International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Data Security Policies in Cloud: A Survey

Kasundra Punitkumar R.[1], **Shikha J. Pachouly**[2]

[1]Student, Department of Computer engineering, AISSMSCOE, Kennedy Road, Pune University, Pune, India

[2]Department of computer engineering, AISSMSCOE, Kennedy Road, Pune University, Pune, India

Abstract: ***In our modern day and age, many enterprises are embracing cloud computing. However, one of the major concerns regarding cloud computing has always been security. Encryption in cloud is still in a state of flux and infancy. Some vendors provide encryption, while others don't. There are different kinds of encryption schemes for securing data in the cloud, sometimes integrated within a system. Whenever a company decides it move its applications to the cloud, it considers several pros and cons before doing so.in this paper various cloud data security policies like Mediated certification ,Role based access control, data deduplication and data self-destruction are presented which assist in protection of data in cloud.***

Keywords*:* Encryption, Data deduplication, data self-destruction, role base access, reencryption

1. Introduction

Cloud computing has transformed the way organizations approach IT, enabling them to become more agile, introduce new business models, provide more services, and reduce IT costs. Cloud computing technologies can be implemented in a wide variety of architectures, under different service and deployment models, and can coexist with other technologies and software design approaches. Data protection tops the list of cloud concerns today. When it comes to public private, and hybrid cloud solutions, the possibility of compromised information creates tremendous angst. Organizations expect third-party providers to manage the cloud infrastructure, but are often uneasy about granting them visibility into sensitive data. Protecting your data in the cloud is done by implemented to access control lists to define the permissions. Attached to the data objects. Storage encryption to protect against unauthorized access at the data center (especially by malicious IT staff).Transport level encryption to protect data when it is transmitted. Firewalls to include web application firewalls to protect against outside attacks launched against the data center. Hardening of the servers to protect against known, and unknown, vulnerabilities in the operating system and software. Physical security to protect against unauthorized physical access to data. This paper is organized in following manner Section. 2 discusses the policies used in data security Mediated certificateless public key encryption [1] also Secure Authorized Deduplication [9] also other policies Section. 3 gives Finding and section.4 provides Conclusion.

2. Existing Policies in Cloud Security

2.1Mediated certificateless public key encryption

In An Efficient Certificateless Encryption for Secure Data Sharing in Public Clouds [1] Seung-Hyun Seo, Mohamed Nabeel and Xiaoyu Ding proposed Mediated certificateless public key encryption (mCL-PKE).This algorithm presents a feasible replacement for traditional public key cryptosystem that requires Certificate Authority (CA)to issue digital certificate to bind user to their public keys .CA generates its own signature on each user public keys and manage and manage user certificate , thus overall key management in this system is way too expensive and complex . The newly proposed system uses Attribute Based Encryption (ABE). In ABE Access control policy is employed to encrypt each data to solve the key escrow problem Certificateless public key cryptography (CL-PKC) [2] is introduced which then further improved in Certificateless proxy reencryption [3] which is based on CL-PKC. The CL-PRE as shown in figure-1 uses a pairing operation. In this algorithm user first provide its ID to the cloud which returns private key to the user in interim also item to the owner ., the owner of plain text then encrypt the data using the public key .and transmit the encrypted data along the public keys to the cloud and the user downloads and decrypt the data. Now we will look at the mCL-PKE algorithm.

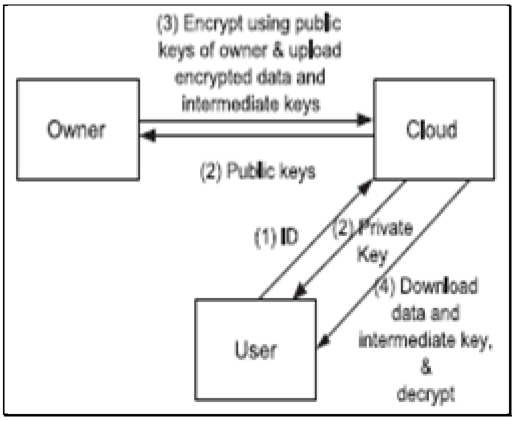

(3) Encrypt using public keys of owner & upload encrypted data and intermediate keys
Owner
Cloud
(2) Public keys
(1) ID
(2) Private Key
(4) Download data and intermediate key, & decrypt
User

Figure 1: Operation scheme of CL-PRE

The mediated certificateless public key encryption scheme algorithm is defined by seven tuple which consist of SetUp, *SetPrivateKey, SetPublicKey, SEM-KeyExtract, Encrypt, SEM-Decrypt, and USER-Decrypt)* and each of them is defined below:

SetUp: The input to the setup is **K** and **mk**
Where,
k- Security Parameter
mk- Secret Master Key
SetPrivateKey: Input- param and ID (owner) Output- SK_{id}
Secret valueSK_{id} is returned to the owner.
SetPublicKey: Input – params and SK_{id}

Output – user's public key PK_{id}

SEM-KeyExtract: In this stepuser ID and PK_{id} are registered with basic algorithm KGC the authorization of public key to its private key is done by KGC.

Input to KGC: parameters mk and the ID of the user. Output: SEM-key corresponding to ID required during decryption time by the SEM.

Encrypt- Input: params, user identity ID, PK_{id} and Message M

Output:C_{id} or symbol ⊥

Where,

C_{id} - cipher text

⊥ - Error

SEM-Decrypt- Input: params, SEM-key and C_{id}

Output:C_{id}' or ⊥

Where,

SEM-key- Security mediator key.

C_{id}' -partially cipher text

USER-Decrypt- Input: params, SK_{id} and c'_{id} .

Output: fully decrypt message M

Decryption of Security Model of Mediated CL-PKE (mCL-PKE):

SetUp: The Challenger runs Setup by taking a security parameter *k* as input in order to return system parameters params and a master key mk. The Challenger gives params to the adversary *A* and Keeps mk secret.

- Phase 1: The adversary *A* can adaptively make various queries and the Challenger can respond to the queries as follows:

SEM-key for ID Extraction: The Challenger runs SEM-KeyExtract to generate the SEM-key d_0 using an identity *ID* and params as the input. Public Key Request for *ID*: The Challenger runs SetPrivateKey to generate SK_{id} and then runs SetPublicKey to generate the public key PK_{id} using *ID*, *SKID* and params as the input. It returns PK_{id} to *A*.

Public Key Replacement: The adversary A can repeatedly replace the public key for any identity with any value of its choice. The SEMkeyis also updated if the Challenger bundles the public key with the identity for SEM key creation. The replaced public key will be used in the rest of the game unless replaced again.

Private Key Extraction for ID: The Challenger runs SetPrivateKey to generate SK_{id} using *ID as* the input. It returns *SKID* to *A*. However, if the publickey of *ID* has been already replaced by the adversary*A*, this query is disallowed.

SEM-Decryption: The adversary provides C_{id} an identity *ID* and a cipher text. The Challenger responds with the partial decryption C_{id} under the SEM-keyC_0 that is associated with the identity *ID*.

USER-Decryption: The adversary provides an identity *ID* and a cipher textC_{id}. The Challenger responds with the decryption of C_{id}under the private key *SKID* that is associated with the identity *ID*. Challenge Phase: Once *A* determines that Phase 1 is over, it outputs a challenge identity *ID*and a pair of plaintext *(M*0, *M1)* with an equal length. In case that *A* is an*AI*, it chooses a public key of identity *ID*, $PK_{id}$by using the Public Key Replacement query. For the Identity*ID*, *A* $_i$cannot ask both the SEM-Key. Extraction query and Private Key Extraction query.

