Volume No:2, Issue No:1 (June-2016) ISSN No : 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

Data Sharing Among Multiple Groups Using Encryption in Cloud

R.Sinduja, M.E
Student,
Dept of CSE,
Dhanalakshni College of Engineering,
Chennai, Tamilnadu, India.

C.Kayalvizhi B.E, M.E
Assistant Professor,
Dept of CSE,
Dhanalakshni College of Engineering,
Chennai, Tamilnadu, India.

Abstract:

With the character of low maintenance, cloud computing provides an inexpensive and economical resolution for sharing cluster resource among cloud users. Sadly, sharing data in extremely during a exceedingly multi-owner manner.Whereas preserving data and identity privacy from an un-trusted cloud continues to be a troublesome issue, due to the frequent modification of the membership. Throughout this paper, we've an inclination to propose a secure multi owner knowledge sharing theme, named Mona, for dynamic groups inside the cloud. By investment cluster signature and dynamic broadcast secret writing techniques, any cloud user can anonymously share data with others. Meanwhile, the storage overhead and encryption computation worth of our theme square measure freelance with the number of revoked users. in addition, we've an inclination to research the protection of our theme with rigorous proofs, and demonstrate the efficiency of our theme in experiments.

Keywords:

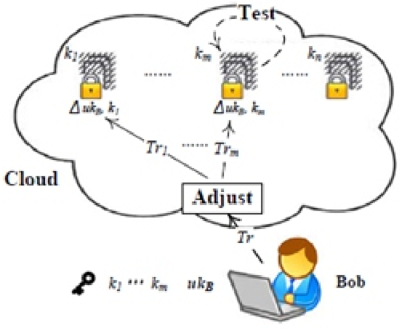
broadcast, encryption, signature.

INTRODUCTION:

CLOUD computing is recognized as AN alternate to ancient data technology due to its intrinsic resource-sharing and low-maintenance characteristics. In cloud computing, the cloud service suppliers (CSPs), like Amazon, area unit able to deliver various services to cloud users with the help of powerful knowledge centres. By migrating the native info management systems into cloud servers, users can relish high-quality services and save important investments on their native infrastructures. One in all the foremost basic services offered by cloud suppliers is info storage. Permit United States to require under consideration a wise info application. A company permits its staffs inside an equivalent cluster or department to store and share files inside the cloud. By utilizing the cloud, the staffs could also be totally discharged from the troublesome native info storage and maintenance. However, it in addition poses a serious risk to the confidentiality of these keeps files. Specifically, the cloud servers managed by cloud suppliers do not appear to be whole positive by users whereas the information files keep inside the cloud might even be sensitive and confidential, like business plans. To preserve info privacy, a basic resolution is to cipher info files, so transfer the encrypted info into the cloud. Sadly, coming up with academic degree economical and secure info sharing theme for teams inside the cloud is not a simple task due to the following difficult issues.First, identity privacy is one in all the foremost very important obstacles for the wide activity of cloud computing.

While not the guarantee of identity privacy, users are unwilling to hitch in cloud computing systems as a results of their real identities could also be merely disclosed to cloud suppliers and attackers. On the other hand, unconditional identity privacy might incur the abuse of privacy. As AN example, misbehaved staff can deceive others inside the corporate by sharing false files whereas not being traceable. Therefore, traceability, that allows the cluster manager (e.g., an organization manager) to reveal the necessary identity of a user, is additionally extraordinarily fascinating. Second, it's extraordinarily advised that any member throughout a bunch got to be ready to fully fancy the data storing and sharing services provided by the cloud that's printed because the multiple-owner manner.

Cloud computing may be a virtual, scalable, versatile open supply technology. And it should be an excellent price savings within the cloud, wherever our servers run on native servers that you simply share the data with alternative customers.



