ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
(IJARTET) Vol. 5, Special Issue 14, April 2018

# CDSPD: A NOVEL CERTIFICATELESS DATA SHARING IN CLOUD THROUGH PARTIAL DECRYPTION

**JEEVITHA B K, SHWETHA MARY A, THRIVENI J AND VENUGOPAL K R**
Department of Computer Science and Engineering, University Visvesvaraya College of Engineering,
Bangalore University, Bengaluru- 560056
Contact: bkjeevitha87@gmail.com, shwethamary.rose@gmail.com

**ABSTRACT**

Storing the important data in the cloud has become an essential argument in the computer territory. The cloud enables the user to store the data efficiently and access the data securely. It avoids the basic expenditure on hardware, software and maintenance. Protecting the cloud data has become one of the burdensome tasks in today's environment. Our proposed scheme "Certificateless Data Sharing in Cloud through Partial Decryption" (CDSPD) makes use of Shared Secret Session (3S) key for encryption and double decryption process to secure the information in the cloud. CDSPD does not use pairing concept to solve the key escrow problem. Our scheme provides an efficient secure way of sharing data to the cloud and reduces the time consumption nearly by 50 percent as compared to the existing mCL-PKE scheme in encryption and decryption process.
***Keywords:*** Certificateless Authority; Cloud Data Security; Shared Secret Session (3S) key; Security Mediator; Partial Decryption

## I. INTRODUCTION

Cloud computing has become a significant mold in recent days because of its high demand for the facility provided by different data centers for storing the information. Due to this facility in the cloud, organizations are ready to accept services provided by the various service center to manage their data. While giving services to the organization, the data centers have the responsibility of securing the information from attackers.

Organizations make use of public cloud services like Dropbox to manage their data. Due to the extensive use of cloud services, the design of the data storage faces the problem of data confidentiality and accessing the data by unauthorized users. To store the data in the cloud, the standard approach used is to encrypt the data and store securely. The organizations are needed to enforce the concept of Fine-Grained Access Control (FGAC) [1] of the data, in which it gives a transparent security policy through a low level of granularity. By using FGAC scheme, policies are generated by the policy enforcer for each data. Initially, the data is uploaded by the owner to the owner. To generate the policy, the user has to send his/her credentials to the policy enforcer who resides in the cloud. This policy is transferred to both the user and the owner. Then the user requests to get register in the owner database along with the policy. The owner verifies the policy, if successful, sends the authentication success to the cloud. In turn, the cloud receives the success message and sends the data to the user. If the authentication fails, then the registration also fails. Policy enforcer is not having any authority to expose the information about the policies and the credentials of the access control. So the encryption method is used to support fine-grained encryption-based access control.

Fig 1 shows the fine-grained encryption based access control approach. After registering with the owner, the user gets the keys. Then the owner selectively encrypts the data and uploads to the cloud. The user downloads the corresponding file and decrypts to get the original file. The owner also downloads the file to re-encrypt the updated data.

Public-key cryptosystem is used to reduce the overhead of key management. Certificate Authority (CA) generates the certificates that verify the authenticity of the public key in traditional public key cryptography. Managing these certificates is both complex as well as costly. So Shamir et al., [2] eliminated the concept by using a new idea called Identity-based Cryptography (ID-PKC) by delivering the public keys directly to the email address of the user. Private Key Generator (PKG) is the trusted third party, which is used to generate private key by verifying the credentials of the user. Sometimes this PKG may be compromised with the int uders that reveal the private key which leads to Key Escrow Problem (KEP). This KEP of ID-PKC is a security issue because a third party is involved which is solved by using CertificateLess Public Key

All Rights Reserved © 2018 IJARTET

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
(IJARTET) Vol. 5, Special Issue 14, April 2018

Cryptography (CL-PKC) by Al Riyami et al., [3]. In this scheme, each user is influenced by the combination of partial private key and another user-chosen secret key which is produced by Key Generation Center (KGC). CL-PKC deletes the concept of managing the certificates as CL-PKC itself validates the user's public key. The idea of bilinear pairing makes the CL-PKE costly. Lei et al., [3] proposed the CertificateLess Proxy Re-Encryption (CL-PRE) scheme in securing the shared data in the environments of the public cloud. The author uses pairing concept that leads to computational costs which achieves Chosen Plaintext Attack (CPA). CPA is not enough for security purpose in sharing data which is against Chosen Ciphertext Attack (CCA).

