iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** XII **Month of publication:** December 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Multi-level Security: Data Sharing Using Cryptography and Steganography in Cloud Computing

Deepika.J[1], Bharathi Dasan.V.S[2], Vidhya.K[3]
Sri Venkateswara College of Engineering and Technology, India

Abstract: ***In cloud computing, sharing of data is considered as a challenging security problem. The sharing of the user's critical data on a third party cloud server does not guarantee the promised level of security and there is a threat to compromise user data. In order to safeguard the data and improve the security measures in the consumer side, cryptographic encryption technique is used where aggregate key and steganography concepts are implemented.. This project aspires to provide secured data sharing in cloud storage with the help of steganography concepts in addition with the sharing of aggregate keys between data owners and data users.***

Keywords— Cloud storage, data sharing, aggregate key encryption, steganography.

I. INTRODUCTION

Cloud computing (so-called, cloud) represents one of the magnificent shifts in information technology which can enhance collaboration, agility, scaling and availability, and provide the potential for cost reduction through optimized and efficient computing Different from the existing technologies and computing approaches, cloud is defined with five essential characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service),SPI service models (Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS)), and deployment models (Public, Private, Hybrid, Community). However, security concern has become the biggest obstacle to adoption of cloud because all information and data (including real location of data, and security management level) are completely under the control of cloud service providers. For such a reason, CSA, ENISA and NIST published general security guidance and recommendations for the cloud usage in order to provide some level of protection ranging from physical security to network/system/application security. In March 2010, CSA announced the top threats in cloud computing which, if not properly secured, may cause devastating impacts on the mission-critical cloud services. Among those threats,

A. Data Loss or Leakage: Some examples to compromise data are
  1) operational failures (e.g., improper data deletion or alteration);
  2) inconsistent use of encryption and software keys (or loss of encryption keys);
  3) unreliable data storage; and
  4) Insufficient AAA controls to allow unauthorized parties to access sensitive data. The threat of data compromise is mainly due to the architectural or operational characteristics of the cloud environment.

B. Account or Service Hijacking: Account and service hijacking, usually with stolen credentials, remains a top threat. With the stolen credentials, an attacker can access critical areas of the deployed cloud services to compromise confidentiality, integrity, and availability of data and those services. If cloud service providers also provide SSO (Single Sign-On) or ID management services, the account and service hijacking would cause the collateral damage on those services as well.

Some general security guidance to deal with the above threats can be found in the following

C. Encryption and Key Management: Encryption provides data protection while key management enables access to protected data. It is strongly recommended to encrypt detain transit over networks, at rest, and on backup media. In particular, data encryption at rest (e.g., for long-term archival storage) can avoid the risk of malicious cloud service providers or malicious multi-tenants

abuse. At the same time, secure key stores (including key backup and recoverability) and access to key stores must be securely implemented since improper (or access to) key storage could lead to the compromise of all encrypted data.

D. Identity and Access Management: Secure management of identity and access control is a critical factor to prevent account and service hijacking. It is strongly recommended to prohibit sharing of account credentials, to leverage strong (multi-factor) authentication if possible, and to consider delegated authentication and managing trust across all types of cloud services. In order to guarantee security against 'Data Loss or Leakage' and 'Account or Service Hijacking' threats

The cloud computing encounters various security issues, as it comprises of many technologies namely, Networks, Database, virtualization, operating systems, Resource scheduling, Transaction management, Load balancing, Concurrency control, Memory management.

Therefore security issues for many of these systems and technologies are also applicable to cloud computing.

For example,

a) The network that interconnects the system in a cloud has to be secure.

b) Mapping the virtual machines to physical machine has to be carried out securely

c) Data security involves encrypting the data as well as ensuring that appropriate policies are enforced for data sharing.

Considering data privacy, a traditional way to ensure it is to rely on the server to enforce the access control after authentication, which means any unexpected privilege escalation will expose all data. In a shared-tenancy cloud computing environment, things become even worse. Data from different clients can be hosted on separate virtual machines (VMs) but reside on a single physical machine in a target VM could be stolen by instantiating another VM coresident with the target one. Regarding the availability of files, there are the series of cryptographic schemes which as go a s far as allowing the third party auditor to check the availability of files on behalf on the data owner without leakage of anything about the data or without compromising the data owner anonymity . A cryptographic solution, for example, with proven security relied on number-theoretic assumption is more desirable, whenever the user is not perfectly happy with trusting the security of the VM or the honesty of the technical staff. These are motivated to encrypt their data with their own keys before uploading them to server. Users can download the encrypted data from the storage, decrypt them, then send them to others for sharing, but it loses the value of cloud storage. users should be able to delegate the access rights of the sharing data to others so that they can access these data from the server directly. Drop box is an example .Assume that data owner puts all her private photos on drop box and she does not like to share her photos to everyone.

