INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 3)
Available online at: www.ijariit.com

Ciphertext attribute-based encryption algorithm to confiscate de-duplication in hybrid cloud storage

Shobhanjaly P Nair
anjaly.cse@gmail.com
Anand Institute of Higher Technology, Kazhipathur, Tamil Nadu

Dr. A. Kathirvel
ayyakathir@gmail.com
Misrimal Navajee Munoth Jain Engineering College, Chennai, Tamil Nadu

ABSTRACT

The attribute-based encryption algorithm is widely used in cloud computing to make encrypted data outsourced by data providers to be shared or accessed only by users possessing specific credentials (or attributes). However, the standard ABE system could not ensure secure de-duplication, which leads to less storage space and network bandwidth due to duplicate copies of identical data. An attribute-based storage system is introduced which helps to overcome de-duplication in hybrid cloud storage, where a private cloud is responsible for duplicate detection and a public cloud handles the storage. The core advantages of this system are, primarily it can be used to confidentially share data with users by scrutinizing their specific credentials rather than sharing decryption keys. Also, the system attains the standard notion of semantic security for data privacy while existing systems can only achieve this through weaker security notion. Also, the equality checking algorithm will help in detecting de-duplication and confiscate the duplicate copy in private cloud storage.

Keywords: *Ciphertext- attribute-based encryption algorithm, Equality checking algorithm, Huffman technique*

1. INTRODUCTION

Cloud computing greatly facilitates data providers who want to outsource their data to the cloud without disclosing their sensitive data to external parties and would like users with certain credentials to be able to access the data. This requires data to be stored in encrypted forms with access control policies such that no one except users with attributes (or credentials) of specific forms can decrypt the encrypted data. An encryption technique that meets this requirement is called attribute-based encryption (ABE). Where a user's private key is associated with an attribute set, a message is encrypted under an access policy (or access structure) over a set of attributes, and a user can decrypt a cipher text with his/her private key if his/her set of attributes satisfies the access policy associated with this cipher text. But, the standard ABE system fall short in achieving secure de-duplication [1], which is a technique to save storage space and network bandwidth by eliminating redundant copies of the encrypted data stored in the cloud.

Also, to the best of our knowledge, existing constructions for secure de-duplication are not built on attribute-based encryption. Nevertheless, since ABE and secure de-duplication have been widely applied in cloud computing, it would be desirable to design a cloud storage system possessing properties. Consider the scenario, where the design of an attribute-based storage system supporting secure de-duplication of encrypted data in the cloud, in which the cloud will not store a file more than once even though it may receive multiple copies of the same file encrypted under different access policies. A data provider intends to upload a file in the cloud, and share it with users having certain credentials. In order to data provider encrypts file under an access policy over a set of attributes, and uploads the corresponding cipher text to the cloud, such that only users whose sets of attributes satisfying the access policy can decrypt the cipher text. Later, another data provider uploads a cipher text for the same underlying file but ascribed to a different access policy. Since the file is uploaded in an encrypted form, the cloud is not able to discern that the plaintext corresponding to data provider cipher text is the same as that corresponding to users, and will store twice. Obviously, such duplicated storage wastes storage space and communication bandwidth.

The concept of de-duplication is to eliminate the redundant data. We provide definitions of privacy and integrity peculiar to this domain. Now having created a clear, strong target for designs, we make contributions that may broadly be divided into two parts: practical and theoretical. Analyze existing schemes and new variants, breaking some and justifying others with proofs in the random-oracle-model (ROM). MLE (Message Locked Encryption)[8,9]appears as a primitive that combines practical impact with theoretical depth and challenges, making it well worthy of further study and a place in the cryptographic pantheon. Many enterprises and other

organizations need to store and compute on a large amount of data. Cloud computing rents resources based on requirement. Cloud providers offer both, highly available storage and massively parallel computing resources with High Performance Computing (HPC) at low costs, as they can share resources among multiple clients.