BACKGROUND OF PROBLEM:

Suppose that Client 1 uploads all her private pictures and videos on Dropbox, and she does not want to see her photos by everyone. Due to various data leakages in cloud there may be possibility that client 1 cannot feel satisfied by just relying on the privacy protection provided by Dropbox, so she encrypts all the pictures using her own keys before uploading. One day, Client 1's friend, say client 2, asks her to share her pictures taken during all these years which client 2 appeared in. client 1 then uses the share function of Dropbox, but the problem is how to delegate the decryption rights for these pictures to client 2. A possible option client 1 can choose is to securely send client 2 the secret keys included .Therefore there are two ways for her under the traditional encryption paradigm: 1)client 1 encrypts all files with a single encryption key and gives client 2 the corresponding secret key directly. 2)client 1encrypts files with distinct keys and sends client 2 the corresponding secret keys surely, the first technique is inadequate since all data which is not yet chosen may be also leaked to client 2. For the second method, there are practical concerns on efficiency. The number of keys is equivalent to the number of the shared photos, say, a thousand. Sending these secret keys requires a more secure channel, and storage of these keys requires expensive secure storage. The cost and complexities included generally rise with the number of the decryption keys to be shared. In short, it is much heavy and costly to do[2]

VARIOUS SEARCHABLE ENCRYPTION SCHEMES AND THEIR RELATIONSHIP TO OUR WORK:

A.Multi-user Searchable Encryption(MUSE):

There is a large amount of literature on searchable encryption, including SSE and PEKS 's schemes . In contrast to those existing schemes, in the cloud storage, keyword search under the multi-tenancy is a more used scenario. In such a scenario, the data owner will to share a document with a group of authorized users, and each user who has the access authority can provide a trapdoor to perform the process of keyword search over the shared document, namely, the multiple-users searchable encryption (MUSE) scenario [1]. Schemes are created by sharing the documents searchable encryption key with all users who have access on it, and broadcast encryptions used to reach coarse-grained access control. As a result, in MUSE, the big problem is how to manage which users can access which documents, whereas how to decrease the number of shared keys and trapdoors is not taken in account. Key aggregate searchable encryption can provide efficient solution and it can make MUSE more efficient and practical.

B.Multi-Key Searchable Encryption(MKSE):

In this ,the number of trapdoors is equivalent to the number of documents to search over the documents (if user provides to the server a keyword trapdoor under every key along which a matched document can be encrypted). The objective of MKSE is to assure the cloud service provider can perform keyword search by using only one trapdoor over different documents, whereas the objective of Key Aggregate Searchable Encryption is delegate the right of keyword search to any user by distributing the aggregate key to user in a group data sharing system[1].

C. Searchable symmetric encryption (SSE):

It allows a client to encrypt its data in such a way that this data can get searched still. The most significant application of SSE to the cloud storage is where it enables a client to securely transfer its data to an untrusted cloud provider without losing the ability to search over it[1].SSE is active research and various functionalities of schemes can achieve various levels of security and efficiency. Any practical SSE scheme, however, should satisfy the following properties: sub linear searching time, security, indexes and the ability to modify files efficiently [7].

Previous existing-known SSE schemes cannot achieve all these properties at the simultaneously. This limits the practical value of SSE and reduces its chance of deployment in real-world cloud storage system.

D. Attribute Based encryption (ABE):

It contains every ciphertext to be associated with an attribute, and the master-secret key holder can be extract a secret key for a policy of these attributes so that the ciphertext can be decrypted by this key if its associated attribute confirms to the policy. In this technique the user's secret key and ciphertext is dependent on attributes[2].

EXISTING SYSTEM:

The existing system of cloud storage blogger will let their friends read subsets of their personal info AN enterprise might grant his/her staff access to some of data or information. The difficult drawback is a way to effectively share encrypted knowledge. Users will transfer the encrypted knowledge from the storage unit, and rewrite them, then send them to others for sharing the info; however it will loses the worth of cloud storage knowledge. Users ought to be ready to delegate the access rights of the sharing knowledge to others so they'll access this knowledge directly from the server. However, finding economical and secure thanks to share partial knowledge in cloud storage isn't trivial. The receiver decrypting the initial Message mistreatment cruciform key algorithmic rule. With a lot of mathematical tools and crypto logic ways have gotten extremely versatile and involve several variety of keys for one application meaning there a may be a doable of forgetting the keys in an exceedingly application.