Due to data leakage just relying on the privacy protection mechanisms provided by drop box ,so she encrypts all the photos using her own keys before uploading .One day, Data owner friend ,i.e data user asks her to share the photos taken over all these years which data user appeared in .Data owner can then use the share function of drop box, but the problem now is how to delegate the decryption rights for these photos to data user. A possible option data owner can choose is to securely send data user secret keys involved. In a proxy re-encryption scheme a semi-trusted proxy converts a ciphertext for data owners into a ciphertext for data requestor without seeing the underlying plaintext. The fundamental property of proxy re-encryption schemes is that the proxy is not fully trusted.

Several efficient proxy re-encryption schemes that offer security improvements over earlier approaches. The primary advantage of our schemes is that they are unidirectional (i.e., Alice can delegate to Bob without Bob having to delegate to her) and do not require delegators to reveal all of their secret key to anyone – or even interact with the delegatee – in order to allow a proxy to re-encrypt their ciphertexts.

In these schemes, only a limited amount of trust is placed in the proxy. Identity management is one of the most common services deployed within companies and organizations because of its key role in access control, authorization and accountability processes. However, it introduces an overhead in cost and time, and in most cases it requires specific applications and personnel for managing, integrating and maintaining this service the cloud offers an innovative opportunity of externalizing this workload; this is what has been called Identity Management as a service (IDaaS or IDMaaS). In a proxy re-encryption (PRE) scheme, a proxy is given special information that allows it to translate a ciphertext under one key into a cipher text of the same message under a different key. The proxy cannot, however, learn anything about the messages encrypted under either key.

E. Our contributions

The main feature of the encryption/decryption program implementation is the generation of the encryption key. Now a day, cryptography has many commercial applications. If we are protecting confidential information then cryptography is provide high level of privacy of individuals and groups. However, the main purpose of the cryptography is used not only to provide confidentiality, but also to provide solutions for other problems like: data integrity, authentication, non-repudiation. Cryptography is the methods that allow information to be sent in a secure from in such a way that the only receiver able to retrieve this information.

Cryptographic algorithms play a dominant role in securing the files without getting attacked by the intruders.But in this current scenario there are numerous number of chances to break an encryption algorithm by performing cryptanalytic attacks. So the demand for a strong encryption algorithm becomes vital. become It is obvious that if a message is encrypted by using more than one encryption algorithm then it cannot be easily breaked by the eavesdroppers. Cloud users probably will not hold the strong belief that the cloud server is doing a good job in terms of confidentiality. A cryptographic solution says whenever the user is not perfectly happy with trusting the security of the Honesty of the technical staff, these users are motivated to encrypt their data with their own keys before uploading them to the server.

F. Benefits of a Cryptographic Storage Services

The core properties of a cryptographic storage service are that control (1) control of the data is maintained by the customer and 2) the security properties are derived from cryptography, as opposed to legal mechanisms, physical security or access control. Therefore, such a service provides several compelling advantages over other storage services based on public cloud infrastructures is outlined below,

1) Regulatory compliance: Most countries have laws in place that make organizations responsible for the protection of the data that is entrusted to them. This is particularly so for the case of personally identifiable information, medical records and financial records. And since organizations are often held responsible for the actions of their contractors, the use of a public cloud storage service can involve significant legal risks. In a cryptographic storage service, the data is encrypted on-premise by the data processor(s). This way, customers can be assured that the confidentiality of their data is preserved irrespective of the actions of the cloud storage provider. This greatly reduces any legal exposure for both the customer and the provider.

2) Geographic restrictions. Data that is stored in certain legal jurisdictions may be subject to regulations even if it was not collected there. Because it can be difficult to ascertain exactly where one's data is being stored once it is sent to the cloud (i.e., many service providers have data centers deployed throughout the world) some customers may be reluctant to use a public cloud for fear of increasing their legal exposure. In a cryptographic storage service data is only stored in encrypted form so any law that pertains to the stored data has little to no effect on the customer. This reduces legal exposure for the customer and allows the cloud storage provider to make optimal use of its storage infrastructure, thereby reducing costs.