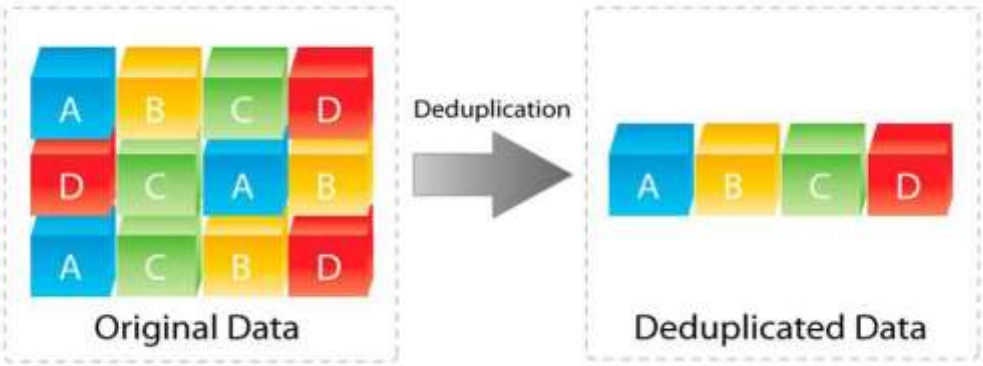

Fig. 1: Simple De-duplication Concepts

Several attribute-based constructions have been presented; a common classification property is whether a system is a "small universe" or "large universe" constructions. In "small universe" constructions the size of the attribute space is polynomial bounded in the security parameter and the attributes were fixed at setup. Moreover, the size of the public parameters grew linearly with the number of attributes. In "large universe" constructions, on the other hand, the size of the attribute universe can be exponentially large, which is a desirable feature. A technique which has been proposed to meet these two conflicting requirements is convergent encryption whereby the encryption key is usually the result of the hash of the data segment. Although convergent encryption seems to be a good candidate to achieve confidentiality and de-duplication at the same time, it unfortunately suffers from various well-known weaknesses including dictionary attacks: an attacker, guess or predict files that can easily derive the potential encryption key and validate whether the file is previously stored at the cloud storage provider or not. CLF is an emerging field of data security use to analyze data inside cloud log files for the investigation of malicious behavior [4]. However, cloud log files are only accessible to a Cloud Service Provider (CSP) through cloud resource ownership. For instance, in cloud computing Software-as-a-Services (SaaS), a user is provided with developed software to run its applications. Each application generates log files during its execution on the cloud that are inaccessible to the users. Cloud services and applications may require all standard security functions including data confidentiality, integrity, privacy, robustness and access control. Hence securing the cloud and its data is a challenging task. There are several cryptographic methods to secure the data stored in cloud storage systems.

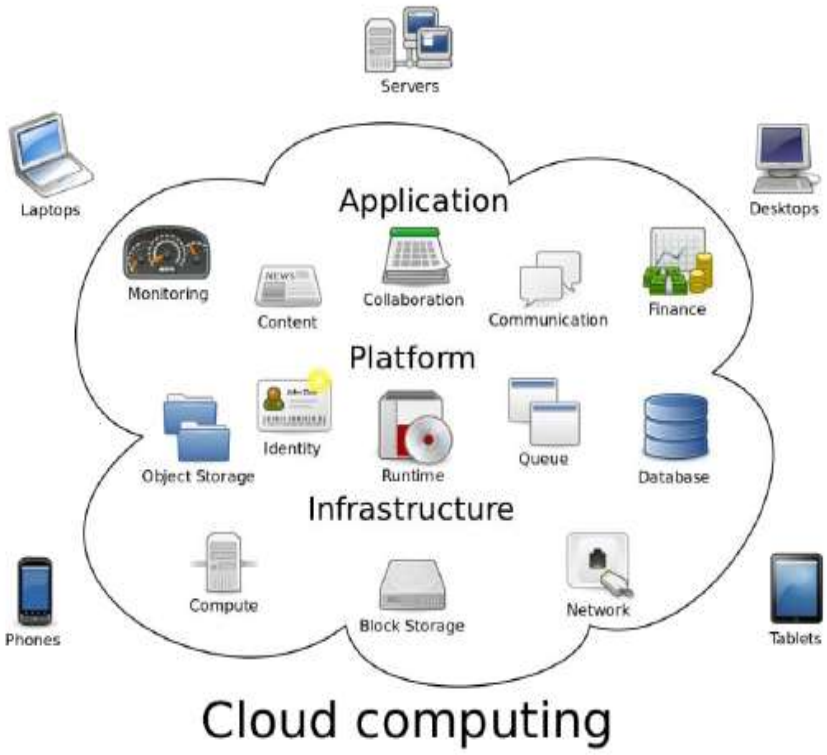

Fig. 2: Simple Cloud Concepts

Proxy re-encryption is a relatively new data encryption technique devised primarily for distributed data and files security [2]. The target of proxy re-encryption is allowing the re-encryption of one cipher text to another cipher text without relying or trusting the third party that perform the transfer. In situations where one user wishes for another user decrypt a message using its own or a new secret key instead of the first user's secret key, one technique involves the assistance of a proxy.

2. LITERATURE SURVEY

Hui Cui et al (2017) Attribute Based Encryption (ABE) system can be used as an attribute that need not be enumerated at system setup. The first construction establishes a large universe Cipher text-Policy ABE scheme on prime order bilinear groups, while the second achieves a significant efficiency improvement over the large universe Key-Policy ABE system. Both schemes are selectively secure in the standard model under two "q-type" assumptions similar to ones used in prior works. The process brings back "program and cancel" techniques to this scheme and aims in providing practical large universe ABE implementations. The standard model defines the presented KP-ABE construction, secure in the standard model. The system was proved selectively secure under static assumptions. The construction was an "unbounded" scheme, in the sense that the public parameters do not impose additional limitations on the functionality of the systems. The scheme is indeed large universe, since the size of the attribute universe is exponentially large in the security parameter.

Raghi Roy and Paul P. Mathai, (2017) together explained Proxy re-encryption techniques with respect to secure cloud data and its application. To keep sensitive user data confidential against un-trusted servers, cryptographic methods are used to provide security and access control in clouds. As the data is shared over the network, it is needed to be encrypted. There are many encryption schemes that provide security and access control over the network. Proxy re-encryption enables the semi-trusted proxy server to re-encrypt the cipher text encrypted under user1 public key to another cipher text encrypted under user2 public key. The re-encryption is done without the server being able to decrypt the cipher text. Cloud services and applications should follow the standard security measures including data confidentiality, integrity, privacy, robustness and access control. In this paper the proxy re-encryption(PRE) schemes, Conditional PRE, Identity based PRE and broadcast PRE, Type based PRE, Key private PRE, Attribute based PRE, Threshold PRE and its role in securing the cloud data are explained. Cloud computing is emerging as an inevitable option for internet based applications and services. Cloud computing is a distributed computing architecture where the computing resources such as hardware, software, processing power are delivered as a service over a network infrastructure. The cloud computing model allows the users to access information and other resources from anywhere that a network connection is available. In cloud computing all data are stored on distributed servers at remote location. The remote locations are data centre. The client can purchase or rent, such as handling time, network bandwidth, disk storage and memory. Data owners can remotely store their data in the cloud and no longer posses the data locally. Cloud computing migrates the application software and database to the large datacenter, where the data management and services may not fully trustworthy.

K. R. Choo et al. (2016) has explained about cloud computing, a convenient way of accessing services, resources and applications over the internet, shifts the focus of industries and organizations away from the deployment and day-to-day running of their IT facilities by providing an on-demand, self-service, and pay-as-you- go business model. It is, therefore, unsurprising that cloud computing has continued to increase its popularity in recent times. While cloud computing provides various benefits to users, there are underlying security and privacy risks. For example, multi-tenancy, resource pooling and share ability features can be exploited by cybercriminals and anyone with a malicious intent to the detriment of both cloud users and cloud service providers. That the cloud computing has emerged as a salient area of inquiry for security researchers with the cloud computing based process. Where it maintain secure with appropriate data in the cloud.. For example, when user data like (e.g documents, videos and photos) are uploaded or stored in a cloud computing service then the (documents, videos and photos) are stored securely in the cloud service.

K. R. Choo et al. (2016) discussed another new technique, Cloud log forensics (CLF) that helps in mitigating the investigation process by identifying the malicious behavior of attackers through profound cloud log analysis. However, the accessibility attributes of cloud logs obstruct accomplishment of the goal to investigate cloud logs for various susceptibilities, because cloud log files are only accessible to a Cloud Service Provider (CSP) through cloud resource ownership. Accessibility involves the issues of cloud log access, selection of proper cloud log file, cloud log data integrity, and trustworthiness of cloud logs. Therefore, forensic investigators of cloud log files are dependent on cloud service providers (CSPs) to get access of different cloud logs. Accessing cloud logs from outside the cloud without depending on the CSP is a challenging research area, as CSP restrict access to third-party investigators for cloud log files due to user data privacy and organizational Standard Operating Procedures (SOPs) therefore data integrity is preserved in CLF. This paper reviews the state of the art of CLF and highlights different challenges and issues involved in investigating cloud log data. The performance analysis of the cloud log files to produce potential evidence to help the investigator to track the attacker by re-generating the malicious activities again is continuously monitored.

Dayananda RB and Prof. Dr. G.Manoj Someswar (2015), has discussed about time based proxy Re-encryption scheme which describes the system model and security model and provide the design goals and related assumptions such as time based re-encryption process in secure data sharing in cloud. A cloud computing environment consisting of a cloud service provider (CSP), a data owner and many users maintains cloud infrastructures, which pool the bandwidth, storage space, and CPU power of many cloud servers to provide 24/7 services . The data owner outsources a set of data to the cloud. Each piece of data is encrypted before outsourcing. The data owner is responsible for determining the access structure for each data, and distributing user attribute secret keys (UAKs) corresponding to user.

Mart´ın Abadi and Dan Boneh's (2013), the first construction deviates from the approach of Bellare et al. by avoiding the use of cipher text components derived deterministically from the messages. We design a fully randomized scheme that supports an equality-testing algorithm defined on the cipher text. Their second construction has a deterministic cipher text component that enables more efficient equality testing. Security for lock-dependent messages still holds under computational assumptions on the message distributions produced by the attacker. The first approach is to avoid using tags that are derived deterministically from the messages. To end this, a fully randomized scheme that supports an equality-testing algorithm is defined on the cipher text. To illustrate that this method enables to satisfy a strong definition of security for an extension of the Message Locked Encryption notion, is done by allowing the adversary to specify the distribution of the plaintexts adaptively with no further restrictions on the

distribution than its min-entropy. The second approach continues using deterministic tags. Security for lock-dependent messages is guaranteed by limiting the computational power of the adversarial message distributions. The construction can be based on any semantically secure encryption scheme. Its overhead, defined as the increase in the length of the cipher text, is additive and depends only on the security parameter.

M. Bellare et al. (2013) has explained the Message-Locked Encryption (MLE) method to secure de-duplication. A MLE key is used to perform encryption and decryption of message which helps to achieve the secure de-duplication (space-efficient secure outsourced storage). This method is been used by numerous cloud-storage providers since it provides definitions for privacy and integrity that it is also called as tag consistency. Based on the foundation, MLE makes both practical and theoretical contributions. On the practical side, it offers ROM (random- oracle-model) security analyses a usual kind of MLE schemes that includes deployed schemes. On the theoretical side, addressing the challenging issue in finding a standard-model MLE scheme and establishing connections with deterministic public-key encryption. The correlated-input-secure hash functions will provide schemes for exhibiting different trade-off between assumptions made and the message distributions through the hash function with security.

S. Bugiel et al. (2011) introduced a Cloud Computing technique which enables a cost effective technology to outsource storage and massively parallel computations. The existing system approaches for probably secure outsourcing of data and arbitrary computations are either based on tamper-proof hardware or fully homomorphism encryption. Protocols that accumulate slow secure computations over time and provide the possibility to query them in parallel, on demand by leveraging the benefits of cloud computing. In this approach, the user communicates with a resource-constrained Trusted Cloud (either a private cloud or built from multiple secure hardware modules) which encrypts algorithms and data to be stored and later on queried in the powerful but un-trusted Commodity Cloud.

Mihir Bellare and Adriana Palacioy (2002) discussed Guillou-Quisquater (GQ) and Schnorr identification schemes are amongst the most efficient and best-known Fiat-Shamir follow-on, but the question of whether they can be proven secure against impersonation under active attack has remained open. This paper provides proof for GQ, based on the assumed security of RSA less than one more inversion, an extension of the usual one-way assumption that was introduced. It also provides a proof for Schnorr scheme, based on a corresponding discrete-log related assumption. These are the security proofs for these schemes under assumptions related to the underlying one-way functions. Both results extend to establish security against impersonation under concurrent attack, as well as its extension to even stronger attacks. The process of security attacks with the proof will be maintained with the identification schemes of GQ and Schnorr.
J. R. Douceur et al. (2002), The Far site distributed file system provides availability by replicating each file onto multiple desktop computers. Since this replication consumes significant storage space, it is important to reclaim used space where possible. Measurement of over 500 desktop file systems shows that nearly half of all consumed space is occupied by duplicate files. We present a mechanism to reclaim space from this incidental duplication to make it available for controlled file replication. The proposed mechanism includes 1) convergent encryption, which enables duplicate files to coalesced into the space of a single file, even if the files are encrypted with different users' keys, and 2) SALAD, a Self- Arranging, Lossy, Associative Database for aggregating file content and location information in a decentralized, scalable, fault-tolerant manner. Large- scale simulation experiments show that the duplicate-file system is scalable, highly effective, and fault-tolerant. The problems of identifying and coalescing identical files in the Far site distributed file system, for the purpose of reclaiming storage space consumed by incidentally redundant content. Far site is a secure, scalable, server less file system that logically functions as a centralized file server but that is physically distributed among a networked collection of desktop workstations. Process must tolerate a high rate of system failure, operate without central coordination, and function in tandem with cryptographic security. Since the disk space of desktop computers is mostly unused and becoming less used over time, reclaiming disk space might not seem to be an important issue.

3. CLOUD STORAGE

There are three forms of Cloud computing available: public clouds, private clouds, and hybrids clouds. Based on the nature of data used all the three Cloud Computing storages are evaluated in terms of their security level and management offered. The three forms are discussed,

3.1 Public Clouds

A public cloud offers an easily accessible, inexpensive service provider which is available for general users. Basically a internet can be termed as public cloud. Examples of public clouds include Amazon Elastic Compute Cloud (EC2), IBM's Blue Cloud, Sun Cloud, Google App Engine and Windows Azure Services Platform. There are some drawbacks, as the public cloud cannot fit for organization storage purpose.

3.2 Private Clouds

A Private clouds are designed specifically for organization storages. Where the cloud is owned by a specific organization and provides flexibility, provisioning, scalability, monitoring and automation of data stored. This kind of cloud storage offers better security, but it's expensive and not recommended for medium sized company.

3.3 Hybrid Clouds

A Hybrid approach, is mix of both public and private cloud. It can maintain control of an internally managed private cloud while relying on the public cloud as needed. It provides security for all the data stored in the cloud as it provides access to user only based on the authenticity of the user. During catastrophic times it is difficult and expensive to maintain organization data; therefore in hybrid cloud all data or portion of data can be transferred to public cloud. It will help in easy data recovery and helps in avoiding any loss of data.

4. RELATED WORK

The key aspect of Attribute-based storage system which employs cipher text-policy attribute-based encryption (CPABE) is to supports secure de-duplication. In the proposed system, Equality Checking Algorithm is employed to verify the files/data whether it's replicated. Information about duplicate files/data is intimated to data owner. The Symmetric Algorithm is used to encrypt the files/data for security reasons. Huffman techniques used to compress the size of file, to reduce the storage space optimized in cloud. To check and eradicate de-duplication of files/data, the system uses hybrid cloud architecture. The private keys for authentic users are not issued to them directly; it is kept and managed by the private cloud server. The privileged user's identity is verified using the credentials (attributes) and then, on basis of their requested file/data key is been issued, since each file/data is encrypted and key generated is stored in private cloud. This prevents unidentified users from accessing the files and provides at most security. To get a file token, the user needs to send a request to the private cloud server. The authorized duplicate check for this file is performed in the public cloud before uploading the file. Based on the results of duplicate check, the user either uploads this file or runs POW.

The advantages of the proposed system are:

- De-duplication done on the target side(in cloud) and if any duplicate copies found deletes the duplicate files thereby provides storage system with secure de-duplication.
- The file uploaded is encrypted and key generated is stored in private cloud. Without an authenticate key, even privileged user cannot access any file. After verifying the user based on their credentials and file requested a key is issued.
- This helps in reducing the storage space, increase network bandwidth and provides fastest recoveries and high security.

5. ALGORITHM DESIGN

5.1 Cipher text abe algorithm

In the cipher text-policy attribute-based encryption scheme, each user's private key (decryption key) is tied to a set of attributes representing that user's permissions. When a cipher text is encrypted, a set of attributes is designated for the encryption, and only users tied to the relevant attributes are able to decrypt the cipher text.
The example presented on the website presents a cipher text encrypted such that only employees with the attributes "Human Resources" UNION "Executive" are able to decrypt it. HR employees have the "Human Resources" attribute tied to their private keys, and Executive employees have the "Executive" attribute tied to their private keys. Both groups, therefore, are able to decrypt the encrypted message. Unlike other Role-Based Access Control (RBAC) systems, CPABE does not require a trusted authority, or any form of storage. The encryption itself serves as the RBAC mechanism.

- SETUP (): This algorithm is run by a trusted authority. It takes as input a security parameter, and outputs public parameters *PK* and a master secret key *MK*.
- KEY GENERATION (): This algorithm is also run by the trusted authority. It takes as input the public parameters *PK*, the master secret key *MK* and a set of user's attributes. The output of this step is the secret key for a user with the attribute set. Here is composed of two parts, i.e. and, where can be used by *proxy B* to assist in decryption, while is used directly by the user to recover a plain message from the partially decrypted cipher-textˆconstructed by *proxy B*.
- ENCRYPTION (): The encryption algorithm takes as input the public parameters *PK*, a message *M*, and an access tree over a universe of attributes. It partial cipher text is produced includes the access tree (structure), but no cryptographic access policy associated.
- POLICY CREATION (): This algorithm is run by *proxy A* in order to create the final cipher text *CT*. It takes as input the partial cipher textˇand the proxy encryption secret key and creates the cryptographic access policy related to the access tree.
- POLICY VERIFICATION():This algorithm is run by *proxy B* that takes as input the proxy decryption key related to and the cipher text . The output of this algorithm is a partially decrypted cipher textˆ(called ElGamal style cipher text) if satisfies access tree.
- DECRYPTION (): The user runs the decryption algorithm. The decryption algorithm takes as input the partially decrypted cipher textˆand a user's secret key (called ElGamal style private key). The output of this stage is the decrypted message if satisfies, otherwise the output is an error.

5.2 EQUALITY CHECKING ALGORITHM

The fundamental form of this encryption is taking uploaded file and calculating a hash from it. Then using this hash as the key, encrypt the rest of the file. Finally using the password, hash key is encrypted and stored. The user can get the password of the file only if the user owns the original file. When two or more users have the same file, they have to calculate the same hash, and the same encrypted file. Since the encrypted version is the same, only one copy is required. To decrypt the file, user has to first decrypt the hash using their password, then the decrypt the file using the hash.
STEP 1: Client! Server: The client asks for the de-duplication of new data m_, and the server returns the tag of the current node (gri; gr0_h(mi)). (Initially, the current node is the root of the tree and its tag is (gr0; gr0_h(m0)).
STEP 2: Client: The client computes gri_h(m_) and verifies gri_h(m_)?=gri_h(mi).
STEP 3: Client! Server: If gri_h(m_) = gri_h(mi), the client sends "duplication find" to the server. Otherwise, it computes b = B(gri_h(m_)) 2 f0; 1g and sends b to the server.
STEP 4: Server: The server moves the current pointer of the tree according to b. If b = 0, the server moves the pointer to its left child. Otherwise, it moves the current pointer to its right child. Then, return to step 1. The algorithm stops, when the server receives "duplication find" or it needs to move the pointer to an empty node.

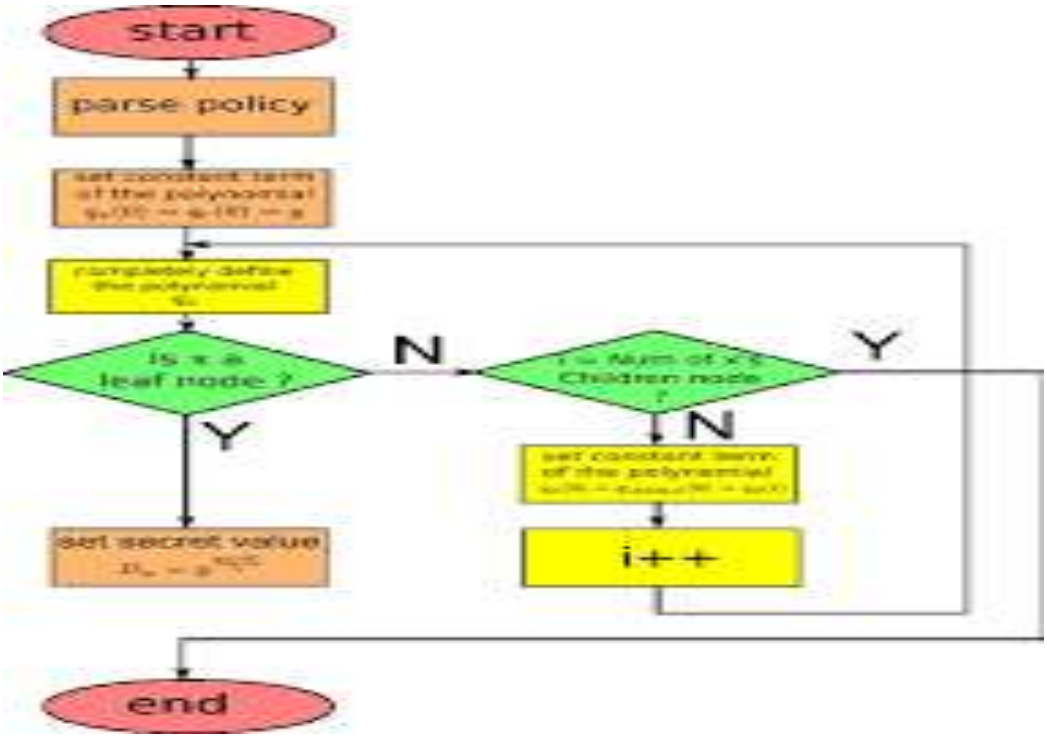

Fig. 3: Cipher Text ABE Algorithm Flow Chart

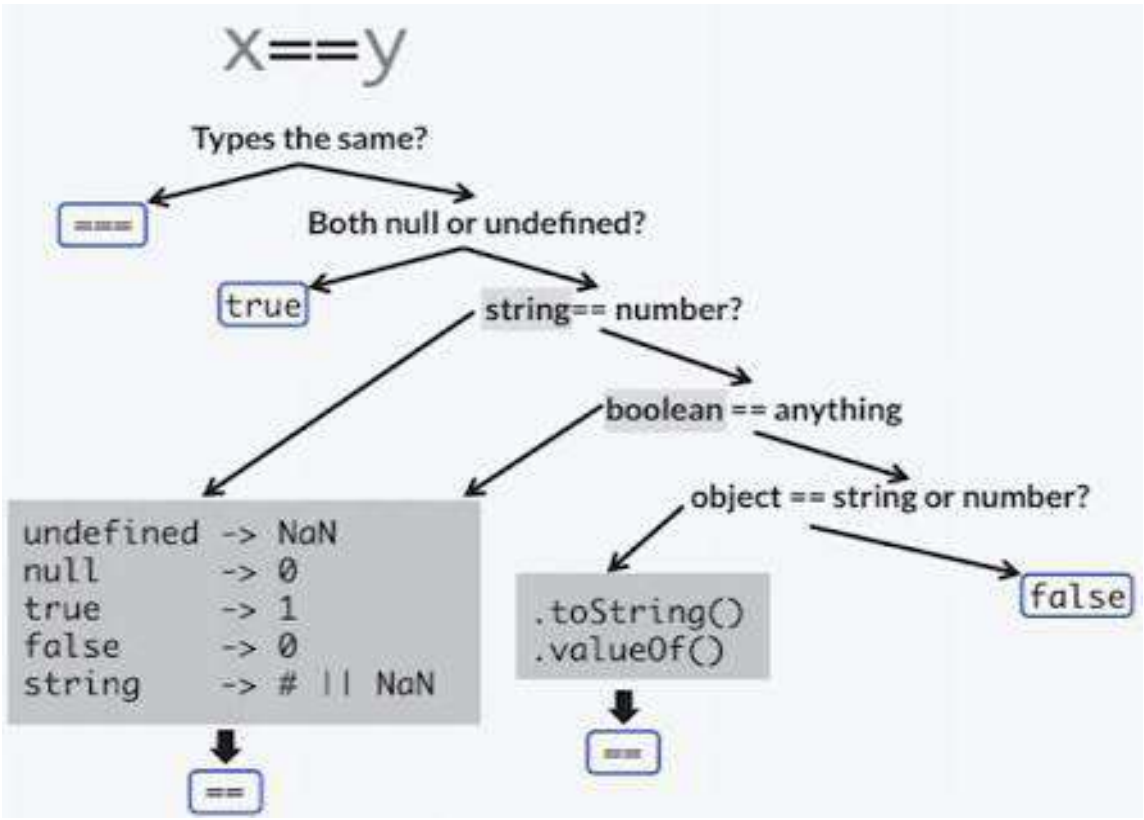

Fig 4: Equality Checking Algorithm Flow Chart

6. SYSTEM ARCHITECTURE

The Figure describes overall system architecture where a user can login and upload files. Also uploaded files are encrypted and a key is generated and stored in the private cloud later checked for de-duplication. The user is given access only after verifying their credentials with the data sets. The system begins with authorization and establishing control and key generation for the uploaded file and introducing a hybrid cloud where the uploaded files are stored for later access. Next phase includes detecting de-duplication of the uploaded file, when a user wants to retrieve any file, based on credentials and authenticity of the user is verified and key is exchanged and file can be retrieved.

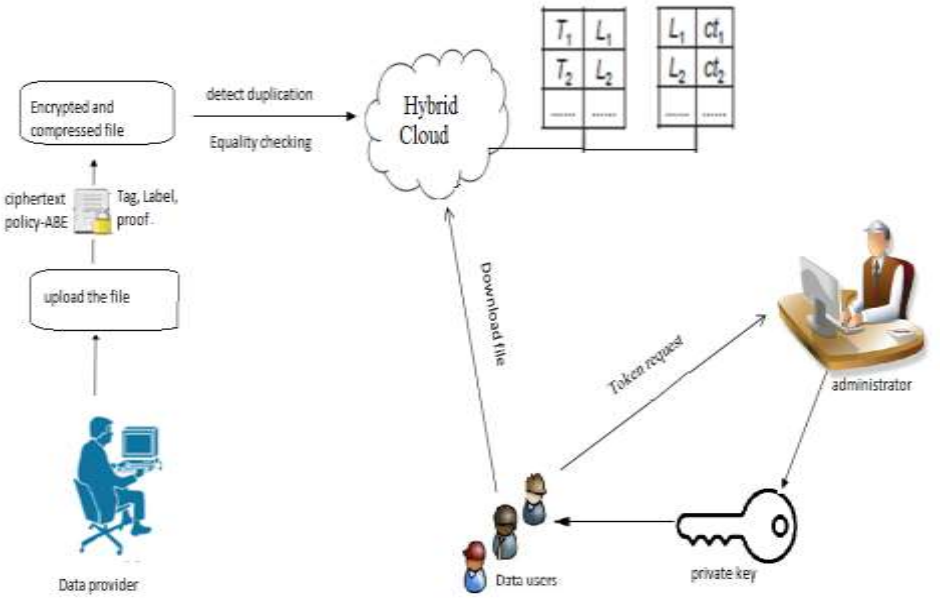

Fig. 5: System Architecture of Cipher Text Attribute-Based System in hybrid cloud storage

6.1 Authorization Control Creation and Key Generation

Authorized user is able to use their individual private keys to generate query for certain file and the privileges owned with the help of private cloud. A user can upload a file only if they possess credentials to log in to the cloud storage. Unauthorized users without appropriate privileges or file can be prevented from accessing a file or generating key. In system, the S-CSP performs the duplicate check upon receiving the request from users. The system validates the user and only after verifying them to be legitimate permission is given to upload a file/data. It requires that any user without querying the private cloud server for some file token cannot achieve any useful information from the token, which includes the file information or the privilege information.

6.2 Uploading File in Hybrid Cloud

Hybrid cloud architecture is introduced to solve the problem of de-duplication and to validate legitimate users. It has a private cloud where encryption process and key generation takes place. The public cloud will manage and store the encrypted data and does de-duplication checking. Private keys of legitimate users are also and managed by the hybrid cloud and won't be issued to users directly. To get a file token or key, the user needs to send a request to the private cloud server. To perform the duplicate check for uploaded file, the system will perform equality check algorithm and checks with the entire file in public cloud. The private cloud server will also check the user's identity before issuing the corresponding file token to the user. The authorized duplicate check for this file is performed in the public cloud before uploading this file into hybrid cloud. Based on the results of duplicate check, the user either uploads this file or runs PoW.

6.3 Detect De-duplication:

Convergent encryption provides data confidentiality in de-duplication. A user derives a key from each original data copy and encrypts the data copy with the same key. In addition, the user also derives a *tag* for the data copy, such that the tag will be used to detect duplicates. Here, we assume that the tag correctness property holds, i.e., if two data copies are the same, then their tags are the same. To detect duplicates, the user first sends the tag to the server side to check if the identical copy has been already stored. The key generated and the tag is independently derived and the tag cannot be used to deduce the key and compromise data confidentiality. Both the encrypted data copy and its corresponding tag will be stored on the server side. Now the tags are checked with equality checking algorithm and if they match with each other then the file is already uploaded and current file to be uploaded will be denied the access and information will be passed on to the user.

6.4 Key Exchanging:

The private keys of the legitimate user and encrypted files are stored and managed in the private cloud. The file token requests from the users are considered after verifying the user's authenticity and then based on the file requested. The interface offered by the private cloud allows user to submit files and queries to be securely stored and computed. The private cloud server after checking and verifying user's identity issues the corresponding file token or key to the user. The authorized duplicate check for this file can be performed by the user with the public cloud before uploading this file.

6.5 Verification and File Retrieving:

A symmetric key for each user is selected and set of keys will be sent to the private cloud. An identification protocol equals to proof and verify is also defined, where Proof and Verify are the proof and verification algorithm respectively. In each user *U* is assumed to have a secret key *to* perform the identification with servers. Assume that user *U* has the privilege set *PU*. It also initializes a PoW protocol POW for the file ownership proof. The private cloud server will maintain a table which stores each user's public information pkU and its corresponding privilege. It first sends a request and the file name to the S-CSP. Upon receiving the request and file name, the S-CSP will check whether the user is eligible to download file. If failed, the S-CSP sends back an abort signal to the user

to indicate the download failure. Otherwise, the S-CSP returns the corresponding cipher text *CF*. upon receiving the encrypted data from the S-CSP; the user uses the key *kF* stored locally to recover the original file.

7. RESULT ANALYSIS

Data de-duplication is useful for organizations dealing with highly redundant operations that requires constant copying and storing of data for future reference or recovery purpose N^{th} term is explained as an approach that eliminates duplicate copies of data from the system. For instance, a file that is backed up every week results in a lot of duplicate data and thus, eats up considerable disk space. De-duplication run an analysis and eliminates these sets of duplicate data and keeps only what is unique and essential, thus significantly clearing storage space. Here are some benefits of data de-duplication for organizations. This concept results in eliminating the redundant copy of different formats of dataset in Amazon cloud which saves Clears storage space, Adept replication, Effective use of network bandwidth, cost-effective.

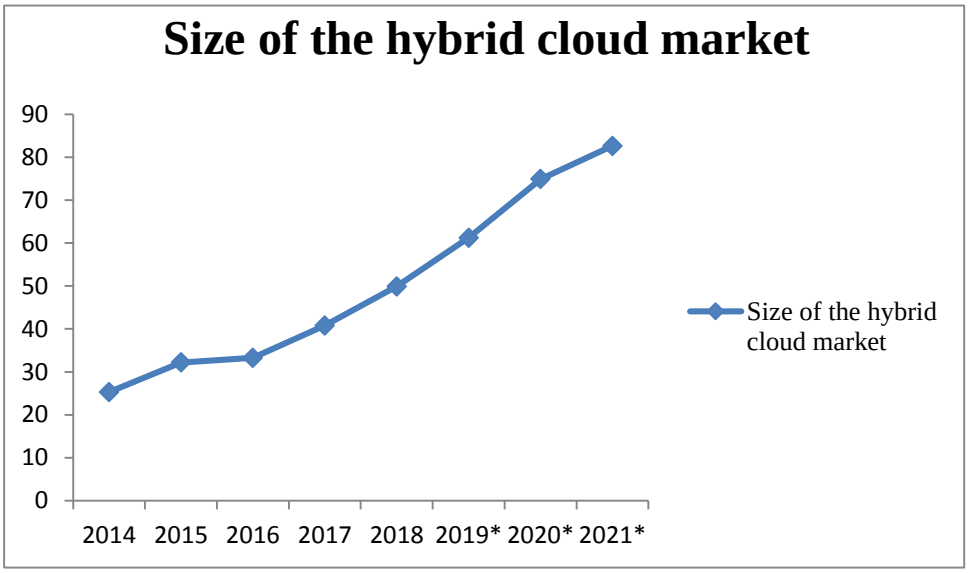

Fig 6: Graph indicating the growth in the size of Hybrid Cloud market

Though the above solution supports the differential privilege duplicate, it is inherently subject to brute force attacks launched by the public cloud server, which can recover files falling. The main idea of this technique is that the novel encryption key generation algorithm. For simplicity, the hash function is used to define the tag generation functions and convergent keys in this section.

8. CONCLUSION

The Cipher Text Attribute-based encryption (ABE) is used in cloud computing, as it is convenient for users to outsource their encrypted data and can share their data only with legitimate users with specific credentials. Also to save the storage space and network bandwidth it is necessary to check and confiscate duplicate copies, using equality check algorithm it is possible to eliminates duplicate copies of identical data. The Cipher Text Attribute-based encryption (ABE) system will support to confiscate de-duplication. This storage system is built under a hybrid cloud architecture, where a private cloud controls the computation and a public cloud manages the storage. The private cloud is provided with a key associated with the corresponding ciphertext, with which it can transfer the cipher text over one access policy into cipher text of the same plaintext under any other access policies without being aware of the underlying plaintext. After receiving a storage request, the private cloud first checks the validity of the uploaded item through the attached proof. If the proof is valid, the private cloud runs a tag matching algorithm to see whether the same data underlying the ciphertext has been stored. If so, whenever it is necessary, it regenerates the ciphertext into a ciphertext of the same plaintext over an access policy which is the union set of both access policies. The proposed storage system provides two major advantages such as, it can be used to confidentially share data with other users by specifying an access policy rather than sharing the decryption key and, it achieves the standard notion of semantic security while existing deduplication schemes only achieve it under a weaker security notion.

9. REFERENCES

[1] Hui Cui, Robert H. Deng, Yingjiu Li, and Guowei Wu, "Attribute-Based Storage Supporting SecureDeduplication of Encrypted Data in Cloud", IEEE Transactions on Big Data, 2017.

[2] Raghi Roy and Paul P. Mathai "Proxy Re-encryption Schemes for Secure Cloud Data and Applications: A Survey", International Journal of Computer Applications (0975 - 8887) Volume 164 - No.5, April 2017.

[3] K. R. Choo, J. Domingo-Ferrer, and L. Zhang, "Cloud cryptography: Theory, practice and future research directions," Future Generation Comp. Syst., vol. 62, pp. 51–53, 2016.

[4] K. R. Choo, M. Herman, M. Iorga, and B. Martini, "Cloud forensics: State-of-the-art and future directions," Digital Investigation, vol. 18, pp. 77–78, 2016.

[5] Y. Yang, H. Zhu, H. Lu, J. Weng, Y. Zhang, and K. R. Choo, "Cloud based data sharing with fine-grained proxy re-encryption," Pervasive and Mobile Computing, vol. 28, pp. 122–134, 2016.

[6] M. Bellare and S. Keelveedhi, "Interactive message-locked encryption and secure deduplication," in Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings, ser. Lecture Notes in Computer Science, vol. 9020. Springer, 2015, pp. 516–538.

[7] D. Quick and K. R. Choo, "Google drive: Forensic analysis of data remnants," J. Network and Computer Applications, vol. 40, pp. 179– 193, 2014.

[8] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings, ser. Lecture Notes in Computer Science, vol. 7881. Springer, 2013, pp. 296–312.

[9] M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev, "Message-locked encryption for lock-dependent messages," in Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I, ser. Lecture Notes in Computer Science, vol. 8042. Springer, 2013, pp. 374–391.

[10] S. Keelveedhi, M. Bellare, and T. Ristenpart, "Dupless: Serveraided encryption for deduplicated storage," in Proceedings of the 22th USENIX Security Symposium, Washington, DC, USA, August 14-16, 2013. USENIX Association, 2013, pp. 179–194.

[11] S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider, "Twin ¨ clouds: Secure cloud computing with low latency - (full version)," in Communications and Multimedia Security, 12th IFIP TC 6 / TC 11 International Conference, CMS 2011, Ghent, Belgium, October 19- 21,2011. Proceedings, ser. Lecture Notes in Computer Science, vol. 7025. Springer, 2011, pp. 32–44.

[12] Mihir Bellare and Adriana Palacio, "GQ and Schnorr Identification Schemes: Proofs of Security against Impersonation under Active and Concurrent Attacks" Advances in Cryptology – CRYPTO '02, LNCS 2442, pp 162-177, 2002.

[13] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer," Reclaiming Space from Duplicate Files in a Server less Distributed File System Prediction", pp. 617–624, ICDCS, 2002.

[14] Dayananda RB1, Prof. Dr. G.Manoj Someswar," Time-Based Proxy Re-encryption Scheme for Secure Data Sharing in a Cloud Environment", International Journal of Emerging Trends in Science and Technology, Vol.02, Issue01,Pages 1699-1710, January 2015, ISSN 2348-9480.

