ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Secure and Scalable Method to Share Data in Cloud Storage

K.Omsekhar Reddy
PG Student
Department of CSE
MJR College of Engineering & Technology
Chittoor Road, Near Agraharam, Piler, Andhra Pradesh.

Mareddy Swathi
Assistant Professor
Department of CSE
MJR College of Engineering & Technology
Chittoor Road, Near Agraharam, Piler, Andhra Pradesh.

Karamala Suresh
Assistant Professor & HoD
Department of CSE
MJR College of Engineering & Technology
Chittoor Road, Near Agraharam, Piler, Andhra Pradesh.

Abstract

Cloud storage is a model of data storage in which the digital data is stored in logical pools, the physical storage spans multiple servers, and the physical environment is typically owned and managed by a hosting company. These cloud storage providers are responsible for keeping the data available and accessible, and the physical environment protected and running. People and organizations buy or lease storage capacity from the providers to store user, organization, or application data.

The ability of specifically offering scrambled information to diverse clients by means of open distributed storage might extraordinarily ease security worries over unintentional information spills in the cloud. A key test to planning such encryption plans lies in the productive administration of encryption keys. The wanted adaptability of imparting any gathering of chose reports to any gathering of client's requests diverse encryption keys to be utilized for distinctive archives. On the other hand, this additionally infers the need of safely conveying to clients an extensive number of keys for both encryption and seeks, and those clients will need to safely store the got keys, and present a just as substantial number of catchphrase trapdoors to the cloud to perform look over the common information. The inferred requirement for secure correspondence, stockpiling, and multifaceted nature unmistakably renders the methodology unfeasible. In this paper, we address this down to earth issue, which is generally disregarded in the writing, by proposing the novel idea of Single-Key Explore encryption and instantiating the idea through a solid SKEE plan, in which an information proprietor just needs to disseminate a solitary key to a client for sharing countless, and the client just needs to present a solitary trapdoor to the cloud for questioning the mutual reports. The security examination and execution assessment both affirm that our proposed plans are provably secure and for all intents and purposes productive.

Keywords: - *SKEE; data privacy & sharing; cloud storage*

Introduction

Cloud storage has emerged as a promising solution for providing ubiquitous, convenient, and on-demand accesses to large amounts of data shared over the Internet. Today, millions of users are sharing personal data, such as photos and videos, with their friends through social network applications based on cloud storage on a daily basis. Business users are also being attracted by cloud storage due to its numerous benefits, including lower cost, greater agility, and better resource utilization. However, while enjoying the convenience of sharing data via cloud storage, users are also increasingly concerned about inadvertent data leaks in the cloud. Such data leaks, caused by a malicious adversary or a misbehaving cloud operator, can usually lead to serious breaches of personal privacy or business secrets (e.g., the recent high profile incident of celebrity photos being leaked in I Cloud). To address users' concerns over potential data leaks in cloud storage, a common approach is for the data owner to encrypt all the data before uploading them to the cloud, such that later the encrypted data may be retrieved and decrypted by those who have the decryption keys. Such cloud storage is often called the cryptographic cloud storage. However, the encryption of data makes it challenging for users to search and then selectively retrieve only the data containing given keywords. A common solution is to employ a Explore encryption (SE) scheme in which the data owner is required to encrypt potential keywords and upload them to the cloud together with encrypted data, such that, for retrieving data matching a keyword, the user will send the corresponding keyword trapdoor to the cloud for performing search over the encrypted data.

Existing System:

Cloud computing provides users with a convenient mechanism to manage their personal files with the notion called database-as-a-service (DAS). In DAS schemes, a user can outsource his encrypted files to untrusted proxy servers. Proxy servers can perform some functions on the outsourced cipher texts without knowing anything about the original files. Unfortunately, this technique has not been employed extensively. The main reason lies in that users are especially concerned on the confidentiality, integrity and query of the outsourced files as cloud computing is a lot more complicated than the local data storage systems, as the cloud is managed by an untrusted third party. After outsourcing the files to proxy servers, the user will remove them from his local machine. Therefore, how to guarantee the outsourced files are not accessed by the unauthorized users and not modified by proxy servers is an important problem that has been considered in the data storage research community. Furthermore, how to guarantee that an authorized user can query the outsourced files from proxy servers is another concern as the proxy server only maintains the outsourced cipher texts. Consequently, research around these topics grows significantly.