3) Subpoenas. If an organization becomes the subject of an investigation, law enforcement agencies may request access to its data. If the data is stored in a public cloud, the request may be made to the cloud provider and the latter could even be prevented from notifying the customer. This can have severe consequences for customers. First, it preempts the customer from challenging the request. Second, it can lead to law enforcement having access to data from clients that are not under investigation, Such a scenario can occur due to the fact that service providers often store multiple customer's data on the same disks. In a cryptographic storage service, since data is stored in encrypted form and since the customer retains possession of all the keys, any request for the (unencrypted) data must be made directly to the customer.

4) Security breaches. Even if a cloud storage provider implements strong security practices there is always the possibility of a security breach. If this occurs the customer may be legally responsible. In a cryptographic storage service data is encrypted and data integrity can be verified at any time. Therefore, a security breach poses little to no risk for the customer.

5) Electronic discovery. Digital information plays an important role in legal proceedings and often organizations are required to preserve and produce records for litigation. Organizations with high levels of litigation may need to keep a copy of large amounts of data on-premise in order to assure its integrity. This can obviously negate the benefits of using a cloud storage service. Since, with a cryptographic storage service, a customer can verify the integrity of its data at any point in time (e.g., every hour) a

provider has every incentive to preserve the data's integrity.

6) Data retention and destruction. In many cases a customer may be responsible for the retention and destruction of the data it has collected. If this data is stored in the cloud, however, it can be difficult for a customer to ascertain the integrity of the data or to verify whether it was properly discarded. A cryptographic storage service alleviates these concerns since data integrity can be verified and since the information necessary to decrypt data (i.e., the master key) is kept on-premise. Secure data erasure can be effectively achieved by just erasing the master key.

We solve this problem by introducing a special type of public-key encryption which we call key -aggregate cryptosystem.In KAC users encrypts a message not only under a public-key,but also under an identifier of ciphertext called class. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes. More importantly, the extracted key have can be an aggregate key is as compact as a secret key for a single class, but aggregates the power of many such keys, i.e., the decryption power for any subset of ciphertext classes. our solution, Data owner can simply send dta user a single aggregate key via a secure e-mail. Bob can download the encrypted photos from Alice's Dropbox space and then use this aggregate key to decrypt these encrypted photos.The public system parameter has size linear in the number of ciphertext classes, but only a small part of it is needed each time and it can be fetched on demand from large (but nonconfidential) cloud storage. We propose several KAC schemes with different security levels and extensions in this paper. All constructions can be proven secure in the standard model. To the best of our knowledge, our aggregation mechanism.

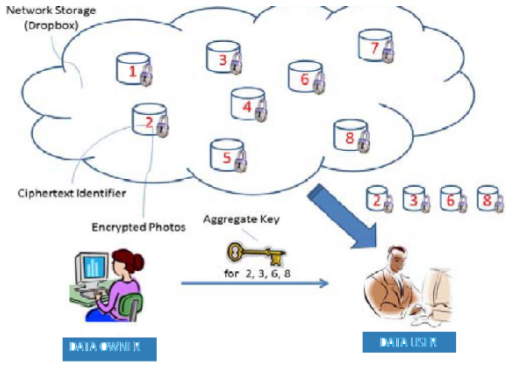


Figure 1. Alice shares files with identifiers 2, 3, 6, and 8 with Bob by sending him a single aggregate key.

II. RELATED WORK

A. Aggregation of Secret Keys

Introducing a special type of public-key encryption which we call key-aggregate cryptosystem (KAC).In KAC, users encrypt a message not only under a public key, but also under an identifier of ciphertext called class. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes. More importantly, the extract key can be an aggregate key which is compact as a secret key for a single class, but aggregates the power of many such keys ,i.e., the decryption power for any subset of ciphertext classes.

1) Framework: The data owner establishes the public system parameter via Setup and generates a public/master- secret3 key pair via KeyGen. Messages can be encrypted via Encrypt by anyone who also decides what ciphertext class is associated with the plaintext message to be encrypted. The data owner can use the master-secret to generate an aggregate decryption key for a set of ciphertext classes via Extract. The generated keys can be passed to delegates securely (via secure e-mails or secure devices) Finally, any user with an aggregate key can decrypt any ciphertext provided that the ciphertext's class is contained in the aggregate key via Decrypt.

a) Setup(1^λ,n): executed by the data owner to setup an account on an untrusted server. On input a security level parameter 1 λ and the number of ciphertext classes n (i.e., class index should be an integer bounded by 1 and n), it outputs the public system parameter

param, which is omitted from the input of the other algorithms for brevity.

b) KeyGen: executed by the data owner to randomly generate a public/master-secret key pair (pk; msk).