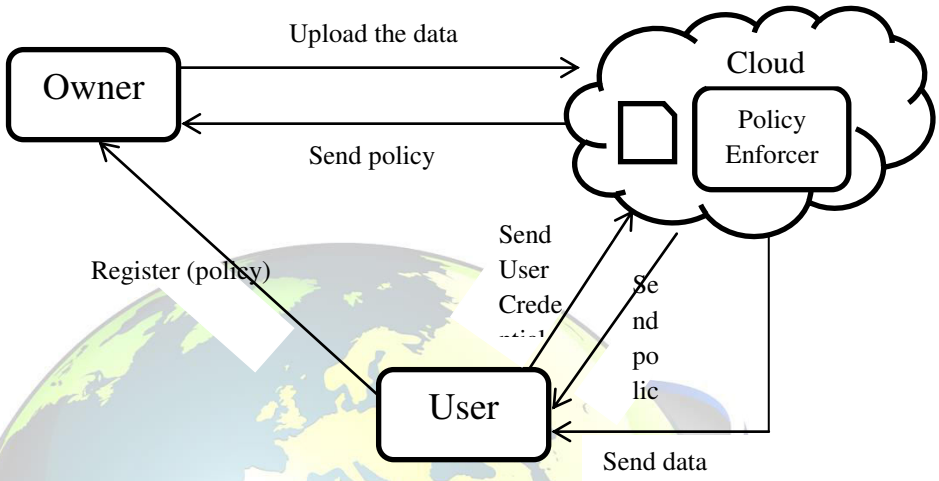


**Fig 1: Fine-grained Encryption**

This paper mainly concentrates on the drawback of the existing approach like a mediated Certificateless Public Key Encryption (mCL-PKE) [14]. mCL-PKE makes use of asymmetric encryption approach which leads to difficulty in managing the two keys. Hence "Certificateless Data Sharing in Cloud through Partial Decryption" (CDSPD) has been proposed that does not use pairing (bilinear) operations which is more expensive that has been proved by the recent research. This paper mainly focuses on the security of the data that guarantees the confidentiality of the data that is stored in public clouds by using encryption and double decryption. To give the double security, partial decryption process of data takes place at the cloud end, and complete decryptionat the user end.

***Motivation:***

Asymmetric keys are used which increases the overhead to the user while decrypting the shared data, and single decryption will put the entire overhead on the user. The main point is to decrease the overhead in maintaining two keys.

***Contribution:***

In this paper, we have developed a novel approach known as CDSPD, to achieve the confidentiality of the data.

- CDSPD scheme provides the high security on sharing the data.
- CDSPD reduces the key escrow problem.
- Only valid users can decrypt the stored information.
- The overhead is reduced on user side as the partial encryption is done in the cloud.
- The CDSPD scheme is more flexible and highly efficient than the existing scheme.

***Organization:***

The remainder of this paper is formed as follow: Section II discuss about the Literature Survey made to present this approach. Section III describes the System Architecture of the CDSPD. Section IV gives the algorithm used. Section V shows the Results and Analysis of the proposed scheme and Section VI Concludes the paper.

## II. RELATED WORK

Predicate Encryption (PE) [4] is a new standard that is the generalization of Identity Based Encryption (IBE). In PE, the secret keys are combined with predicates and ciphertext are linked with attributes. It generates

All Rights Reserved © 2018 IJARTET

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
(IJARTET) Vol. 5, Special Issue 14, April 2018

a master secret key associated with fine-grained control access to the encrypted data. In PE, the policy which is used for the encryption is made public. The public key is a known variable like user id, email id that is related to the users. Jan et al., [5] proposed a secure protocol to improve the privacy of the data by using bilinear Diffie-Hellman exponent and the Strong Diffie-Hellman assumptions. The author makes use of permissions to control the access of the data. These permissions can be attributes or the roles that the user has. The protocol assures the security for both database and the user like

1. Only authorized users can access the data.
2. The database provider doesn't know anything about the records and also the attributes of the user.

M. Bellare et al., [6] compared the security strengths of public key encryption schemes. The author considered both goals and attack models to define the secure encryptions. The goals consist of indistinguishability of encryption and nonmalleability. Indistinguishability (IND) declares an attacker's inability to know the information about the plaintext concealed with the challenging ciphertext. Non-Malleability (NM) also declares the inability of an attacker of getting the plaintext from the ciphertext, instead the attackers gets the different ciphertext.

Gerome et al., [7] proposed a framework that accomplishes the policy of the access control on XML documents using cryptography. The data owner sends the key to the user. The author achieves access control policy and constructs an XML document by using encryption techniques to implement the policies.

A. Sahai [8] has proposed a scheme that allows one to conceal the policy of the access control by encrypting the message. The message can be taken back by the user only if the user is satisfied by the encoded policy. Attribute-Based Encryption (ABE) is more expressive predicate encryption. The public keys of PE are clarified by the user's attribute set. Key- Policy ABE (KP-ABE) and Cipherte t-Policy ABE (CP-ABE) are two popular extensions of ABE. Whenever there is a change in the group, then the private keys are needed to be updated to the users by re-keying operations to provide backward/forward secrecy. Later, the ABE undergo key escrow problem. N. Shang et al., [9] used different keys to encrypt the data. These keys are issued to users at the registration phase. Then the encrypted data is broadcast to all the users. Due to this process, maintaining the secrecy of the keys is difficult when user groups are dynamic or when ACP policy changed.

M. Nabeel et al., [10] used selective encryption method so that the organizations are requested to manage the keys and also the encryptions before uploading. The authorization policy change lead to communication and computation cost. Two layered encryption approach has been proposed by the author in controlling the above problem and increases the responsibility of the cloud in access control administration.

