International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958 (Online), Volume-9 Issue-4, April, 2020

S^{3}DCE: Secure Storing and Sharing of Data in Cloud Environment using user Phrase

Jeevitha B K, Neetha P U, Pushpa C N, Thriveni J, Venugopal K R

Abstract*: Distributed Cloud Environment (DCE) focuses mainly on securing the data and safely shares it to the user. Data leakage may occur by the channel compromising or with the key managers. It is necessary to safeguard the communication channel between the entities before sharing the data. In this process of sharing, what if the key managers compromises with intruders and reveal the information of the user's key that is used for encryption. The process of securing the key by using the user's phrase is the key concept used in the proposed system "Secure Storing and Sharing of Data in Cloud Environment using User Phrase (S^{3}DCE). It does not rely on any key managers to generate the key instead; the user himself generates the key. In order to provide double security, the public key derived from the user's phrase also encrypts the encryption key. S^{3}DCE guarantees privacy, confidentiality and integrity of the user data while storing and sharing. The proposed method S^{3}DCE is more efficient in terms of time, cost and resource utilization compared to the existing algorithm DaSCE (Data Security for Cloud Environment with Semi Trusted Third Party) [22] and DACESM (Data Security for Cloud Environment with Scheduled Key Managers) [23].

***Keywords*: FADE, Key Derivation Function, OTK Algorithm, Owner Data, SFADE.**

I. INTRODUCTION

Cloud computing means offering the system resources especially for data sharing and for computing to the users without any direct active management by them. The data transformation is carried over the internet. The users need not worry about the data storage in the system. Each entities like software, hardware and platform are considered as a utility/service by the cloud. These services attracts the organization and customers to avail the benefits and flexibilities of the cloud. Cloud helps the organizations in achieving the ability to store and retrieve their data through cloud data centres. The cloud service provider offers various services to the user in which storage is one of the service provided by the cloud. Because of the flexibility and cost effective nature of the cloud, the IT infrastructures are gaining benefit from the cloud services. By storing, the data on cloud helps the organization/ users to concentrate on their company goals rather than the storage system and its maintenance. To achieve this, the cloud has to adopt the best security measures. Storage is one of the service provided by the cloud along with other resources namely Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) [1]. SaaS is a model that gives quick access to cloud based web applications. IaaS provides the entire range of computing infrastructure such as storage, servers networking hardware along with maintenance and support. PaaS is the base of the cloud where development, testing and organization of the business application takes place. It simplifies the process of enterprise software development.

Data centres [2] plays an important role in storing the user's data and access them through the internet. This greatly reduces the users cost on spending on the other storage devices like flash-drives, pen-drives and hard disks. The cloud service providers works on pay-as-you-use model where the providers offers the storage space so that the users can scale up/down the storage space based on their requirement. These data centres does not belong to the users, instead they are owned by the cloud service provider.

The cloud storage service providers are Google, Amazon, and Dropbox etc. The user hire some space in data center where data can be stored and accessed through internet. Since data centres are away from user's physical connection, the data can be accessed in any geographical region where internet is available.

The cloud service providers make multiple copies of these data and stores in the multiple data centres, which are geographically apart. If one data centre goes down, then the user can get their data from other data centre from where the copy is stored. This is an important benefit offered by the cloud known as "Disaster Recovery". It is also cost benefit because the user need not to concentrate on the storage space management. So, Storage-as-a-Service is widely used among the cloud users.

Even though the cloud acquires a good amount of benefits, it is still considered as third party. Therefore, it is better to protect the data before uploading it into the cloud. To protect the data, the approach used until now are the traditional encryption techniques that encrypts the raw data into an unreadable format at the user end and stores safely at the cloud end.

Revised Manuscript Received on April 25, 2020.
* Correspondence Author
Jeevitha B K*, Computer Science and Engineering, University Viseveswaraya College of Engineering, Bangalore University, Bengaluru, Karnataka, India Email: bkjeevitha87@gmail.com
Neeta P U, Computer Science and Engineering, University Viseveswaraya College of Engineering, Bangalore University, Bengaluru, Karnataka, India Email: neethapud@gmail.com
Pushpa C N, Computer Science and Engineering, University Viseveswaraya College of Engineering, Bangalore University, Bengaluru, Karnataka, India Email: pushpacn@gmail.com
Thriveni J, Computer Science and Engineering, University Viseveswaraya College of Engineering, Bangalore University, Bengaluru, Karnataka, India Email: drthrivenij@gmail.com
Venugopal K R, Vice-Chancellor, Bangalore University, Bengaluru, Karnataka, India Email: venugopalkr@gmail.com

So even the attacker attacks the data center, it is impossible to read the data unless he/she gets information used to encrypt the data.