c) Encrypt (pk; i;m): executed by anyone who wants to encrypt data. On input a public-key pk, an index i denoting the ciphertext class, and a message m, it outputs a ciphertext C.

d) Extract (msk,S):executed by the data owner for delegating the decrypting power for a certain set of ciphertext classes to a delegatee. On input the master-secret key msk and a set S of indices corresponding to different classes, it outputs the aggregate key for set S denoted by KS.

e) Decrypt(Ks,S,i,C): executed by a delegate who received an aggregate key KS generated by Extract. On input KS, the set S, an index i denoting the ciphertext class the ciphertext C belongs to, and C, it outputs the decrypted result m if i ε S.

B. Steganography

Steganography is the practice of concealing a message, image, or file within another message, image, or file. Stegnography sometimes is used when encryption is not permitted. More commonly, steganography is used to supplement encryption. An encrypted file may still hide information using steganography, so even if the encrypted file.

Steganography and Cryptography are both pretended to protect information from unwanted third parties. Most of the experts would suggest using both to add multiple layers of security. The data formats used in the steganography are .bmp, .doc, .gif, .jpeg, .mp3, .txt and .wav. This formats are used because of the following advantage, as the redundant or noisy data can be easily removed from them and can be easily replaced with a hidden message.

The reason for the emerging of stenographic technique is due to the lack of strength in the cryptographic system. There are three types of steganography namely, pure steganography, secret key steganography, public key steganography. In this scheme we are using the pure steganography.

1) *Public Key Steganography:* Public key steganography used the ideas of the public key cryptography. Public Key Steganography is said to be steganography system that uses a public key and a private key to secure the communication between the data owners and the data users wanting to communicate secretly. The sender will use the public key during the encoding process and only the private key, which has a direct mathematical relationship with the public key, can decipher the secret message.

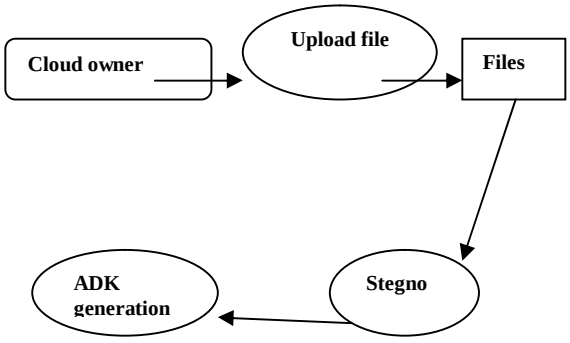


Figure 4 Key Generations and Implementation of Steganography

2) *Encoding Messages in Images:* Basically steganography is to code a secret message in to a digital images it is a wide trend in the digital word .As this hold its place in popularity because it can take advantage of the limited powerof the human visual system (HVS). Almost any plain text, cipher text, image and any other media that can be encoded into a bit stream can be hidden in a digital image.

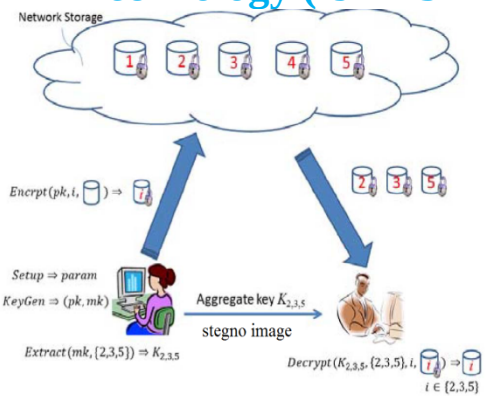


Figure 3. Alice sending the aggregate key along with the stegno image

III. ARCHITECTURE DESIGN

In wide-ranging perception say data sharing, we conscious of parties like data owner and data requestor. The data owner who are willing to share data with data requestor will upload an original data in server, where as an data requestor get an requested original data .Now the big deal is whether the uploaded data is still an original one. Obviously the answer is No!. So the solution to the above revealed problem is explained in this section

