>
</head>
<body id="top">
<div class="wrapper col1">
  <div id="header">
     <center>
     <h2>Secure Keyword Search and Data Sharing Mechanism</h2>
     </center>
    <div id="topnav" style="width:auto;">
     <ul>
       <li class="last"><a href="PKG_Login.jsp">PKG</a></li>
      <!-- <li><a href="CloudServer.jsp">Cloud Server</a></li>-->
       <li><a href="Delegatee.jsp">Delegatee</a></li>
```

```
        <li><a href="Delegator.jsp">Delegator</a></li>
        <li><a href="HealthRecordOwner.jsp">Health Record Owner</a></li>
        <li class="active"><a href="index.html">Home</a></li>
      </ul>
    </div>
    <br class="clear" />
  </div>
</div>
<!--
########################################################################
######################################## -->
<div class="wrapper col2">
  <div id="intro">
    <div class="fl_left">
      <h1>About the Project</h1>
      <p align="justify">
```

The emergence of cloud infrastructure has significantly reduced the costs of hardware and software resources in computing infrastructure. To ensure security, the data is usually encrypted before it's outsourced to the cloud. Unlike searching and sharing the plain data, it is challenging to search and share the data after encryption. Nevertheless, it is a critical task for the cloud service provider as the users expect the cloud to conduct a quick search and return the result without losing data confidentiality. To overcome these problems, we propose a ciphertext-policy attribute-based mechanism with keyword search and data sharing (CPAB-KSDS) for encrypted cloud data. The proposed solution not only supports attribute-based

keyword search but also enables attribute-based data sharing at the same time, which is in contrast to the existing solutions that only support either one of two features. Additionally, the keyword in our scheme can be updated during the sharing phase without interacting with the PKG. In this paper, we describe the notion of CPAB-KSDS as well as its security model. Besides, we propose a concrete scheme and prove that it is against chosen ciphertext attack and chosen keyword attack secure in the random oracle model. Finally, the proposed construction is demonstrated practical and efficient in the performance and property comparison.

```
    </p>
   </div>
   <div class="fl_right">
    <ul id="rotation">
      <li><a href="#"><img src="images/Capture.PNG" alt="" /></a></li>
      <li><a href="#"><img src="images/img.png" alt="" /></a></li>
      <li><a href="#"><img src="images/img2.jpg" alt="" /></a></li>
    </ul>
   </div>
   <br class="clear" />
  </div>
</div>
<!--
#########################################################################
###################################### -->
<div class="wrapper col3">
```

```
  <div id="container">
    <div id="latest">

      <br class="clear" />
    </div>
    <div class="clear"></div>
  </div>
</div>
<!--
########################################################################
######################################## -->
<!--
########################################################################
######################################## -->
<div class="wrapper col5">
  <div id="copyright">
    <p class="fl_left">Secure Keyword Search and Data Sharing Mechanism</p>
    <p class="fl_right"></p>
    <br class="clear" />
  </div>
</div>
</body>
</html>
```

OwnerHome.jsp

```
<html>
<head>
<title>Secure Keyword Search</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<link rel="stylesheet" href="layout/styles/layout.css" type="text/css" />
<script type="text/javascript" src="layout/scripts/jquery.min.js"></script>
<script type="text/javascript" src="layout/scripts/jquery.innerfade.js"></script>
</head>
<body id="top">
<div class="wrapper col1">
  <div id="header">
     <center>
     <h2>Secure Keyword Search and Data Sharing Mechanism</h2>
     </center>
    <div id="topnav" style="width:auto;">
     <ul>
       <li><a href="HealthRecordOwner.jsp">Logout</a></li>
         <li><a href="Viewcipherrequest.jsp">Ciphertext Request</a></li>
         <li><a href="ViewRe_EnData.jsp">Re-Encrypt Data</a></li>
```

```
        <li><a href="Re_EnRequest.jsp">Re-Encrypt Request</a></li>
         <li><a href="ViewRecord.jsp">Records</a></li>
         <li><a href="UploadRecord.jsp">Upload</a></li>

         <li class="active"><a href="OwnerHome.jsp">Home</a></li>
       </ul>
     </div>
     <br class="clear" />
   </div>
</div>

