A
Project Report
On

“Attribute-Based Encryption Approach for Storage, Sharing and Retrieval of Encrypted Data in the cloud”

Submitted in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications (MCA)

By

NAME: AVUSUNAPALLI SAMYUKTHA
ROLL NO:1009-21-862-029

Under the guidance of

Mr. T. RAMDAS NAIK
Asst. Professor

Department of Informatics
Nizam College (Autonomous)
(A Constituent College, O.U)
Basheer Bagh, Hyderabad
2022- 2023

DEPARTMENT OF INFORMATICS

NIZAM COLLEGE (AUTONOMOUS)

(A Constituent College, O.U)

CERTIFICATE

This is to certify that the project entitled **"Attribute Based Encryption Approach for Storing, sharing and retrieval of Encrypted Data in the Cloud "** has been submitted **"AVUSUNAPALLI SAMYUKTHA"** bearing Roll No: "**1009-21-862-029**" in the partial fulfillment of the requirements for the award of the degree of **Master of Computer Applications** in **Faculty of Informatics**, Nizam college, Osmania University, Hyderabad.

PROJECT GUIDE
Mr. T. RAMDAS NAIK
BE, MCA, M. Tech, (Ph. D)
Assistant Professor

EXTERNAL EXAMINER

Mr. M. PURNA CHARY
MCA (Ph. D), NET, SET.
Head of the Department,
Dept. of Informatics.

windows Information Technology

WINDOWS INFORMATION TECHNOLOGY

Near Raithu Bazaar, New N.G.O's Colony, Subash Nagar

Regd No:158/2009

CERTIFICATE

This is to certify that **AVUSUNAPALLI SAMYUKTHA** bearing the Roll No:**"1009-21-862-029"** from "**NIZAM COLLEGE"**, student of M.C.A has successfully completed the project entitled as **"ATTRIBUTE BASED ENCRYPTION FOR SHARING, STORING AND RETRIEVAL OF THE ENCRYPTED DATA INTO THE CLOUD"** part of the course curriculum in our Organization.

She has done the project using **Java** during the period **23-05-2023 to 21-28-2023 under** the guidance and supervision of **Mr.B.Gnaneshwer** from **Windows Information Technology Computers, Nizamabad.**

She has completed the assigned project well within the time frame. She is sincere, hardworking and her conduct during period is commendable.

We wish all the best in her future endeavor.

For **WIT Computers**

WINDOWS INFORMATION TECHNOLOGY
NIZAMABAD.

K. Uday Kumar
(Managing Director)

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely fortunate to have got this all along the completion of my project work. Whatever I have done is only due to such guidance and assistance and I would not forget to thank them.

I would like to thank the **ALMIGHTY,** who gave me enough strength and health to complete this task without any interruption and **My Parents** who gave me all the support during the project work.

I owe my profound gratitude to my project guide **Mr. T. RAMDAS NAIK**, Asst. Professor and other teachers who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

I thank to **Mr. M. PURNA CHARY**, Head of the Dept. Of Informatics, Nizam College, For his moral Support and Guidance.

I express my whole hearted gratitude to **Prof. B. BHIMA, Principal, Nizam College,** and to **Management Nizam College** for providing us the conductive environment to carry through our academic schedules and project with ease.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of Department of Informatics which helped us in successfully completing our project work. Also, I would like to extend our sincere regards to all the non-teaching staff of department of Informatics for their timely support.

Last but not the least; I would like to thank all my friends for their compliment

Name: AVUSUNAPALLI SAMYUKTHA

Roll No:1009-21-862-029

ABSTRACT

One of the most cost-effective services in cloud computing is storage, used by businesses and individuals to outsource their massive data to untrusted servers. Efforts have studied problems around this application scenario in different fronts: efficiency, flexibility, reliability, and security. In this paper we address the security concerns of cloud storage under the scenario where users encrypt-then-outsource data, share their outsourced data with other users, and the service provider can be queried for searching and retrieval of encrypted data. As main distinctive, we propose a security approach for storage, sharing and retrieval of encrypted data in the cloud fully constructed on the basis of attribute-based encryption (ABE) thus enabling access control mechanisms over both the encrypted data and also for the information retrieval task through search access control. Compared to related works, our approach considers efficient encryption at three different levels:

i) bulk encryption of data outsourced to the cloud,
ii) keys management for access control over encrypted data by means of digital envelopes fromattribute-based encryption, and
iii)novel construction for attribute based searchable encryption(ABSE).

Our underlying ABE algorithms are carefully selected from the body of knowledge and novel constructions for ABSE are provided over the asymmetric setting (Type-III pairings)to support security levels of 128-bits or greater. Experimental results on benchmark data sets demonstrate the viability of our approach for practical realizations using Barreto-Naehrig curves.

Index

LIST OF FIGURES

Chapter 1

1.INTRODUCTION

1.1 Problem Statement

Although ABSE theoretically meets the basic security requirements and guarantees the searching capabilities for the cloud storage service provider, in the literature there are no ABSEconstructions that have been evaluated in real scenarios. That is, there are no proposals in the state of the art that have been evaluated to determine their feasibility in real applications in such a way that the functional and non-functional requirements are satisfied in the context of HBC and SHBC scenarios. In other words, a practical scheme for secure cloud storage must beable to provide an appropriate solution to all the possible problems involved for example, among these problems are

1) An efficient distribution and management of the symmetric encryption keys, and

2) The null reliability on the service provider, since it could be either HBC or SHBC, while atthe same time it must be preserved

3) The data confidentiality

4) The retrieval and sharing capabilities of the storage service. ABSE relies in Attribute-BasedEncryption ABE), which has many variants and settings to be deployed. There are several ABSE schemes in the literature, each one suitable for different applications or use cases, and considering different properties and added functionalities. So, a carefully selection of ABE- related algorithms is required.

.

1.2 Objective

The high availability (access anytime, anywhere) and reliability of data at low cost are the main incentives for organizations and individuals to adopt cloud storage services. These services are in increasing demand due to the high amount of data generated by different sources (Internet of Things) and cloud enabled applications. However, data owners (DOs) outsourcing their datato untrusted servers in the cloud face the security concern that the cloud service provider (CSP)honestly stores the DO’s data and follows the agreed protocol but tries to learn as much as possible from the computations and interactions with the users (DU) that access the DO's data. This issue can be solved by providing DOs with a confidentiality security service for DOs toencrypt data before uploading it to the cloud. A straightforward encryption approach to prevent DO's data disclosure and to keep DO's data private from CSP or from any other entity, causesthe provider cannot manipulate data, that is, loss of utility appears as the encrypted data cannot be used by the CSP for retrieval/searching purposes. Due to that inconvenience, DUs should download large volume of encrypted data, decrypt, and then search over the plaintext data (locally), re-encrypt and upload again its data to the cloud. Of course, so one approach incurs in huge communications and computations overhead and is completely inefficient. Searchable encryption (SE) has been the most known approach to cope with the problem of searching overencrypted data stored in untrusted servers. SE is defined as the ability to identify and retrieve a set of objects from an encrypted collection that satisfy a query. In SE, the CSP executes DU'sencrypted queries over encrypted data without decryption, so it does not learn anything about the data content, search criteria, nor search patterns back to DU.

1.3 Motivation

Technological advances and the constant increase in the amount of data created every day

have made cloud storage services vital for users who need to store or back up

their information

More often, both people and organizations choose to upload their information to external servers through cloud storage services. This is because cloud computing facilitates the access to several assets on demand with broad access to the provider network through standard mechanisms. Also, cloud storage services allow users to reduce the volume of data manipulatedlocally, and prevent the acquisition of expensive storage infrastructure. Despite there are several storage models, the public cloud model has the greatest number of users. However, using public clouds can lead to certain security risks, especially when the information of data owners is sensitive, for example, clinical records, corporate financial documents, and personal data, just to name a few. As mentioned above, the use of cloud storage services provides important benefits, since the access to the resources is generally achieved through theInternet. However, it is also true that there are still pending challenges to be solved, especiallythose concerned with information security. Among these challenges are data confidentialityand its secure transmission through insecure communication channels, as well as authenticationand fine-grained access control for data sharing.

1.4 Existing System

A straightforward encryption approach to prevent DO’s data disclosure and to keep DO’s dataprivate from CSP or from any other entity, causes the provider cannot manipulate data, that is, loss of utility appears as the encrypted data cannot be used by the CSP for retrieval/searching purposes. Due to that inconvenience, DUs should download large volume of encrypted data, decrypt, and then search over the plaintext data (locally), re-encrypt and upload again its data to the cloud. Of course, so one approach incurs in huge communications and computations overhead and is completely inefficient. Searchable encryption (SE) has been the most known approach to cope with the problem of searching over encrypted data stored in untrusted servers.SE is defined as the ability to identify and retrieve a set of objects from an encrypted collectionthat satisfy a query. In SE, the CSP executes DU’s encrypted queries over encrypted data without decryption, so it does not learn anything about the data content, search criteria, nor search patterns.

1.5 Proposed System

We present a security approach for storing, sharing and retrieving of encrypted data in the cloud, fully constructed on the basis of attribute-based encryption (ABE). Our approach is wellsuited for a known cloud-based storage and sharing model, where DO uploads encrypted datato the cloud to ensure confidentiality (by means of symmetric data encryption) and establishesaccess control mechanisms for data sharing using attribute based encryption; DU can selectively locate specific documents using an index-based structure and retrieve documents ofinterest in encrypted form, without revealing any information to the CSP and under a fine- grained search control. Our proposed approach aims at meeting the following four requirementsto enable practical storage, sharing and retrieval of encrypted data in the cloud:

R1 - DO can execute Ek1 (D) efficiently to provide confidentiality over outsourced data to thecloud at the same time that enables fine-grained data access control and secure distribution of k1 for DUs, thus enabling secure data sharing.

R2 - DUs can query Ik2 (W) (via the CSP) by computing and using Tk3 (wq) at the time that secure fine-grained search control is enabled.

R3 - DUs can ask the CSP to return the k-most relevant documents from the retrieval task results, ordered accordingly to their relevance to the query.

R4 - Both R1 and R2 comply with recommended security levels1 (i.e. $\lambda \geq 128 - \text{bit}$).

We called our approach FABECS (Fully Attribute-Based Encryption scheme for Cloud Storage, Sharing and Retrieval) which fulfils requirements R1-R4. FABCS includes a novel Cipher-text policy ABSE (CP-ABSE) construction to achieve R2 and R3 requirements. At thesame time, FABECS reuses the settings of ABSE (pairings and curve parameters) for the setupof DET-ABE which provides cryptographically enforced fine-grained access controls needed to meet requirement R1

1.6 Scope

Searchable encryption (SE) [1] has been the most known approach to cope with the problem of searching over encrypted data stored in untrusted servers. SE is defined as the ability to identify and retrieve a set of objects from an encrypted collection that satisfy a query. In SE, the CSP executes DU's encrypted queries over encrypted data without decryption, so it does not learn anything about the data content, search criteria, nor search patterns. The most generalmethod to implement SE in any of its existing families [2] is as follows:

1) DO creates cryptographic keys {k1, k2, k3} with security strength λ;

2) Given a set of documents D, DO extracts representative keywords W in D;

3) DO encrypts D with a symmetric cipher E using key k1 (Ek1 (D));

4) DO creates a secure index from W, using key k2 (Ik2 (W));

5) DO uploads Ek1 (D) and Ik2 (W) to the cloud;

6) DO gives authorization to some DUs to search over its data, using k3.

7) DU generates trapdoors for a given query word wq in W using k3, Tk3 (wq);

8) DU queries the CSP using Tk3 (wq), to retrieve D 0 documents containing wq.

9) CSP searches over encrypted data using Tk3 (wq) and Ik2 (W) (CSP will not learn anythingabout data, search criteria nor query patterns)

10)The encrypted documents Ek1 (D 0) found by CSP (satisfying the query) are sent back toDU.

11)If DO authorizes DU the access D 0 in plaintext form, then DO shares k1 with DU andthus DU decrypts Ek1 (D 0) and get access to the documents set D 0 in clear

1.7 Software & Hardware Requirements

Hardware requirements

➢ Processor	: i3.
➢ Ram	: 4GB
➢ Hard Disk	: 500 GB.

Software requirements

➢ Database Server	: MySQL
➢ Database Client	: SQLYOG
➢ Server	: Apache Tomcat
➢ Platform	: Java
➢ Server-side Technologies	: JEE (Servlets, JSP)
➢ Client-Side Technologies	: HTML, CSS, JavaScript, AJAX
➢ IDE	: Eclipse
➢ UML Design/E-R Modelling Tools	: Rational Rose, SQL-Developer
➢ Testing	: JUNIT
➢ Cloud	: DriveHQ

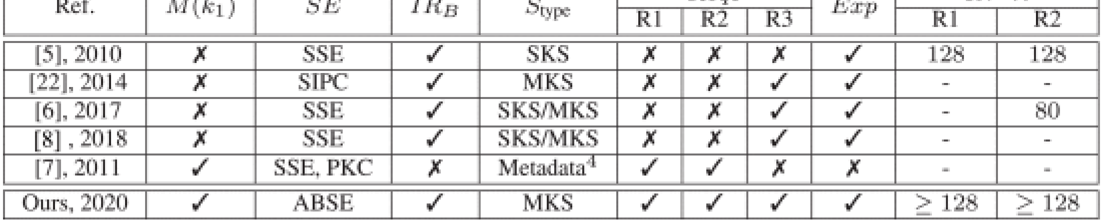
Chapter 2

2.1 Survey of Major Area Relevant to Project

Attribute-based encryption (ABE) has become an enabler technology for secure storage and sharing in the cloud, as described in Section I-A. ABE allows many-to-many encryption, which is not possible to achieve in traditional public key cryptosystems (PKC), i.e., RSA. Thus, potential receivers may not necessarily be known at the time of encryption, a task that is done by using an access policy that enforces all those whose attributes satisfy the policy can decrypt and gain access to plaintext data. Atomically, ABE provides both confidentiality and fine-grained access control over data. Attributes are taken from a universe U. The access policy, expressed as an *access structure* A, restricts the decryption capabilities of the intended destinations depending on the attributes set possessed.

The two prominent approaches for attribute-based encryption that have been proposed in the state-of-the-art are KP-ABE and CP-ABE. In KP-ABE [20], policies are associated to decryption keys and attributes to the ciphertext. Contrary, in CP-ABE [21] the ciphertext is created with a policy and decryption keys are associated to user's attributes. CP-ABE is conceptually closer to Role Based Access Control and more natural to apply than is KP-ABE to enforce access control over encrypted data. Therfore, the cryptosystems based on CP-ABE are considered an attractive option for providing confidentiality service and fine-grained access control mechanisms at the same time [11]. Attribute-based searchable encryption (ABSE) is defined using either KP-ABE or CP-ABE as basis.

Table 1 summarizes the main aspects of previous works proposing searchable encryption approaches for cloud-based storage systems. None of them use ABE for both access control over outsourced encrypted data neither for the information retrieval task (ABSE) as we proposein this work.



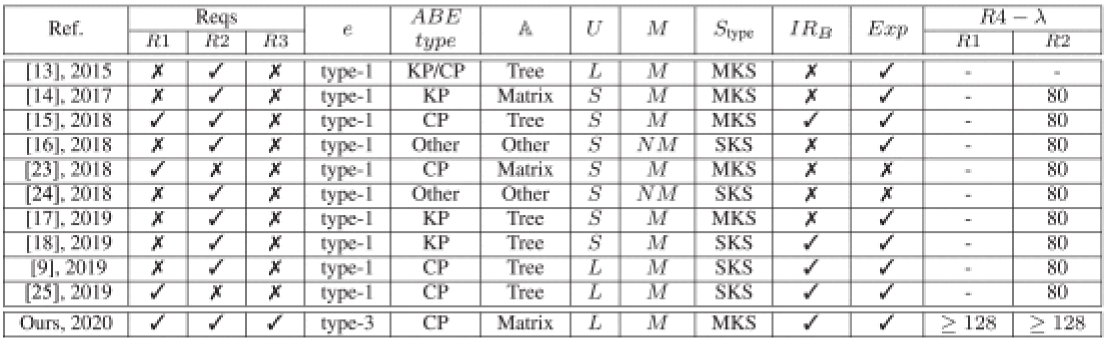
Ref.	$M(k_1)$	SE	IR_B	S_{type}	*Reqs*			Exp	R4 - λ	
					R1	R2	R3		R1	R2
[5], 2010	✗	SSE	✓	SKS	✗	✗	✗	✓	128	128
[22], 2014	✗	SIPC	✓	MKS	✗	✗	✓	✓	-	-
[6], 2017	✗	SSE	✓	SKS/MKS	✗	✗	✓	✓	-	80
[8] , 2018	✗	SSE	✓	SKS/MKS	✗	✗	✓	✓	-	-
[7], 2011	✓	SSE, PKC	✗	Metadata^4	✓	✓	✗	✗	-	-
Ours, 2020	✓	ABSE	✓	MKS	✓	✓	✓	✓	≥ 128	≥ 128

TABLE 1 Generalities of Searchable Encryption Approaches (Non-Attribute-Based) in the Literature for Cloud-Based Storage Scenarios

It is relevant to note from Table 1 is that most of the reported works do not consider any key management mechanism M(k1) (as it is specified in the R1 requirement), and such a mechanism is crucial in practice, otherwise data could not be decrypted neither accessed by DUs. For the information retrieval task, the most common technique used has been SSE, either for single keyword search (SKS) or multi-keyword search (MKS). Furthermore, it is worth to mention that some solutions consider evaluation based on known benchmarks (IRB), which include a ranking mechanism (R3 requirement) to evaluate the effectiveness of the retrieval task. It is also worth to mention, from Table 1, that some works have carried out experiments (Exp) that have been limited only to the retrieval task (if R3 is checked) or only considering a specific security value in R4 for the encryption (R1) and search (R2) tasks.

