PRIVACY AND OWNER AUTHERISATION FRAMEWORK TO MANAGE KEYS

Darshan Chavan[1], Prof.P.M.Yawalkar[2]
[1] *M.E., Computer Engineering Department, Maharashtra, India*
[2] *Computer Engineering Department, Maharashtra, India*

ABSTRACT

The number of encryption keys increases due to limit of web app users get exceeds. Outsourcing keys to the professional password managers is an attractive trend. Traditional outsourcing scenarios are not capable to meet the security requirements for outsourcing keys such as, privacy & confidentiality of keys, privacy on attribute ties to keys at the time of search. In proposed system, SC-PRE scheme is utilized. It combines two approaches such as, Proxy Re-encryption (PRE) and Hidden Vector Encryption (HVE). It encrypts key tuple similar to normal data encryption. The proposed cloudKeyBank framework achieved three security requirements for outsourcing of keys such as, confidentiality of key privacy, attributes of search ties with key and owner controllable authorization on shared keys. Bloom filter is space-efficient probabilistic data structure designed to specify whether an element is present or not in the set. It is rapid and memory efficient strategy which we contribute in our proposed work. It can save search space and time of searching.

Keyword: *SC-PRE, search privacy, key management, keys outsourcing*

1. INTRODUCTION

Password and key management is the challenging task which occurred in the scenario of outsourcing data. In this era, there is huge growth in the use f web applications. Users have multiple online accounts for various purposes. Web development deploys 'n' number of applications such as, social networking (Facebook, twitter, linkend), shopping sites (Amazon, eBay, snapdeal etc) and data storages such as, googledrive. To provide authorization for individuals account there is registration and login page, by using this new user can registered their details to web application and further use this registered credentials for accessing their accounts. Similar approach is carried out in case of data outsourcing to the cloud server or cloud storage. Generally, user uploads their data to cloud in encrypted format for the perspective of data security. Data encryption is performed using cryptographic functions or using encryption algorithm. In the process of data encryption, some secret keys are generated at the user's end which may uploaded to cloud server for their appropriate management and these keys are also forwarded to other user's for data retrieving purposes. Due to cost efficient and scalable data storage, many users and huge business industries take benefit of it for management and maintenance of their heavy data. According to survey analysis, 90% of students were concern about privacy to their key or password. There are two types of situations about privacy concern such as, they do not fully trust the service providers because there is no governance about how keys can be used by them and whether the key owner can actually control their keys on their own OR they trust the service providers, but keys could be disclosed if there exists an misbehaving internal employee or broken server. Hence to provide privacy for key is to encrypt key tuple before outsourcing them and encryption process is similar to the normal data encryption. It is promising solution to maintain trust and also to ensure key privacy and owner control on outsourced keys.

Proposed system is based on SC-PRE i.e. Searchable Conditional Proxy Re-Encryption scheme. It combines two techniques namely, hidden vector encryption (HVE) and proxy re-encryption (PRE). It can efficiently solves the challenging issues occurred in key tuple encryption. During key tuple encryption some critical issues are identified such as, keys are highly sensitive and they need to be secured from honest-but-curious service provider and malicious attacks.

In proposed cloudkeybank framework, there are three main entities are included such as, cloud, trusted client and cloudkeyBank. Trusted client is the intermediate between cloud and cloudkeyBank. Two types of protocols used for key traversal i.e. depositeKey protocol and withdrawKey.

DepositKey protocol is used to distribute key DB under the support of ACP i.e. access control policies. It distributes the query token for authorized users only.

CloudkeyBank provider is the professional password manager. It gives the access control policies on EDB. It mainly carried out two tasks such as, To enforce the privacy of identity attributes in the Search attribute group, he/she can perform search query directly by evaluating the submitted Query token against the encrypted key tuples in EDB; and To enforce the key authorization he/she can transform an encrypted key into the authorized re-encrypted key under the corresponding Delegation token stored in authorization table (AuT). There are two types of users involved in this system user and data owner. Data owner deposit keys to cloudkeyBank and it can only access by himself whereas, users having query token access authorized key using withdraw key protocol.

Along with the management of outsourced keys and provide privacy for them proposed system also make searching efficiently by implementing bloom filter.

Bloom filter based index in one server, and access policy enforcement in another server to support scalable operations on encrypted key database.

2. RELATED WORK

2.1 SQL over Encrypted Data

DAAS is "Database as a service" concept of storing outsourcing data on cloud. Data owner stores the data in encrypted format using some cryptographic techniques for privacy preservation [2]. Privacy preservation technique is discussed by Tracey Raybourn[3], this technique is known as bucketisation encryption. This technique partitioned the encrypted attributes into querytable bucket or table. Bucketisation solves the problem of usually required a tradeoff in which greater security is achieved. OPES is order Preserving Encryption Scheme. It can precisely employd on an encrypted data [4]. Correspondance, range queries and MIN, MAX and COUNT queries directly processes over an encrypted data. It is efficient encryption technique for avoiding data misuse. OPES has limitation on numeric data encryption. Authenticated index structures based on various cost metrics is proposed by F. Li, M. Hadjieleftheriou et al. for cryptographic operations and index maintenance. This technique formulates the problem of query freshness. For index maintenance B+ tree approach is implemented. But this technique provides less support for privacy leakage. Whereas, previous discussed topics provides the confidentiality guarantee and privacy for data tuples [5]. A group Key management scheme is discussed Xiaoling Wang, Aoying Zhou, allows an efficient access to extract the decryption key for specific portion that permitted for them on the basis of subscription information. The problem of enforcing access control in DSP to make more utilization of system DSP-reencryption concept is suggested by Aoying Zhou. Two types of combination namely ASBE scheme and DSP re-encryption is introduced in [9] for flexible dual fined-grained access control enforcement mechanism. These techniques work against efficient key management. Another techniques discussed by Ling Huangin [7],, Yong Wang, focused on gaining identity and authorization privacy of users. To provide compressive protection for outsourced medical data there are two techniques combined namely, digital watermarking and binning which is discussed by Elisa Bertino[10].

2.2 Searching an encrypted data

There are many techniques and algorithms are available to perform searching on encrypted data. SSE is Searchable symmetric encryption and PKES public encryption with keyword search discussed in by Dawn Xiaodong Song David Wagner etal[11] [12]. SSE supports the collective search and basic Boolean queries on outsourced symmetrically encrypted data. This scheme supports both structural and textual data with basic Boolean queries. To provide proof of security remote searching techniques discussed byElisa Bertino[11]. Remote searching techniques have many advantages such as; they are more secure, controlling support, hidden search and isolation. A new generalized identity based encryption approach known as predicate encryption is proposed by j. Katz etal [10]. In predicate encryption privacy keys are correlated with predicates and cipher-texts are identified with attributes. For any supported query predicate token is produced in public key systems that supporting query on encrypted data. Hidden Vector encryption is technique used for conjunctive query search over an encrypted data. It is essentially anonymous IBE scheme as they construct a bilinear group with a composite order. This work is extended to support predicate encryption to disjunction and inner product. HVE is hidden vector encryption scheme provides ciphertext associated with a binary attribute vector and k-key associated with vector [14]. There is condition for decryption of ciphertext such that k-key have to satisfy the predicate of key. This technique is used to access fined-grained control

on an encrypted data. Predicate encryption is mechanism which gives master secret key owner fine-grained control over access for encrypted data [15]. A novel technique to realize a tag-based dual system encryption in prime-order groups HVE scheme is introduced byJong Hwan Park , Kwangsu Lee, which is based on bilinear maps (pairings), provides efficiency advantages in that it requires O(1)-sized private keys and O(1) pairing evaluations for data decryption.

2.3 Proxy re-encryption

M. Blaze introduced Divertible Protocols and Atomic Proxy Cryptography scheme. Both are the security properties. Atomic Proxy Cryptography is extension for existing public key cryptography. Previous schemes encrypt the data without granting the ability to decrypt it. Also there exist some systems that re-encrypt the data without granting ability to decrypt it [17]. PRE is proxy re-encryption scheme which re-encrypts the ciphertext from sender. Two types of proxy encryptions are available namely, unidirectional and bi-directional. From both of this unidirectional scheme is most trustworthy as asymmetric proxy functions are involved in it. In unidirectional proxy encryption scheme [18] do not required any delegator for revealing their secret keys to anyone in order to permit proxy to re-encrypt their ciphertexts. Identity based proxy encryption is proposed byRan Canetti_ Susan Hohenberger, to identify the problem in identity based proxy re-encryption while transferring ciphertext from one identity to another one. Multiple different schemes are proposed for proxy re-encryption, in that some PRE schemes work against chosen ciphertext attacks. CCA-Secure Proxy Re-Encryption is introduced by J. Shao[21], , a semi-trusted proxy can transform the ciphertext under one's public key into another ciphertext that can decrypt by other user. Due transformation scheme of proxy it can be used in many applications for example in encrypted mail forwarding. CCA is chosen ciphertext attack that work against security issues. CCA –secure and collusion resistant unidirectional PRE scheme implemented to solve the problem of ciphertext attacks. Searchable public key encryption scheme with designated testers (dPEKS) is implemented to keyword guessing attacks as there exist some vulnerable to keyword guessing attacks by malicious servers. This because an outer attacker can make the use of server as a test oracle to verify the correctness of the keyword that guessed by attackers.

2.4 Bloom Filters

Generally, bloom filter could be defined as the probabilistic data structure used to test whether the element is the member of a set. Bloom filters has 100% recall rate as, there is more possibilities of false positive matches than false negative matches. It is concept based on hashing. Bloom filters has fixed or constant time complexity for adding and determining whether the element is present or not. Many cases there is need to perform quick look-up for deciding how to respond for incoming request. It is the com3pact representation of membership in set. In this, incremental result will automatically halt after getting fixed number of results.

3. PROBLEM DEFINITION

To design CloudKeyBank framework to provide privacy for outsourcing keys as well as search privacy on identity attributes.

4. SYSTEM ARCHITECTURE

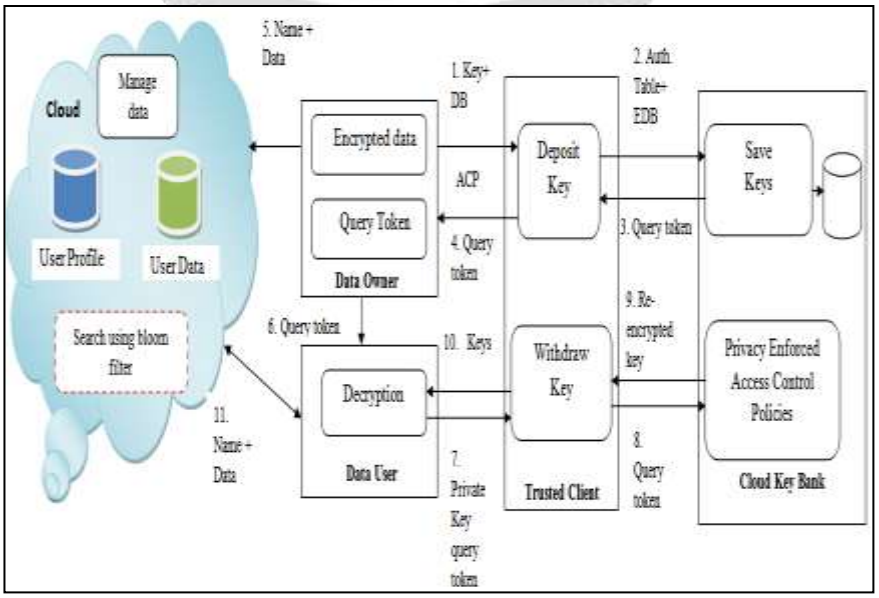


Fig -1: System Architecture

Above fig. 1 depicts system architecture of proposed system. There are four types of entities present in system architecture. Each entity has some responsibility to which represents the flow of system. It given as below:

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:

To: In this field, enter the email addresses of the person or persons to whom you are sending the message

From: It contains email of data owner

Subject: Enter the **Subject of your Mail** which lets people know what the email is all about.

Compose mail: Type the **Body of the Message** in this box

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|) → ($\vec{pk}$, 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 $x \in x$ and message $m \in M$. It generates the ciphertexts $CT \in \vec{C}$ 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, $\vec{w}$) which outputs the token $TK_{\vec{w}}$. 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 cloudKeyBank.

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. CloudKeyBank:

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.

5. ALGORITHMS

5.1 RSA Algorithm:

Input:

-Two large primes random numbers: p, q

-M = Numeric Block of Plaintext

Output:

C: Ciphertext

OM: Recovered Plaintext

Processing Steps:

Step1: Public Key Generation: (KU)

-Measure system modulus N=p.q

- Measure φ=(p-1)(q-1)

-Choose random no as encryption key e >1

-gcd(e,φ)=1

-Public encryption key KU={e,N}

Step 2: Private key Generation: (KR)

d = e-1 (mod φ)

Private encryption key KR={d,p,q}

Step 3: If M modN !=0, then

Me.d mod N = M

Ciphertext C=Me mod N.

Step4: Decryption

OM = Cd mod N

6. EXPERIMENTAL SETUP

System is implementing on java-jdk 1.7.0 platform.

At server side 64-bit CentOS 6.x having 2GB RAM, with Apache-7 and mysql-5.6 is configured on same system at server side.

At client side we have used system having i3 processor and 4GB RAM. Netbeans 8.0.1 IDE is used for implementation of client side system, it is designed using swing components whereas, Eclipse indigo is used for cloud server system implementation.

A. Dataset Used:

1. Text files: For testing enron [22] dataset is used. It is email dataset containing 6000,000 emails of 158 users. It is in .txt format. Fig. shows the sample format of dataset.
2. PDF Files: Synthetic dataset is generated for pdf files. Random 100 IEEE document files and some 50 pdf books are collected.
3. Image files: The Holidays dataset [23] is a set of images which mainly contains holiday traveling photos. It contains 1491 set of images
4. Video Dataset: synthetic dataset is generated for various video clips from 10 to 100MB.

7. RESULT TABLES AND DISCUSSION

Table-1: Time evaluation for PDF file

				Download	
File size(in MB)	**Encryption**	**Upload**	**Decry-ption**	**With bloom filter**	**With-out bloom filter**
1	0.44	0.6	1.45	2.05	9.58
3	1.39	1.74	2.07	3.24	11.26
5	3.2	3.45	3.78	4.45	12.97
7	5.77	5.86	6.24	5.74	17.68
10	9.78	9.93	10.38	14.29	21.1

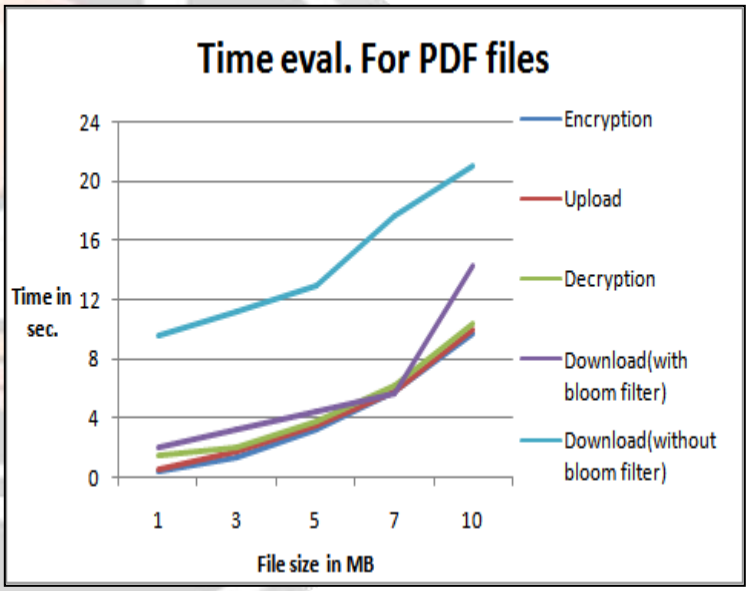


Chart-1: Time evaluation (for PDF file)

Table 1 represents the time evaluation for PDF file which get varies from 1MB to 10MB in size. Along with the file uploading time, encryption time is also included into file upload time.

In chart 1, graph of PDF file encryption, upload, download and decryption is shown. In this, X-axis represents file size in MB and Y-axis represents the time in second.

Table-2: Time evaluation for Image files

Image size	Encryption	Upload	Decryption	With bloom	Without bloom
10	3.62	4.12	7.12	8.17	14.26
20	5.32	4.85	7.84	10.59	16.75
30	6.59	5.66	8.38	11.17	17.61
40	6.88	6.11	8.45	11.65	17.78
50	7.23	7.67	8.67	11.73	17.89
60	7.86	8.17	8.97	12.01	18.31
70	8.14	8.58	8.99	12.14	18.57
80	8.78	8.83	9.17	12.28	19.02
90	8.97	10.25	10.73	12.87	19.45
100	9.43	10.63	11.04	13.12	19.88

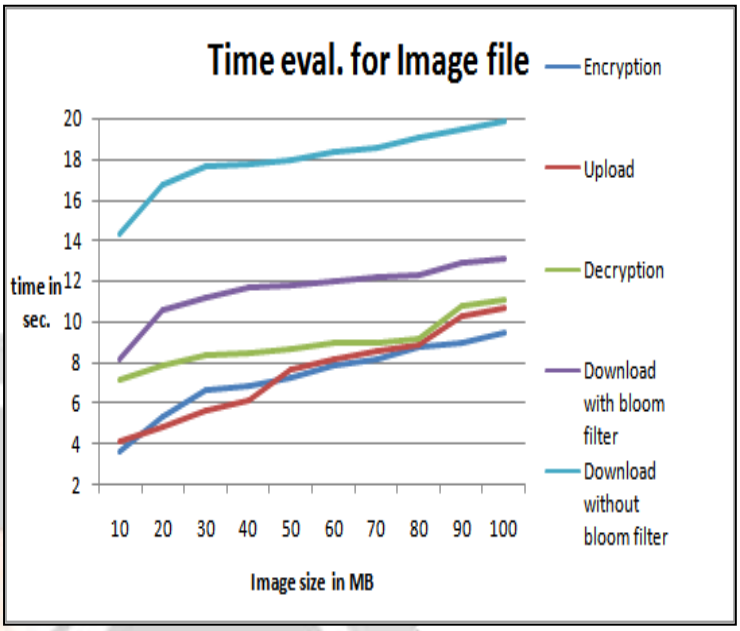


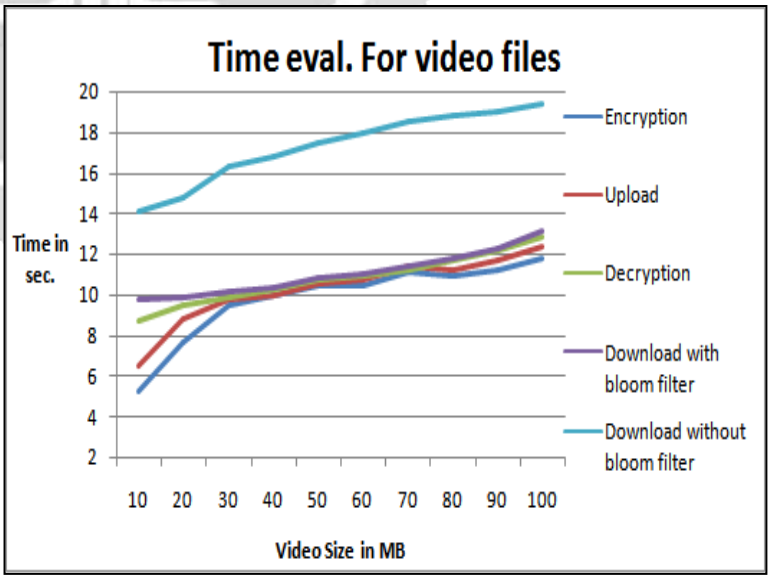
Chart-2: Time evaluation (image file)

Table 2 consists of image encryption, upload, download and decryption timing. The respective readings are taken into seconds(s). For testing we have used images various from 10MB to 100MB. Image encryption is also included into image upload time.

In chart-2, graphical form of images encryption, upload, download and decryption is shown. In this X-axis represent the file size in MB and Y-axis represents the time in second

Table-3: Time evaluation for video file

File size	Encryption	Upload	Decryption	With bloom	Without bloom
10	5.25	6.53	8.74	9.77	14.11
20	7.64	8.78	9.51	9.87	14.78
30	9.47	9.78	9.89	10.12	16.32
40	9.95	9.92	10.27	10.33	16.79
50	10.42	10.57	10.76	10.82	17.47
60	10.48	10.72	10.89	11.05	17.89
70	11.14	11.37	11.19	11.41	18.55
80	10.87	11.17	11.65	11.78	18.79
90	11.23	11.68	12.16	12.25	19.02



100	11.74	12.32	12.79	13.09	19.34

Chart-3: Time evaluation (video file)

Table-3 consists of video file encryption, upload, download and decryption timing. The respective readings are taken into seconds(s). Various sizes of video files which vary from 10MB to 100MB are used for testing. In this, video encryption is also included into video upload time.

In chart-3, graphical form of video encryption, upload, download and decryption is shown. In this X-axis represent the video file size in MB and Y-axis represents the time in second.

Table-4: Time evaluation for compressed data

File size	**Encryption**	**Upload**	**Decryption**	**Download with bloom filter**	**Download without bloom filter**
10	8.76	9.13	9.85	10.09	15.27
20	9.64	9.79	10.51	10.56	15.87
30	9.89	10.24	10.73	11.02	17.21
40	10.21	10.78	11.07	11.23	17.45
50	10.79	11.17	11.59	11.78	17.78
60	11.19	11.48	11.92	12.07	18.19
70	11.74	11.81	12.17	12.34	18.48
80	12.14	12.34	12.78	13.11	19.06
90	12.64	12.72	13.03	13.26	19.87
100	13.84	13.93	13.66	13.97	20.12

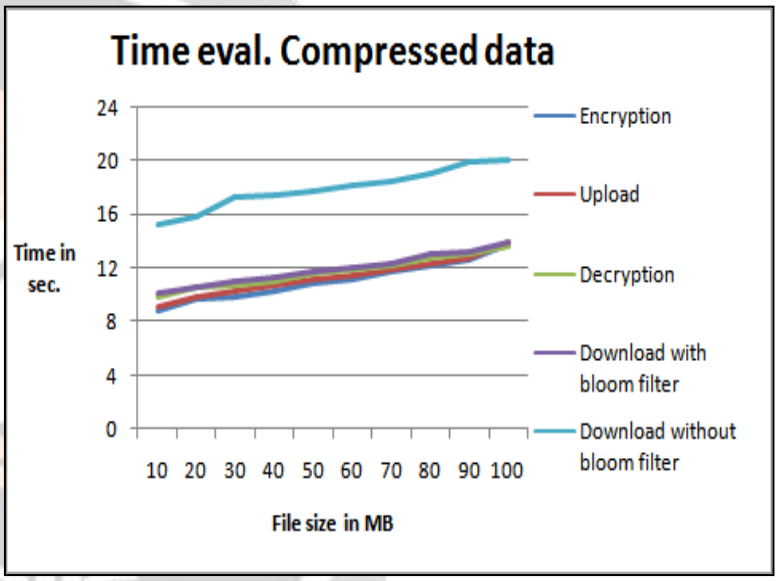


Chart-4: Time evaluation (compressed data)

Table 4 consists of compressed data encryption, upload, download and decryption timing. The respective readings are taken into seconds(s).

Various sizes of video files which vary from 10MB to 100MB are used for testing. In this, compressed data encryption is also included into compressed data upload time.

In chart-4, graphical form of compressed data encryption, upload, download and decryption is shown. In this X-axis represent the compressed data size in MB and Y-axis represents the time in second.

Table-5: Query Response Time

No. of keys in database	Query response time(s) without bloom filter	Query response time(s) with bloom filter
20	1478	623
40	1523	721
60	1637	820
80	1795	947
100	1913	1011

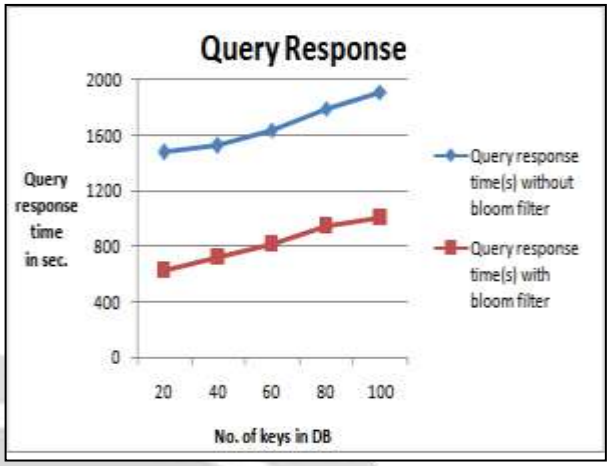


Chart-5: Query response time

Table-5 represents query response time for number of keys in database such as, 20,40,60,80,100. According to observation, query response time required for 100 key tuples is 0.945 seconds.

Proposed system utilizes the bloom filter for efficient key retrieval. Hence, as per observation proposed system gives response for query in very less time as compared to existing or without bloom filter approach.

Table-6: Comparative Analysis

			Owner controllable authorization	
Scheme/Security	**Key confidentiality**	**Search privacy**	**Key Authorization**	**Efficient Key Retrieval**
Hacigumus[2]	Y	N	N	N
Cash[12]	N	Y	N	N
Shang[6]	Y	N	N	N
Nabee[6]	Y	N	N	N
Tian[7]	Y	N	Y	N
Li[5]	N	Y	N	Y
CloudKeyBank[1]	Y	Y	Y	Y
Proposed Solution	Y	Y	Y	Y

Table-6 represents the comparative analysis between existing key or password management techniques. According observation we can predict that proposed privacy and owner authorization framework provides more privacy guarantee than existing techniques.

In proposed framework, efficient key retrieval approach is also provided. Table-5 shows the performance of key retrieval with and without bloom filter.

8. CONCLUSIONS

Privacy and owner authorization framework is proposed to address key challenges in security requirements for outsourcing keys. The proposed framework is based on SC-PRE scheme. It is a cryptographic primitive which proves all three requirements which does not solve existing key outsourcing scenario. Proposed system contributes bloom filter based index on single server which result into search efficiency and can give an efficient solution for outsourced key management. Due to bloom filter proposed system improves the performance in terms of query response time is 21.01%.

9. ACKNOWLEDGEMENT

10. REFERENCES

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

[2] 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.

[3] Tracey Raybourn ,"Bucketisation Technique for Encrypted Databases:Quantifying the impact of Query Distribution",a thesis of master of science, May 2013

[4] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in Proc ACM SIGMOD Int. Conf. Manag. Data, 2004, pp. 563–574.

[5] F. Li, M. Hadjieleftheriou, G. Kollios, and L. Reyzin, "Dynamic authenticated index structures for outsourced databases," in Proc ACM SIGMOD Int. Conf. Manag. Data, 2006, pp. 121–132

[6] 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.

[7] 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.

[8] X. X. Tian, X. L. Wang, and A. Y. Zhou, "DSP Re-encryption based access control enforcement management mechanism in DaaS," Int. J. Netw. Security, vol. 15, no. 1, pp. 28–41, 2013.

[9] 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

[10] X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. IEEE Symp. Security Privacy, Oakland, California, USA, May 2000, pp. 44–55.

[11] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in Proc. 27th Annu. Int. Conf. Adv. Cryptol. Theory Appl. Cryptograph. Techn., vol. 4965, pp. 146–162, 2008.

[12] D. Cash, Stanislaw, C. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, "Highly-scalable searchable symmetric encryption with support for boolean queries," in Proc 33th Int. Conf. Crypography Conf., 2013, pp. 353–373.

[13] V. Iovino and G. Persiano, "Hidden-vector encryption with groups of prime order," in Proc. Int. Conf. Pairing-Based Cryptography, 2008, vol. 5209, pp. 75–88.

[14] J. Hwan Park, K. Lee, W. Susilo, and D. Hoon Lee, "Fully secure hidden vector encryption under standard assumptions," Inf. Sci., vol. 232, pp. 188–207, 2013

[15] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in Proc. Int. Conf. Theory Appl. Cryptograph. Techn., 1998, pp. 127–144.

[16] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in Proc. 5th Int. Conf. Appl. Cryptography Netw. Security, 2007, pp. 288–306

[17] R. Canetti and S. Hohenberger, "Chosen-ciphertext secure proxy Re-encryption," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, pp. 185–194.

[18] L. M. Fang, W. Susilo, C. P. Ge, and J. D. Wang, "Hierarchical conditional proxy re-encryption," Comput. Standards Interfaces, vol. 34, pp. 380–389, 2012

[19] H. S. Rhee, W. Susilo, and H.-J. Kim, "Secure searchable public key encryption scheme against keyword guessing attacks," IEICE Electronics Express, vol. 6, no. 5, pp. 237–243, 2009.

[20] L. Fang, W. Susilo, and J. Wang, "Anonymous conditional proxy re-encryption without random oracle," in Proc. 3rd Int. Conf. Provable Security, 2009, pp. 47–60.

[21] 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

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

[23] http://lear.inrialpes.fr/~jegou/data.php