What if the entity that is responsible to generate/store the user's key compromise with the intruder and reveal the information about the key? Therefore, it is very important to secure the key as well as the data. The proposed method S³DCE provides the privacy, confidentiality and integrity to the user's data by encrypting the key that is used to secure the data.

A. Motivation

In the previous method, the third party key managers are responsible to generate the keys and scheduler is used to monitor the workload of the key managers and tried to assign the work who has less task to complete. By considering the scheduler as single point of failure, the concept of key managers is removed from the proposed method. Shamir's concept is eliminated from the proposed method by considering it as time consuming.

B. Contributions

In this paper, a secure storing and sharing of data in cloud environment using user's phrase S³DCE is proposed that use both symmetric and asymmetric keys to provide data confidentiality. Symmetric key is generated using the proposed OTL algorithm and asymmetric key is generated using the proposed key derivation algorithm. The proposed algorithm is efficient in terms of time, cost and resource utilization compared to the existing system.

C. Organizations

The organization of the paper is as follows: Section II discusses the Literature Survey that are referred to the paper. Section III describes the Background Work. Section IV tells about the Proposed method S³DCE. Section VI describes the Algorithms used. Section V gives the Performance Analysis obtained. Section VI concludes the paper.

II. LITERATURE SURVEY

The main purpose of cryptography [3] to secure the information while storing and securing the information. Elliptic Curve Cryptography (ECC) [4] plays an important role in the world of cryptography as public-key crypto-system. The shorter key size in ECC provides an equivalent protection level of public key algorithms, which utilized the largest key size of Rivest Shamir Adleman (RSA) [5]. In addition, the ECC offers more security compared to the RSA algorithm based on the prime number factorization problem.

Michael et al., [6] made IT companies to concentrate on the work rather than maintenance of the data. The developers can easily deploy their project with the help of cloud service provider instead of thinking about their management cost. The companies with large batch-oriented task can get result quickly because of using 1000 servers for one hour instead of using one server for 1000 hours. This reduces cost for the large company. The authors gave a clarification about cloud computing by comparing it with conventional computing and provides top technical and non-technical obstacles and opportunities of cloud computing.

In 2010, the US Government started to process more about the cloud there was no good definition for that. So National Institute of Standards and Technology (NIST) [7] in US to the task of trying to put a description together and came up with a simple model. They described the cloud with the three major things. First is the set of central characteristics of actually what makes the cloud as " The Cloud". Second is the set of service description that gives a consistent way to describe the kinds of services that the cloud is trying to offer and the third is the deployment models that tells how somebody is getting the access to deployed cloud environment.

Marjory [8] draws the multiple views of cloud such as technical view, business view and policy view. The author enumerates key issues, including risks associated with the cloud service provider as well as by the users. As the users most commonly use public cloud, more security is required for the public cloud. The author tells about the absence of security mechanism, it is important to be careful about data or processes assigned to the public cloud.

H. Takabi et al., [9] discusses the various approaches to boost the secrecy challenges of the cloud. The author presents the main aspects of cloud that shows the transparent nature of the cloud. The key aspects includes on-request self-service, secure network access and the estimated resources. The control of computing resources are optimized through traffic balancing, and resource allocation. Different privacy and security issues are defined along with the solutions including the trustworthy cloud.

Juels et al., [10] makes use of Iris File System (IFS) as a solution to the security of the data. IFS provides the services to freshness, similarity and availability of the data. The author also uses Merkle tree that checks for the originality and similarity of the data. In Merkle tree, the data is divided into file blocks, uses version numbers of the file, and places these files blocks at different stages of the tree. So the author proposed a portal based application that manages the keys and assures the data newness, recollection and data resistant against failure of disk.

H Lin et al., [11] proposed a distributed network in a de-centralized way so that each messages are distributed independently to storage servers. Each servers can execute the encoding process so that each key server can object the servers independently. This distributed network assures the privacy of the message even if all storage servers are compromised.

A. R Khan et al., [12] presents a survey on mobile cloud computing. Smart-phones are now capable of supporting a wide range of applications, many of which demand an ever-increasing computational power. This postures a challenge since smart-phones are resource-constrained gadgets with constrained computation control, memory, and capacity. The cloud computing innovation offers for all intents and purpose boundless energetic assets for computation, capacity and benefit arrangement.

Hence, analyst imagine amplifying cloud-computing administrations to portable gadgets to overcome the smart-phones imperatives.

This article presents versatile cloud engineering, offloading choice influencing substances, application models classification, the most recent portable cloud application models, their critical analysis and future research directions.

SeDaSc [13] is a cloud storage security system that concentrates on data sharing within the group. It provides data confidentiality, integrity and access control. It shares the data without re-encryption. It provides assured access control by deleting the parameters, which is necessary for the decryption process. The SeDaSc methodology can be used in both conventional and mobile cloud computing environment.