<div class="wrapper col3">
  <div id="container">
   <%
     String id=(String)session.getAttribute("id");
       String username=(String)session.getAttribute("username");
        String email=(String)session.getAttribute("email");
   %>
   <center>
       <h2 style="margin-bottom:300px;">Owner Home Page::'<%=email%>'.</h2>
```

```
        </center>


    <div class="clear"></div>
  </div>
</div>
<!--
#########################################################################
######################################## -->


<!--
#########################################################################
######################################## -->
<div class="wrapper col5">
  <div id="copyright">
    <p class="fl_left">Secure Keyword Search and Data Sharing Mechanism</p>
    <p class="fl_right"></p>
    <br class="clear" />
  </div>
</div>
</body>
</html>
```

```
<%@page import="java.sql.ResultSet"%>
<%@page import="com.database.Queries"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<%
try{
    String uname=request.getParameter("uname");
    String pwd=request.getParameter("pwd");
  String query="select * from owner where username='"+uname+"'and password='"+pwd+"'";
    ResultSet i=Queries.getExecuteQuery(query);
    if(i.next()){
     session.setAttribute("id",i.getString("id"));
       session.setAttribute("username",i.getString("username"));
        session.setAttribute("email",i.getString("email"));
%>
      <script type='text/javascript'>
         window.alert("Login Successful...!!");
         window.location="OwnerHome.jsp";
      </script>
      <%}else{
```

```
    %>
      <script type='text/javascript'>
          window.alert("Login Failed..!!");
          window.location="HealthRecordOwner.jsp";
      </script>
      <%
}
}catch(Exception e){
    out.println(e);
}
%>


<%@page import="java.sql.ResultSet"%>
<%@page import="com.database.Queries"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<title>Secure Keyword Search</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<link rel="stylesheet" href="layout/styles/layout.css" type="text/css" />
<script type="text/javascript" src="layout/scripts/jquery.min.js"></script>
```

```
<script type="text/javascript" src="layout/scripts/jquery.innerfade.js"></script>
</head>
<body id="top">
<div class="wrapper col1">
  <div id="header">
     <center>
     <h2>Secure Keyword Search and Data Sharing Mechanism</h2>
     </center>
    <div id="topnav" style="width:auto;">
     <ul>
       <li><a href="Delegatee.jsp">Logout</a></li>
       <li><a href="DLTViewReEncRecodsresult.jsp">View Result</a></li>
      <li class="active"><a href="DLTViewReEncRecods.jsp">View Re_Encrypted Records</a></li><li><a href="DelegateeHome.jsp">Home</a></li>
     </ul>
    </div>
    <br class="clear" />
  </div>
</div>
<div class="wrapper col3">
  <div id="container">
```

```
<%
  String id=(String)session.getAttribute("id");
    String username=(String)session.getAttribute("username");
     String email=(String)session.getAttribute("email");
%>
<center>
    <h2 style="margin-bottom:150px;">Delegatee Home Page::'<%=email%>'.</h2>

    <h4>View All Re_Encrypted Records</h4>
      <table>
      <tr>
        <th>File ID</th>
          <th>File Name</th>
            <th>Request</th>
      </tr>
<%
      try{
        String query="select * from re_enc where skey!='waiting' and re_cipher!='waiting'";
        ResultSet r=Queries.getExecuteQuery(query);
```

```
            while(r.next()){
                String rid=r.getString("id");
                String fid=r.getString("fid");
                String fname=r.getString("fname");
                  String owner=r.getString("owner");
               %>
                 <tr>
              <td><%=fid%></td>
                <td><%=fname%></td>
                <td><a
href="SendreEncRequest.jsp?id=<%=rid%>&fid=<%=fid%>&owner=<%=owner
%>">Ciphertext Request</a></td>
          </tr>
            <%}
          }catch(Exception e){
            out.println(e);
          }
%>
      </table>
\       </center>
\    <div class="clear"></div>
  </div>
```

```
</div>
<!--
########################################################################
######################################## -->


<!--
########################################################################
######################################## -->
<div class="wrapper col5">
  <div id="copyright">
    <p class="fl_left">Secure Keyword Search and Data Sharing Mechanism</p>
    <p class="fl_right"></p>
    <br class="clear" />
  </div>
</div>
</body>
</html>
```