Authors in [7] proposed to implement M(k1) by means of public key encryption (PKE) as a kind of digital envelope. Destinations of encrypted data receive a key from their public keys. So, public keys must be distributed previously and digital certificates are mandatory for practical realizations. In addition, PKE only allows coarse-grain access control so fine-grained and many-to-many encryption is not possible. Experimental evaluation was not reported.

Since our approach is fully based on attribute-based encryption (ABE) to fulfill R1-R4, we provide in Table 2 a summary of previous works that have considered ABE or ABSE approaches for enabling a cloud-based searchable encryption system. It is worth noting that in most of the cases, the four requirements (R1 - R4) for a functional cloud-based searchable encryption system have not been met neither both ABE and ABSE have been considered at the same time to achieve R1-R4. Furthermore, provided ABE realizations have been limited to constructions only for outdated security levels (symmetric pairing, 80-bits).



Ref.	Reqs			e	ABE $type$	$\mathbb{A}$	U	M	S_{type}	IR_B	Exp	$R4-\lambda$	
	$R1$	$R2$	$R3$									$R1$	$R2$
[13], 2015	✗	✓	✗	type-1	KP/CP	Tree	L	M	MKS	✗	✓	-	-
[14], 2017	✗	✓	✗	type-1	KP	Matrix	S	M	MKS	✗	✓	-	80
[15], 2018	✓	✓	✗	type-1	CP	Tree	S	M	MKS	✓	✓	-	80
[16], 2018	✗	✓	✗	type-1	Other	Other	S	NM	SKS	✗	✓	-	80
[23], 2018	✓	✗	✗	type-1	CP	Matrix	S	M	MKS	✗	✗	-	80
[24], 2018	✗	✓	✗	type-1	Other	Other	S	NM	SKS	✗	✗	-	80
[17], 2019	✗	✓	✗	type-1	KP	Tree	S	M	MKS	✗	✓	-	80
[18], 2019	✗	✓	✗	type-1	KP	Tree	S	M	SKS	✓	✓	-	80
[9], 2019	✗	✓	✗	type-1	CP	Tree	L	M	SKS	✓	✓	-	80
[25], 2019	✓	✗	✗	type-1	CP	Tree	L	M	SKS	✓	✓	-	80
Ours, 2020	✓	✓	✓	type-3	CP	Matrix	L	M	MKS	✓	✓	≥ 128	≥ 128

TABLE 2 Attribute-Based Approaches for Searchable Encryption in Cloud-Based Storage Systems, Focusing on Either Access Control Over Encrypted Data (ABE) or on Searchable Encryption (ABSE)

In Table 2, we have highlighted the following relevant aspects of ABE in the provided constructions: the pairing used (e) to deploy ABE; the ABE type used, which is important in terms of access control applicability; the access structure being used, being the most common the tree structure (for ABE constructions based on [21]) or the matrix structure (for LSSS); attribute universe U , small (S) or large (L) (in the former, all the attributes must be known in advance and once ABE is deployed no more attributes can be added); access structure expressiveness M , being better preferred monotonic policies (M) instead of non-monotonic ones (*NM*); Stype indicates how the searching process is based (on single word *SKS* or multi-keyword *MKS* search); IRB indicates if benchmarks were used to evaluate the information retrieval efficiency, very important to fulfill R3 requirement; Exp indicates if some experimental evaluation was done.

Authors in [13] proposed to use either KP- or CP- ABE versions only to fulfill R2, however, they do not provide a specific construction and their experiments are vaguely explained (there are no details about datasets, security levels, ABE settings, etc). The approach in [14] also only focuses on the R2 requirement, limited to small universe KP-ABE, and with scarce experimental details to evaluate if R3 was or not met.

In [15], there is a first attempt to fulfill R1 and R2. However, it fails to achieve R3 and R4, besides the construction is for an ABE small attribute universe. In [16], [17], [24], and [18] only R2 is addressed and the construction is limited to a small attribute universe. Theusage of non-monotonic access structures in [16] and [24] led to less expressiveness of accesspolicies. In [23], R1 is fulfilled by using ABE, but R2 is achieved by deriving a special key depending on DU’s attribute set. However, the secure index is not encrypted with an access structure as required in ABSE. Besides, the construction is limited to a small attribute universe. The main advantage of the construction provided in [9] over [14], [16]–[18] is the usage of CP-ABE in large attribute universe. However, it still fails to fulfill R1. Although in [25] the idea is to use ABE to fulfill R1, R2 is met by using SSE, not with ABSE. So, R2 is not completelyfulfilled as the secure index is not encrypted with an access structure as required in ABSE.

For all proposals in Table 2, the provided ABE realizations to fulfill either R1 or R2 requirements are limited to the type-1 pairing, which is a symmetric pairing not practical for realizing ABE for higher security levels as demanded by R4 requirement (the greater the security level, the greater the size of the operands and hence the processing times). ABE construction based on the Type 1 pairing have been only efficiently realized for the outdated

80-bit security level. In [9], authors proposed a ciphertext-policy attribute-based searchable encryption schema (CP-ABSE), relying on the CP-ABE scheme proposed in [21] (hereafter referred to as BSW07 scheme). Although sufficient for practical use, the BSW07 construction has the main disadvantage that its security analysis is limited to the generic group model.

In this work, we provide a novel CP-ABSE construction over Type-III pairings taking as basis the well-known CP-ABE model proposed by Waters [26] (hereafter referred as W11 scheme). The W11 construction was proven to be secure in a more solid model, the standard one. Furthermore, W11 construction was proved to be both foundationally sound and practical, supporting expressive access control structures. Because of that, most of the actual realizations of CP-ABEs are compliant with that construction. The original W11 scheme was proposed for the Type-I pairing, not for searchable encryption and limited to the symmetric pairings. Additionally, we include in our CP-ABSE construction the LSSS scheme reported in [27] in order to reduce the space of the ABE ciphertext, not considered in [9]. Our CP-ABSE construction considers a large attributes universe setting, which allows supporting expressive access structures.

2.2Techniques and Algorithms

Pairings are mathematical objects defined over groups and efficient computable pairings are used to construct several cryptographic algorithms and protocols such as ABE.

Bilinear Pairings A bilinear pairing [28] is defined as the mapping $e: G1\times G2\rightarrow GT$, where $G1=\langle g1\rangle$, $G2=\langle g2\rangle$, and GT are cyclic groups of prime order r , with g1 and g2 the generators of G1 and G2 respectively. The paring e must satisfy the properties of

2.2.1 Bilinearlity: $\forall g\in G1$, $\forall g^\wedge\in G2$, and $a,b \in Z*r$, $e(ga, g^\wedge b)=e(g, g^\wedge)ab$.

2.2.2 Non-degeneracy: $e (g1, g2) \neq 1$.

$Z*r$ is the set $\{1,2,\ldots,r-1\}$. In practice, G1 and G2 are subgroups of a so-called friendly elliptic curve E defined over a finite field Fq , and GT is the multiplicative group of extensionfield Fqk with k referred to as the embedding degree of the elliptic curve, the smallest positiveinteger such that r divides to $qk-1$. If G1=G2 the pairing is symmetric (Type-I), otherwise it is asymmetric. If the pairing is asymmetric and no efficient isomorphism is known between G1 and G2, then the pairing is said to be of Type-III.

While a pairing-based cryptographic construction for a Type-III pairing is easy to transform to Type-I (by taking G1=G2), the opposite is not trivial. The main motivation to use Type-III pairing constructions of pairing-based cryptographic protocols such as ABE is because of performance and security [29]. Type-I pairings has shown serious security issues [30]. Several applications of Type-III paring-based constructions are in practical use, for example for protecting privacy of transactions with the zk-SNARKs algorithm [31] and in several Blockchain projects [32]. A kind of friendly elliptic curve is the Barreto-Naehrig curve (BN curve) [33], which constructs Type-III pairings well suited for practical usage.

Security Assumptions

Definition 1 (Discrete Logarithm (DL) Assumption):

Let G be a cyclic group with λ -bit prime order r. Let g be a generator of G. Given g and ga,the DL assumption is defined as: no probabilistic polynomial-time adversary A can compute a∈Z∗r with a non-negligible advantage Adv DLA(λ) in the security parameter λ.

The security level of pairing-friendly curves is estimated by the computational cost of the most efficient algorithm to solve the discrete logarithm problem (DLP) in the groups G1,G2 and GT . This estimation is reflected in recommended group size λ=log2r givenby the size in bits for q and k by several standards and international projects [34]. In this work,all the underlying constructions of ABE are for the Type-III pairing realized with updated groups size λ as recommended in [35].

As other pairing-based cryptographic schemes, the ABE constructions proposed in this work are based on the following security assumptions for the Type-III pairings. Definition 2 (Decisional Diffie-Hellman (DDH) Assumption): Let G be a cyclic group with λ -bit prime order r. Let g be a generator of G. Given the tuple (g, ga, gb) ∈G with a,b,c∈Z∗r (random), the DDH assumption is defined as: no probabilistic polynomial-time adversary A can decide whether gab is equal to gc with a non-negligible advantage Adv DDHA(λ). The DDH holds if for any A with output in {0,1}

$$\mathrm{Adv}\ \mathrm{DDHA}(\lambda)=||\Pr [A\ (g,\ ga,\ gb,\ gab)] -\Pr [A\ (g,\ ga,\ gb,\ gc)]\ ||$$

is negligible in the security parameter λ.

Definition 3 (Symmetric External Diffie-Hellman (SXDH) Assumption):

Let Pλ be the setting for a Type-III pairing consisting on the tuple {G1, G2, GT, g1, g2, r}. The SXDH assumption holds if DDH holds in both G1 and G2.

Definition 4 (Decisional Bilinear Diffie-Hellman (DBDH) Type-III Assumption [36]):
Let Pλ be the setting for a Type-III pairing consisting on the tuple {G1, G2, GT, g1, g2, r}. The DBDH assumption is defined as: no probabilistic polynomial-time adversary A can distinguish {g1, g2, ga1, gb1, gc1, gb2, gc2, e (g1, g2) abc} from {g1, g2, ga1, gb1,gc1,gb2,gc2,e(g1,g2)z} for random elements a,b,z∈Z∗r with a non-negligible advantage Adv DBDHA(λ).

B. CP-ABE and DET-ABE

A Secret Sharing Scheme (SSS) in attribute-based encryption (e.g., CP-ABE) is crucial. In this context, an access structure A is defined as a non-empty subset of the power set P(U), with U a set of attributes. SA is called an authorized set if SA∈A. An SSS involves a dealer that sharesa secret s with a set of n parties defined by U. In CP-ABE, SAs are authorized sets of attributesto decrypt a ciphertext. A is said to be monotone, if given SA∈A and SA⊂SB, then SB∈A. That is, decryption privileges are not lost even if more attributes are acquired by decryptors. Inthe W11 ABE construction, A is an n×l matrix (for an access policy including n attributes) instead of the tree access structure used in the BSW07 construction, without loss of efficiency. In this work, the linear SSS (LSSS) proposed by Liu and Cao in [27] is used, where the conceptof Formatted Boolean Formula (FBF) over attributes is proposed to represent the access policy associated to the access structure A.

CP-ABE (as other PKC schemes) relies on mathematical operations over large numbers (hundreds of bits). That is why CP-ABE is not used to encrypt data massively. For this purpose, symmetric ciphers E using the encryption key k1 (i.e., Ek1(D)) are faster and preferred. Digital envelopes [37] have been effective methods to encrypt large data with E and securely distribute k1 to intended decryptors. In [38], digital envelopes realized from attribute-based encryption (called DET-ABE) were proposed by using the BSW07 construction. Later, in [11] the construction was extended to the ABE construction based on W11. In DET-ABE, data D is encrypted with the Advanced Encryption Standard (AES) using a session key k1, which is encrypted with CP-ABE with an access policy. Only those authorized entities with a valid set of attributes satisfying the ABE encryption policy could decrypt and recover the dataencryption key k1 to then decrypt and recover D. DET-ABE can be formally defined by the four polynomial-time algorithms [11]:

1. **DET-ABE.setup**(1λ) → {*MK*, *PK*}: Creates *MK* (master private key) and *PK* (master public key), such that the length of both keys is compliant with the λ security strength. Initializes the attributes universe set U.

2. **DET-ABE.privateKeyGen**(*MK*, Su) → SK_u: Creates the user private key SK_u by using *MK*, and given a set of attributes Su⊂U, specific for the user u.
3. **DET-ABE.encrypt**(D, *PK*, A) → *CT*D: Encrypts data D using the AES cipher with a private session key k1. Then, encrypts k1 with CP-ABE using *PK* and an access structure A, which corresponds to an access policy over attributes in U. The resulting encryption is the package *CT*D= {*AES*k1(D), CP-ABE (k1, A)}.
4. **DET-ABE.decrypt**(*CT*D, SK_u) → D: Decrypts CP-ABE k1, A) using decryptor's private key SK_u to recover k1. Then, decrypts *AES*k1(D) using k1. Decryption works only if the attribute set Su used to generate SK_u is in A.

DET-ABE provides effective access control mechanisms and confidentiality over a large documents dataset. It has been successfully tested in other systems [39], [40], however, DET-ABE does not support searchable encryption (DET-ABE only meets requirement R1).

In [9], authors proposed CP-ABSE, a realization of ABSE using the CP-ABE BSW07 construction as basis. As it is the case of other searchable encryption techniques, CP-ABSE constructs a secure index, that is, the index is encrypted so it does not support traditional queries over plaintext keywords; the matching is cryptographically tested. The secure index maps encrypted keywords to encrypted documents, so the searching process does not leak information about the data being queried. CP-ABSE consists of the following five polynomial-time algorithms [9].

1. **CP-ABSE.setup**(1λ) → {dk_1, dk_2}: Creates two keys dk_1 and dk_2; dk_1 is used to encrypt an index keyword given an access structure A and dk_2 is used to generate user'sprivate key given its attribute set SA in an attribute universe U. With private keys, users can create encrypted queries for data retrieval.
2. **CP-ABSE.privateKeyGen**(dk_2, Su) → SK_u: Creates the user private key SK_u using dk_2 from a set of attributes Su⊂U, specific for the user u.
3. **CP-ABSE.encInd**(dk_1, w, A) → CT_w: Creates a secure index by encrypting with CP-ABE each keyword w in the index. The result is the set of each encrypted keyword *CT*w= CP-ABE (dk_1, w, A).
4. **CP-ABSE.Trpdr**(SK_u, wq) →Tu(wq): Generates an encrypted query from keyword wq, by using the user private key SK_u. The encrypted query is the trapdoor Tu(wq).

5. **CP-ABSE.search**(CTw,Tu(wq))→{0,1} : Given the encrypted keyword CT_w and a trapdoor Tu(wq) , the algorithm returns ‘1’ if w=wq (cryptographically tested) and if the attribute set Su that generated SK_u and used to create Tu(wq) is in the access structure A used to encrypt CT_w, simultaneously; otherwise it outputs ’0’.

The original CP-ABSE construction in [9] was provided for the Type-I pairing (symmetric), based on the BSW07 CP-ABE construction. In this work, we provide a novel CP-ABSE construction over the Type-III pairing, which allows realizations for security levels equal or greater than 128-bits. Furthermore, our construction is not based on the CP-ABE BSW07 construction but on the W11, which has been proved to be both foundationally sound and practical, supporting expressive access control structures using a large attributes universe setting.