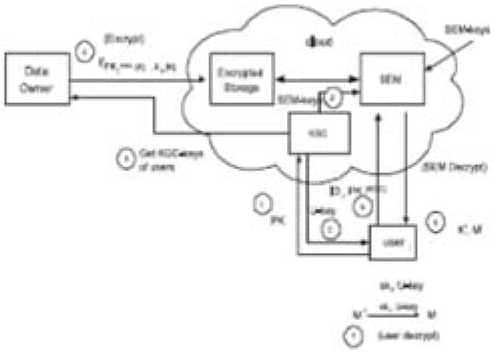
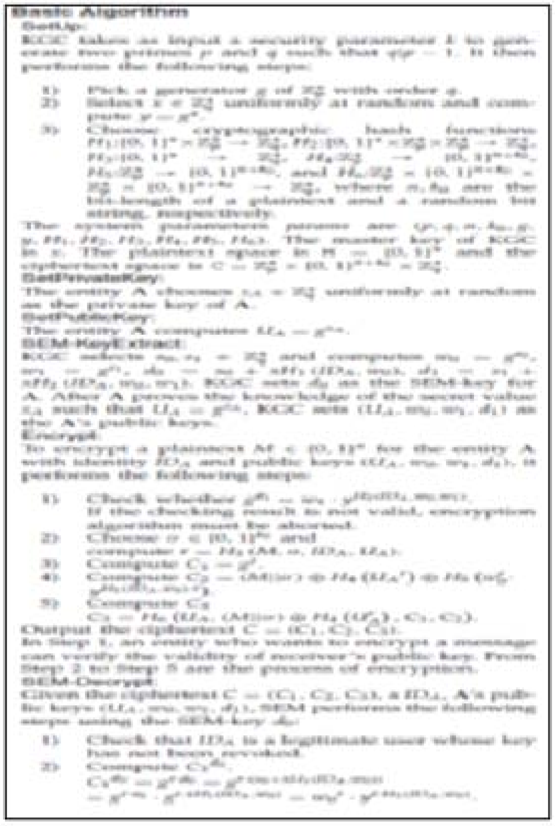


Figure 2: Architecture for mCL-PKE



Basic Algorithm

SetUp:

KGC takes as input a security parameter k to generate two primes p and q such that $q|p-1$. It then performs the following steps:

1) Pick a generator g of $\mathbb{Z}_p^*$ with order q.
2) Select $x \in \mathbb{Z}_q^*$ uniformly at random and compute $y = g^x$.
3) Choose cryptographic hash functions $H_1:\{0,1\}^* \times \mathbb{Z}_p^* \to \mathbb{Z}_q^*$, $H_2:\{0,1\}^* \times \mathbb{Z}_p^* \times \mathbb{Z}_p^* \to \mathbb{Z}_q^*$, $H_3:\{0,1\}^* \to \mathbb{Z}_q^*$, $H_4:\mathbb{Z}_p^* \to \{0,1\}^{n+k_0}$, $H_5:\mathbb{Z}_p^* \to \{0,1\}^{n+k_0}$, and $H_6:\mathbb{Z}_p^* \times \{0,1\}^{n+k_0} \times \mathbb{Z}_p^* \to \mathbb{Z}_q^*$, where n, k_0 are the bit-length of a plaintext and a random bit string, respectively.

The system parameters params are $(p, q, n, k_0, g, y, H_1, H_2, H_3, H_4, H_5, H_6)$. The master key of KGC is x. The plaintext space is $\mathcal{M} = \{0,1\}^n$ and the ciphertext space is $\mathcal{C} = \mathbb{Z}_p^* \times \{0,1\}^{n+k_0} \times \mathbb{Z}_q^*$.

SetPrivateKey:

The entity A chooses $z_A \in \mathbb{Z}_q^*$ uniformly at random as the private key of A.

SetPublicKey:

The entity A computes $U_A = g^{z_A}$.

SEM-KeyExtract:

KGC selects $s_0, s_1 \in \mathbb{Z}_q^*$ and computes $w_0 = g^{s_0}$, $w_1 = g^{s_1}$, $d_0 = s_0 + xH_1(ID_A, w_0)$, and $d_1 = s_1 + xH_2(ID_A, w_0, w_1)$. KGC sets d_0 as the SEM-key for A. After A proves the knowledge of the secret value z_A such that $U_A = g^{z_A}$, KGC sets (U_A, w_0, w_1, d_1) as the A's public keys.

Encrypt:

To encrypt a plaintext $M \in \{0,1\}^n$ for the entity A with identity ID_A and public keys (U_A, w_0, w_1, d_1), it performs the following steps:

1) Check whether $g^{d_1} = w_1 \cdot y^{H_2(ID_A, w_0, w_1)}$. If the checking result is not valid, encryption algorithm must be aborted.
2) Choose $\sigma \in \{0,1\}^{k_0}$ and compute $r = H_3(M, \sigma, ID_A, U_A)$.
3) Compute $C_1 = g^r$.
4) Compute $C_2 = (M\|\sigma) \oplus H_4(U_A^r) \oplus H_5((w_0 y^{H_1(ID_A, w_0)})^r)$.
5) Compute C_3 $C_3 = H_6(U_A, (M\|\sigma) \oplus H_4(U_A^r), C_1, C_2)$.

