Cloud Key Depository: A Novel Key Management Framework for Owner Authorization and Privacy Preservation Over The Cloud

P. M. Gund[1], Prof. S.M.Rokade[2]
[1, 2] Department of Computer Engineering
[1, 2] PREC, Loni, Maharashtra.

Abstract- *CloudKey depository is specifically designed for management of keys. In modern era, there is huge growth in the use of web app. To get access in web applications, there is need of user registration and login. While registering on web application user get password or secret key as privacy parameters for unique user. Key or password management is very hectic task as a single user may have multiple accounts for every single web app for various purposes. Similar to data outsourcing, users also outsourced keys or passwords to cloud server. Therefore, we proposed cloudkey Depository framework which can efficiently manages keys using two protocols namely, DepositeKey protocol and WithdrawKey protocol. Traditional approaches of key management does not fulfills the requirements such as, , privacy & confidentiality of keys, search privacy on identity attributes tied to keys and owner controllable authorization over his shared keys. In proposed approach SC-PRE scheme is utilized. It combines HVE and PRE techniques for better implementation of Cloudkey Depository. Along with key management, email search over an encrypted data is provided.*

Keywords- SC-PRE, search privacy, key management, keys outsourcing

I. INTRODUCTION

There is explosive growth in the use of web based applications. Encryption keys or password required to access web based applications are then outsourced to cloud server. Keys or passwords are outsourced to cloud server for heavy management of them. There are several web applications exists for various purposes such as, online banking, shopping, social networks and data storage. Previously, password managers are provided to manage passwords and encryption keys such as, Lastpass and PasswordBox, it is used by millions of users over three years in 2013. Also other similar tools used to outsource their passwords to centralized key management for relieving them from crushing the burden of memorization and management. Privacy problem is the main concern of cloud users in outsourcing storage of data. It also consists of key outsourcing with outsourcing the data. There are mainly two situations such as, 1. Users do not fully trust on CSP due no longer control about how keys are used by them and whether the key owner actually control their keys on their own and 2. They may trust on CSP, keys could be disclosed if there exists misbehaving internal employee or corrupt server. Hence, key tuples are encrypted similar like normal data encryption. Data outsourcing is the promising solution to maintain trust and ensuring key ower's control over their own privacy. Privacy requirements of key attributes in the group of key attribute is higher than that of identity attribute in search attribute group because resulted information leak by former is much higher than that by latter. Therefore, based on different sensitive attributes in key tuples, proposed system identifies critical issues that are given below:

1. Keys having high sensitivity need to be hidden from service providers and malicious attackers. It involves confidentiality and privacy keys i.e. only authorized users can derive shared keys of key owner through authorized decryption computation.
2. Keys are saved with multiple sensitive identity attributes of key owners and searched based on them. It contains search privacy and identity attributes.
3. Keys have strong ownership because they preserve many other sensitive information of other key owner. It contains owner controllable authorization including authorization of keys and authorization of query.

Traditional approaches of key outsourcing and management just support one or two identified security requirements to some extent. Key tuple encryption similar to data tuples provides key privacy and confidentiality. But it does not consider authorization of keys and different privacy requirements of sensitive attributes in key tuples. Search keyword encryption based on identity attributes of key tuples based on symmetric encryption (SSE) or hierarchical predicate encryption provides guarantee of search privacy on key tuples. To efficiently solve identified secure problems, we proposed Cloudkey Depository. It is unified framework for key

management. It enforced privacy and owner controllable authorization.

II. RELATED WORK

SQL queries over an encrypted data have been introduced in [1]. Proposed mechanism processes much of SQL queries without data decryption. The remainder required for query processing as well as data decryption is placed at client side. "Coarse index" deploys the partial execution of SQL query on the provider side. Service provider retains the responsibility to handle constancy of the data. Therefore, client does not need to manage persistency of data. To guarantee privacy and access authorization of outsourced data, data owners employ different cryptographic techniques to encrypt data so as to implement different goals of privacy protection. An experimental result only guarantees the confidentiality and privacy of keys but does not consider the key authorization and the different privacy requirements of sensitive attributes in key tuples. Challenging issue of access control management by the database service providers (DSP) discussed in [2]. In flexible access control enforcement management by applying a DSP re-encryption mechanism user does not required many computations. User can use their private key to decrypt all authorized data tuples. Dynamic policies are update and manage whenever a revocation operation takes places. Moreover, a new architecture satisfies the secure performance of the confidentiality and can reduce the computation complexity of the client by eliminating the public catalog of tokens. Subsequent researches under the new architecture are on designing of the efficient query transformation function in the client, the additional mechanisms such as the integrity and query guarantee in the DSP and the efficient implementation of role-based and the secret share based DSP re-encryption mechanisms. C-PRE scheme is conditional proxy re-encryption in which only ciphertext satisfying a specific condition set by one user can transform to the proxy, it is then decrypted by another user. C-PRE method is extended to MC-PRE which satisfies multiple conditions and supports AND gates on conditions.

APKS is authorized private searches over an encrypted PHR's in the cloud computing. In this, multiple PHR owners encrypt their health records along with a keyword index to allow searches by multiple users in the public domain. To restrain exposure of sensitive patient health information from unrestricted query capabilities a scalable, fine-grained authorization framework is proposed in which user access their search capabilities from local trusted authorities according to their "eligible attributes". Two approaches have been proposed as solutions for APKS over encrypted PHR based on HPE. The proposed solutions also enjoy the advantages of supporting multi-dimensional multi-keyword query, search capability delegation and efficient revocation.

Two layers of encryption which gives assure data confidentiality from cloud and delegate the enforcement of fine-grained access control to the cloud is discussed by M. Nabeel et al [5]. Challenging issue of decomposing access control policies i.e. ACP is introduced by them. Cloud security and privacy are the major concerns. Encryption assures the confidentiality of data against cloud. Existing techniques of encryption not comfortable to support the enforcement of fine-grained organizational access control policies. It is based on a privacy preserving attribute based key management scheme. It saves privacy of users at the time of enforcing attribute based ACPs. A single layer encryption (SLE) technique is proposed by them to enforce access control through process of encryption. A document broadcasting approach is based on access control policies specifying which users can access which documents, or subdocuments. It is supported by new group key management scheme. It is secure approach which permits qualified supporter to efficiently extract decryption keys for the portions of documents they allowed to access, based on the subscription information.

An attribute based encryption is discussed in [6]. Fine-grained access control in DaaS is also implemented. Issues like, data privacy, policy privacy and key privacy has been addressed. The proposed approach gives the privacy guarantee against any of the malicious parties, such as only the curious DSP or malicious user. The proposed scheme provides guarantee of multi-privacy such as, identity attribute privacy, data privacy and attributed conditions privacy in ACP. It gives assurance of any malicious parties. For the flexible attribute set combination. "Multi-owner" setting for data in encrypted format was outsourced by data owner and it can be searched and retrieved by multiple users. Keyword searching for authorized private keyword searches over encrypted data where multiple data owners encrypt their records along with a keyword index. Multiple users can search encrypted records on cloud. A scalable, fine-grained authorization framework is also represented to restrict performance of sensitive information which occurred due to unbounded capabilities of query. There are two novel solutions have been proposed in APKS based on HPE over an encrypted data. APKS enhances search efficiency using attribute hierarchy. The proposed solution supports the multi-dimensional range query, search capability delegation and revocation. An atomic proxy re-encryption application is discussed in [7]. Semi-trusted proxy translates ciphertext of one user into a ciphertext for other user without analyzing basic plaintext.

III. PROBLEM DEFINITION

"To design and develop keyDepository framework for key management to provide owner authorization and privacy preservation over cloud environment."

IV. SYSTEM ARCHITECTURE

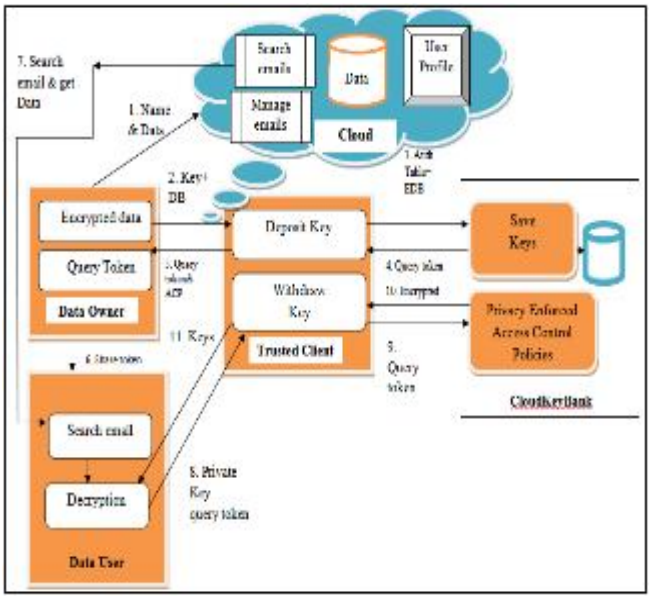

Figure 1. System Architecture

Cloudkey Depository architecture consists of four entities such as, Key owner, Cloudkey Depository, Trusted client and Users.

1. Compose mail:

It is primary module of proposed system using this user can compose or write mail. It is similar to other email system. It has other input fields explain below:

i. To : In this field, enter the email addresses of the person or persons to whom you are sending the message
ii. From : It contains email of data owner
iii. Subject: Enter the Subject of your Mail which lets people know what the email is all about.
iv. Compose mail : Type the Body of the Message in this box
v. Attached file: Attached file displayed in the body of the message. After clicking on click on "Send" button to send the entire email.

2. Create key:

It is denoted as keyGen(|x|) □ (pk, sk). The number of identity attribute-value pairs |x| are specified as an input for it. It generates the public as well as private key using RSA algorithm.

3. Encryption:

In encryption phase, data owner takes as input the delegated user's public key pk , a chosen vector x ∈ x and message m ∈ M. It generates the ciphertexts CT ∈ C of given input data.

4. Upload File or Send mail:

After data encryption composed mail send to entire emails included in mail.

5. Token generation:

In this phase, delegated user takes the private key as input (sku, w) which outputs the token TK w. This token is given to email sender or data owner when file is successfully saved on cloud server.

6. Trusted client:

It is trusted entity which preserve and manages the user token, key and key parameters. There are two protocols included in this entity for the purpose of key deposit and withdraw i.e. Deposit key and withdraw key. It is intermediate entity between user and cloudkey Depository.

7. Inbox :

When user request for specific file from cloud server, it get list of mails those shared with him. By referring this mail list user can download required file from it.

8. Download file:

To download required file from cloud server, user have to specify token which is given by cloud server after his registration process. This token gets verified by trusted client. For valid token file get downloaded into destination path.

9. Decryption:

In the decryption phase, on input the private key sk of data owner and ciphertext CT, there exist a decryption algorithm Di ∈ D which outputs the message m i.e. plain texts.

10. Cloudkey Depository :

This entity is special introduced in proposed system for management of all keys. It save all keys into database and enforce some privacy policies on them i.e. ACP. With the help of this entity key privacy and confidentiality can efficiently achieve.

V. MATHEMATICAL MODEL

Table1.

C is cloud O: {OI, OF, OO}	
OI: {OI1, OI2, OI3, OI4} Where,	OI1= Registration OI2= Login OI3= Document 'm' OI4= Key 'k'
OF: {OF1, OF2, OF3, OF4}	OF1= Select document OF2=→Encrypt document (pk_u; x; m) OF3= Upload Key OF4= Upload document
OO: {OO1, OO2}	OO1= → Encryption → Key Enc(pk_o, x, k)-→ CT. OO2= Encrypted Document
UI: {UI1, UI2}	UI1: User Details UI2: Generate download request 'QuTGen'
UF: {UF1, UF2, UF3, UF4, UF5}	UF1=Document →download request QuTGen(sk_o, w)-→ QuT_o, w UF2= → Download → Key DelGen(sk_o, pk_{du})→ $dk_{o\rightarrow du}$ UF3= download document UF4= Decrypt document Dec (sk_{du},rek)-→ k UF5= Save document
UO: {UO1, UO2}	UO1= Keys from trusted client UO2=Requested document from cloud
CB: {CBI, CBF, CBO}	
CBI: {CBI1}	CBI1= Keys
CBF: {CBF1}	CBF1= Save Keys
CBO: {CBO1}	CBO1= Retrieved Keys
TC: {TCI, TCF, TCO}	
TCI: {TCI1, TCI2}	TCI1= Keys from data owner
	TCI2= User Req. for key
TCF: {TCF1, TCF2}	TCF1= Deposit Key
	TCF2= Withdraw Key
TCO: {TCO1}	TCO1= Keys
C: {CI, CF, CO}	
CI: {CI1}	CI1= Encrypted document
CF: {CF1, CF2, CF3}	CF1 =Validate user details CF2= Save User details CF3= Save document
CO: {CO1}	CO1= File Storage

VI. ALGORITHMS

A. RSA Algorithm:

Input:
-Two large primes random numbers: p, q
-M = Numeric Block of Plaintext
Output:
Ciphertext(C)
Recovered Plaintext (RP)
Processing Steps:

1: Public Key Generation: (PK)

-Evaluate system modulus N=p.q
- Evaluate φ=(p-1)(q-1)
-Choose random no just as encryption key e >1
-gcd(e,φ)=1
-Public encryption key PK={e,N}

2: Generation of Private key: (SK)
d = e-1 (mod φ)
Private encryption key SK ={d,p,q}

3: If M modN !=0, then
$M^{e.d}$ mod N = M
Ciphertext $C=M^e$ mod N.

4: Decryption
$RP = C^d$ mod N

VII.EXPERIMENTAL SETUP

Using java environment system is implemented as client server architecture. For development jdk1.7, tomcat 7 and mysql 5.5 is used. At the servers end. Web services are written and at the client end GUI is designed using swing components. Netbeans-8.1 and Eclipse-Indigo IDE are used for development.

Dataset :

- **Synthetic Dataset:**

For Synthetic dataset generation files of following types are collected from different resources.

1. Text files: For testing enron [11] dataset files are used.
2. Image files: The Holidays dataset [12] contains 1491 set of images. Random images are selected from this dataset.
3. Compressed Files: These files are manually created by adding text, images in folder and compressed with zip extension.

VIII. RESULT TABLE AND DISCUSSION

Table 2: Efficiency Evaluation For Multimedia Files(For 2mb)

	Image	Audio	Video	Compressed
Encryption	0.47	0.39	0.45	1.5
Upload	0.7	1.34	1.4	1.54
Decryption	1.62	3.56	4.21	4.56
Download	4.04	4.11	4.45	4.73

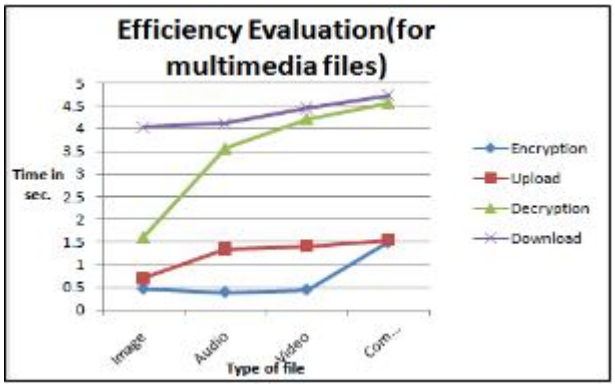

Figure 2. Graph of efficiency evaluation (for 2MB file size)

Table 3. Efficiency Evaluation For Multimedia Files(For 10mb)

	Image	Audio	Video	Compressed
Encryption	3.816	3.627	3.902	4.8
Upload	4.11	4.347	4.354	4.97
Decryption	7.84	8.11	8.42	8.67
Download	8.96	9.23	9.87	9.98

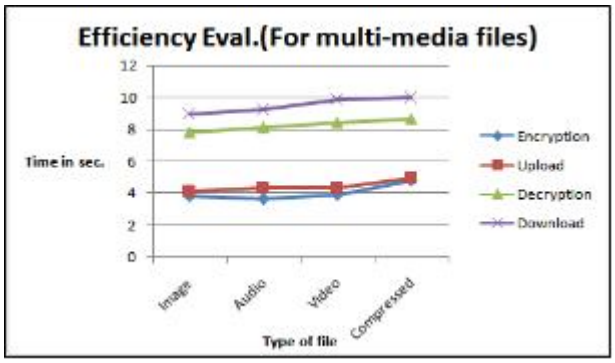

Figure 3. Graph of efficiency evaluation (for 10MB file size)

Table I & II, represents the efficiency evaluation readings for 2MB & 10MB multi-media files respectively, such as image, audio, video and compressed. According to observations time required for file encryption is less whereas, time required for file download is more.

Figure 2 & 3 depicts the graph of efficiency evaluation. In this X-axis represents the type of file and Y-axis represents the time in seconds.

Table 4. Efficiency Evaluation For Text Files

File size (in KB)	Index Gen.	Encry-ption	Upload	Decry-ption	Download
200	8.75	0.23	0.26	0.24	0.31
400	9.87	0.47	0.49	0.49	0.52
600	11.21	0.63	0.73	0.68	0.68
800	14.56	0.74	0.87	0.83	0.79
1000	17.33	0.94	0.96	0.98	0.99

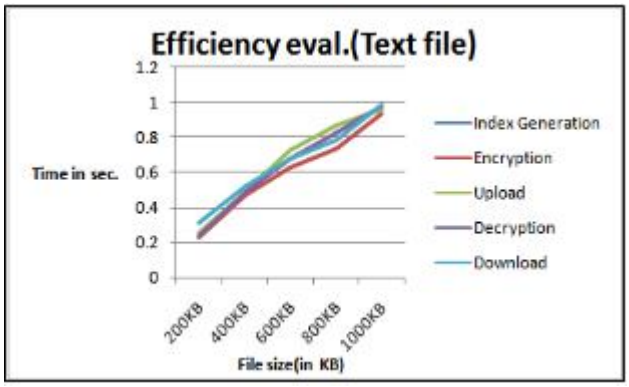

Figure 4. Graph of efficiency evaluation (text file)

The complete system is modelled in the ARENA software by the Rockwell and the method of verification s cross checked by the inbuilt model. The complete model is simulated for an 218 minutes and the statistics are observed

Table III contains the readings for text file encryption, upload, index generation, decryption and download. For system testing we have used text files various from 200KB to 1000KB in size. File upload time includes index generation and file encryption time. Whereas, file download time includes file decryption time in it.

Figure 4, depicts graph of text file efficiency evaluation. In this x-axis represents file size in KB and y-axis represents the time in second.

Table 5. Cloud key bank (Performance Of Query Response)

No. of keys in DB	Query response time(s)
20	0.13
40	0.21
60	0.39
80	0.55
100	0.63

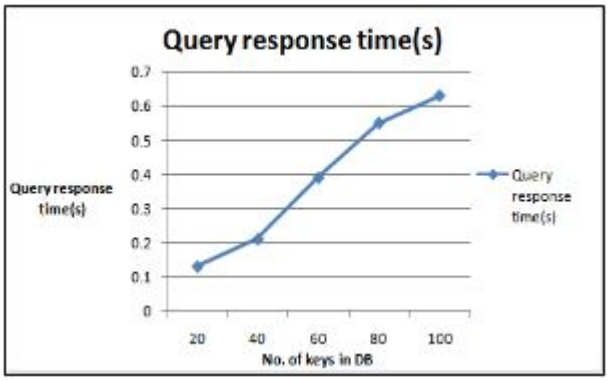

Figure 5. Graph of query response time

Table IV, represents the performance of cloud key Bank entity of proposed system. In this query response time or key retrieval time is calculated. In proposed system, for 20 key tuples required 0.13 seconds whereas, 100 tuples required 0.63 sec. The query response is more efficient because the Cloud Key Bank provider does not need to return the whole encrypted key tuple like that in most of the existing proposed approaches.

Figure 5, depicts the graphical format for the performance of cloud Key Bank entity. In this, X-axis represents number of keys in DB and Y-axis represents the query response time in seconds.

Table 6. Search Efficiency

No. of files	Search time (in sec.)
50	0.04
100	0.06
150	0.8
200	0. 9

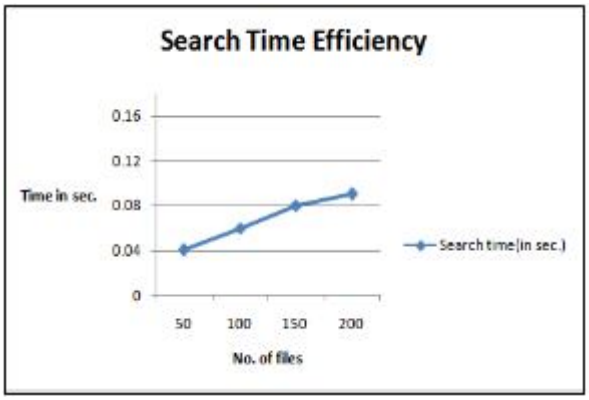

Figure 6. Graph of search efficiency

Table V, represents search time efficiency of proposed system. As per observation, search time is linearly grows with the increasing number of files.

Figure 6, represent search time efficiency graph in which X-axis represents number of files and Y-axis represents time in second.

Table 7. Comparative Analysis

No. of documents	Existing system	Proposed system
50	50	20
100	100	45
150	150	97
200	200	116

Figure 7. Graph of comparative analysis

In table VI, comparative analysis between existing and proposed system is given in terms of document retrieval.

In case of certain document retrieval (i.e. the document for which no search details are available), existing system retrieves all documents from cloud server which is an inefficient task which required extra bandwidth.

Such issue is overcome in proposed system. Table VI, shows search results for 2- query keywords. Existing system retrieve all documents whereas, proposed system only retrieves specific and query relevant documents. It preserves the bandwidth and increases the search efficiency.

IX. CONCLUSION

Multiple methods have been proposed in the literature survey regarding to outsourcing of data which are unable to meet the security requirements of outsourced keys. Therefore, in this system we identified the problem of security requirements for outsourcing keys such as, to preserve confidentiality and privacy of keys, owners controllable authorization and Search privacy on identity attributes tied to keys. For efficient key management we introduced cloudkey Depository framework. It is represented by implementing novel scheme SC-PRE and comparative analysis of security requirements of key outsourcing. In experimental analysis, we have show that proposed solution can be a worth solution for previously discussed security problems.

X. ACKNOWLEDGMENT

I wish to express my sincere gratitude to H.O.D Prof. S. M. Rokade of M.E. Computer Engineering Department for providing me an opportunity for presenting the topic " Mining of URSTP's". I sincerely thank to my guide Prof. S.M.Rokade for his guidance and encouragement in the completion of this work.

I also wish to express my gratitude to the officials and other staff members who rendered their help during the period. Last but not least I wish to avail myself of this opportunity, to express a sense of gratitude and love to my friends and my parents for their manual support, strength, help and for everything.

REFERENCES

[1] H. Hacigumus, B. Iyer, C. Li, and S. Mehrotra, "Executing SQL over encrypted data in the database-service-provider model," in Proc. 18th Int. Conf. Data Eng., 2002, pp. 216–227

[2] X. Tian, X. Wang, and A. Zhou, "DSP re-encryption a flexible mechanism for access control enforcement management in DaaS, in Proc. IEEE Int. Conf. Cloud Comput., 2009, pp. 25–32.

[3] J. Shao and Z. Cao, "CCA-secure proxy re-encryption without pairings," in Proc. 12th Int. Conf. Practice Theory Public Key Cryptography, 2009, pp. 357–376

[4] M. Li, S. C. Yu, N. Cao, and W. Liu, "Authorized private keyword search over encrypted data in cloud computing," in Proc. 31st Int. Conf. Distrib. Comput. Syst., 2011, pp. 383–394

[5] N. Shang, F. Paci, M. Nabeel, and E. Bertino, "A privacypreserving approach to policy-based content dissemination," in Proc 26th Int. Conf. Data Eng., 2010, pp. 944–955.

[6] M. Nabeel and E. Bertino, "Privacy preserving delegated access control in public clouds," IEEE Trans. Knowl. Data Eng., vol. 26, no. 9, pp. 2268–2280, Sep. 2014

[7] X. X. Tian, L. Huang, Y. Wang, C. F. Sha, and X. L. Wang, "DualAcE: Fine-grained dual access control enforcement with Multi-privacy guarantee in DaaS," Secure Commun. Netw., vol. 8, no. 8, pp. 1494–1508, 2015

[8] M. Li, S. C. Yu, N. Cao, and W. Liu, "Authorized private keyword search over encrypted data in cloud computing," in Proc. 31st Int. Conf. Distrib. Comput. Syst., 2011, pp. 383–394.

[9] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," in Proc. 12th Annu. Netw. Distrib. Syst. Security Symp., 2005, pp. 29–44.

[10] Xiuxia Tian, Ling Huang, Tony Wu, Xiaoling Wang ,"Cloudkey Depository: Privacy and Owner Authorization Enforced Key Management Framework", IEEE transaction on knowledge and data engineering, dec.2015,vol.27, no.12.

[11] https://www.cs.cmu.edu/~./enron/

[12] http://lear.inrialpes.fr/people/jegou/data.php