SECURE KEYWORD SEARCH AND DATA SHARING FOR THE CLOUD

BILLAKURTHI LAKSHMI BHAVANI #[1], K.RAMBABU [#2]

[#1] MCA Student, Master of Computer Applications,

D.N.R. College, P.G.Courses & Research Center, Bhimavaram, AP, India.

[#2] Head & Assistant Professor, Master of Computer Applications,

D.N.R. College, P.G.Courses & Research Center, Bhimavaram, AP, India.

ABSTRACT

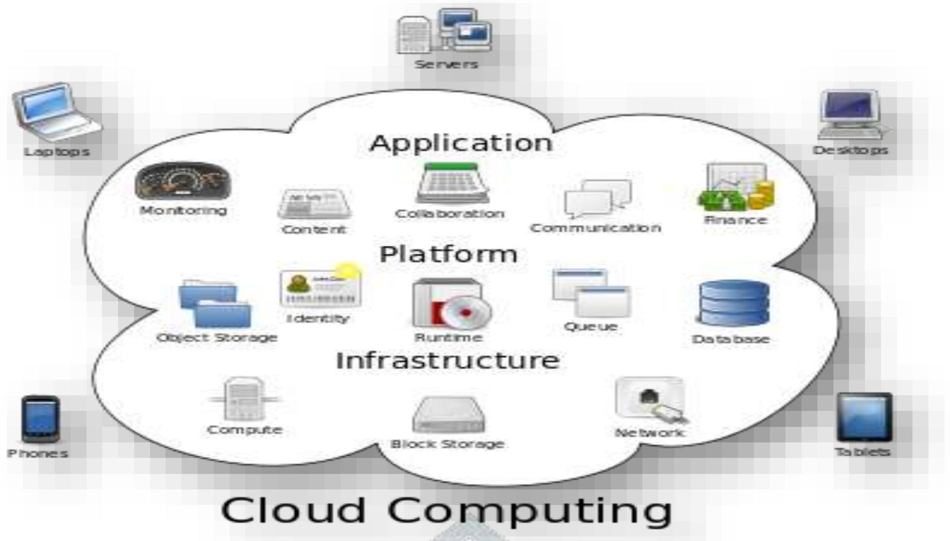
In current days cloud computing has become one of the fascinating domain which was accessed by almost all users in order to store ,retrieve and access the data from remote systems rather than from the local machines. We propose a novel and efficient scheme that guarantees data confidentiality even if the encryption key is leaked and the adversary has access to almost all cipher text blocks. We analyze the security of our proposed application and we evaluate its performance by means of a prototype implementation.

KEYWORDS:

Cloud Server, Cipher Text, Encryption, Efficient Scheme.

1. INTRODUCTION

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.



HOW CLOUD COMPUTING WORKS?

The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games.

The cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing.

All the existing cloud servers try to store the data in a plain text manner rather than in a encrypted manner. If the encryption key is exposed, the data can be easily accessed by the intruder. In the existing system there is no security for the data even it is encrypted because there is only single cipher key is generated for that data.

In this proposed system, we study data confidentiality against an adversary which knows the encryption key and has access to a large fraction of the cipher text blocks. The adversary can acquire the key either by exploiting flaws or backdoors in the key-generation software, or by compromising the devices that store the keys (e.g., at the user-side or in the cloud). To counter such an adversary, we propose a novel and efficient scheme which ensures that plaintext data cannot be recovered as long as the adversary has access to at most all but *two* ciphertext blocks, even when the encryption key is exposed. The main objective of this present application is design a secure keyword search and provide security by providing multiple keys for encryption or decryption rather than providing single key for encryption or decryption. This is mainly achieved by using polynomial time series algorithm.

2. LITERATURE SURVEY

2.1 INRODUCTION

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next steps are to determine which operating system and language used for developing the tool. Once the programmers start building the tool, the programmers need lot of external support. This support obtained from senior programmers, from book or from websites. Before building the system the above consideration r taken into account for developing the proposed system.

1 “Secret-Sharing Schemes: A Survey,”

AUTHORS: A. Beimel,

A secret-sharing scheme is a method by which a dealer distributes shares to parties such that only authorized subsets of parties can reconstruct the secret. Secret-sharing schemes are important tools in cryptography and they are used as a building box in many secure protocols, e.g., general protocol for multiparty computation, Byzantine agreement, threshold cryptography, access control, attribute-based encryption, and generalized oblivious transfer. In this survey, we will describe the most important constructions of secret-sharing schemes, explaining the connections between secret-sharing schemes and monotone formulae and monotone span programs. The main problem with known secret-sharing schemes is the large share size: it is exponential in the number of parties. We conjecture that this is unavoidable. We will discuss the known lower bounds on the share size. These lower bounds are fairly weak and there is a big gap between the lower and upper bounds. For linear secret-sharing schemes, which is a class of schemes based on linear algebra that contains most known schemes, super-polynomial lower bounds on the share size are known. We will describe the proofs of these lower bounds. We will also present two results connecting secret-sharing schemes for a Hamiltonian access structure to the NP vs. coNP problem and to a major open problem in cryptography – constructing oblivious-transfer protocols from one-way functions