Disadvantages

- Users are especially concerned on the confidentiality, integrity and query of the outsourced files as cloud computing is a lot more complicated than the local data storage systems, as the cloud is managed by an untrusted third party.
- The outsourced files are not accessed by the unauthorized users and not modified by proxy servers is an important problem that has been considered in the data storage research community.

Proposed System:

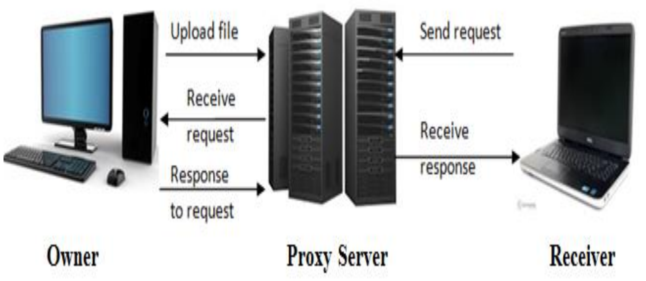
In this paper, we propose two identity-based secure distributed data storage (IBSDDS) schemes in standard model where, for one query, the receiver can only access one of the owner's files, instead of all files. In other words, access permission (re-encryption key) is bound not only to the identity of the receiver but also the file. The access permission can be decided by the owner, instead of the trusted party (PKG). Furthermore, our schemes are secure against the collusion attacks.

Advantages:

- It has two schemes of security, the first scheme is CPA secure, the second scheme achieves CCA security.
- To the best of our knowledge, it is the first IBSDDS schemes where an access permission is made by the owner for an exact file and collusion attacks can be protected in the standard model.

- To achieve a stronger security and implement file based access control, the owner must be online to authenticate requesters and also to generate access permissions for them. Therefore, the owner in our schemes needs do more computations than that in PRE schemes. Although PRE schemes can provide the similar functionalities of our schemes when the owner only has one file, these are not flexible and practical.

Architecture diagram



Design involves identification of classes, their relationships as well as their collaboration. In objectory, classes were divided into Entity classes, interface classes and the control classes. The Computer Aided Software Engineering tools that are available commercially do not provide any assistance in this transition. Even research CASE tools take advantage of Meta modeling are helpful only after the construction of class diagram is completed. In the Fusion method ,it used some object-oriented Class-Responsibility-Collaborator(CRC) and Objectory, used the term Agents to represent some of the hardware and software systems .In Fusion method, there was no requirement phase ,where in a user will supply the initial requirement document. Any software project is worked out by both analyst and designer. The analyst creates the Use case diagram. The designer creates the Class diagram. But the designer can do this only after the analyst has created the Use case diagram. Once the design is over it is need to decide which software is suitable for the application.

IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Implementation is the process of converting a new system design into operation. It is the phase that focuses on user training, site preparation and file conversion for installing a candidate system. The important factor that should be considered here is that the conversion should not disrupt the functioning of the organization.

MODULES:

1. Owner
2. Proxy Server
3. Receiver
4. Storage scheme modules

Data owner:

In this module, first the new data owner registers and then get a valid login credentials. After logged in, the data owner has the permission to upload their file into the Cloud Server. The data owner encrypts his data and outsources the cipher texts to the proxy servers.

Proxy server:

Proxy servers store the encrypted data and transfer the cipher text from the owner to the receiver when they obtain access permission (re-encryption key) from the owner. In these systems, proxy servers are assumed to be trusted. They authenticate receivers and validate access permissions.

Receiver module:

The receiver authenticates himself to the owner and decrypts the re-encrypted Cipher text to obtain the data. An end to-end security is provided by cryptographic protocols which are executed by the file owner to prevent proxy servers and unauthorized users from modifying and accessing the sensitive files. These systems can be divided into two types: shared file system and non-shared system.

Algorithm

Waters's Identity-based Encryption

This identity-based encryption (IBE) works as follows:

Setup. This algorithm takes as input the security parameters

1_, and outputs a bilinear group GG (1_) →

(e, p, G,Gτ) with prime order p, where e : G × G → Gτ .

Let g and η be generators of the group G, u0

R←

G and

U = (u1, u2, · · · , un) where ui

R←

G for i = 1, 2, · · · , n.

It sets g1 = gα where α

R←

Zp. The public parameters

are (e, p,G,Gτ, g, η, u0, g1, U) and the master secret key

is ηα.

KeyGen. Let ID represent an identity which is an n bit

string, IDi be the ith bit of ID, and I be the set which

consists of all i with IDi = 1. This algorithm chooses

r

R←

Zp, and computes

KID,1 = ηα(u0

i∈I

ui)r and KID,2 = gr.

The secret key for the identity ID is SKID =

(KID,1,KID,2).

Encryption. To encrypt a message M ∈ Gτ , this algorithm

chooses s

R←

Zp and computes

C1 = M · e(g1, η)s, C2 = gs and C3 = (u0

i∈I

ui)s.

The ciphertext for the message M is CT = (C1, C2, C3).

Decryption. To decrypt the ciphertext CT = (C1, C2, C3),

this algorithm takes as input the secret key SKID =

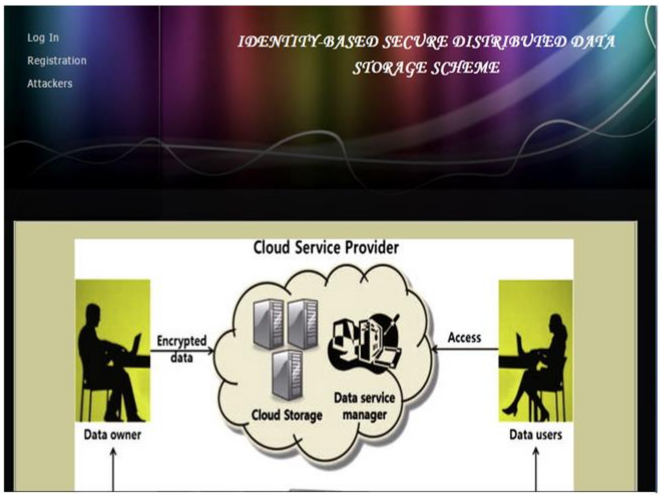
(KID,1,KID,2) and computes

M = C1 ·

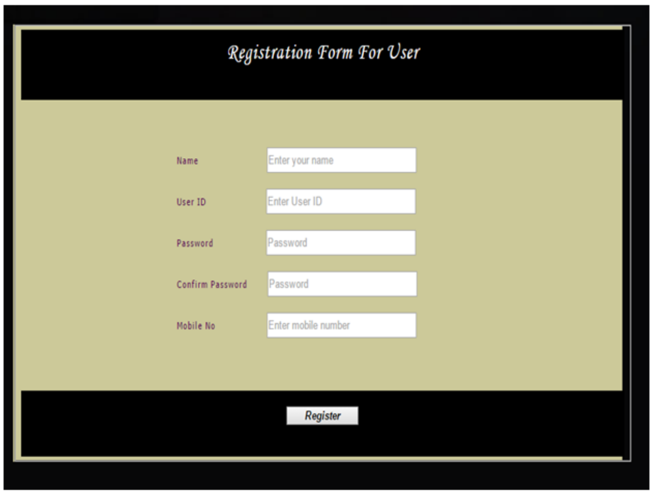
e(KID,2, C3)

e(KID,1, C2)

