Security Enhancement in Cloud Environment using Secure Secret Key Sharing

Sakshi Chhabra, *Member, IEEE* ,and Ashutosh Kumar Singh, *Senior Fellow, IEEE*

Original scientific paper

***Abstract*—Securing the data in distributed cloud system is considered one of the major concern for the cloud customers who faces security risks. The data leakage or data tampering are widely used by attackers to extract the private information of other users who shares the confidential data through virtualization. This paper presents Secure Secret Sharing (SSS) technique which is being recognized as one of the leading method to secure the sensitive data. It shares encrypted data over cloud and generated secret key is split into different parts distributed to qualified participants (Q_n) only which is analyzed by malicious checkers. It verifies the clients based on their previous performances, whether these users proved to be authorized participant or not. The key computation is evaluated by the Key handler (KH) called trusted party which manages authorized control list, encryption/decryption and reconstruction of key shares. The Lagrange's interpolation method is used to reconstruct the secret from shares. The experimental results shows that the proposed secure data sharing algorithm not only provides excellent security and performance, but also achieves better key management and data confidentiality than previous countermeasures. It improves the security by using secure VM placement and evaluated based on time consumption and probability computation to prove the efficacy of our algorithm. Experiments are performed on cloudsim based on following parameters i.e. time computation of key generation; response time and encryption/decryption. The experimental results demonstrate that this method can effectively reduce the risks and improves the security and time consumption upto 27.81% and 43.61% over existing algorithms.**

***Index Terms*—Secret Sharing, Cloud Security, Cloud Computing, Malicious attacker analysis, Secret Splitting.**

I. Introduction

IN the field of cloud computing and Internet of Things (IOT), latest developments have increasingly changed the way of computing as well as the notion of computing resources [1]. According to a recent report, 94% of companies expects more than half of their workloads to be in the cloud and IOT is envisioned to grow rapidly due to the proliferation of communication technology [2]. The rapid development of applications has increased the traffic rate exponentially in the cloud data center networks. Today being an emerging approach, these techniques [36] has attracted significant attraction as a means of reducing the capital investment and increasing the system efficiency [23] [32]. It stores user data to a large virtual storage system comprising of multiple servers on a network. The storage of data at remote locations has many benefits, but there is always a risk of alteration, leakage or regeneration of the private data. Because every user expects confidentiality, integrity and authentication of the message which is sent by them through the computer [15]. As IOT plays the role of a data source unit, IoT security is an area of concern in order to safeguard the hardware and the networks in the IoT system. The main security issues includes verification of eligible user's credentials, insider threats or vulnerabilities, malware attacks, external connectivity to an organizational network, lack of skilled security workforce and many more. To overcome these risks, the data owners demand the high level of protection and provides the effective security measures from the cloud servers. This security can be provided by some effective methods and practical security algorithms [3]. So our goal is to make the cloud secure and to minimize these security threats so that the cloud servers can be used trustfully [5], [17], [19].

This paper presents secure key management framework which ensures the minimization of these attacks and reduce the data leakage along with efficient use of computing resources. The SSS model not only facilitates the cloud consumers by data confidentiality and key computation time, response time etc. but also maximizes the data integrity and prevents the data leakage attacks. These attacks can be averted by blocking the malicious VMs if it can be identified prior to their execution. But it is so difficult to achieve it in real time scenario. So, malicious checker is introduced which identifies the authorized or unauthorized clients based on their previous actions. It analyzes the historical interpretations of the clients based on their performance and leakage considerations that how much % of clients are trustworthy. The emphasis of this paper is to reduce the possibility of attacks among different users. It surely reduces the information leakage and improves the data confidentiality in the cloud architecture. It achieves more security and privacy concerns named as a Secure Secret Sharing in Clouds that deals with the aforementioned security requirements of shared group data within the cloud. The problem that we have considered in this paper can be outlined as follows:

A. Our Contribution

The paper is introduced with theoretical models to more accurately quantify and explain the problem of secret key

Manuscript received December 7, 2019; revised May 22, 2020. Date of publication July 20, 2020. Date of current version July 20, 2020. The manuscript has been submitted in Special Issue on Internet of Things: Hardware and Software Solutions

Authors are with the Department of Computer Applications, National Institute of Technology Kurukshetra, Haryana, INDIA email: sakshichhabra555@gmail.com and ashutosh@nitkkr.ac.in

Digital Object Identifier (DOI): 10.24138/jcomss.v16i3.964

sharing, especially when more participants are present in different locations with varied mechanisms of safeguards. It includes:

- The model is defined with security metrics for assessing the attacks and evaluate these metrics quantitatively by encryption/decryption, file size, key time computation where eligible members can recover the secret by merging their shares together.
- A security model which utilizes the probability methods to minimize the likelihood of these attacks by sharing the key among a group, exercise the access control, and manage the secret keys in an effective manner to safeguard the data security [35] and confidentiality.
- The idea of selecting the secure virtual machine is employed by placing the encryption as well as secure VMs [19] to access these significant reliable indices. It ensures the high execution efficiency, secure key management and optimized the external service performance. Implementation, test, and verification of the effectiveness of policy on the popular platform Cloudsim.
- In the case of new participant addition, it can only be entered by the permission of the key owner and that key holder will give access rights to a new user like read only, write only or read-write both. Same in the case of quit group participants, the key owner will eliminates all the records of the departing user from the related files.

The rest of the paper is organized as follows: In section II we discuss the related work, it identifies the key issues of cryptography based on the secret sharing framework. In section III, secure secret sharing framework is proposed based on key sharing, distribution and reconstruction. The detailed description of the malicious checker analysis for the purpose of finding authorized or unauthorized clients based on historical observations is given in section IV. An experimental setup and analysis are presented on the basis of computation time and probability in section V followed by conclusion.

II. Related Work

There are several techniques that have been proposed by various authors related to single and multi-cloud security using secret sharing key algorithms [16] [25] and a few of them are explained. In a multi-cloud strategy, it associates the use of two or more cloud services to minimize the risk with separate instances that run parallel to one another in a cloud computing environment. Distributed Cloud is one of the cloud application that interconnects the data and applications are to be served from the multiple geographic locations [26]-[29]. Doyel Pal et al. [5] introduced a threshold secret sharing scheme widely used mechanism to secure different computing environment and to enhance the security of secret key in a distributed cloud environment. The whole information is shared among multiple systems which may also be in different locations. However, the computation time and response time are higher than other approaches. Cas Cremers et al. [32] presented the secured key technique with ISO/IEC 11770 standard method and achieves better analysis of the protocols and uncovers several incorrect claims. It solves the problem of key agreement, key establishment, and key transport protocols effectively. But it has not implemented on cloud and didn't talk about response time, cost etc. Chenyutao Ke et al. [10] discussed two-threshold secret sharing scheme in order to enforce a new type of cross-group policy. A method of malicious provider may recover secret data illegally through manipulation on servers that holds enough shares to recover the secret. However, this approach is not aware about effective key management. Chen and Tzeng et al. [4] designed a policy based on the shared key derivation method for securing the data sharing among a group . This methodology uses a binary tree for the computation of the keys. However, the computational cost of the proposed scheme is high as the rekeying mechanism is heavily employed in the proposed scheme. M. Breezely George et al. [31] introduced the aggregate key after losing the key with which it is impossible to access the data. Therefore, to solve this problem, they proposed a novel technique to utilize the key sharing with proper security, such as Triple DES algorithm and Elliptic Curve Cryptography (ECC). The Triple DES algorithm is used for file encryption and decryption process. A Private information retrieval (PIR) is one of the encouraging security primitive that assures the privacy of user's interests. It is because of PIR technique confessed to access any data from database server without having the knowledge of the server through which the information has been accessed at that time. Now in this ramp secret sharing approach states that it needs low communication cost, which results in less expenditure and/or better quality of service compared with what may be achieved if traditional information-theoretic PIR and anonymizers are used [13]. Felix Günther et al. [30] announced a notion based on solving the key distribution problem of signature schemes. This linkable message tagging scheme (LMT) helps to identify the pair of messages and accompanying these authentication tags as related if and only if these tags are created using the same secret key. Xiao-Fen Wang et al. [33] proposed the idea of data sharing and shows how to securely and flexibly search and share cloud data among a group of users without a group manager. Md Kausar Alam et al. [15] introduced the uses of multi-cloud and data security, to reduce the security risks and its affect on cloud users by using a Shamir's Secret sharing algorithm. The security and performance of a secret sharing scheme based on storage requirement, the time consumed for splitting and recovering the data analyzed by Aisha Abdallah et al. [24]. It also reduces the required capital investment to store the share and ensures the security of each secret's share. An information theoretic approach to Secret Sharing is suggested by Shaofeng Zou et al. [21]. This problem has been solved by compound wiretap channel. The control of this approach is further demonstrated by MIMO (Multiple input Multiple output) with layered decoding and secrecy constraints. Many researchers had worked out in this area, but at rest we want more security to store the sensitive or confidential data. In proposed SSS framework, it is based on cryptographic algorithms and malicious checkers and makes this methodology a trivial mode. Moreover the access controls are decided by the key owners to a portion of the key that disallow insiders initiate individuals or coordinated attacks on the data, Table-1 gives a summarized comparison of work done in Secure Secret Sharing techniques.

TABLE I. SUMMARIZED VIEW OF STATE-OF-ART WORK DONE IN SECRET SHARING TECHNIQUES

Cloud	Computing techniques	Control	Splitting the keys
Distributed Cloud	BOINC,SET, PlanetLab	Resource Selection and Encryption	Multiple shares [5]
	Map Reduce Secure functions and HMAC	Secure data transmission and Integrity	Pieces of processes and assigns to mappers and reducers [9]
Public Cloud	Ramp secret sharing, Ray casting, Modular prime Operations	Security and Privacy Concerns	Splitting into three color components, i.e. red, green, blue [6]
	Bilinear pairing and BDH	Security, Data confidentiality protection and Key management	Public and Private key generator (PKG) [8]
	Shamirś Secret Sharing, SHSS, Information-Theoretic Security	Security and Availabilityt	Two-threshold (t, m) - (k, n) secret [10]
	Query processing, e.g. equality, range, aggregation, projection, joins	Confidentiality and Privacy	Multiple partitioning methods [11]
	Triple DES algorithm, Elliptic Curve cryptography	Authorization, Data availability, Accessibility	Splitting in two ways: Symmetric and Asymmetric [13]
Hybrid Cloud	Probabilistic model, data segregation, Reverse engineering	Confidentiality, Cloud storage security	Splitting into optimal chunks [12]
Multi-Cloud	Homomorphic algorithms, Private information retrieval (PIR)	Confidentiality, Integrity and Availability	Multiple shares (Deepsky Architecture) [7]

All reseachers mentioned above worked against the secret sharing and key management problems. Though different from the previous studies, we focus on a real-time application request types with secure VM placement and encryption/decryption of different file sizes. The model shares the data among secure VMs in the cloud for effective data confidentiality is evaluated based on time consumption during the key generation, response time of uploading/downloading to the cloud. Through analysis of our proposed algorithm, the SSS scheme is found suitable and guarantees that data among virtual machines is well secured and comparatively superior to the above works.

III. SECRET SHARING FRAMEWORK

A. Mathematical Definitions

A secret sharing scheme (SSS) is a tuple $(S_k, P, R_K, Q_n, NQ_n, K, t, S_k, S_H, SH, RE)$ consisting of [16]:

- A positive integer S_n, called the number of senders;
- A finite set P, whose elements are called secrets;
- A positive set R_n, called the number of receivers;
- A finite set Q_n with $|Q_n| \geq 2$, whose elements are called qualified set;
- A finite set NQ_n, whose elements are called non-qualified set;
- A finite set K, called the number of participants.
- A threshold value t, where at least 85% part of the secretś shares recovers the whole
secret value.
- A positive integer S_K, called the secret key which we shared among shareholders.
- A positive integer S_H, called the number of shares in the secret.
- An algorithm SH (usually probabilistic), called the sharing algorithm, that takes as input a secret $s \in P$, and outputs a vector of n shares; and
- An algorithm RE, called the reconstruction algorithm that takes as input a vector and outputs either a secret or $\perp$. Here, $\perp$ is a fixed symbol, not contained in $P \bigcup Q$ that represents a missing share in the input, and failure to reconstruct the secret in the output.

Definition 1: *(Secret Sharing). Let S_i be a finite set of secret, where $|K| \geq 2$. The distribution algorithm takes the secret key and splits into $(K-1)$ shares. A k-out-of-n secret sharing threshold value SS= $(t(85\% of n), n)$ is called (t, n) realizing an access structure A called qualified sets $Q_n = \{1, 2, \ldots, n\}$. The reconstruction algorithm assembles all the shares of key with the assistance of Lagrange's interpolation method, i.e. $f(x) = \sum_{i=1}^{k} y_i L_j(x_i)$.*

According to the definition 1, we encrypt the data with a single secret key to counter the insider threats. A secret key is to be split into k shares, where k should be more than 2. This sharing algorithm takes secret S as an input and produces a set of n shares i.e. $(s_1, s_2, \ldots, s_n)$. These

TABLE II. DESCRIPTION OF NOTATIONS TO BE USED IN OUR SCHEME

Notation	Definition
S_n	No. of Senders
K	No. of Participants
T	Threshold value
Q_n	Qualified set
SK	Secret Key
NQ_n	Non-Qualified set
KH	Key Handler
R_n	No. of Receivers
D_n	No. of Data
$\Re$	Authorized Structure
$gn(x)$	Polynomial degree x in the range of n
$l_v n$	Degree of polynomials
v_n	No. of vectors
S_P	Prime Numbers
$MemBuf$	Memory Buffer
SYN/ACK	Synchronization/Acknowledgement
Per_{CL}	Performance of every client
Non_{GLT}	Recognition of Non-Guilty Agent
Key_{AVL}	Key available on mentioned time
NM_{ANLY}	Non-Maliciousness Analysis

shares are on the receiver side $(s_1, s_2, \ldots, s_t)$ where t is a threshold value in which 2 $\leq$ t $\geq$ n which retrieves the secret key S_k if these shares are authorized otherwise halts. The number of shares involved in this reconstruction should be at least 85% of n of the secret, while shares with less than t will not reveal any secret. This merging of shares is done with the help of Lagrange's interpolation method [24]. This methodology presents confidentiality, secure key sharing among groups and secure data from unauthorized access of valid insiders within the group.

Definition 2: *(Authorized structure $\Re$) A set of participants $\wp = \{1, 2, \ldots, K\}$ to share a secret key S_K with K shareholders (Qualified sets Q_n) of the following: (1) A secret S_K which uniformly distributed over authorized sets $\Re$. (2) An encoder $g : W \rightarrow K^n$ mapping each secret 's share with a codeword. (3) A decoder interprets the shares with assistance of Key Handler (KH).*

Note that, in the authorized structure, there are number of qualified participants $\{Q_1, Q_2, \ldots, Q_n\}$ which are eligible for preservance of share the secret wisely by the analysis of non-malicious checker test i.e. $((Per_{CL} + Non_{GLT} + Key_{AVL})/3)$. Because shares should be distributed to the only authorized sets and secret key is encrypted that helps in mapping of each share with some secret code. Key Handler collects the key 's parts from the qualified participants, then reconstruct from an authorized set of shares and decode the keys and pass to the secure cloud. Table-2 demonstrates the description of notations which we utilized in our paper.

B. Key Management

Suppose a cloud is composed of multiple Senders $S_n = \{1, 2, \ldots, i\}$, Receivers $R_n = \{1, 2, \ldots, i\}$ and amount of data $D = \{D_1, D_2, \ldots, D_n\}$. We have to send the data from sender to the clients by using the secure secret sharing technique for protecting this sensitive data. Let there be any sender and a set of shareholders $\wp = \{1, 2, \ldots, K\}$. The sender has a secret key S_K which wants to share that secret information with K shareholders. There are qualified sets $\{Q_1, Q_2, \ldots, Q_n\}$ and non qualified sets $\{NQ_1, NQ_2, \ldots, NQ_n\}$ at the receiver side. We cannot give the shares of that secret to everyone. Out of n clients only k clients are under qualified sets on which we can trust and give the shares confidently. Those qualified sets Q_n should be specified by an authorized structure and should pass the test of maliciousness i.e. that user is legitimate or not that preserves the secret's parts. Thus, Secret Sharing key Framework assures that the qualified sets of participants can decrypt the secret key while non qualified set of participants cannot recover the secret key (S_K). We define an authorized structure $\Re$ which includes all subsets of $\wp$ that are required to reconstruct our secret key. The set $A \in \Re$ is defined as qualified set. In secret sharing scheme, we involve that if the shareholders are in the qualified set (Q_i) $A \in \Re$ collect their information together with a negligible error probability. We define a non-authorized structure ω , which does not include any subset of $\wp$ which are required to reconstruct our secret key. The set $B \in \omega$ is defined as non-qualified set. In secret sharing scheme, we can involve the shareholders of $NQ_n B \notin \Re$. Even if they collect their corresponding information together but they cannot obtain that secret.

We can say, $\omega = \Re^C$. In our scheme, the main objective is to make the secret key inaccessible to the unauthorized users. The idea of a secret sharing scheme is what we can say that two points are sufficient to define a line, three points are sufficient to define a triangle, four points are sufficient to define a rectangle. An authorized structure contains all set with the threshold value at least $t(85\% of n)$ i.e. if we are getting at least 85% shares of that S_K that are enough to easily recover the S_K while shares with less than t will not reconstruct any secret such that 2 $\leq$ (85% of n) $\leq$ n. In our scenario, we initialized the threshold value i.e. 85% because it meets the federally recommended results and the shares will just adapt themselves to adjust the secret key. If we assume any biased value, we still can do shares accordingly, but it might lead to poor performance every times. We are dividing S_K i.e. secret key into the number of shares and gives them to the only qualified sets. Fig. 1 identifies the keys that are used to distribute a secret (S) value amongst a group of individuals ($shareholders$) each of which is allocated with some information ($share$) related to the secret that have to reconstruct all parts of keys together to get the actual secret key. The secret can only be reconstructed when the shares are combined together. Individual shares are of no use on their own.

More specifically, our scheme consists of mainly three phases: Key Generation, Distribution of keys and Secret key reconstruction. In key generation phase, the sender S_i (trusted dealer) produces the encrypted key from input data $\{D_1, D_2, \ldots, D_n\}$ to protect our sensitive data. Then this generated key are distributed to shareholders through a secured channel. Here, Key Sharing algorithm is used

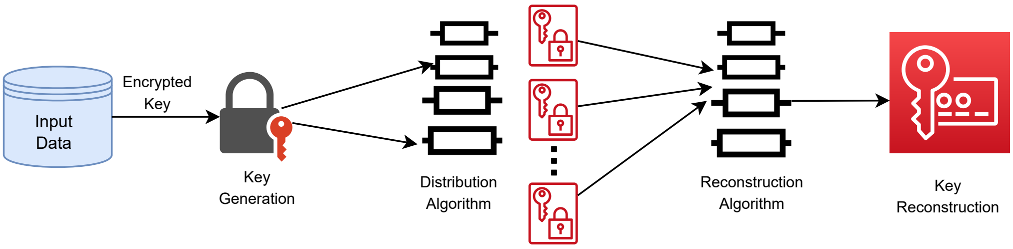


Fig. 1. Key Generation, Distribution, Reconstruction.

for allocating the key's shares. In reconstruction phase, it defines the key reconstruction methods and its mathematical estimations. By using these methods the secret key will be recovered only from an authorized set of shares.
S_i represents the shares of the secret S:

$$\sum_{i=1}^{n} S_i, i = 1, 2, ..., n$$

C. Estimation and Components

Our scheme uses the polynomial interpolation evaluation method [29] for allocating the shares to the shareholders. It is based on threshold values at any t or more shares are sufficient to restructure the secret key (S_K). S_K is expressed as a random $(k-1)$-degree polynomial $g(x).\{g_1(x), g_2(x), \ldots, g_n(x)\}$ are called the split pieces of $g(x)$, the degree of $g_1(x)$ is lv_1, the degree of $g_2(x)$ is lv_2, …, the degree of $g_n(x)$ is lv_n; The degree $lv_i \geq 1, i = 1, 2, \ldots, n$ Maximum $\{lv_1, lv_2, \ldots, lv_n\} = k-1$. Then the polynomial estimation starts with considering polynomial g(x) over commutative ring R. [comparative]

$$g(x) = g_0 + g_1 x + \cdots + g_{k-1} x^{k-1} \tag{1}$$

Polynomial g(x) is estimated on a given vector v = $[v_1, v_2, \ldots, v_n] \in R^n$. Polynomial estimation plotting is designed as:

$$estm(g) := R[x] \rightarrow R^n \tag{2}$$

The output of a vector is given by:

$$estm(g) := [g(v_0), \ldots, g(v_n)]^T \tag{3}$$

In many of the practical applications, evaluation of polynomial has got noticed over a finite field. We consider operations like addition and multiplication for the given prime numbers, $S_P := [0, 1, \ldots, P-1]$ over S_P performed modulo P. To estimate the polynomial coefficients for vectors v_n and these polynomials has been taken from section SP. This whole evaluation is performed under the Galois Field $[GF(p)]$.

For the polynomials reconstruction, our scheme collects threshold value $t\,(85\% of n\, S_H)$ shares to recover the shared secret data. By using Lagrange 's interpolation method, which uses the polynomial construction that passes through k points The polynomial 's construction of degree n that passes through k points, a set of basic polynomial $M_p(x_i)$ defined as:

$$M_p(x_i) = \prod_{p=1, p\neq 1}^{k} \frac{x - x_p}{x_i - x_p} \tag{4}$$

where,

$$M_p(x_i) = \begin{cases} 1, & when \;\; j = i \\ 0, & when \;\; j \neq i \end{cases}$$

Here basic function has been defines above, now we get $(k-1)^{th}$ degree. Lagrangeś interpolation polynomial identifies:

$$g(x) = \sum_{i=1}^{k} y_i \prod_{j=1, j\neq 1}^{k} \frac{x - x_j}{x_i - x_j} \tag{5}$$

$$g(x) = \sum_{i=1}^{k} y_i L_j(x_i) \tag{6}$$

Here, x-value classifies the user and y-value classifies their corresponding share's value.

In public-key infrastructure it is frequently useful to be able to reconstruct private keys. For example, if a user has lost his smartcard that contains his private decryption key, then he cannot decrypt any encrypted file on his computer anymore. So those encrypted files are then inaccessible for the user unless it is possible to reconstruct the decryption key. However, for security reasons it might be important that the key could not be reconstructed by a single person. That person could abuse the knowledge of the private key. It would be more secure if a group of people has to be involved in the reconstruction. In this framework, we present the design of secret sharing in the cloud, we propose several operations to achieve the security goals. In Fig. 2 the sender is sending their data to the receiver through a secure cloud and keys are sending differently and then the whole keys are encrypted with the help of encryption algorithm.Here we can adopt any one of the encryption algorithm but we are applying RSA [18] for the implementation. Encrypted key is split into parts here like $\{p_1, p_2, \ldots, p_n\}$. There are various components which we concerned in achieving better security.

Attackers- Attackers can be anywhere, i.e. on the receiver side, sender side or on the cloud so we have to maintain any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy information system resources or the information itself. The easiest way to protect a network from an outside attack is to close it off completely from the outside world. A closed network provides connectivity only to a known trusted parties and sites; a closed network does not allow a connection to public networks. But in cloud, itś not possible to maintain the closed networks so we have to distinguish malicious traffic from normal traffic.

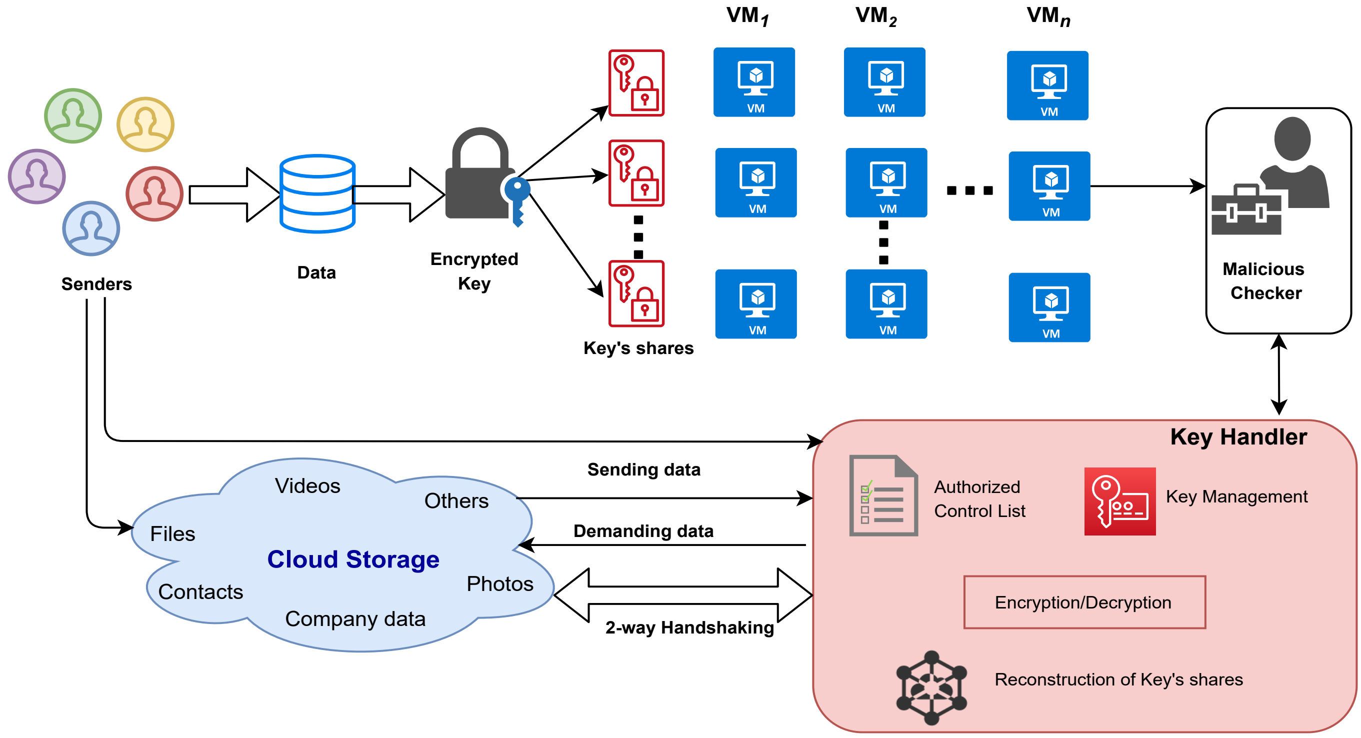


Fig. 2. Secret Sharing Framework

New participant Addition- In any case if new client adds a link to the group of users, joining of the new user is made at the request of the group header. The request includes the new user's official information along with all the access control rights that an owner wants to give. The access details should include the facts of read/write, or read-write approaches granted to the participants. Here, date can also be declared from, when the client has become a part of our key shareholder's group and which access rights are authorized for the participants. This ensures the upcoming access control for the joining member. Then the key holder should update our participant's list.

Remove Quitted Participant- In the case of quitted participant, the key owner should eliminate all the records of that departing participant and update all the records of the related files regarding that participant. The remove option will take the client out of the shareholderś group, but doesn't delete the participant 's previous contributions. They can request to join the same group again if key owner permits.

Key Handler- Key Handler (KH) manages all the issues of cryptographic keys in cryptography. It deals with key generator, storage, sharing of keys. Along with these security services, key infrastructure is also provided. While transferring the keys from sender (S_n) to receiver (R_n) key handler plays a big role during transmission. In this technique, it should be a trusted party and security of the system should be dependent on the security of key's shares. Cloud is requesting for the key to the key handler and then KH assembles the key's parts from qualified clients $Q_n\{Q_1, Q_2, \dots, Q_n\}$. Then all the key's parts are reconstructed with the help of Lagrange's interpolation method as we discussed and decrypt the keys. After all this, key handler sends the data to the cloud $(secured\,party)$. While sending or receiving the data we must require a secure connection so that there is no leakage between them.

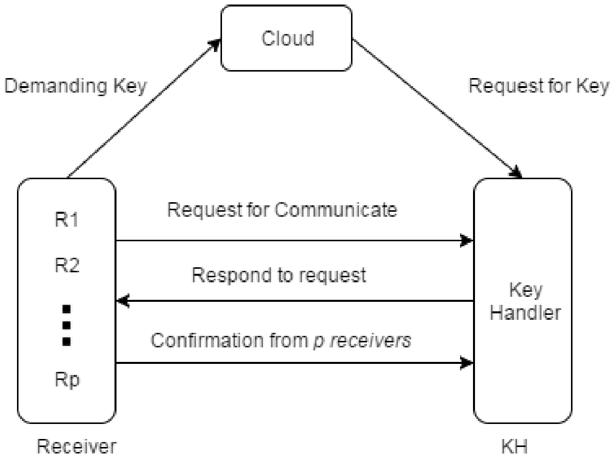


Fig. 3. Three-way Handshaking between receivers and Key handler.

With the help of 2-way handshaking, we can make a secured connection between two channels i.e. key handler and qualified clients. KH requests for the key's parts and sends a SYN packet with sequence number $'x'$, the connection is established. Then Q_i on receiving a SYN packet responds with a SYN packet with sequence number $'y'$ and ACK with sequence number $'x+1'$ to KH to assure that they received the secured data or not.
With the help of three-way handshaking, in Fig. 3 it ensures the communication between two sides without delaying of

the message among them. Any receiver $R_1, R_2, \ldots, R_p$ sends a SYN, data packet over an IP network. The objective of this is to ask/request the key handler for the communication. Then KH responds to their request and acknowledge or SYN packet. After this, Receiver node received the SYN/ACK form KH and responds with an ACK packet.

IV. Malicious Checker Analysis

The malicious checker identifies the authorized or unauthorized clients based on their previous performances. It helps to predict the future performances based on their preceding routine. We used malicious factor mathematical model to analyze the historical interpretations of the clients based on their performance and leakage considerations that how much % of clients are trustworthy? We consider three phases mathematically:

- Performance: The overall effective performance is calculated based on history of the client. Performance is illustrated as *”How many total number of work assigned to each client or machine”*, *“Available time”* and *”Speed of work done”*. Here, we are assuming that Speed=work per hour.

$$Per_{CL} = \frac{Tot_{Wrk}}{Avl_{Tm} \times Spd_{Wrk}} \tag{7}$$

- Non-Guilty Client: We are predicting the non-guilty client based on their *“Total Leakage occurs in the enterprise”* till now and *“Leakage occurs by that client/machine”*. It helps to identify that how much %age of particular client will be guilt? Guilty person are recognized by its own untrustworthiness, irresponsibility, faulty, maliciousness etc.

$$Non_{GLT} = \frac{Tot_{LOCC} - Lk_{Clt}}{Tot_{Lkg}} \times 100 \tag{8}$$

- Key available on mentioned time by particular client: We can presume about the key available by that client (C_i) on that previous times. Key_{AVL} is based on *“Total time when key was available on time”*, *“Breakdowns during transferring the keys”* and *“Wrong entry of the key ever by that client”*. When key handler has requested for key to the client. At that time, is that client has given the key on mentioned time or not? If (C_i) has given key’s part on mentioned time then we can assume that client is trustworthy.

$$Key_{AVL} = \frac{Tot_{tmeky} - Br_{dwn} - Wrg_{etrky}}{Tot_{tmeky}} \tag{9}$$

- Non-Malicious Analysis: We are considering a technique, i.e. required for the analysis of secured clients or non-secured client’s information to determine legitimate or non-legitimate from their previous performances. It inquires safe and unsafe states during VM allocations and estimates the reliability of the client as per the historic performances by malicious checker analysis. The emphasis of this paper is to reduce the possibility of data leakage attacks among different users. When cloud data centers receive requests for the task deployment, then the proposed system will find out the secure virtual machine under VM allocation policies for data confidentiality while avoiding the threats. Finally, legitimate client can be found on the basis of average on above three analyses such as client’s performance, client’s non-guilty analysis and key’s share available on previous times by a particular client.

$$NM_{ANLY} = \frac{Per_{CL} + Non_{GLT} + Key_{AVL}}{3} \tag{10}$$

With the help of this analysis, the legitimate receiver is recognized and we can give the share of the secret to that client by eliminating the risks involved. Here, we find their corresponding %ages and if %, $\geq$ 80% then we can identify that particular client is more than 80% genuine and we can share the secret securely as shown in Table 3. And if %age $\geq$ 90% then we can surely contribute the top secret with that particular client consistently.

TABLE III. Sensitive Level Measurements

Analysis	Sensitive Level
If result $<$ 80%	Less Secret
If result $\geq$ 80%	Secret
If result $\geq$ 90%	Top Secret

TABLE IV. Probability Analysis

Shares	Probability	Security
2	1	Highly Secured
16	0.5	Much Secured
256	0.25	Secured
More than 256	Less than or equal to 0.25	Less Secure

After the findings of trust clients, CSP (Cloud Service Providers) will collect the shares of the secret and it will pass those shares to the Key handler. In such a way only qualified clients (Q_n) can pass the shares or reconstruct the value of S whereas Non-qualified clients (NQ_n) cannot determine anything about the secret key. One drawback is also there, if shares given to participants are too long then we need more memory requirements. At this particular time our share distribution algorithm can become inefficient. Otherwise, this concept has been proved very useful for both practically, as a means to keep important information both from overexposure and from possible loss, as well as theoretically. Naturally, the size of each share grows as a function of the number of performers and the security parameters. If we are taking more shares of the secret, then probability of getting the secret is minimized and if we are receiving a less number of shares then probability of getting the secret is high. If we are having 2 shares of the secret then probability of getting the secret is higher and we assumed:

- If $n = 0$, Probability=1 when $S(n) = 2$
- If $n = 1$, Probability=$\frac{1}{2^n}$ when $S(n) = 2^{4n}$
- If $n = 2$, Probability=$\frac{1}{2^n}$ when $S(n) = 2^{4n}$

TABLE V. ALGORITHM 1

```
Key Generation and Encryption

Input:Any text file with $.txt$ extension, Prime field $S_p$, Users $n$, threshold $t$.
Get a file from user.
If $.txt$ extension not found then print Please choose a file with $.txt$ extension
Get file from the requesting user or download from the cloud
Get from the user: Input Secret Key-Input the number of parts you want to generate:
setPolynomialValue(long $data$, int $n$ Parts)
polynomial[0] = $data$
Set $s_0$=secret
Arbitrarily selects polynomial constants
$g(x) = g_0 + g_1 x + \cdots + g_{k-1} x^{k-1}$
Given vector v = $[v_1, v_2, \ldots, v_n] \in R^n$ estimated by g(x) polynomial
$estm(g) := R[x] \rightarrow R^n$
The output as a vector is given by:
$estm(g) := [g(v_0), \ldots, g(v_n)]^T$
set of basic polynomial $M_p(x_i)$ defined as:
$M_p(x_i) = \prod_{p=1,p\neq 1}^{k} \frac{x-x_p}{x_i-x_p}$
where
$M_p(x_i) = \begin{cases} 1, & when \;\; j = i \\ 0, & when \;\; j \neq i \end{cases}$
Construct a confidential polynomial
Calculate shares using constructed polynomial $sh_i$=g(i) for 1$\leq$ i $\leq$ n.
Generated Shares $sh_i$.

Output: It generates key, complete encryption process and secret's shares assigned to participants.
```

TABLE VI. ALGORITHM 2

```
Decryption Algorithm
Input: Shares $s_{t1}, s_{t2}, \ldots, s_{tk} \in P$ ,where ($t_k \in 1, 2, \ldots, n$ ), Qualified shareholders ($Q_n$)

Secret data $S$
Use Lagrange's Interpolation method to reconstruct the shares of $S_K$,
$g(x) = \sum_{i=1}^{k} y_i \prod_{j=1,j\neq 1}^{k} \frac{x-x_j}{x_i-x_j}$
$g(x) = \sum_{i=1}^{k} y_i L_j(x_i)$
Collects shares of SK, threshold value 85% shares of that $S_K$ are enough to
easily recover the $S_K$ such that $2 \leq (85\% of n) \leq n$. Print: Decypted Secret Key

Output:It decrypts the secret key and reconstruct the complete secret $S_K$.
```

- And, If $(n : n > 2)$, Probability $\leq$ 0.25 when $S(n) > 256$.

Table 4 shows how many shares are there, probability and how much level of security we achieved in this proposed structure. So, we can say our shares (s) are the inversely proportional to probability (p) such that:

Shares of the secret $\propto$ *Probability analysis*

To identify the complete concept of Secret Sharing, we have to understand the whole perception gradually of this implementation part. Here we are providing the authorized security to any documentation like pdf or any doc file, etc., but we are experimenting on any text files, i.e. $(.txt)$ that consists of tables, alphanumeric text or images. Firstly, (S_i) wants to preserve their data confidential and their secret key S_K send by number of K participants. Then the whole secret key is split into shares that how many shares we want to generate. At present secret's shares generated are in encrypted outward appearance. But, (S_i) give a share of the secret to only (Q_i) participants whose are checked by malicious checker whether they are approved or not. Through the help of these shares we can restructure the entire secret and can decrypt the secret key with the assistance of decryption algorithm by Key Handler. Algorithm 1 and Algorithm 2 illustrates the encryption and decryption process.

V. EXPERIMENTAL EVALUATION

Initiating over a supercomputer sometimes is a difficult task, but here an archetype is to be designed for the execution of our program. The experiments are carried out by using Cloudsim 3.0 and Java-Eclipse IDE on a machine equipped with Intel® Core™ I5-3230M processor of 2.60 GHZ clock speed and 8 GB of main memory to evaluate the efficacy of the proposed framework. The simulated cloud network is used to communicate between these three entities: cloud, user and key handler. The whole communication in this model was secured using the SSLStream class and it uses the SHA-256 hash function for generating the keys and RSA used for encryption and decryption. For accessing all the methods of SHA-256 we have used the class SHA256 CryptoServiceProvider. The results are performed on 0 to 200 physical hosts with different configurations and had 10 virtual machines on each host. We have compared our model's performance to that of widely adopted well-known methods employed in literature, in terms of well-defined set of metrics. A series of randomly generated files with different sizes has been carried on each experiment and every task request has obvious need for computing amount of resources. The experiments are performed with some benchmarks and intending to provide security for them. Benchmarks are 2013 and 2015 CASE Fundraising in International Schools Survey Reports [21] [22].The parameters used in simulation is illustrated in Table VII.

TABLE VII. Parameters Used in Simulation

Parameters	Value
Hardware Sprecifications	
CPU	Intel® CoreTM i5 3217U CPU2@1.80GHz
RAM	8GB
STORAGE	1024GB
Graphics Card Recommended Graphics:	1024 × 768 × 32-bit color
Other	CD-ROM Drive
VM Setup of Data Center	
CPU Computing ability	1860 MIPs, 2660 MIPs
Disk I/O	8 GB
RAM	4096 MB
Bandwidth	100 M/s
Storage	10 G
Task Setup of Data Center	
Length (CPU)	[250-1000] MIPs
File Size	[100-2000] MB
Output size (Memory)	[20-40] MB

A. Results & Analysis

1) Computation time for Secret Key Generation: The experiment tests the computation time for key generation which defines the total time needed for processing the keys from beginning to the end which is shown in Table VIII. It is performed by using CryptoServiceProvider class for SHA-256 by varying the number of shares and providing these shares to distinct clients. The processing time increases with the number of increasing number of users from 10 to 100. It is compared with three secret sharing algorithms presented in [12], [17] and [20].

TABLE VIII. Time Computation for Key Generation

No. of users	Computation time (ms)			
	[12]	[17]	[20]	[SSS]
10	1.494	1.594	1.534	0.0281
20	1.598	1.741	1.606	0.0564
30	1.673	2.321	1.684	0.0926
40	1.791	1.888	1.799	0.0989
50	1.907	1.952	1.866	0.0456
60	1.954	2.193	1.923	0.0886
70	1.944	2.286	2.034	0.0942
80	2.092	2.694	2.129	0.164
90	2.401	2.827	2.388	0.1856
100	2.495	2.887	26.12	0.1987

The second experiment is used to calculate the response time between different file sizes from 0.1 MB to 500 MB. It shows the response time for uploading (UL) and downloading (DL) the data to the cloud for encryption and decryption. SSS is compared with other three techniques and it reveals that SSS framework outperforms the existing methods due to absence of heavy computations. It minimises the response time while maximising the performance and its usage of their data centres. As shown in Table IX, SSS improves the turnaround time up to 34.72%, 62.29% and 84.76% for [12], [17] and [20] respectively.

Fig. 4 and 5 shows the time computation of file encryption/decryption with other existing techniques in the simulated cloud network data center. As key computation is also evaluated during the encryption and decryption with varying file sizes. The file size is differentiated with in the range 0.1, 0.5, 1, 10, 50, 100 and 500 MB and compared with SeDaSC methodology [14]. It consumes less time because resources are available so that easily distribution of the shares to resource providers. startTime=System.currentTimeMillis();
endTime=System.currentTimeMillis();
timetook=endTime-startTime;
However, total time is composed from the time of submission of request to the server at which the file is encrypted or decrypted on the cloud. The time for the computation of the secret key is independent of the file size.

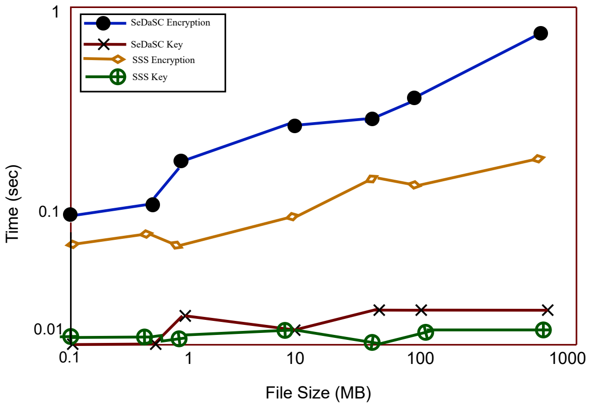


Fig. 4. Time Computation comparision of File Encryption

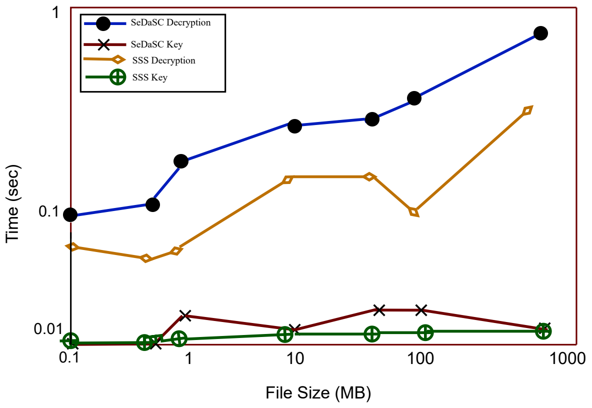


Fig. 5. Time Computation comparision of File Decryption

Fig. 6 shows the average total time to split the secret into shares and distribute them to other nodes. It is shown

TABLE IX. Response Time Calculation

File Size (MB)	[12]		[17]		[20]		[SSS]	
	UL	DL	UL	DL	UL	DL	UL	DL
0.1	0.90	0.81	1.4	0.99	1.48	1.15	0.84	0.72
0.5	1.18	0.96	1.48	1.03	1.89	1.34	0.92	1.04
1	1.80	1.39	2.06	1.48	2.90	1.85	1.39	1.22
10	13.05	9.91	14.95	9.90	14.59	10.45	7.68	5.59
50	53.68	33.45	58.56	35.57	60.37	35.90	8.23	12.64
100	99.69	57.14	112.41	59.14	155.15	61.59	21.69	31.22
500	369.72	215.3	492.03	229.81	872.09	400.21	51.29	44.89

TABLE X. Non-Maliciousness analysis based on predicted Values

Client	Performance	Non-Guilty Client	Key Available	Non-Malicious Analysis	Sensitive-ness
2	80%	70%	72%	74%	Non-Reliable
4	90%	85%	80%	85%	Secret
8	90.6%	86.5%	86.2%	87.76%	Secret
16	98%	95%	90%	94.30%	Top Secret
32	63%	54%	89%	68.6%	Non-Reliable
64	91%	98%	89%	92.66%	Top Secret

that number of nodes increases, the time computation to find nodes and distribution to closed ones decreases. It is compared with multilevel threshold mechanism [5] which conveys that SSS takes less time for processing the number of nodes than other heuristics. SSS improves the average processing time by 30.89% over multilevel threshold policy respectively.

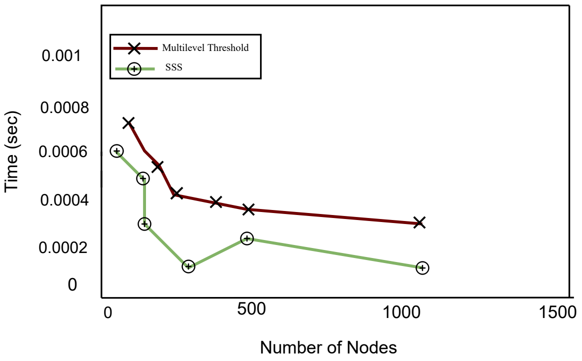


Fig. 6. Time Computation to split shares and distribution

2) Malicious Analysis : In this evaluation, we analyzed the percentage level of guilty clients and reliable clients. Here, Non-Reliable clients are those which do not pass the malicious checker analysis test based on client's historical observations with these three components: Performance, Calculate % of malicious clients, Key available on mentioned time by particular client. If %age $\geq$ 90% then sender can give share of the top secret level to the particular client whose passed this malicious analysis test, If $\%age \geq$ 80% then sender can give share of the secret and If %age $<$ 80% then sender don't give the share of secret to that guilty client. Fig. 7 shows the malicious level of various clients which we done in our experiment.

Based on their preceding performances, identification of guiltiness of each client and key accessible by that client on that declared time, we have found the predicted values for each client. By using these above mentioned investigations, we can find analysis of non-maliciousness for each client by taking the average of these three as calculated in Table X.

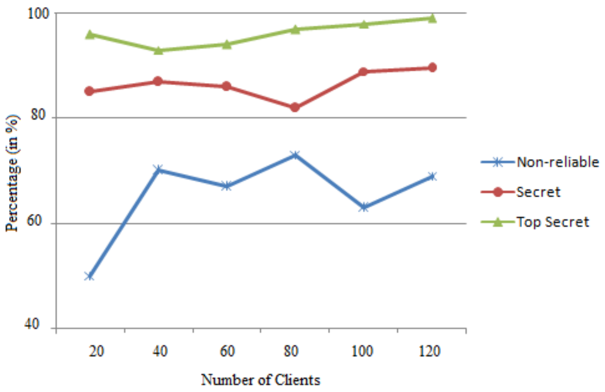


Fig. 7. Malicious analysis of various clients

VI. Conclusion

This paper has provided a new perspective of secure cloud computations for the data confidentiality. The main purpose of this research work is to mitigate the threats by minimising the probability of attackers co-locating with the targets and avoid data leakage and data disposals. It is well-known that security for cloud environment is a non-compromised requirement. The encryption and decryption functionalities are performed by using different file sizes. The difficulty of achieving security has been compared under three basic and widely used secret sharing policies. Moreover, it is adaptive to the dynamic network environment and reduces key time computation. The proposed solution minimizes the wastage of resources and reduces the uploading/downloading rate. The trust level of each client is also evaluated with the help of malicious or non-malicious analysis. Key Handler (trusted party) determines the authority control list of the particular clients. The results revealed that the SSS methodology can be practically used in the cloud for secure data sharing among the group. It validates that the framework can successfully handle both defending against the data leakage attacks and key management. Although, the budget awareness has not been considered into this process. In future work, we will study the adaptive methods to better balance the tradeoff between security, resource efficiency and monetary costs. Besides, add more objectives into our model and then implementing the algorithm in a real cloud data centre constitute our future work.

Acknowledgment

The authors would like a wonderful thanks to the DeitY (Department of Electronics and Information technology) for this research project. And we are extremely fortunate to get constant encouragement, support as well as appreciation.

References

[1] Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P. and Sikdar, B., 2019. A survey on IoT security: application areas, security threats, and solution architectures. IEEE Access, 7, pp.82721-82743. DOI: 10.1109/ACCESS.2019.2924045.

[2] Farras, O. and Padró, C., 2012. Ideal hierarchical secret sharing schemes. IEEE transactions on information theory, 58(5), pp.3273-3286. DOI: 10.1007/978-3-642-11799-2_36.

[3] Butoi, Alexandru, and Nicolae Tomai. "Secret sharing scheme for data confidentiality preserving in a public-private hybrid cloud storage approach." In 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, pp. 992-997. IEEE, 2014. DOI: 10.1109/UCC.2014.163.

[4] Roukounaki, Aikaterini, Sofoklis Efremidis, John Soldatos, Juergen Neises, Thomas Walloschke, and Nikos Kefalakis. "Scalable and Configurable End-to-End Collection and Analysis of IoT Security Data: Towards End-to-End Security in IoT Systems." In 2019 Global IoT Summit (GIoTS), pp. 1-6. IEEE, 2019. DOI: 10.1109/GIOTS.2019.8766407.

[5] Doyel Pal, Praveen Kumar Khethavath, Johnson P. Thomas, Tingting Chen, " Multilevel Threshold Secret Sharing in Distributed Cloud", Communications in Computer and Information Science (CCIS), Springer, Issue-536, pp. 13-23, August, 2015. DOI: 10.1007/978-3-319-22915-7_2.

[6] Manoranjan Mohanty, Wei T. Ooi, P.K. Atrey, " Secret Sharing approach for securing cloud-based pre-classification, volume Ray-casting", Multimedia Tools and Applications (Springer), Volume 75, Issue 11 pp. 1-29, March, 2015. DOI: 10.1007/s11042-015-2567-8.

[7] M. Muhil, U. Hemanth Krishna, R. Kishore Kumar, E.A. Mary Anita, "Securing Multi-Cloud Using Secret Sharing Algorithm", Big Data, Cloud and Computing Challenges, Science Direct, Vol. 50, pp 421-426, 2015. https://doi.org/10.1016/j.procs.2015.04.011.

[8] Meneghello, Francesca, Matteo Calore, Daniel Zucchetto, Michele Polese, and Andrea Zanella. "IoT: Internet of Threats: A Survey of Practical Security Vulnerabilities in Real IoT Devices." IEEE Internet of Things Journal 6, no. 5, 2019, pp: 8182-8201. DOI: 10.1109/JIOT.2019.2935189.

[9] Hua Yi Lin, Che-Yu Yang, Meng-Yen Hsieh, "Secure map Reduce Data Transmission Mechanism in Cloud Computing Using Threshold Secret Sharing Scheme", Advances in Intelligent and Soft Computing (Springer), pp 761-769, January 2012. DOI: 10.1007/978-3-642-25349-2_101.

[10] Chenyutao Ke, Hiroaki Anada, Junpei Kawamoto, Kirill Morozov, "Cross-group Secret Sharing for Secure Cloud Storage Service", Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication in ACM Digital Library, January 2016. DOI: 10.1145/2857546.2857610.

[11] Mohammad Ali Hadavi, R.Jalili, E. Damiani, S. Cimato, "Security and Searchability in secret sharing-based data outsourcing", International Journal of Information Security (Springer), Vol. 14 pp. 513-529, November 2015. DOI: 10.1007/s10207-015-0277-x.

[12] Xu, Lei, Xiaoxin Wu, and Xinwen Zhang. "CL-PRE: a certificateless proxy re-encryption scheme for secure data sharing with public cloud." In Proceedings of the 7th ACM symposium on information, computer and communications security, pp. 87-88, 2012. https://doi.org/10.1145/2414456.2414507.

[13] Lichun Li, Michael Militzer, Anwitaman Datta, "rPIR: ramp secret sharing-based communication-efficient private information retrieval", International Journal of Information Security (Springer), Vol-15, pp-1-23, September, 2016. DOI: 10.1007/s10207-016-0347-8.

[14] Mazhar Ali, R. Dhamotharan, E. Khan,Samee U. Khan, "SeDaSC: Secure Data Sharing in Clouds", IEEE Systems Journal (IEEE), April 2015. DOI: 10.1109/JSYST.2014.2379646.

[15] Md Kausar Alam, Sharmila Banu K, " An Approach Secret Sharing Algorithm in Cloud Security over Single to Multi-Clouds", International Journal of Scientific and Research Publications, Vol. 3, Issue 4, April 2013. ISSN 2250-3153 .

[16] Alfonso Cevallos Manzano, Ronald Cramer, Serge Fehr, " Reducing the Share Size in Robust Secret Sharing", Mathematisch Institute Universiteit Leiden, Master Thesis, October 2011.

[17] Seo, Seung-Hyun, Mohamed Nabeel, Xiaoyu Ding, and Elisa Bertino. "An efficient certificateless encryption for secure data sharing in public clouds." IEEE transactions on Knowledge and Data Engineering 26, no. 9, 2013 pp: 2107-2119. DOI: 10.1109/TKDE.2013.138.

[18] Suli Wang, Ganlai Liu, " File encryption and decryption system based on RSA algorithm", International Conference on Computational and Information Sciences, IEEE, 2011. DOI: 10.1109/ICCIS.2011.150.

[19] Satoshi Takahashi, Keiichi Iwamura, "Secret Sharing Scheme Suitable for Cloud Computing", 27th International Conference on Advanced Information Networking and Applications (IEEE), 2013. DOI: 10.1109/AINA.2013.124.

[20] Khan, Abdul Nasir, ML Mat Kiah, Sajjad A. Madani, Mazhar Ali, and Shahaboddin Shamshirband. "Incremental proxy re-encryption scheme for mobile cloud computing environment." The Journal of Supercomputing 68, no. 2, 2014, pp: 624-651. DOI: 10.1007/s11227-013-1055-z.

[21] Shaofeng Zou, Yingbin Liang, Lifeng Lai and Shlomo Shamai (Shitz), "An information theoretic approach to secret sharing", IEEE transations on information theory (IEEE), Vol. 61 June 2015. DOI: 10.1109/TIT.2015.2421905.

[22] Lin, Changlu, Huidan Hu, Chin-Chen Chang, and Shaohua Tang. "A Publicly Verifiable Multi-Secret Sharing Scheme With Outsourcing Secret Reconstruction." IEEE Access 6, 2018, pp: 70666-70673. DOI: 10.1109/ACCESS.2018.2880975.

[23] Chen, Dong, Wei Lu, Weiwei Xing, and Na Wang. "An Efficient Verifiable Threshold Multi-secret Sharing Scheme with Different Stages." IEEE Access, 2019. DOI: 10.1109/ACCESS.2019.2929090.

[24] Aisha Abdallah, Mazleena Salleh, "Secret Sharing Scheme Security and Performance Analysis", International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (IEEE), pp-173-180, 2015. DOI: 10.1109/ICCNEEE.2015.7381357.

[25] Muhammad Adeka, Simon Shepherd, Raed Abd-Alhameed, Nuredin A. S Ahmed, "A Versatile and Ubiquitous Secret Sharing: A cloud data repository secure access", Internet Technologies and Applications (ITA) (IEEE), 2015. DOI: 10.1109/ITechA.2015.7317449.

[26] MIAO Fuyou, FAN Yuanyuan, WANG Xingfu, XIONG Yan, Moaman Badawy, "A (t,m,n)-Group Oriented Secret Sharing Scheme", Chinese Journal of Electronics (IEEE): Vol. 25, No. 1, January 2016. DOI: 10.1049/cje.2016.01.026.

[27] Yi SUN, Gaochao LI, Zhaowen LIN, Fei XIAO, Xiaoming YANG, "A Completely Fair Secret Sharing Scheme without Dealer" International Conference on Consumer Electronics-Taiwan (ICCE-TW) (IEEE), 2016. DOI: 10.1109/ICCE-TW.2016.7520905.

[28] Shalini I S, Mohan Naik R, Dr. S V Sathyanarayana, "A Comparative Analysis of Secret Sharing Schemes with Special Reference to e-Commerce Applications", International Conference on Emerging Research in Electronics, Computer Science and Technology (IEEE), 2015. DOI: 10.1109/ERECT.2015.7498980.

[29] Min Huang, Vernon J. Rego, "Polynomial Evaluation in Secret Sharing Schemes", 2010. Corpus ID: 16343331.

[30] Felix Günther, Bertram Poettering, "Linkable message tagging: solving the key distribution problem of signature schemes" International Journal of Information Security (Springer), pp-1-17, March 2016. DOI: 10.1007/s10207-016-0327-z.

[31] M. Breezely George, S.L.Sabasti Prabu, "Secured Key Sharing in Cloud Storage using Elliptic Curve Cryptography" Proceedings of the International Conference on Soft Computing Systems pp 21-31 (Springer), January 2016. DOI: 10.1007/978-81-322-2674-1_3.

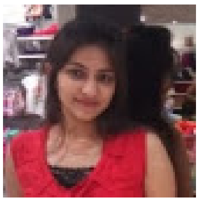
[32] Cas Cremers, Marko Horvat, "Improving the ISO/IEC 11770 standard for key management techniques" International Journal of Information Security (Springer), Vol-15, pp-659-673, November 2015. DOI: 10.1007/s10207-015-0306-9.

[33] Xiao-Fen Wang, Yi Mu, Rongmao Chen and Xiao-Song Zhang, "Secure Channel Free ID-Based Searchable Encryption for Peer-to-Peer Group", Journal of Computer Science and Technology (Springer), pp-1012-1027, September 2016. DOI: 10.1007/s11390-016-1676-9

[34] Farrukh Nadeem and Rizwan Qaiser, "An Early Evaluation and Comparison of Three Private Cloud Computing Software Platforms", Journal of Computer Science and Technology (Springer), pp- 639-654, May 2015. DOI:10.1007/s11390-015-1550-1.

[35] A. K. Raina and S. C. Kak, ”Data security: A cryptographic approach”, Proceedings of the Indian Academy of Sciences, Engineering Sciences, Sadhana (Springer),pp-65-83, Volume 5, Issue 1, March 1982. https://doi.org/10.1016/j.procs.2015.03.232.

[36] Murat Yesilyurt and Yildiray Yalman, ”New approach for ensuring cloud computing security-using data hiding methods”, Sadhana (Springer), pp-1289-1298, Volume 41, Issue 11, November 2016. DOI: 10.1007/s12046-016-0558-8.



Sakshi Chhabra received the BCA degree in Computer Applications from the Punjab University, Chandigarh in 2012, and the MCA degree in Computer Applications in 2015. She began her research on Cloud Computing in 2015. Currently, she is a Research Scholar with the National Institute of Technology, Kurukshetra in Department of Computer Applications. Her main research interests include Cloud Computing and Information Security. She has published the research papers in SCI, Scopus journals and International Conferences.

Ashutosh Kumar Singh is working as a Professor in National Institute of Technology, Kurukshetra, India. He has more than 15 years research and teaching experience in various Universities of India, UK, and Malaysia. Prior to this appointment, he has worked as an Associate Professor and Head of Department Electrical and Computer Engineering in School of Engineering Curtin University Australia offshore Campus Malaysia, Sr. Lecturer and Deputy Dean (Research and Graduate Studies) in Faculty of Information Technology, University Tun Abdul Razak Kuala Lumpur Malaysia, Post Doc RA in the Department of Computer Science, University of Bristol, Faculty of Information Science and Technology, Multimedia University Malaysia and Sr. Lecturer in Electronics and Communication Department at NIST, India. His research area includes Web Technology, Big Data, Verification, Synthesis, Design and Testing of Digital Circuits. He has published more than 300 research papers now in different journals, conferences and news magazines.