DISADVANTAGE:

* Increases the prices of storing and transmitting cipher texts.
* Secret keys square measure typically holds on within the tamper-proof memory that is comparatively valuable.
* This may be a versatile approach.
* The prices and complexities involve usually which will increase with the quantity of the decoding keys to be shared.

PROPOSED SYSTEM:

In this paper, we've an inclination to make a cryptography key as lots of powerful inside the sense that it permits cryptography of multiple cipher texts, whereas not increasing its size. we've an inclination to unit of measurement introducing a public-key encryption that we've an inclination to call key-aggregate cryptosystem they practice AES formula. In kac, users write a message not exclusively below a public-key, but put together below Associate in nursing image of cipher text referred to as class. Which suggests the cipher texts unit of measurement any classified into whole completely different categories? The key owner holds a master-secret referred to as master-secret key, which can be accustomed extract secret keys for numerous classes. Lots of considerably, the extracted key have is Associate in nursing mixture key that's as compact as a secret key for one class, but aggregates the power of the numerous such keys, i.e., the cryptography power for any set of cipher text classes.

ADVANTAGES:

* The delegation of decoding method will be expeditiously enforced with the mixture key, that is merely of mounted size.
* Number of cipher text categories is massive. It is straightforward to key management for secret writing and decoding

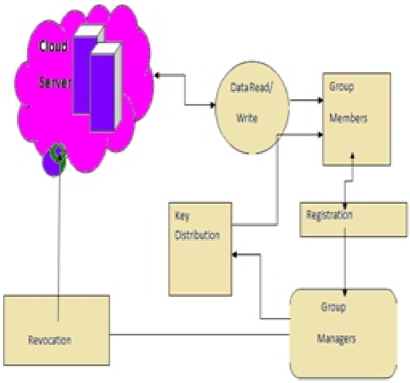


Fig: 1 Architecture Diagram

LITRETURE SURVEY:

1) Scalable Hierarchical Access Control in Secure Group Communications

Several cluster communications want a security infrastructure that maintains a lot of levels of access privilege for cluster members.Access management in hierarchy is rife in transmission applications, that carries with it users that take utterly different quality levels or different sets of knowledge streams. During this paper, we’ve an inclination to gift a multi-group key management theme that achieves such a hierarchical access management by mistreatment AN integrated key graph Associate in Nursing by managing cluster keys for all users with varied access schemes. Compare with applying existing tree-based cluster key management schemes on to the hierarchical access management drawback, the planned them considerably reduces the communication price, process and storage overhead associated with key management and achieves higher quality once the amount of access levels can increase. Additionally, the planned key graph is acceptable for every centralized and tributary environment.

2) Plutus: Scalable secure file sharing on untrusted storage

This paper has introduced novel uses of crypto logic primitives applied to the matter of secure storage within the presence of un-trusted servers and a want for owner managed key aggregation. Eliminating all reserve necessities for server trust (we still need servers to not destroy knowledge on server– though we will sight if they do) and keeping key distribution (and so access control) within the hands of individual knowledge house owners provides a basis for a secure storage system services which will defend and share knowledge at terribly massive scale and across trust boundaries.

3) SiRiUS: Securing Remote Untrusted Storage

This paper presents Canicula, a secure filing system designed to be stratified over insecure network and purpose a pair of purpose file systems like Network file systemFS, cifs, Ocean Store, and yahoo, briefcase. Canicula assumes the network storage service is untrusted and provides its own read-write crypto logic access management for file level sharing.Key management theme and revocation is straightforward with bottom band communication. Filing system guarantees square measure supported by Canicula mistreatment hash tree constructions. Canicula contains a completely unique methodology for performing arts file random access in an exceedingly crypto logic filing system while not the employment of a block server.

4) Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing

During this paper planned theme is characterised by providing the data confidentiality on sensitive documents hold on in cloud, anonymous authentication on user access, and root following on controversial documents. With the demonstrable security techniques, we tend to formally demonstrate the planned theme is secure within the normal model.

5) Cipher text-Policy Attribute-Based Encryption: An Expressive, E_cient, and Provably Secure Realization

This Paper gift a greenhorn methodology for realizing Cipher text-Policy Attribute secret writing (CP- ABE) below concrete and non interactive science assumptions inside the traditional model. Our solutions alter any encryptor to specify access management in terms of any access formula over the attributes inside the system. In our most e_cient system, cipher text size, encryption, and writing time scales linearly with the standard of the access formula. The only previous work to comprehend these parameters was restricted to a sign inside the generic cluster model.

6) Key-Aggregate Cryptosystem for Scalable Data Sharing In Cloud Storage

during this paper, we’ve an inclination to ponder the thanks to “compress” secret keys in public-key cryptosystems that support delegation of secret keys for numerous cipher text classes in cloud storage. Withal that one all told the power set of classes, the delegate can forever get academic degree mixture key of constant size. Our approach is extra versatile than stratified key assignment which can exclusively save areas if all key-holders share a consistent set of privileges. A limitation in our work is that the predefined sure of the number of most cipher text classes.

In cloud storage, the number of cipher texts generally grows quickly.

APPROACHES:

AES:

Rijndael is a block cipher. What this means is that messages are broken into blocks of a predetermined length, and each block is encrypted independently of the others. Rijndael operates on blocks that are 128-bits in length. There are actually 3 variants of the Rijndael cipher, each of which uses a different key length. The permissible key lengths are 128, 192, and 256 bits. Even the smallest of these is large enough to prevent any exhaustive search. Of course, a large key is no good without a strong design. The details of Rijndael may be found in [12], but we give an overview here. 2.1. Mathematical Preliminaries. Within a block, the fundamental unit operated upon is a byte, that is, 8 bits. Bytes are thought of in two different ways in Rijndael. Let the byte be given in terms of its bits as b7b6 . . . b0. We may think of each bit as an element in GF(2), the finite field of two elements. First, one may think of a byte as a vector, (b7, b6. . . b0) in GF(2)8 . Second, one may think of a byte as an element of GF(28), in the following way: Consider the polynomial ring GF(2)[X]. We may mod out by any polynomial to produce a factor ring. If this polynomial is irreducible, and of degree n, then the resulting factor ring is isomorphic to GF(2n). In Rijndael, we mod out by the irreducible polynomial X8 + X4 + X3 + X + 1, and so obtain a representation for GF(28). A byte is then represented in GF(28) by the polynomial b7X7 + b6X6 + . . . + b0. It is also convenient to refer to bytes (in either setting) by their hexadecimal representations. Of course, we may then define polynomial rings over GF(28). Later on, the ring GF(28)[Y]/(Y 4 + 1) we be used. We note that while this is not a field (as Y 4 + 1 is not irreducible in GF(28)[Y], being equal to (Y + 1)4), elements are invertible if they are coprime to Y 4 + 1, that is, if they are not divisible by Y + 1. 2.2. The State. For simplicity, we limit ourselves to describing Rijndael with a 128-bit key. The other variants are essentially the same. Operations are done on intermediate results known as the state. The state is 128-bits long. We think of the state as divided into 16 bytes, a(i,j) where 0 ≤ i, j ≤ 3. We think of these 16 bytes as an array, or matrix, with 4 rows and 4 columns, like so: [a(0,0) a(0,1) a(0,2) a(0,3) a(1,0) a(1,1) a(1,2) a(1,3) a(2,0) a(2,1) a(2,2) a(2,3) a(3,0) a(3,1) a(3,2) a(3,3)]