2.3Applications

- Attribute-based encryption (ABE) can be used for log encryption.
- Traditionally, everyone with the right key can decrypt and thus access the encrypted file. ABE is not that different. However instead of using the public key to encrypt the files, it uses attribute(s) or a key based on attributes to encrypt the files.

Chapter 3

System Design

3.1 System Architecture

MVC stands for Model View and Controller. It is a design pattern that separates the business logic, presentation logic and data.

MVC Structure has the following three parts:

Controller acts as an interface between View and Model. Controller intercepts all the incoming requests.

Model represents the state of the application i.e., data. It can also have business logic.

View represents the presentation i.e., UI (User Interface).

Advantage of MVC Architecture

1.Navigation Control is centralized

2.Easy to maintain the large application

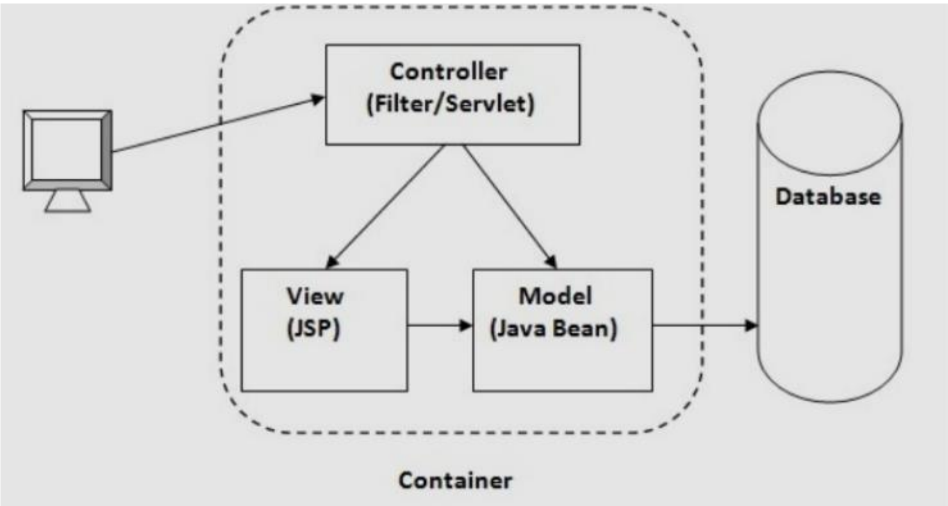


Fig: 3.1.1 System Architecture

Technical Architecture

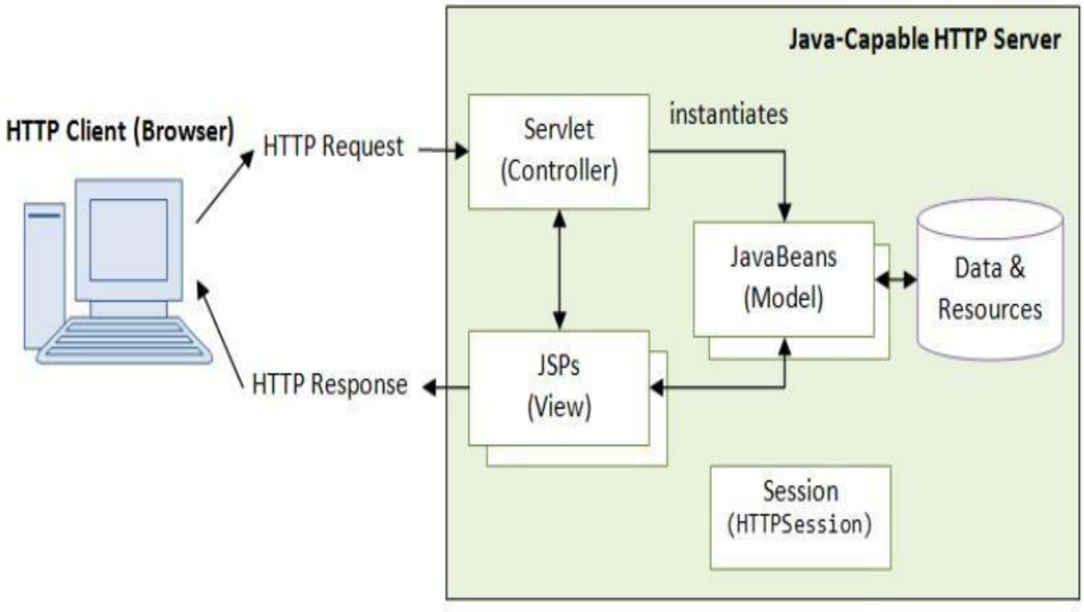


Fig:3.1.2 Technical Architecture

3.2 System Flow

UML Design

Unified Modelling Language (UML) is a general-purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

UML is not a programming language; it is rather a visual language. We use UML diagrams to portray the behaviour and structure of a system, UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object Management Group (OMG) adopted Unified Modelling Language as a standard in 1997. It's been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

Do we really need UML?

Complex applications need collaboration and planning from multiple teams and hence requirea clear and concise way to communicate amongst them.

Businessmen do not understand code. So, UML becomes essential to communicate with non-programmer's essential requirements, functionalities and processes of the system.

A lot of time is saved down the line when teams are able to visualize processes, user interactions and static structure of the system.

UML is linked with object-oriented design and analysis. UML makes the use of elements and forms associations between them to form diagrams. Diagrams in UML can be broadly classified as:

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

Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

Provide extendibility and specialization mechanisms to extend the core concepts.

Be independent of particular programming languages and development process.

Provide a formal basis for understanding the modeling language.

Encourage the growth of OO tools market.

Support higher level development concepts such as collaborations, frameworks, patterns and components.

Integrate best practices.

Types of UML Diagrams

Structural Diagrams

Capture static aspects or structure of a system. Structural Diagrams include: Component Diagrams, Object Diagrams, Class Diagrams and Deployment Diagrams.

Behaviour Diagrams

Capture dynamic aspects or behaviour of the system. Behaviour diagrams include: Use Case Diagrams, State Diagrams, Activity Diagrams and Interaction Diagrams.

The image below shows the hierarchy of diagrams according to UML

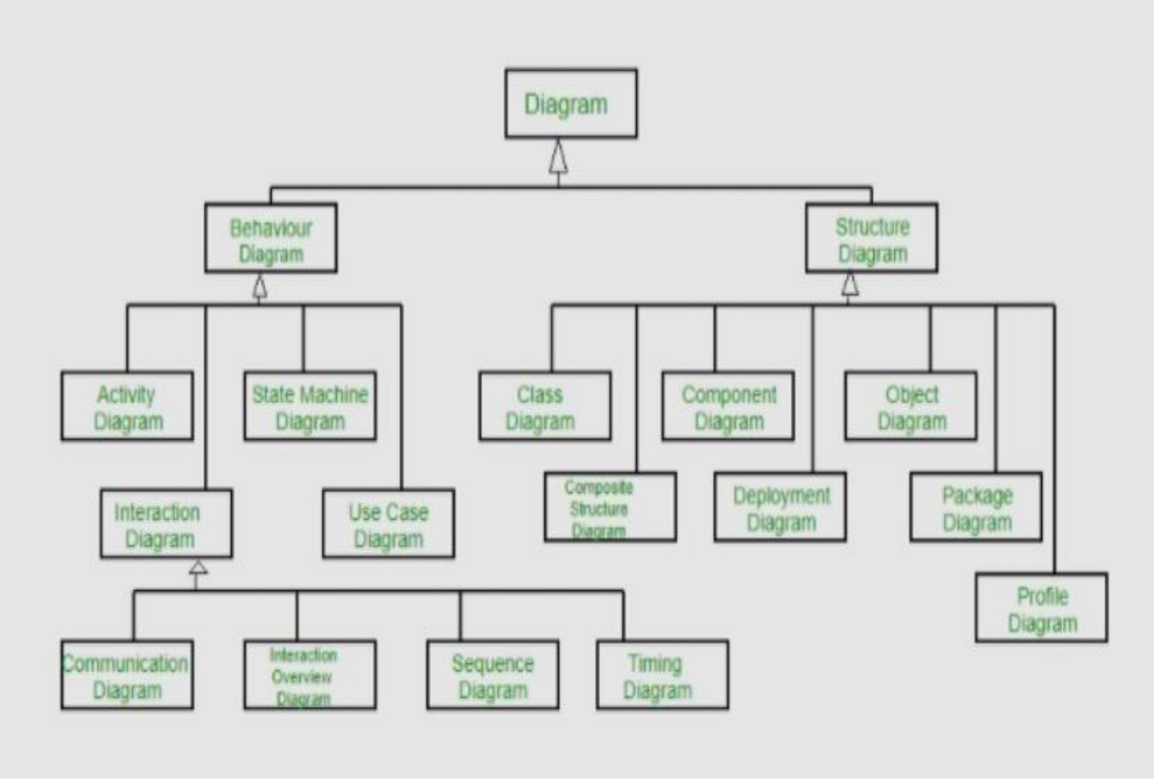


Fig:3.2.1 Classification of UML

Use Case Diagram

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural 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.

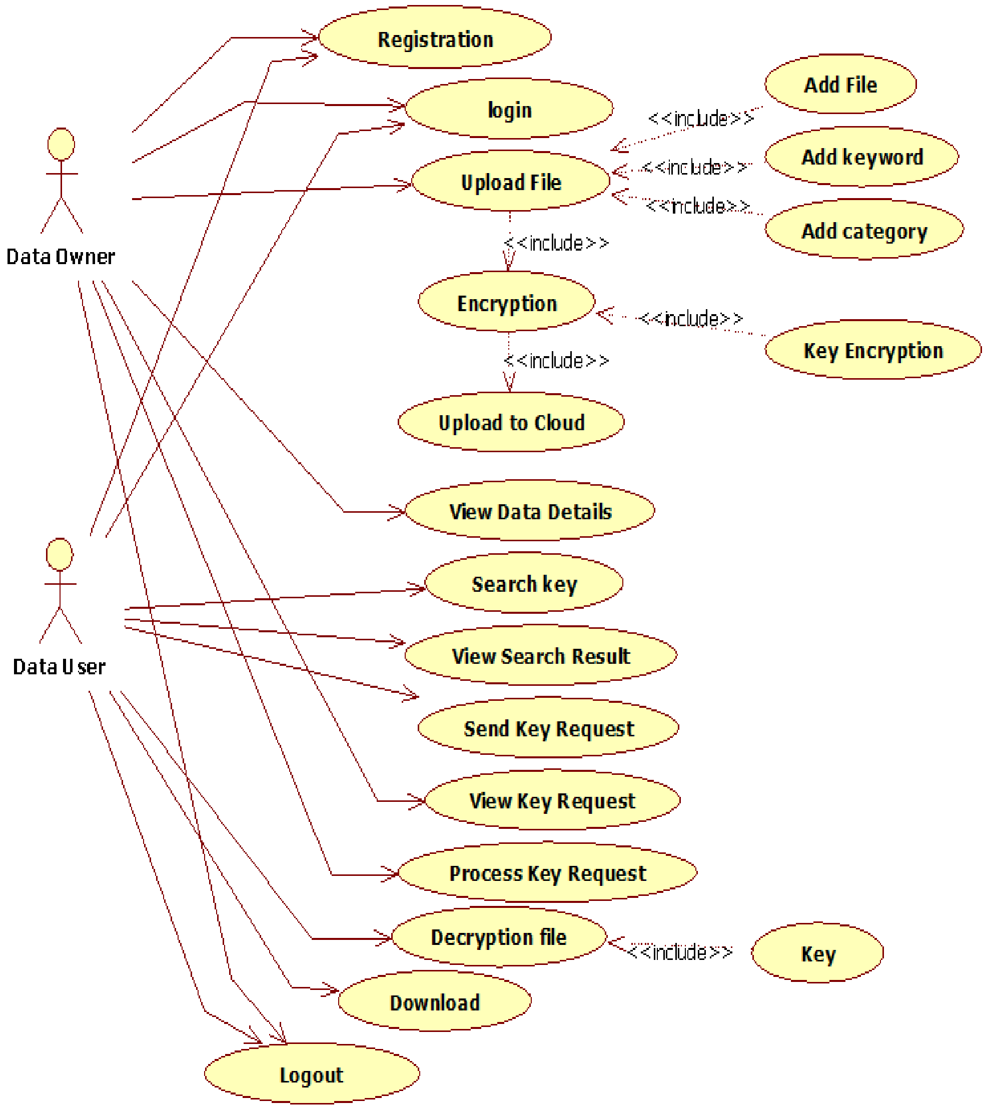


Fig:3.2.2 Use Case Diagram

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.

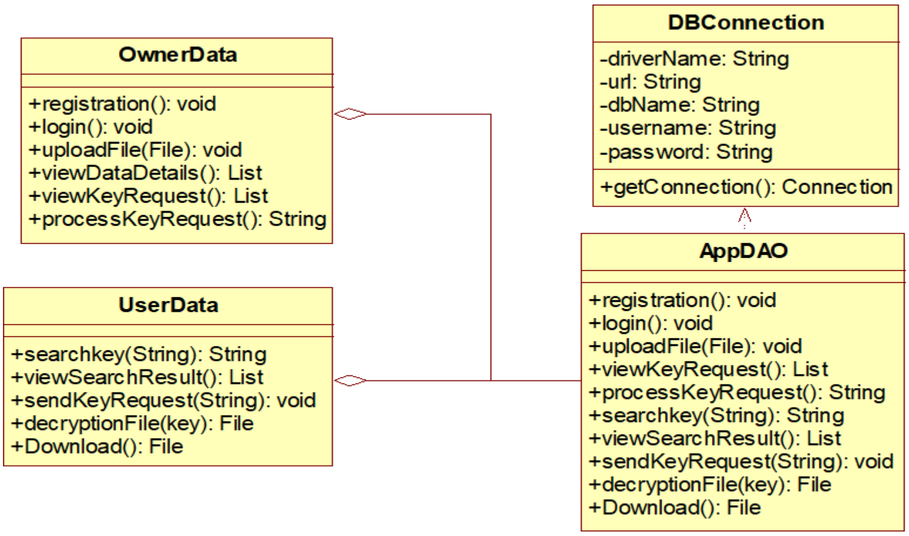


Fig :3.2.3 Class Diagram

Sequence Diagram

A sequence diagram in Unified Modelling 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.