In order to provide the random access of the file along with integrity and confidentiality, the author Yan et al., [14] constructed MAC tree that uses the universal hash based stateful MAC. It provides better performance and low cost integrity protection than merkle hash tree based scheme for distributed cryptographic file systems.

Hsiao et al., [15] proposed a threshold proxy re-encryption scheme that adds the additional functionalities in securing the data over encrypted data. It is integrated with decentralized erasure code that lets the user to forward their data to other user without retrieving the data back.

M. Ali et al., [16] reduces the security issues for the data that is shared among the group of users. This type of sharing faces the problem both at the cloud end and the threat from insiders. So the author uses the concept of key shares in which it uses two different shares: one share will be with the user and the other share with the cryptographic server. This methodology reduces the threats from insiders. By this, the author achieves data confidentiality, access control and assured deletion of files.

Y. Tang, P et. al., [17] proposed a framework that guarantees the deletion of the file and consistently works on cloud administrations. The framework decouples the organization of mixed data and encryption keys, such that mixed data remains on third party (untrusted) cloud service providers. It provides assured deletion of the file based on time bound and the most fine-grained approach called policy-based file guaranteed deletion. This policy is based on time or read/write access given to the client's data. Therefore, if the policies are denied or end up with time expiry, then the file cannot be accessed and deleted completely.

III. BACKGROUND WORK

A. FADE

File Assured DElection (FADE) addresses the issue that even after the files are deleted by the user, some traces are still found as multiple copies of the data stored to achieve reliability. FADE makes use of third party key manager to store the keys that is used to encrypt the user file.

Fig 1 shows the FADE architecture. In FADE, there are two variants of meta-data: file and policy meta-data. The file meta-data consists of file size part of 8 bytes and Hash based Message Authentication Code (HMAC) of 20 bytes for integrity checking. These two are attached to the encrypted file at the initial stage. Policy meta-data is an assumption made that each individual policy and this policy is of 4 byte of unique integer identities. This provides the value-added security features associated with outsourced applications.

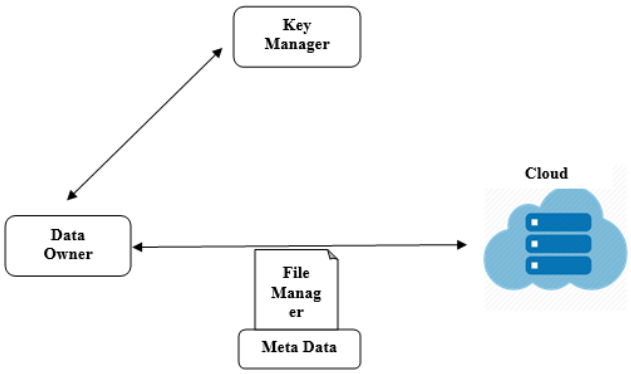

Fig 1. FADE Architecture

B. SFADE+

By the facilities provided by the cloud, the dependency on the pubic cloud storage has increased immensely; hence, the security is important even after the data is deleted. The basic idea of the FADE [18] is to encrypt the file from the ephemeral keys where these keys are maintained by the "Ephemeras", i.e., a key manager. It describes a system that supports high availability of data, until the data is completely deleted. It supports two types of assured deletions.

1) Deletion by knowing the expiry time at the time of file creation.
2) Deletion by on-demand of the individual files.

It assures the previous work that describe how to do assured deletion when the expiry time of a file is known at file creation time and was particularly suited to supporting many independent clients. This system reviews the work and describes how to support an on-demand deletion of individual files.

FADE is introduced that uses semi-trusted third party [19] for key generation, which makes the system bit complex and makes it suitable for corporate clients. However, it will be difficult for the ordinary users. Hence, the SFADE [20] model is easier to implement and friendlier to the users. This technique avoids key-manager while ensuring assured deletion that benefits the cloud storage users to maintain their data confidentiality. Both FADE and SFADE lacks of sharing feature.

SFADE+ is proposed for sharing the file with more than one users, so the SFADE is enhanced to SFADE+ that includes:

1) Implementing version-control system.
2) Sharing files with other users.
3) Mobile version of the system.

SFADE+ [21] is a system that retain data integrity and access. It maintains data security through encryption techniques and helps to access these encrypted data in a secure way. SFADE+ provides data sharing as well as accessibility and deletion of data. However, SFADE+ leads to revocation problem and the algorithms used for key generation is not much secure and key length is also very large.

