A Shared Protocol for Key Management in Encryption based on Encrypted Text Policy Properties to Share Data in the Cloud

G. Muni Nagamani[1], Dr. J. Suneetha[2]

[1]M.Tech Student, Dept of CSE, Siddhartha Institute of Engineering & Technology, Puttur, AP, India
[2]Associate Professor, Dept of CSE, Siddhartha Institute of Engineering & Technology, Puttur, AP, India

Abstract - *Encryption based on encrypted text policy attributes is a promising cryptographic technique for detailed access control of outsourced data in the cloud. However, some inconveniences of key management make it difficult to popularity of the application. A disadvantage in the urgent need for the solution is the key problem of custody. We indicate that the front-end customer devices such as smart phones generally have limited privacy protection, so if the private keys are entirely in their hands, customers risk key exposure that is barely noticeable, but that inherently exists in previous investigation. In addition, huge decryption of the client overload limits the practical use of Attribute Based Encryption. In this work, we propose a collaborative protocol for key management in Ciphertext Policy – Attribute Based Encryption. Our construction is distributed generation, issuance and storage of private keys without adding any additional infrastructure. An attribute of fine and immediate grain revocation is provided for the key update. The proposed collaboration mechanism effectively resolves not only the key problem of escrow, but also key exposure. Meanwhile, it helps sharply reduce the customer decryption overload. A comparison with another representative Ciphertext Policy Attribute Based Encryption schemes show that our scheme has better performance in terms of outsourced data based on the cloud sharing on mobile devices. Finally, this scheme provides security proof for the proposed protocol.*

***Key Words*: Key Management, Cloud data sharing, Security, Privacy**

1. INTRODUCTION

With improved enhancements in computing technology and large-scale networks, data sharing with others becomes simpler. In addition, digital resources are easily accessed by cloud computing and storage. Since the cloud database acquisition of offset infrastructure is needed, some organizations are jointly organized, threatening any privacy of remote storage database owners, so personal, confidentiality stored in the cloud And protection of serious data is extremely important. Data sharing requires fine grained access control with a large number of users participating. Feature-based encryption is a passionate cryptographic primitive which offers an interesting solution to sharing safe and flexible data. Attribute Based Encryption (ABE) is more than one casting property, which means that a key can wipe different ciphertexts or different keys can reject the same aspirant text. There are two types of ABE, in which the Cipher Text Policy ABE (CP-ABE) and Key Policy ABE (KP-ABE). For CP-ABE, the access policy is embedded in a ciphertext and the feature set is embedded in a personal key. For KP-ABE, access policy is embedded in a private key and feature set is included in a secret form. CP-ABE allows database owners to define their own access policy. Anyone who wants to get data first must first meet the required access policy set. Due to this property, CP-ABE is quite suitable to build secure, reflexive access control for sharing cloud data.

However, there are still many open challenges regarding ABE's practical feelings, especially in private terms of key management. For a large number of previous ABE projects, key authority must be fully trustworthy, because it can waste all the use of the ciphertext generated private key without the permission of its owner. This is the place where the main Escrow problem is usually called and is a reasonable damage that threatens user's privacy. Together Mobile applications, mobile cloud services development in cloud computing has been introduced as a potential trend. Current research work is also difficult to mobile front end Devices, such as smart phones, are far more dangerous servers with respect to privacy protection. Such private key protection can be weakened easily by displaying keys to unauthorized users. Besides that, the current ABE key management schemes also need more bilinear pair accounting, exponentiation and multiplication, especially in the depression phase Resulting runtime possibly unacceptable.

In this paper, we propose a new collaborative key management protocol in Cipher text Policy Attributes-based Encryption (CKM-CP-ABE), aimed at improving the security and efficiency of key management in cloud data sharing system. The most important contributions are summarized as follows:

1) A new advanced protocol is presented. With the help of the interaction between the key authority, a cloud server and a client with access to data, distributed generation, issue and storage of private keys are realized. Thus, secure key management is guaranteed without adding additional physical infrastructure, which is easier to distribute than previous multisystem.

2) We present attribute groups to build the private key update algorithm. A unique attribute group's key is assigned to each attribute group containing clients sharing the same attribute. Via the update of the attribute group's key, a fine-grained and immediate attribute recall is offered.

