I.J. Wireless and Microwave Technologies, **2016, 1, 23-33**
Published Online January 2016 in MECS(http://www.mecs-press.net)
DOI: 10.5815/ijwmt.2016.01.03

Available online at ***http://www.mecs-press.net/ijwmt***

A Secure and Efficient Client-Side Encryption Scheme in Cloud Computing

Tania Gaur[a], Divya sharma[b]

[a,b] *CSE Department, The Northcap University (Formerly Itmu), Gurgaon, India*

Abstract

Computation process has changed itself from desktop computing to cluster computing and to grid computing and now a days it goes towards cloud computing. In today's trend, Cloud computing is being preferred by majority of the organization because of its major advantages in terms of scalability and multi tenancy. Among all these advantageous features of cloud computing, the only hitch is the security of the data that is stored in the cloud. The security clause rests with the cloud service provider but security can be applied on client side also and it is possible if the user knows how his data is secured. This paper proposes a mechanism which can help the user to ensure the security of his data at the client-side.

Index Terms: Cloud computing security, cryptography techniques and client-side encryption.

1. Introduction

From an individual user to an entire organization, cloud has found its use in many spheres of life. There are various cloud service providers, like Amazon, Google, Salesforce.com and many more, which have so many offers for the users. These options are offered to the user as different service models. The cloud services are offered majorly in the form of the following three models:

a) Software as a Service (SaaS)
b) Platform as a Service (PaaS)
c) Infrastructure as a Service (IaaS)

The security of the data that is stored in these clouds is one the major concern in today's time. The activities' of the hackers and malignant users has been on the increase lately. Thus, both the corporate data and the private data which is sensitive in nature need to be provided utmost protection. This hitch is the only "flaw in the plan"

* Corresponding author
E-mail address: gaurtania@gmail.com, divya84kkr@gmail.com

where cloud is concerned because cloud comes with these risks[1]. The various attacks in the past on the data stored in the cloud are enough to raise an alarm and call for urgent remedial actions to be taken place to secure the data. And as these risks become more dangerous, the cloud users become more uncomfortable in trusting the security mechanisms applied on the server-end by the cloud service providers. The following section discusses the difference in applying security techniques applied at the server-side as compared to those applied at the client-end.

2. Server-Side Security Vs Client-Side Security

The purpose of this paper is to propose a model that takes care of the data at the client-side. So, it is important to discuss as to how the process of securing the data at the client-side is different from securing the data at the server-side. This is important in the sense that it will help the cloud user in understanding the measures taken-up at both sides and help him make a choice as to which type of security mechanism provides better security to the data.

In the server-side security process, the security measures are taken up by the cloud-service providers. Here, along-with providing cloud services, the cloud service providers adopts certain security measures to safeguard the data that is stored in the cloud. The cloud service providers usually employ cryptographic techniques of encryption, digitally signing the data along with the use of keys. These keys are encryption keys and access keys which are also kept with the cloud service providers. This is done by the cloud consumer's terms of service for a cloud which is a legally binding agreement between two parties. This agreement is done in two parts – service agreement and SLA [2] (Service Level Agreement). The service agreement is the legal contract between the cloud consumer and cloud service provider. An SLA is a small document consisting f the technical performance promises made by the cloud-service providers.

When the security of data is ensured at the client-end, the major difference when compared to the security at the server-end is that at the client-side, the cloud user becomes responsible for ensuring security of the data. The user has control over the encryption keys. The cloud user also controls who can have access permission to his data. There are many applications available that provides the above said level of security at the client-end.

3. Related Work

The Internet users depend upon Cloud Computing heavily for various computing applications and resources. The main motive of the Cloud service providers is to provide these services in a virtualized manner. One of the main issue that requires constant attention of cloud computing is the security of the Cloud Storage. The security of the data stored in the Cloud Storage is wholly in the hands of the Cloud service providers. The Cloud Providers assures the Cloud users that their data, that is stored on their servers, is safe. The users do not take part in securing the data. The various cloud service providers claim that they take appropriate and efficient steps in securing the cloud storage. But there have been attacks on famous cloud service providing companies such as Google, Salesforce.com and Dropbox [3]. Many of the cloud service providers employee third party companies for services which has caused the users to lose their trust with these companies. Thus, the steps taken by these cloud service providers should be equally strong. The privacy and security of cloud computation depend mostly on whether these providers have implemented adequate and robust security controls as required in order to secure the data or not.

As the need was felt to ensure the security of the data at the client-side, many organizations/ individuals attempted to do this. Resultantly, many applications were developed in the last decade to take the control of security of data from the hands of the server and give it in the hands of the clients. A few of such applications providing cloud data security at client-side along with their respective features are given in Table 1.

Table 1. Comparison of available Client-side Applications

Features Applications	Free Storage	Encryption algorithm	License Type	Product Type	Distinguishing Features
Tresorit[4]	5GB	AES-256 HMAC-SHA-512	Unrestricted freeware	Combines a web service with a stand-alone program	• End-to-end encryption • Encrypted backup • Application for mobile and desktop • Dashboards for user and device • Email attachments required no more
OwnCloud[5]	NA	AES-256	Unrestricted freeware	Runs as a stand-alone program on a user's computer	• Encryption of files • Synchronization • Calendar • Task scheduler • Bookmarking • PDF viewer
BoxCryptor[6]	2GB	AES-256 RSA	Free for private use only	Runs as a stand-alone program on a user's computer	• Fly-based on the –fly encryption • Permission management • Filename encryption • Password reset • Master key
Viivo[7]	NA	AES-256	Unrestricted freeware	Runs as a stand-alone program on a user's PC	• File compression • Multi cloud support • Rights management • Dropzone • BoxEdit
CloudFogger[8]	NA	AES-256 RSA	Unrestricted freeware	Combines a web service with a stand-alone program	• Platform independent • File sharing • Preferenece manamegent • Receiver bound to have the app to access files
Wuala[9]	5GB	AES-256(encryption) RSA-2048(signatures) SHA-256(integrity)	Freeware	Combines a web service with a stand-alone program	• Cross-platform compatibility • Mounted network drive • Backup • Syncing • Groups • Sharing • Tags • Versioning • Time Travel
SpiderOak[10]	2GB	AES256 in CFB mode, HMAC-SHA256	Unrestricted freeware	Combines a web service with a stand-alone program	• Version management • Secure and sync • Restore everything • Trash recovery • Unlimited devices
TeamDrive[11]	2GB	AES-256, RSA	Free for private use only (limited)	Runs as a stand-alone application	• Version management • Multi-location backup • Conflict file management • Access rights management • supported file formats

4. Proposed Model

The proposed work will ensure the security of the data at the client-end. The proposed implemented application will ensure the above mentioned level of security by applying cryptographic technique of encrypting the data before uploading to the cloud. In this proposed mechanism to ensure safety of data at the client-end, three participants are involved. These three participants are Cloud User (CU), Third Party Auditor (TPA) and Cloud Service Provider (CSP). Table 2 describes the role of these three participants:

Table 2 Participants and their functions in the proposed model

Participant	Function
Cloud User (CU)	CU is the end user who accesses and enjoys the services provided by the Cloud Service Provider.
Third Party Auditor (TPA)	TPA ensures the authenticity of the entities over the network. It ensures secure means of sharing the key between two users.
Cloud Service Provider (CSP)	CSP is the one who is responsible for providing the cloud services to the end users.

In this paper, a proposal is made to ensure the security of the data at the client-end. This proposal employs the concept of Diffie-Hellman algorithm to generate the shared key that will be further used for encryption and decryption of data. The proposal also has a Third Party Auditor who is a very well trusted party. TPA checks the authenticity of the end users and also keeps a watch over the secure exchange of key between the end users. The following are the operations that are performed in the above stated proposed mechanism:

a) Key generation – (KeyGen())

CU uses Diffie-Hellman algorithm to generate the shared key. This key is used to encrypt the file before uploading this data to the cloud storage. The algorithm for Diffie-Hellman algorithm is given below:

Sender:
1. Pick a secret number "a"
2. Calculate "A" using the following:
(Values of g and p are pre-determined)

$$g^a mod\ p = A \tag{1}$$

3. Send "A" to the receiver
4. Receive "B" from the receiver
5. Calculate the shared secret code from the following:

$$B^a mod\ p = S_{AB} \tag{2}$$

Receiver:
1. Pick a secret number "b"
2. Calculate "B" using the following:
(Values of g and p are pre-determined)

$$g^b mod\ p = B \tag{3}$$

3. Send "B" to the receiver
4. Receive "A" from the receiver
5. Calculate the shared secret code from the following:

$$A^{b} mod\ p = S_{AB} \tag{4}$$

b) Key sharing –

The shared key that is being generated with the help of Diffie-Hellman algorithm via the KeyGen() operation needs to be sent over the network to another user who wishes to access the data. This is because the data that is stored in the cloud is in the encrypted format and when it is retrieved from the cloud, it needs to be decrypted. To decrypt the file, the shared key is required at this end. The key generated using Diffie-Hellman algorithm between the CUs by using the secure channel.

c) Encryption -

In the first step, the author who generated the data, encrypted the data (D) using the shared key that was generated through the KeyGen() process using the Diffie-Hellman algorithm. The URL and the file are encrypted with the help of AES Algorithm. The file, thus encrypted, is uploaded to the cloud storage.

d) Decryption –

Once the file is retrieved from the cloud, to read the contents of the file and to retrieve the URL, thus received by the CU2 from CU1, it needs to be decrypted as it was stored in the cloud in an encrypted format and the URL was encrypted before sharing it with CU2, respectively. To decrypt this retrieved file and the URL, the end user CU2 needs the shared key generated through the KeyGen() process generated by the owner of the file with the help of Diffie-Hellman algorithm. This shared key is received by the recipient of the file via a secure channel.

e) URL generation –

Using the Amazon Web Services, the URL can be generated by using the Generate Web URL option. When the owner of the data wishes to share the file with another user, he will share the URL with that user. This URL is encrypted before transmission to another user to ensure that even if it is attacked during transmission, it will be safeguarded against it.

5. Proposed Architecture

This section describes the structure of the proposed system with the help of a flowchart. Fig. 1 represents the flowchart for the operations at the sender's end and fig. 2 gives the flowchart for the operations at the receiver end.

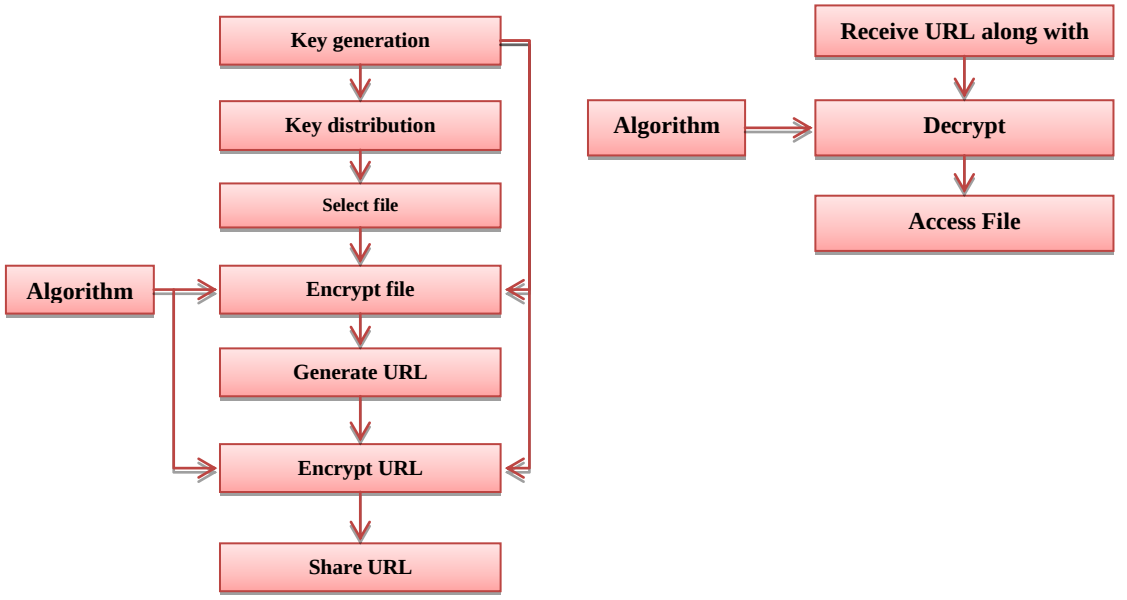

Fig.1. (a) Flowchart for the Operations at the Sender's End; (b) Flowchart for the Operations at the Receiver End

In the first step at the sending side (fig. 1(a)), the key that will be used to encrypt or decrypt the data and the URL will be generated. The key thus generated will be shared between the users with the help of TPA. The next step involves encrypting the file before uploading to the cloud storage. Once the data is stored in the cloud, a URL link of that particular location is retrieved. Refer to Fig.3. This URL link is encrypted in the next step which will be sent by the sender to the client requesting the file. At the receiving end (fig. 1(b)), the receiver will acquire the key from TPA and will receive the encrypted URL from the sender. This is represented in Fig. 4. The receiver will decrypt the link and access the data.

6. Proposed Mechanism

In this section, a scheme has been proposed to provide the client-side security measures that can be adopted to secure the data stored in the cloud. This proposal employs the cryptographic techniques of encryption using Advanced Encryption Standard [12] (AES) and Diffie-Hellman [13] algorithms. AES will be used to encrypt the data before uploading to the cloud. Diffie-Hellman algorithm is a public key cryptography algorithm that will be used to generate the key pair.

In the first step, the key pair is generated with the help of Diffie-Hellman algorithm. The file that needs to be secured is then encrypted by using the AES algorithm. This is represented by fig. 2. In the next step, this encrypted file is then uploaded to the cloud.

Note: The cloud storage service used for demonstration of this proposal is Amazon Web Services [14] Simple Storage Service (AWS S3).

Once the file is uploaded to the cloud in the encrypted format, in the next step the address of the location where the file is stored is generated. The next step is what is different from the tradition. Refer the fig. 3, the URL is not shared between users as such; it will be encrypted before sharing with another user. TPA is responsible for authenticating the entities and ensuring the integrity of the data. The keys that are generated using Diffie-Hellman algorithm are shared in a secure manner. Fig. 4 shows the different steps of the system that are performed at the receiving end. The shared and encrypted URL is decrypted by the receiver, thereby

accessing the file. The proposed architecture is shown below:

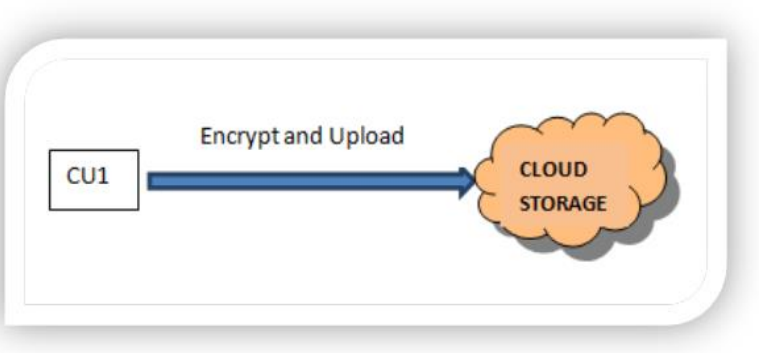

Fig.2. To Encrypt and Upload the File

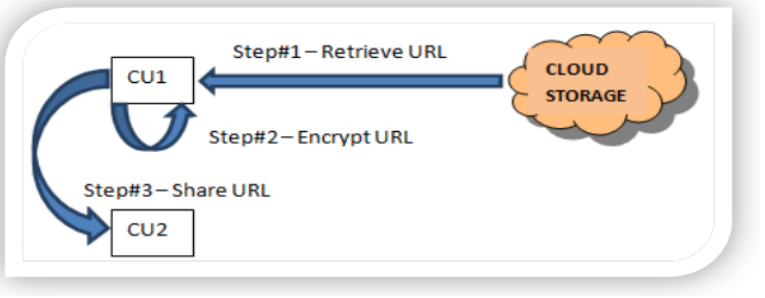

Fig.3. To Encrypt the URL at Sender's End

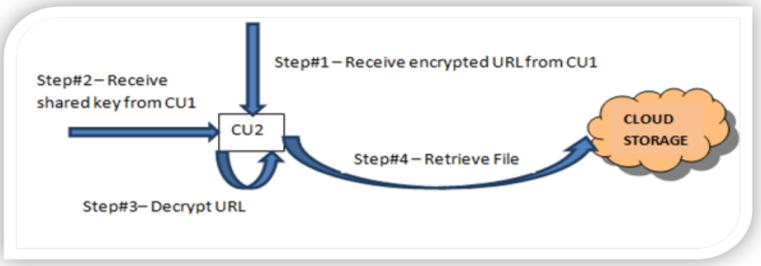

Fig.4. To Share URL and Decrypt URL Along with File at Receiver's End

7. Implementation of the Proposed System

This subsection describes in detail the implementation phase of the proposed system. In order to encrypt our data before uploading to the cloud storage, we need a key to encrypt it. In this prototype, we generate the key by using the symmetric key-agreement algorithm which is the Diffie-Hellman algorithm which is explained in the section 4,it also covers mechanism of sharing of keys with the help of a third party auditor (TPA), in case of sharing the data with multiple users and the role of TPA in our proposed model.

1. **Diffie-Hellman:** Diffie-Hellman is used for the key generation in the first step, which is later on shared with the help of Third party auditor. Where a key is exchanged in between sender and receiver and by the help of that in future they can use encrypted file for decryption. This is the first step in algorithm.
2. **TPA:** Third party auditor is an agency that authenticates and distributes the keys between two parties for the exchange of data. The role of TPA is to ensure authenticity of various entities in a network. The key sharing between a user and TPA will take place by using Diffie-Hellman algorithm. If a user A wants to share his file with user B, he will send the request to TPA. TPA will contact user B and tell him that A has requested a session with him. If B agrees, TPA will create a session key for both the users.
3. **Implementing the AES Algorithm:** It is a non-Fiestel cipher. Each round in AES, except the last round, consists of four transformations that are invertible. The last round has only three transformations. (1) *KeyExpansions*-round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more. (2)*InitialRound*-AddRoundKey—each byte of the state is combined with a block of the round key using bitwise xor. (3)*Rounds*-(3.1)SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.(3.2)ShiftRows—a transposition step where the last three rows of the state are shifted cyclically a certain number of steps. (3.3)MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column. (3.4)AddRoundKey. (4)*Final Round* (no MixColumns)- (4.1)SubBytes (4.2)ShiftRows(4.3)AddRoundKey.

8. Experimental Result

An application has been developed and implemented in C# programming Language using .Net framework and LAN to attain the desired functionalities of the cloud user, TPA and the CSP. All the operations that are performed in the proposed solution, are performed at the client-end. This makes the users feel a little more secure as compared to the prior situation where the users were required to rely completely on the cloud service providers and the servers for the security.

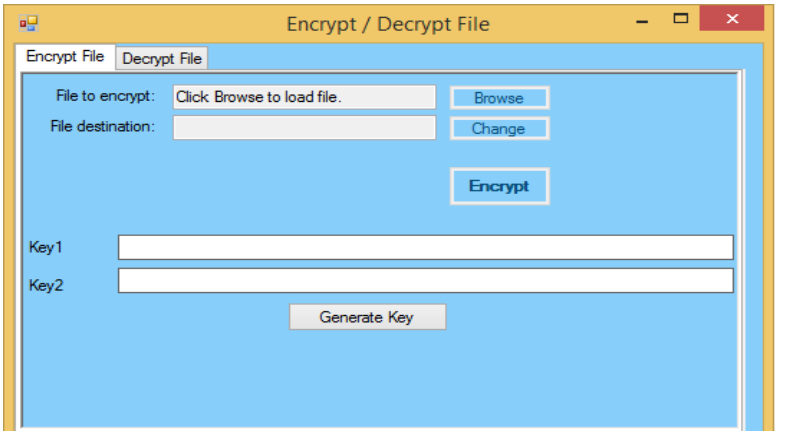

Fig.5. Encrypt/ Decrypt Files

The user can then encrypt (Figure 5) the file by using AES algorithm before uploading to the cloud. This process requires the generation of keys that is done with the help of Diffie-Hellman algorihtm. The key thus generated is stored with the TPA in reference to the sender.

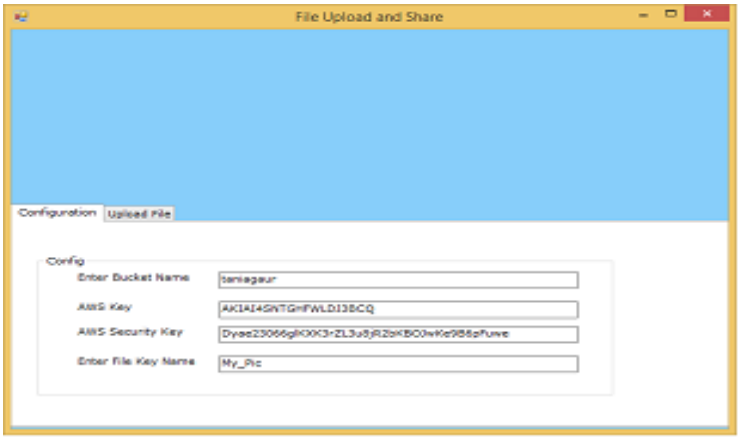

Fig.6. Uploading the File

After the file is encrypted, it is uploaded to the cloud. This process is depicted in figure 6.

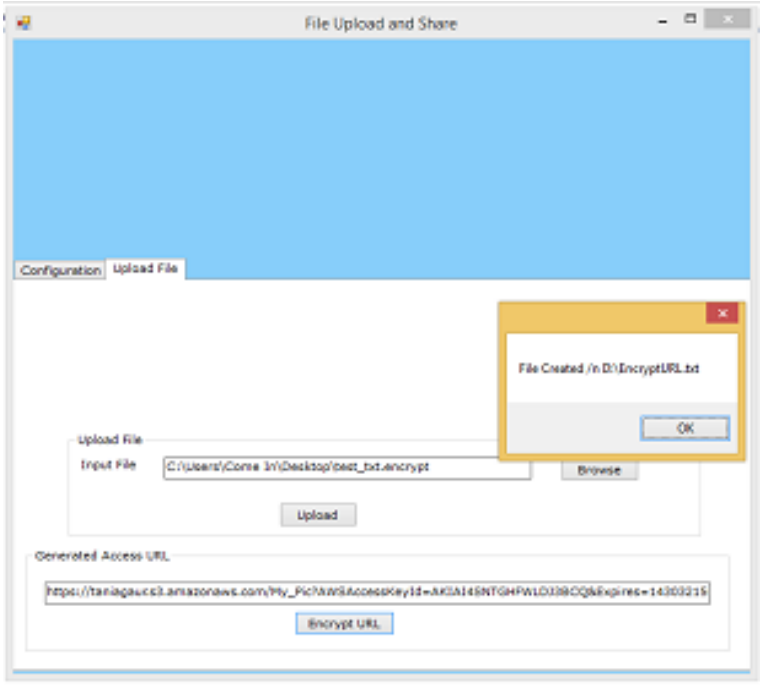

Fig.7. Encrypt URL

The next step is to encrypt the URL before it can be send over the network. Figure 7 shows how URL is generated and encrypted.

Figure 8 depicts the role of TPA in our application. It is used to entertain the requests various users make in order to access keys regarding to various files. The user who wishes to decrypt a file asks the TPA for the key related to that file. The TPA then sends the requested key to that particular user.

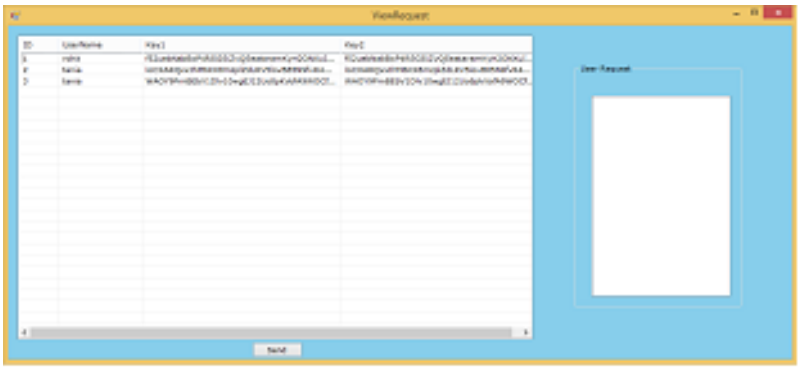

Fig.8. Function of TPA

The elements that make our solution feasible are the encryption of the files before they are uploaded to the cloud storage servers, and the encryption of link before they are transferred via insecure channels. The application runs as a stand-alone program that can be integrated with another web based application in order to share the URL of the data stored in the cloud. It is assumed that these three participants lie in the same system domain and are sharing the same parameters. With the help of this application, messages can be transferred between these three entities and the desired results are obtained.

9. Conclusions

This research paper has proposed a model that will help the end user to secure their data at the client-side instead of trusting the cloud service providers blindly with their data. The proposed mechanism uses encryption to secure the data when it is stored in the cloud. It also employs the use of encrypted URL link to share the data between two users over the Internet. The URL link that is being shared between users is also encrypted to safeguard it against the attacks that could occur on it during transmission.

The shared key is generated with the help of Diffie-Hellman algorithm. The data and the URL link are encrypted by using AES algorithm. The security analysis shows that the application that has been developed fulfils the requirements of the cloud user when it comes to the security of the data that is stored in the cloud. As there is always a scope for improvement in all spheres of life, so here is as well. One of the assumptions is that the TPA is neutral. And it is also assumed that the CU, TPA and the CSP all lie in the same domain and share the similar parameters. So work could be done in near future to make changes in this situation.

References

[1] T. Gaur and N. Kharb, “Security of Data Storage in Cloud Computing,” in *International Journal of Computer Applications*, Vol. 110, No. 10, pp: 15-18, 2015.

[2] P. Mell and T. Grance, “The NIST definition of cloud computing,” 2011.

[3] Dropbox confirms security glitch--no password required, http://news.cnet.com/8301-31921_3-20072755-281/dropbox-confirms-security-glitch-no-password-required/.

[4] Tresorit, “Tresorit,” [Online]. Available: https://tresorit.com/.

[5] Ben Martini and Kim-Kwang Raymond Choo, "Cloud storage forensics: own Cloud as a case study." *Digital Investigation* 10.4, pp: 287-299, 2013.

[6] BoxCryptor, "Introduction," [Online]. Available: https://www.boxcryptor.com/en.

[7] Viivo, “Viivo – Introduction,” [Online]. Available: https://viivo.com/.

[8] CloudFogger, "CloudFogger - Introduction," [Online]. Available: http://www.cloudfogger.com/en/.
[9] Mager Thomas, Ernst Biersack, and Pietro Michiardi. "A measurement study of the Wuala on-line storage service." In *Peer-to-Peer Computing (P2P), 2012 IEEE 12th International Conference*, pp. 237-248, IEEE, 2012.
[10] SpiderOak, "SpiderOak – Features," [Online]. Available: https://spideroak.com/features/.
[11] TeamDrive, "TeamDrive – Introduction," [Online]. Available: www.teamdrive.com/.
[12] B. A. Forouzan, "*Cryptography & Network Security*." McGraw-Hill, Inc, 2007.
[13] N. Tirthani and R. Ganesan, "Data Security in Cloud Architecture Based on Diffie Hellman and Elliptical Curve Cryptography," in *IACR Cryptology ePrint Archive,* pp:49, 2014.
[14] http://aws.amazon.com/s3/.
[15] Wazir Zada Khan, Yang Xiang, Mohammed Y Aalsalema, Quratulain Arshada, "*The Selective Forwarding Attack in Sensor Networks: Detections and Countermeasures*", I.J. Wireless and Microwave Technologies, Volume 2, pp 33-44, 2012.

Authors' Profiles

Tania Gaur has completed her Master's in Technology in Computer Science and Engineering from ITM University, Gurgaon in 2015. She has done her Bachelor's in Technology in Computer Science and Engineering from MDU, Rohtak. She has published 2 research papers, one in an International Journal and another in a National Conference. Her research interests lies in Cryptography and Cloud Computing.

Dr. Divya Sharma employed as Assistant Professor in Department of Computer Science, The NORTHCAP UNIVERSITY (Formerly ITMU), Gurgaon. She has done Ph.D. in Computer Science from Kurukshetra University Kurukshetra. She has more than 15 research papers to her credit in various International/National Journals and Conferences. She is CCNA certified. Her research interests are in Mobile Ad hoc Networks and Cloud Computing.

How to cite this paper: Tania Gaur, Divya sharma,"A Secure and Efficient Client-Side Encryption Scheme in Cloud Computing", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.6, No.1, pp.23-33, 2016.DOI: 10.5815/ijwmt.2016.01.03