2.Using Erasure Codes Efficiently for Storage in a Distributed System,”

AUTHORS:M. K. Aguilera, R. Janakiraman, and L. Xu

Erasure codes provide space-optimal data redundancy to protect against data loss. A common use is to reliably store data in a distributed system, where erasure-coded data are kept in different nodes to tolerate node failures without losing data. In this paper, we propose a new approach to maintain ensure-encoded data in a distributed system. The approach allows the use of space efficient k-of-n erasure codes where n and k are large and the overhead n-k is small. Concurrent updates and accesses to data are highly optimized: in common cases, they require no locks, no two-phase commits, and no logs of old versions of data. We evaluate our approach using an implementation and simulations for larger systems.

3.**“Security amplification by composition: The case of doublyiterated, ideal ciphers,”**

AUTHORS:W. Aiello, M. Bellare, G. D. Crescenzo, and R. Venkatesan

One concern in using cloud storage is that the sensitive data should be confidential We investigate, in the Shannon model, the security of constructions corresponding to double and (two-key) triple DES. That is, we consider Fk1 (Fk2 ()) and Fk1 (F 1 k2 (Fk1 ())) with the component functions being ideal ciphers. This models the resistance of these constructions to \generic" attacks like meet in the middle attacks. sense. We compute a bound on the probability of breaking the double cipher as a function of the number of computations of the base cipher made, and the number of examples of the composedcipher seen, and show that the success probability is the square of that for a single key cipher. meet in the middle is the best possible generic attack against the double cipher. local revocable group signature and identity-based broadcast encryption with constant size ciphertext and private keys. To realize our concept, we equip the broadcast encryption with the dynamic ciphertext update feature, and give formal security guarantee against adaptive chosen-ciphertext decryption and update attacks.

4.“The security of all-or-nothing encryption: Protecting against exhaustive key search,”

AUTHORS:A. Desai,

We investigate the all-or-nothing encryption paradigm which was introduced by Rivest as a new mode of operation for block ciphers. The paradigm involves composing an all-or-nothing transform (AONT) with an ordinary encryption mode. The goal is to have secure encryption modes with the additional property that exhaustive key-search attacks on them are slowed down by a factor equal to the number of blocks in the ciphertext. We give a new notion concerned with the privacy of keys that provably captures this key-search resistance property. We suggest a new characterization of AONTs and establish that the resulting all-or-nothing encryption paradigm yields secure encryption modes that also meet this notion of key privacy. A consequence of our new characterization is that we get more efficient ways of instantiating the all-or-nothing encryption paradigm. We describe a simple block-cipher-based AONT and prove itsecure in the Shannon Model of a block cipher. We also give attacks against alternate paradigms that were believed to have the above keysearch resistance property.

3. EXISTING SYSTEM

All the existing cloud servers try to store the data in a plain text manner rather than in a encrypted manner. If the encryption key is exposed, the data can be easily accessed by the intruder. In the existing system there is no security for the data even it is encrypted because there is only single cipher key is generated for that data.

LIMITATION OF EXISTING SYSTEM

The following are the limitation of existing system. They are as follows:

1. All the existing schemes are limited to the single-owner model.
2. All the existing cloud servers has search in a normal manner under plain text model, but they don't have any facility to search in a ENRYPTED manner
3. The existing cloud servers are almost operated in a centralized manner, where all the access can be viewed and monitored by the cloud service providers.

4. PROPOSED SYSTEM

In this proposed system, we study data confidentiality against an adversary which knows the encryption key and has access to a large fraction of the cipher text blocks. The adversary can acquire the key either by exploiting flaws or backdoors in the key-generation software, or by compromising the devices that store the keys (e.g., at the user-side or in the cloud). To counter such an adversary, we propose a novel and efficient scheme which ensures that plaintext data cannot be recovered as long as the adversary has access to at most all but *two* ciphertext blocks, even when the encryption key is exposed.

ADVANTAGES OF THE PROPOSED SYSTEM

The following are the advantages of the proposed system. They are as follows:

1. Here we use multiple cipher text keys for decrypting the data
2. It is very hard for the user to break the keys.
3. The proposed system is really complex and tough to break the encrypted data without having key permissions.

5. SOFTWARE PROJECT MODULES

Implementation is the stage where the theoretical design is converted into programmatically manner. In this stage we will divide the application into a number of modules and then coded for deployment. We have implemented the proposed concept on Java programming language with JEE as the chosen language in order to show the performance this proposed application. The front end of the application takes JSP,HTML and Java Beans and as a Back-End Data base we took My-SQL Server. The application is divided mainly into following 4 modules. They are as follows:

1. System Construction Module
2. Data Owner
3. Data User
4. Admin

Now let us discuss about each and every module in detail as follows:

5.1 System Construction Module

n this module we try to construct an system with following attributes like : Single admin with multiple data owners and multiple data users.Here the data owner try to upload the files into the cloud server and before he upload he try to encrypt the data with a polynomial based encryption and then try to add those files into cloud server.Here the data users are those who try to enter into their account and search for the files which are uploaded by the various data owners

5.2 Data Owner Module

In Data Owner module, Initially Data Owner must have to register their detail and admin will approve the registration by sending signature key and private key through email. After successful login he/she have to verify their login by entering signature and private key. Then data Owner can upload files into cloud server with Polynomial key generation. He/she can view the files that are uploaded in cloud by entering the secret file key.

5.3 End User Module

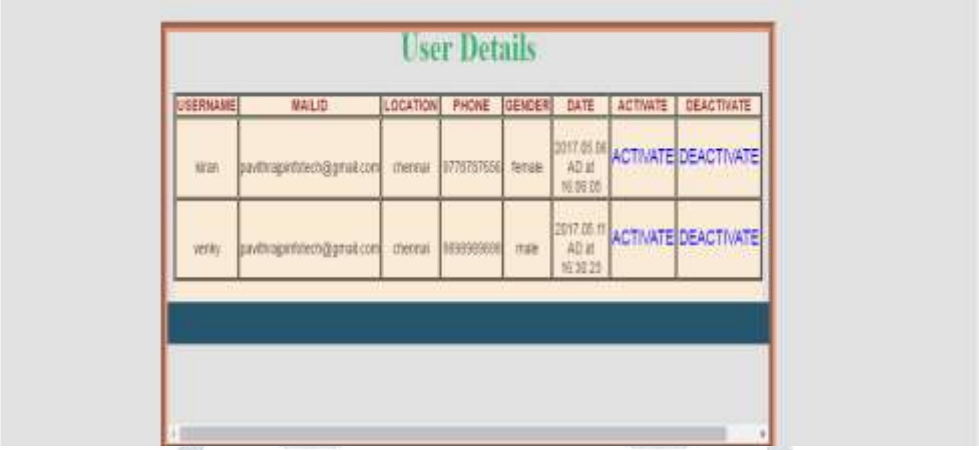
In Data User module, Initially Data Users must have to register their detail and admin will approve the registration by sending signature key and private key through email. After successful login he/she have to verify their login by entering signature and private key. Data Users can search all the files upload by data owners. He/she can send search request to admin then admin will send the search key. After entering the search key he/she can view the file

5.4 Admin Module

In Admin module, Admin can view all the Data owners and data user's details. Admin will approve the users and send the signature key and private key to the data owners and data users. Also admin will send the search request key to the users. Admin can able see the files in cloud uploaded by the data owners.

6. EXPERIMENTAL REPORTS

1) ADMIN MAIN PAGE

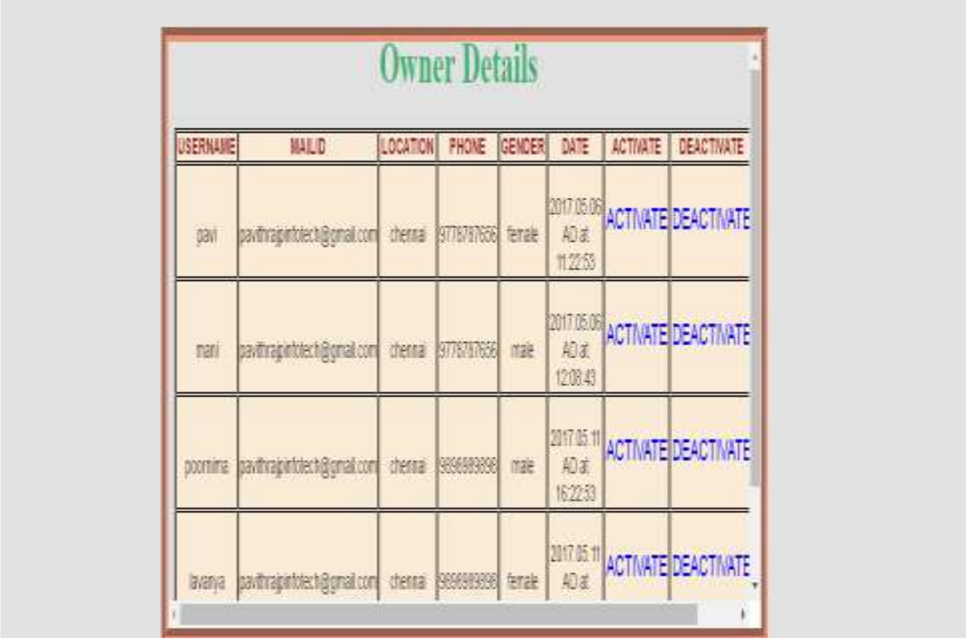


User Details

USERNAME	MAILID	LOCATION	PHONE	GENDER	DATE	ACTIVATE	DEACTIVATE
kiran	pavithrajpinfotech@gmail.com	chennai	9778787656	female	2017.05.06 AD at 16:06:05	ACTIVATE	DEACTIVATE
venky	pavithrajpinfotech@gmail.com	chennai	9898989898	male	2017.05.11 AD at 16:30:23	ACTIVATE	DEACTIVATE