J Bethencourt et al., [11] proposed a scheme called Ciphertext Policy Attribute-Based Encryption (CP-ABE) that keeps the data secretly even if the server is not fully trusted and stored in encrypted form. CP-ABE is a cryptographic technique for FGAC of the shared data. In CP-ABE, each user is collaborated with a set of attributes. The user data is encrypted based on the attributes to construct the access structure. The user can decrypt the encrypted data only if the attributes satisfies the ciphertext access structure.

W. Lou et al., [12] concentrate on attribute revocation problem which is inconvenient for CP-ABE scheme. The author revokes the user attributes by integrating proxy re-encryption with CPABE which reduces the load on authority. The scheme is secure against Chosen Ciphertext Attack (CCA) [13] and can be applicable to KP-ABE. While storing the encrypted data in the third party, the user selectively shares the data at a coarse-grained level. So Vipul et al., [15] develop a new cryptosystem for fine-grained sharing of encrypted data as Key-Policy Attribute-based Encryption (KP-ABE). In KP-ABE, attributes are considered to generate the private keys associated with access structure.

Zhang et al., [16] proposed the first secure certificateless public key encryption scheme without redundancy. It provides optimal bandwidth and efficient decryption process compared with the existing CL-PKE schemes. The author uses the twinning technique in developing Certificateless Encryption (CLE) that secures the data from adaptive chosen ciphertext attacks in random oracle model. Y. Sun et al., [17] propose a CL-PKE without pairing operations that improve the efficiency by reducing the concept of pairing concept. This scheme surpasses the existing CL-PKE scheme on both ciphertext length and decryption process.

All Rights Reserved © 2018 IJARTET

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
(IJARTET) Vol. 5, Special Issue 14, April 2018

Security Mediator (SEM) is a mediated cryptographic concept which controls the user interaction when the user's public key is revoked. The public key is revoked when the private key is compromised, and it is not safe to use the corresponding public key. S. S. M Chow et al.,[18] addresses this issue by introducing a lightweight extended version of mediated cryptography known as Security-Mediated Certificateless (SMC) cryptography which maintains the revoked keys. It also provides a shield against attackers of ciphertext, crook key generation center or any combination of any crook users. This scheme presents a general construction of particular algorithm which is efficient than identity-based mediated encryption scheme which is based on bilinear pairing.

C Yang et al., [19] solves key escrow problem of traditional Identity-based cryptosystem and provides the revocation property. The SEM contains the information about the partial private key which is involved in decryption or the sign procedure method.

X.W. Lei Xu and X.Zhang [20] proposed a proxy based re-encryption approach that delivers keys of encryption to the access control policy from the data owner. It introduces slight overhead on the user by eliminating the direct interaction between the data owner and its recipients. This CLPRE approach is documented with certificateless public key cryptography which avails the benefits of cloud for not only stored data but also for distributing the keys in safer mode and storing the data.

D. Boneh et al., [1] proposed a new technique of semi-trusted mediator under online mode combined with a simple threshold variant of RSA cryptosystem. The author uses this technique for fine-grained control on security benefits. The combination of online SEM with RSA cryptosystem gives various advantages compared to recent revocation techniques that include simplified validation of digital signatures, efficient certificate revocation for legacy systems and fast revocation of signature and decryption capabilities.

Jeevitha et al., [21] have surveyed on different data security approaches like confidentiality approaches, integrity approaches and data access technologies.

## III. BACKGROUND WORK

### *A. Key-Escrow Problem*

The Key-Escrow problem (KEP) is a data security risk in which a cryptographic key is stored in escrow/saved by the third party. It is a major security issue of identity-based and security-mediated cryptosystems where the trusted third party (TTP) is involved in generating the keys. This TTP may be compromised with the intruders and can decrypt the file in favor of any users. This KEP problem can be resolved temporarily by distributing the capability of third party over several entities.

### *B. Certificateless Cryptography*

It is a variant of ID-based cryptography, intended to prevent the key-escrow problem. Generally, KGC and Certificate authority is given full authority to generate the keys. By combining the merits of traditional PKC and ID-PKC, the approach of cerificateless public-key cryptography was introduced by the Al-Riyami et al., [3]. ID-PKC and CL-PKC rely on the existence of a semi-trusted key-generation center generates a master key and sends the partial key to the user based on the identity of the user. User calculates its own key pair (public/private), and the decryption makes use of both partial private key and the user private key. The key escrow problem will be resolved along with retaining the concept of certificate free nature of the system.

### *C. Security Mediator*

Security Mediator (SEM) [22] can generate signature on the data to the data owner to provide security. SEM does not have any idea of who is the owner, the uploaded data. While verifying the data, SEM cannot recognize the identity of the data uploader, thus the trust on the SEM is minimized. SEM can authenticate each data owner by anonymous credential supporting both revocation and reputation. With these low computational and bandwidth requirements and the low trust level, a typical server of the organization can serve as the SEM.

Fig 2 contains four entities: Data owners, Cloud server, Data users and a Security Mediator.

- Data owner generates and uploads the data for sharing.
- Data user can only access the data uploaded by the owner but cannot modify.
- Cloud server provides data storage and sharing services to the data owner and the users.
- SEM generates the signature on the data before outsourced to the cloud to provide security services.

All Rights Reserved © 2018 IJARTET

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
(IJARTET) Vol. 5, Special Issue 14, April 2018

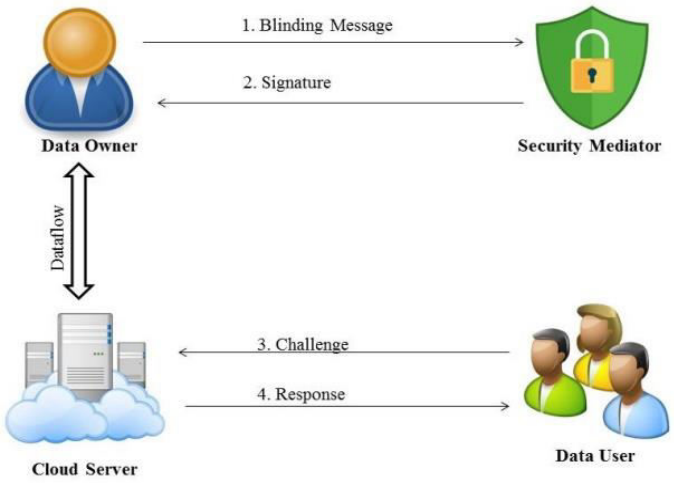


**Fig 2: Security Mediator**

### *D. Triple DES*

Triple DES or 3DES or DES-EDE (Encrypt-Decrypt-Encrypt) is an improvised version of DES algorithm as DES makes use of small key of 56 bit and very easy to crack. Hence in order to provide more security with the bigger key size TDES came into existence. TDES is a block-cipher based symmetric encryption. It uses a secret key for both encryption and decryption operation of the data. It takes the input as fixed-length bits of the plaintext and produces the complicated bits of ciphertext. Both plaintext and ciphertext are of same length. Actual key length of TDES is 64 bits length. Every $8^{th}$, $16^{th}$, $24^{th}$, $32^{nd}$, $40^{th}$, $48^{th}$, $56^{th}$, and $64^{th}$ positioned bit are discarded. So TDES uses only 56 bits of the 64 bits.

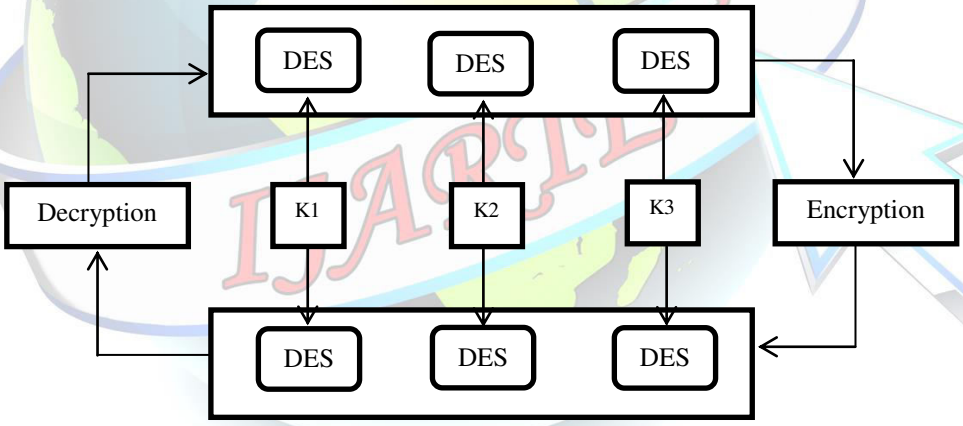


**Fig 3: Working of Triple DES**

3DES is the current standard adopted by the National Institute of Standards and Technology (NIST). Fig 3 shows the working of the Triple DES algorithm using three keys i.e., $K_1$, $K_2$ and $K_3$.

## IV. PROPOSED SYSTEM ARCHITECTURE

The proposed system has three primary entities i.e.,

1. The Data owner
2. The Data user
3. The Cloud

The *data owner*, the cloud and the user must authenticate themselves to have trust among them. The authenticated details are maintained in a database. The *Cloud* entity has three parts, i.e., the database, The Key Generation Center (KGC) and the Security Mediator (SEM). The KGC generates the key after getting the key

All Rights Reserved © 2018 IJARTET

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
(IJARTET) Vol. 5, Special Issue 14, April 2018

request from the owner. The *Data Owner* wants to upload the data to the cloud must request for the key from KGC. After getting the key, the owner encrypts the data and sends it to the cloud.

If Data user wants to access the secret information which is stored in the cloud, sends the request by sending its identity. Here the user gets the partially decrypted data from the cloud. The user fully decrypts the information to get the complete data.