3) We indicate that not only the key escrow issue but also key exposure threatens the privacy of private keys, which is not noticed in previous research. Compared with previous key management protocols for attribute based data sharing in cloud, our proposed protocol effectively addresses both issues using collaboration key management. Finally, we provide proof of security for the proposed protocol.

4) The collaborative mechanism significantly helps reduce client encryption overhead by using a decryption server to perform most decryption, but leaves no knowledge of information about it.

2. EXISTING SYSTEM

The key authority must be completely reliable as it can decrypt all coding texts using one generated private key without permission from the data owner. This is usually called the key entry problem and is a disadvantage. Mobile front-end devices, such as smart phones, are far more vulnerable to privacy protection servers. Thus, the vulnerability of private key protection can easily lead to the exposure of keys to unauthorized users. In addition, the current ABE key management systems also require lots of automotive calculation, exposure and multiplication, especially in decryption. The resulting runtime can be horribly unacceptable.

2.1. Disadvantages

1. Security problems and key exposure chances are there.
2. The time required for decryption takes large value.

3. PROPOSED SYSTEM

We propose a new collaborative key management protocol in encoding text policy Attribute Based Encryption (CKM-CP-ABE) aimed at increasing the security and efficiency of key management in cloud data sharing system

1) A new cooperation protocol is presented. With the help of the secure key management is guaranteed that is easier to distribute compared to previous multi-authorization arrangements.

2) A unique attribute group's key is assigned to each attribute group containing clients like sharing the same attribute. Via update of the attribute group's key, a fine-grained and immediate Attribute recall is given.

3) The collaborative mechanism significantly helps reducing the client encryption overhead hire a decryption server to perform most decryption while leaving no knowledge of information to it.

3.1. Advantages

1. Secure key management and unique group key generation provided so that security problems will not be there.

2. For reducing the decryption time overhead we have proposed special methods here.

4. SYSTEM ARCHITECTURE

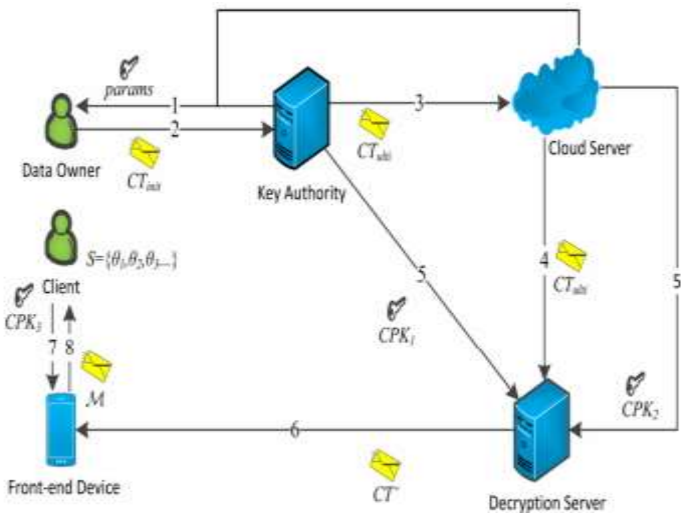

Fig -1 System Architecture

In this shared protocol for key management in encryption based on encrypted text policy attributes, there are mainly five modules as represented below:

- Client
- Key Authority
- Encryption
- Decryption Server
- Data Owner

The brief explanation of the above modules is as follows.

A client is the user who intends to access the data in the cloud storage via front-end devices. If the client's attribute set satisfies an access policy, then only the client will be allowed to access the plaintext.

A data owner is an authorized person in the system that possesses the data to be uploaded.

The key authority is responsible for most calculating tasks, including key generation, key update etc. Key Authority is the main component in the system and the generation of the private keys and updations of the private keys is the main responsibility of the key authority component in the system.

Encryption corresponds to the process of encoding a message or information in such a way that only authorized people can access it and those who are not authorized cannot. This encryption is the most basic thing which is used for the security of the information.

Decryption server is used to reduce the clients decryption overhead while execution without leaving any information to it. Decryption server has more powerful capabilities. It guarantees the data security.

Data owner is the authorized user in the system that possesses the data to be uploaded into the cloud. Data owner can define their own explicit policies so that their desirable clients are granted permission to access the plaintext.