Represents the Admin Views all User Details

2) CLOUD SERVER VIEW ALL OWNER DETAILS

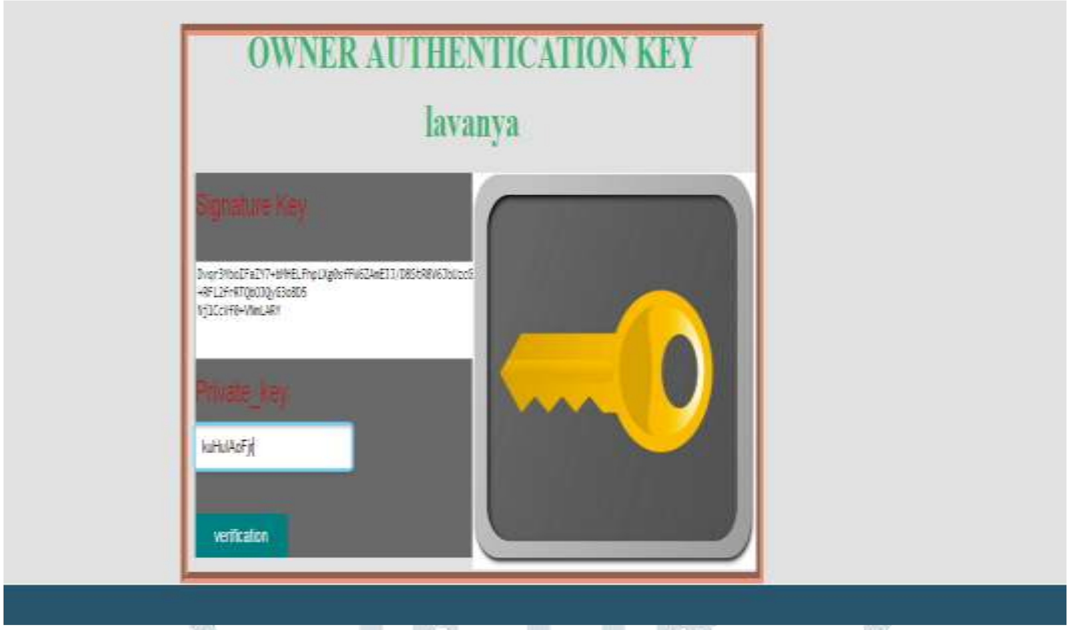


Owner Details

USERNAME	MAILID	LOCATION	PHONE	GENDER	DATE	ACTIVATE	DEACTIVATE
pavi	pavithrajpinfotech@gmail.com	chennai	9776797656	female	2017.05.06 AD at 11:22:53	ACTIVATE	DEACTIVATE
mani	pavithrajpinfotech@gmail.com	chennai	9776797656	male	2017.05.06 AD at 12:08:43	ACTIVATE	DEACTIVATE
poornima	pavithrajpinfotech@gmail.com	chennai	9696989898	male	2017.05.11 AD at 16:22:53	ACTIVATE	DEACTIVATE
lavanya	pavithrajpinfotech@gmail.com	chennai	9696989898	female	2017.05.11 AD at	ACTIVATE	DEACTIVATE

Represents the Cloud ADMIN Authorize Owners

3) OWNER AUTHENTICATE ITS KEY



OWNER AUTHENTICATION KEY

lavanya

Signature Key

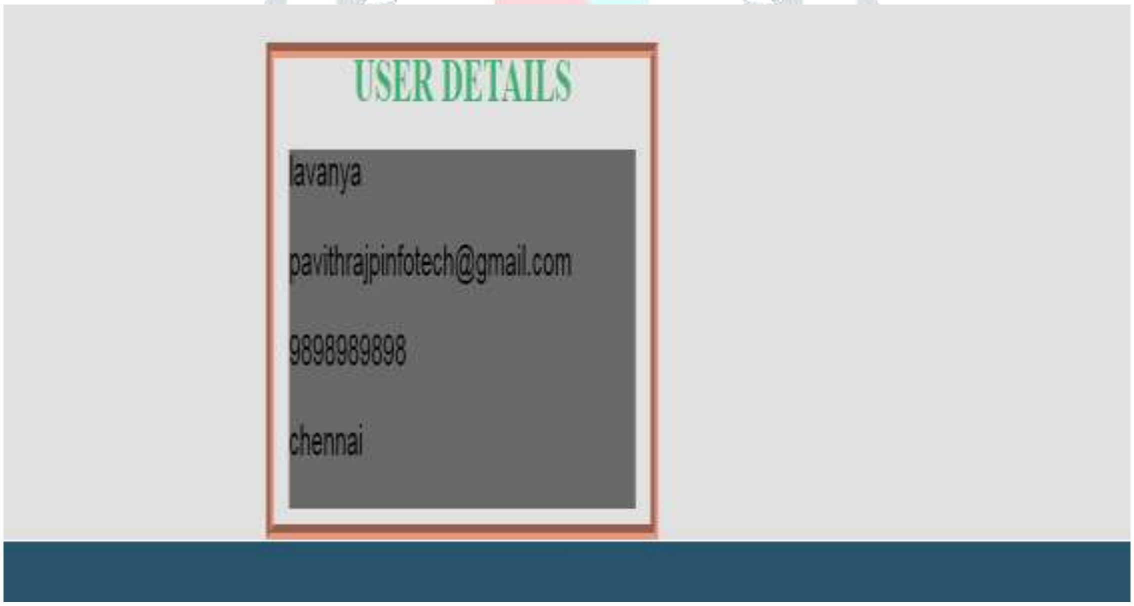
Dvqr3YbsIFaZY7+WMHELFnp1Xg0sfFW6ZAmEI3/D8StR8V6JbUzcG
+RFL2H+RTQbO3QyE3s8D5
NjICcHF0+VWmLARY