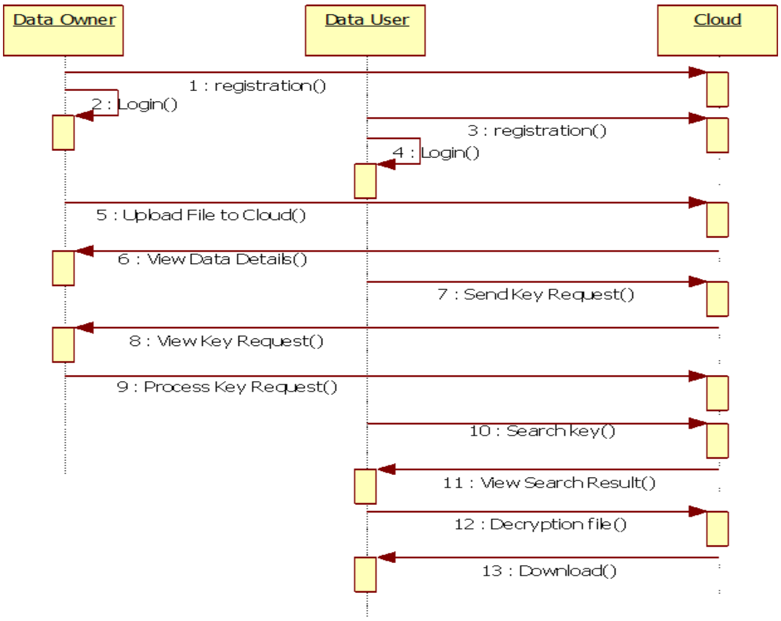


Fig :3.2.3 Sequence Diagram

Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

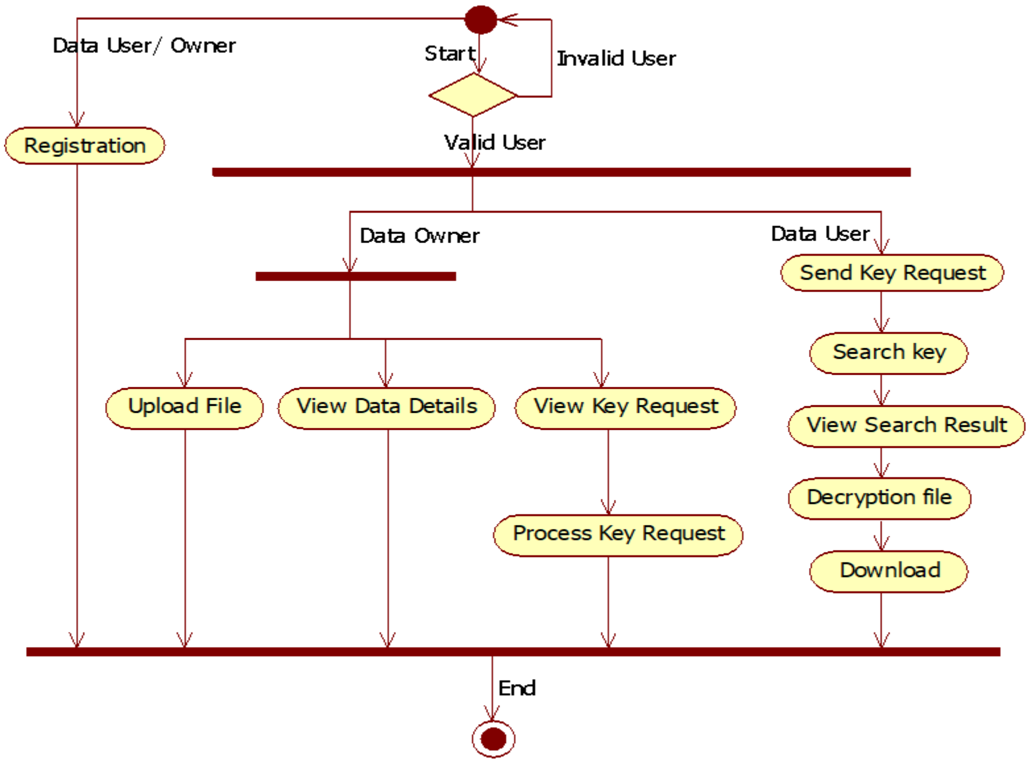


Fig: 3.2.4 Activity Diagram

Collaboration Diagram

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

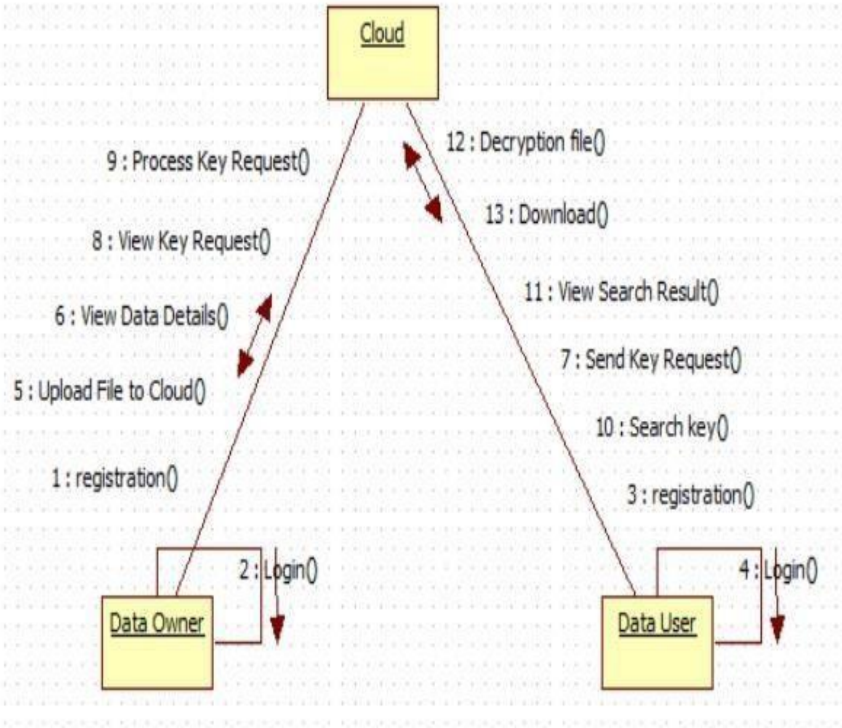


Fig 3.2.5 Collaboration Diagram

Deployment Diagram

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

The deployment diagram maps the software architecture created in design to the physical system architecture that executes it. In distributed systems, it models the distribution of the software across the physical nodes.

The software systems are manifested using various artifacts, and then they are mapped to the execution environment that is going to execute the software such as nodes. Many nodes are involved in the deployment diagram; hence, the relation between them is represented using communication paths.

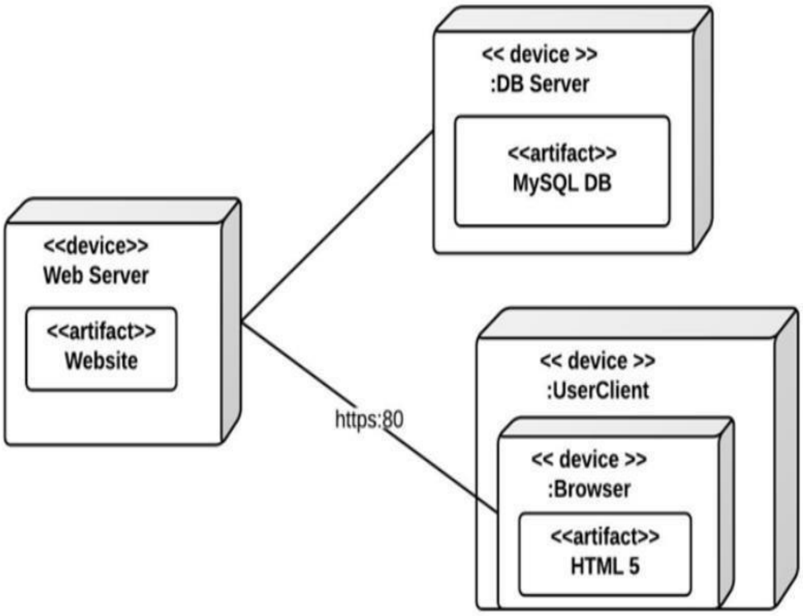


Fig:3.2.6 Deployment Diagram

Entity-Relationship Diagrams

ER-modelling is a data modelling method used in software engineering to produce a conceptual data model of an information system. Diagrams created using this ER-modelling method are called Entity-Relationship Diagrams or ER diagrams or ERDs.

Purpose of ERD

The database analyst gains a better understanding of the data to be contained in the database through the step of constructing the ERD.

The ERD serves as a documentation tool.

Finally, the ERD is used to connect the logical structure of the database to users. In particular, the ERD effectively communicates the logic of the database to users.

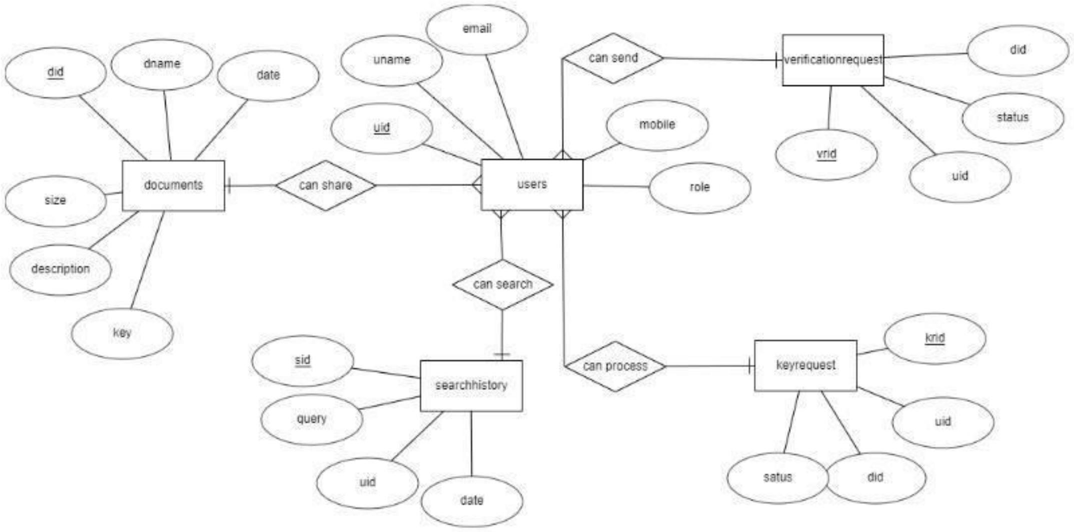


Fig: 3.2.7 Entity-Relationship Diagram

3.3 Module Description

Our approach, fully constructed on attribute-based encryption is named FABECS. It is createdon the basis of DET-ABE and CP-ABSE, each one relying on CP-ABE. The system model inthis work is for a cloud-based document sharing and retrieval system with enabled searchableencryption, graphically shown in Fig. 1. The main actors in this model and their capabilities are summarized in Table 3. DO possesses a set of documents D. It creates the secure index SIfrom keywords set in D by using CP-ABSE (with search control policies given by access structure as). DO encrypts D using DET-ABE (with data access control policies given by accessstructure Ad). Thus, DO achieves R1 and R2 requirements. By using the available service provided by the CSP, DO uploads to the cloud both the encrypted documents CTD (as digital envelopes) and the secure (encrypted) index SI. DUs with properly authorized attribute set Sucreates encrypted queries (trapdoors) Tu(wq) using ABSE private keys for given keywords ofinterest wq. DU sends the encrypted query to the CSP. The CSP uses Tu(wq) to query the encrypted index SI and to locate the documents set Ek1 (D 0) (D 0 ⊂ D) that satisfies the search criteria. For efficiency and bandwidth savings, the CSP can rank the results and returns back to DU the k-most relevant encrypted documents that satisfy the query. Finally, DU uses its DET-ABE private key (derived from its attributes) to decrypt and obtain D 0 (documents in clear).

System module consist of Three modules :

1.Data owner (DO)

2.Data User(DU)

3.Cloud Service Provider (CSP)

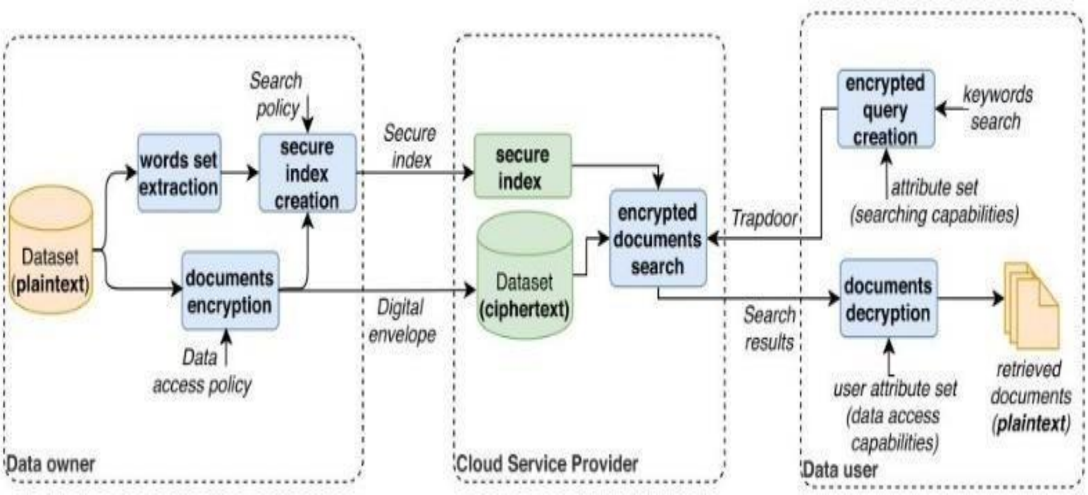
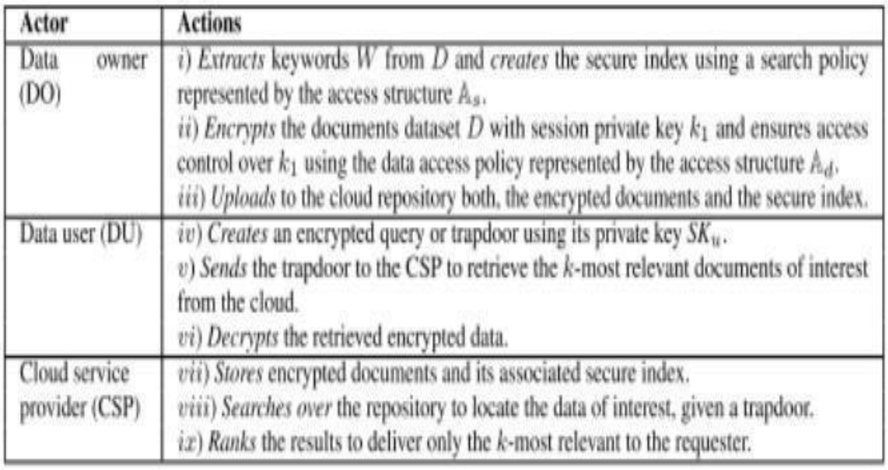


Fig: 3.3.1 Module Description

TABLE 3. Actors and their main actions in the system model.



Actor	Actions
Data owner (DO)	*i) Extracts* keywords W from D and *creates* the secure index using a search policy represented by the access structure $\mathbb{A}_s$. *ii) Encrypts* the documents dataset D with session private key k_1 and ensures access control over k_1 using the data access policy represented by the access structure $\mathbb{A}_d$. *iii) Uploads* to the cloud repository both, the encrypted documents and the secure index.
Data user (DU)	*iv) Creates* an encrypted query or trapdoor using its private key SK_u. *v) Sends* the trapdoor to the CSP to retrieve the k-most relevant documents of interest from the cloud. *vi) Decrypts* the retrieved encrypted data.
Cloud service provider (CSP)	*vii) Stores* encrypted documents and its associated secure index. *viii) Searches over* the repository to locate the data of interest, given a trapdoor. *ix) Ranks* the results to deliver only the k-most relevant to the requester.

In the system model, it is assumed that DO is fully trusted. The adversary is the CSP, e.g., a system administrator that maliciously obtain remote access to storage infrastructure of CSP but cannot access the volatile memory. The adversary does not modify/destroy the stored data but tries to derive sensitive information from the stored documents, DU's queries as well as search outcomes. The CSP is supposed to only know the DO's encrypted data set and the DO's searchable index I (known ciphertext model). During search operations, the CSP could posse more knowledge as correlation relationship of given search requests (trapdoors), as well as statistical information of data sets (Known background model) [22]. The system model previously described to support query-based retrieval over encrypted data relies on the following assumptions. 1) There are means to authenticate each actor in the system. 2) There is a secure way for DU to generate and obtain the encrypted queries. 3) Key management, encryption and decryption of documents and queries are operations of proven semantically secure ciphers for data encryption and secure index generation. 4) Denial-of-service attacks are excluded. The adversary aims to compromise the confidentiality of data or searching privileges by forging existing access policies generated by the corresponding data owners.

V. OUR APPROACH FULLY BASED ON ATTRIBUTE-BASED ENCRYPTION FABECS

is constructed taking advantage of shared components in DET-ABE and CP-ABSE. Setup in DET-ABE and CP-ABSE are in essence the same, with MK = dk2 and PK = dk1. So, one single setup can serve for both realizations. With DET-ABE and ABSE defined over a large universe scenario, the attribute set U is dynamically updated. The private Keygen function in DET-ABE and CP-ABSE, with the proper attribute set, are the same. CP-ABE encryption can be displayed as a single module used to realize completely CP-ABSE. EncInd and the symmetrickey encryption in DET-ABE. Encrypt. In each case, a different access structure can be used if searching and data access capabilities are different. Finally, the core of CP-ABE decryption consisting in recovering a secret from the attributes in the decryption key (for DET-ABE. Decrypt) or in the trapdoor (for CP-ABSE. search) is the same, but the final result is different. This module can also be reused to serve as a common engine for DET-ABE and CP-ABSE.