Screenshots

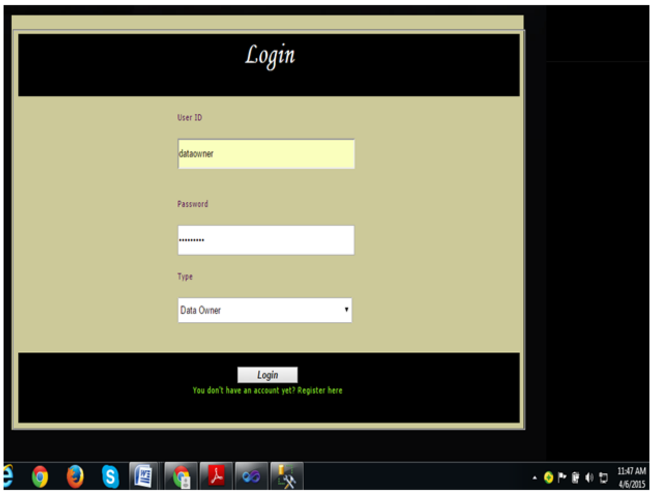


Screen shot 1: Welcome page


Registration Form For User

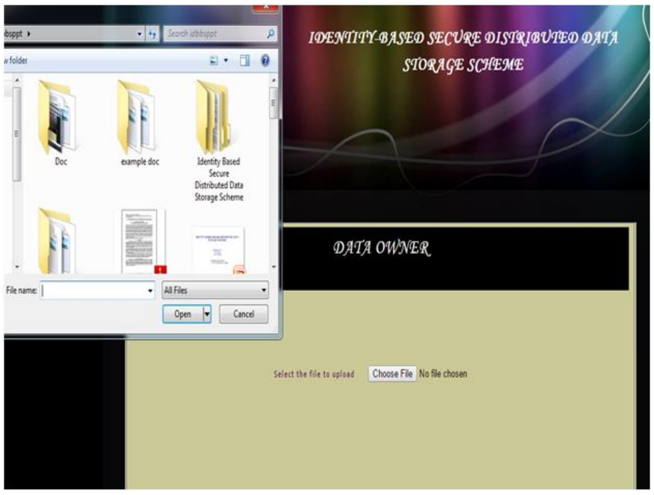
Name: Enter your name
User ID: Enter User ID
Password: Password
Confirm Password: Password
Mobile No: Enter mobile number

Screen shot 2: Registration page


Login

User ID: dataowner
Password: ••••••••••
Type: Data Owner

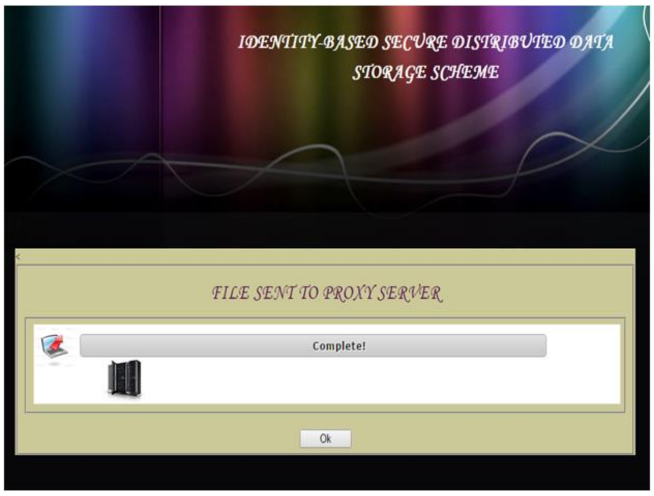
Screenshot 3: Login page


IDENTITY-BASED SECURE DISTRIBUTED DATA STORAGE SCHEME

DATA OWNER

Select the file to upload: Choose File No file chosen

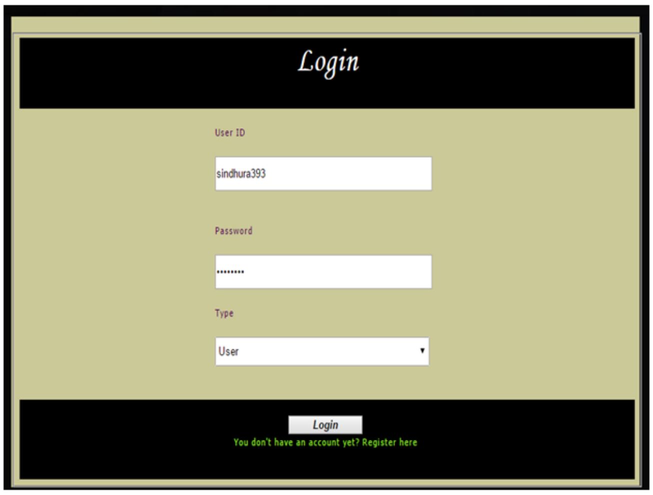
Screen shot 4: Choosing a file from system


IDENTITY-BASED SECURE DISTRIBUTED DATA STORAGE SCHEME

FILE SENT TO PROXY SERVER

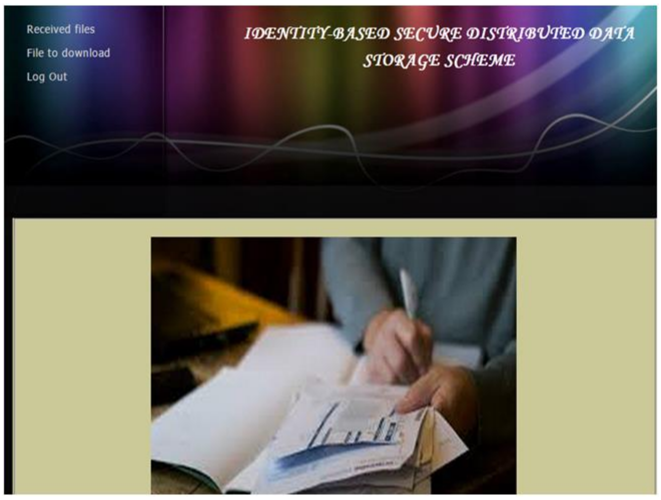
Complete!

Screen shot 5: Uploading file

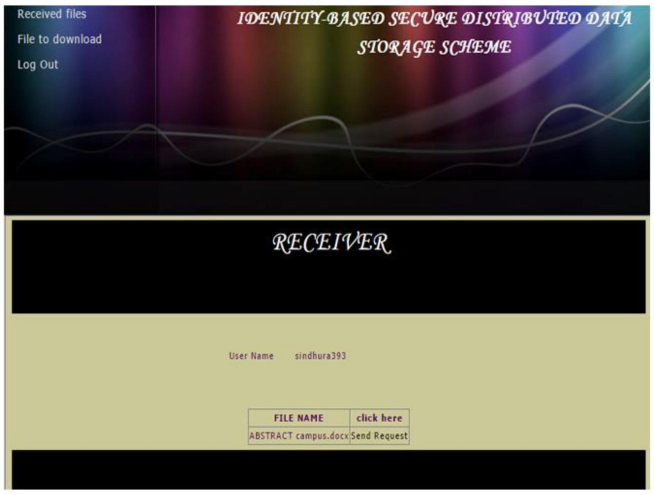

Login

User ID: sindhura393
Password: ••••••••
Type: User

Screen shot 6: login page for user


IDENTITY-BASED SECURE DISTRIBUTED DATA STORAGE SCHEME

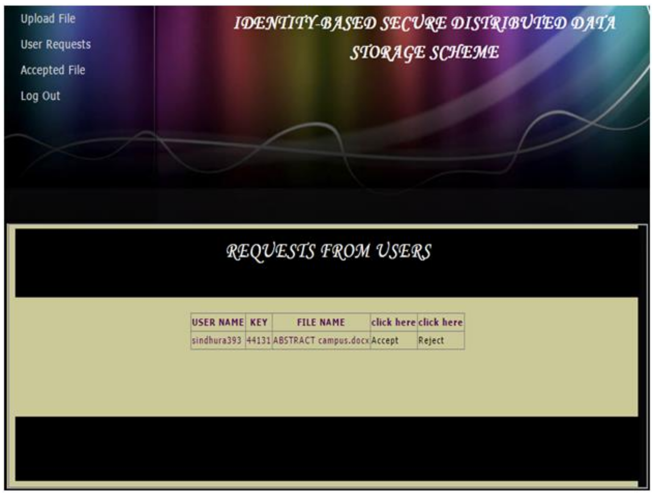
Screenshot 7: Menu page for user



User Name sindhura393

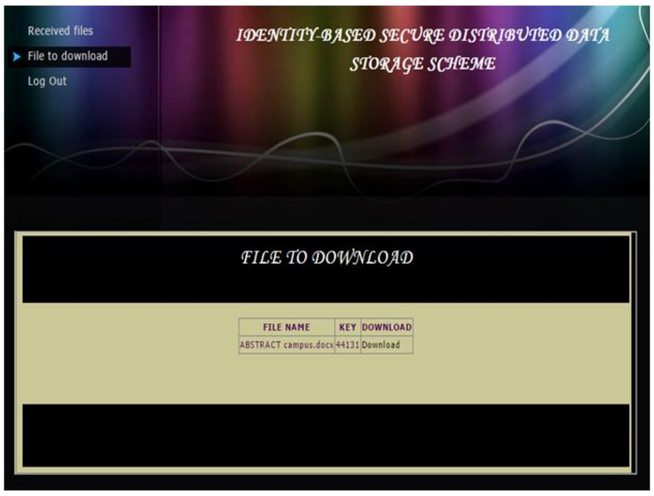
FILE NAME	click here
ABSTRACT campus.docx	Send Request

Screenshot 8: User sending request for file



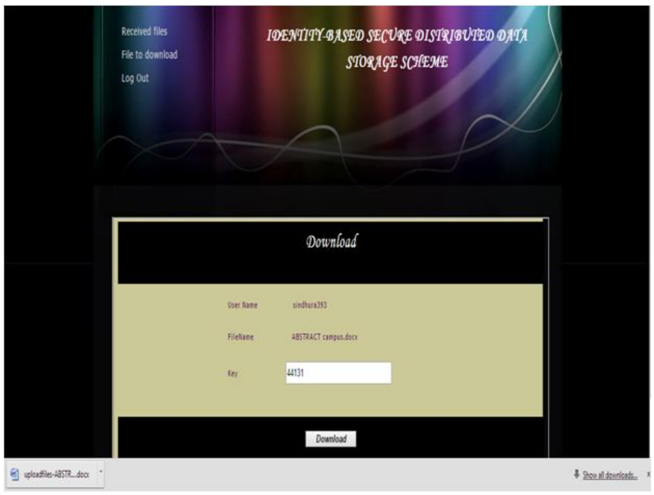
USER NAME	KEY	FILE NAME	click here	click here
sindhura393	44131	ABSTRACT campus.docx	Accept	Reject

Screenshot 9: requests from users



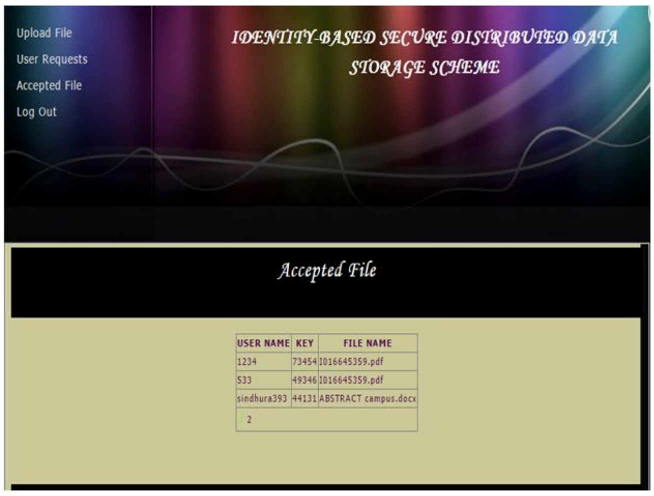
FILE NAME	KEY	DOWNLOAD
ABSTRACT campus.docx	44131	Download

Screen shot 10:File to download for user


Received files
File to download
Log Out
IDENTITY-BASED SECURE DISTRIBUTED DATA STORAGE SCHEME
Download
Download
uploadfiles-ABSTR....docx
Show all downloads...