Private_key

kuHulAoFjx

Represents the OWNER AUTHENTICATION

4)USER LOGIN



USER DETAILS

lavanya

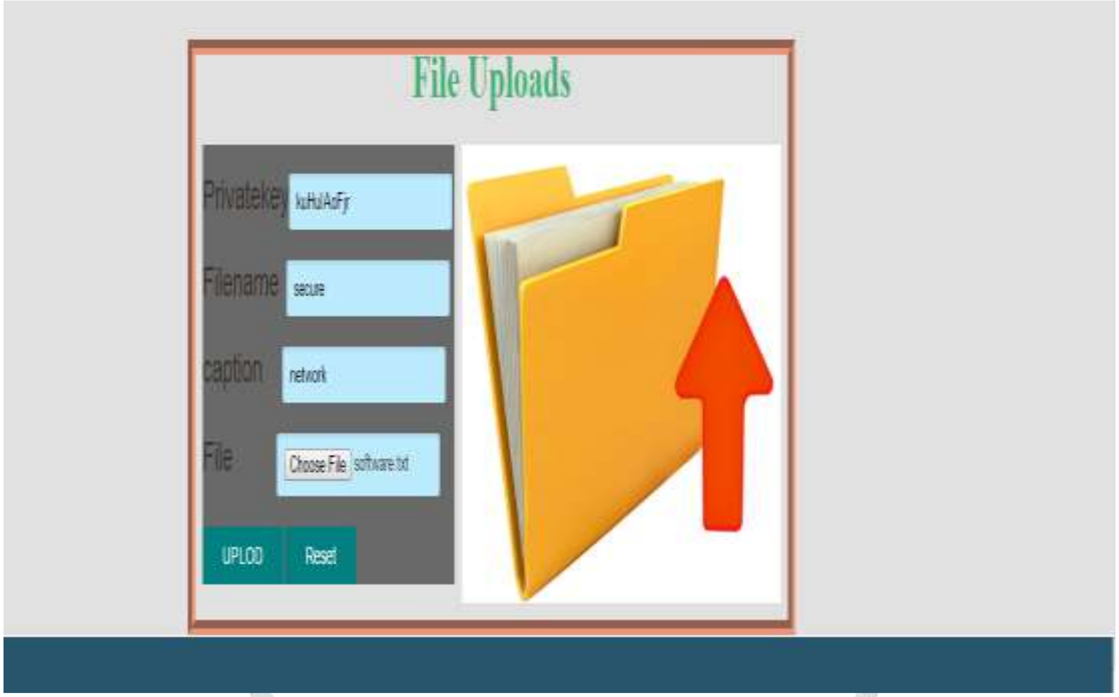
pavithrajpinfotech@gmail.com

9898989898

chennai

Represents the USER LOGIN

5) OWNER UPLOAD A FILE



File Uploads

Privatekey: kuHuiAoFjr

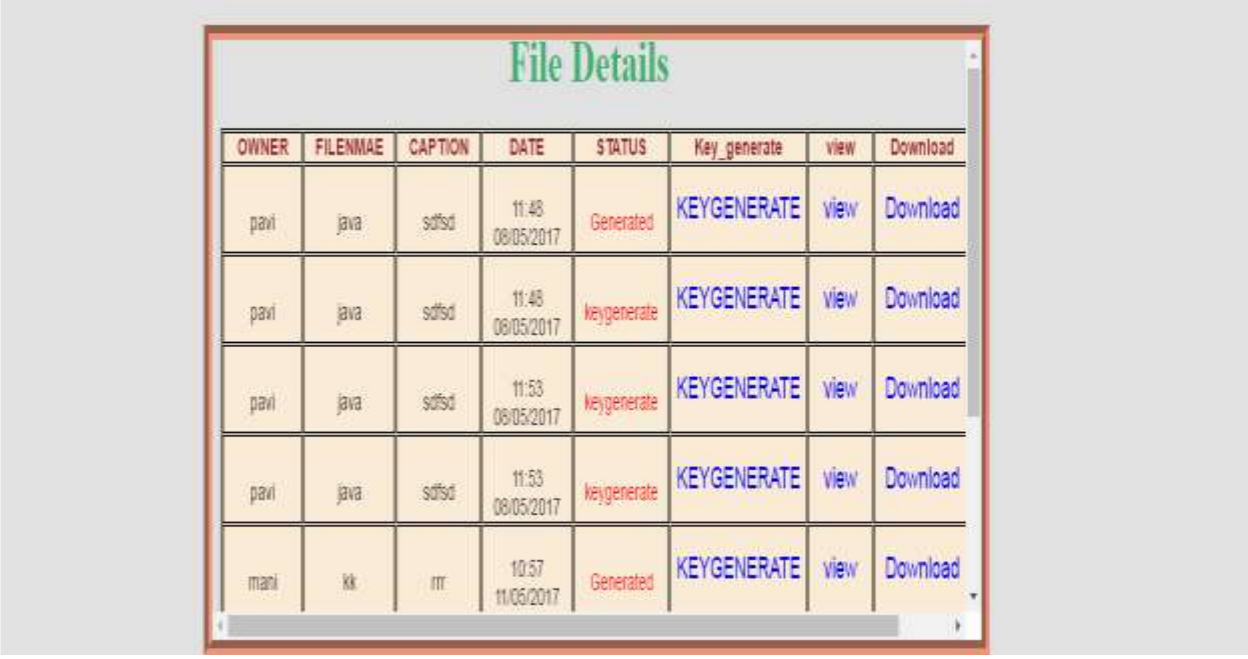
Filename: secure

caption: network

File: Choose File software.txt

Represents the OWNER FILE UPLOAD

6) OWNER CAN VIEW ALL FILE DETAILS

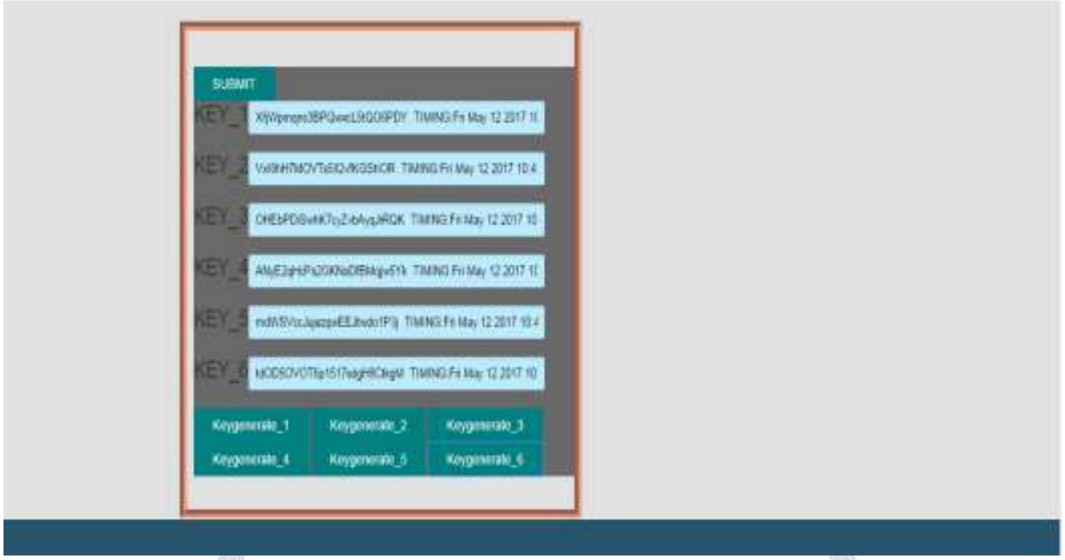


File Details

OWNER	FILENMAE	CAPTION	DATE	STATUS	Key_generate	view	Download
pavi	java	sdfsd	11:48 08/05/2017	Generated	KEYGENERATE	view	Download
pavi	java	sdfsd	11:48 08/05/2017	keygenerate	KEYGENERATE	view	Download
pavi	java	sdfsd	11:53 08/05/2017	keygenerate	KEYGENERATE	view	Download
pavi	java	sdfsd	11:53 08/05/2017	keygenerate	KEYGENERATE	view	Download
mani	kk	rrr	10:57 11/06/2017	Generated	KEYGENERATE	view	Download

Represents the ALL FILE DETAILS

7) CLOUD GENERATES POLYNOMIAL TIME SERIES KEYS



KEY_1 XfjWpmqesJBPQwcL9GOXPDY TIMING:Fri May 12 2017 1(

KEY_2 VxI9hH7MOVTsEl2vlKGStiOR TIMING:Fri May 12 2017 10:4

KEY_3 OHEbPDiSwhK7cyZxbAvqJRQK TIMING:Fri May 12 2017 10

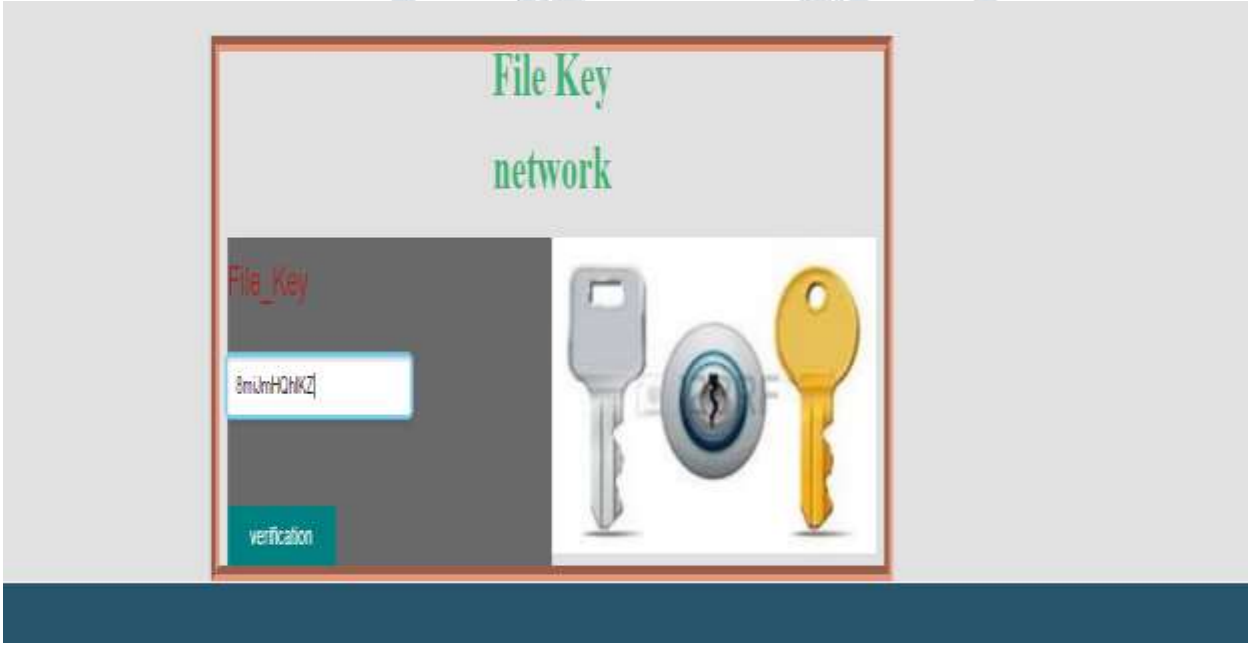
KEY_4 ANyE3qHrPu2GKNsDfBbMgv5Yk TIMING:Fri May 12 2017 1(

KEY_5 mdWSVbcJqazqxEELhvdo1P'j TIMING:Fri May 12 2017 10:4

KEY_6 ktODSOVOTEp1517edgH8CkgM TIMING:Fri May 12 2017 10

Represents the TIME SERIES KEYS

8) FILE KEY ACCESS

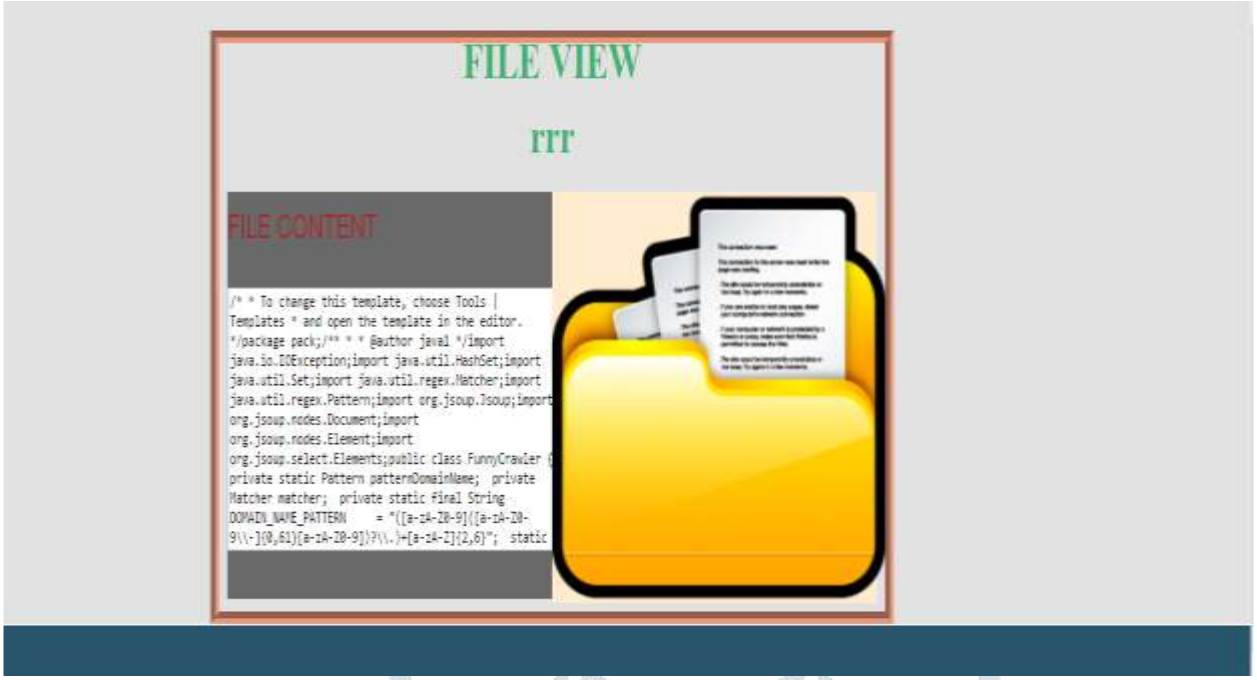


File_Key

8miJmHQhKZ

represents the FILE KEY ACCESS

9) **IF USER GIVE CORRECT KEY SEQUENCE**



```
/* * To change this template, choose Tools |
Templates * and open the template in the editor.
*/package pack;/** * * @author java1 */import
java.io.IOException;import java.util.HashSet;import
java.util.Set;import java.util.regex.Matcher;import
java.util.regex.Pattern;import org.jsoup.Jsoup;import
org.jsoup.nodes.Document;import
org.jsoup.nodes.Element;import
org.jsoup.select.Elements;public class FunnyCrawler {
private static Pattern patternDomainName;  private
Matcher matcher;  private static final String
DOMAIN_NAME_PATTERN     = "([a-zA-Z0-9]([a-zA-Z0-
9\\-]{0,61}[a-zA-Z0-9])?\\.)+[a-zA-Z]{2,6}";  static
```

Represents the file can be viewed in plain text manner

7. CONCLUSION

In this proposed work, we addressed the problem of securing data outsourced to the cloud against an adversary which has access to the encryption key. For that purpose, we introduced a novel security definition that captures data confidentiality against the new adversary. Here we proposed a time series keys so that those who hold all the multiple keys can able to decrypt and download the data and if the intruder try to gain illegal key access ,he need to know the sequence of that keys for decrypting the data.

8. REFERENCES

[1] M. Abd-El-Malek, G. R. Ganger, G. R. Goodson, M. K. Reiter, and J. J. Wylie, “Fault-Scalable Byzantine Fault-Tolerant Services,” in ACM Symposium on Operating Systems Principles (SOSP), 2005, pp. 59–74.

[2] M. K. Aguilera, R. Janakiraman, and L. Xu, “Using Erasure Codes Efficiently for Storage in a Distributed System,” in International Conference on Dependable Systems and Networks (DSN), 2005, pp. 336–345.

[3] W. Aiello, M. Bellare, G. D. Crescenzo, and R. Venkatesan, “Security amplification by composition: The case of doublyiterated, ideal ciphers,” in Advances in Cryptology (CRYPTO), 1998, pp. 390–407.

[4] C. Basescu, C. Cachin, I. Eyal, R. Haas, and M. Vukolic, "Robust Data Sharing with Key-value Stores," in ACM SIGACTSIGOPS Symposium on Principles of Distributed Computing (PODC), 2011, pp. 221–222.

[5] A. Beimel, "Secret-sharing schemes: A survey," in International Workshop on Coding and Cryptology (IWCC), 2011, pp. 11–46.

[6] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, "DepSky: Dependable and Secure Storage in a Cloud-ofclouds," in Sixth Conference on Computer Systems (EuroSys), 2011, pp. 31–46.

[7] G. R. Blakley and C. Meadows, "Security of ramp schemes," in Advances in Cryptology (CRYPTO), 1984, pp. 242–268.

[8] V. Boyko, "On the Security Properties of OAEP as an Allor-nothing Transform," in Advances in Cryptology (CRYPTO), 1999, pp. 503–518.

[9] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, "Deniable Encryption," in Proceedings of CRYPTO, 1997.

[10] Cavalry, "Encryption Engine Dongle," http://www. cavalrystorage.com/en2010.aspx/.

[11] C. Charnes, J. Pieprzyk, and R. Safavi-Naini, "Conditionally secure secret sharing schemes with disenrollment capability," in ACM Conference on Computer and Communications Security (CCS), 1994, pp. 89–95.

[12] A. Desai, "The security of all-or-nothing encryption: Protecting against exhaustive key search," in Advances in Cryptology (CRYPTO), 2000, pp. 359–375.

[13] C. Dubnicki, L. Gryz, L. Heldt, M. Kaczmarczyk, W. Kilian, P. Strzelczak, J. Szczepkowski, C. Ungureanu, and M. Welnicki, "HYDRAstor: a Scalable Secondary Storage," in USENIX Conference on File and Storage Technologies (FAST), 2009, pp. 197–210.

[14] M. Dürmuth and D. M. Freeman, "Deniable encryption with negligible detection probability: An interactive construction," in EUROCRYPT, 2011, pp. 610–626.

[15] EMC, "Transform to a Hybrid Cloud," http://www.emc. com/campaign/global/hybridcloud/index.htm.

[16] IBM, “IBM Hybrid Cloud Solution,” http://www-01.ibm. com/software/tivoli/products/hybrid-cloud/.

[17] J. Kilian and P. Rogaway, “How to protect DES against exhaustive key search,” in Advances in Cryptology (CRYPTO), 1996, pp. 252–267.

[18] M. Klonowski, P. Kubiak, and M. Kutylowski, “Practical Deniable Encryption,” in Theory and Practice of Computer Science (SOFSEM), 2008, pp. 599–609.

[19] H. Krawczyk, “Secret Sharing Made Short,” in Advances in Cryptology (CRYPTO), 1993, pp. 136–146.