IV. S³DCE: SECURE STORING AND SHARING OF DATA IN CLOUD ENVIRONMENT USING USER PHRASE

The proposed method S³DCE has 3 modules i.e., the owner, the user and the cloud as shown in Figure 2. The owner is the one who either store his own data, or share it to the user. The owner has to generate his own secret key using OTK algorithm to encrypt the file that is to be stored in the cloud. After file encryption, the key used should also be secured from the intruders so that the owner generated the phrase, which he can be easily remembered and not known to anyone. This phrase is sent to Key Derivation Function (KDF) to generate the public key and then this public key is used to encrypt the secret key. Both the encrypted file and the encrypted key is stored in the cloud.

The owner can share the file with any of the registered user. The user has to register himself in the cloud. By this, a profile will be created along with public key, which is generated using his phrase as an input to the KDF. When the owner wants to share the file to a user *x*, encrypt the file using public key of *x* and store the link of the location where the encrypted file and encrypted key in the repository. The cloud is the one who stores the data that is sent by the owner. The cloud has one more entity known as repository where the user's profiles will be stored.

The interactions are considered in three scenario between the user, the cloud and the owner:

1) Single Interaction.
2) One-to-One Interaction.
3) One-to-Many Interaction.

A. Single Interaction

When the user acts both the roles of owner and the user, where the user wants to upload his data to the cloud and only he/she has to access the uploaded data. While uploading the file to the cloud, the owner has to remember the phrase that is used to generate the key. This phrase should be known only to the owner in order to maintain the privacy of the data. The process of uploading and downloading of the personal data of the owner is as follows:

Uploading

1) The owner has to select the file F.
2) The secret encryption key S_k is generated using OTK algorithm.
3) The file F is encrypted using encryption algorithm and S_k i.e., $E(S_k(F))$.
4) The user phase Ph is used to generate the public key P_k with the help of KDF algorithm.
5) Public key P_k and encryption algorithm is used to encrypt the secret encryption key i.e., $E(P_k(S_k))$.
6) Both encrypted file $E(S_k(F))$ and encrypted secret key $E(S_k)$ is stored in the cloud.

Downloading

1) Both encrypted file $E(S_k(F))$ and encrypted secret key $E(S_k)$ is fetched from the cloud.
2) The phrase Ph is sent to the KDF algorithm to generate the private key P_r.
3) The private key P_r and the encrypted secret key is sent to decrypted to get the encryption key S_k.
4) The encrypted secret key S_k and the encrypted file $E(S_k(F))$ is sent to decryption algorithm to get file F.
5) By using the private key P_r, the user first decrypts the secret key $D(P_r,E(Sk))$.
6) Then the user makes use of secret key S_k to decrypt the encrypted file $D(S_k(F))$.
7) The user can download the file for further use or can only view the file.

B. One-to-One Interaction

The interaction is between only one owner and one user at a time and vice versa. If owner has to send some file to the user, has to register him/herself and a profile will be created and only the public key of them will be stored in the repository. This public key is generated by the phrase of the user using the KDF algorithm.

The process of uploading and downloading of the data between a user and an owner is as follows:

Uploading

1) The owner selects the file F to be shared to the user.
2) The secret encryption key S_k is generated by using OTK algorithm.
3) The file is encrypted by using the symmetric key S_k to get encrypted file i.e., $E(S_k(F))$.
4) The public key P_k of the user is obtained from the repository that is stored in the cloud.
5) Then secret key is encrypted by the public key P_k of the user i.e., $E(P_k(S_k))$.
6) Both the encrypted file $E(S_k(F))$ and encrypted secret key $E(S_k)$ are stored in the cloud.
7) The location of the stored data is updated in the user's profile, which is in repository of the cloud.

Downloading

1) The user uses the link of the file location that is stored in the cloud.
2) The user downloads both encrypted file $E(S_k(F))$ and encrypted secret key $E(S_k)$.
3) The user uses the phrase to generate the private key P_r by using KDF algorithm.
4) By using the private key P_r, the user first decrypts the secret key $D(Pr,E(S_k))$.
5) Then the user makes use of secret key S_k to decrypt the encrypted file $D(Sk(F))$.
6) The user can download the file for further use or can only view the file.

Retrieval Number: D7303049420/2020©BEIESP
DOI: 10.35940/ijeat.D7303.049420
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