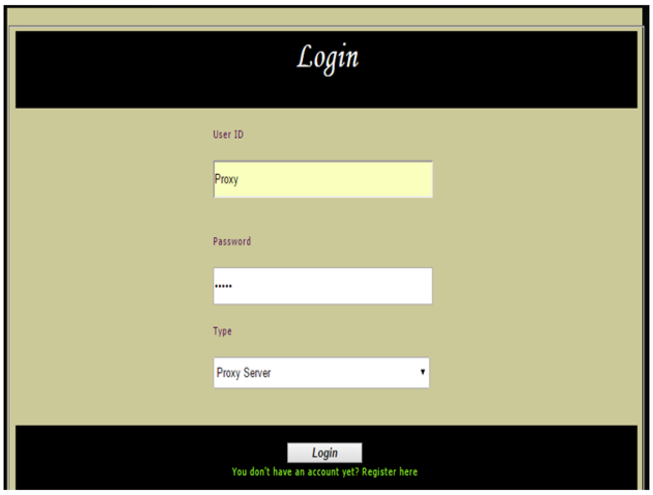
User Name sindhura393
FileName ABSTRACT campus.docx
Key 44131

Screen shot 11: User enters the private key for downloading file



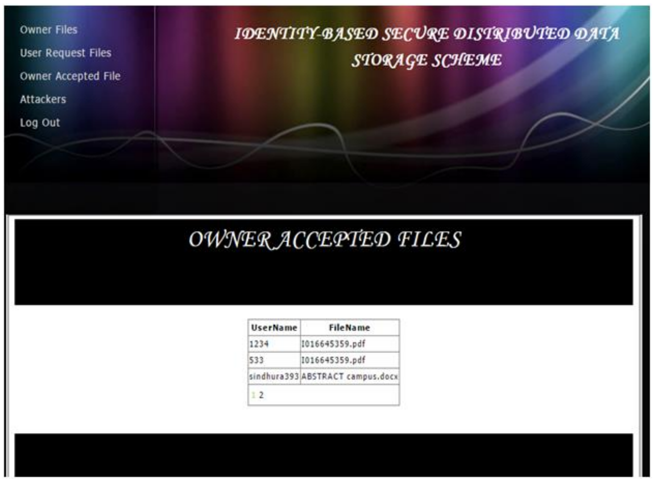
USER NAME	KEY	FILE NAME
1234	73454	I016645359.pdf
533	49346	I016645359.pdf
sindhura393	44131	ABSTRACT campus.docx
2		

Screen shot 12: Owners accepted file list



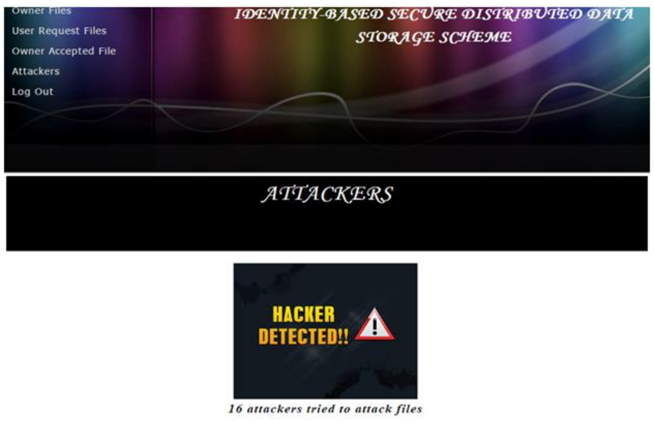
User ID Proxy
Password •••••
Type Proxy Server

Screen shot 13: Proxy login page



UserName	FileName
1234	I016645359.pdf
533	I016645359.pdf
sindhura393	ABSTRACT campus.docx
1 2	

Screen shot14: Details saved by Proxy server.



16 attackers tried to attack files

Screenshot 15: Hackers Detected in Proxy Server

CONCLUSION

Distributed data storage schemes provide the users with convenience to outsource their files to untrusted proxy servers. Identity-based secure distributed data storage (IBSDDS) schemes are a special kind of distributed data storage schemes where users are identified by their identities and can communicate without the need of verifying the public key certificates. In this paper, we proposed two new IBSDDS schemes in standard model where, for one query, the receiver can only access one file, instead of all files. Furthermore, the access permission can be made by the owner, instead of the trusted party. Notably, our schemes are secure against the collusion attacks. The first scheme is CPA secure, while the second one is CCA secure.

Future Enhancements

Our future enhancements for identity-based secure distributed data storage (IBSDDS) are to allow user to upload PDF files and excel sheets. Future research will include advancements like uploading the pictures, images, videos in encrypted format for user convenience.

References

1.H. Hacig¨ um¨ us, B. R. Iyer, C. Li, and S. Mehrotra, “Executing SQL over encrypted data in the database-service-provider model,” in Proceedings: SIGMOD Conference - SIGMOD’02 (M. J. Franklin, B. Moon, and A. Ailamaki, eds.), vol. 2002, (Madison, Wisconsin, USA), pp. 216–227, ACM, Jun. 2002. [2]

2. L. Bouganim and P. Pucheral, “Chip-secured data access: Confi- dential data on untrusted servers,” in Proc. International Conference on Very Large Data Bases - VLDB’02, (Hong Kong, China), pp. 131– 142, Morgan Kaufmann, Aug. 2002.

3. U. Maheshwari, R. Vingralek, and W. Shapiro, “How to build a trusted database system on untrusted storage,” in Proc. Symposium on Operating System Design and Implementation - OSDI’00, (San Diego, California, USA), pp. 135–150, USENIX, Oct. 2000.

4. A. Ivan and Y. Dodis, “Proxy cryptography revisited,” in Proc. Network and Distributed System Security Symposium - NDSS’03, (San Diego, California, USA), pp. 1–20, The Internet Society, Feb. 2003. A. Shamir, “Identity-based cryptosystems and signature scheme,” in Proc. Advances in Cryptology - CRYPTO’84 (G. R. Blakley and D. Chaum, eds.), vol. 196 of Lecture Notes in Computer Science, (Santa Barbara, California, USA), pp. 47–53, Springer, Aug. 1984.

5. D. Boneh and M. Franklin, “Identity-based encryption from the weil pairing,” in Proc. Advances in Cryptology - CRYPTO’01 (J. Kil- ian, ed.), vol. 2139 of Lecture Notes in Computer Science, (Santa

Barbara, California, USA), pp. 213–229, Springer, Aug. 2001.

6. M. Green and G. Ateniese, "Identity-based proxy re-encryption," in Proc. Applied Cryptography and Network Security - ACNS'07 (J. Katz and M. Yung, eds.), vol. 4521 of Lecture Notes in Computer Science, (Zhuhai, China), pp. 288–306, Springer, Jun. 2007.

7. Jinguang Han, Student Member, IEEE, Willy Susilo, Senior Member, IEEE, and Yi Mu, Senior Member, IEEE-"Identity-Based Secure Distributed Data Storage Schemes"-IEEE TRANSACTIONS ON COMPUTERS, 2013.

8.L. Wang, L. Wang, M. Mambo, and E. Okamoto, "New identity- based proxy re-encryption schemes to prevent collusion attacks," in Proc. Pairing-Based Cryptography - Pairing'10 (M. Joye, A. Miyaji, and A. Otsuka, eds.), vol. 6487 of Lecture Notes in Computer Science, (Yamanaka Hot Spring, Japan), pp. 327–346, Springer, Dec. 2010. L. Wang, L. Wang, M. Mambo, and E. Okamoto, "Identity- based proxy cryptosystems with revocability and hierarchical confidentialities," in Proc. International Conference on Information and Communications Security - ICICS'10 (M. Soriano, S. Qing, and J. L´ opez, eds.), vol. 6476 of Lecture Notes in Computer Science, (Barcelona, Spain), pp. 383–440, Springer, Dec. 2010.

9. B. Waters, "Efficient identity-based encryption without ran- dom oracles," in Proc. Advances in Cryptology - EUROCRYPT'05 (R. Cramer, ed.), vol. 3494 of Lecture Notes in Computer Science, (Aarhus, Denmark), pp. 114–127, Springer, May 2005.

10. R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in Proc. Advances in Cryptology - EUROCRYPT'04 (C. Cachin and J. Camenisch, eds.), vol. 3027 of Lecture Notes in Computer Science, (Interlaken, Switzerland), pp. 207–222, Springer, May 2004.

11.D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in Proc. Advances in Cryptology - Crypto'05 (V. Shoup, ed.), vol. 3621 of Lecture Notes in Computer Science, (Santa Barbara, California, USA), pp. 258–275, Springer, Aug. 2005.