International Journal of Grid and Distributed Computing
Vol. 10, No. 10 (2017), pp.1-20
http://dx.doi.org/10.14257/ijgdc.2017.10.10.01

Safety as a Service (SFaaS) Model - The New Invention in Cloud computing to establish a Secure Logical Communication Channel between Data Owner and the Cloud Service Provider before Storing, Retrieving or Accessing any Data in the Cloud

Debabrata Sarddar[1], Himadri Biswas[2] and Priyajit Sen[3]

[1]Assistant Professor in the Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, India
[2]Assistant Professor in the Department of Computer Applications, Bengal College of Engineering & Technology, Durgapur, West Bengal, India
[3]Pursuing M.Tech in Computer Science and Engineering from University of Kalyani, Kalyani, Nadia, West Bengal, India
[1]dsarddar@klyuniv.ac.in, [2]mr.himadri.biswas@gmail.com, [3]priyajit91@gmail.com

Abstract

To store data securely in the Cloud Server is one of the key objectives of Cloud Computing. Always the Data Owners (DW) want the top level security from the Cloud Service Provider (CSP) to store their sensitive data as pay and use basis and enjoy the high quality service without hacking or misuse or unauthorized access of the data. This paper proposes new security architecture - Safety as a service (SFaaS) which helps to store data safely on the cloud server and enforce the access control on the data. The main concept is the Safety Service Provider (SSP) offers SFaaS, plays the vital role in secured data storage or access control on the data. So before storing any data on the cloud server or accessing the existing data, SSP registers the DW, the Authorized Users (AU) and the CSP to establish a secure logical communication channel between them. Here the DW or AU is not directly related to the CSP, rather than there is a direct mutual trust relation between the DW & the SSP, as well as the SSP & the CSP, i.e. without the SSP's concern the DW, AU or CSP no one is able to communicate each others.

Keywords*: Safety as a Service, Data Owner, Authorized Users, Cloud service provider, Safety Service Provider, Service Level Agreement, Encryption, Decryption*

1. Introduction

As one of the most stirring technologies which have advanced in the world today, Cloud Computing refers to the supply of computing resources over the Internet, where shared pool of resources such as software, platform, storage space and information are provided to customers on claim. Cloud computing is regarded as concentration that is considerable from both industry and academic circles due to a number of essential advantages namely: elasticity to extent up and down information technology competence, low managing overhead, cost efficiency instant access to a broad range of applications, and mobility which includes the consumers that can access information at any place wherever they are, without having them to work at their workplace. So, outsourcing data the data owners just have to pay the billed amount that are priory decided. Since the data owner actually releases responsive data to a remote CSP, so may give rise to certain privacy implications.

Received (March 14, 2017), Review Result (June 27, 2017), Accepted (July 10, 2017)

ISSN: 2005-4262 IJGDC

The proposed model provides the new computing environment- Safety-as-a-Service (SFaaS), which ensures the data security, integrity and access control on the data between the DW & the CSP, as well as the AU & the CSP. This means that the data stored remotely should be accessed only by the DWs or the AUs. So both the DW and the CSP needs to be safeguard from any false accusation may claimed by each other to get illegal compensations.

2. Related Works

On cloud storage security many research works had done. The idea of trusted entity [1], where the CSP receives the files from trusted entity other than the Data Owner to impose the security issues. Ateniese *et al.* [3] have introduced a secure distributed storage protocol based on proxy re-encryption [2]. Using this protocol, the blocks are encrypted by the data owner with symmetric data keys, which are encrypted using a master public key. On the other hand the data owner holds a master private key to decrypt the symmetric data keys. The owner generates proxy re-encryption keys with the help of master private key and the authorized user's public key. The proxy re-encryption keys are then used by a semi-trusted server to translate a cipher text into a form that can be decrypted by a specific granted user, and thus enforces access control for the data.] For secure sharing of data on un-trusted servers a cryptography-based file system called Plutus is designed by Kallahalla *et al.* [4]. Some certified users have the freedom to read and write on the data, while others can only read the data. In a hypervisor-based virtualization environment [15,16] the security architecture imposed against malicious attacks, but no such security issues between users, data owners and the service providers.

In Plutus, a set of files with similar attributes represented by a file-group, and each file-group is related with a symmetric key called file-lockbox key. A data file is split into blocks and a unique symmetric key called a file-block key is required for encryption of each block. Now the file-block key is encrypted by the file-lockbox key of the file-group to which the data file belongs. The file-lockbox key is distributed to the data owner only if they want to share a file-group with a set of users. them. Plutus allowed two operations read and write/modify on the file blocks, where delete operation can be carried out by overwriting an existing block with null. To share read/ write data Popa *et al.* [5] have introduced a cryptographic cloud storage system called Cloud Proof, which has been designed to offer security assurance in the service level agreements of cloud storage systems. the concept of over-encryption introduced by Wang *et al.* [6] to enforce access control and the owner encrypts the data block-by-block, and creates a binary tree of the block keys. The owner helped out by the binary tree to reduce the number of keys given to each consumer, where different keys in the tree can be generated from one common parent node. Over-encryption performed by the remote storage server to prevent from getting access to updated data blocks by the revoked users. Third party auditor's idea has been used before in outsourcing data storage systems, especially for clients with constrained computing resources and capabilities, *e.g.*, [7-10]. The idea of TTP is used in a slightly different fashion in the proposed work [11]. CSP provides the auditing process of the data, which is done by the authorized users and routed to the TTP only to resolve disputes that may arise regarding data integrity or originality. Decreasing the storage overhead on the CSP side is cost-effectively a key feature to lesser the fees paid by the consumers. Furthermore, reducing the overall computation cost in the system is another key aspect.

A little amount of the owner's work is hand over to the TTP to reach these goals. To access control and secure sharing of data on untrustworthy servers have focused on the proposal [3], [4], [12], [13]. For achieving mutual trust between the data owners and the remote servers and the full block-level dynamic operations are concerned like insert, delete, modify and append are not inside their scope. Though an proficient access control

technique and full data dynamics are handled over remote servers have mentioned [6], whereas data integrity, newness property, and mutual trust are not addressed. In [14] the SLAM (Service Level Agreement Manager) module acts as a Resource Locator – for availability of VMs, Resource Allotter – to initiate operations on the service request to the VMs, Resource Collector / Observer – for observation & Collection of data along with resources and Bill generator Agent – which not only generates bill but runs at frequent intervals to update account logs of individual users or consumers or brokers. So, as and when required this model provides the updated information (consumed resources and storage spaces) to the user, though security encryption services is the missing link for collecting and collating the records.

3. Overview and Rationale

3.1. Cloud Service Models

The cloud computing service models are –

3.1.1. Software as a Service (SaaS)

A readymade application, along with any necessary software, operating system, hardware, and network are provided by this SaaS model.

3.1.2. Platform as a Service (PaaS)

Hardware, network and an operating system are provided by this PaaS model, whereas the consumer can install or builds up its own software and applications.

3.1.3. Infrastructure as a Service (IaaS)

Only the hardware and networks are supplied by this IaaS model, the customer set up or develops its own operating systems, software and applications.

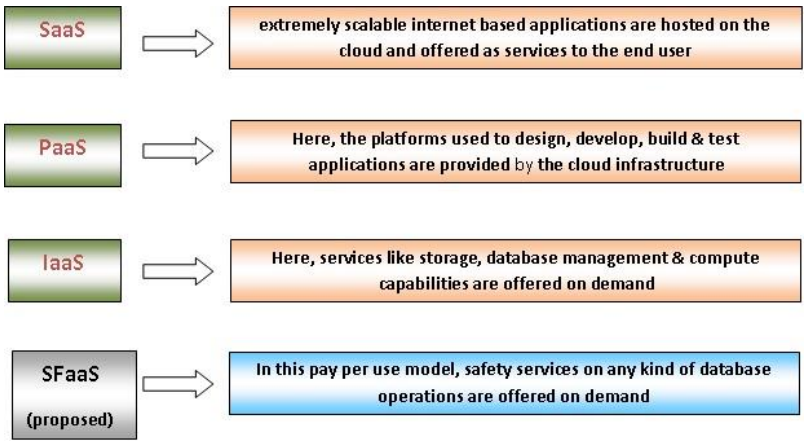

Figure 1. Cloud Service Models

3.2. Proposed Service Model

3.2.1. Safety as a Service (SFaaS)

SFaaS provides the security to store data securely on the cloud server and impose the access control on the data. The main concept is the Safety Service Provider (SSP) under SFaaS, plays the important role to store or access control on the data safely. So before storing any data on the cloud server or accessing the existing data, SSP registers the Data

Owners (DW), the Authorized Users (AU) and the Cloud Service Providers (CSP) to establish a secure logical communication channel between them.

3.3. Deployment Models

Cloud services are classically made available via a private, community, public or hybrid clouds.

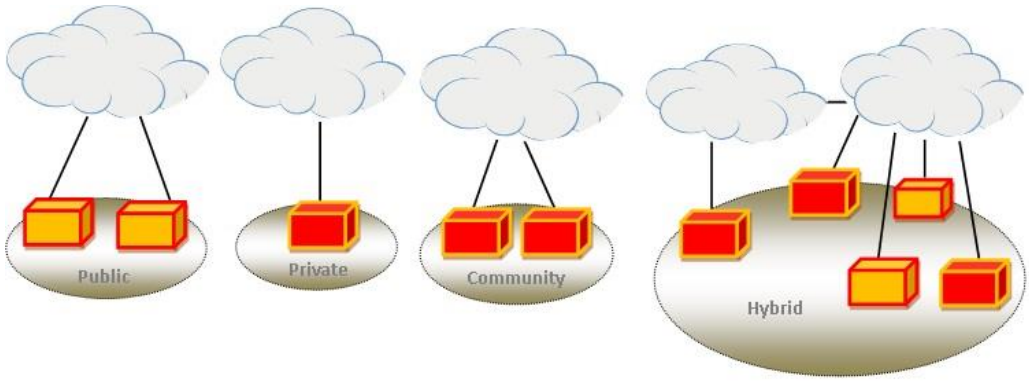

Figure 2. Deployment Models in Cloud Computing

3.3.1. Public Cloud

Public cloud offers services through the internet, possessed and are maintained by a cloud service provider. The services provided publicly like email, online image storage services, communal networking sites *etc.*, except these different enterprise services can also make available in a public cloud.

3.3.2. Private Cloud

In a private cloud, the cloud infrastructure is controlled and fully dedicated for a particular organization, and is directed by the organization or an intermediary.

3.3.3. Community Cloud

In a community cloud, the provision of sharing services among numerous organizations and the services not only available to those organizations, the infrastructure may be possessed and controlled by the organizations itself or by a cloud service provider.

3.3.4. Hybrid Cloud

Hybrid cloud is a mixture of two or more clouds (private, public or community) that remain single unit but are bound together and provides the benefits of scalability, reliability, quick response and potential cost savings of public cloud storage and with the security and complete control of private cloud storage.

3.4. Data Owner (DW)

A data owner that can be an organization or a group producing perceptive data to be stored in the cloud and made available for controlled external use.

3.5. Authorized users (AU)

The authorized users are a set of clients who have the right to access the remote data over the cloud.

3.6. Cloud Service Provider (CSP)

A service provider who provides some components and services to the customers for storage or software or hardware or network services via SaaS or IaaS or PaaS is referred to as Cloud Service Provider. A cloud service provider is not only provides services to the businesses or individuals, it guarantees that for taking the responsibility of everything related to the components or services that used by the customers.

3.7. Safety Service Provider (SSP)

The Safety Service Provider (SSP) under SFaaS, plays the vital role in secured data storage or access control on the data. Actually SSP offers the SFaaS. So before storing any data on the cloud server or accessing the existing data, SSP registers the Data Owners (DW), the Authorized Users (AU) and the Cloud Service Providers (CSP) to establish a secure logical communication channel between them. SSP holds a security log table which holds all the public keys (in indexed order) of the communicating nodes or the nodes that want to communicate each other.

3.8. Service Level Agreement (SLA)

It is an agreement or a deal between a service provider and its inside or outside customers that define what services the provider will provide and defines the performance standards the provider is compelled to meet.

3.9. Cryptography

Cryptography means private messages are confined from unauthorized access, *i.e.* messages are protected to prevent grasp. Two types of operations employed by cryptography:

3.9.1. Encryption or Enciphering

It is the process of transforming the simple text (understandable format) into cipher text (not understandable format). The mathematical notation of Encryption is: CT=Enc(ST, Ek), where CT is the Cipher Text, Enc ( ) is the Encryption Function, ST is the Simple Text to be encrypted by the Encryption Key, Ek.

3.9.2. Decryption or Deciphering

It is the process of transforming the cipher text back to simple text. The mathematical notation of Decryption is: ST=Dec (CT, Dk), *i.e.* ST=Dec(Enc(ST, Ek), Dk); Where Dec ( ) is the Decryption Function, and CT is decrypted by the Decryption Key Dk. So the resulting text is the Simple Text.

3.10. Representation of Different Types of Data

As we know that Cloud is the vast area of storing, retrieving and accessing of different types of data. So, it is very essential to know about the nature of different types of data to provide the security to all types of data. Whatever thing we wish to represent in a computer, we need to find a way of converting it into numbers. A text is broken down into its characters and each character is assigned a number (like ASCII code or Unicode). An image is broken down into a fine grid of little squares (Pixels) and each pixel's color is further broken down into Red, Green and Blue components, Each component is measured on a scale of 0 to 255, say (which is 8 bits. So an image is converted into a stream of numbers. Audio data (a continuous wave to be exact) is broken down into very close points (a discrete or "digital" wave) then the amplitude of these points is measured and converted into numbers (called sampling). A video is just an image with an added

dimension of time. In the physical world time is unremitting, but for the profit of our computer we need to sample time, much like we did with audio. So, the basics are easy: capture heaps of images at a adequately high rate, convert each image into numbers, add the necessary metadata and we have our video file
Let's do the math for a standard definition, 90-minute movie:

90 minutes→60 seconds for each minute→30 frames for each second→480 rows for each image→ 640 pixels for each row→3 bytes (24 bits) for each pixel and this would take up 150 gigabytes (GB) of disk space.

4. Proposed Work

Emerging Cloud computing infrastructures provide computing resources on demand based on pay as you need. Figure 3 describes the proposed architecture of "A new strategy- **Safety as a service (SFaaS)** in cloud Computing to establish a secure logical communication channel between Data Owner and the Cloud Service Provider before storing, retrieving or accessing any data on the cloud" is based on the ability of the SFaaS model, where Safety Service Provider (SSP) helps to make a secure logical communication path between the Data Owner (DW), the Cloud Service Provider (CSP) and the Authorized Users (AU).

4.1. System Components and Relations

From Figure 3 we conclude that---

i) DW that can be an organization generating sensitive data to be stored in the cloud.
ii) The AUs' are a set of owner's clients who have the right to access the remot data. The AUs' are solely managed by the DWs.
iii) A CSP who manages cloud servers and provides paid storage space on its infrastructure for storing the owner's files and make them accessible for authorized users.
iv) The SSP offers SFaaS, plays the vital role to make a secure communication channel between the nodes before storing, retrieving or updating any data on the cloud server. So before storing any data on the cloud server or accessing the existing data, SSP registers the Data Owners (DW), the Authorized Users (AU) and the Cloud Service Providers (CSP) to establish a secure logical communication channel between them. SSP holds a log table which holds all the public keys (in indexed order) of the communicating nodes or the nodes that want to communicate each other.
v) Cloud Servers (CS) are directly related to the CSPs. One DW may store their Sensitive data under different CSPs. Similarly, one CSP is not only engaged with one DW, but many DWs. Whatever may be, actually the Owner's data is stored in the CSs through CSP with the consent of SSP.

In our work SSP take the responsibility to make the secured connection between the communicating nodes (*i.e.* DW & the CSP), that means no third party can hijack the data at the time of communication, because the receiving nodes decrypts the data by its confidential key (CK), which is self generated and only known to itself.

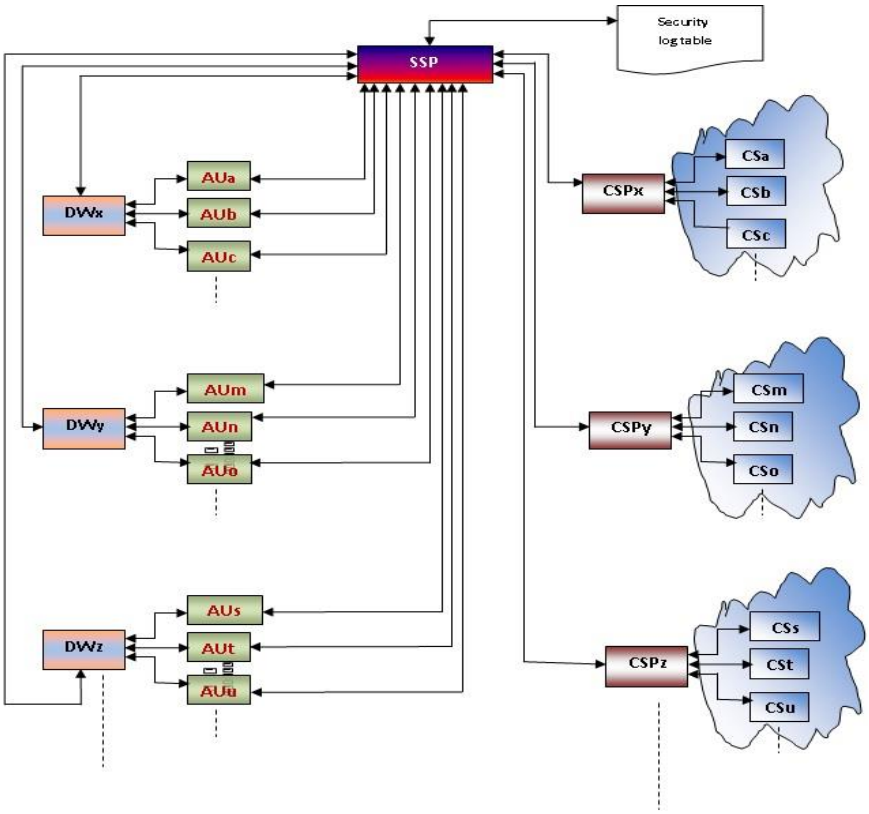

Figure 3. Proposed Architecture of "A New Strategy- Safety as a Service (SFaaS) in Cloud Computing to Establish a Secure Logical Communication Channel between DW & CSP as well as AU & CSP via SSP"

4.2. Safety as a Service (SFaaS) Model

Cloud computing atmosphere provides the computing resources on demand based on pay as you need. Different organizations store their secret data in the cloud storage and only the authorized clients can access the data. Most of the time it observes that there is a misunderstanding between the DW and the CSP, can claim to each other of any false accusation because of the direct mutual trust relation between them, *i.e.* a DW make a request directly to the CSP for getting the cloud services. So, no mediator in between them, who looks into the whole matter and find out the victim.

Main objective of our proposed work is to provide a secured cloud computing environment SFaaS Model, which provides the security of the communicating nodes to send and receive data securely on the cloud server and impose the access control on the data. The main concept is the Safety Service Provider (SSP) under SFaaS, plays the critical role to store data securely or to access control on the data. SSP generates a security log table (contains the public keys of the communicating nodes or the nodes want to communicate) in the indexed order of the public keys. So before storing any data on the cloud server or accessing the existing data, SSP registers the Data Owners (DW), the Authorized Users (AU) and the Cloud Service Providers (CSP) to establish a secure logical communication channel between them. Once the connection is established between two nodes, they can communicate data securely and no chance of interference by others.

The nodes that want to communicate shortly or in near future, first send their public keys to the SSP to register themselves. If any node changed its public key, then have to inform to the SSP and re-registered again. So public keys can be changed dynamically, so

unauthorized nodes can't send not a single fake message to the authorized node, *i.e.*, not a single chance of accepting the fake messages by the authorized nodes.

4.2.1. Working Process of Sfaas

One thing is remember that, in our proposed work all the request and response messages of the communicating nodes in an encrypted format. For encryption and decryption, two types of keys are required- Public key (available to the communicating nodes) and Confidential key (self generated and only known to the owner). When the two nodes want to communicate each other, suppose node A want to communicate to node B, then A sends a requesting message (encrypted by B's public key) to node B and B receives the message (decrypts by B's confidential key). On the other hand, when B sends a reply message (encrypts by the public key of A) to A and A receives the message (decrypts by A's confidential key).

Figure 4 describes a request made by the DW to the SSP to store the data on the cloud server (CS). After receiving the request, SSP retrieves the public key of the authorized CSP from its security log table and forwards to the DW. Getting the reply message from SSP, DW decrypts it with its own confidential key and send a test data (encrypt by the public key of CSP) to the CSP. After receiving the test data (decrypts by CSP's confidential key), CSP realize that DW wants to communicate with it, as soon as CSP request (encrypts by SSP's public key) to SSP for getting the public key of the DW. Now SSP decrypts the request (by its confidential key), retrieves the public key of DW from its security log table and forwards (encrypts by CSP's public key) to CSP. Receiving the reply (decrypts by CSP's confidential key) from SSP, CSP generates a test data & attached with the test data (previously sent by DW) and finally sends as a message (encrypts by DW's public key) to DW. DW decrypts the

message (by its confidential key) and retrieves the test data (previously sent to CSP), compares with its original data; if both are same, then DW make sure about the reply from CSP that CSP is reliable. Otherwise again generate another test data and send to CSP for authentication. Now DW sends a reply message (encrypts by the public key of CSP) to CSP. CSP received the message from DW decrypts by its own confidential key and retrieves the test data (previously sent to DW), compares with its original data; if both are same, then CSP make sure about the reply from DW that DW is reliable. So DW and CSP both realize that the communication channel between them is secure and the regular communication can be started. Before start the communication, CSP sends a Service Level Agreement (SLA) to DW, if DW agreed upon the SLA then sends an accepted message to CSP and communication starts.

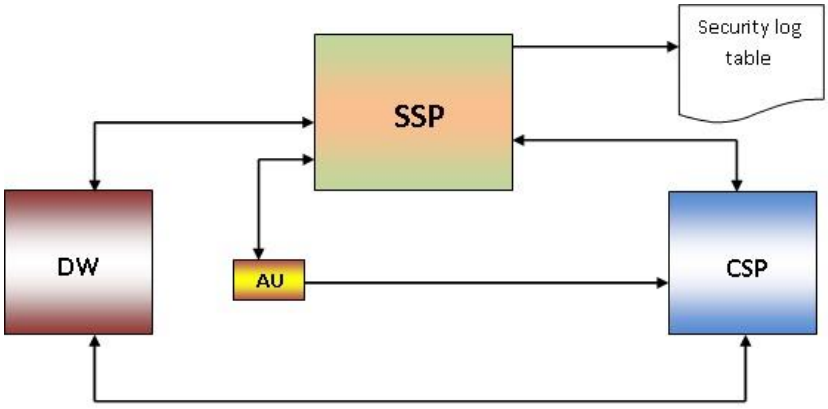

Figure 4. SSP Offering SFaaS

5. Algorithm & Flowchart

5.1. Encryption-Decryption Algorithm

Step 1: Read a packet.
Step 2: Each packet is divided into number of blocks.
Step 3: Convert each block of data into an array of unsigned characters (1D array).
Step 4: Convert each block of data from unsigned characters array to array of integers (2D array).
Step 5: Number of bits for each element is 8 (Let n= 8).
Step 6: Go to Step 7 for Encryption and Step 8 for Decryption.
Step 7: Process Encryption
7.1: Read a block
7.1.1: Read an element of the block
7.1.1.1 Convert the element into its equivalent binary.
7.1.1.2 N1 = Decimal equivalent of the binary number.
7.1.1.3 N2 = Decimal equivalent of the last 4 bit numbers (starting from the LSB).
7.1.1.4 Multiply N2 by 2 and Subtract from N1.
7.1.1.5 Multiply the number of bits n by 2 and subtract 1 from it.
7.1.1.6 Now add (2n-1) with (N1-2N2) and we get the Encrypted Element N which is equal to (N1-2N2) + (2n-1) *i.e.*

$$N = (N1-1)-2(N2-n) \quad \text{----------(i)}$$

7.1.1.7 Store as an element of the block.
7.1.1.8 **If** next element is available
Go to Step 7.1.1 to read another element.
7.1.1.9 **Else if** next block is available
Go to Step 7.1 to read another block
7.1.1.10 **Else if** next packet is available
Go to Step 1 to read another packet
7.1.1.11 **Else**
Go to Step 9
Step 8: Process Decryption
8.1: Read a block
8.1.1: Read an element of the block
8.1.1.1 Convert the number into equivalent binary.
8.1.1.2 N = Decimal equivalent of the binary number.
8.1.1.3 N`2 = Decimal equivalent of the last 4 bit numbers (starting from the LSB).
8.1.1.4 Multiply N`2 by 2 and Subtract from N.
8.1.1.5 Multiply the number of bits n by 2 and subtract 1 from it.
8.1.1.6 Now add (2n-1) with (N- 2N`2) and we get the Decrypted Number N` which is equal to (N-2N`2) + (2n-1) *i.e.*

$$N`=(N-1)-2(N`2-n) \quad \text{----------(ii)}$$

8.1.1.7 Store as an element of the block.
8.1.1.8 **If** next element is available
Go to Step 8.1.1 to read another element.
8.1.1.9 **Else if** next block is available
Go to Step 8.1 to read another block
8.1.1.10 **Else if** next packet is available
Go to Step 1 to read another packet
8.1.1.11 **Else**

Go to Step 9.

Step 9: End

Note: From the general definition of Encryption and Decryption and from Equation (i) and (ii) we conclude that N=Enc[N1,{(N1-1)-2(N2-n)}] & N`=Dec[N,{(N-1)-2(N2-n)}], where {(N1-1)-2(N2-n)} and {(N-1)-2(N2-n)} are the Encryption and Decryption key respectively.

5.2. Flowchart for Encryption-Decryption

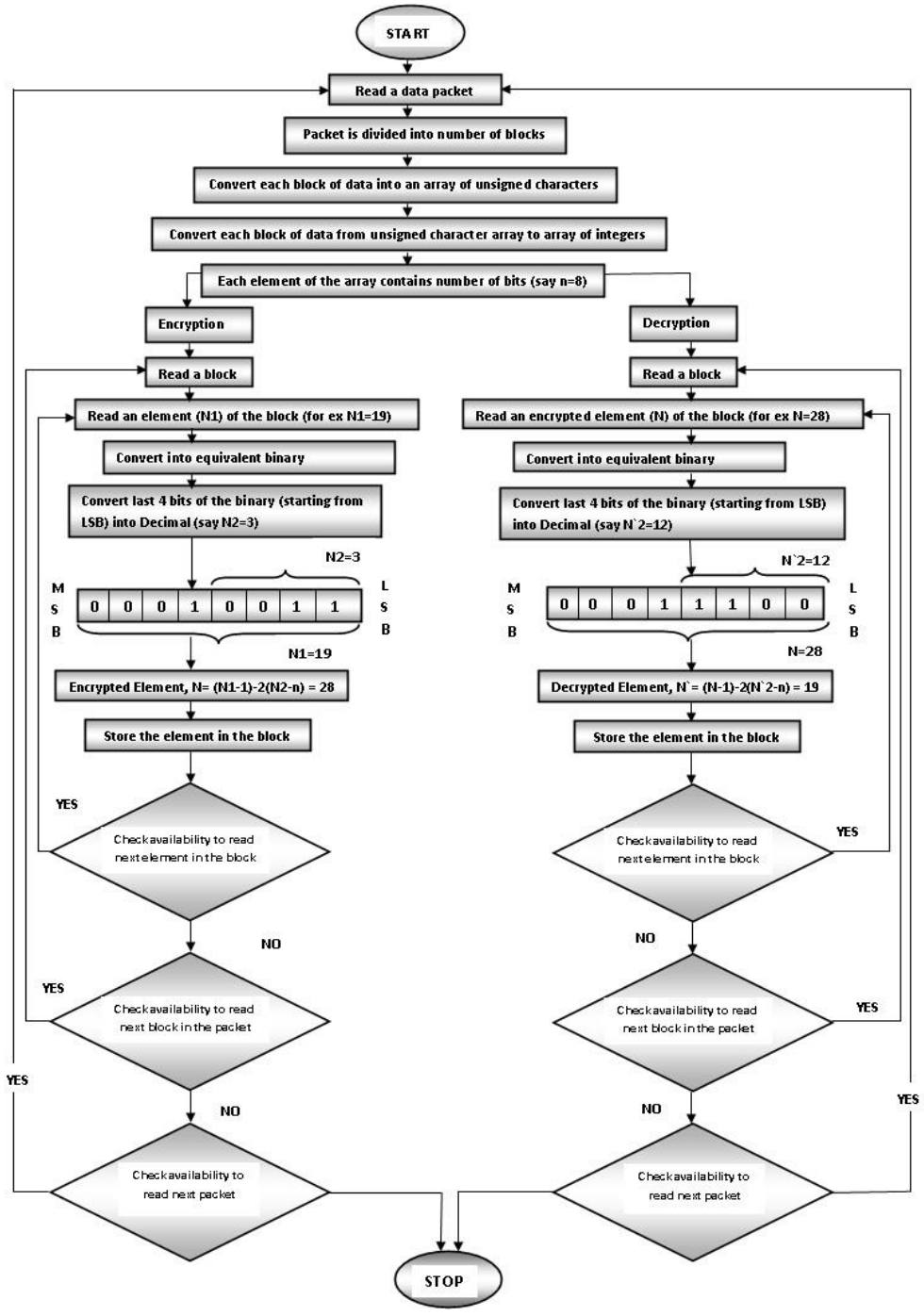

Figure 5. Flowchart of Encryption and Decryption Process

5.3. Algorithm for Establishing a Secure Logical Communication Channel between the DW & the CSP

Before writing the algorithm let us know the following—

Let us assume that SSP receives the public keys of all the communicating nodes or the nodes want to communicate each other (*i.e.* the public keys of DW, CSP, AU); Similarly, the public key of SSP is known to all the communicating nodes or the nodes want to communicate each other. Each communicating nodes generate its own confidential key, that only known to that particular node. So, Safety Service Provider (SSP) acts as a safety as a service (SFaaS) which holds one confidential key (generated by itself and known to itself only) and all the public keys in indexed order of Data Owner (DW), Cloud Service Provider (CSP) and the Authorized Users (AU).

1. DW sends an encrypted data packet to SSP requesting for an authorized CSP to store cloud data securely, *i.e.*, to establish a secure logical communication channel between DW and the CSP.
    - 1.1 The requesting message contains- the request for authorized CSP
    - *1.2* The packet contains – *{identity of DW, public key of DW, requesting message, a time stamp set by DW}*
    - 1.3 DW encrypts the Packet by the public key of SSP and sends to SSP.

2. SSP decrypts the received packet by its confidential key and searches for the available CSP.
    - 2.1 **IF CSP is available** –
        - 2.1.1 SSP finds out the location of the CSP.
        - 2.1.2 SSP searches for the public key of the CSP from its own list.
        - *2.1.3* SSP sends an encrypted packet as a reply to the DW, which contains –*{the previously sent requested message by DW (step 1.1), the time stamp set by the DW (step 1.2), CSP's identity, CSP's public key}*
        - 2.1.4 Packet encrypts by the public key of DW.
    - 2.2 **ELSE**
        - *2.2.1* SSP creates a packet which contains – *{the previously sent requested message by DW (step 1.1) & the time stamp set by the DW (step 1.2), a NULL value}*
        - 2.2.2 SSP encrypts the packet by the public key of DW as a reply to the DW.

3. Getting the reply message from SSP, DW decrypts it with its own confidential key and compares the received values (of the previously sent requested message by DW (step 1.1), the time stamp set by DW (step 1.2)) with the actual values.
    - 3.1 **IF the values are same** –
        - 3.1.1 DW make sure about the reply from the SSP
        - 3.1.2 **IF** CSP's identity & public key are available—
            - 3.1.2.1 DW generates a test data.
            - *3.1.2.2* DW creates the packet which contains – *{DW's identity, test data created by DW, a time stamp set by DW}*
            - 3.1.2.3 DW encrypts the packet by the public key of CSP and sends to the CSP.
        - 3.1.3 **ELSE**
            - 3.1.3.1 Go to step 1.

3.2 **ELSE**

3.2.1 Go to step 1.

4. CSP receives the packet, decrypts by its confidential key and realize that DW wants to establish a communication path with it. CSP sends an encrypted data packet to SSP requesting for DW's public key.

4.1 Requesting message contains- request for the public key of the DW.

4.2 The packet contains—*{identity of CSP, identity of DW, requesting message, a time stamp set by CSP}*

4.3 CSP encrypts the packet by the public key of SSP and sends to the SSP.

5. SSP receives the packet, decrypts by its confidential key, search for the public keys of DW & CSP and sends an encrypted packet to the CSP.

5.1 Retrieves the public key of DW from its list.

5.2 SSP sends an encrypted packet as a reply to the CSP, which contains – *{the previously sent requested message by CSP (step 4.1) & the time stamp set by the CSP (step 4.2), public key of DW}*

5.3 SSP encrypts the Packet by the public key of CSP and sends to CSP.

6. Getting the reply message from SSP, CSP decrypts it with its confidential key and compares the received values (of the previously sent requested message by CSP (Step 4.1) & the time stamp set by CSP (step 4.2)) with the actual values.

6.1 **IF the values are same** –

6.1.1 CSP make sure about the reply from the SSP

6.1.2 **IF DW's public key is available** –

6.1.2.1 CSP generates a test data.

6.1.2.2 CSP creates the packet which contains – *{test data & the time stamp set by DW(step 3.1.2.2), test data generated by CSP, a time stamp set by CSP }*

6.1.2.3 CSP encrypts the packet by the public key of DW and sends to the DW.

6.1.3 **ELSE**

6.1.3.1 Go to step 4.1

6.2 **ELSE**

6.2.1 Go to step 4.1

7. DW receives the packet, decrypts by its confidential key and compares the received values (of the previously sent test data by DW & the time stamp set by DW (Step 3.1.2.2)) with the actual values.

7.1 **IF the values are same** –

7.1.1 DW makes sure about the reply from the CSP that CSP is reliable.

7.1.2 DW encrypts the packet *{test data sent by CSP & the time stamp set by the CSP (in step 6.1.2.2)}* with the public key of CSP and sends to the CSP.

7.2 **ELSE**

7.2.1 Go to step 3.1.2.1

8. CSP receives the packet, decrypts by its confidential key and compares the received values (of the previously sent test data by CSP & the time stamp set by CSP (in Step 6.1.2.2)) with the actual values.

8.1 **IF the values are same** –

8.1.1 CSP make sure about the reply from the DW that DW is reliable.

8.2 **ELSE**

8.2.1 Go to step 6.1.2.1

9. CSP sends the Service Level Agreements (SLA) with a time stamp to the DW, encrypted by the public key of DW. So the encrypted packet contains – *{SLA, time stamp set by CSP}*.

10. DW receives the packet and decrypts by its own confidential key.

10.1 **IF agreed upon SLA** –

10.1.1 DW encrypts the packet *{SLA, time stamp previously set by CSP(in step 9)}* with the public key of CSP and sends to CSP.

10.2 **ELSE**

10.2.1 Go to step 1 for another CSP.

11. CSP receives packet and decrypts by its confidential key, and sends an encrypted packet *{Ack, time stamp set by CSP}* as an acknowledgement to DW. The packet encrypts by the public key of DW.

12. Now DW receives the message decrypts by its confidential key and keeps it up for documentation.

13. Secure logical communication channel is established between DW & the CSP and regular communications can be started.

So after checking the reliability between the communicating nodes, the above algorithm is sufficient to establish a secure logical communication channel between the DW and the CSP.

Note: The above algorithm ensures that the secure channel can be established between any two nodes like the DW & AUs, as well as the AUs & the CSP and so on.

5.4. Flowchart for Establishing a Secure Logical Communication Channel between the DW & the CSP

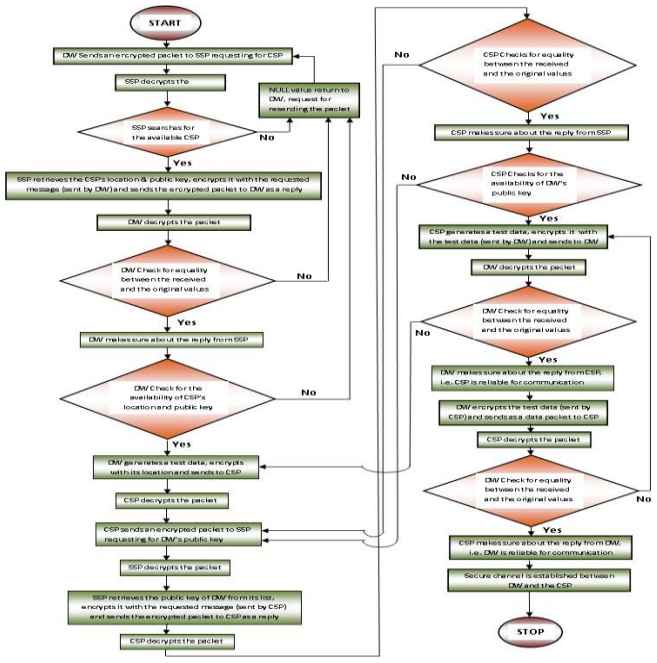

Figure 6. Flow Chart: Establishing a Secure Logical Channel between DW &the CSP using SSP before Storing or Modifying the Data in the Cloud Server

6. Detailed analysis of SFaaS Model (A case study)

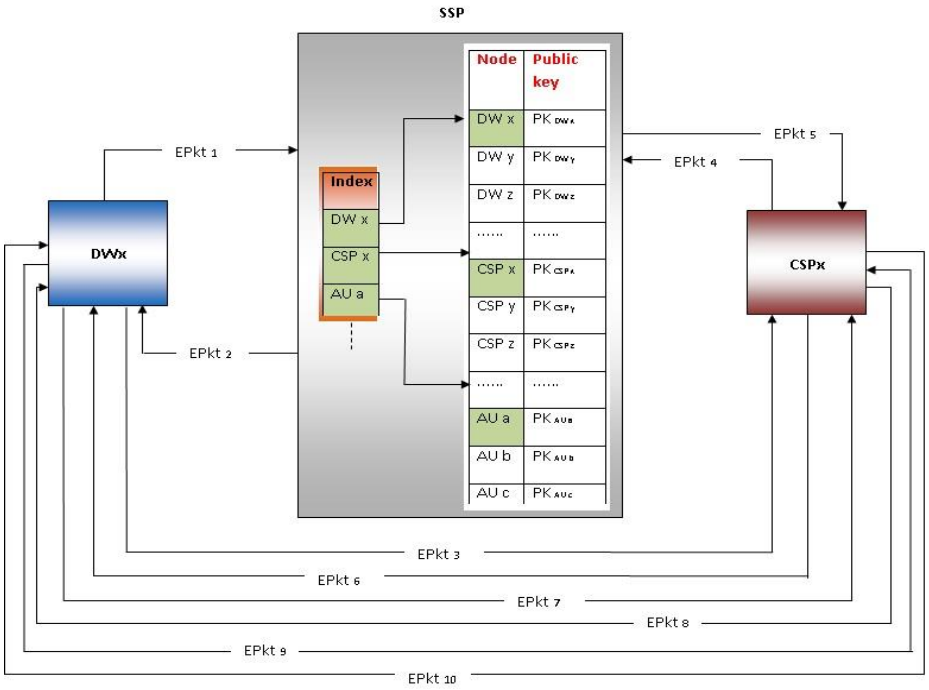

Figure 7. Safety as a service (SFaaS) module -- Establish a Secure Logical Communication Channel between Data Owner and the Cloud Service Provider through SSP before Storing Any Data in the Cloud"

In SFaaS Model, to establish the connection, the communicating nodes have to follow some protocols for improving security purposes. Only when they fulfil all the protocols, connection will establish. Messages sent between the DW, CSP and the SSP are always in an encrypted packet. Once a connection is established (between DW and CSP), only the communicating two nodes sends and receives data securely without the help of SSP until and unless the connection is off. If the connection is off, again they have to re-establish the connection before sending any data between them. SSP maintains a security log table, holds the public keys of all the communicating nodes. To improve the security, the public keys are changed dynamically, *i.e.* when the established connection is cut off due to any reason and at the time of reestablishment of the connection, they may change their public keys, but they must have to inform to the SSP because of maintaining the security log table properly.

Let us an example of two nodes DW_x and CSP_x want to establish a secure communication channel between them, and they have to go through the following procedures:

Notations:

DW_x	:	x is the name of the Data Owner
CSP_x	:	x is the name of the Cloud Service Provider
SSP	:	Safety Service Provider

PK_{SSP}	:	Public key of SSP
CK_{SSP}	:	Confidential Key of SSP
PK_{DWx}	:	Public key of DW_x
CK_{DWx}	:	Confidential Key of DW_x
PK_{CSPx}	:	Public key of CSP_x
CK_{CSPx}	:	Confidential Key of CSP_x
RQ_{DWx}	:	Request from DW_x
RQ_{CSPx}	:	Request from CSP_x
TS_{aDWx}	:	Time stamp a set by DW_x
TS_{bDWx}	:	Time stamp b set by DW_x
TS_{cCSPx}	:	Time stamp c set by CSP_x
TS_{dCSPx}	:	Time stamp d set by CSP_x
TS_{eCSPx}	:	Time stamp e set by CSP_x
TS_{fDWx}	:	Time stamp f set by DW_x
TS_{gCSPx}	:	Time stamp g set by CSP_x
TD_{aDWx}	:	Test Data a generated by DW_x
TD_{bCSPx}	:	Test Data b generated by CSP_x
$EPkt_1$	:	An encrypted packet sends from DW_x to SSP
$EPkt_2$	:	An encrypted packet sends from SSP to DW_x
$EPkt_3$	:	An encrypted packet sends from DW_x to CSP_x
$EPkt_4$	:	An encrypted packet sends from CSP_x to SSP
$EPkt_5$	:	An encrypted packet sends from SSP to CSP_x
$EPkt_6$	:	An encrypted packet sends from CSP_x to DW_x
$EPkt_7$	:	An encrypted packet sends from DW_x to CSP_x
$EPkt_8$	:	An encrypted packet sends from CSP_x to DW_x
$EPkt_9$	:	An encrypted packet sends from DW_x to CSP_x
$EPkt_{10}$	:	An encrypted packet sends from CSP_x to DW_x

SLA_{CSPx} : Service Level Agreement sends from CSP_x to DW_x
SLA_{DWx} : DW_x agreed upon Service Level Agreement and sends to CSP_x
ACK_{CSPx} : Acknowledgement of agreement sends from CSP_x to DW_x

Step 1: DW_x sends an encrypted data packet ($EPkt_1$) to SSP requesting for an authorized CSP_x to store cloud data securely, *i.e.,* to establish a secure logical communication channel between DW_x and the CSP_x. The packet encrypts by the public key of SSP. So the packet contains -
$EPkt_1 = Enc\{(DW_x, PK_{DWx}, RQ_{DWx}, TS_{aDWx}), PK_{SSP}\}$

Step 2: SSP decrypts the received packet ($EPkt_1$) by its confidential key *i.e.* Dec ($EPkt_1$, CK_{SSP}) and searches for the available CSP. If find out then searches for the public key (PK_{CSPx}) of CSP_x from its own list and sends an encrypted packet ($EPkt_2$) as a reply to DW_x, which contains – $EPkt_2 = Enc\ \{(RQ_{DWx}, TS_{aDWx,} CSP_x, PK_{CSPx}), PK_{DWx}\}$. Otherwise, a NULL value will return to DW_x, *i.e.* $EPkt_2 = Enc\ \{(RQ_{DWx}, TS_{aDWx,} NULL)\ PK_{DWx}\}$.

Step 3: Getting the reply message from SSP, DW_x decrypts it with its own confidential key,*i.e.* Dec ($EPkt_2$,CK_{DWx}) and compares the received values (RQ_{DWx}, TS_{aDWx}) with the actual values. If the values are same and if CSP_x's identity & public key are available then DW generates a test data and creates an encrypted packet ($EPkt_3$) and sends to CSP_x, *i.e.* $EPkt_3 = Enc\ \{(DW_x, TD_{aDWx}, TS_{bDWx}), PK_{CSPx}\ \}$.
Otherwise, go to step 1 and re send the request packet, $Epkt_1$ again.

Step 4: CSP_x receives the packet ($Epkt_3$), decrypts by its confidential key *i.e.* Dec ($Epkt_3$,
CK_{CSPx}) and realize that DW_x wants to establish a communication path with it. Immediately CSP_x sends an encrypted data packet ($Epkt_4$) to SSP requesting for DW_x's public key. The packet encrypts by the public key of SSP.
So, the packet contains, $Epkt_4 = Enc\ \{(CSP_x, DW_x, RQ_{CSPx}, TS_{cCSPx}), PK_{SSP}\}$

Step 5: SSP receives the packet, decrypts by its confidential key, *i.e.* Dec ($Epkt_4$, CK_{SSP}); retrieves the public keys of DW_x & the CSP_x and sends an encrypted packet ($Epkt_5$) to the CSP_x. The packet encrypts by the public key of CSP_x. So the packet contains- $Epkt_5 = Enc\ \{(\ RQ_{CSPx}, TS_{cCSPx}, PK_{DWx}), PK_{CSPx}\}$

Step 6: Getting the reply message from SSP, CSP_x decrypts it with its confidential key, *i.e.* Dec ($Epkt_5$, CK_{CSPx}) and compares the received values (RQ_{CSPx}, TS_{cCSPx}) with the actual values. If the values are same and if DW_x's public key is available then CSP_x generates a test data (TD_{bCSPx}) and creates an encrypted packet ($EPkt_6$) and sends to DW_x. The packet encrypts by the public key of DW_x. So the packet contains - $Epkt_6 = Enc\ \{(TD_{aDWx}, TS_{bDWx}, TD_{bCSPx}, TS_{dCSPx}), PK_{DWx}\ \}$.
Otherwise, go to step 4 and re send the request packet, $Epkt_4$ again.

Step 7: DW_x receives the packet, decrypts by its confidential key, *i.e.* Dec ($Epkt_6$, CK_{DWx}) and compares the received values (TD_{aDWx}, TS_{bDWx}) with the actual values.
If the values are same, DW_x make sure about the reply from the CSP_x that CSP_x is reliable and DW_x sends an encrypted packet ($EPkt_7$) to CSP_x. The packet encrypts by the public key of CSP_x. So the packet contains – $EPkt_7 = Enc\ \{(TD_{bCSPx}, TS_{dCSPx}), PK_{CSPx}\ \}$. Otherwise, go to step 3 and re send the test data *i.e.* packet send packet $Epkt_3$ again.

Step 8: CSP_x receives the packet, decrypts by its confidential key *i.e.* Dec ($Epkt_7$, CK_{CSPx})

and compares the received values (TD_{bCSPx}, TS_{dCSPx}) with the actual values. If the values are same, CSP_x make sure about the reply from the DW_x that DW_x is reliable. Otherwise, go to step 6 and re send the test data *i.e.* send packet $Epkt_6$ again.

Step 9: CSP_x sends the Service Level Agreements, SLA_{CSPx} with a time stamp encrypted by the public key of DW_x. So the encrypted packet contains – $EPkt_8$ = Enc {(SLA_{CSPx}, TS_{eCSPx}), PK_{DWx} }.

Step 10: DW_x receives the packet and decrypts by its own confidential key, *i.e.* Dec ($EPkt_8$, CK_{DWx}). If DW_x agreed upon Service Level Agreements, SLA_{DWx}, DW_x encrypts the packet ($EPkt_8$) by the public key of CSP_x and sends to CSP_x. So the encrypted packet contains— $EPkt_9$ = Enc {(SLA_{CSPx}, TS_{eCSPx}, SLA_{DWx}, TS_{fDWx}), PK_{CSPx} }. Otherwise, go to step 1 and request for another CSP.

Step 11: CSP_x receives packet and decrypts by its confidential key, *i.e.* Dec ($EPkt_9$, CK_{CSPx}) and sends an encrypted packet ($EPkt_{10}$) as an acknowledgement to DW_x. The packet encrypts by the public key of DW_x. So the encrypted packet contains— $EPkt_{10}$ = Enc {(SLA_{DWx}, TS_{fDWx}, ACK_{CSPx}, TS_{gCSPx}), PK_{DWx} }.

Step 12: Now DW_x receives the packet ($EPkt_{10}$), decrypts by its confidential key, *i.e.* Dec ($EPkt_{10}$, CK_{DWx}) and keep it for documentation.

Step 13: Secure logical communication channel is established between DW_x & the CSP_x and Communications can be started safely.

7. Discussion

In our proposed model, we have developed an encryption-decryption algorithm which encrypt any type of data, the public keys are changed dynamically so unauthorized nodes can't send not a single fake message to the authorized node, *i.e.* not a single chance of accepting the fake messages by the authorized nodes. Therefore, we say that our proposed model is highly secured as well as robust. Here we use the index structure of public keys and the indexing is based on the categories of the nodes, so it is easy to retrieve the public keys by the SSP from its security log table, a much amount of time will save to retrieve the public keys from its list. Moreover, there is no chance of cheating by the SSP because it only helps to establish the secure connection between the nodes; not to store, retrieve or update the owner's data.

8. Conclusion

Some aspects related to outsourcing data storage are based on the trusted third party (TTP), an entity trusted by all other system components, and has proficiency and ability to detect and specify unfair parties. TTP is an self-regulating entity, and thus has no inducement to conspire with any party in the system. So TTP act as a verifier that can validate the correctness of data. However, any possible leakage of data towards the TTP must be prevented to keep the outsourced data private, but the TTP may still able to cheat and return stale data to authorized users after the auditing process is done.

The proposed scheme in this work uses the SSP, totally different from TTP. Here SSP allows the data owner to outsource data that is responsive to a Cloud Service Provider, and it also ensures that only the authorized users receive the data that is outsourced. Further, it enables direct mutual trust between the Cloud service provider and the data owner, *i.e.* SSP acts as a media which only help to make the secure connection between the communicating nodes or the nodes that want to communicate each other, but not directly store or retrieve or supply of any secured data. So there is no chance of cheating by the SSP. To achieve these goals, a small amount of fees to be paid by the owner to the SSP.

9. Future Scope

Our proposed work is basically based to establish the secure logical communication channel between the nodes that want to communicate each other. In future we enhance our work to protect the cloud environment, *i.e.* the direct involvements of SCaaS is required to secure the cloud storage, as well as cloud computing resources also.

References

[1] U.R. Godase and J. Shinde, "Study of Providing Security for Cloud Storage System", in International Journal of Innovative Research in Advanced Engineering (IJIRAE), vol. 2, Issue 1, **(2015)**, pp. 313.
[2] M. Blaze, G. Bleumer and M. Strauss, "Divertible protocols and atomic proxy cryptography", in EUROCRYPT, **(1998)**, pp. 127-144.
[3] G. Ateniese, K. Fu, M. Green and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage", in Proceedings of the Network and Distributed System Security Symposium, NDSS, The Internet Society, **(2005)**.
[4] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang and K. Fu, "Plutus: Scalable secure file sharing on untrusted storage", in Proceedings of the FAST 03 Conference on File and Storage Technologies, USENIX, **(2013)**.
[5] R. A. Popa, J. R. Lorch, D. Molnar, H. J. Wang and L. Zhuang, "Enabling security in cloud storage SLAs with cloud proof", in Proceedings of the 2011 USENIX Conference on USENIX Annual Technical Conference, Ser. USENIXATC'11, USENIX Association, **(2011)**.
[6] W. Wang, Z. Li, R. Owens and B. Bhargava, "Secure and efficient access to outsourced data", in Proceedings of the 2009 ACM workshop on Cloud computing security, ser. CCSW 09. ACM, (**2009)**, pp. 55-66.
[7] M. A. Shah, M. Baker, J. C. Mogul and R. Swaminathan, "Auditing to keep online storage services honest", in HOTOS'07: Proceedings of the 11th USENIX workshop on hot topics in operating systems, Berkeley, CA, USA, **(2007)**, pp. 1-6.
[8] M. A. Shah, R. Swaminathan and M. Baker, "Privacy-preserving audit and extraction of digital contents," Cryptology PrePrint Archive, Report 2008, **(2008)**.
[9] Z. Hao, S. Zhong and N. Yu, "A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability", IEEE Transactions on Knowledge and Data Engineering, vol. 99, no. PrePrints, **(2011)**.
[10] C. Wang, Q. Wang, K. Ren and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing", in INFOCOM, **(2010)**, pp. 525-533.
[11] A. F. Barsoum and M. Anwar, Hasan Department of Electrical and Computer Engineering, University of Waterloo, Ontario, Canada, "Enabling Data Dynamic and Indirect Mutual Trust for Cloud Computing Storage Systems".
[12] E. J. G. H. Shacham, N. Modadugu and D. Boneh, "Sirius: Securing remote untrusted storage", in Proceedings of the Network and Distributed System Security Symposium, NDSS. The Internet Society, **(2003)**.
[13] S. D. C. Di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi and P. Samarati, "Over-encryption: Management of access control evolution on outsourced data", in Proceedings of the 33rd International Conference on Very Large Data Bases. ACM, **(2007)**, pp. 123-134.
[14] Rajesh Bose, Himadri Biswas, Debabrata Sarddar, Manas Kumar Sanyal, "Cloud Billing & Verification of Consumed Resources and Storage Spaces by a Cloud User", in International Journal of Applied Engineering Research, vol. 11, no. 9, **(2016)**, pp. 6568-6576.
[15] F. Sabahi, "Secure Virtualization for Cloud Environment Using Hypervisor-based Technology", In Journal of Machine Learning and Computing, vol. 2, no. 1, **(2012)**.
[16] R. Bose and D. Sarddar, "A Secure Hypervisor-based Technology Create a Secure Cloud Environment" in International Journal of Emerging Research in Management &Technology, vol. 4, Issue2.

Authors

Debabrata Sarddar, he is Assistant Professor in the Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, India, completed Ph.D at Jadavpur University. He completed M. Tech in Computer Science & Engineering from DAVV, Indore in 2006, and B.E in Computer Science & Engineering from NIT, Durgapur in 2001. He published more than 150 research papers in different journals and conferences. His research interest includes wireless and mobile system and Cloud computing. Email: dsarddar@klyuniv.ac.in.

Himadri Biswas, he is Assistant Professor in the Department of Computer Applications, Bengal College of Engineering & Technology, Durgapur, West Bengal, INDIA. He has also 14 years of teaching experiences and worked under several institutions. He is currently pursuing PhD from University of Kalyani. He has completed his M. Tech in Computer Science & Engineering from WBUT in 2010, and his MCA from St. Xavier's College, Kolkata under IGNOU, in 2004. His research interest includes cloud computing, mobile computing. Email:mr.himadri.biswas@gmail.com

Priyajit Sen, he is presently pursuing M.Tech in Computer Science and Engineering in the Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, India. He completed his MCA from Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, India in 2015. His research interest includes Mobile Computing, Wireless Sensor Network and Cloud Computing.
priyajit91@gmail.com