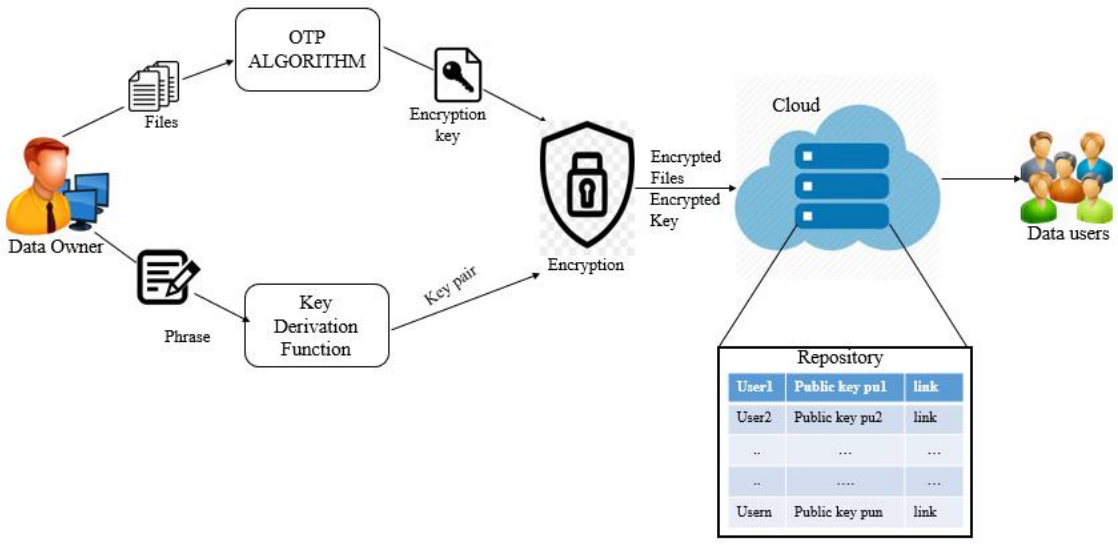

Fig 2: S^{3}DCE Architecture

C. One-to-Many Interaction

The interaction is between one owner and a group of users. If the owner has to send the file to a group, instead of sending it to single member, a group is maintained based on their category of working, status. The one who created the group will frame the phrase, get the key by passing the phrase to KDF, and get the common public key and stored it in the repository.The process of uploading and downloading the file between the owner and the group is as follows:

Uploading

1) The owner has to select the file F that need to be shared among groups.
2) The secret key S_k is generated using OTK algorithm.
3) The file is encrypted using $E(S_k(F))$.
4) Group public key is retrieved from repository.
5) This key is used to encrypt the secret key and will obtain $E(P_u(S_k))$.
6) The encrypted secret key $E(P_u(S_k))$ and encrypted file $E(Sk(F))$ both are stored in the cloud.
7) The location of the data stored will be updated in the repository.

Downloading

1) The member of the group retrieves both encrypted file and the encrypted secret key from the cloud.
2) Request is sent to generate group private key P_r.
3) Private key generated using the KDF algorithm and the phrase of the group admin.
4) Secret key is obtained by decryption using P_r and decryption algorithm.
5) This secret key is used to decrypt the file and the file is obtained that is shared by the owner.
6) The group members can either view or download the file F.

V. ALGORITHMS USED

A. OTK Algorithm: The key of length 4 to 6 bytes are framed that contains the combination of 0-9, a-z, A-Z and special characters. Based on the user requirement, the length of the key is considered. Let the length of the key be 4 bytes. Algorithm 1 tells about the secret key generation algorithm.

Algorithm 1: One-Time Key Algorithm

Input: Number of digits required K_i
Output: Encryption key of specified length
begin
Step1: Initialize all the characters to $W = W_0,W_1,..,W_n$, i.e., a-z, A-Z,0-9 and Special characters.
Step 2: Select a random index i
Step 3: Randomly select the number of digits required from W.
Step 4: Concatenate the digits.
Step 5: Shuffle the digits.
Step 6: The Encryption key Ki with specified number of digits is generated.
end

B. Key Derivation Function: The user phrase cannot directly be used to generate the keys. So the user phrase is let into Key Derivation Function to first convert to a string of characters because the keys are not stored anywhere and the user has to generate the keys every-time either to encrypt or decrypt the file. It is necessary to derive a common string of characters with the user phrase. This string is given as an input to ECC key pair algorithm to generate a pair of keys i.e., public and private key whenever required. Algorithm 2 tell about the key derivation function.

Algorithm 2: Key-Derivation Function

Input: User Phrase.
Output: Encoded String/key Pair.
begin
Step 1: Split the user phrase letter by letter.
Step 2: Convert each letter to its binary value (ASCII Value).
Step 3: Concatenate all the binary values together.
Step 4: Divide the binary values into group of 6 bits each.
Step 5: Convert this 6 bits to its equivalent ASCII character.
Step 6: The encoded string is given as input to ECC key Pair Algorithm.
Step 7: A pair of private and public is generated.
end

VI. PERFORMANCE ANALYSIS

It is assumed that the three entities i.e., the user, the owner and the cloud have authenticated themselves in a secure way. As the cloud is considered as the semi-trusted third party, the process of encrypting the file as well as the secret key is done at the client side. The performance is analysed by considering the 2nd scenario where the interaction takes place between single user and single owner. Scenario 3 is not considered for the analysis as the group owner is only responsible to generate keys using group owner phrase and to communicate within the group that intern considered as a single user. Owner will be communicating with the group owner that makes the interaction as one-to- one communication.