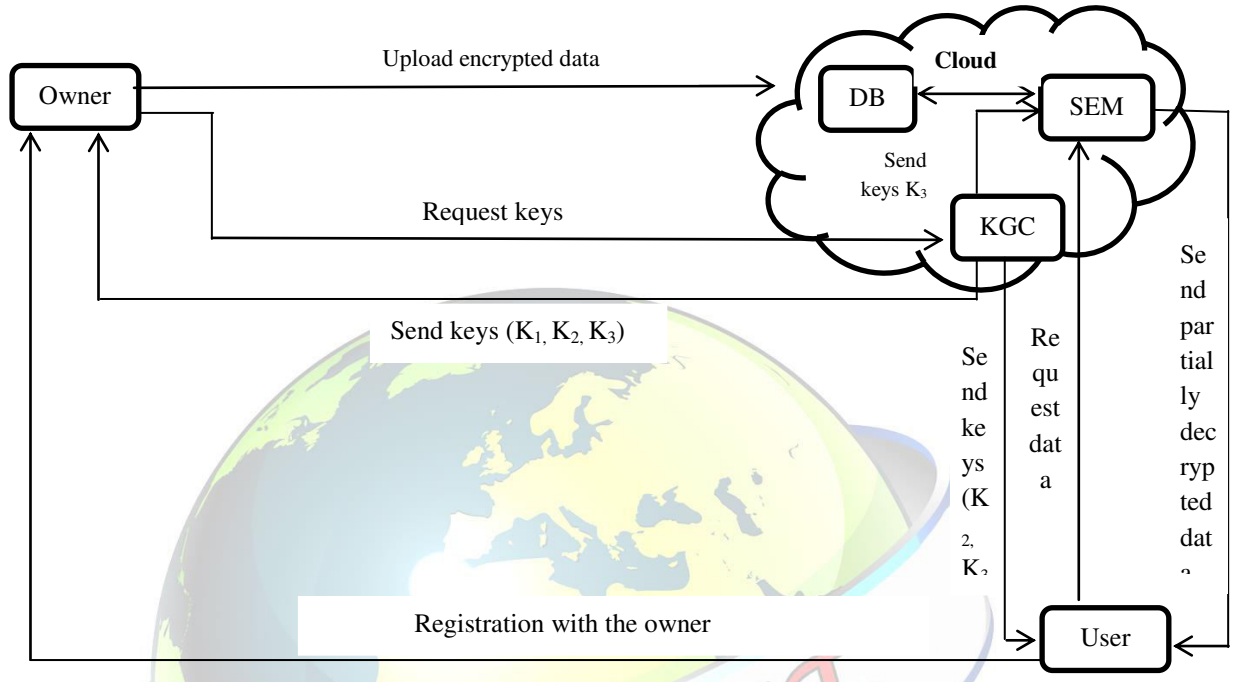


**Fig 4: Overall architecture of the CDSPD**

Fig 4 describes the overall architecture of the CDSPD and follows the below sequence:

- The user registers him/her to get the data of the user from the cloud.
- By identifying the user authentication, the owner request the cloud to send the keys to upload and store the data and further sends it to the user.
- The KGC generates the key and sends the key to all the three entities: user, owner, and SEM.
- Upon receiving the keys, the owner encrypts the data and sends to the cloud for uploading.
- The cloud partially decrypts the chunks of the uploaded data and sends to the user.
- The user completes the partially decrypted data.

## V. IMPLEMENTATION

The proposed scheme has four phases, i.e., Key Generation phase (KGC), Encryption phase, Partial decryption phase at the cloud end and Decryption phase at the user end. Each of these phase run using KGC(), Encrypt(), SEMDecrypt() and Decrypt() functions. The notations used in the algorithm are shown in Table I.

The Initial phase is creating the session so that the owner wants to store some data in the cloud to send it to the intended user. After receiving the request from the owner, the KGC generates the keys, i.e., the Shared Secret Session (3S) key for a particular session. The KGC() function is used to generate the keys. This generated key will be used for encryption and decryption.

**Table 1: Notations**

| Symbol | Definitions |
|---|---|
| *KGC* | Key Generation Center |
| *SEM* | Security Mediator |
| *M* | Plaintext |

All Rights Reserved © 2018 IJARTET

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
(IJARTET) Vol. 5, Special Issue 14, April 2018

| | |
|---|---|
| *C* | Ciphertext |
| *K* | Shared Secret Session (3S) key |
| *E* | Encrypt |
| *D* | Decrypt |
| *ID* | Identity of the user |
| *KEP* | Key Escrow Problem |

**Algorithm 1: Key Generation**

Step 1: Generate 192 bit key K using Random() K is assigned as Shared Secret Session (3S) Key
Step 2: Divide the key K into 3 subkeys i.e., $K_1$, $K_2$, $K_3$.
$K_1$64 bit, $K_2$64 bit
$K_3$64 bit

According to Algorithm 1, the KGC generates the session key K which includes three subkeys $K_1$, $K_2$ and $K_3$. The key length of each subkey will be 56 bits each. This generated key is shared to both the Data Owner and also to the User for the session.

After the key generation phase the next phase is for the Data Owner to send his data by encryption. This is done by Encrypt() function.

**Algorithm 2: Encryption**

Input: Imagebits, ID, File
Output: Encrypted data of the owner
Step 1: Convert image into imagebits by using ByteArrayOutputStream and encodeBase64()
Step 2: Encryption Phase: Encrypt();
$C = E_{K3} (D_{K2} (E_{K1} (M)))$

The Algorithm 2 provides the encryption phase. The Algorithm used for encryption and decryption is Triple DES (TDES) or 3DES or DES-EDE (Encrypt-Decrypt-Encrypt) [4]. TDES is an improvised version of DES algorithm as DES makes use of small key of 56 bit and very easy to crack. So in order to give more security, TDES is used which is three times bigger key compared to DES. TDES is the block cipher based symmetric encryption technique which means both sender and receiver uses a shared secret key to encrypt/decrypt the data. It takes fixed-length string of plain text bits and transforms it through a series of complicated operations into another cipher text bit string of same length.

While considering the image as input, the input image must be converted into image bits by using ByteArrayOutputStream and encodeBase64(). Then these image bits are encrypted using Encrypt() function. The plain text M is encrypted using the subkey $K_1$, decrypted using the subkey $K_2$ and again encrypted using the subkey $K_3$ and then uploaded to the cloud for further transfer it to the intended user. This encrypted data is the cipher text of the data.

**Algorithm 3: Decryption**

Input: C, ID, Encrypted File, Key K
Output: Original data and image
Step 1: SEM Decrypt();
$C_1 = D_{K3}$
Step 2: User Decrypt();
$M = D\ K3(E_{M1} (C_1))$

The cloud verifies that the requested files that is present or not. The partial decryption takes place in the cloud using the chunk key i.e., $K_3$ which partially decrypts the chunk of the encrypted data in SEMDecrypt Phase [3]. The next step is the downloading process at the user end shown in UserDecrypt Phase. In this phase, the user uses the secret session key that fully decrypts the partially decrypted data and gets the original data.

## VI. EXPERIMENTAL RESULTS AND ANALYSIS

All Rights Reserved © 2018 IJARTET

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
(IJARTET) Vol. 5, Special Issue 14, April 2018

In this section, we present the preliminary results of our approach CDSPD and is compared with the existing mCLPKE (mediated Certificateless Public Key Encryption) [14] by considering the encryption and decryption time. The size of the files used and the time taken for both encryption and decryption are shown in Table II.

**Table II: Comparison of Time Taken for Encryption and Decryption of mCL-PKE and CDSPD**

| Size in KB | Encryption Time in ms | | Decryption Time in ms | |
|---|---|---|---|---|
| | mCL-PKE | CDSPD | mCL-PKE | CDSPD |
| 100 | 34 | 16 | 21.8 | 8 |
| 200 | 64.2 | 30 | 43 | 15.3 |
| 300 | 121 | 56.7 | 84.6 | 27 |
| 400 | 236 | 110 | 168 | 52.6 |
| 500 | 469 | 218.5 | 334.7 | 104.8 |
| 600 | 930 | 435 | 669.3 | 210 |
| 700 | 1852 | 870 | 1340 | 412 |
| 800 | 3700 | 1740.4 | 2678.7 | 815 |
| 900 | 7417 | 3481 | 5360 | 1632 |
| 1000 | 14830 | 6956 | 10720 | 3260 |

Fig 5 shows the time required to perform the encryption operation of CDSPD scheme, and mCL-PKE scheme for different message sizes vary from 100 kb to 1000 kb. Since our scheme does not use pairing operations, it performs encryption efficiently by using symmetric encryption approach.

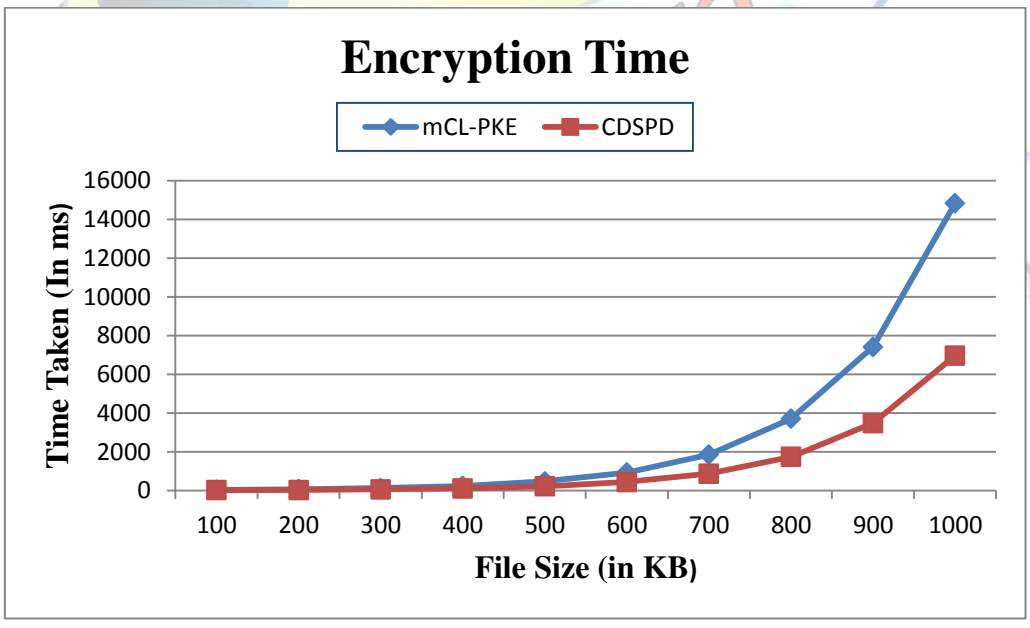


**Fig 5: Comparison of time taken for encryption by mCL-PKE and CDSPD**

Fig. 6 shows the time taken to perform decryption of CDSPD and mCL-PKE for different file sizes that varies from 100 kb to 1000kb. One part of the decryption is done in the SEM of the cloud, and then the cloud will send the partially decrypted data to the user for further decryption process. The user completes the decryption using Shared Secret Session (3S) key and gets the original data. As only chunks are sent to the user and the cloud, the attackers will not be having any idea of these chunk keys. This improves the security strength of the data. The division of the decryption process reduces the overhead of the user for complete decryption and

All Rights Reserved © 2018 IJARTET

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
(IJARTET) Vol. 5, Special Issue 14, April 2018

also make sure that intruders do not attack the data. Using of Triple-DES algorithm decreases the time consumption in both encryption and decryption of the data and also secures the data.

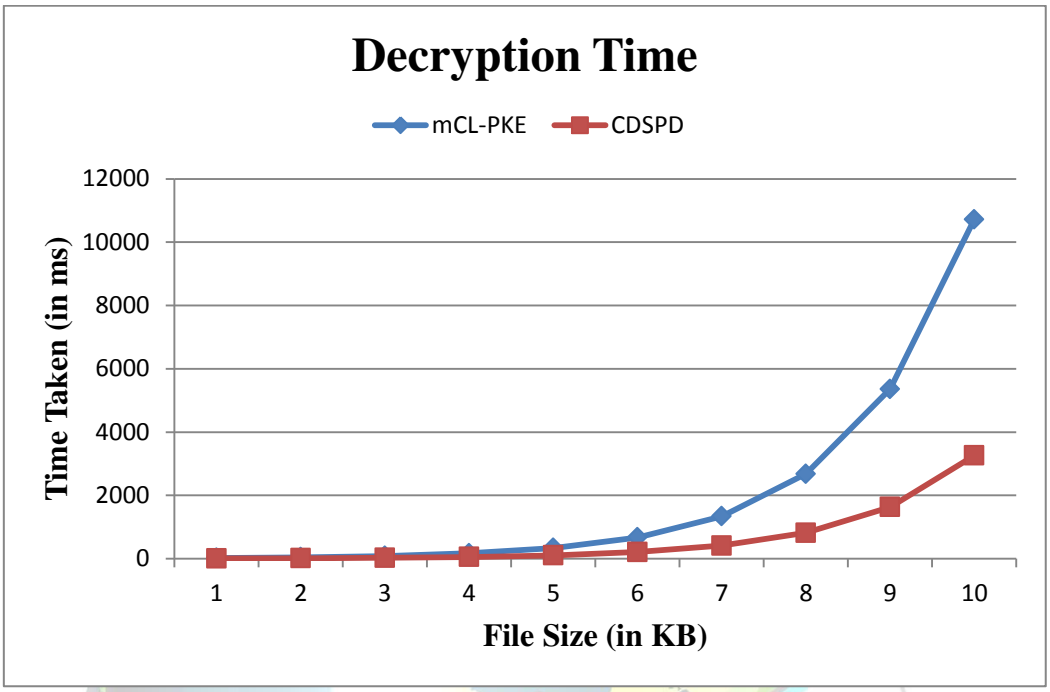


**Fig 6: Comparison of time taken for decryption by mCL-PKE and CDSPD**

It is observed from the Fig 5 and 6 that initially when the file size is small, the time taken by both the algorithm is same.
As the file size increases, the difference in encryption time and decryption time increases. Thus our approach takes less time to encrypt and decrypt.

## VII. CONCLUSIONS

In this paper, "Certificateless Data Sharing in Cloud through Partial Decryption" (CDSPD) scheme has been proposed that solves the key escrow problem and avoids usage of asymmetric keys. This scheme is more flexible and consumes less time for encryption and decryption compared to the existing mediated Certificateless Public Key Encryption (mCL-PKE). CDSPD reduces the time by almost 50 percent the time taken by mCLPKE. It provides more confidentiality because of the partial decryption that provides double security to the data that is shared to the user.

## REFERENCES

[1] D. Boneh, X. Ding and G. Tsudik"Fine-grained control of securitycapabilities", *ACM Transactions in Internet Technologies*, vol. 4, no. 1, pp. 60-82, Feb 2004.
[2] Adi Shamir, "Identity-based Crytosystems and Signature Schemes*", Lecture Notes in Computer Science*, vol. 196, pp. 47-53, November 2000.
[3] Sattam S Al-Riyami and K. Paterson, "Certificateless Public Key Cryptography", *International Conference on the Theory and Application of Cryptology and Information Security*, vol. 2894, pp. 452-473, ASIACRYPT 2003.
[4] Jonathan Katz, Amit Sahai and Brent Waters, "Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products", *International Conference on Advances in Cryptology (EUROCRYPT)*, pp. 146–162, 2008.
[5] Jan Cameniseh, Maria Dubovitskaya, Gregory Neven, "Oblivious Transfer with Access Control", *ACM Conference on Computer and Communication Security*, pp. 131-140, Nov 2009.
[6] M. Bellare, A. Desai, D. Pointcheval, P. Rogaway, "Relations Among Notions of Security for Public-key Encryption Schemes", *Advances in Cryptology,*CRYPTO '98', vol. 1462, pp. 26-45, May 2008.
[7] Gerome Miklau and Dan Sucice, "Controlling Access to Published Data using Cryptography", *Proceedings of 29th International Conference on Very Large Data Bases*, vol. 29, pp. 898-909, Berlin, Germany, December 2003.
[8] A. Sahai and B. Waters, "Fuzzy Identity- based Encryption", *International Conference on Advances in Cryptology (EUROCRYPT)*, vol. 457–473, pp. 457-473, 2005.

All Rights Reserved © 2018 IJARTET

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
(IJARTET) Vol. 5, Special Issue 14, April 2018

[9] N. Shang, M. Nabeel, F. Paci and E. Bertino, "A Privacy Preserving approach to Policy-based Content Dissemination", *International Conference on Data Engineering*, pp. 944-955, 2010.
[10] M. Nabeel, N. Shang and E. Bertino, "Privacy Preserving Policy based Content Sharing in Public Clouds",*IEEE transaction on Knowledge and Data Engineering,* vol. 25, no. 11, pp. 898-909, Berlin, Germany,2003.
[11] J. Bethencourt, A. Sahai and B. Waters, "Ciphertext-Policy Attributebased Encryption", *IEEE Symposium on Security and Privacy*, pp. 321- 334, Taormina, Italy, May 2007.
[12] Shucheng Yu, Cong Wang, Kui Ren and Wenjing Lou, "Attribute based Data Sharing with Attribute Revocation*", Proceedings of the 5th International Symposium on Information, Computer and Communication Security, ASIACCS*, pp. 261-270, January 2010.
[13] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, "Relations among Notions of Security for Public-key Encryption Schemes*", International Conference on Advances in Cryptology*, pp 26-45, 2006.
[14] Seung-Hyun Seo, Mohamed Nabeel, Xiaoyu Ding and Elisa Bertino, "An Efficient Certificateless Encryption for Secure Data Sharing in Public Clouds", *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2107-2119, September 2014.
[15] Vipul Gopal, Omkant Pandey, Amit Sahai and Brent Waters, "Attributebased Encryption for Fine-Grained Access Control of Encrypted Data", *Proceeding of the ACM Conference on Computer and Communication Security*, pp. 89-98, January 2006.
[16] Yinxia Sun and Futai Zhang, "Secure Certificateless Public Key Encryption without Redundancy*", International Association of Cryptologic Research ePrint Archive*, 2008.
[17] Y. Sun, F. Zhang and J. Baek, "Strongly Secure Certificateless Public Key Encryption without Pairing", *6th International Conference on Cryptography and Network Security*, pp. 194-208, Singapore 2007.
[18] S. S. M. Chow, C. Boyd and J. M.G. Nieto, "Security Mediated Certificateless Cryptography", *International Conference on Theory*, vol. 3958, pp. 508-524, New York, USA, 2006.
[19] C. Yang, F. Wang and X. Wang, "Efficient Mediated Certificates Public Key Encryption Scheme without Pairings*", International Conference on Advanced Information Networking and Applications*, vol. 1, pp. 109-112, May 2007.
[20] X. W Lei Xu and X. Zhang, "CL-PKE: A Certificateless Proxy Re-Encryption Scheme for Secure Data Sharing with Public Cloud", *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pp. 87–88 May 2012.
[21] Jeevitha B K, Thriveni J and Venugopal K R, "Data Storage Security and Privacy in Cloud Computing: A Comprehensive Survey*", InternationalJournal of Computer Applications*,vol. 156, no. 12, pp. 16-27, December 2016.
[22] Boyang Wang, Sherman S. M. Chow, Ming Li and Hui Li, "Storing Shared Data on the Cloud via Security-Mediator", *International Conference on Distributed Computing Systems*, pp. 124-133, July 2013.

All Rights Reserved © 2018 IJARTET 