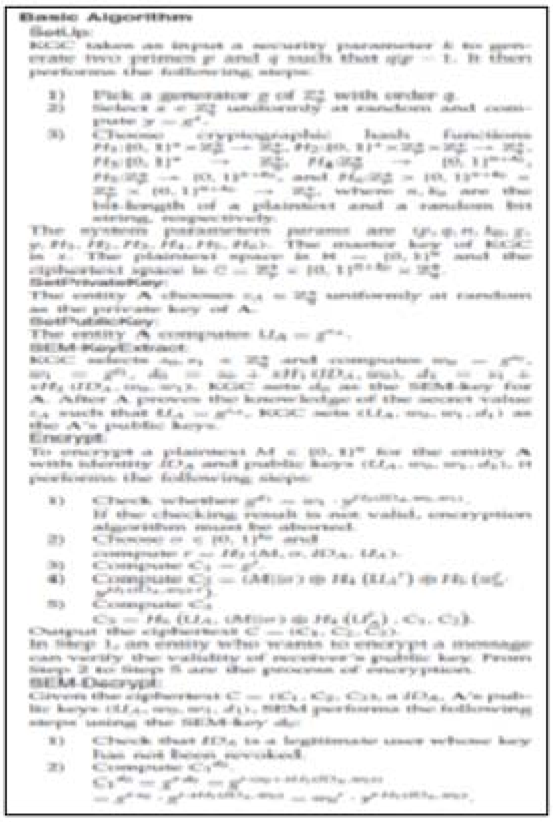
Output the ciphertext $C = (C_1, C_2, C_3)$.

In Step 1, an entity who wants to encrypt a message can verify the validity of receiver's public key. From Step 2 to Step 5 are the process of encryption.

SEM-Decrypt:

Given the ciphertext $C = (C_1, C_2, C_3)$, a ID_A, A's public keys (U_A, w_0, w_1, d_1), SEM performs the following steps using the SEM-key d_0:

1) Check that ID_A is a legitimate user whose key has not been revoked.
2) Compute $C_1^{d_0}$.
$C_1^{d_0} = g^{rd_0} = g^{r(s_0 + xH_1(ID_A, w_0))}$
$= g^{rs_0} \cdot g^{rxH_1(ID_A, w_0)} = w_0^r \cdot y^{rH_1(ID_A, w_0)}$.



Basic Algorithm

SetUp:

KGC takes as input a security parameter k to generate two primes p and q such that $q|p-1$. It then performs the following steps:

1) Pick a generator g of $\mathbb{Z}_p^*$ with order q.
2) Select $x \in \mathbb{Z}_q^*$ uniformly at random and compute $y = g^x$.
3) Choose cryptographic hash functions $H_1: \{0,1\}^* \times \mathbb{Z}_p^* \rightarrow \mathbb{Z}_q^*$, $H_2: \{0,1\}^* \times \mathbb{Z}_p^* \times \mathbb{Z}_p^* \rightarrow \mathbb{Z}_q^*$, $H_3: \{0,1\}^* \rightarrow \mathbb{Z}_q^*$, $H_4: \mathbb{Z}_p^* \rightarrow \{0,1\}^{n+k_0}$, $H_5: \mathbb{Z}_p^* \rightarrow \{0,1\}^{n+k_0}$, and $H_6: \mathbb{Z}_p^* \times \{0,1\}^{n+k_0} \times \mathbb{Z}_p^* \times \mathbb{Z}_p^* \rightarrow \mathbb{Z}_q^*$, where n, k_0 are the bit-length of a plaintext and a random bit string, respectively.

The system parameters params are $(p, q, n, k_0, g, y, H_1, H_2, H_3, H_4, H_5, H_6)$. The master key of KGC is x. The plaintext space is $\mathcal{M} = \{0,1\}^n$ and the ciphertext space is $\mathcal{C} = \mathbb{Z}_p^* \times \{0,1\}^{n+k_0} \times \mathbb{Z}_q^*$.

SetPrivateKey:

The entity A chooses $z_A \in \mathbb{Z}_q^*$ uniformly at random as the private key of A.

SetPublicKey:

The entity A computes $U_A = g^{z_A}$.

SEM-KeyExtract:

KGC selects $s_0, s_1 \in \mathbb{Z}_q^*$ and computes $w_0 = g^{s_0}$, $w_1 = g^{s_1}$, $d_0 = s_0 + xH_1(ID_A, w_0)$, $d_1 = s_1 + xH_2(ID_A, w_0, w_1)$. KGC sets d_0 as the SEM-key for A. After A proves the knowledge of the secret value z_A such that $U_A = g^{z_A}$, KGC sets (U_A, w_0, w_1, d_1) as the A's public keys.

Encrypt:

To encrypt a plaintext $M \in \{0,1\}^n$ for the entity A with identity ID_A and public keys (U_A, w_0, w_1, d_1), it performs the following steps:

1) Check whether $g^{d_1} = w_1 \cdot y^{H_2(ID_A, w_0, w_1)}$. If the checking result is not valid, encryption algorithm must be aborted.
2) Choose $\sigma \in \{0,1\}^{k_0}$ and compute $r = H_3(M, \sigma, ID_A, U_A)$.
3) Compute $C_1 = g^r$.
4) Compute $C_2 = (M \| \sigma) \oplus H_4(U_A^r) \oplus H_5(w_0^r \cdot y^{rH_1(ID_A, w_0)})$.
5) Compute C_3: $C_3 = H_6(U_A, (M \| \sigma) \oplus H_4(U_A^r), C_1, C_2)$.

Output the ciphertext $C = (C_1, C_2, C_3)$.

In Step 1, an entity who wants to encrypt a message can verify the validity of receiver's public key. From Step 2 to Step 5 are the process of encryption.

SEM-Decrypt:

Given the ciphertext $C = (C_1, C_2, C_3)$, a ID_A, A's public keys (U_A, w_0, w_1, d_1), SEM performs the following steps using the SEM-key d_0:

1) Check that ID_A is a legitimate user whose key has not been revoked.
2) Compute $C_1^{d_0}$: $C_1^{d_0} = g^{rd_0} = g^{r(s_0 + xH_1(ID_A, w_0))} = g^{rs_0} \cdot g^{rxH_1(ID_A, w_0)} = w_0^r \cdot y^{rH_1(ID_A, w_0)}$.

Figure 3: Algorithm for mCL-PKE

2.2 Role-Based Access Control on Encrypted Data

Lan Zhou, Vijay and Michael in Achieving Secure Role-Based Access Control on Encrypted Data in Cloud Storage (RBAC) [4] proposed a frame work for data security in cloud. In RBAC the access control policies are enforced by a new Role-Based Encryption (RBE) on Encryption (RBE) proposed by the author. The enforcement of policies is done is done by RBE on encrypted data by revocation using broadcast encryption mechanism[5].In this methodology the owner of the data encrypts the data in such a way that only user with appropriate role as specified by RBAC policy can decrypt and view the data. The architecture of this framework is given in figure. 4. RBAC works on environment as given in figure. We can have a look at the component of this framework in step by step manner.

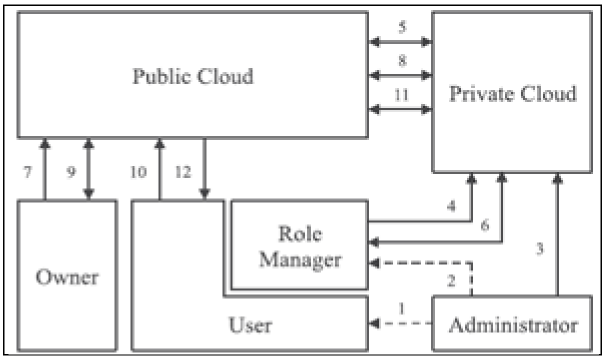


Figure 4: RBE system architecture

A) Public cloud-A hybrid cloud structure is employed which consist of a private cloud for the storage of sensitive data. The job of the private cloud is to provide interface to the administrator and the role manager of the role based system and to public cloud
B) .there is no access permission for the user in the private cloud.
C) User- user are the parties who wish to acquire certain data from public cloud. the authorization of the user is done by the administrator of role based system
D) Role manager - Role manager a role manager is the party who manages the relationship between the user and their roles each role has its own parameter which defines user membership.
E) Administrator – the administrator is the certificate authority. All the necessary parameters and credentials are issued by the administrator.
F) Owner-data and the encrypted data is stored on cloud for other users to access belongs to the owner.
G) Here ID-based signature (IBS) [6] scheme in the system to certify the data communicated between the different parties. Using this technique the receiving party can verify the integrity of data content and authenticate.
H) **Setup** **(**λ**)** takes as input the security parameter λ and outputs a master secret key mk and a system public key pk. mk is kept secret by the SA while pk is made public to all users of the system.

Extract **(**mk**,** ID**)** is executed by the SA to generate the key associated with the identity ID. If ID is the identity of a user, the generated key is returned to the user as the decryption key. . If ID is the identity of a role, the generated key is returned to the RM as the secret key of the role, and an empty user list *RUL* which will list all the users with Manage Role **(**mk**,** ID*R***,** *PRR***)** is executed by the SA to Manage a role with the identity ID*R* in the role hierarchy. PR_r is the set of roles which will be the ancestor roles of the role. This operation publishes a set of public parameters $pub_r$to cloud. Q is the member of that role is also returned to the RM.

Add User **(**P_k**,** SK_r**,** RUL_r**,** ID$_u$**)** is executed by the role manager R_M of a role *R* to grant the role membership to the user ID_u, which results in the role public parameters $PUB_r$and role user list RUL_r, being updated in cloud.

Revoke User **(**P_k**,** SK_r**,** RUL_r**,** ID$_u$**)**is executed by a role manager R_m of a role *R* to revoke the role membership from a user ID*U* , which also results in the role public parameters $PUB\ _r$ and role user list RUL_r, being updated in cloud.

Encrypt **(**P_k,$PUB\ _r$**)** is executed by the owner of a message *M*. This algorithm takes as input the system public keyP_k, the role public parameters$PUB\ _r$, and outputs a tuple *C, K*, where *C* will be a part of the cipher text, and *K* ∈*K* is the key that will be used to encrypt the message *M*.

Decrypt (P$_k$, PUB$_r$, d$_k$, C**)** is executed by a user who is a member of the role *R*. This algorithm takes as input the system public key pk, the role public parameters PUB$_r$, the user decryption keyd$_k$,, the part *C* from the cipher text downloaded from cloud, and outputs the message encryption key *K* ∈*K*. The key *K* can then be used to decrypt the cipher text part *EncK (M)* and obtain the message *M*.

2.3 Secure Authorized Deduplication

In A Hybrid Cloud Approach for Secure Authorized Deduplication, Jin Li, Yan, Xiaofeng and Patrick [9] proposed a system for data deduplication. Data deduplication is one of the popular compression technique used for eliminating duplicate copies of repeating data. This policy uses convergent encryption scheme for encrypting the data before outsourcing. The architecture of the authorized deduplication is shown in figure 5.

System Model
Hybrid Architecture for Secure Deduplication- There are three components defined in the architecture in this system i.e. user, private cloud, S-CSP.

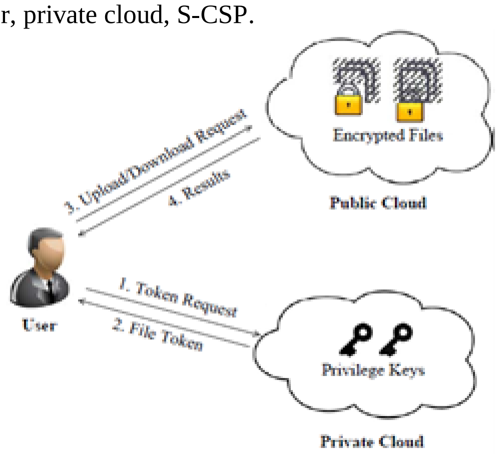


Figure5: Architecture for secure data Deduplication

S-CSP- It performs the deduplication task. The deduplication is performed by checking whether two files have same content. If the files are same the one of the copy is eliminated.

Data Users. A user is an entity that wants to outsource data storage to the S-CSP and access the data later. In a storage system supporting deduplication, the user only uploads unique data but does not upload any duplicate data to save the upload bandwidth, which may be owned by the same user or different users. In the authorized deduplication system, each user is issued a set of privileges in the setup of the system. Private Cloud Compared with the traditional deduplication architecture in cloud computing, this is a new entity introduced for facilitating user's secure usage of cloud service.

Advisory model- Under this, two kinds of adversaries are considered, that is,

1) External adversaries which aim to extract secret information as much as possible from both public cloud and private cloud;
2) Internal adversaries who aim to obtain more information on the file from the public cloud and duplicate-check token information from the private cloud outside of their scopes.

2.4 Data Self-Destructing Scheme

Jinbo Xiong and Zhiqiang in A secure data self-destructing scheme in cloud computing [7] proposed a system for key-policy attribute-based encryption with time specified attributes (KP-TSABE), a novel secure data self-destructing scheme in cloud computing. Attribute based encryption (ABE) [8] is used for public key encryption instead of one-to-one encryption because it achieves flexible one-to-many Encryption.

1) *Data Owner*- Data owner can provide data or files that contain some sensitive information, which are used for sharing with his/her friends (data users). All these shared data are outsourced to the cloud servers to store.
2) *Authority*-It is an indispensable entity which is responsible for generating, distributing and managing all the private keys, and is trusted by all the other entities involved in the system.
3) *Time Server*- It is a time reference server without any interaction with other entities involved in the system. It is responsible for a precise release time specification.
4) *Data Users*- Data users are some peoples who passed the identity authentication and access to the data outsourced by the data owner. Notice that, the shared data can only be accessed by the authorized users during its authorization period.
5) *Cloud Servers*- It contains almost unlimited storage space which is able to store and manage all the data or files in the system. Other entities with limited storage space can store their data to the cloud servers.
6) *Potential Adversary*- It is a polynomial time adversary and described in the security model of the KP-TSABE scheme.

Formal Model of KP-TSABE- the KP-TSABE scheme can be described as a collection of the following four algorithms: Setup,Encrypt, KeyGen, and Decrypt.Setup (1*to U*): This algorithm is run by the Authority and takes as input the security parameter and attribute universe *U*, generates system publicParameters *params* and the master key MSK. The Authority publishes *params* and keeps MSK secret to itself.

3. Findings

In Certificateless Encryption [10] we achieve the following goals. Firstly, data owner encrypts the data encryption key once for a data item and provides some additional information to the cloud so that authorized users can decrypt the content using their private keys. Secondly the idea is similar to Proxy Re-Encryption (PRE) where the content encrypted using the data owner's public key is allowed to be decrypted by different private keys after some transformation by the cloud which acts as the proxy. Thirdly, encryption and decryption operation are time-efficient. We can deduce from Figure.6 that the time required performing the encryption operation in the mCL-PKE scheme for different message sizes. Since this scheme does not use pairing operations, it performs encryption efficiently. As can be seen from the graph, the encryption time increases linearly as the message size increases.

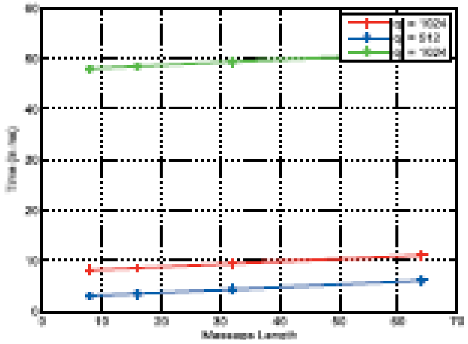


Figure 6: Certificateless encryption vs. Lie al’s scheme [10]

Also deduction can be made from analyzing Figure .7 that it is evident from the graph that as more users are allowed to access the same data item, the better the improved scheme performs compared to the basic scheme. The cost of the basic scheme is high since the encryption algorithm is executed for each user.

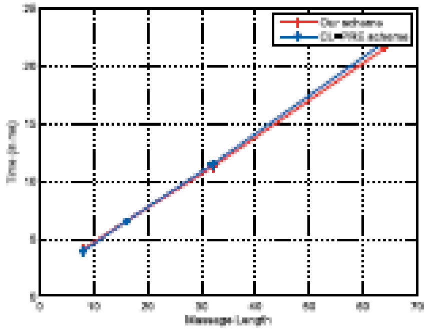


Figure7: Comparison of time to perform encryption and decryption certificateless encryption [10]

Though, Secure Authorized Deduplication [9] is armored with advanced deduplication system supporting authorized duplicate check employing hybrid cloud architecture.

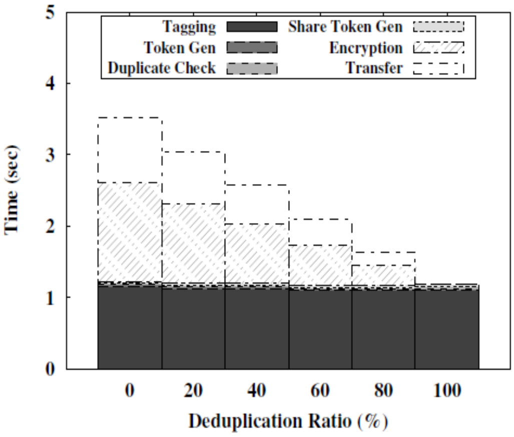


Figure 8: Time Breakdown for Different Deduplication Ratio [9]

4. Conclusion

We have done in-depth survey on advance policies adopted in cloud which efficiently handles the problem of defining who, and under which circumstances, can gain legal permission to access data stored on the cloud. Users believe that their information is confidential and protected from everyone just because it belongs to them and is their property. Therefore we can jump to a conclusion that Cloud computing is the future but not if security problems persist.

References

[1] Seung-Hyun Seo, Mohamed Nabeel, Xiaoyu Ding, and Elisa Bertino, “An Efficient Certificateless Encryption for Secure Data Sharing in Public Clouds”, Knowledge and Data Engineering, IEEE Transactions on, vol. 26, no. 9, pp. 2107–2119, 2014.

[2] S. Al-Riyami and K. Paterson, “Certificateless public key cryptography,” in *Proc.ASIACRYPT* 2003, C.-S.Laih, Ed.Berlin, Germany: Springer, LNCS 2894, pp. 452–473.

[3] X. W. Lei Xu and X. Zhang, “CL-PKE: A certificateless proxy reencryption scheme for secure data sharing with public cloud,” in *ACM Symp. Inform. Comput. Commun. Security*, 2012.

[4] Lan Zhou, Vijay Varadharajan, and Michael Hitchens,” Achieving Secure Role-Based Access Control on Encrypted Data in Cloud Storage”, Information Forensics and Security, IEEE Transactions on, vol. 8, no. 12, pp. 1947 – 1960, 2013.

[5] C. Delerablée, “Identity-based broadcast encryption with constant size ciphertexts and private keys,” in *ASIACRYPT* (Lecture Notes in Computer Science), vol. 4833. New York, NY, USA: Springer-Verlag, 2007, pp. 200–215.

[6] P. S. L. M. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater, “Efficient and provably-secure identity-based signatures and signcryption from bilinear maps,” in *ASIACRYPT* (Lecture Notes in Computer Science), vol. 3788. New York, NY, USA: Springer-Verlag, Dec. 2005, pp. 515–532.

[7] Jinbo Xiong, Ximeng Liu, Zhiqiang Yao, Jianfeng Ma, Qi Li, Kui Geng, and Patrick S. Chen, “A secure data self-destructing scheme in cloud computing”, Cloud Computing, IEEE Transactions on, vol. pp, no. 99, pp.1, 2014.

[8] J. Xiong, Z. Yao, J. Ma, F. Li, X. Liu, and Q. Li, “A secure self-destruction scheme for composite documents with attribute based encryption,” Acta Electronica Sinica, vol. 42, no. 2, pp. 366–376, 2014.

[9] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou, “A Hybrid Cloud Approach for Secure Authorized Deduplication”, Parallel and Distributed Systems, IEEE Transactions on, vol. pp, no. 99, pp. 1, 2014.

[10]X. W. Lei Xu and X. Zhang, “CL-PKE: A certificateless proxy reencryptionscheme for secure data sharing with public cloud,” in*ACM* Symp. Inform. Comput. Commun. Security, 2012.