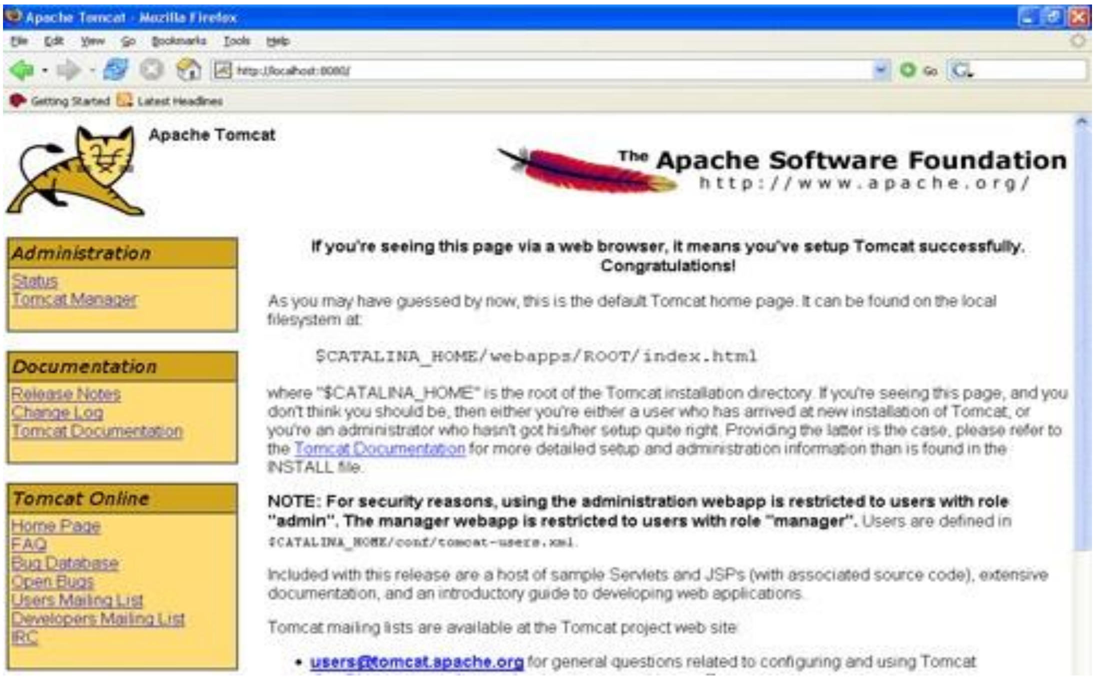
Chapter 4

IMPLEMENTATION

4.1 Environmental Setup

Tomcat 6.0 web server

Tomcat is an open source web server developed by Apache Group. Apache Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and Java Server Pages technologies. The Java Servlet and Java Server Pages specifications are developed by Sun under the Java Community Process. Web Servers like Apache Tomcat support only web components while an application server supports web components as well as business components (BE As Web logic, is one of the popular application server). To develop a web application with jsp/servlet install any web server like Run, Tomcat etc., to run your application.



If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

```
$CATALINA_HOME/webapps/ROOT/index.html
```

where "$CATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the Tomcat Documentation for more detailed setup and administration information than is found in the INSTALL file.

NOTE: For security reasons, using the administration webapp is restricted to users with role "admin". The manager webapp is restricted to users with role "manager". Users are defined in `$CATALINA_HOME/conf/tomcat-users.xml`.

Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation, and an introductory guide to developing web applications.

Tomcat mailing lists are available at the Tomcat project web site:

- **users@tomcat.apache.org** for general questions related to configuring and using Tomcat

Fig 4.1.1 Tomcat 6.0 web server

The Java Platform

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.

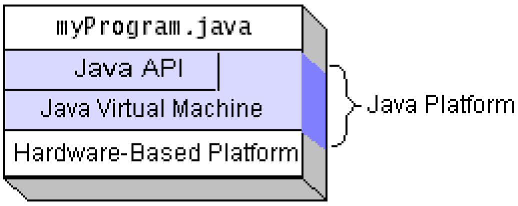
The Java platform has two components:

- The Java Virtual Machine (Java VM)
- The java Application Interface (Java API)

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

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 bit slower 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.

4.2 Implementation of Modules

In this work, we present a security approach for storing, sharing and retrieving of encrypted data in the cloud, fully constructed on the basis of attribute-based encryption (ABE). Our approach is well suited for a known cloud-based storage and sharing model [8], where DO uploads encrypted data to the cloud to ensure confidentiality (by means of symmetric data encryption) and establishes access control mechanisms for data sharing using attribute based encryption; DU can selectively locate specific documents using an index-based structure and

retrieve documents of interest in encrypted form, without revealing any information to the CSP and under a fine-grained search control. Our proposed approach aims at meeting the following four requirements to enable practical storage, sharing and retrieval of encrypted data in the cloud:

R1 - DO can execute Ek1 (D) efficiently to provide confidentiality over outsourced data to the cloud at the same time that enables fine-grained data access control and secure distribution of k1 for DUs, thus enabling secure data sharing.

R2 - DUs can query Ik2 (W) (via the CSP) by computing and using Tk3 (wq) at the time that secure fine-grained search control is enabled.

R3 - DUs can ask the CSP to return the k-most relevant documents from the retrieval task results, ordered accordingly to their relevance to the query.

R4 - Both R1 and R2 comply with recommended security levels 1 (i.e. $\lambda \geq 128 - \text{bit}$).

We called our approach FABECS (Fully Attribute-Based Encryption scheme for Cloud Storage, Sharing and Retrieval) which fulfills requirements R1-R4. FABCS includes a novel.

4.2 Integration and Development

To integrate and develop an attribute-based encryption (ABE) approach for storage, sharing, and retrieval of encrypted data in the cloud, we need to consider the following steps and aspects:

1. **System Design:** Begin by designing the overall architecture of your system, taking into account the different modules and their interactions. Define the roles and responsibilities of each module and how they will communicate with each other.
2. **ABE Scheme Selection:** Choose an appropriate ABE scheme that aligns with your requirements. There are various types of ABE schemes, such as key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). Consider the complexity, security

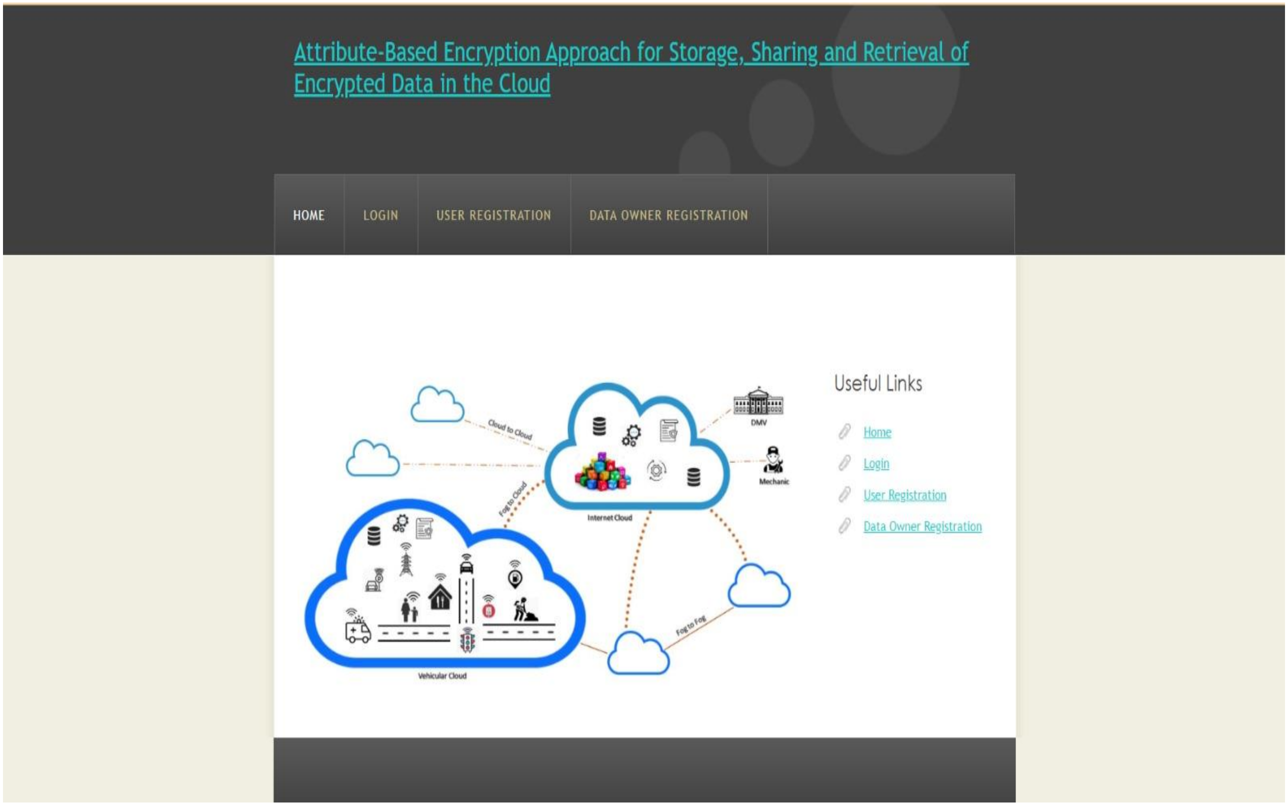
guarantees, and suitability for your specific use case.

3. **Environment Setup:** Set up the development environment by installing the necessary software libraries, frameworks, and dependencies required for implementing ABE. For example, you might need cryptographic libraries like OpenSSL or existing ABE libraries such as Charm-Crypto or Bouncy Castle.
4. **Implementation of Key Generation Module:** Develop the key generation module that generates the master key and user-specific keys based on the attributes. This module should handle attribute management, key generation, and storage of keys securely.
5. **Implementation of Encryption Module:** Implement the encryption module that performs data encryption using ABE. This module should convert the access policy into an access structure, generate a random symmetric key, encrypt the data using the symmetric key, and encrypt the symmetric key using ABE with the access structure.
6. **Implementation of Decryption Module:** Develop the decryption module that allows authorized users to decrypt the data. This module should retrieve the user's attributes, generate the user's private key based on the attributes, decrypt the symmetric key using the private key, and use the decrypted key to decrypt the data.
7. **Access Control Implementation:** Implement the access control module responsible for enforcing access policies based on attributes. This module should compare the user's attributes with the access policy requirements and grant or deny access accordingly.
8. **Integration with Cloud Storage:** Integrate your ABE system with the cloud storage service. This involves developing the storage and retrieval module that handles storing the encrypted data, associated metadata, and coordinating with the decryption module to decrypt data when authorized users access it.
9. **Testing and Security Evaluation:** Thoroughly test your implementation to ensure its correctness and evaluate the security properties of your ABE system. Perform testing scenarios to validate the functionality, access control enforcement, and encryption/decryption processes.
10. **Deployment and Maintenance:** Once you have tested and validated your ABE system, deploy it in a cloud environment and monitor its performance and security. Regularly update and maintain the system to address any potential vulnerabilities or improvements.

4.3 Integration and Development

1.Home Page

Home page is the first page in our website it shows login, User Registration and Data owner Registration Details



Attribute-Based Encryption Approach for Storage, Sharing and Retrieval of Encrypted Data in the Cloud

Useful Links

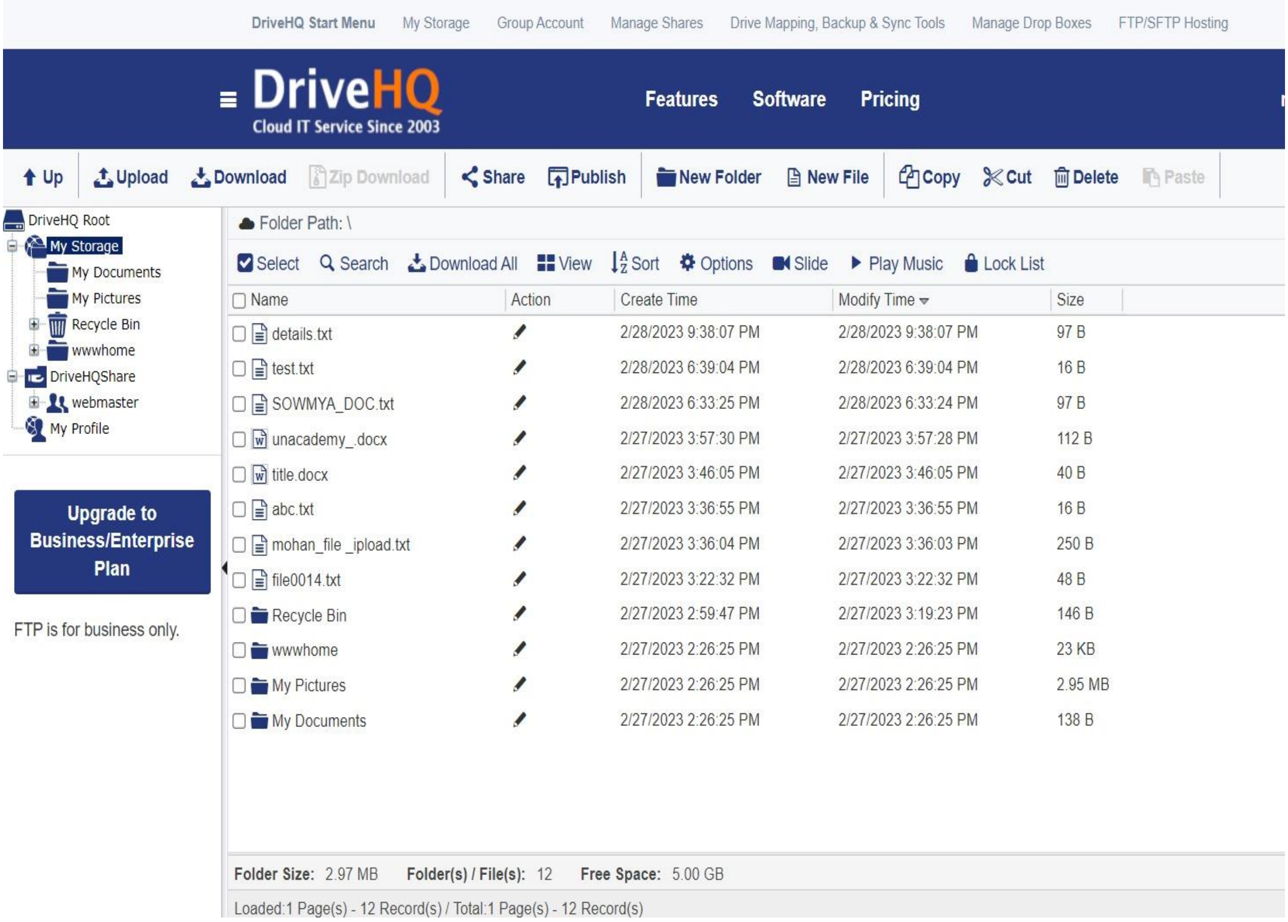
- Home
- Login
- User Registration
- Data Owner Registration

Fig 4.3.1: Home page

From home page we redirect to owner registration page and user registration page from their we should first register the user and owner. After completion registration user can login and view portal. Data Owner can login modify the data according to user requirements.

2.Drive Page

In this page we can see the Documents, files that are saved in the “My Storage” file.
Files are selected from the drive page and uploaded to the cloud.



□ Name	Action	Create Time	Modify Time	Size
□ details.txt	✎	2/28/2023 9:38:07 PM	2/28/2023 9:38:07 PM	97 B
□ test.txt	✎	2/28/2023 6:39:04 PM	2/28/2023 6:39:04 PM	16 B
□ SOWMYA_DOC.txt	✎	2/28/2023 6:33:25 PM	2/28/2023 6:33:24 PM	97 B
□ unacademy_.docx	✎	2/27/2023 3:57:30 PM	2/27/2023 3:57:28 PM	112 B
□ title.docx	✎	2/27/2023 3:46:05 PM	2/27/2023 3:46:05 PM	40 B
□ abc.txt	✎	2/27/2023 3:36:55 PM	2/27/2023 3:36:55 PM	16 B
□ mohan_file _ipload.txt	✎	2/27/2023 3:36:04 PM	2/27/2023 3:36:03 PM	250 B
□ file0014.txt	✎	2/27/2023 3:22:32 PM	2/27/2023 3:22:32 PM	48 B
□ Recycle Bin	✎	2/27/2023 2:59:47 PM	2/27/2023 3:19:23 PM	146 B
□ wwwhome	✎	2/27/2023 2:26:25 PM	2/27/2023 2:26:25 PM	23 KB
□ My Pictures	✎	2/27/2023 2:26:25 PM	2/27/2023 2:26:25 PM	2.95 MB
□ My Documents	✎	2/27/2023 2:26:25 PM	2/27/2023 2:26:25 PM	138 B

Folder Size: 2.97 MB Folder(s) / File(s): 12 Free Space: 5.00 GB

Loaded:1 Page(s) - 12 Record(s) / Total:1 Page(s) - 12 Record(s)

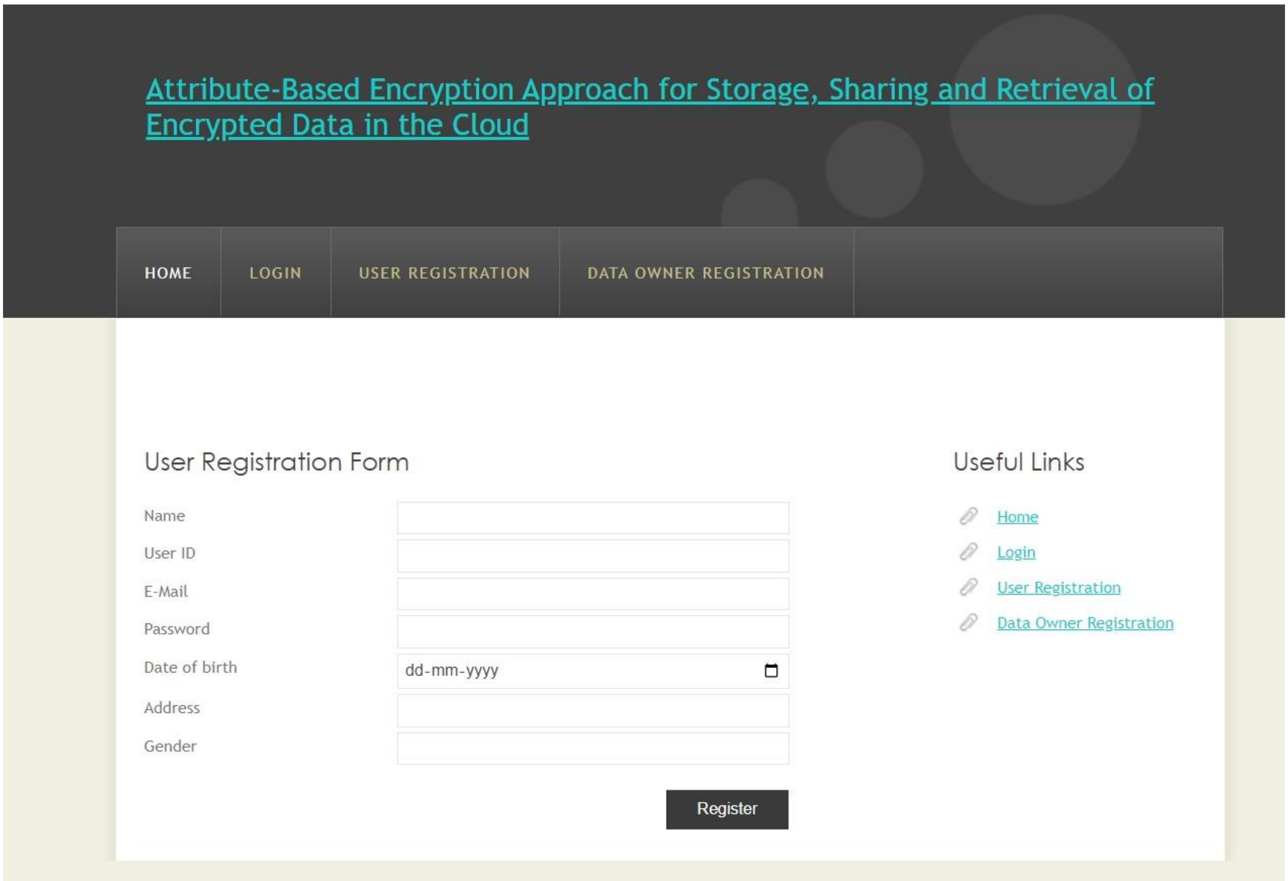
Fig 4.3.2: Drive Page

In drive page, while the Data Owner is uploading a file to the cloud, he can view the all the files which are stored in drive and can a select a particular file. The selected file is uploaded so that user can view his files by user login.

To keep a file secure here, Data Owner can give permissions accordingly to the user. Data owner has all the rights to give the permissions to all the users. Only specified permissions are given according to user requirements.

3.User Registration Page

New user will register in this page by providing below shown credentials.



Attribute-Based Encryption Approach for Storage, Sharing and Retrieval of Encrypted Data in the Cloud

User Registration Form

Name

User ID

E-Mail

Password

Date of birth dd-mm-yyyy

Address

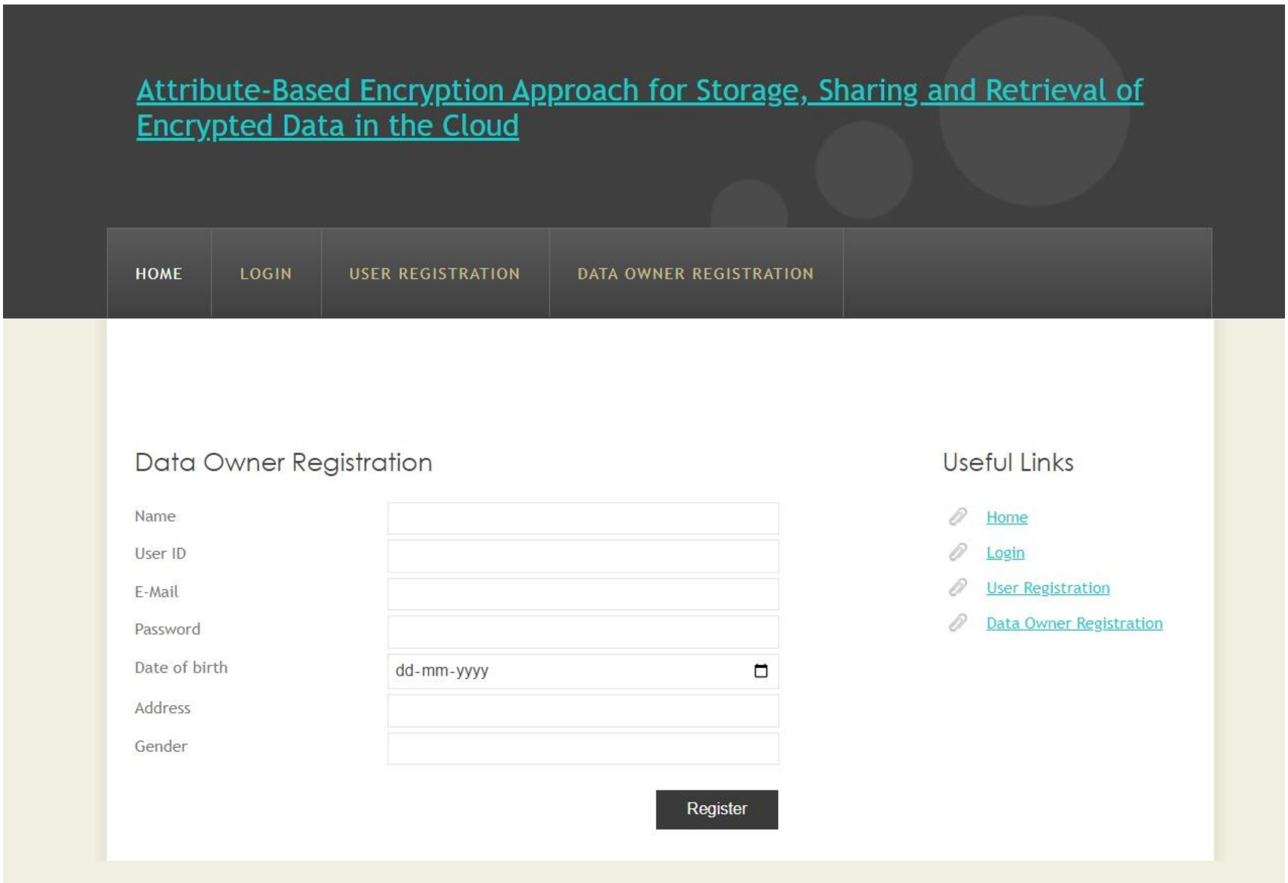
Gender

Fig 4.3.3: User Registration page

User should registered first with their name, user ID, E-mail, and few more details here User should be remember his User-ID and password which was given by him then only user can login with that details.

4.Data Owner Registration

The Data owner will register in this page by providing his credentials and click on the Register button provided there



Attribute-Based Encryption Approach for Storage, Sharing and Retrieval of Encrypted Data in the Cloud

Data Owner Registration

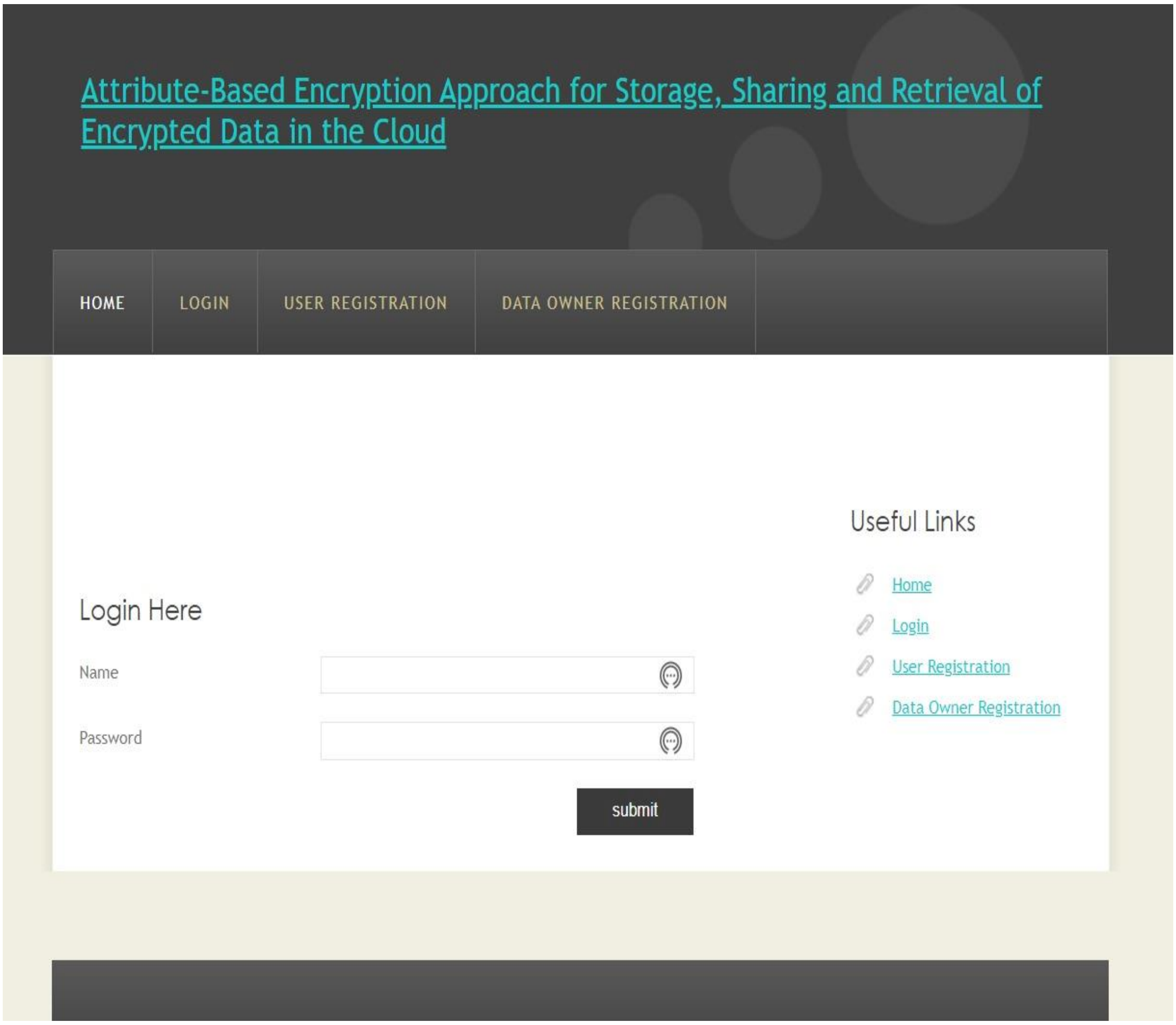
Name	
User ID	
E-Mail	
Password	
Date of birth	dd-mm-yyyy
Address	
Gender	

Fig 4.3.4: Data owner registration

Data Owner should be registered with his name and user-Id. Data owner should also remember his user-Id and password which he was given while the time of registration process.
Whenever Data owner want to login or modify any changes, he should be login with the user-Id and password then, he can modify the details and permissions.

5.Login page

Ther User can login to the website by using his credentials he mentioned at the time of registering



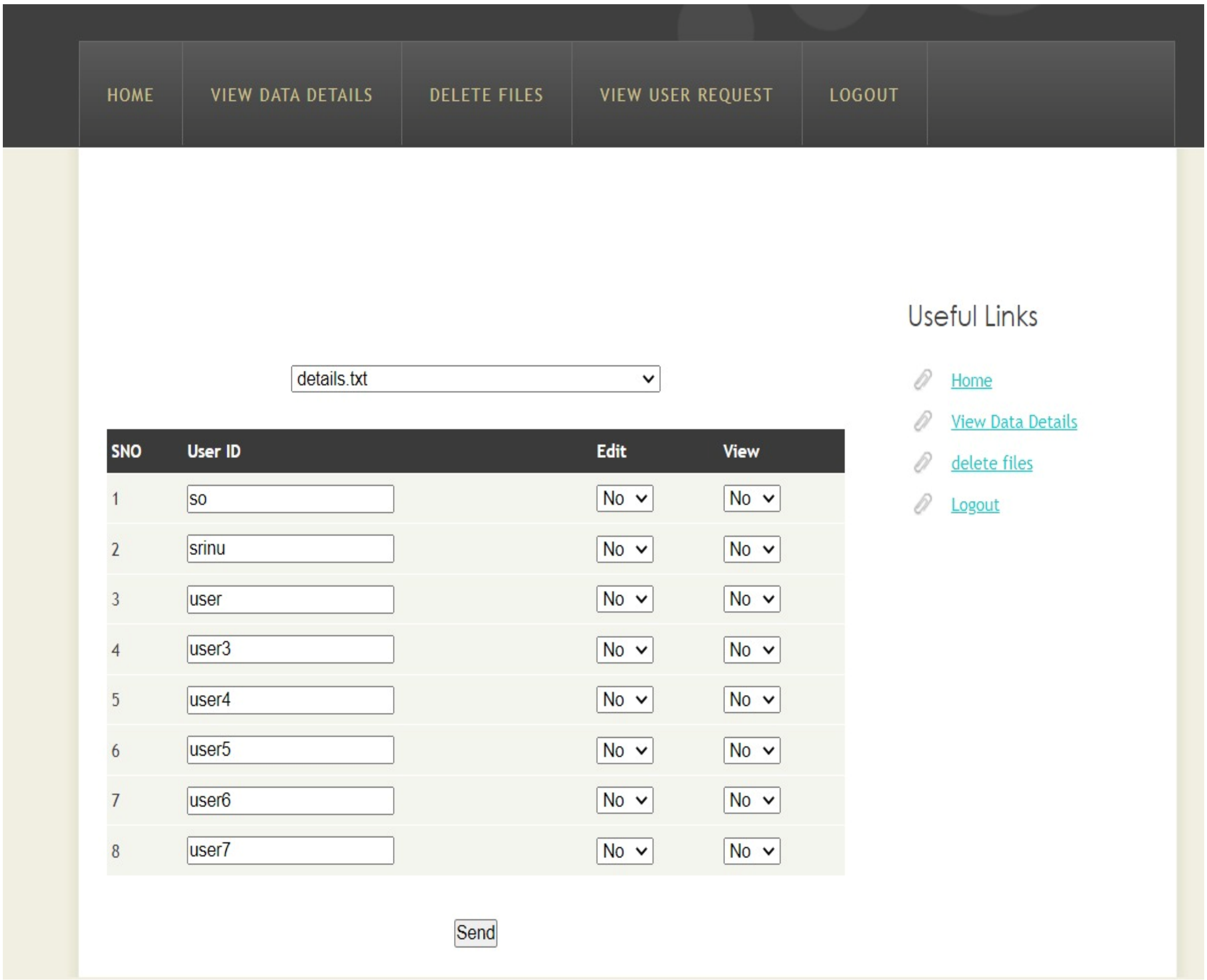
Attribute-Based Encryption Approach for Storage, Sharing and Retrieval of Encrypted Data in the Cloud

Fig 4.3.5: Login page

From the login form, user and data owner can login and view their page.
User-id and password are mandatory to login and view the page.

6.View Page

The user can view the files and Edit them so that no other unauthorized user can view and edit them



details.txt

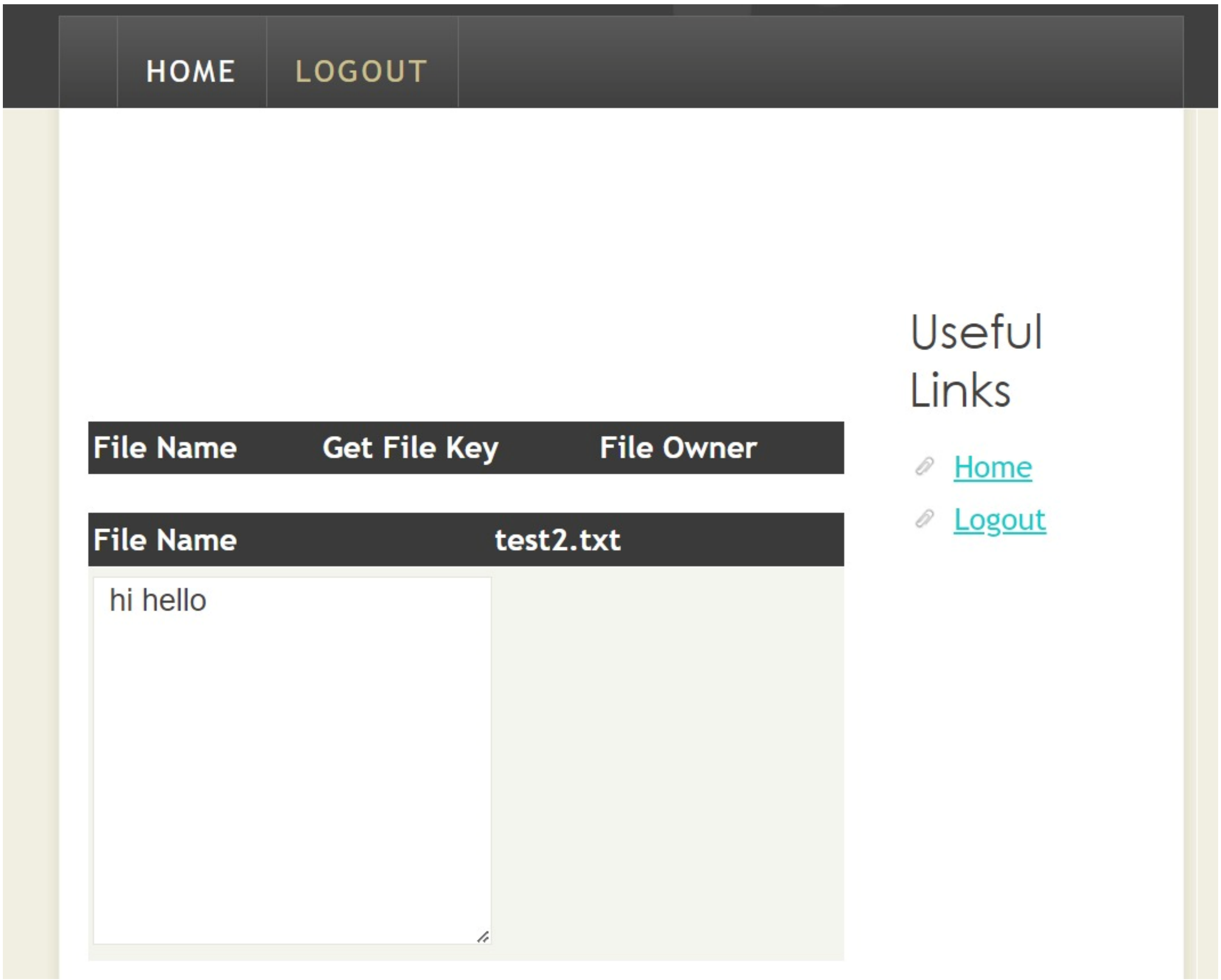
SNO	User ID	Edit	View
1	so	No	No
2	srinu	No	No
3	user	No	No
4	user3	No	No
5	user4	No	No
6	user5	No	No
7	user6	No	No
8	user7	No	No

Fig 4.3.6: View page

The user can edit and view the file.This permissions are given by the Data Owner. According to user Data owner can modify the permissions. Some of the users can only view the file cannot modify the file.

7.Logout Page

Here the user can view the file he uploaded and logout from the page and the user can login anytime by using his details.



File Name	Get File Key	File Owner

File Name	test2.txt

hi hello

Fig 4.3.7: Logout Page

Chapter 5

System Testing

5.1. Datasets:

Tables in DBMS. A database is a collection of related data. Each of these data’s is grouped into different related groups. Each of groups is stored in the physical memory like disks in theform of bits. But when a user wants to see some specific data, if he is given in the form of bits,he will not understand what it is.

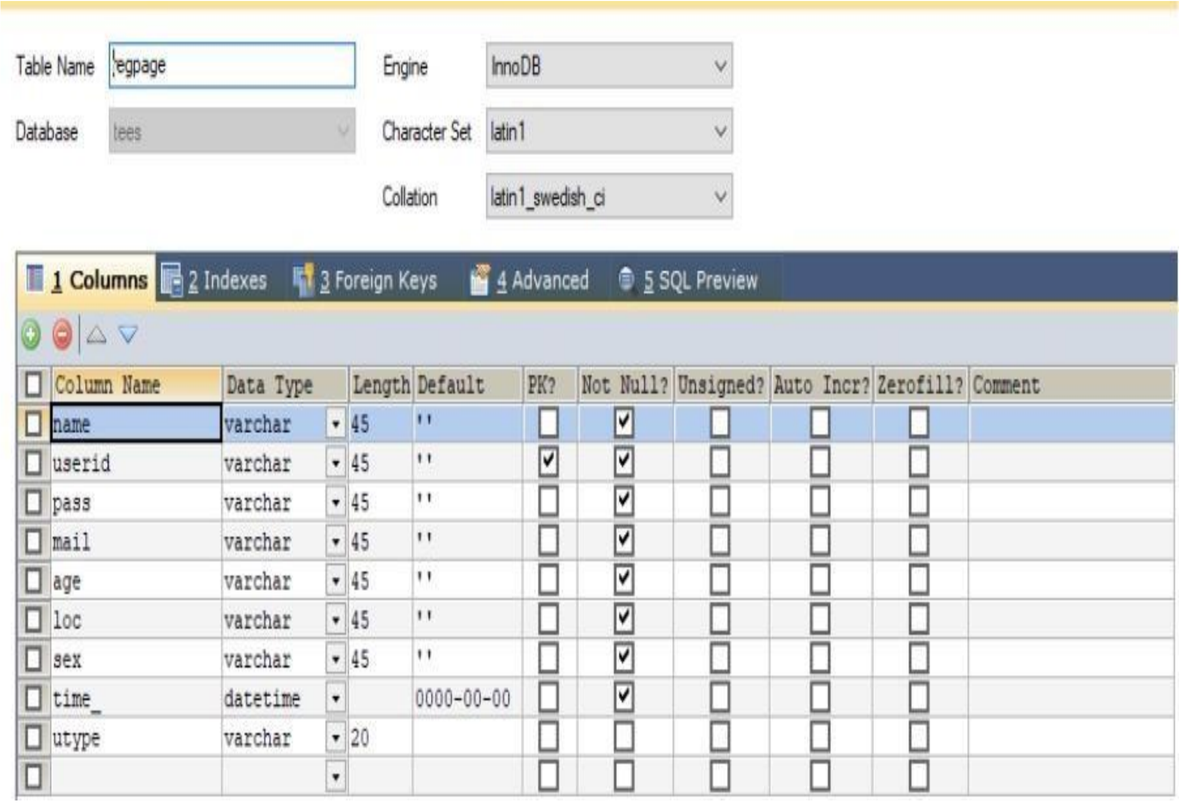


Table Name: regpage
Database: tees
Engine: InnoDB
Character Set: latin1
Collation: latin1_swedish_ci

☐	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
☐	name	varchar	45	''	☐	☑	☐	☐	☐	
☐	userid	varchar	45	''	☑	☑	☐	☐	☐	
☐	pass	varchar	45	''	☐	☑	☐	☐	☐	
☐	mail	varchar	45	''	☐	☑	☐	☐	☐	
☐	age	varchar	45	''	☐	☑	☐	☐	☐	
☐	loc	varchar	45	''	☐	☑	☐	☐	☐	
☐	sex	varchar	45	''	☐	☑	☐	☐	☐	
☐	time_	datetime		0000-00-00	☐	☑	☐	☐	☐	
☐	utype	varchar	20		☐	☐	☐	☐	☐	
☐					☐	☐	☐	☐	☐	

Fig 5.1.1: Registration page Table

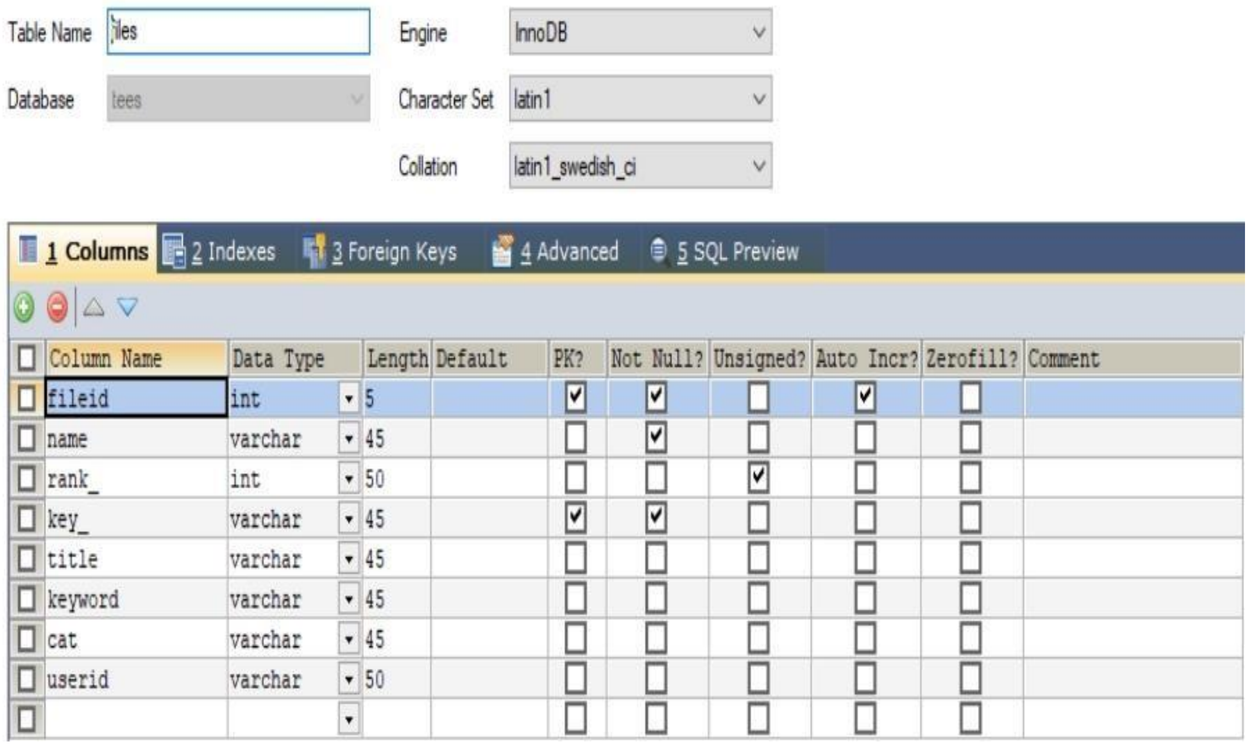


Table Name: files

Database: tees

Engine: InnoDB

Character Set: latin1

Collation: latin1_swedish_ci

☐	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
☐	fileid	int	5		☑	☑	☐	☑	☐	
☐	name	varchar	45		☐	☑	☐	☐	☐	
☐	rank_	int	50		☐	☐	☑	☐	☐	
☐	key_	varchar	45		☑	☑	☐	☐	☐	
☐	title	varchar	45		☐	☐	☐	☐	☐	
☐	keyword	varchar	45		☐	☐	☐	☐	☐	
☐	cat	varchar	45		☐	☐	☐	☐	☐	
☐	userid	varchar	50		☐	☐	☐	☐	☐	
☐					☐	☐	☐	☐	☐	

Fig 5.1.2: Files Table

User can view all the files which are uploaded by the Data Owner. User can read the file and modify the file but all users can do that only specified users are able to modify. This specifications are given by the Data owner.

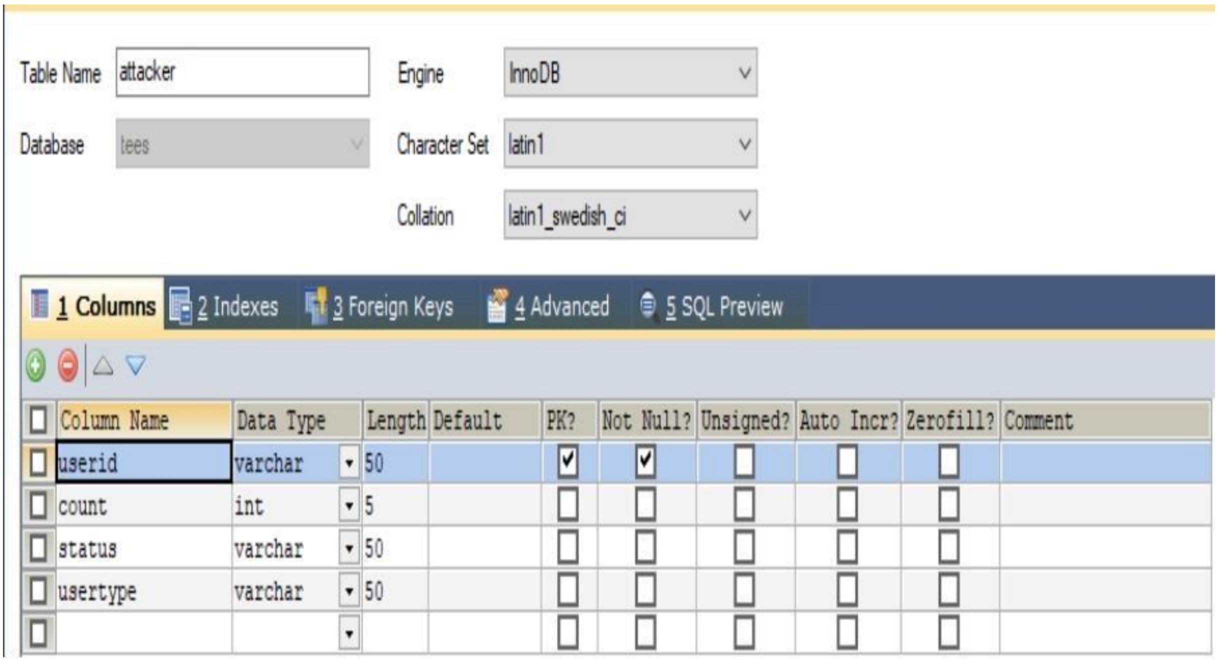


Table Name: attacker

Database: tees

Engine: InnoDB

Character Set: latin1

Collation: latin1_swedish_ci

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
userid	varchar	50		☑	☑	☐	☐	☐	
count	int	5		☐	☐	☐	☐	☐	
status	varchar	50		☐	☐	☐	☐	☐	
usertype	varchar	50		☐	☐	☐	☐	☐	
				☐	☐	☐	☐	☐	

Fig 5.1.3: Attacker Table

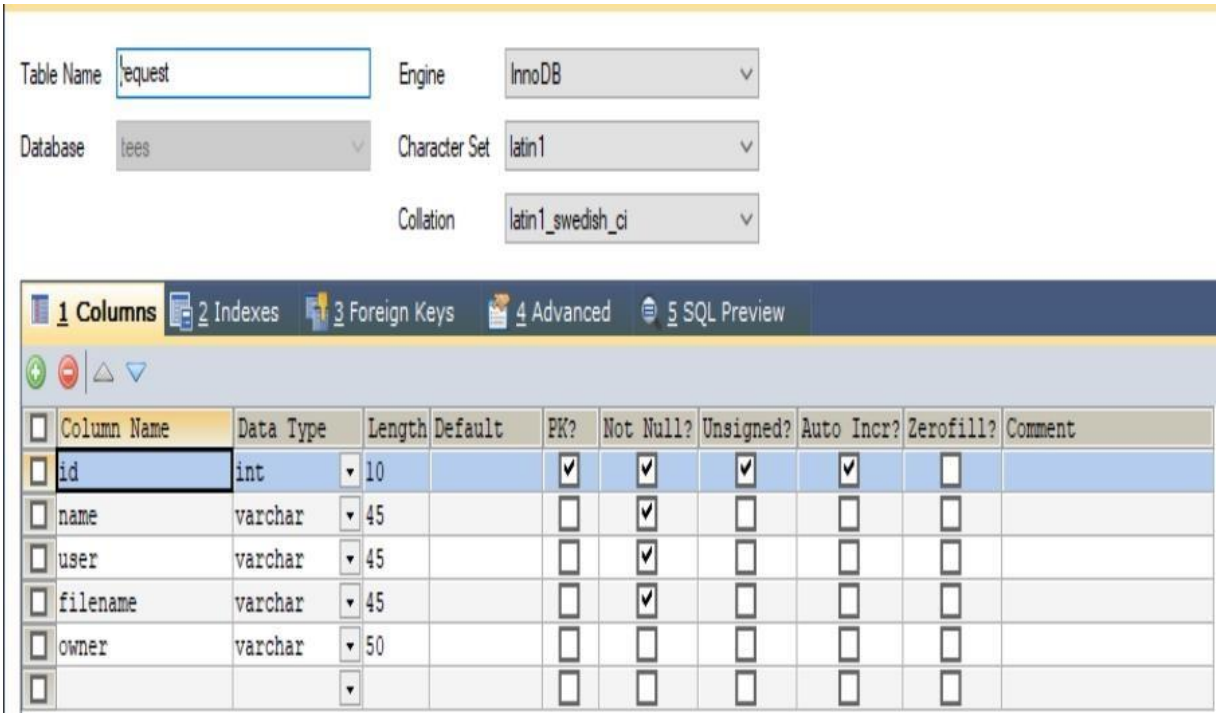


Table Name: request

Database: tees

Engine: InnoDB

Character Set: latin1

Collation: latin1_swedish_ci

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
id	int	10		☑	☑	☑	☑	☐	
name	varchar	45		☐	☑	☐	☐	☐	
user	varchar	45		☐	☑	☐	☐	☐	
filename	varchar	45		☐	☑	☐	☐	☐	
owner	varchar	50		☐	☐	☐	☐	☐	
				☐	☐	☐	☐	☐	

Fig 5.1.4: Request Table

5.2 Evaluation Metrics

- During a retrieval process, the index is consulted for each word or phrase in the query.Also, a ranking is done to select the k-most relevant documents to the query as the resultof the retrieval process.

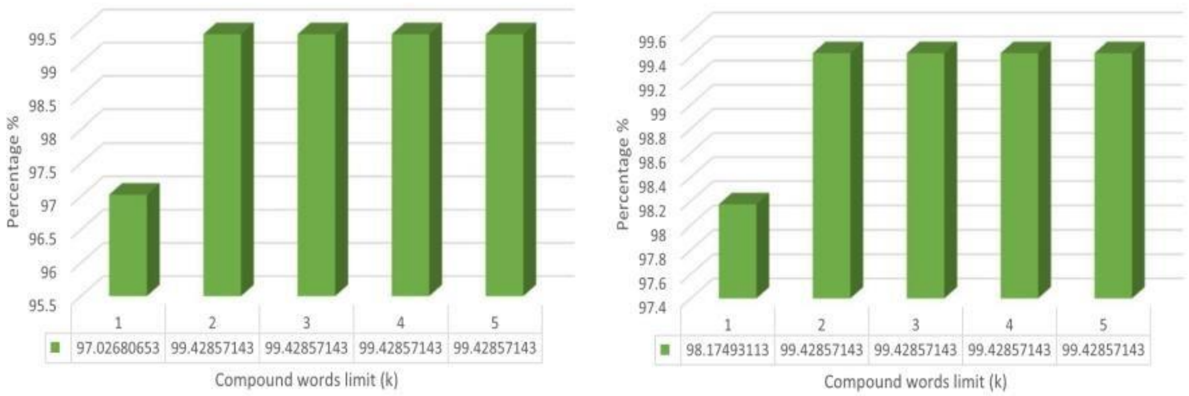
- Let RL be the set of documents relevant to a given query. Let RT the set of documentsretrieved. The documents of interest for users that are both relevant and retrieved are I
  = RL∩ RT . The metrics that assess the quality of text retrieval are:

  Precision = RL ∩

  RT RTRecall = RL

  ∩ RT RL

- We consider two cases in the experimentation for the retrieval task, one on which synonyms (SYN case) are considered during the index creation and the other case whensynonyms are not considered (NO-SYN case).
- The former case causes an increase in the time for building the secure index, the size ofthe index grows and as a consequence, it also increases the searching time.



a) With words synonyms

b) Without words synonyms

Fig 5.2.1: Mean average precision for the retrieval task of FABECS

- We consider two cases in the experimentation for the retrieval task, one on which synonyms (SYN case) are considered during the index creation and the other case whensynonyms are not considered (NO-SYN case).

5.3. Test Cases

Test ID	Test Name	Inputs	Process	Excepted Output	Actual Output	Status

1	Login Test	UserName, Password	Validate the Username and Password on database	Need to Redirect to User home page	It’s redirected to user home	Success
2	Registration Test	Username, password, email, mobile number, etc…	Insert the users into database	Need to Insert the user details into database	It’s inserted	Success
3	Upload file	File, User id.	Insert the files into the cloud	Need to Insert the files into the database	It’s inserted	Success
4	Grant access	User id, Permission	Grant the access to the user to access the files	Need to grant permission to user	permission granted	Success
5	Send Mail	User id, Mail id, key	Send these Inputs to Specified user	Need to send the details to mail id	Mail forwarded	Success
6	Download File	User id, file id, key, permission	If given user having access to files and if he entered.	Download the file if he specified the right details	File downloaded	Success

Chapter 6

CONCLUSION AND FUTURE ENHANCEMENT

Conclusion

We presented for the first time a secure scheme fully based on attribute-based encryption to ensure both, the confidentiality and access control over data outsourced (in encrypted form) by data owners to the cloud and the fine-grained search control for data users when retrieving encrypted data from the cloud; we called this scheme FABECS. Through a formal analysis and experimentation, we proved the correctness and efficacy of FABECS to be used for storing, sharing and retrieval of documents in a cloud-based environment. Furthermore, we provided for the first time Type-III constructions for CP-ABSE and DET-ABE as main building blocks of FABECS. This setting allows using more efficient pairing-friendly curves to achieve recommended security levels, as minimum of 128-bits. These constructions where detailed and their efficacy proved by means of experimentation, over the LISA benchmark for the retrieval task. Further work is focused in the efficiency aspect, as the results presented in this paper did not consider acceleration strategies. For example, parallelization at several levels is possible, besides the scheme is friendly enough to be deployed using parallel patterns such as the manager-worker (for processing a group of attributes at a time) or data encryption (AES on GPUs). Also, as FABECS can be realized with other efficient pairing friendly curves, experimental evaluation could consider the Barreto-Lynn-Scott Curve (BLS) that is also being promoted to be used in practical applications.

EFERENCES

[1] A. Bagherzandi, B. Hore, and S. Mehrotra, Search over Encrypted Data. Boston, MA, USA: Springer, 2011, pp. 1088–1093.

[2] H. Pham, J. Woodworth, and M. A. Salehi, ‘‘Survey on secure search over encrypted data on the cloud,’’ Concurrency Comput. Pract. Exper., vol. 31, p. 1– 15, Apr. 2019.

[3] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, ‘‘Searchable symmetric encryption: Improved definitions and efficient constructions,’’ in Proc. 13th ACM Conf. Comput. Commun. Secur., New York, NY, USA, 2011, pp. 79–88, 2006.

[4] M. Zeng, H.-F. Qian, J. Chen, and K. Zhang, ‘‘Forward secure public key encryption with keyword search for outsourced cloud storage,’’ IEEE Trans. Cloud Comput., early access, Sep. 27, 2019, doi: 10.1109/TCC.2019.2944367.

[5] S. Kamara, C. Papamanthou, and T. Roeder, ‘‘Cs2: A searchable cryptographic cloud storage system,’’ Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-2011-58, May 2011.

[6] W. Song, B. Wang, Q. Wang, Z. Peng, W. Lou, and Y. Cui, ‘‘A privacypreserved full-text retrieval algorithm over encrypted data for cloud storage applications,’’ J. Parallel Distrib. Comput., vol. 99, pp. 14–27, Jan. 2017.

[7] A. G. Kumbhare, Y. Simmhan, and V. Prasanna, ‘‘Designing a secure storage repository for sharing scientific datasets using public clouds,’’ in Proc. 2nd Int. workshop Data Intensive Comput. Clouds, 2011, pp. 31–40.

[8] Z. Yang, J. Tang, and H. Liu, ‘‘Cloud information retrieval: Model description and scheme design,’’ IEEE Access, vol. 6, pp. 15420–15430, 2018.

[9] H. Yin, J. Zhang, Y. Xiong, L. Ou, F. Li, S. Liao, and K. Li, ‘‘CP-ABSE: A ciphertext-policy attribute-based searchable encryption scheme,’’ IEEE Access, vol. 7, pp. 5682–5694, 2019.

[10] A. Sahai and B. Waters, ‘‘Fuzzy identity-based encryption,’’ in Advances in Cryptology (Lecture Notes in Computer Science), vol. 3494, R. Cramer, Ed. Berlin, Germany: Springer-Verlag, 2005, pp. 457–473.

[11] M. Morales-Sandoval, J. L. Gonzalez-Compean, A. Diaz-Perez, and V. J. Sosa-Sosa, ‘‘A pairing-based cryptographic approach for data security in the cloud,’’ Int. J. Inf. Secur., vol. 17, no. 4, pp. 441–461, Aug. 2018.

[12] D. Khader, ‘‘Introduction to attribute based searchable encryption,’’ in Communication Multimedia Security. Berlin, Germany: Springer, pp. 131–135, 2014.

[13] T. Bouabana-Tebibel and A. Kaci, ‘‘Parallel search over encrypted data under attribute based encryption on the cloud computing,’’ Comput. Secur., vol. 54, pp. 77–91, Oct. 2015.

[14] S. Wang, D. Zhao, and Y. Zhang, ‘‘Searchable attribute-based encryption scheme with attribute revocation in cloud storage,’’ PLoS ONE, vol. 12, no. 8, pp. 1–20, Aug. 2017.

[15] Y. Miao, J. Ma, X. Liu, J. Weng, H. Li, and H. Li, ‘‘Lightweight finegrained search over encrypted data in fog computing,’’ IEEE Trans. Services Comput., vol. 12, no. 5, pp. 772–785, Sep. 2019.

[16] H. Wang, X. Dong, Z. Cao, and D. Li, ‘‘Secure and efficient attributebased encryption with keyword search,’’ Comput. J., vol. 61, no. 8, pp. 1133–1142, Aug. 2018.

[17] Mamta and B. B. Gupta, ‘‘An efficient KP design framework of attribute– based searchable encryption for user level revocation in cloud,’’ Concurrency Comput., Pract. Exper., vol. 32, no. 18, p. 5291, Sep. 2020.

[18] H. Yin, Y. Xiong, J. Zhang, L. Ou, S. Liao, and Z. Qin, ‘‘A key-policy searchable attribute-based encryption scheme for efficient keyword search and fine-grained access control over encrypted data,’’ Electronics, vol. 8, no. 3, p. 265, Feb. 2019.

[19] H. Z. Rad, ‘‘Lisa test collection,’’ Sheffield Univ., Sheffield, U.K., Tech. Rep., 2019.

[20] V. Goyal, O. Pandey, A. Sahai, and B. Waters, ‘‘Attribute-based encryption for fine-grained access control of encrypted data,’’ in Proc. 13th ACM Conf. Comput. Commun. Secur., 2006, pp. 89–98.

[21] J. Bethencourt, A. Sahai, and B. Waters, ‘‘Ciphertext-policy attribute-based encryption,’’ in Proc. IEEE Symp. Secur. Privacy, Jul. 2007, pp. 321–334, 2007.

[22] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, ‘‘Privacy-preserving multikeyword ranked search over encrypted cloud data,’’ IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 1, pp. 222–233, Jan. 2014.

[23] S. Wang, J. Ye, and Y. Zhang, ‘‘A keyword searchable attribute-based encryption scheme with attribute update for cloud storage,’’ PLoS ONE, vol. 13, no. 5, pp. 1–19, May 2018.

[24] W. Guo, X. Dong, Z. Cao, and J. Shen, ‘‘Efficient attribute-based searchable encryption on cloud storage,’’ J. Phys., Conf. Ser., vol. 1087, Sep. 2018, Art. no. 052001.

[25] A. Michalas, ‘‘The lord of the shares: Combining attribute-based encryption and searchable encryption for flexible data sharing,’’ in Proc. 34th ACM/SIGAPP Symp. Appl. Comput., Apr. 2019, pp. 146–155.

[26] B. Waters, ‘‘Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization,’’ in Public Key Cryptography. Berlin, Germany: Springer, 2011, pp. 53–70.

[27] Z. Liu, Z. Cao, and S. D. Wong. (2010). Efficient Generation of Linear Secret Sharing Scheme Matrices From Threshold Access Trees. https://eprint.iacr.org/2010/374

[28] D. Boneh, ‘‘Pairing-based cryptography: Past, present, and future,’’ in Advances Cryptoligy. Berlin, Germany: Springer, 2012, p. 1.

[29] S. D. Galbraith, K. G. Paterson, and N. P. Smart, ‘‘Pairings for cryptographers,’’ Discrete Appl. Math., vol. 156, no. 16, pp. 3113–3121, Sep. 2008.

[30] S. Galbraith, (2014). New Discrete Logarithm Records, and the Death of Type 1 Pairings. [Online]. Available: https://ellipticnews. wordpress.com/2014/02/01/new-discrete-logarithm-records-and-thedeath-of-type-1-pairings/

[31] L. Wang, X. Shen, J. Li, J. Shao, and Y. Yang, ‘‘Cryptographic primitives in blockchains,’’ J. Netw. Comput. Appl., vol. 127, pp. 43–58, Feb. 2019.

[32] R. Blum and T. Bocek, ‘‘Superlight – a permissionless, light-client only blockchain with self-contained proofs and bls signatures,’’ in 2019 IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM), pp. 36–41, 2019.

[33] S. L. M. Paulo Barreto and M. Naehrig, ‘‘Pairing-friendly elliptic curves of prime order,’’ in Selected Areas Cryptography. Berlin, Germany: Springer, 2006, pp. 319–331.

[34] Bluekrypt. (2020). Cryptographic Key Length Recommendation. [Online]. Available: https://www.keylength.com/en/

[35] R. Barbulescu and S. Duquesne, ‘‘Updating key size estimations for pairings,’’ J. Cryptol., vol. 32, no. 4, pp. 1298–1336, Oct. 2019.

[36] S. Chatterjee and A. Menezes, ‘‘On cryptographic protocols employing asymmetric pairings the role of ψ revisited,’’ Discrete Appl. Math., vol. 159, no. 13, pp. 1311–1322, 2011.

[37] M. Morales-Sandoval and A. Diaz-Perez, ‘‘DET-ABE: A Java API for data confidentiality and fine-grained access control from attribute based encryption,’’ in Proc. Int. Conf. Inf. Secur. Theory Pract., Heraklion, Greece, Aug. 2015, pp. 104–119.

[38] G. A. Vazquez-Martinez and J. L. Gonzalez-Compean, ‘‘Cloudchain: A novel distribution model for digital products based on supply chain principles,’’ Int. J. Inf. Manage., vol. 39, p. 90–103, Apr. 2018.

[39] J. L. Gonzalez-Compean, O. Telles, I. Lopez-Arevalo, M. MoralesSandoval, V. J. Sosa-Sosa, and J. Carretero, ‘‘A policy-based containerized filter for secure information sharing in organizational environments,’’ Future Gener. Comput. Syst., vol. 95, pp. 430–444, Jun. 2019.

[40] A. De Caro and V. Iovino, ‘‘JPBC: Java pairing based cryptography,’’ in Proc. IEEE Symp. Comput. Commun. (ISCC), Jun. 2011, pp. 850–855.

[41] A. Menezes, P. Sarkar, and S. Singh, ‘‘Challenges with assessing the impact of nfs advances on the security of pairing-based cryptography,’’ in Paradigms Cryptology. Cham, Switzerland: Springer, 2017, pp. 83–108.

APPENDIX

ABSE: attribute based searchable encryption

ABE: attribute-based encryption

CSP: cloud service provider

DO: data owners

SE: Searchable encryption

CP-ABE: cipher text-policy attribute-based encryption

SSE: Symmetric Searchable Encryption

PEKS: public key encryption with Keyword search

FABECS: Fully Attribute-Based Encryption scheme for Cloud Storage, Sharing and Retrieval

PKC: Public key cryptosystems.

INDEXING

INDEXING

INDEXING