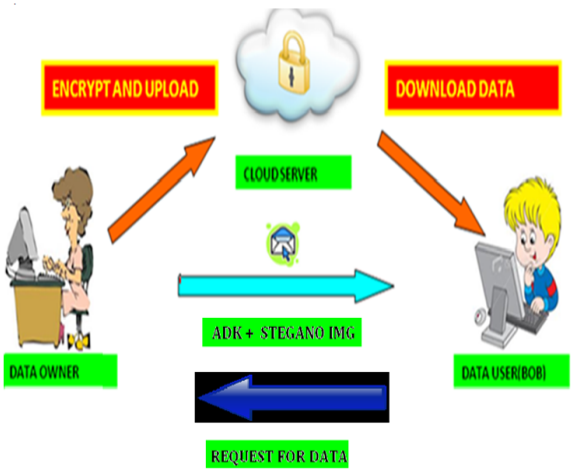


Figure 4.Architecure diagram

The Cloud Computing is vast developing technology, the challenging problem is how to effectively share encrypted data in cloud computing. In the EXISTING SYSTEM user's directly uploads the data in cloud drop box without encryption. So the hacker easily hacks the data and damage the originality of the content.so it end up with less security and missing of integrity.

In the PROPOSED SCHEME, Data owner randomly generates public/master-secret key pair after account is created in the server. Data owner encrypts the data, public key and data index & then uploaded in the Cloud Server. In additional steganography concept is used. Encrypted Outlet of original Data, Public Key and Index is made stegno into an Image. Data owner has to share the selected Image along with the ADK to download the Original Data. The advantage of this concept it provides high security. Preserving data integrity and confidentiality. Minimizing the number of ciphertext storage.

we are going to create an User application by which the User is allowed to access the data from the Server of the Cloud Service Provider. Here first the User wants to create an account and then only they are allowed to access the Network. Once the User creates an account, they are to login into their account and request the Job from the Cloud Service Provider. Based on the User's request, the Cloud Ser vice Provider will process the User requested Job and respond to them. All the User details will be stored in the Database of the Cloud Service Provider. In this Project, we will design the User Interface Frame to Communicate with the Cloud Server through Network Coding using the programming Languages like Java/ .Net. By sending the request to Cloud Server Provider,the User can access the requested data if they authenticated by the Cloud Service Provider

Cloud Service Provider will contain the large amount of data in their Data Storage. Also the Cloud Service provider will maintain the all the User information to authenticate the User when are login into their account. The User information will be stored in the Database of the Cloud Service Provider. Also the Cloud Server will redirect the User requested job to the Resource Assigning Module to process the User requested Job. The Request of all the Users will process by the Resource Assigning Module. To communicate with the Client and the with the other modules of the Cloud Network, the Cloud Server will establish connection between them. For this Purpose we are going to create a User Interface Frame. Also the Cloud Service Provider will send the User Job request to the Resource Assign Module in Fist in First out (FIFO) manner.
Cloud servers are constructed with the files and the index information are maintained in the main cloud server. The data are added in each cloud servers, and network construction is made with the entire data index present in each cloud server. Query is given to the main cloud server, so that the main cloud server will verify the index information present in it & divert the query to the corresponding cloud servers.

Data owner uploading the file it is encrypted with AES algorithm and provided with public key and private key and also master key this private and public key is used to see the data in the cloud and the master is used to create aggregate key. when a cloud owner upload file into the cloud if the cloud user want to see the file then the cloud user has to make a request to the cloud owner and then cloud has to generate the aggregate key using master key, and then send the key to the cloud user through the email. In this project is user will encrypt the file ,public key and index into an image called stenography to cloud user will be giving their username, password, user public key and send the request to the owner. if owner is interested to share then it will forward ADK, private key and key to the user. User is authenticated after verification so that the data is shared securely.

A. Algorithm Details

Encryption keys comes with two flavors namely, symmetric keys encryption or asymmetric (public) key encryption. The public key encryption gives more flexibility for our application. The decryption key and the encryption key is different in the public key encryption. For example, in an enterprise every employee can make use of an uploading option and upload encrypted data on the cloud storage server without the prior knowledge of enterprise master secret key. The algorithm details is given below:

1) AES (Asymmetric Encryption Standard): This algorithm is flexible in supporting any combination of data and key size of 128, 192, and 256 bits. However, AES merely allows a 128 bit data length that can be divided into four basic operation blocks. These blocks operate on array of bytes and organized as a 4×4 matrix that is called the state. For full encryption, the data is passed through Nr rounds (Nr = 10, 12, 14). These rounds are governed by the following transformations,

a) Byte Substitution: This is a non linear byte Substitution, using substation table (s-box), which is constructed by multiplicative inverse and affine transformation.
b) Shifting the rows: This is a simple byte transposition, the bytes in the last three rows of the state are cyclically shifted; the offset of the left shift varies from one to three bytes.
c) Mixing of columns: Is equivalent to a matrix multiplication of columns of the states. Each column vector is multiplied by a fixed matrix. It should be noted that the bytes are treated as polynomials rather than numbers.
d) Adding round key: Is a simple XOR between the working state and the round key. This transformation is its own inverse.
The following describes these four stages briefly,

a) Substitution Box(s-box): The Rijndael S-box is a matrix (square array of numbers) used in the Rijndael cipher, which the Advanced encryption standard (AES) cryptographic algorithm was based on.The S-box (substitution box) serves as a lookup table.The S-box is generated by determining the multiplicative inverse for a given number in $GF(2^8)$ = $GF(2)[x]/(x^8 + x^4 + x^3 + x + 1)$, Rijndael's finite field (zero, which has no inverse, is set to zero). The multiplicative inverse is then transformed using the following affine transformation:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Figure 4. Affine Transformation

where $[x_0, ..., x_7]$ is the multiplicative inverse as a vector. This affine transformation is the sum of multiple rotations of the byte as a vector, where addition is the XOR operation.

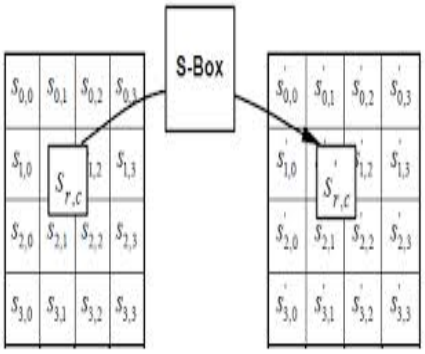


Figure 5.Substitution Box

The matrix multiplication can be calculated by the following algorithm:

1. Store the multiplicative inverse of the input number in two 8-bit unsigned temporary variables: *s* and *x*.
2. Rotate the value *s* one bit to the left; if the value of *s* had a high bit (eighth bit from the right) of one, make the low bit of *s* one; otherwise the low bit of *s* is zero.
3. Exclusive OR the value of *x* with the value of *s*, storing the value in *x*
4. For three more iterations, repeat steps two and three; steps two and three are done a total of four times.
5. The value of *x* will now have the result of the multiplication.

b) Shifting of Rows: The shift rows transformation operates on the rows of the state. The bytes present in the last three rows of the state are cyclically shifted over different number of bytes or offsets. The first row of the array is not changed. Each byte of the second row is shifted one to the left in a cyclic order. The second and third rows are shifted two and three bytes respectively. The shifting pattern

is the same for 128 bits and 192 bits of block size. In the case of 256 bit block, the first row is left unchanged and the second, third and fourth row are shifter 1 byte, 3 bytes and 4 bytes respectively.

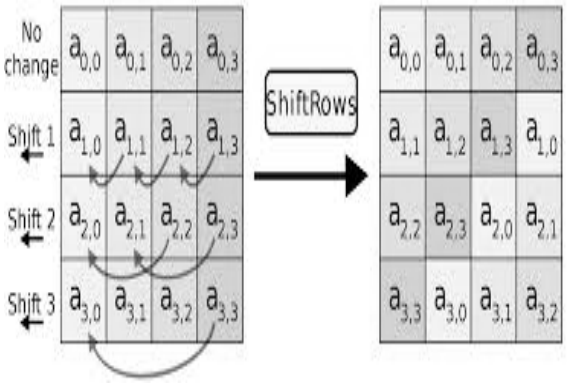


Figure 6.Shifting Of Rows

c) Mixing of Columns: The four bytes of each column of the state are combined using a invertible linear transformation in the mixcolumns function. The mixcolumns transformation takes four bytes as input and outputs four bytes. All the input bytes affect the output bytes. Mixcolumns and shiftrows provide diffusion in the cipher. Each column of the mixrows is treated as a polynomial over GF (28) and is then multiplied modulo x4+1 with a fixed polynomial c(x) = 0x03

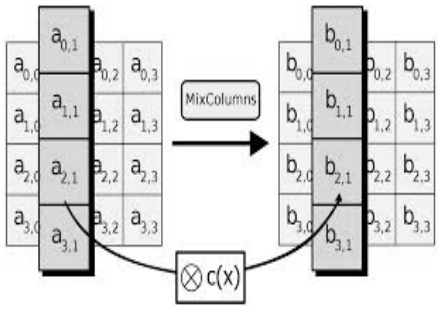


Figure 7.Mixing of columns

d) *SubByte Transformation:* The subBytes transformation can be described as a non-linear byte substitution that works or operates independently on each byte of the state using a substitution table also known as the S-box, derived from the Rijndael algorithm. The multiplicative inverse over GF(28) is used to derive the S-box. The S-box is constructed by combining the inverse function with an invertible affine transformation (a transformation that prevents straight lines) to avoid attacks. Any fixed points and opposite fixed points are avoided while choosing the S-box.

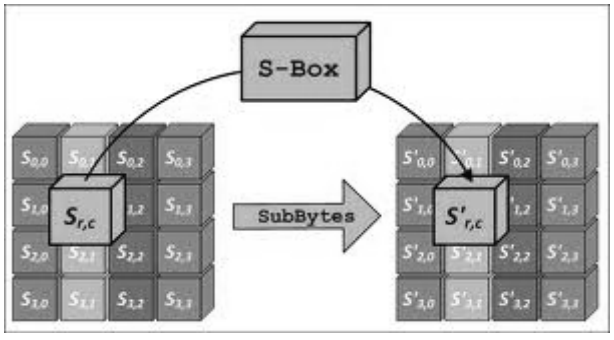


Figure 8.subByte Transformation

Cascade Encryption: The cascade encryption is otherwise said to be multiple encryption .Multiple encryption is such that encrypting the data more then twice,as we aware that triple time encryption named 3DES,but we are not using an DES instead AES is used,the reason behind this is explained below as an advantage of AES over 3DES.

e) Advantages of AES over 3DES

i. AES is more secure (it is less susceptible to cryptanalysis than 3DES).
ii. AES supports larger key sizes than 3DES's 112 or 168 bytes.
iii. AES is faster in both hardware and software.
iv. AES's 128-bit block size makes it less open to attacks via the birthday problem than 3DES with its 64-bit block size.

IV. CONCLUSION

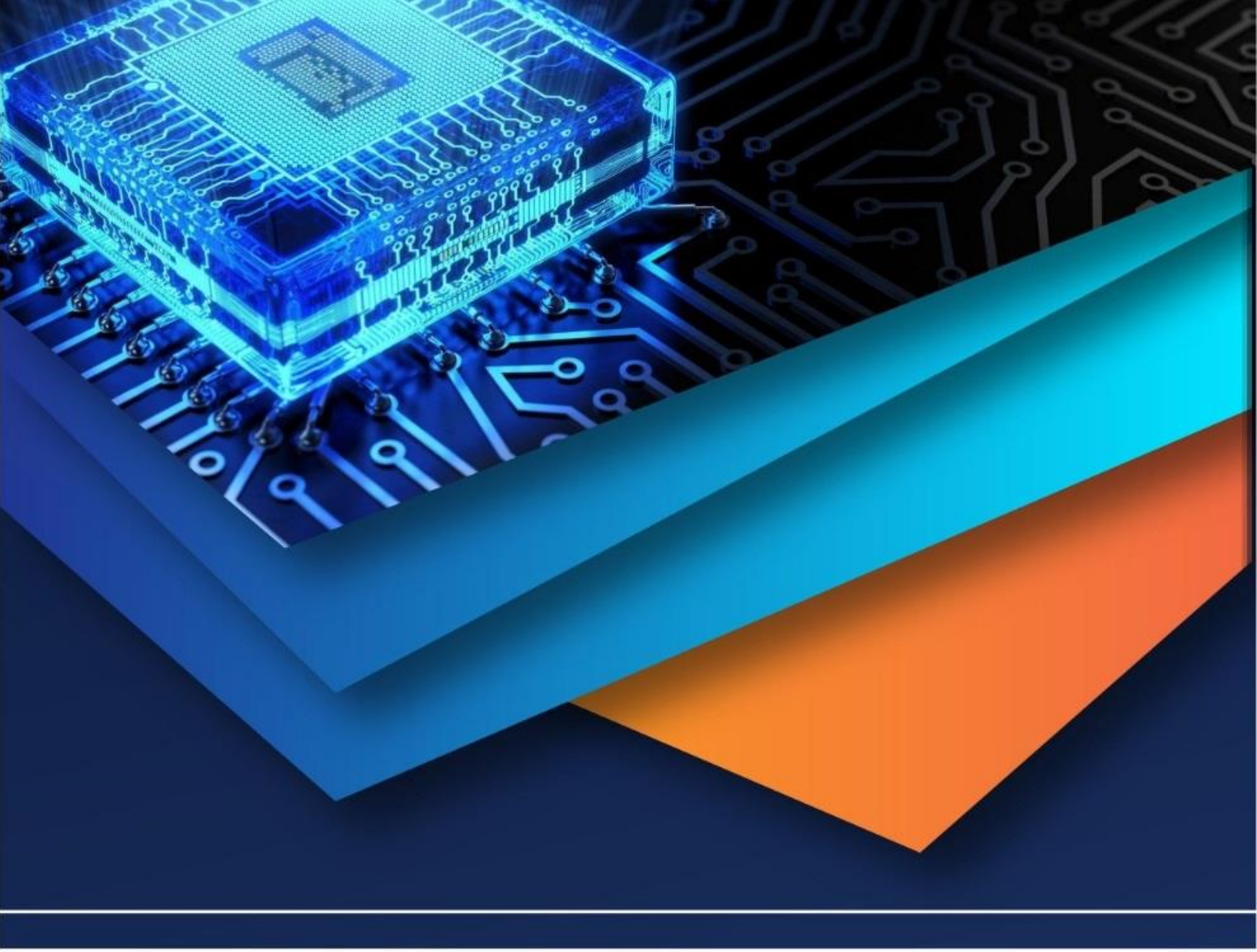
To protect user's data privacy is an ultimate question of cloud storage. Mathematical tools like cryptographic schemes are getting more and more versatile. And there is a drawback called multiple keys for single application. In this paper, we considered the fore mentioned drawbacks and decided to compress the key requirements. So a new approach called steganography technique is used along with the KAC approach to provide multiple layer of security for data owner who upload the data and data user who download the data. More preserving data integrity and confidentiality.so that it is an challenging problem for a breaker/interrupter to inject or to remove data from an original content. And it's a toughest work for an other parties to predict the original data from an cipher format of the original one, because of the introduction of the multiple time encryption (Cascade Encryption).

V. ACKNOWLEDGMENT

I would like to thank my guide Mr.V.S Bharathi Dasan for assisting me in this paper work.

REFERENCES

[1] D. Boneh, C. Gentry, B. Lynn, and Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '03), pp. 416-432, 2003.
[2] C.-K. Chu and W.-G. Tzeng, "Identity-Based Proxy Re-encryption without Random Oracles," Proc. Information Security Conf. (ISC '07), vol. 4779, pp. 189-202, 2007.
[3] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Trans. Information and System Security, vol. 9, no. 1, pp. 1-30, 2006.
[4] S.S.M. Chow, J. Weng, Y. Yang, and R.H. Deng, "Efficient Unidirectional Proxy Re-Encryption," Proc. Progress in Cryptology(AFRICACRYPT '10), vol. 6055, pp. 316-332, 2010.
[5] S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu, "SPICE – Simple Privacy-Preserving Identity-Management for Cloud Environment,"Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS), vol. 7341, pp. 526-543, 2012.
[6]"Integrating OpenID with proxy re-encryption to enhance privacy in cloud-based identity services" David Nu˜nez, Isaac Agudo, and Javier Lopez Network, Information and Computer Security Laboratory Universidad de M´alaga M´alaga, Spain
[7] L. Hardesty, Secure Computers Aren't so secure .MITpress,http://www.physorg.com/news176107396.html, 2009
[8] C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy- Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.
[9] B. Wang, S.S.M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS), 2013.
[10] R. Canetti and S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re-Encryption," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 185-194, 2007.
[11] S.S.M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R.H. Deng, "Dynamic Secure Cloud Storage with Provenance," Cryptography and Security, pp. 442-464, Springer, 2012.
[12] C.-K. Chu, J. Weng, S.S.M. Chow, J. Zhou, and R.H. Deng, "Conditional Proxy Broadcast Re-Encryption," Proc. 14th Australasian Conf. Information Security and Privacy (ACISP '09), vol. 5594, pp. 327-342, 2009.
[13] D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," Proc. Advances in Cryptology Conf. (CRYPTO '05), vol. 3621, pp. 258-275, 2005.
[14] S.S.M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters, "Practical Leakage-Resilien Identity-Based Encryption from Simple Assumptions,"Proc. ACM Conf. Computer and Comm. Security, pp. 152-161, 2010.
[15] M. Chase and S.S.M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," Proc. ACM Conf. Computer and Comm. Security, pp. 121-130. 2009,
[16] T.H. Yuen, S.S.M. Chow, Y. Zhang, and S.M. Yiu, "Identity-Based Encryption Resilient to Continual Auxiliary Leakage," Proc. Advances in Cryptology Conf. (EUROCRYPT '12), vol. 7237, pp. 117-134, 2012.
[17] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical Identity Based Encryption with Constant Size Ciphertext," Proc. Advances in Cryptology Conf. (EUROCRYPT '05), vol. 3494, pp. 440-456, 2005



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY