Computer Systems Science & Engineering
DOI:10.32604/csse.2022.019622
Article

Tech Science Press

Secure Data Sharing with Confidentiality, Integrity and Access Control in Cloud Environment

V. Rajkumar[1,*], M. Prakash[2] and V. Vennila[3]

[1]Department of Computer Science and Engineering, Krishnasamy College of Engineering and Technology (Affiliated to Anna University, Chennai), Cuddalore, 607109, India
[2]Department of Computer Science and Engineering, School of Computing, SRM Institute of Science & Technology, Kattankulathur, 603203, India
[3]Department of Computer Science and Engineering, K.S.R. College of Engineering (Affiliated to Anna University, Chennai), Tiruchengode, 637215, India
[*]Corresponding Author: V. Rajkumar. Email: raj7win@gmail.com
Received: 20 April 2021; Accepted: 28 May 2021

Abstract: Cloud storage is an incipient technology in today's world. Lack of security in cloud environment is one of the primary challenges faced these days. This scenario poses new security issues and it forms the crux of the current work. The current study proposes Secure Interactional Proof System (SIPS) to address this challenge. This methodology has a few key essential components listed herewith to strengthen the security such as authentication, confidentiality, access control, integrity and the group of components such as AVK Scheme (Access List, Verifier and Key Generator). It is challenging for every user to prove their identity to the verifier who maintains the access list. Verification is conducted by following Gulliou-Quisquater protocol which determines the security level of the user in multi-step authentication process. Here, RSA algorithm performs the key generation process while the proposed methodology provides data integrity as well as confidentiality using asymmetric encryption. Various methodological operations such as time consumption have been used as performance evaluators in the proposed SIPS protocol. The proposed solution provides a secure system for firm data sharing in cloud environment with confidentiality, authentication and access control. Stochastic Timed Petri (STPN) Net evaluation tool was used to verify and prove the formal analysis of SIPS methodology. This evidence established the effectiveness of the proposed methodology in secure data sharing in cloud environment.

Keywords: Secure interactional proof system; access control; multi-step authentication; Gulliou-Quisquater protocol

1 Introduction

Cloud computing is the next-gen technology which finds its applications across different sectors for information storage and security concerns. In cloud computing model, data privacy and prevention of data loss are the major concerns to be addressed [1]. In this scenario, the current research work proposes

a methodology to overcome data security challenges in cloud. There is a drastic growth experienced in cloud computing in the recent years, thanks to its wide range of applications, flexibility and cost-effective implementation. Most of the organizations that deploy cloud technology handle their operations in a cost-effective and flexible manner. It further reduces the total cost incurred by the ownership, a highly competitive advantage for emerging users and it provides time flexibility which is much needed to achieve market objectives [2]. In spite of the business benefits rendered by cloud technology, it still poses few challenges [3]. Data residency and security of the deployed data are key concerns raised upon cloud computing. The main concerns with data residency are as follows; who holds the authority to manage data, who can access the data and in case of data breach, alternative options for data storage and rule of law to recover from data breach [4].

Data encryption and limited access rights are the key solutions to overcome data residency concerns. Data encryption is a mathematical process that converts clear text data into cipher text so that the ciphered text cannot be read by anyone other than the intended user [5]. Access rights act as a protector against external threats and the clear text data can only be accessed by the user who has the permission to access the cloud database. Encryption protects the data from internal and external threats. The proposed methodology i.e., Secure Interactional Proof System provides secure deployment of technology in any organizations to improve business performance and the collaborative solution provider for secure data sharing in cloud environment. Secure International Proof System (SIPS) focuses on security concern in cloud environment. The proposed SIPS methodology has four basic objectives as given herewith. (1) Key agent (2) Access list (3) GQ authentication protocol and (4) Key pair. The architecture proposed system is pictorial represented in Fig. 1.

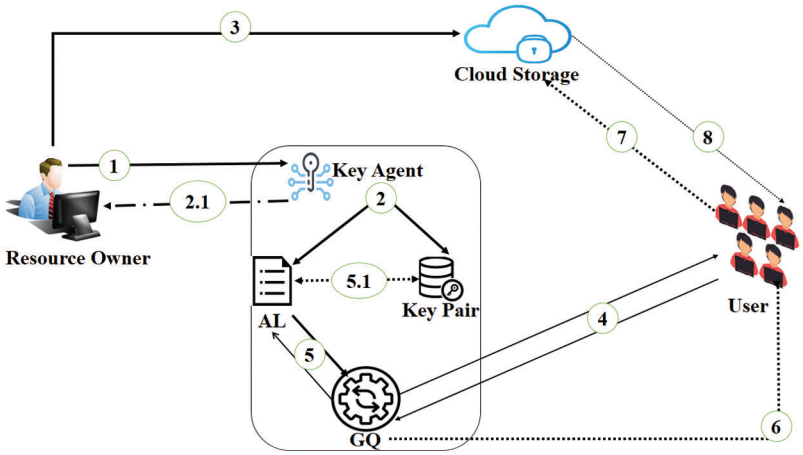

Figure 1: The system architecture

The resource owner provides a data list to key agent. Then, the key agent forwards the list as access list where a user's access rights are generated and maintained during Access listing process. The list provided by the key agent to access list tend to have a forwarded copy to key pair database as well since it helps in maintaining the key pair or security manners towards the data.

The generated prototypes of access list and key pair are exchanged to ensure the data integrity of the users. A multistep authentication protocol is followed for the users to have several verifications so as to maintain data security. These verification processes also ensure authentication and authorization of the users. Key management is also derived and monitored by Guillou-Quisquarter protocol. The direct access of the user is first ensured through multistep authentication protocol. After passing multistep

authentication, only an approved user can enter the cloud environment and can go through the data list required by the users.

2 Related Works

A number of methodologies has been proposed and implemented earlier to over the challenge i.e., to enable data security while sharing data in cloud. Ali et al. [6,7] proposed CL-PRE certificate-less proxy re-encryption scheme which is a worthy approach in those domain. In this study, the data owner shares the data to cloud in which they are mentioned as recipients. At first, the file (or) data is encrypted using symmetric data encryption key DEK by the owner itself. Then the data is stored in cloud with Access Control List (ACL) [8]. ACL contains the access rights and the names in recipient group who can access the data. In the second step, the major and important task i.e., re-encryption occurs in which the DEK is encrypted again using public key and this process enables high security for the data. The encrypted DEK is also stored in public cloud [9]. The recipient holds a private key which is developed in the form of a proxy server [10]. Proxy server in cloud considers the re-encrypted data which is sent by the data owner. Then, the re-encryption algorithm is applied to the encrypted DEK so that the decrypted recipients' private key is converted. With the help of private key, a user can download the encrypted data from cloud. For each recipient group, different DEK keys are produced to ensure confidentiality. The major advantage of this work is re-encryption key which is generated from data owner's private key and recipient's public key. Certificate-less based encryption security properties such as unidirectionality, non-interactivity, non-transistive and single use were obtained in this research that paved the way for gaining data security in cloud.

Seo et al. [11,12] conducted a research with regards to mediated certificateless encryption (or) double encryption scheme. This work was applied to achieve confidentiality and security performance in cloud. Authorization has played a vital role in increasing the applicability and success of this scheme. The researchers proposed CL_PKE scheme to overcome the existing certificateless based encryption schemes which are not only expensive in pairing operation, but also were vulnerable to decryption attacks [13]. The proposed scheme works without pairing operation for sensitive information shared in cloud. Based on access control policies, the sensitive data is encrypted using clod generated user's public keys and the data is uploaded to cloud. The cloud performs partial decryption and encryption for the authorized users [14]. In subsequent process, the user fully encrypts or decrypts the data using their own private keys. This method proved to be an efficient approach in overcoming the pairing operations. Further, certificate-less cryptography was also applied with several theorems and explanations. This scheme was established as an efficient and practical method in achieving the intended outcome.

To overcome certain drawbacks in the past two approaches, the study conducted earlier [15] implemented a special feature for advent users on cloud security through another proposal which introduced identity-based auditing for data sharing in cloud. This method promoted identity-based auditing scheme with information hiding. The method was promising in terms of hiding information to provide security. It is a different method since the scheme allows the user to share their plaintext without any encryption with researchers and makes the sensitive data go invisible [16]. To overcome the failure of previously-constructed approaches, this method implemented an identity-based auditing scheme to hide the sensitive information from malicious attackers. Integrity and authenticity were heavily achieved in this method [17,18]. A novel mechanism for sensitive information hiding was proposed with unique signing which is unique to the user. The responsibility of the manager remains the same alike computer network gateway and they possess the rights to check whether the file contains content with sensitive information. In the study conducted earlier [19], an efficient identity-based auditing scheme was proposed for shared data model to achieve high concurrency. The main aim of this approach was not to show the

sensitive data of the organization to both senders as well as the receivers. This was achieved by centralized computing tasks, which are redundant to manager and are distributed to the users. A portion of the user's private key is used to hide sensitive information, instead of selecting a random variable. The author implemented Herss's efficient identity-based signature scheme to overcome some disadvantages in this method especially during signature algorithm process [20,21]. Data processing and integrity are the major disadvantages found in this approach.

As per the review of literature, some disadvantages are found in earlier methods and are yet to be overcome such as security, integrity confidentiality, access control and authorization. The current study proposes a novel method and implements the same to overcome the challenges faced in this domain. The experimentation procedure is conducted with performance data sets and the output is discussed in detail. Following section details about the advantages of the proposed scheme.

3 SIPS Methodology

The proposed methodology that supports authentication is briefly discussed in this section. The method has the ability to store the encrypted data before it reaches the cloud and perform secure data sharing in cloud environment.

The following realities are applied in SIPS methodology.

3.1 Realities Part: I

Cloud Storage: The storage service is provided by the cloud to users. All the stored information on cloud should be secured against internal and external threats [22]. Both confidentiality and integrity of the information should be secured by storing the encrypted data in cloud [23,24]. Cloud storage in SIPS methodology plays a vital role in basic cloud operations such as data uploading and data downloading during when both data integrity and data confidentiality are heavily accomplished.

SIPS: SIPS remains the heart of the secure system that helps in bringing out the desired objectives, for instance, authentication (GQ Key Management, Key Generation and Key Pair Storage) whereas AL provides the access rights to the users. A user is required to register themselves with AL in order to obtain security service. The SIPS methodology ensures the accomplishment of secure reality for authentication. Authentication is mainly provided to avoid data loss and to ensure data integrity. SIPS can be implemented by any organization or can be maintained by a private trusted party too. However, the SIPS generates more trust in the system in organization setting.

Resource owner: Resource owner or data owner is the one who provides the data to user. The data provided by the resource owners are encrypted and stored in cloud storage. Access permission is given by the resource owner to cloud through access list. The access list contains the list of protocols for user who can access the data derived accordingly by the resource owner. The access list was maintained in SIPS methodology to qualify the access control with worthy users and to notify the user as a competent person and achieve owner satisfaction.

Users: The clients are said to be users in the cloud. For certain data, at least one client is present to access the data. A user has to get access management from AL to access the data.

3.2 Realities Part: II

Key agent: SIPS has an asymmetric cryptographic key for each resource file. The key is split into two major parts which are used for two different operations. The first process is encryption whereas the other process is decryption. The following keys are used in the proposed SIPS methodology.

Asymmetric Key $[R_k,\ U_k]$: Two large primes P_L and Q_L, generated by key agent, are selected for each key request made by the resource owner. To be secure, the recommended size for each prime, P_L or Q_L, is 512 bits (almost 154 decimal digits). This makes the size of T, the modulus 1024 bits (309 digits) to calculate $(R_k,\ U_k)$ in a step-by-step process. In first step, two unique large prime numbers such as P_L and Q_L of length 512 bits are selected in such a way that P_L is not equal to Q_L. In the next step, T is obtained by multiplying two prime numbers (P_L and Q_L) while the output is 1024 bit and the equation $\varphi(T) \leftarrow (P_L - 1)*(Q_L - 1)$ is computed. Then, select the R_k and R_k is a co-prime to $\varphi(T)$ finally calculate the 'U_k', inverse of R_k modulo $\varphi(T)$. This asymmetric key encryption for securing the data.

Key Agent/Key generation $(\boldsymbol{R_{ki}},\ \boldsymbol{U_{ki}})$: For each of the users in the group, the key agent generates $(\boldsymbol{R_{ki}},\ \boldsymbol{U_{ki}}$ such that $\{\boldsymbol{R_{ki}},\ \boldsymbol{U_{ki}}\} = \{0, 1\}$ 512. $\boldsymbol{R_{ki}},\ \boldsymbol{U_{ki}}$ serves as the portion of key agent and is used to compute (R, U), whenever a key request is obtained by the key agent. Furthermore, it is ensured by contrasting the distinct values $(\boldsymbol{R_{ki}},\ \boldsymbol{U_{ki}})$ generated for every key request.

3.3 Algorithmic Representation of SIPS Methodology

Algorithm: key generation process

Input:

AL, Key req, 512 bits;

COMPUTE:

Read P_L, Q_L|| must prime|| $P_L \neq\ Q_L$

DETERMINE T;

$T = P_L{*}Q_L$

find $\varphi(T) \leftarrow (P_L - 1)*(Q_L - 1)$

Select $\boldsymbol{Rk}$; || $1 < \boldsymbol{Rk} <\ \varphi(T)$

Calculate U_k; $||U_k = \boldsymbol{Rk}^{-1}\ mod\ \varphi(T)$

End

For each user i in AL do

$\{\boldsymbol{R_{ki}},\ \boldsymbol{U_{ki}}\}$ = {0,1} 512

Compute $\boldsymbol{R_{ki}},\ \boldsymbol{U_{ki}} \leftarrow$ each user in AL

Store $\boldsymbol{U_{ki}}$ for user i;

End for

Share $(\boldsymbol{R_{ki}})$ to Resource owner $(\boldsymbol{R_0})$

Store $(\boldsymbol{U_{ki}})$ to key pair

Return

3.4 Realities Part: 3

SIPS Design: In this part, the access for the proposed SIPS methodology is provided which secures the data in cloud among several users.

3.4.1 File Upload

There should be a secure way to protect sensitive data. This data further needs to be stored and shared among several users or in group. A key request is sent by resource owner to the key agent (KA). Figs. 2 and 3 show the processed involved in uploading a file.

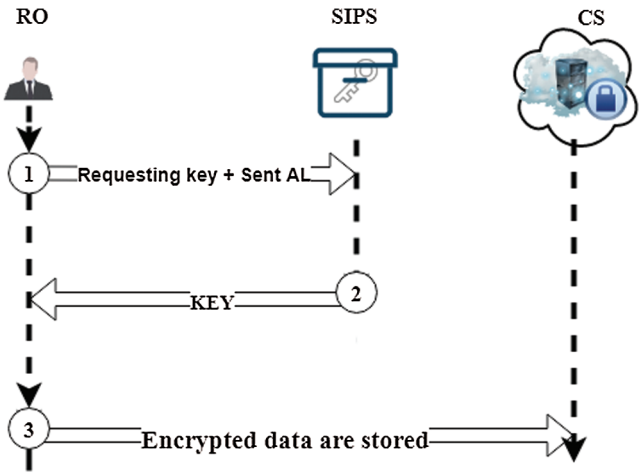

Figure 2: Upload process

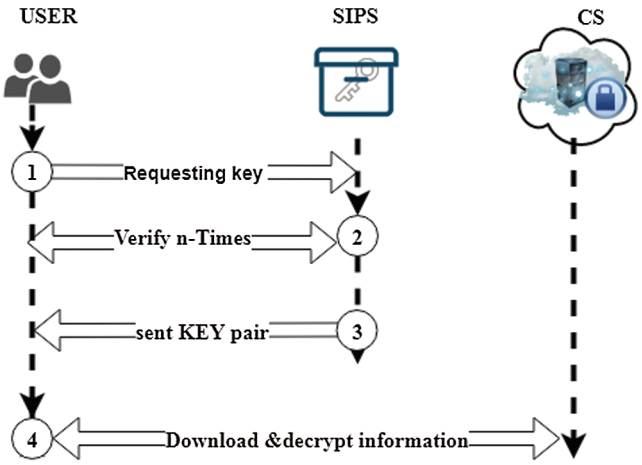

Figure 3: Download process

AL database contains the key request and access list that are granted by resource file access of the user. There are different types of access rights used by the user to access the file. There are many other constraints also can be set to get the access control over data. The key agent generates the key according to the process defined in section (iii). To generate the ACL for respective data, AL is used by KA. Resource owner, after receiving the encryption key, encrypts the data and stores the same in cloud. For each file, ACL is separately maintained. ACL holds some major information about file such as file ID, size, Owner Information (ID) and the list of user IDs with other metadata.

Decryption key is stored in keypair database. Subsequently, the key agent generates R_{ki} and U_{ki} for every user and the information is stored into AL database for later use.

3.4.2 File Download

The authorized user requests the key to decrypt the file. Before that, the user must prove its identity to SIPS. GQ (Guillou-Quisquater protocol) selects two numbers for every user Such as 'PU' i.e., public and 'SE' i.e., secret. However, in this case, the relationship between 'PU' and 'SE' is different i.e., SE r * PU = 1 mod T. The GQ constitutes three exchanges. Verification is repeated several times at a random

value of challenge between 1 and r. The user must pass several rounds of tests for verification. If a user fails in single round authentication, the process is aborted and user is not authenticated. After user authentication, they receive the session key and decryption key (U). Now, the user can download the file from cloud storage and decrypt it.

3.4.3 File Update

The method of updating the data is similar alike uploading the file in cloud. The peak difference between them is when you update the access list-related activities, the key generation activities are not carried out. When a resource owner downloads the file and make any changes, they have to encrypt the file again and store it in cloud. If the resource owner of the asset wants to change the access list, they can ask the key agent to re-generate the key pair and update the access list. Ultimately, the resource owner has the rights to add (or) delete the user against the access rights in the file.

3.5 Algorithm 2 Encryption Process (Resource Owner Side)

$(Read\_\boldsymbol{R_{ki}})$

do

{

$CT = \boldsymbol{R_{ki}}\ (Res.file) \leftarrow$ encryption:

// calculation of Res.file $\boldsymbol{R_{ki}}$ mod n;

WRITE_[CT]

//Upload the CT in cloud

End

}

3.6 Algorithm 3 Decryption Process [User Side]

(GET_$\boldsymbol{U_{ki}}$,ID)

do

{

PT = $\boldsymbol{U_{ki}}$ [CT file] ← decryption:

READ_[PT]

End

}

4 Discussion on Sips

The SIPS methodology is proposed in this study to provide the following services for electronic records.

- Authorization and Integrity
- High Confidentiality
- Secure data sharing among the group
- Secure data from unauthorized access
- Provide Access control to the user.

The following discussion briefly describes the working principle of SIPS methodology and how the service are achieved. The proposed methodology has a few main components such as Access List, Key Agent and GQ Protocol. These components act as Secure Interactional Proof system that enables its users to interact securely in cloud.

4.1 Access List

Access control is provided to the user based on the access list. This access list plays an important role by mutually interacting with Gulliou-Quisquater Protocol and ensuring the access of data for the user in cloud. The access list is generated and provided by the resource owner who shows the users' authorization. The ultimate goal of access list is to provide the access to cloud information only to correct users (or) authorized users. Access rights are provided by data owner to the authorized users.

4.2 Key Agent/Generator

The goal of the key agent is to generate keys. A pair of keys is generated using the key pair data that is encrypted and stored securely in cloud. Data confidentiality and data integrity are achieved through this encryption method.

4.3 GQ Protocol

GQ protocol is a multi-authentication protocol which verifies the user in multiple steps. Through multistep authentication, the fraudulent users can be get rid of. It is an identification protocol that provides authentication by processing 'n' number of rounds.

- Authentication system through GQ Protocol

For user authentication process, GQ protocol enables numerous rounds in SIPS.

- One-time setup:

SIPS chooses two unique primes i.e., S and R and generates a T = SR module.

SIPS specifies a public variable i.e., $P_U > 4$, with gcd $[(P_U, (S-1)(R-1)] = 1$ in order to allow SIPS to measure the security $S = {P_U}^{-1} \bmod (S-1)(R-1)$

The parameters are defined by SIPS.

- Selection of parameters for each user

Each user has a unique identifier ID(A) that can be used in the determination of value J(A) = f(Id(A)) mod n. [Redundant identity]

SIPS offers private data to each user which can be determined using $(A) = J(A)^{-S}$.

- Protocol:

The user proves their identification to SIPS using 'N' rounds. Each of them is composed of the following elements.

i) A user chooses a random private R_P and sends ID (a) and $X = R_P P_U \, mod \, T$ to SIPS
ii) SIPS chooses a randomized challenge in $\{1, 2 \ldots\ldots\ldots\ldots r\}$
iii) The user calculates and replies to SIPS: $Y = R_P \, private \, (user)^e \, mod \, T$.
iv) SIPS collects, Y, constructs J(User) = f(id (user) mod T calculates

$Z = J(user)^e 7 P_U \, and \, f \, Z = mod \, T$ accepts and authenticates the user.

4.4 Algorithm: The user Authentication Process

Input: User request to SIPS for key

Begin_GQ;

For Authenticate do

Obtain Unique Prime S,R;

SET public variable; Gcd

$[(P_U,\ (S-1)\ (R-1)] = 1 \leftarrow P_U > 4:$

Determine ¥ $= P_U{}^{-1}\ mod\ (S-1)\ (R-1)$

Write_Parameters:

End

Initialization:Parameters for each_user;

$Id(user) -> unique\ identifier$

Compute $J(user) = F(Id(user)) mod\ T$

Write Each user;

$User - j(user)^s\ private$

End;

For proves identification

Each user do

While (ID=True)

Initialize $R_P; \leftarrow Private$

$Send\ Id(U)\ ||\ X = R_P\ P_U mod\ T \leftarrow SIPS$

SIPS do

Challenge e

Calculate y; ← *by user.*

Reply $y = R_P$. *Private* $(u)^e\ mod\ T$

Get y;

Compute $J(u) \leftarrow F(Id(u)\ mod\ n.$

Then $Z = J(u)^e Y^Y$

if $Z = Xmod\ T$ *Then*

Accespts User :

Else

Terminate process

End while

5 Formal Analysis

Time Net is a software which is used in modelling and analysis of Stochastic Petri Nets (STPN). The following section briefly introduces STPN prior to discussion of the analysis.

5.1 Stochastic Petri Nets (STPN)

Time Net tool is used in the evaluation of STPNs in which the transition firing times can be exponentially distributed. Graphical User Interface (GUI) is used to specify the models and the results are defined with special purpose syntax. Both continuous and discrete time scale models are supported in this method.

The analysis is conducted based on Markov regenerated theory. The supplementary variable method is used for transient analysis. This tool provides different techniques for simulation experiments.

5.2 Analysis Theme of STPN's

It consists of five tuples $STPN = (S, T, R, M_0, \lambda)$ where P denotes a set of states and is said to be places. T denotes a set of transitions, $\boldsymbol{R}$ where $R = (S*T)\ U\ (T*S)$ is a flow relation set called as arc. $\boldsymbol{M_0}$, is denoted as initial marking. λ is the firing rate array λ which is associated with transition. The function $\lambda(\boldsymbol{m})$ denotes the firing rate of the random valuable for current marking. STPN's reach ability graph can be directly mapped to Markov properties. Each state of the graph is relatively mapped with the state of Markov process. Firing state λ of the graph is correspondingly equal to Markov state transition with λ probability.

→ Step 1: The key agent generates asymmetric key (i.e.,) K. The following formula is generated on transition gen_key to describe the process

$$SIPS\ (gen\_key) = select\ [P_L, Q_L] \mid\mid cal\ [T] \mid\mid R_K = gcd[\varphi(T, R_K)] \mid\mid U_K = {R_K}^{-1}\ mod\ \varphi\ (T) \tag{1}$$

→ Step 2: This process is further carried out to next level of encryption. The data owner encrypts the file (F) which is then uploaded to the cloud in a secured manner.

$$R_0\left(En_{file}\right) = CipherText||R_{Ki}(Res.file)|| \tag{2}$$

→ Step 3: The key agent generates a pair of keys in which one key is shared to the user and another key is stored in key pair.

$$SIPS\left(gen_{key}\right) = key\ pair(R_K, U_K)||R_K - -Share\ (R_0)||U_K - -Store\ (user) \tag{3}$$

→ Step 4: At most of the times, authentication is the primary step in this approach. This is performed using Guillou Quisquarter protocol (GQP). This protocol conducts the multistep authentication process using a one-time setup by choosing several parameters such as User ID, file name etc., which creates data privacy. The steps involved and the procedure are discussed in detail under section (iii).

$$SIPS(Authen\_user) = F(Id(user))mod\ T||\ J(user)^{-s}\big|\big|\ X = R_{P,}\ P_{U,}\ mod\ n\big|\big|$$

$$Y = {R_P}^{private}(user)^{R_K} mod\ T \tag{4}$$

→ Step 5: After uploading the data and successfully achieving the authentication, next step is processed on user side i.e., file downloading which is often referred to decryption process. The following formula relates the downloading process. The key generation receives a decryption request from the user. After verifying the authentication and authorization status of the user by key generation using GQP, the key is figured out based on predefined steps. Key generation decrypts the data and replies the user. This is to ensure privacy and to secure the generated keys, it is deleted subsequently.

5.3 Properties for Verification

- Unauthorised users are not allowed
- An authorized user from cloud cannot generate a valid key by acting as another user and granting a random key.

- An authorized user can access the data by generating a valid key which is contributed by key manager (or) generator.
- A malicious party cannot access the data since the proposed methodology is highly secure and authenticated.

6 Performance Evaluation

6.1 Experimental Setup

To specify the performance of the proposed SIPS methodology, the curren approach was implemented using code Dx which provides a set of correlated results. The main goal of code Dx is to prioritize and manage attacks. It is an interactive visualization of the metrics which is highly required for the current scenario since it covers features such as system security, authentication and also the integrity of the data. The main protocol used in code Dx API uses a REST-full design built on HTTP such as GET, POST, and DELETE etc. HTTP 200 ok is used to communicate status in the server. Authentication relies on passing an API-Key whereas HTTP header is present in all API requests. HTTP 403 Forbidden is used as a request header for any invalid users or invalid point which is generally returned as an empty response.

E.g.: The output of the API-Key Header look as follows

API-Key: 650e8300 – e286 – 40d4 – a617 – 557744550000

In general, the UUID's are used to generate API keys in code Dx. To upload data, a new analysis in code dx is as follows.

POST/api/project/:pid/analysis

All the cryptographic operations were implemented in RSA Algorithm.

6.2 Result Analysis

The proposed SIPS methodology was evaluated under different scenarios.

1) Key generation

The asymmetric keys were generated for every file as discussed earlier. Key sharing was done separately for every user. The proposed SIPs methodology was evaluated with specific reference to time taken during key generation.

The researcher analyzed the consumption of time for different number of users. The set of users considered were 20, 40, 60, 80, and 100. Fig. 4 shows the results attained i.e., time consumed to generate keys. With increase in the number of users, the time consumed for key generation. It is to be noted that the increase in time consumption is not uniformly proportional to the increase in the number of users. The time consumption did not increase alike the increase in the number of users. A slight decline was observed at the time of data submission.

6.2.1 Encryption and Decryption

The researcher analyzed the time taken for encryption and decryption processes with varying data (or) file size.The file size used were 1, 10, 50, 100, 500 MB. As defined earlier, key generation plays a vital role in this methodology before encrypting and decrypting the data. The time required for key generation was compared with total encryption and decryption times. The main purpose is to check and maintain the overhead of key computation across the total number of encryption and decryption processes. Figs. 5 and 6 shows the results attained from the analysis of encryption and decryption. The figure shows the expected time for encryption and decryption processes with increase in file size. This shows that the proposed SIPS methodology was highly helpful in maintaining the computational time. The results

inferred that the time was almost constant with negotiable change occurring during processing. The comparative analysis results infer that the small-sized file had high percentage of key computational time compared to total encryption time. As per the comparison made, 200 kb file size took 15% high computation time than the total encryption time. Though the file size increased to 2 MB, it reduced the time proportion to 10%. When the file size was increased up to 20 MB, the percentage of time consumption got reduced to 4%. With 1000 MB file size, the percentage of time consumption remained 0.54%. It is to be noted that the overall key computation time was in the ragne of 0.010 and 0.015 s. The decryption results were also in line with the trends observed in encryption process. The major percentage of key computation was in the range of 15% in case of 200 kB and 2% in case of 1000 MB file size.

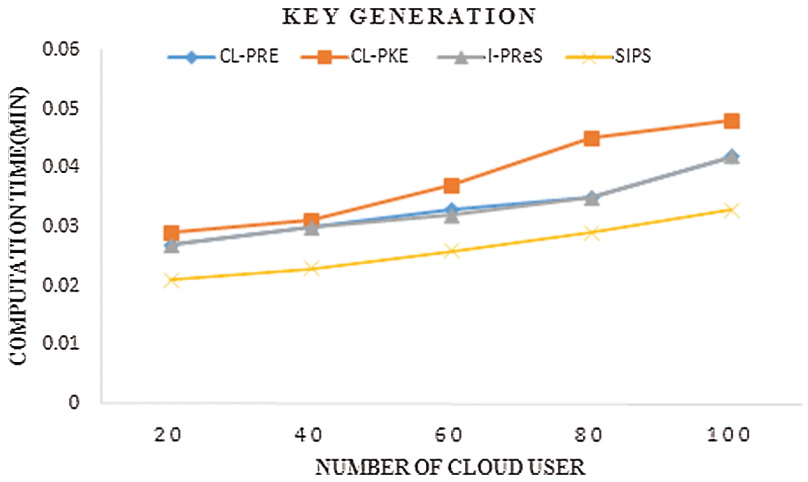

Figure 4: Computation time for key generation

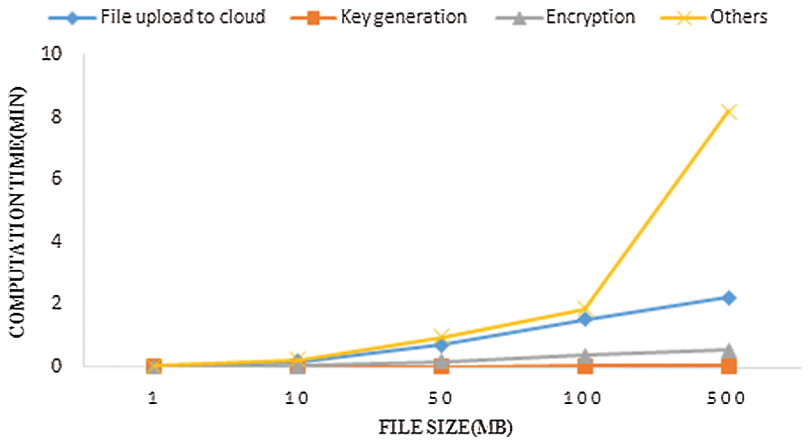

Figure 5: Performance of file uploads for SIPS

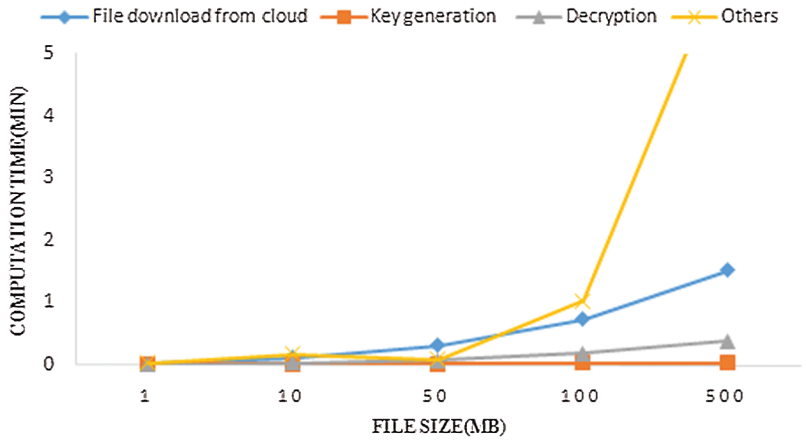

Figure 6: Performance of file download for SIPS

6.2.2 File Upload/Download

The researcher evaluated the proposed SIPS methodology for total time consumed to upload and download a file from cloud. The following times were taken into consideration to perform the above-mentioned scenario.

1) Key generation time

2) Encryption/Decryption time

3) Uploading/downloading time

4) Time request for data submission.

Fig. 4 shows the results for time taken to upload the data. Fig. 5 shows the results for downloading the data from cloud followed by subsequent decryption process. The time consumed for both uploading and downloading the data was same. Tab. 1 represents the comparison of key generation times and Tab. 2 compares the turnaround times. The proposed SIPS methodology was compared and show in Fig. 4 for key generation, Fig. 6 for file uploading and Fig. 5 for file downloading. These comparisons were based on time consumption during key generation and turnaround time taken for both encryption and decryption processes. To conclude, the comparison reveals that the SIPS methodology performed far better than other techniques due to small overhead time.

Table 1: Comparison of key generation times

No. of user	CL-PRE	CL-PKE	I-PReS	SIPS
20	0.027	0.029	0.027	0.021
40	0.03	0.031	0.03	0.023
60	0.033	0.037	0.032	0.026
80	0.035	0.045	0.035	0.029
100	0.042	0.048	0.042	0.033

Table 2: Comparison of turnaround times

FS(MB)	CL-PRE		CL-PKE		I-PReS		SIPS	
	FUL	FDL	FUL	FDL	FUL	FDL	FUL	FDL
1	0.03	0.023	0.034	0.025	0.048	0.031	0.028	0.02
10	0.218	0.165	0.249	0.165	0.243	0.174	0.178	0.126
50	0.895	0.558	0.976	0.593	1.006	0.098	0.719	0.312
100	1.662	0.952	1.874	0.986	2.586	1.027	1.531	0.741
500	6.162	3.588	8.201	3.83	14.535	6.67	2.225	1.522

Descriptions for the table are following.

FS = File Size, FUL = File upload, FDL = File download.

7 Conclusion

The current study proposed and designed a novel methodology i.e., SIPS for secure data sharing in cloud. The proposed methodology has the ability to achieve data confidentiality, authentication, authorization, integrity and perform secure data sharing without double encryption process. The main aim of the proposed methodology is to ensure access control for the data so as to avoid malicious attackers. Moreover, the SIPS methodology assures the integrity of the data in case if it is unmodified. Both encryption and decryption processes were performed with the help of key generator that acted as a trusted third party in SIPS methodology. The proposed methodology can also be implemented in mobile cloud computing. The working of SIPS was formally analyzed using STPN and Code Dx. The performance was evaluated based on time consumption during three scenarios such as key generation, uploading and downloading the data from cloud. The results infer that the proposed SIPS methodology can be implemented in cloud for secure data sharing. In future, the proposed model can be incorporated in real time application areas. Besides, the presented model can be extended to the use of light weight cryptographic techniques.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Rady, T. Abdelkader and R. Ismail, "Integrity and confidentiality in cloud outsourced data," *Ain Shams Engineering Journal*, vol. 19, no. 2, pp. 275–285, 2019.

[2] J. M. M. Perez, G. M. Perez and F. Skarmeta, "SecRBAC: Secure data in the cloud," *IEEE Transaction on Services Computing*, vol. 10, no. 5, pp. 726–740, 2017.

[3] K. Karthikeyan, R. Sunder, K. Shankar, S. K. Lakshmanaprabu, V. Vijayakumar *et al.,* "Energy consumption analysis of virtual machine migration in cloud using hybrid swarm optimization (ABC-BA)," *Journal of Supercomputing*, vol. 76, no. 5, pp. 3374–3390, 2020.

[4] Y. Fan, Y. Liao, F. Li. and S. Zhou, "Identity-based auditing for shared cloud data with efficient and secure sensitive information hiding," *IEEE Access*, vol. 7, pp. 114246–114260, 2019.

[5] N. Agrawal and S. Tapaswi, "A trustworthy agent-based encrypted access control method for mobile cloud computing environment," *Pervasive and Mobile Computing*, vol. 52, pp. 13–28, 2019.

[6] M. Ali, R. Dhamotharan, E. Khan, U. Samee, U. Khan *et al.,* "Secure data sharing in clouds," *IEEE Systems Journal*, vol. 11, no. 2, pp. 395–404, 2017.

[7] L. Xu, X. Wu and X. Zhang, "CL-PRE: A certificateless proxy re-encryption scheme for secure data sharing with public cloud," *ACM Sym. on Information, Computer and Communications Security*, Korea, pp. 87–88, 2012.

[8] I. E. Ghoubch, R. Abbou and F. Mrabti, "A secure and efficient remote data auditing scheme for cloud storage," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 5, pp. 1–7, 2019.

[9] S. Aldossary and W. Allen, "Data security, privacy, availability and integrity in cloud computing: Issues and current solutions," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 4, pp. 485–498, 2016.

[10] Y. Sun, J. Zhang, Y. Xiong and G. Zhu, "Data security and privacy in cloud computing," *International Journal of Distributed Sensor Networks*, vol. 10, no. 7, pp. 1909–1913, 2014.

[11] S. Seo and M. N. D. Bertino, "An efficient certificateless encryption for secure data sharing in public clouds," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2107–2119, 2013.

[12] K. Xue, W. Chen, J. Hong, W. Li and P. Hong, "Combining data owner-side and cloud-side access control for encrypted cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2062–2074, 2018.

[13] J. Wei, W. Liu and X. Hu, "Secure and efficient attribute-based access control for multiauthority cloud storage," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1731–1742, 2018.

[14] S. Xu, G. Yang, M. Yi and R. H. Deng, "Secure fine-grained access control and data sharing for dynamic groups in cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2101–2113, 2018.

[15] S. Sicari, A. Rizzard, L. A. Grieco and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[16] A. Muthumari, J. Banumathi, S. Rajasekaran, P. Vijayakarthik, K. Shankar *et al.,* "High security for de-duplicated big data using optimal simon cipher," *Computers Materials & Continua*, vol. 67, no. 2, pp. 1863–1879, 2021.

[17] S. Shafeeq, M. Alam and A. Khan, "Privacy aware decentralized access control system," *Future Generation Computer Systems*, vol. 101, pp. 420–433, 2015.

[18] D. He, N. Kumar, M. K. Khan, L. Wang and J. Shen, "Efficient privacy-aware authentication scheme for mobile cloud computing services," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1621–1631, 2018.

[19] Y. Liu, J. Ryoo and S. Rizvi, "Ensuring data confidentiality in cloud computing: An encryption and trust-based solution," in *2014 23rd Wireless and Optical Communication Conference (WOCC)*, Newark, NJ, USA, pp. 1–6, 2014.

[20] S. Zhou, R. Du, J. Chen, H. Deng, J. Shen *et al.,* "SSEM: Secure, scalable and efficient multi-owner data sharing in clouds," *China Communications*, vol. 13, no. 8, pp. 231–243, 2016.

[21] A. N. Khan, M. L. M. Kiah, A. Sajjad, M. Ali, A. R. Khan *et al.,* "Incremental proxy re-encryption scheme for mobile cloud computing environment," *Journal of Supercomputing*, vol. 68, no. 2, pp. 624–651, 2014.

[22] J. Singh, T. Pasquier, J. Bacon, H. Ko and D. Eyers, "Twenty security considerations for cloud-supported internet of things," *Internet of Things Journal*, vol. 3, no. 3, pp. 269–284, 2016.

[23] V. Manikandan, M. Sivaram, A. S. Mohammed, V. Porkodi and K. Shankar, "Secure localization based authentication (SLA) strategy for data integrity in WNS," *Computers Materials & Continua*, vol. 67, no. 3, pp. 4005–4018, 2021.

[24] A. F. S. Devaraj, M. Elhoseny, S. Dhanasekaran, E. L. Lydia and K. Shankar, "Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments," *Journal of Parallel and Distributed Computing*, vol. 142, pp. 36–45, 2020.