Data owner first sends the request to multiple key authorities for uploading the file with high security. After sending the request gets keys from multiple key authorities to upload files. Based on the keys, the data owner uploads the files in the cloud storage with high data security. If the data user wants to access the files, the data user sends the request to the data owner. If the client's attribute set satisfies an access policy associated with the cipher text, the client will be allowed to get the keys from the data owner. Finally, by using the keys, the client can able to access the secured files that are in the cloud.

5. IMPLEMENTATION

Here, we offer a secured algorithm called Key Management Algorithm for the high data security for the files uploaded in the cloud. The description and the steps will be as follows.

Key Management Algorithm

The CP-ABE scheme consists of six algorithms whose details are as follows.

A. *Setup*

The setup algorithm takes as input the security parameter and it returns a group of public parameters. It consists of the following three steps:

1) *TrustSetup* $(1k, L)$:

The trust launcher selects a group of random elements and each element corresponds with an authorized attribute. The trust launcher then outputs the public parameter.

2) *KASetup* :

The key authority chooses a random exponent a master secret and computes a public parameter.

3) *CSSetup:*

The cloud server chooses a random exponent and then it computes the secret or public parameter pair.

For our scheme, we maintain the trust setup algorithm run by a trust launcher to keep theoretical correctness.

B. Key Generation

The Key Authority runs the key generation algorithm by taking the input as the public parameter and clients attribute set. The initial key is kept by the KA for the subsequent private key updates.

C. Encryption

The Data Owner runs the encryption algorithm by taking the input as public parameters, an access structure and a plain text. Data owner generates a random vector and a linear secret sharing scheme associated with access structure. And finally the data owner outputs the initial ciphertext.

D. Re-encryption:

Cloud server is responsible for running this re-encryption algorithm. It takes the input as public parameters, initial cipher text and the collection of attribute groups. We are using the group-based algorithm to re-encrypt the cipher text. After receiving the cipher text, the cloud server selects two random exponents and generates set of attribute group keys, where each client has a unique constant identity code.

The ultimate cipher text is stored in the cloud. Once an access query is received, the cloud server will launch retrieval. The main goal of the re-encryption algorithm is realizing the effective attribute revocation.

E. Private Key Update

The private key update algorithm, which is the principal invocation of the cipher text policy scheme, which takes the input as parameters and initial keys. For this algorithm, the collaborative key management protocol is implemented to generate and distribute three different private key components. The protocol consists of two sub protocols.

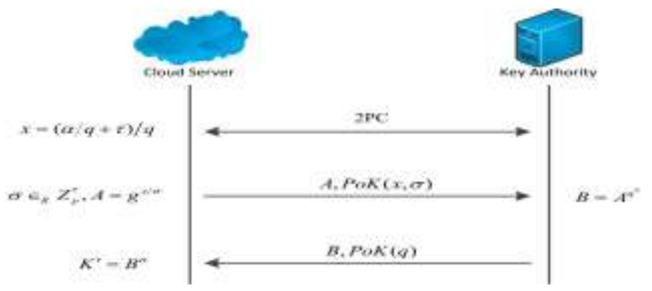

Fig - 2 The first sub-protocol flows

According to the two party computations, the sub-protocol involves the Key Authority and Cloud Server in the following simple arithmetic computation.

1) The KA chooses a consistent random exponent $\tau \in_R Z^*_p$.
2) The CS and KA engage in a secure 2PC, in which CS's Input is α and KA's input is (q,τ) . The 2PC returns a private output $x = (\alpha/ q + \tau)/ q$ to CS.
3) The CS chooses a random $\sigma \in_R Z^*_p$ and returns $A = g^{x/\sigma}$ to the KA.
4) The KA computes $B = A^{q2}$ and sends it to the CS.
5) The CS obtains an initial key component by computing $K' = B^{\sigma} = g^{\alpha+q\tau}$.

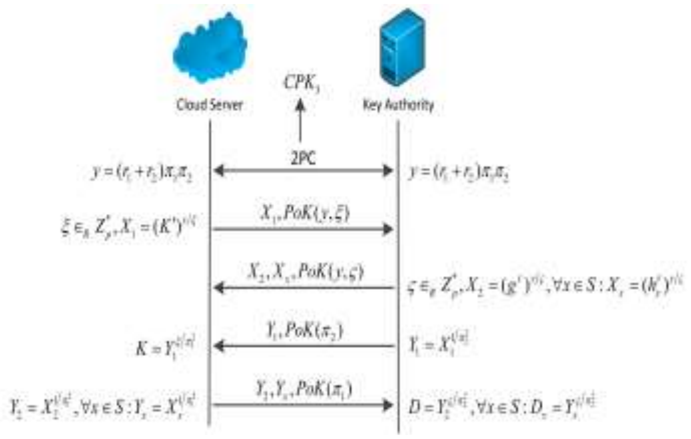
Cloud Server
Key Authority
CPK_1
2PC
$y=(r_1+r_2)\pi_1\pi_2$
$y=(r_1+r_2)\pi_1\pi_2$
$X_1, PoK(y,\xi)$
$\xi \in_R Z^*_p, X_1=(K')^{y/\xi}$
$X_2, X_x, PoK(y,\varsigma)$
$\varsigma \in_R Z^*_p, X_2=(g^\tau)^{y/\varsigma}, \forall x \in S: X_x=(h^\tau_x)^{y/\varsigma}$
$Y_1, PoK(\pi_2)$
$K=Y_1^{\xi/\pi_1^2}$
$Y_1=X_1^{1/\pi_2^2}$
$Y_2, Y_x, PoK(\pi_1)$
$Y_2=X_2^{1/\pi_1^2}, \forall x \in S: Y_x=X_x^{1/\pi_1^2}$
$D=Y_2^{\varsigma/\pi_2^2}, \forall x \in S: D_x=Y_x^{\varsigma/\pi_2^2}$

Fig – 3 The second sub-protocol flows

After the process of first sub-protocol flows, the Cloud Server holds k' and the Key Authority possesses the initial key $PK_{init.}$ These two facts are crucial to the proposed second sub-protocol that involves the KA and CS and also the CL. The computation process of the second sub-protocol flows are as follows.

1) The CS randomly selects r_1, $\pi_1 \in_R Z^*_p$ as its private input, and the Key Authority randomly selects r_2, $\pi_2 \in_R Z^*_p$ as its private input.

2) The protocol returns private output $y = (r_1+r_2)\ \pi_1\pi_2$ to both KA and CS.

3) The CS selects random exponent $\xi \in_R Z^*_p$ and returns $X = (K')^{y/\xi}$ to the KA.

4) The KA returns $Y_1 = X_1^{1/\pi 2_2}$ to the CS.

5) The CS outputs $K = Y_1^{\xi/\pi 2_1}$ as its component of the private key.

6) The KA selects random exponent $\varsigma \in_R Z^*_p$ and returns $X_2 = (g^\tau)^{y/\varsigma}$ and $\forall x \in S$: $X_x = (h^\tau_x)^{y/\varsigma}$ to the CS.

7) The CS returns $Y_2 = X_2^{1/\pi 2_1}$ and $\forall x \in S$: $Y_x = X_x^{1/\pi 2_1}$ to the KA.

8) The KA outputs $D = Y_2^{\varsigma/\pi 2_2}$ and $\forall x \in S$: $D_x = Y_x^{\varsigma/\pi 2_2}$.

9) The protocol secretly sends $(r_1+r_2)/ \pi_1\pi_2$ to the client as its component of the private key.

The components of the private key are:

$CPK_1 = (K = (g^{\alpha+q\tau})^{(r1+r2)/\pi1\pi2})$

$CPK_2 = (D = (g^\tau)^{(r1+r2)/\pi1\pi2}, \forall x \in S: D_x = (h^\tau_x)^{(r1+r2)/\pi1\pi2})$

$CPK_3 = (r_1+r_2)/ \pi1\pi2$

All private key component generation and transmission is executed under the collaborative key management protocol. Neither KA nor Client can decrypt ciphertext on their own due to the proposed key management protocol. Thus, this proposed scheme helps in preventing key exposure along with key escrow problem.

F. Decryption

The decryption algorithm takes the input as clients attribute set and the ultimate ciphertext and also all the private key components i.e., CPK_1, CPK_2, $CPK_{3.}$ The Key authority and cloud server send their private key components to the decryption server after receiving the decryption query from the client.

6. RESULTS

This section represents the experimental results of the application.

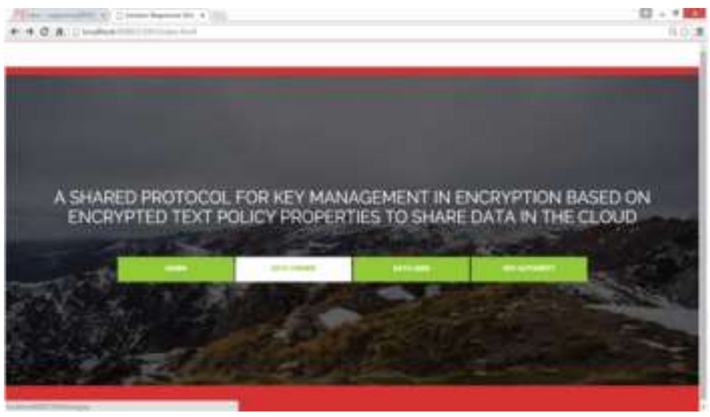
A SHARED PROTOCOL FOR KEY MANAGEMENT IN ENCRYPTION BASED ON ENCRYPTED TEXT POLICY PROPERTIES TO SHARE DATA IN THE CLOUD

Fig – 4 Home Page

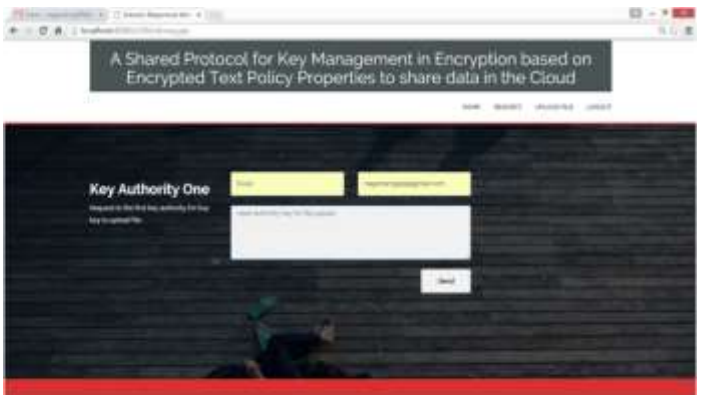
A Shared Protocol for Key Management in Encryption based on Encrypted Text Policy Properties to share data in the Cloud
Key Authority One
Send

Fig – 5 Data owner sending the request to the Key Authority

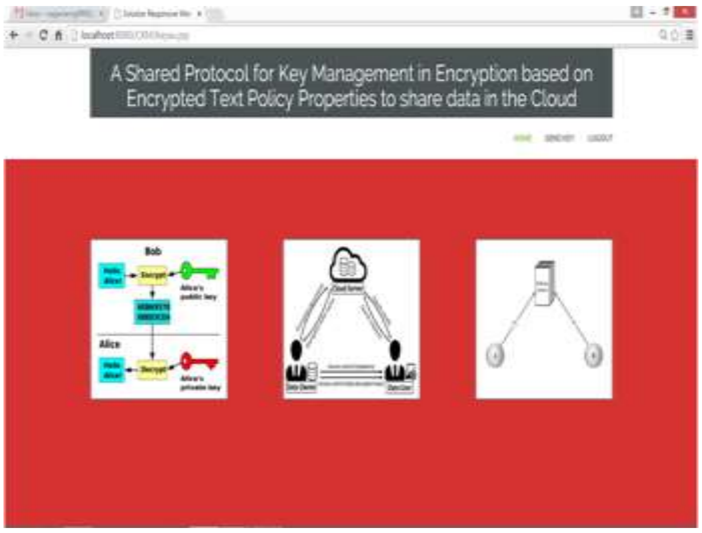
A Shared Protocol for Key Management in Encryption based on Encrypted Text Policy Properties to share data in the Cloud
Bob
Alice

Fig – 6 Key Authority Home Page

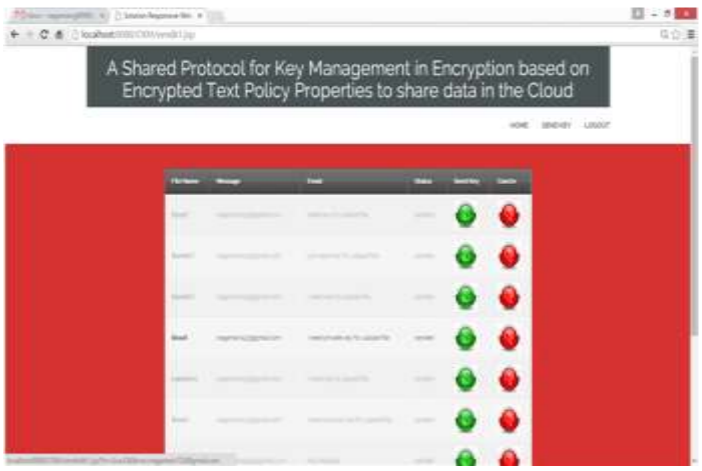
A Shared Protocol for Key Management in Encryption based on Encrypted Text Policy Properties to share data in the Cloud

Fig – 7 Key Authority accepting/rejecting the request

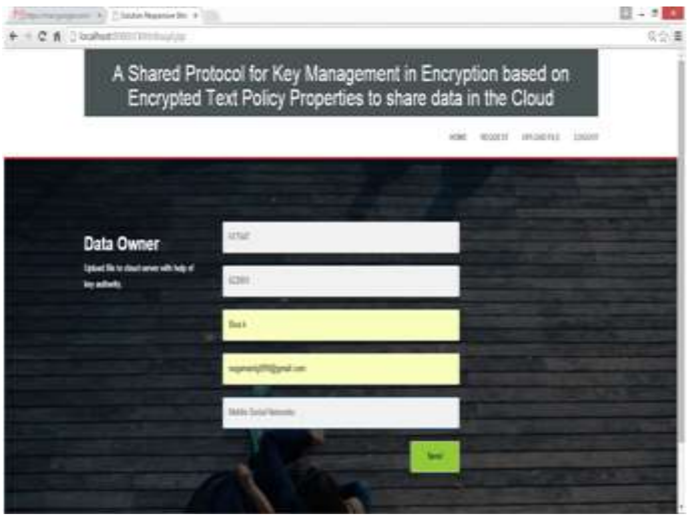
A Shared Protocol for Key Management in Encryption based on Encrypted Text Policy Properties to share data in the Cloud
Data Owner
Send

Fig – 8 Data owner uploading the file page

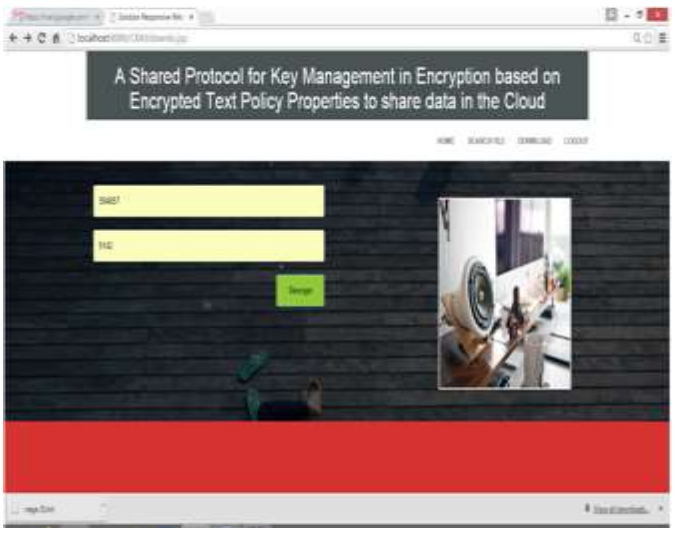
A Shared Protocol for Key Management in Encryption based on Encrypted Text Policy Properties to share data in the Cloud

Fig – 9 Decrypting the file using keys

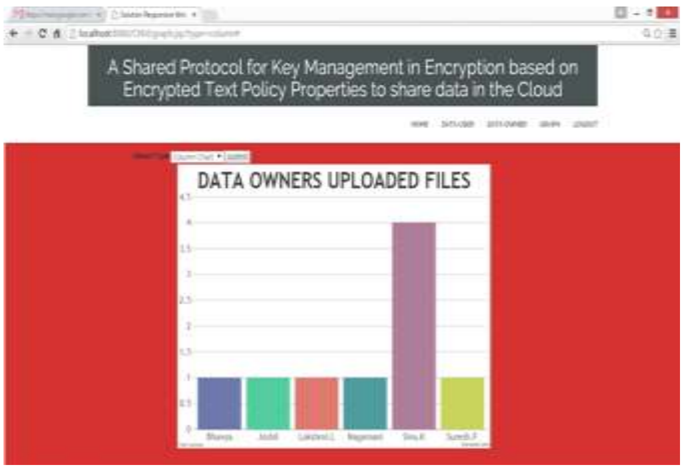

Fig – 10 Column chart representing the uploaded files

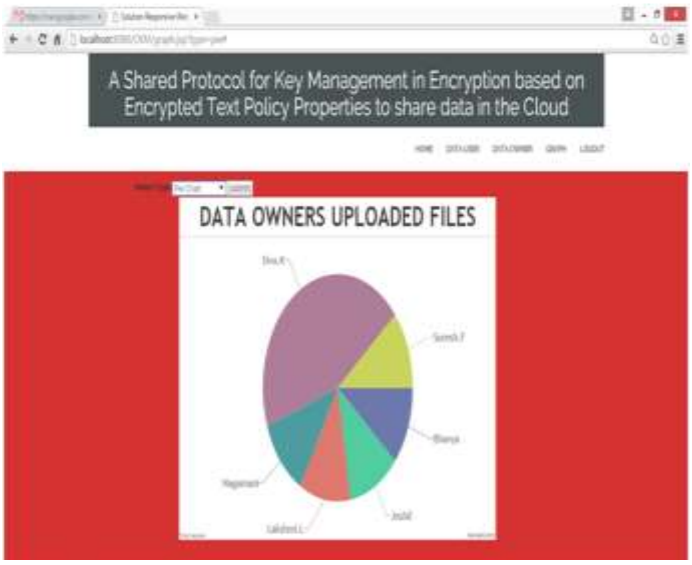
A Shared Protocol for Key Management in Encryption based on Encrypted Text Policy Properties to share data in the Cloud
DATA OWNERS UPLOADED FILES
Siva.K
Suresh.T
Bhanya
Jesfal
Lakshmi.L
Nagamani

Fig – 11 Pie chart representing the uploaded files

7. CONCLUSION

Cipher text policy attribute-based encryption is a promising cryptographic technique to realize fine-grained access control in secure cloud storage. In this paper, we propose a novel collaborative key management protocol to enhance both security and efficiency of key management in cipher text policy attribute-based encryption for cloud data sharing system. Distributed key generation, issue and storage of private keys are realized without adding any extra physical infrastructure. We introduce attribute groups to build a private key update algorithm for fine-grained and immediate attribute revocation. The proposed collaborative mechanism perfectly addresses not only key escrow problem but also a worse problem called key decryption overhead.

8. REFERENCES

[1] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. ACM CCS, 2006, pp. 89-98.

[2] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," IEEE Trans. Comput., vol. 62, no. 2, pp. 362-375, 2013.

[3] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 1, pp. 131-143, 2013.

[4] J. Bethencourt, A. Sahai, and B.Waters, "Ciphertext-policy attribute-based encryption," in Proc. IEEE Symp. Secur. Privacy, 2007, pp. 321-334.

[5] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian, "Mobile cloud computing: a survey, state of art and future directions," Mobile Networks and Applications, vol. 19, no. 2, pp. 133-143, 2014.

[6] M. S. Ahmad, N. E. Musa, R. Nadarajah, R. Hassan, and N. E. Othman, "Comparison between android and iOS operating system in terms of security," in Proc. CITA, 2013, pp. 1-4.

[7] H. Hong, Z. Sun, "High efficient key-insulated attribute based encryption scheme without bilinear pairing operation," SpringerPlus, vol. 5, no. 1, pp. 131, 2016.

[8] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the Internet of Things," Future Generation Computer Systems, vol. 49, pp. 104-112, 2015.

[9] G. Zhang, L. Liu, and Y. Liu, "An attribute-based encryption scheme secure against malicious KGC," in Proc. TRUSTCOM, 2012, pp. 1376-1380.

[10] J. Hur, "Improving security and efficiency in attribute-based data sharing," IEEE Trans. Knowl. Data. Eng., vol. 25, no. 10, pp. 2271-2282, 2013.

[11] P. P. Chandar, D. Mutkurman, and M. Rathinrai, "Hierarchical attribute based proxy reencryption access control in cloud computing," in Proc. ICCPCT, 2014, pp. 1565-1570.

[12] X. A. Wang, J. Ma, and F. Xhafa, "Outsourcing decryption of attribute based encryption with energy efficiency," in Proc. 3PGCIC, 2015, pp. 444-448.

[13] L. Cheung, and C. Newport, "Provably secure ciphertext policy ABE," in Proc. ACM CCS, 2007, pp. 456-465.

[14] J. Hur, and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 7, pp. 1214-1221, 2011.