The state starts out as the 128-bit input. We operate on the state by performing successive rounds. A round is made up of three parts: application of the S-box, linear diffusion, and subkey addition. We discuss each part below. 2.3. The S-Box. S-boxes, or substitution boxes, are common in block ciphers. These are objective functions on the blocks that are, ideally, highly non-linear. Much of the security of block ciphers can be thought of as 'residing' in their S-boxes. In AES, the S-box has a relatively ALGEBRAIC CRYPTANALYSIS OF AES: AN OVERVIEW 5 simple form. The S-box is the same in every round, and it acts independently on each byte. It has two parts. For the first part, we think of each byte as living in GF(28). We then simply apply the 'patched inverse'. This sends a byte a to a −1 if a is non-zero, and sends it to 0 if it is zero. This can also be expressed as sending a 7→ a 254. This inversion is actually optimal with respect to several measures of non-linearity, and non-linearity is important to protect against several common families of attack. For the second part, we apply an affine (over GF(2)) transformation. Think of the byte a as a vector in GF(2)8 . Consider the invertible matrix A, [1 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1] . Much like inversion, the structure of A is relatively simple, successively shifting the prior row by 1. If we define the vector v GF(2)8 to be (1, 1, 0, 0, 0, 1, 1, 0), then the second half of the S-box sends a byte a to A • a + v T . As a whole then, the action of the S-box is a(i,j) 7→ A • a −1 (i,j) + v T . 2.4. Linear Diffusion. Next, we apply two different linear maps to 'mix' the state. The first map is the RowShift. Here, we simply shift the rows around. The first row is unchanged, the second row is shifted to the left by 1, the second by 2, the third by 3. Graphically, if the state after the S-box step is denoted by the matrix (a(i,j)), the new state is [a(0,0) a(0,1) a(0,2) a(0,3) a(1,1) a(1,2) a(1,3) a(1,0) a(2,2) a(2,3) a(2,0) a(2,1) a(3,3) a(3,0) a(3,1) a(3,2)] . The second step, the MixColumn transformation, not surprisingly mixes the columns. We do more, however, than just move around bytes within the columns. We interpret the bytes of each column as the coefficients of a polynomial in GF(28)[Y]/(Y 4 + 1). Then, we multiply each column by the polynomial '03'Y 3 + '02'Y 2 + '01'Y + '02' (which is invertible in our ring) and reduce appropriately. 6 HARRIS NOVER Each step, and hence their composition, is linear, whether viewed over GF(2) or GF(28). Note that in the last round, for reasons of efficiency in decrypting, we will leave out the column mixing. 2.5. Subkey Addition.

From the original key, we produce a succession of 128-bit keys, by means of a key schedule. The details of the key schedule need not concern us here; we simply note that later round keys are produced from earlier round keys by applications of the S-box above and by XORing prior round keys together. Each 128-bit round key may then be divided into bytes, and the bytes placed in a 4x4 matrix. We refer to the (i, j)th byte of the mth round key by km,(i,j) . Then in round m we replace byte a(i,j) of the current state with a(i,j)km,(i,j) . 2.6. Putting It Together. The Rijndael algorithm is then as follows. Put the input into the state. XOR the state with the 0-th round key. We start with this because any actions before the first (or after the last) use of the key are pointless, as they are publicly known and so can be undone by an attacker. Then, apply 10 of the above rounds, skipping the column mixing on the last round (but proceeding to a final key XOR in that round). The resulting state is the ciphertext.

Procedure: EncryptString ()
1) Create Function called “ EncryptString ()”
2) Make Exceptional Try...Catch block
3) Dimensionate the variables upto n depends on your needs.
4) Assign the value to the declarations.
5) Dimensionate unstring as String
6) Write Clear() function to clear all the items available in the Unique Item List
7) Add the first index item into List
8) Create Looping Statements for identifying unique items
Ex:

```
Dim RijndaelCipher As New RijndaelManaged()
Dim PlainText As Byte() = System.Text.Encoding.Unicode.GetBytes(InputText)
Dim Salt As Byte()=Encoding.ASCII.GetBytes(Password.Length.ToString())
Dim SecretKey As New PasswordDeriveBytes(Password, Salt)
Dim Encryptor As ICryptoTransform = RijndaelCipher.CreateEncryptor(SecretKey.GetBytes(16),
SecretKey.GetBytes(16))
Dim memoryStream As New IO.MemoryStream()
Dim cryptoStream As New CryptoStream(memoryStream, Encryptor, CryptoStreamMode.Write)
cryptoStream.Write(PlainText, 0, PlainText.Length)
cryptoStream.FlushFinalBlock()
Dim CipherBytes As Byte() = memoryStream.ToArray()
memoryStream.Close()
cryptoStream.Close()
Dim EncryptedData As String = Convert.ToBase64String(CipherBytes)
Return EncryptedData
```

Procedure: DecryptString ()
1) Create Function called “DecryptString ()”
2) Make Exceptional Try...Catch block
3) Dimensionate the variables upto n depends on your needs.
4) Assign the value to the declarations.
5) Dimensionate unstring as String
6) Write Clear() function to clear all the items available in the Unique Item List
7) Add the first index item into List
8) Create Looping Statements for identifying unique items
Ex:

```
Dim RijndaelCipher As New RijndaelManaged()
Dim EncryptedData As Byte() = Convert.FromBase64String(InputText)
Dim Salt As Byte()=Encoding.ASCII.GetBytes(Password.Length.ToString())
Dim SecretKey As New PasswordDeriveBytes(Password, Salt)
Dim Decryptor As ICryptoTransform = RijndaelCipher.CreateDecryptor(SecretKey.GetBytes(16),
SecretKey.GetBytes(16))
Dim memoryStream As New IO.MemoryStream(EncryptedData)
Dim cryptoStream As New CryptoStream(memoryStream, Decryptor, CryptoStreamMode.Read)
Dim PlainText As Byte() = New Byte(EncryptedData.Length - 1) {}
Dim DecryptedCount As Integer = cryptoStream.Read(PlainText, 0, PlainText.Length)
memoryStream.Close()
cryptoStream.Close()
Dim DecryptedData As String = Encoding.Unicode.GetString(PlainText, 0, DecryptedCount)
Return DecryptedData
Catch exception As Exception
Return (exception.Message)
End Try
```

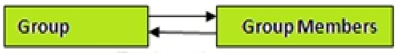
PAST RESEARCH SURVEY

A sequence of searchable symmetric encryption proposal has been projected to facilitate search on cipher text. Long established proposals make possible users to steadily repossess the cipher text, excluding these proposals shore up only Boolean keyword search, that is, whether a key exists in a system or not, without considering the difference of relevance with the queried keys of these encrypted data in the result. Preventing the security from involving in ranking and entrusting work to the user is a natural way to avoid information leakage. However, the limited computational power on the user side and the high computational overhead against information security.

RELATED WORK:

1.User Registration:

For the registration of a user with establish the ID the cluster managers arbitrarily selects with variety. Then the cluster managers add into the cluster user to list that is employed within the traceability state. Once complete the registration of a user, user obtains a key through mail which can be used for cluster signature generation and file decoding.



Registration

2. User Revocation:

User revocation is performed by the cluster manager via a public keys square measure on the market. Revocation list supported that cluster members will write the info files and make sure the confidentiality against the revoked users. Cluster trough update the revocation list every day even no user has being revoked within the day. In alternative words, the others will verify the info of the revocation list from the contained current date.

Group
Cloud

Revocation

3. File Generation and Deletions:

To store and share file within the cloud, a bunch member performs to obtaining the revocation list from the cloud. During this method, the member sends the cluster identity ID to cluster as asking to the cloud. validatory the validity of the received revocation list. File hold on within the cloud will be deleted by either the cluster manager or the info owner.

4. File Access and Traceability:

To access the cloud, a user has to work out a bunch signature for his/her authentication. The used cluster signature theme will be considered a variant of the short cluster signature that inherits the inherent un-forge ability property, anonymous authentication, and following capability. Once a knowledge dispute happens, the tracing operation is performed by the cluster manager to spot the $64000 identity of the info owner.

SCOPE

1) This can be useful in cloud environment where large number of documents is needed to share in a secure way.
2) There is the practical problem of privacy preserving data sharing system based on public cloud storage server which requires a data owner to distribute a huge number of keys to users to enable them to access their documents, here we for the first time proposing the concept of key-aggregate searchable encryption (KASE) and construct a concrete and efficient KASE scheme.

CONCLUSION:

In this paper, we tend to tend to vogue a secure data sharing theme, Mona, for dynamic groups in associate un-trusted cloud. In Mona, a user is prepared to share data with others inside the cluster whereas not revealing identity privacy to the cloud. To boot, island supports economical user revocation and new user amendment of integrity. lots of specially, economical user revocation square measure usually achieved through a public revocation list whereas not amendment the private keys of the remaining users, and new users can directly rewrite files keep inside the cloud before their participation. Moreover, the storage overhead and so the cryptography computation worth unit of measurement constant. Intensive analyses show that our planned theme satisfies the specified security desires and guarantees efficiency equally. Planned a crypto graphical storage system that allows secure file sharing on un-trusted servers, named Plutus.

By dividing files into file teams and encrypting each file cluster with a completely unique file-block key, the information owner can share the file teams with others through delivering the corresponding safe-deposit key, where the safe-deposit secret is accustomed write the file-block keys. However, it brings some of great key distribution overhead for large-scale file sharing. To boot, the file-block key must be updated and distributed all over again for a user revocation.

REFERENCES:

1.Key-Aggregate Cryptosystem for ScalableData Sharing in Cloud StorageCheng-Kang Chu, Sherman S.M. Chow, Wen-Guey Tzeng, Jianying Zhou, andRobert H. Deng, Senior Member, IEEE

2.U.S. Department of Health and Human Services. (2011, Sep.). HIPAA—General Information [Online]. Available: https://www.cms.gov/ hipaageninfo

3.PCI Security Standards Council. (2006, Sep.) Payment Card Industry (PCI) Data Security Standard—Security Audit Procedures Version 1.1 [Online]. Available:https://www.pcisecuritystandards.org/pdfs/pci−audit−procedures−v1-1.pdf

4.Sarbanes-Oxley Act 2002. (2002, Sep.). A Guide to the Sarbanes-Oxley Act [Online]. Available: http://www.soxlaw.com/

5.C. Lonvick, the BSD Syslog Protocol, Request for Comment RFC 3164, Internet Engineering Task Force, Network Working Group, Aug. 2001.

6.6. K. Kent and M. Souppaya. (1992). Guide to Computer Security Log Management, NIST Special Publication 800-92 [Online]. Available:http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf

7.D. New and M. Rose, Reliable Delivery for Syslog, Request for Comment RFC 3195, Internet Engineering Task Force, Network Working Group, Nov. 2001.

8.8.G. Ateniese, K. Fu, M. Green, and S.ohenberger, "ImprovedProxy Re-Encryption Schemes with Applications to SecureDistributed Storage," ACM Trans. Information and System Security,vol. 9, no. 1, pp. 1-30, 2006.

9.D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant BroadcastEncryption with Short Ciphertexts and Private Keys," Proc.Advances in Cryptology Conf. (CRYPTO '05), vol. 3621, pp. 258-275,2005.

10.. L.B. Oliveira, D. Aranha, E. Morais, F. Daguano, J. Lopez, and R.Dahab, "Tiny Tate: Computing the Tate Pairing in Resource-Constrained Sensor Nodes," Proc. IEEE Sixth Int'l Symp. NetworkComputing and Applications (NCA '07), pp. 318-323, 2007.