The performance of the proposed method is compared with the previous existing system DSCESM and DaSCE where both the scheme makes use of semi-trusted third party to generate and manage keys. The proposed system S^{3}DCE does not rely on any key managers for the keys, the owner himself generates the secret key to encrypt the file to store and share it with the group or any individual user.

The existing system DaSCE uses number of key managers that the owner has to wait to get back the key from all the key managers to encrypt the secret key and depends on the Shamir's secret sharing scheme to divide the secret key based on the number of keys obtained by the key managers. To overcome the waiting time for all the key managers to send the keys, a scheduler has been used in DSCESM. Scheduler is used to manage the task of generating the keys based on the workload allotted previously.

Table- I: Key Generation Time

DaSCE and DSCESM (with Key Managers)	27-83 msec
S^{3}DCE (without key managers)	9-11 sec

By considering the scheduler as a single point of failure, the concept of scheduler and key managers are removed in proposed system S^{3}DCE. Instead, the owner himself generates the keys for both encrypting the file and encrypting the secret key. So that the waiting time is removed

The key generation time in the existing system considers either the entire key manager or any number of key managers. The existing system DaSCE uses all the key managers and DSCESM uses only those key managers who have less workload with the help of scheduler. Irrespective of file size, the time taken to generate keys is on an average between 27 to 83 msec. However, the proposed system S^{3}DCE does not depend on any key manager to generate key. Table I shows the time taken generate key with and without key managers by the three methods i.e., DaSCE, DACESM and S^{3}DCE.

The performance is analysed in terms of time required to generate keys, total time to upload and download the encrypted file and encrypted secret key. The file size varies from 10kb to 1000kb that is used to analyse the performance.

The total time taken to upload the file to the cloud includes symmetric key generation by the owner, encrypting the file, getting the public key that is stored in the users profile in the cloud's repository and used to encrypt the symmetric key and then store the encrypted file and the encrypted key in the cloud and update the link of the data stored location in the user's profile.

As the existing system depends on the third party [key managers] to generate the key for encrypting the secret key in the DaSCE and the scheduler to allocate the task of generating the keys in the existing system DSCESM. There is a waiting time that is not there in the proposed system S^{3}DCE, thus the upload time in the proposed system is less as compared to existing system.

Table II: Comparison of Total Upload Time of DASCE, DACESM and S^{3}DCE

Size in KB	Upload Time in ms		
	DaSCE	DSCESM	S^{3}DCE
10	480	135	34
20	484	138	35
50	489	142	42
100	490	146	55
200	495	149	63
500	496	152	77
1000	498	155	89

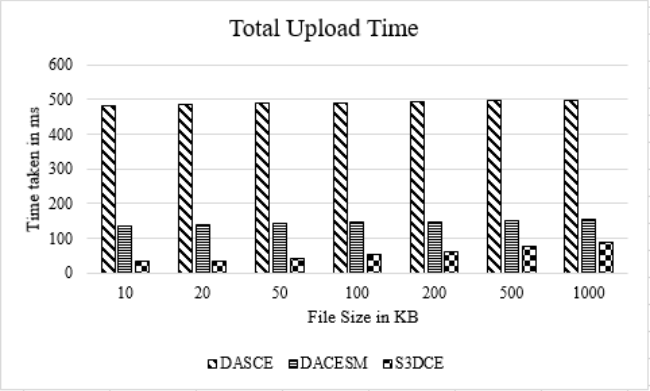

Fig 3: Comparison of Total Upload Time of DASCE, DACESM, S^{3}DCE

Fig 3 shows the time taken to upload the file. The graph is plotted by considering the values given in the Table II with the file size. As the file size increases, the time taken to upload the file also increases. By eliminating the key manager and shamir's concept, the proposed method S^{3}DCE takes much less time compared to the existing system DaSCE and DSCESM. Thus, the cost incurred in storing is reduced. The uploading time reduces by 88% in S^{3}DCE compared to DaSCE and 61% in S^{3}DCE compared to DSCESM.

Table III: Comparison of Total Download Time of DASCE, DACESM and S^{3}DCE

Size in KB	Download Time in ms		
	DaSCE	DSCESM	S^{3}DCE
10	432	118	30
20	436	120	32
50	436	120	37
100	437	121	43
200	437	121	49
500	439	123	56
1000	439	118	75

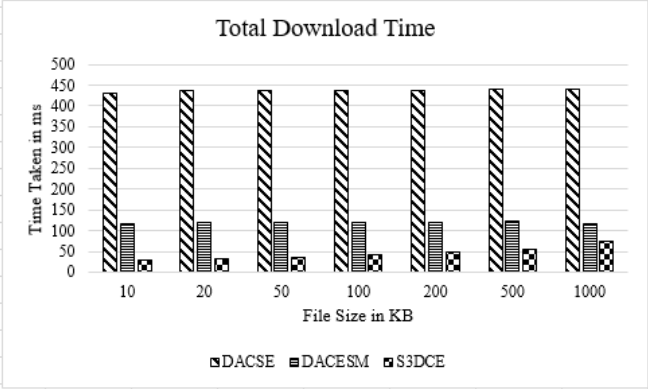

Fig 4: Comparison of Total Download Time of DASCE, DACESM, S^{3}DCE

The total time taken to download the file from the cloud includes retrieving both encrypted file and the encrypted key from the link, generating the private key, decrypting the encrypted secret key, decrypting the file using the secret key. Fig 4 shows the time taken to download the file. The graph is plotted by considering the values given in the Table III with the file size. As the file size increases, the time taken to download the file also increases. By eliminating the key manager and shamir's concept, the proposed method S^{3}DCE takes much less time compared to the existing system DaSCE and DSCESM. The download time reduces by 89% in S^{3}DCE compared to DaSCE and 61% in S^{3}DCE compared to DSCESM.

VII. CONCLUSIONS

S3DCE is a security method that eliminates the concept of key manager and the shamir's concept. The OTK key generation algorithm is used to generate encryption key and it is more efficient compared to the random key generator. The ECC key pair generator provide the same key strength with smaller key size and more secure compared to RSA. This feature of ECC is very appealing with limited storage, processing power and reduced computational requirements. The performance is analysed in terms of time with respect to key generation, total upload time and total download time. It is observed that the proposed method is more efficient as compared to the existing system DACSE and DACESM.

REFERENCES

1. Jeevitha B K, Thriveni J and Venugopal K R, "Data Storage Security and Privacy in Cloud Computing: A Comprehensive Survey", International Journal of Computer Applications, vol. 156, no.12, pp. 16-27, December 2016.
2. Claudio Fiandrino, Dzmitry Kliazovich, Pascal Bouvry and Albert Y. Zomaya, "Performance Metrics for Data Center Communication Systems", IEEE 8th International Conference on Cloud Computing, pp. 98-105, 2015.
3. Kawther Esaa Abdullah and Nada Hussein M. Ali, "Security Improvement in Elliptic Curve Cryptography", International Journal of Advanced Computer Science and Applications, vol. 9, no. 5, pp. 122-131, 2018.
4. Shantha A, Renita J, and Edna Elizabeth N, "Analysis and Implementation of ECC Algorithm in Lightweight Device", International Conference on Communication and Signal Processing, pp. 305-309, 2019.
5. Jagadish Thiruvayipati, "Elliptic Curve Cryptography: faster and lighter encryption protocol for cloud computing environment", International Journal of Engineering Development and Research, vol. 5, issue. 4, pp. 148- 156, 2017.
6. Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia, "A View of Cloud Computing", Communications of the ACM, vol. 53, issue. 4, pp. 50-58, April 2010.
7. M. Hogan, F. Liu, A.Sokol, and J. Tong, "NIST cloud computing standards roadmap", NIST Special Publication, July 2011.
8. Blumenthal Marjory S, "Is Security Lost in the Clouds", International Conference on Communications and Strategies, no. 81, pp. 69-86, 2011.
9. Hassan Takabi, James B.D. Joshi, and Gail-Joon Ahn, "Security and Privacy Challenges in Cloud Computing Environments", IEEE Transaction on Security and Privacy, vol. 8, issue. 6, Nov-Dec 2010.
10. Ari Juels and Alina Opera, "New Approaches to Security and Availability for Cloud Data", Communications of the ACM, vol. 56, no. 2, pp.64-73, 2013.
11. H. Ling and W. Tzeng, "A Secure Decentralized Erasure Code for Distributed etwork Storage", IEEE Transaction on parallel and Distributed Systems, vol. 21, no. 11, pp. 1586-1594, November 2010.
12. A. R Khan, M. Othman, S. A Madani and S. U. Khan, "A Survey of Mobile Cloud Computing Application Models", IEEE Communications Surveys Tutorials, vol. 16, no. 1, pp. 393-413, January 2014.
13. Mazhar Ali, Revathi Dhamotharan, Eraj Khan, Samee U Khan, Athanasios V Vasilakos, Keqin Li and Albert Y Zomaya, "SeSaSC: Secure Data Sharing in Clouds", IEEE Journal on Systems, vol. 11, issue. 2, pp. 395- 404, January 2017.
14. Aaram Yun, Chunhui Shi, and Yongdae Kim, "On Protecting Integrity and Confidentiality of Cryptographic File System for Outsourced Storage", Proceedings of the 2009 ACM Workshop on Cloud Computing Security, pp. 67-76, 2009.
15. Hsiao-Ying Lin and Wen-Guey Tzeng, "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding", IEEE Transactions on Parallel and Distributed Systems, vol. 23, issue. 6, pp. 995-1003 June 2012.
16. A. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A Survey of Mobile Cloud Computing Application Models", IEEE Communications Surveys Tutorials, vol. 16, no. 1, pp. 393-413, Jan 2014.
17. Mitsuru Ito, Akira Saito and Takao Nishizcki, "Secret Sharing Scheme Realizing General Access Structure", Lecture Notes on Fundamental Electronic Science, vol. 72, issue. 9, pp. 56-64, Feb 2010.
18. Yang Tang, Patrick P.C. Lee, John C. S Lui and Radia Perlman, "FADE: Secure Overlay Cloud Storage with File Assured Deletion", International Conference on Security and Privacy in Communication Systems, vol. 50, PP. 380-397, 2010.
19. R Perlmam, "File System Design with Assured Deletion", Third IEEE International Conference on Security in Storage, pp. 82-88, 2005.
20. Ashfia Binte Habib, Tasnim Khanam, Rajesh Palit, "Simplified File Assured Deletion (SFADE)- A User Friendly Overlay Approach for Data Security in Cloud Storage System", International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1640-1644, 2013.
21. Raisa Nusrat, Rajesh Palit, "Simplified FADE with sharing feature (SFADE+): A Overlay Approach for Cloud Storage System", 7th IEEE Annual Computing and COmmunication Workshop and Conference (CCWC), pp. 1-6, Jan 2017.
22. Mazhar Ali, Saif U R Malik, and Samee U Khan, "DaSCE: Data Security for Cloud Environment with Semi-trusted Third Party", IEEE Transaction on Cloud Computing, vol. 5, issue. 4, pp. 642-655, December 2017.
23. Jeevitha B K, Sindhura D, Thriveni J, and Venugopal K R, "DSCESM: Data Security for Cloud Environment with Scheduled Key Managers", International Conference on Advances in Electronics, Electrical and Computational Intelligence, May 2019.

AUTHORS PROFILE

Jeevitha B K received the Bachelor of Engineering degree in Computer Science and Engineering from Vivekananda Institute of Technology, Bengaluru in 2009 and the master of Technology in Jawaharlal Nehru College of Engineering, Shivamogga in 2014, both Visvesvaraya Technological University, Belgaum, India.

She is currently working toward the PhD degree from Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bengaluru, India. Her research interest include cloud computing. She is a member of IEEE since 2015. She has published research papers in reputed International Journals and Conference. She has 5 years of Research Experience.

Pushpa C N has completed Bachelor of Engineering in Computer Science and Engineering from Bangalore University, Master of Technology in VLSI Design and Embedded Systems from Visvesvaraya Technological University and PhD in Computer Science and Engineering from Bangalore University. Presently she is working as Assistant Professor in the Department of Computer Science and Engineering at University Visvesvaraya College of Engineering, Bangalore. She has 18 years of teaching experience. She has presented and published 25 research papers in the International Conferences and Journals. Her research interest includes Web mining, Personalized Web Search, Semantic Web and Data Analytics.

Dr. Thriveni J has completed Bachelor of Engineering, Masters of Engineering and Doctoral Degree in Computer Science and Engineering. She has 4 years of industrial experience and 23 years of teaching experience. Currently she is Professor in the Dept. of CSE, University Visvesvaraya College of Engineering, Bangalore. She has over 90 research papers to her credit. She has produced five doctorate students and guiding 06 Ph.D Students. Her research interests include Networks, Data Mining and Biometrics.

Dr. K. R. Venugopal is currently the Vice Chancellor Bangalore University, Bengaluru. He obtained his Bachelor of Engineering from University Visvesvaraya College of Engineering. He received his Masters degree in Computer Science and Automation from Indian Institute of Science Bengaluru. He was awarded Ph.D. in Economics from Bangalore University and Ph.D. in Computer Science from Indian Institute of Technology, Madras. He has a distinguished academic career and has degrees in Electronics, Economics, Law, Business Finance, Public Relations, Communications, Industrial Relations, Computer Science and Journalism. He has authored and edited 64 books on Computer Science and Economics, which include Petrodollar and the World Economy, C Aptitude, Mastering C, Micro-processor Programming, Mastering C++ and Digital Circuits and Systems etc., He has filed 101 patents. During his three decades of service at UVCE he has over 640 research papers to his credit. His research interests include Computer Networks, Wireless Sensor Networks, Parallel and Distributed Systems, Digital Signal Processing and Data Mining. He is a Fellow of IEEE, ACM and ISTE.

Retrieval Number: D7303049420/2020©BEIESP
DOI: 10.35940/ijeat.D7303.049420
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication