A COMBINED DATA STORAGE WITH ENCRYPTION AND KEYWORD BASED DATA RETRIEVAL USING SCDS-TM MODEL IN CLOUD

***S.V.Divya*[1], *R.S.Shaji*[2], *P.Venkadesh*[3]**

[1,3]Faculty of Information Technology , Computer Science and Engineering , Noorul Islam Centre for Higher Education Kumaracoil,Thuckalay-629180, Tamilnadu, India

[2]Professor, Department of Computer Science and Engineering, St. Xavier's Catholic College of Engineering, Chukandai, Nagercoil 629003, Tamil Nadu, India

Email: divyasadasivam@gmail.com[1]; shajiswaram@yahoo.com[2]; pvenkadesh2002@gmail.com[3]

DOI: https://doi.org/10.22452/mjcs.vol32no3.1

ABSTRACT

With the hasty growth of the Internet technologies and the need for increased computational methods, cloud computing has become a new paradigm in the business and IT endeavors. This technology facilitates the enterprises to move their confidential data into the cloud, where their data is stored in a remote data centers and accessed via the internet. These resources are controlled by third parties who may not be trustworthy and provide adequate data security. While this technology has many benefits, the security issues also increases. Various methods have been proposed for designing a secure cloud storage system.However many of the prior works suffers from security, confidentiality, and integrity issues that limit the functionalities of the storage system. Moreover, the existing schemes suffers delay due to the lack of online verification, and increasein the communication cost as data is retrieved back to the owner for every request made by the user. Hence it remains a challenge for constructing an efficient data storage and data retrieval. In this paper, we present the development of a secure and efficient cloud storage system, "SCDS-TM" that supports data confidentiality, data integrity, and data retrieval functionalities. With the use of Elliptic Curve Cryptography and storage correctness proof, weovercame the data confidentiality and data integrity issues. In order to augment the data retrieval functionality, we exploited a keyword search method called "coordinate matching" technique of capturing the related documents based upon the search query.The performance of the proposed method is analyzed based on various factors such as communication overhead,data transfer rate, query execution time, data security level, probability ratio and precision ratio.Thorough analysis of scrutinizing the privacy and efficiency allow our proposed method to be secure, efficient and offers a higher level of security with low communication overhead.

Keywords: Cloud storage, Integrity, Confidentiality, Elliptic Curve Cryptography, Security.

1.0 INTRODUCTION

Cloud computing is a unique paradigm for service oriented computing that propelle enterprises and IT industry into a new era of computing method; where the applications, resources, and the software are delivered as a service without the installation of any hardware or software.This significantly reduces the infrastructure costs,which gained the attention of business enterprises and the public sectors. This technology reduces the costs of the business activities by utilizing the centralized resources from the large pool of data which are connected in private or public networks. Cloud computing provide a dynamically scalable infrastructure for applications, data, and file storage.Cloud offers its customers to configure, manipulate and access the applications online, and suggests online data storage, infrastructure and application. Moreover, cloud makes business applications mobile and collaborative by eliminating the need to install software on local PC, thereby overcomingthe platform dependency problem.

Despite its benefits the security and the privacy are the major concern that influence cloud's performance and limits its adoptions all over the world. Due to this, many businesses are reluctant to continue using cloud . In cloud storage system, data confidentiality, data robustness, and data integrity are some of the major requirement. Since the cloud services are offered by the chain of cloud service providers, different levels of security services are offered by different service providers. Although cloud brings up new opportunities to the modern era, the technology is still at its infancy. The various security issues with the cloud models and their security issues and challenges are addressed in [1-4].

Similar to the Byzantine failures [5] that arise in the storage servers sporadically may assist the service providers to hide the data errors from its clients for their own profit. Hence, in order to ensure the storage correctness and the integrity of the data, Provable Data Possession (PDP) [6], Dynamic Provable Data Possession (DPDP) [7], Cooperative Provable Data Possession (CPDP) [8], and Proofs of Retrievability (POR) [9] were used.

The DPDP scheme ensures the integrity and ownership of the owners' data without downloading the entire data. However, the DPDP scheme is not efficient in multi-cloud storage environment, meaning that efficient integrity verification and privacy is not possible. Since data can be deleted, modified or neglected by the service providers, or the errors in the storage servers, cloud also posses' security threats to the outsourced data. Due to the outsized storage space, time and the amount depleted for the unused or the un-accessed data, the cloud service provider may sometimes neglect or delete those data without the concent of the owner.Hence,protecting the correctness and the integrity of the cloud data is important and has to be addressed seriously.

Numerous methods sucha as authentication and auditing protocols were employed for providing security to the data in the cloud environment. However, the major core of the problem lies on how the clients achieve the integrity and correctness of the remote data in untrusted servers. The general method of checking the integrity of the cloud data is the digital signatures, which acts as a verification metadata performed by means of owner's public key. Hence, there is a chance of data owners inevitably revealing their identity to the public verifiers. As such,a Security Mediator or a Third Party Auditor (TPA) is used to sign on behalf of all the users and to verify the storage correctness.
The Key contributions of this paper are:

- An SCDS-TM model for efficient data storage and retrieval was proposed by considering both the security and the performance as the primary aspects.
- To improve the security level, encryption based on Elliptic Curve Cryptography (ECC) is used.
- To make the data retrieval process more efficient, a keyword searching methodology is employed. The relevant data with that keyword alone is retrieved to the user, strengthening the confidentiality of the data.
- By introducing a Trust Manager (TM), the communication overhead of the data owner is minimized.
- The SCDS-TM model preserves the anonymity and collusion –resistance to great extent.
- To analyze the performance of the SCDS-TM model through various experimental results,and to demonstrate the effectiveness of the proposed model for secure data storage and retrieval.

This paper is organized as follows: Section 2 describes the literature review, Section 3 describes the proposed SCDS-TM architecture model, Section 4 highlights the experimental evaluations, and Section 5 conludes the study.

2.0 LITERATURE REVIEW

As new technology evolves day by day, the security issues have also threatening them in many aspects, recuising more reseach to be carried in the context of security. With the initiation of major companies data and confidential information is endowed with the cloud, security has to be given a paramount importance and that need to be addressed carefully.

However,some impediments such as data leakage, data privacy, storage overhead, insufficient to handle batch auditing for multiple owners,and difficulty in performing the dynamic data operations still exist. This leads to the proposal of a dynamic auditing protocol.To achieve public auditability and to support dynamic data operations like block insertion and deletion, a public auditing scheme with TPA [10-12] was introduced and data authentication is achieved through Merkle Hash Tree (MHT) construction. To minimize the data management costs, the cloud servers are adopting new outsourcing mechanisms. The outsourced information is managed by trusted third parties who perform the secured file access and deletion policies (FADE) [13] with two goals: (1)policy-based file access control and deletion, and (2) key management with attribute –based encryption.However in some cases, the cloud storage providers may be forced to leak the owner's data, leading to the proposal of Attribute Based Encryption (ABE) by Chi.et.al in [14]. This approach creates fake evidence for the forged data in cipher text to the outside coercer such that the outside coercer cannot deny the evidence. The system thus achieves user privacy and ensures secure data sharing with fine-grained access control mechanism.

A secure cloud storage system with robustness, confidentiality, and functionality was proposed by Lin.et.al in [15]. The system supports data storage, data retrieval, and data forwarding functionalities. By means of threshold proxy re-encryption scheme, the encoding, as well as forwarding operations is carried out. Various parameters such as the number of copies of a message transmitted into the storage servers, and the number of storage servers queried by the key server was also analyzed.

Since the data is delegated to the third party, data confidentiality become one of the serious issue and this resulted in the owners to encrypt the data before storing them in the cloud. To ensure confidentiality and privacy of the data stored in the cloud storage servers, encryption methods were normally offered. However, downloading all the data and decrypting them is impractical due to the huge amount of bandwidth costs. Also, storing data in the cloud serves no other purpose except for searching and utilizing them. So a privacy-preserving data search service over encrypted cloud data [16-17] was proposed and it remains a challenging problem for sharing cloud data and performs searching over those encrypted data.

To maintain data privacy in the cloud, the data has to be encrypted before outsourcing and the traditional data utilization is purely based on the plain text keyword search. As such, a privacy-preserving multi-keyword ranked search over encrypted data (MRSE) for efficient cloud data utilization was proposed in [18]. Based on the multiple keywords in the search query, the relevant documents are sent to the user. Eventhough the scheme achieves low computation overhead,it has a high communication overhead to the data owner as every request is directed to the data owner. The anonymity of the data owner is also not preserved.

Once the data is placed at the cloud, the owner has the fear of losing control over those data. To overcome this, PDP and Proof of Storage (POS) was introduced. The data owners are guaranteed that their data is used according to the SLAs they have agreed when they demand for the particular services. In order to overcome the problems, a Cloud Information Accountability (CIA) [19] framework was proposed. This framework ensures that the owners' data is utilized properly and data usage process becomes more flexible and transparent.

With the advancement of technology and openness, data sharing in pubic environment such as cloud imposes various security challenges such as efficiency, privacy (or anonymity)of the data owner, and data integrity. During data sharing, the data may be compromised by some attacks or nodes within the cloud and therefore high security measures need to be employed to tackle the security and performance issues, and also to optimize the data retrieval time.

On top of that, during data sharing, there is a chance of data leakage in the cloud environment, and hence the need for a communication channel framework was addressed in [20]. The framework intends to minimize the data leakage during data sharing and aims to achieve data confidentiality. Later, DROPS (Division and Replication of Data for Optimal Performance and Security) [21] was proposed. Here, the file is splited into fragments and those fragments are replicated over the nodes .The scheme ensures controlled replication by storing each fragments only once in the nodes so that such that no meaningful information is leaked out. Besides this, there are also a number of researches that proposed the enhancement to the data sharing in cloud [22-31] .

Numerous techniques and methodologies for enhancing the security with respect to confidentiality and integrity in the cloud storage were proposed. However, the existing cloud storage system does not focus on enhancing the security of the cloud data during transactions in the cloud servers. Hence, a conditional source trust attribute encryption with particle swarm-based transaction optimization (CSTAE-PSTO) for enhancing the security during transaction processing over the cloud servers was proposed in [32]. The framework can minimize the transaction completion time and improve the security rate on data layer.

By analyzing the above findings and shortcomings, it is observed that the existing work did not look at the development of an efficient and secure cloud data storage and retrieval that prevent the data owners from leaking their confidential data to unauthorized users.To minimize the communication overhead of the owner and to minimize the data retrieval time, a Secure Cloud Data Storage with Trust Manager (SCDS-TM) model is proposed.

The SCDS-TM model for efficient cloud data storage and retrieval along with a coordinate matching technique is proposed. With elliptic curve encryption and coordinate matching technique, the confidentiality of the data is preserved and strengthens the SCDS-TM model.

3.0 CONSTRUCTION OF SCDS-TM MODEL

A Secure Cloud Data Storage with Trust Manager (SCDS-TM) is proposed for data confidentiality problem, which is one of the issues in cloud infrastructure. In this section, we present our proposed SCDS-TM scenario that we consider for the data confidentiality issue and a discussion for a solution. This framework supports all the basic functionalities of the distributed storage system and facilitates a secure data storage and retrieval. Our proposed framework is compared and analyzed with the existing models with respect to various metrics. To achieve data confidentiality, security as well as for a secure retrieval an efficient model is present here. The workflow diagram of the proposed SCDS-TM model is shown in Fig.1.

The proposed SCDS-TM model not only supports the secure storage and retrieval of confidential data, but also ensures the proper coordination of the storage servers and the key servers effectively. Since the data in the cloud is dynamic, data operations like block modification, insertion and deletion is challenging in the existing systems. This model supports dynamic data operations and verifies the integrity of thedata blocks using Merkle Hash Tree (MHT) construction.

The completeness, correctness, and freshness of the data are accomplished by means of MHT. Moreover, our SCDS-TM framework is simple, efficient, and guarantees public verifiable approach to guarantee cloud data integrity without compromising the anonymity of data owners. To construct a cloud storage system that supports efficient data storage, data integrity and data retrieval, an integration of two features such as data confidentiality, and data integrity are showed in the protocol design.

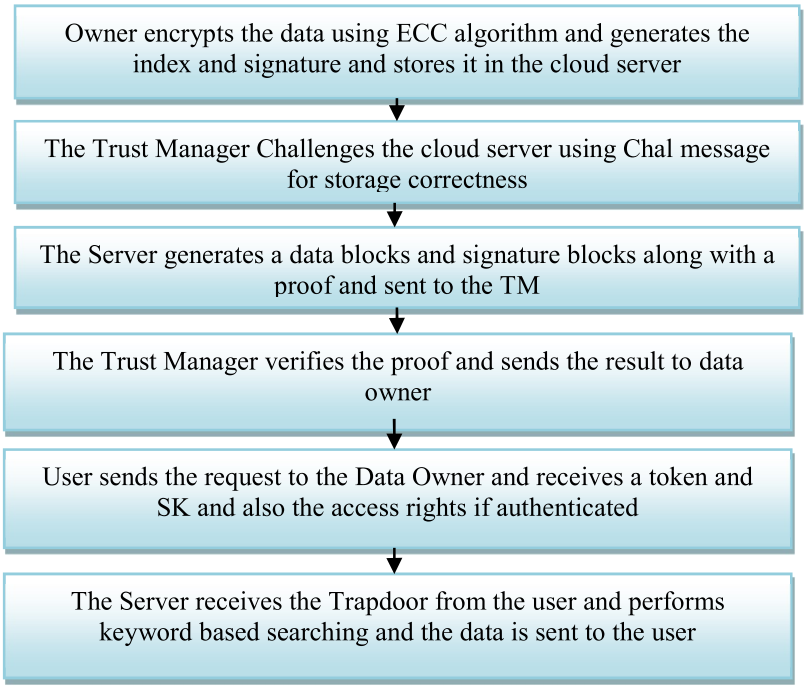

Owner encrypts the data using ECC algorithm and generates the index and signature and stores it in the cloud server
The Trust Manager Challenges the cloud server using Chal message for storage correctness
The Server generates a data blocks and signature blocks along with a proof and sent to the TM
The Trust Manager verifies the proof and sends the result to data owner
User sends the request to the Data Owner and receives a token and SK and also the access rights if authenticated
The Server receives the Trapdoor from the user and performs keyword based searching and the data is sent to the user

Fig.1: Workflow diagram of SCDS-TM model

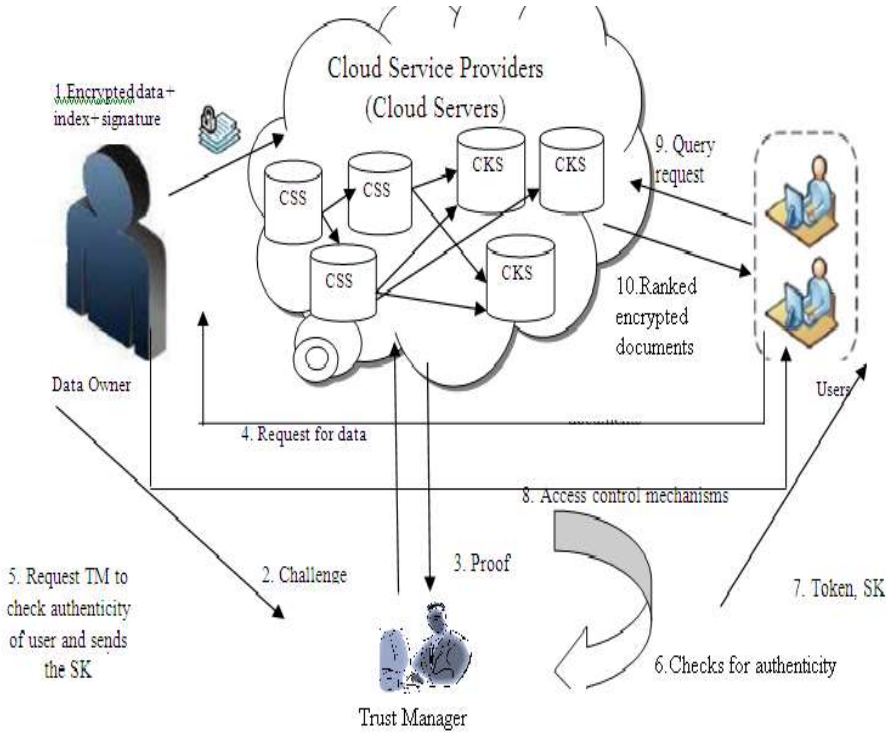

Cloud Service Providers
(Cloud Servers)
1.Encrypted data+ index+signature
CSS
CSS
CSS
CKS
CKS
CKS
9. Query request
10.Ranked encrypted documents
Data Owner
Users
4. Request for data
8. Access control mechanisms
5. Request TM to check authenticity of user and sends the SK
2. Challenge
3. Proof
7. Token, SK
6.Checks for authenticity
Trust Manager

Fig.2: Architecture of SCDS-TM framework

3.1 System Model

We consider the data confidentiality and data integrity issues for data storage purpose. The architectural diagram of the proposed SCDS-TM model is depicted in Fig.2. The cloud server is normally viewed as “honest –but curious”. Even though the server is honest but at some instance, it is also curious in analyzing about the data stored in its storage servers. Hence, our cloud storage framework is constructed which consists of storage servers and key servers. The system is highly distributed so that the storage servers and the key servers independently perform the operations. The storage servers perform the storage functions and the key servers perform the key operations.

The SCDS-TM model consists of 3 entities: 1) Data Owner 2) Trust Manager(TM), and the 3) User. The data owner has all the capabilities of accessing the cloud storage system and they create and host their valuable data in the storage servers. The trust manager has expertise and capabilities similar to the owner. The main objective of the TM is to verify the authenticity of the user. In order to reduce the owners’ burden of responding to users’ request and authenticating them, the trust manager (TM) is used. Moreover, after the data owner outsources his data to the cloud server, it is the responsibility of the trust manager to challenge to the server and acquire the proof for data storage correctness. The TM controls the overall operation of the cloud servers and handles the user requests.

As depicted in Fig2, the data owner initially splits the data into different blocks and encrypts the data by means of elliptic curve cryptography using the ECC Encryption algorithm using Eq.3.1. Then, an index and signature for those blocks are generated using the IndexGen and SigGen algorithm respectively using Eq.3.2, Eq.3.3 and Eq.3.4. The owner hosts the encrypted data in the Cloud Storage Server (CSS) and the corresponding index and the signatures are hosted in the Cloud Key Server (CKS).Once the owner hosts the encrypted data, index and the signature to the cloud servers, the owner seeks the help of the trust manager to ensure the storage correctness of the data using the chal message. After checking the storage correctness of the owner’s data, the cloud server generates the proof (Pr) using the ProofGen algorithm in Eq.3.5 and Eq.3.6. The generated proof is verified by the trust manager and returnthem as either True or False to the owner. The cloud server also runs an ExecuteUpdatealgorithm if a request from the data owner to update the stored data is made. The data is updated by the cloud server and the update is verified by the trust manager using the ProveUpdate algorithm.

Next, when the user requests the data, the data owner demands the trust manager to check the authenticity of the user. If authenciated, the trust manager issues a token and a secret key (SK) to the user and informs to the owner. Once authentication information is received from the trust manager, the data owner sent the access control rights (mechanisms) to the user. The access control rights permits the user to read/ write/ execute the owners' data.

After receiving the access permissions from the data owner, the user sends the request to the cloud server. Once the query request is received by the cloud server, it performs the searching. To make the searching more efficient, a keyword based searching is used. The technique is based upon the principle that the cloud server generates a trapdoor using the dictionary of keywords which is stored in the server by means of TrapGen algorithm. Bygenerating the trapdoor, the user sends the query request which includes (trapdoor, keyword, index). Based upon the index, keyword and the trapdoor the server performs searching and the top –k relevant documents are ranked and sent to the user by the cloud server. To decrypt the documents; user performs the elliptic curve decryption using the Decryption algorithm.

In the proposed SCDS-TM model, the communication and the computation burden of the owners are highly minimized. Also, the necessary and the exact documents are alone retrieved by the user with the help of keyword based searching technique.

Also, the performance of the proposed SCDS-TM model is evaluated with respect to various metrics such as Communication Overhead, Data Transfer Rate, Data Security Level, Query Execution Time, Probability Ratio, and Precision Ratio. With the involvement of the Trust Manager, the Communication Overhead of the owner is minimized. Data Transfer Rate and Data Security Level deals with the rate of transferring the data from the cloud server to the user within a stipulated time period, as well as how the data is sent successfully to the user by the cloud server without any loss respectively.

Query Execution Time is used to analyze the time taken by the cloud server to execute a particular query/request. Probability Ratio explains the probability of successfully retrieving the message by the user and Search Precision investigate how accurate the cloud server performs searching upon receiving the request from the user.

3.2 Design Objectives

With the attacks in the cloud environment, cloud users are worried about the integrity of cloud data. Simultaneously, it is mandatory for the owners to preserve not only their identity privacy, but also the data integrity. The proposed SCDS-TM model is designed to achieve the following properties:

- **Efficiency:** A public verifier or an authenticated Third Party who does not possess the cloud data should check the storage correctness of the data stored in the server withoutretrieving the entire data.Hence, a Trust Manager is used in the proposed SCDS framework.
- **Verifiability:** Here, a public verifier i.e the Trust Manager is used to check the integrity of the cloud data.
- **Anonymity:** In order to preserve the anonymity, the identity of the data owner should not be revealed to the public verifier during integrity checking. Moreover, the Trust Manager does not reveal the identity of the data owners to any users based on the cloud data and the signatures.
- **Collusion-resistance:** The data owners should not collude or share his private/secret key with any other owners to access the cloud data.

3.3 Notations and Preliminaries

The preliminaries which are required for the proposed SCDS-TM model are described here.

3.3.1 Bilinear map: Let B and B1 be any two multiplicative groups which are of cyclic in nature with the prime order p and g Є B be a generator. Let a map m: B*B-> B1 is a bilinear map is bilinear such that for any a,b Є Z_p^*, m(g^a,g^b)=m(g,g)^{ab} ; and degeneracy such that m(g,g) is an un-identity element in B1. The
bilinear map has a distributed hash function DH :$\{0,1\}^*$->B is normally a random .

3.3.2 System Setup (1^k): The system setup takes the input parameter(1^k) and generates α.

Table 1: Notations

Symbol	Physical Meaning
Sk	Secret key
Puk	Public key
C	Cipher Text
T	Set of data tags
B	Bilinear map
Mk	Master key
K	Keywords
D	Dictionary of keywords
τ	Index
$D_{\hat{w}}$	Trapdoor
γ	Signature for each data blocks

3.3.3 Construction of Merkle Hash Tree (MHT): The MHT is an efficient authentication structure which is mainly used to check the integrity of the data blocks in the file. The leaf nodes of the tree represent the authentic hash values. Suppose the verifier requests the prover for authentication of the received blocks, the prover provides the AAI (Auxiliary Authentication Information) of the corresponding root values and the verifier verifies it by comparing the calculated authentic value with the corresponding authentic root value.

3.4 Mathematical modeling

The mathematical modeling of the proposed method is explained below:

3.4.1 KeyGen (1^k): Let the owner divide the file F into different data blocks as F_1, F_2, F_3………F_n; where n_iЄZ_p and p is always a prime. The owner's private key, and the public key is generated by means of invoking the KeyGen algorithm. The owner generates the key pairs (opk, opuk). Choose $\alpha \leftarrow Z_p$ and generate the value $x \leftarrow g^{\alpha}$ .The secret key is sk=(α,opk),and the public key is puk= (x,opukopuk) .Based on these two keys, the owner generates the master key mk=(α, x).

3.4.2 Encryption (): This is used toencrypt the message M by means of Elliptic Curve Cryptography.Before encrypting the data, it is essential to generate the keys for owner. The owner's keys are generated by means of KeyGen() algorithm.

3.4.2.1 C←ECC Encryption (M,r,P,BP)
Input: The message M , the base point BP ,r a random number ,owners' secret key skand a elliptic curve point P.
Output: The encrypted message i.e the cipher text C
1. The message M is mapped with the elliptic curve at a point P.
2. A random integer r is chosen such that rЄ [1,n-1]
3. The encrypted message (cipher text) is determined as a pair:

$$C =\{(r*BP)+(P_m+(r*sk))\} \quad \text{Eq.(3.1)}$$

3.4.3 T←TagGen(C,sk) : The tag generation algorithm is used to generate the tags by considering the data file F and the owner's secret key .Let us assume the encrypted data file F as C where C=(c_1,c_2,….,c_n) , and choose an element $y \leftarrow B$ and generates the tag T .The tag includes:

$name||n||\, y||SSig_{sk}(name||n||y).$ Eq.(3.2)

3.4.4 τ←IndexGen (F, mk) : Based on the data file F, the data owner generates a binary data vector V_i for every documents in the data file F_i, where $V_i[j]$ represents the corresponding keyword which appears in the file F_i and finally the sub-index for all the encrypted documents are generated based on K-Nearest Neighbour computation method. The master key mk is usually composed of one (e+1)-bit vector as X and two (e+1)*(e+1) invertible matrices as $\{A_1,A_2\}$;where e represents the number of fields for each record r_i. The index is built as,
$\tau=\{A_1^T D_i' + A_2^T D_i''\}$;where A_1 and A_2 are matrices for every encrypted document C_i; where $C=(c_1,c_2,\ldots,c_n)$.

3.4.5 γ ←SigGen (sk,F): After generating the tag for the file F, the ownercomputes the signature γ for each data blocksn$_i$(i=1,2,…. ,n) as

$$\gamma i \leftarrow \{H(n_i\ ).\ y^{ni}\}^{\alpha} \qquad \text{Eq.(3.3)}$$

by means of his secret key skand the data File F. The set of signatures for all the data blocks can be represented as, ק={ γi },1≤i≤ n. After generating the signature for each data blocks, the owner with the help of MHT, constructs a root R where the leaf node occupies the ordered set of file tags. The owner then signs the documents by means of his private key

$$\alpha\text{: } sig_{sk}(DH(R)\ ) \leftarrow (DH(R)\ )^{\alpha}\ . \qquad \text{Eq.(3.4)}$$

After signing with the private key, the owner sends (F,T, ק,sig$_{sk}$(DH(R))) to the cloud server and deletes the File, the signature set and the Root R which is signed by the secret key from its local storage.

3.4.6 (Pr)←ProofGen(F,chal, ק): This is used by the cloud server to generate the proof for the data stored in it. It receive the input parameters as the data file F, its signature, and a challenge chaland generates a data integrity proof (Pr) for the blocks which is specified by the challenge chal .
To generate chal, the Trust Manager selects a random element subset I={s$_1$….s$_c$} ;where each i∈I , and chooses a random element $u_i \leftarrow B \subseteq Z_p$. After receiving the challenge chal{(i, u$_i$)} $_{s_1 \le i \le s_c}$ the cloud server generates the data blocks μ and signature blocks σ respectively.

$$\mu=\sum u_i\, n_i \in Z_p \qquad \text{Eq.(3.5)}$$

$$\sigma=\prod \sigma_i^{ui} \in B \qquad \text{Eq.(3.6)}$$

where i=s$_1$…….s$_c$ and μ represents the data blocks and σ represents the signature blocks

3.4.7 (True ,False)←ProofVerf(puk,chal,Pr): This algorithm is used by the owner to validate the proof which is generated by the cloud server. It receives the input parameters as the public key puk, the challenge chal and the proof Pr and outputs the value either true or false.

3.4.8 (ק',F',Pr$_{update}$)←ExecuteUpdate(F,ק,update): This is invoked by the cloud server by accepting the data file F, the signature set and request the operation 'update' from the client side as parameters and outputs ק',F', and Pr$_{update}$- the updated files, signature sets and the proof accordingly.

3.4.9 {(True,False,sig$_{pk}$(DH(R'))}←ProveUpdate(puk,update,Pr$_{update}$) : This algorithm is used to authenticate(verify) the update which is generated by the cloud server by the trust manager(verifier). It outputs either True or False based upon the verification.

3.4.10 $D_{\hat{w}}$←TrapGen(D): With k keywordsof interest , and D be the dictionary of the keywords set that consists D=(D$_1$,D$_2$,……D$_n$) ,this algorithm generates the trapdoor $D_{\hat{w}}$;

3.4.11 $F_{\hat{w}}$←Query ($D_{\hat{w}}$, k, τ): When the cloud server receives the query request as ($D_{\hat{w}}$, k) from the users, it performs the ranked searching on the particular index τ and by means of the trapdoor $D_{\hat{w}}$ and generates $F_{\hat{w}}$the ranked id list of all the top-k documents sorted according to their similarity value.

3.4.12 M ←Decryption(C,upk,P).
Input: The coordinate point C, users' private key upk , owners' secret key skand elliptic curve point P.
Output: The original message M

1. Extract the x co-ordinate of the encrypted message (cipher text) C as,
   x=xcod(C) ; where xcod is the function to compute the x co-ordinate.

Similarly extract the y co-ordinate of the encrypted message C as,
y=ycod(C) ; where ycod is the function to compute the y co-ordinate.

2. Compute Φ, where $\Phi=x*y$

3. Identify the mapped point P by calculating $\{(P_m+(r*sk)) \Phi)\}/upk$

4. The original message M is extracted by un-mapping the point P .

3.5 Steps in SCDS-TM model

Let us assume that the data File F is divided into various data blocks F_1, F_2, F_3………F_n; where $n_i \in Z_p$ and p is always a prime. The owner creates his own data and encrypt the data by means of an elliptic curve cryptography using the Encryption () algorithm. After encrypting the data, the owner creates the index and the signature for those respective data blocks by means of IndexGen () and SigGen () algorithms respectively.

1. The encrypted data -C, index -τand the signatures -ק are hosted in the cloud storage servers. The keys which meant for encryption are stored in the cloud key servers.

2. The Trust manager challenges the server using chalmessage for storage correctness.

3. The cloud server in turn generates the data blocks and signature blocks and also a proof for the challenged blocks and sends this proof to the TM .Next, the proof which was generated by the cloud server was validated by the data owner using ProofVerf() algorithm and updated using ExecuteUpdate() algorithm by the cloud server and it is the task of the TM to executeProveUpdate () algorithm and returns the result as either true or false.

4. When the user requests to the owner, for the data.

5. Upon receiving the request, it is in-turn sent to the Trust Manager(TM) for checking the authenticity of the user and the authenticity of the block tags are verified by running the appropriate MHT algorithm.

6. The TM checks the authenticity of the user.

7. If authenticated, sends a token to the user and the secret key used to decrypt the message; else the request is rejected.

8. Through a proper access control mechanisms, the user gets the access rights from the owner and sends a search request or a Trapdoor to the cloud server.

9. Upon receiving the trapdoor, the server performs the ranked searching on the relevant documents based upon the index and returns all the top k-relevant encrypted documents.

10. By means of the secret key received from the TM, the user decrypts the encrypted documents and the accurate data is obtained.

3.6 Protocol Design of SCDS-TM Model

The data flow diagram of our proposed framework is explained in this section. As illustrated in Fig.3, the storage protocol consists of three phases: Owner Initialization, Data Confirmation, Verification or Auditing.

Phase 1: Owner Initialization

During the system setup phase, the owner generates the keys for encrypting the data and also computes the index and the signature for those respective data blocks.

Phase 2: Data Storage Confirmation

Once the data is stored in the cloud server, the data owner seeks the trust manager to accomplish the confirmation (auditing) to guarantee that the data stored in the cloud is correct. After confirmation is made, the owner erases the data from its local storage.

The auditing phase in our construction is two-way communication as follows:

1. The trust manager invoke the chal() algorithm and generate a chal message for the data blocks in the data file F and sends the chal{(i, u_i)} $s_{1 \leq i \leq s_c}$; to the cloud server .

2. After receiving the chal message from the trust manager, the server computes the data blocks and the signature blocks and generates a proof and sends it to the trust manager.

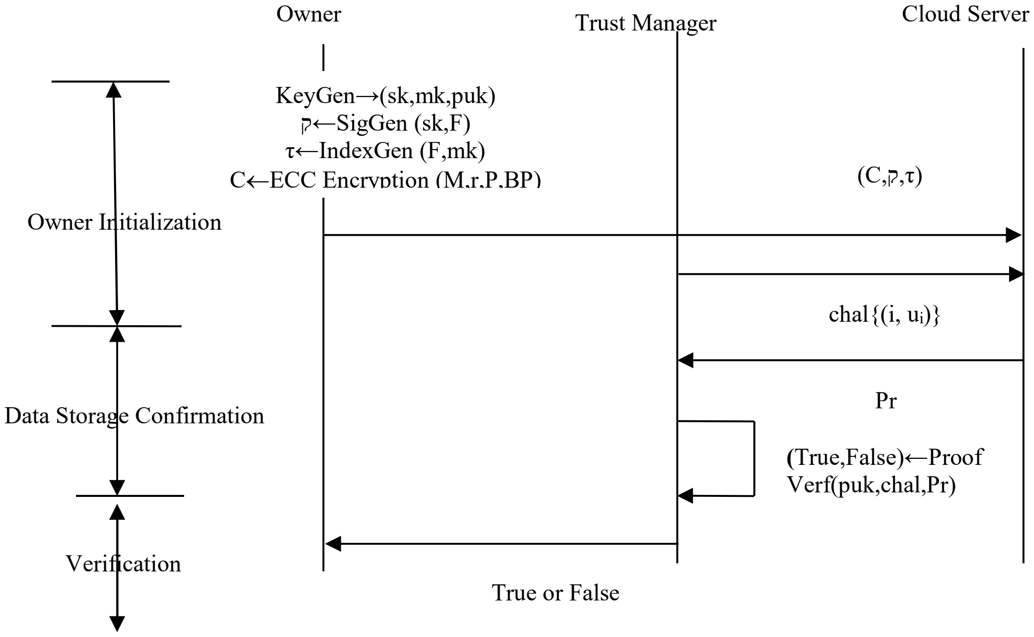


Fig.3: Framework of the Auditing Protocol in our proposed model

Phase 3: Verifying

Upon accepting the proof, the trust manager verifies the proof by means of invoking the ProofVerf() algorithm and return the auditing result as either True or False. The Trust Manager sends these auditing results to the data owner and convinces him that the data is stored correctly. The trust manager carry out the verification periodically by challenging some data blocks in the file F.

3.7 Algorithms for the proposed SCDS-TM model

The algorithms executed by the owner, user, the authentication server and the trust manager in our proposed model are discussed in this section.

Algorithms invoked by the Owner:

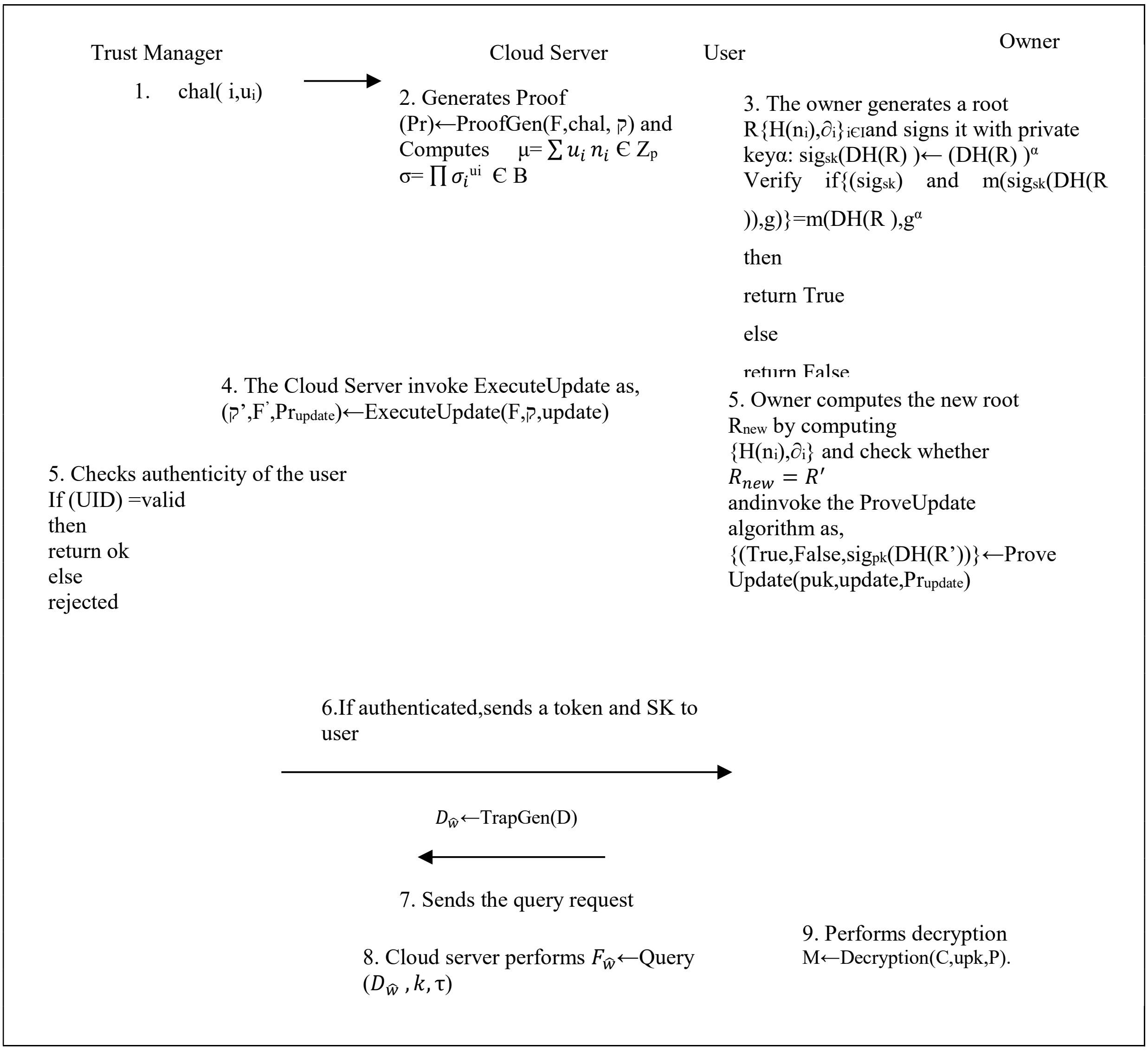
By means of KeyGen() function, the owner's public key, private key and master key is generated. The data file F is preprocessed and the metadata is produced. Initially, the owner takes the input security parameter 1^k and returns the public key puk, secret key sk and master key mk. For generating the index and the signature, the owner takes the input, the master key mk, the secret key sk and the File f which consists of numerous blocks *bi* and returns the ordered signature collection B_i on b_i. The owner outputs the metadata in the form of signature SigSK(D(R)) of the root R of a Distribution tree. Next, the owner generates the index by taking the input. For validation verification, the public key puk, signature SigSK(D(R)) and a request 'True' or 'False'. After verifying successfully, a signature SigSK(D(R`)) with a value of 'True' is returned. Otherwise a 'False' value is returned for old root R.

Algorithm1:Security Validation

```
Step1:Security parameter $1^k$
Begin
  If ($1^k$)
{
 (puk,sk,mk)←KeyGen($1^k$)
return public key puk
return secret key sk
return master key mk
}

Step2:ECC Encryption
Input: Message M, the curve base point BP, a random number r, owner's secret key sk and a curve point P.
Output: The cipher text C.
Step2.1: Initially, the message M is mapped with curve point P.
Step2.2: A random number is chosen such that r€ [1,n-1]
Step2.3: The cipher text is obtained as,C ={(r*BP)+($P_m$+(r*sk))}

Step3:  Tag Generation
Input : The cipher text C and the owner's secret key sk
Output: Tag T
Step3.1: Choose an element $y \leftarrow B$ and generates the tag T
Step3.2: The tag includes$name||n||y||SSig_{sk}(name||n||y)$

Step4: Index Generation
Input: Generates a binary data vector based on data file $F_i$.
Output: sub-index τ
Step4.1: The sub-index τ =$\{A_1^T D_i^{'}+A_2^T D_i^{''}\}$

Step5: Signature Generation
Input: Owner's secret key sk, the File F
Output: Signature Set S
Step4: If R is the root
then
and signs it with private key$\alpha$: $sig_{sk}(DH(R)) \leftarrow (DH(R))^{\alpha}$
and Verify if{($sig_{sk}$) and m($sig_{sk}$(DH(R )),g)}=m(DH(R ),$g^{\alpha}$

then

return True

else

return False
```

Algorithms invoked by the Trust Manager, Cloud Server and the User

The algorithms invoked by the trust manager, cloud server and the user are discussed below:



1. chal(i,u_i)

2. Generates Proof
(Pr)←ProofGen(F,chal, ק) and
Computes $\mu=\sum u_i n_i \in Z_p$
$\sigma=\prod \sigma_i^{ui} \in B$

3. The owner generates a root
R{H(n_i),∂_i}$_{i\in I}$and signs it with private
keyα: sig_{sk}(DH(R))← (DH(R))$^\alpha$
Verify if{(sig_{sk}) and m(sig_{sk}(DH(R
)),g)}=m(DH(R),g$^\alpha$
then
return True
else
return False

4. The Cloud Server invoke ExecuteUpdate as,
(ק',F',Pr_{update})←ExecuteUpdate(F,ק,update)

5. Owner computes the new root
R_{new} by computing
{H(n_i),∂_i} and check whether
$R_{new} = R'$
andinvoke the ProveUpdate
algorithm as,
{(True,False,sig_{pk}(DH(R')))}←Prove
Update(puk,update,Pr_{update})

5. Checks authenticity of the user
If (UID) =valid
then
return ok
else
rejected

6.If authenticated,sends a token and SK to
user

$D_{\hat{w}}$←TrapGen(D)

7. Sends the query request

8. Cloud server performs $F_{\hat{w}}$←Query
($D_{\hat{w}}$, k, τ)

9. Performs decryption
M←Decryption(C,upk,P).

Fig. 4: Algorithms invoked by the server, owner, user and the TM

4.0 EXPERIMENTAL EVALUATIONS

The proposed SCDS framework is compared with the existing schemes such as erasure coded, DPDP scheme MRSE techniques, and various metrics. We have worked out with different execution results which helped us to demonstrate our framework with better results.

4.1 Lab Setup

We analyzed the security of our proposed framework with that of the existing systems such as erasure coded system [16], Dynamic Provable Data Possession (DPDP) [7] scheme and Multi-Ranked Search Encryption (MRSE) [18] techniques. Implementation is done in the Java platform. The user side is implemented on a workstation with an Intel core 2 processor running at 1.86 GHz, 2048 MB of RAM and at a 7200 RPM Serial ATA drive. The cloud server side is implemented on CloudSim software with larger instant type, 7.5 GB memory, and 850 GB instant storage.

4.2 Performance Analysis

The SCDS-TM model not only enables data storage and data retrieval, but also reduces the communication cost of the owner. The performance of the proposed SCDS-TM model is evaluated using various metrics which includes:

- Communication Overhead
- Data Transfer Rate
- Query Execution Time
- Data Security Level
- Probability Ratio
- Precision Ratio

4.2.1 Communication overhead: Communication overhead deals with the server's response to the challenge given. Here, the communication overhead is compared with erasure coded scheme, DPDP, and MRSE. The communication overhead is calculated in terms of KB/ms.The communication overhead is computed as,

Communication Overhead (KB/ms) =Amount of challenged data sent (KB) x delay taken by the server to respond (ms) Eq.(4.1)

Using Eq.(4.1), the Communication overhead of the existing schemes and the proposed SCDS-TM model is computed and the tabulation is shown in Table.2.

Table 2: Comparison chart for File Size vs Communication Overhead

File Size(KB)	Communication Overhead (KB/ms)			
	Existing Schemes			Proposed SCDS-TM Model
	Erasure coded Scheme	DPDP scheme	MRSE scheme	
50	0.35	0.34	0.30	0.28
100	0.45	0.46	0.44	0.43
150	0.42	0.43	0.40	0.35
200	0.46	0.45	0.45	0.42
250	0.55	0.54	0.52	0.50
300	0.54	0.55	0.52	0.50
350	0.68	0.61	0.63	0.60
400	0.62	0.65	0.63	0.61
450	0.69	0.67	0.64	0.62
500	0.73	0.75	0.72	0.71

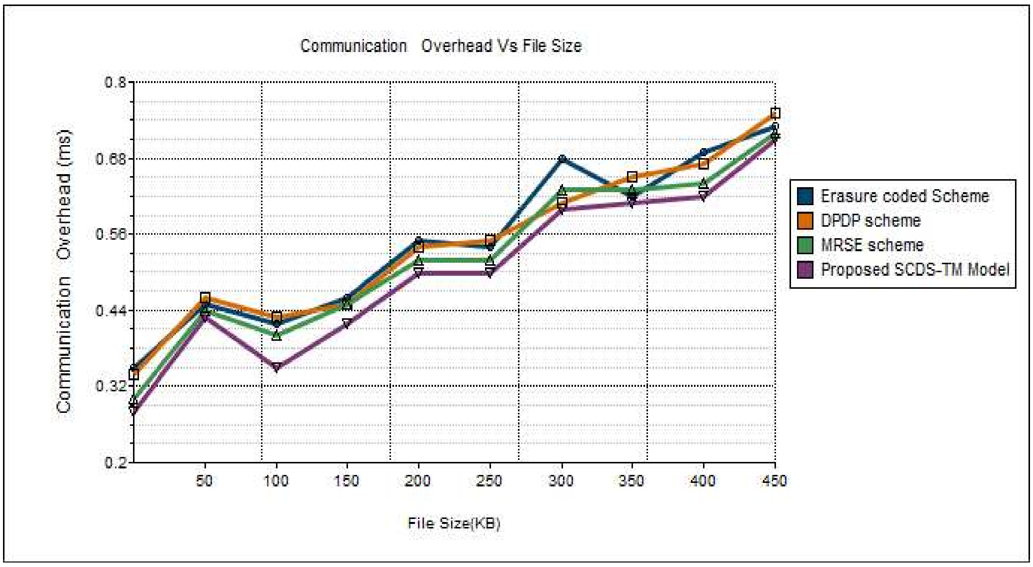


Fig.5:Performance graph ofBlock Size vs Communication overhead

From Fig.5, it is observed that the communication overhead of the proposed SCDS-TM is highly reduced because the Trust Manager (TM) sends the challenge message to the cloud server without the direct involvement of the data owner. In MRSE technique, multiple keywords are sent to the cloud server which costs an additional burden to the server. Hence the communication overhead of the proposed SCDS-TM model is reduced by 4% when compared to other existing schemes like erasure coded, DPDP and MRSE techniques.

4.2.2 Data Transfer Rate: The data transfer rate is defined as transferring the data from the cloud server to user in a given time interval. The data transfer rate of the SCDS-TM model is compared with that of erasure coded technique, DPDP and MRSE techniques.It is measures in terms of KB/ms

$$\textit{Data transfer rate (KB/ms)} = \frac{Data\ transferred\ by\ the\ cloud\ server\ to\ the\ user\ (KB)}{Time\ taken\ (ms)} \text{Eq.(4.2)}$$

Using Eq.(4.2), the data transfer rate of the SCDS-TM model is compared with the existing schemes and the tabulation is shown in Table 3.

Table 3: Comparison chart for File Size vs Data Transfer Rate

File Size(KB)	Data Transfer Rate(KBps)			
	Existing Schemes			Proposed SCDS-TM Model
	Erasure coded scheme	DPDP scheme	MRSE scheme	
50	40	46	42	49
100	44	44	46	48
150	52	52	54	55
200	63	64	65	68
250	62	65	67	70
300	71	73	74	77
350	74	72	75	81
400	88	83	82	88
450	92	91	95	105
500	112	117	121	125

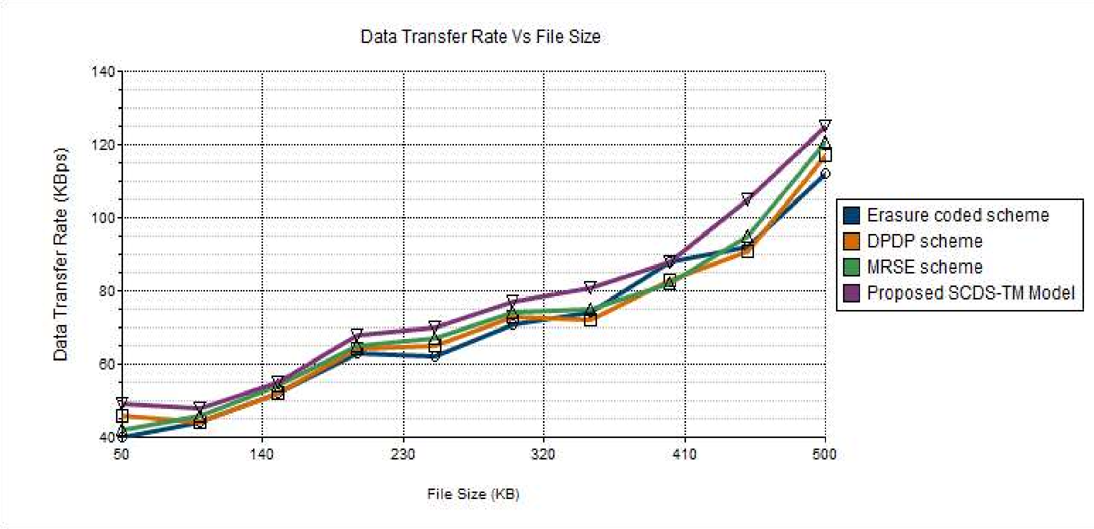


Fig.6: Performance graph of File Size vs Data Transfer Rate

Fig.6 shows the data transfer rate of the existing schemes and the proposed SCDS-TM model. In SCDS-TM model, the data requested by the user is retrieved by the cloud server and performs the search based on the keywords and the index. Only the exact data requested by the user are sent to the userwhile the remaining data are kept confidential. The proposed SCDS-TM model provides a better data transfer rate at 7% above the existing schemes.

4.2.3 Query Execution Time: This explains the time taken to execute a query/request.When users request /queries the cloud server , the time taken by the cloud server to execute the appropriate query and with the help of the keywords, it search the related documents. Based on Fig. 7, the query execution time in term of Block Size is compared with the DPDP scheme.The execution time (in millisecond) is computed as, *Execution Time (ms)=Time taken by the user to send the query(ms) -Time taken by the server to respond to query(ms)* Eq.(4.3)

Using Eq.(4.3), the execution time of the SCDS-TM model is computed and compared with other existing schemes and it is shown in Table 4.

Table 4: Comparison chart for File SizevsQuery Execution Time

File Size(KB)	**Query Execution Time(ms)**			
	Existing schemes			**Proposed SCDS-TM Model**
	Erasure coded scheme	**DPDP scheme**	**MRSE scheme**	
50	10	10	6	5
100	11	12	10	10
150	13	15	11	11
200	15	18	14	12
250	18	20	15	14
300	24	26	22	20
350	28	26	25	24
400	35	36	33	31
450	44	45	43	41
500	49	46	45	43

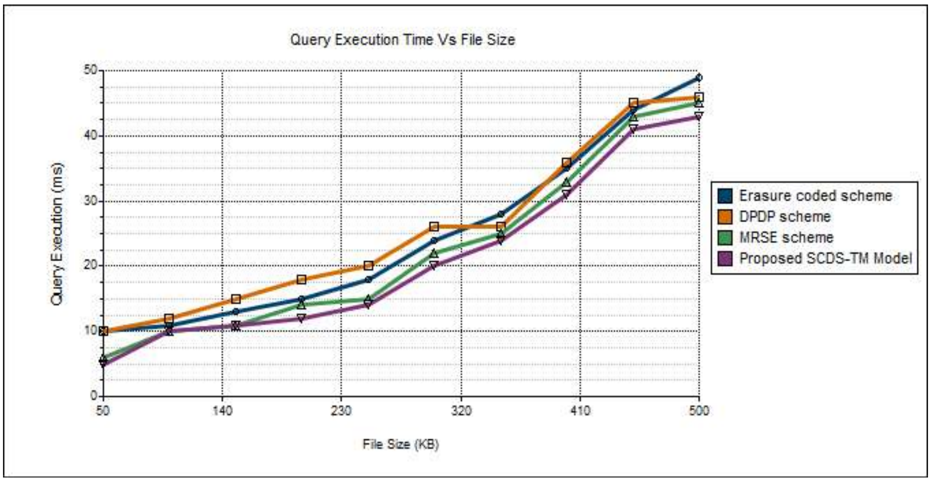


Fig.7: Performance graph of File Size vs Query Execution Time

It is observed from Fig.7, that the query execution time of the proposed SCDS-TM model is low as it uses the keyword based searching technique. When the user requests the data, the cloud server performs searching based upon the appropriate keyword and hence the query execution time is reduced by 3%.

4.2.4 Data Security Level: The data security level is the amount of cloud data securely sent to the clients by the server to that of total data requested by the clients. The data security level is computed in percentage (%). The data security level is calculated as,

$$Data\ security\ level = \frac{Amount\ of\ data\ successfully\ sent\ to\ the\ clients}{Total\ data\ requested\ by\ the\ clients}\ x\ 100 \text{Eq.}(4.4)$$

Using Eq.(4.4), the data security level is computed and the performance analysis of the data security level based on different data in terms of KB is illustrated in Table 5.

Table 5: Comparison chart for File Size and Data Security level

File size (KB)	Data security level (%)			
	Existing Schemes			Proposed SCDS-TM Model
	Erasure coded scheme	DPDP scheme	MRSE Technique	
10	55.35	54.34	57.22	61.34
20	57.68	55.13	58.29	63.74
30	58.20	56.49	58.30	64.55
40	59.05	57.28	62.37	65.45
50	62.83	60.21	63.71	66.42
60	64.26	62.55	65.68	67.22
70	66.39	63.27	69.10	70.42
80	67.02	65.72	70.24	72.11
90	68.75	66.47	73.61	75.81
100	72.46	71.53	74.16	77.58

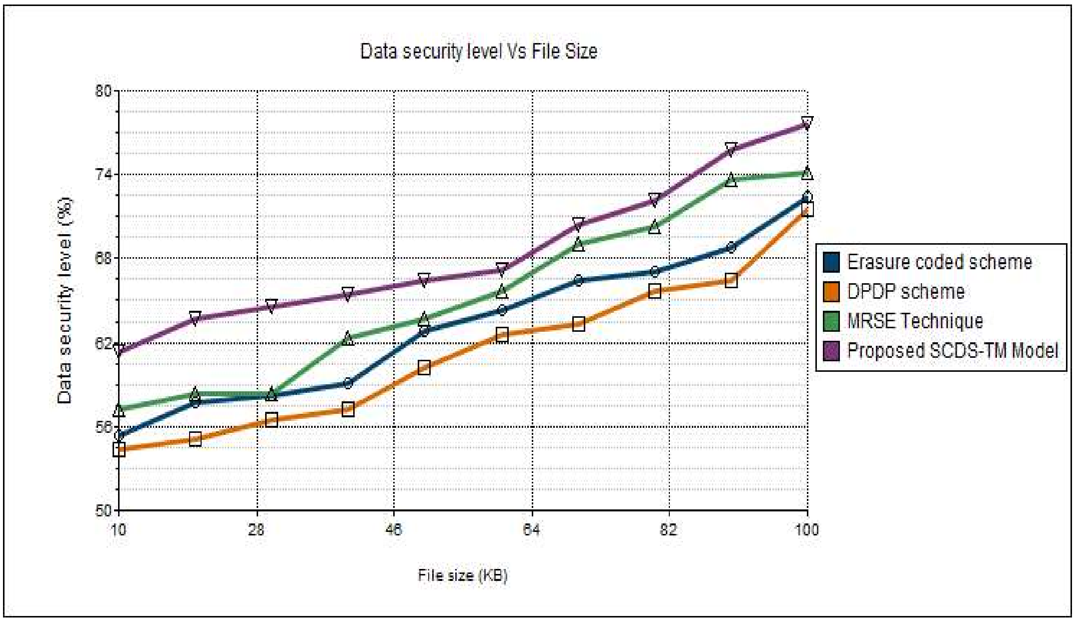


Fig.8: Performance graph for File Size vs Data Security level

Fig.8. shows the data security level of the proposed SCDS-TM model shows a significant improvement of 5% when compared to other existing schemes of erasure coded, DPDP and MRSE techniques.

4.2.5 Probability Ratio: Probability ratio is definedas the action of successfully retrieving the messages. It is measured in percentage. In erasure coded technique, due to the random selection of the storage servers and the key servers and also selecting the random coefficients by the storage servers and the key servers makes the technique a little harder and therefore, the probability mainly depends on these four factors.The proposed SCDS-TM is compared with erasure coded, DPDP and MRSE technique using Eq.(4.5) and it is shown in Table 6.

$$\textit{Probability Ratio (\%)}=\frac{\textit{Number of keywords in the query request}}{\textit{Total number of keywords in the server}} \text{x } \textit{100} \qquad \text{Eq. (4.5)}$$

Table 6: Comparison chart of Number of keywords vsProbability Ratio

Number of Keywords in the Query/Request	Probability Ratio (%)			
	Existing Schemes			Proposed SCDS-TM Model
	Erasure coded scheme	DPDP scheme	MRSE Technique	
10	50.15	53.11	55.63	57.23
20	52.25	54.26	56.27	59.54
30	61.21	63.56	64.37	65.48
40	62.32	63.41	65.04	66.15
50	65.29	64.34	66.10	68.38
60	72.23	73.34	74.47	75.01
70	74.52	75.61	76.59	77.20
80	78.53	75.54	80.52	81.29
90	82.43	79.41	83.43	84.63
100	83.29	80.81	84.26	85.15

As the number of keywords in the search request increases, the probability ratio of retrieving the message in MRSE isaverage when compared to our proposed SCDM-TM method as the probability ratio is greater and hence the accurate message is retrieved.

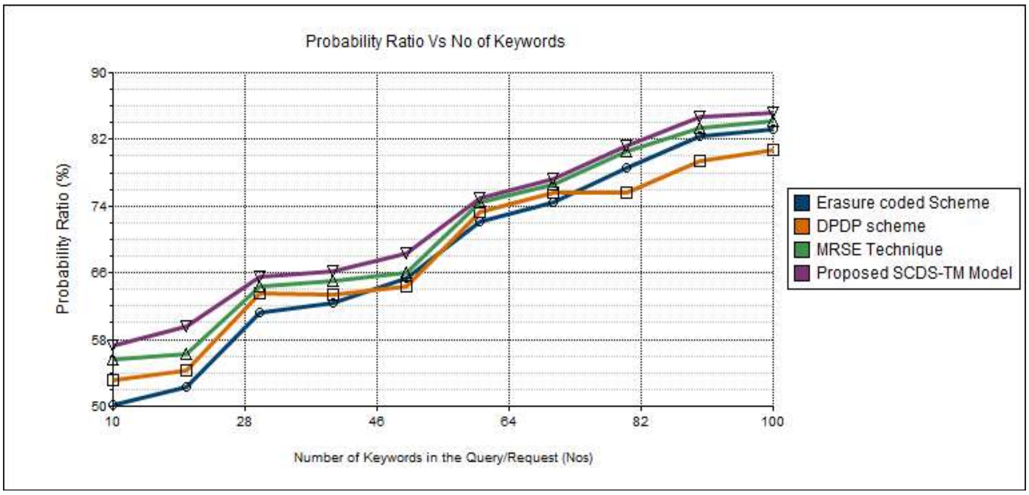


Fig.9: Performance graph of Number ofKeywords vs Probability Ratio

In the proposed SCDS-TM model, since the data file is identified by means of its index and tag, when the user sends search query to retrieve the data file, based upon the coordinate matching technique the top –k documents relevant to that query is captured accurately and hence the probability ratio of the SCDS-TM model offers a better probability ratio of 6% when compared to that of the existing system

4.2.6 Search Precision: When calculating the similarity score of the documents, dummy keywords are sometimes inserted into the data vectors. When the server returns the relevant top k documents based upon the search query, some of the top k-documents may be excluded due to their decrease in the real similarity scores, or the similarity scores of some of the documents may be increased. Hence the similarity scores may not be accurate sometimes and hence a precision value is calculated to evaluate the purity of the k documents recovered by the user.

The search precision is calculated as,

$$\text{Precision Ratio (\%)} = \frac{Relevant\ documents\ retrieved\ by\ the\ user}{Total\ documents} \text{x100} \qquad \text{Eq. (4.6)}$$

Using Eq.(4.6), the search precision ratio for the existing schemes like erasure coded , DPDP and MRSE technique were computed and are compared with the proposed SCDS-TM model and the tabulation is shown in Table 7.

Table 7: Comparison chart for Number of retrieved documentsvsPrecision

Number of retrieved documents	Precision (%)			
	Existing Schemes			Proposed SCDS-TM Model
	Erasure coded scheme	DPDP scheme	MRSE Technique	
50	35.17	38.23	41.19	43.11
100	42.46	45.42	52.15	55.31
150	53.29	54.46	56.37	58.40
200	60.43	61.47	63.82	65.44
250	66.70	65.52	67.28	69.75
300	71.20	70.02	72.65	73.33
350	72.31	73.82	74.33	75.28
400	77.40	74.67	78.15	80.10
450	82.64	83.19	84.66	85.19
500	86.27	87.58	89.43	90.47

For MRSE technique, the search precision may not be accurate as the searching is performed based only on the search queryas shown in Table.7.But in the proposed SCDS-TM model, the searching is based on the search query, index, and the tag generated for all the documents present in the file,increasing the search precision and hence the precision ratio is high.

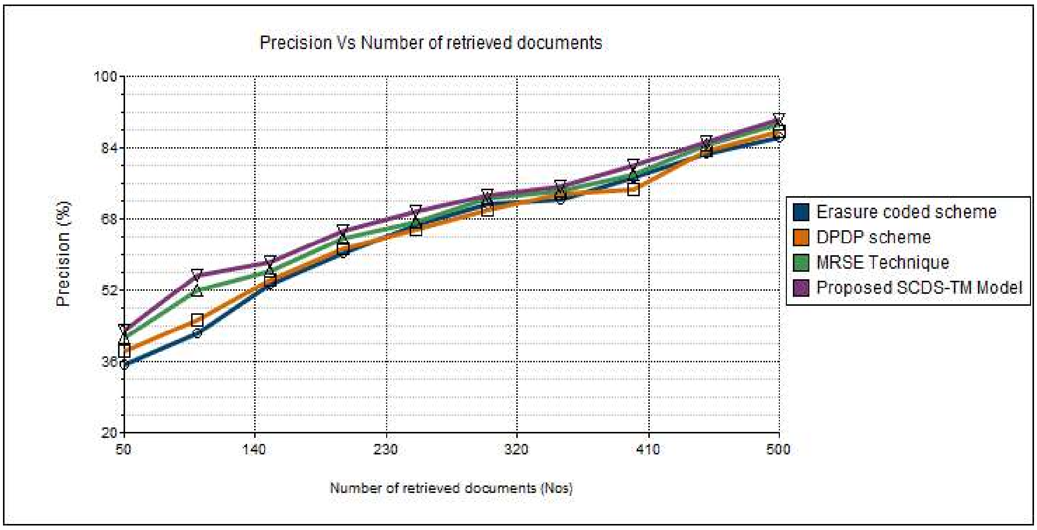


Fig.10:Number of retrieved documents vs Precision ratio

It is observed from Fig.10, that the proposed SCDS-TM model shows a better precision ratio of 5% when compared with the existing systems.

5.0 CONCLUSION

In this paper, we have proposed a new security structure for cloud storage system in the cloud computing environment which includes the ECC file encryption system, tag generation, signature generation, ECC decryption system, and index generation for enhancing the overall security and the performance of the system.

To reinforce the security aspects, the proposed SCDS-TM model also tackles four important issues;.1) Data confidentiality, 2) Data integrity 3) Support of dynamic data operations, and 4) Data retrieval. Data confidentiality arises one of the major issue in cloud environment. Data confidentiality is supported by means of elliptic curve algorithm. The data stored in the cloud is dynamic and can be altered and therefore to address data integrity problem, integrity verification of the stored data is implemented.Moreover, it also supports dynamic data operations.

Another major concern in the model is that, by means of keyword searching, the relevant data with that keyword alone is retrieved to the user and hence the other confidential data is hidden from the user. Apart from this, our proposed method reduces the communication burden of the data owner as the data is retrieved from the cloud server by sending a search query/request by the user without the direct command of the owner, which is mislaid in all the prior works.

By means of experimental evaluations, it is clear that our proposed method outperformed the existing methods with respect to query execution time, communication overhead, probability ratio and also in security level.

REFERENCES

[1] Dimitrios Zissis., DimitriosLekkas., “Addressing cloud computing security issues,” *Future Generation Computer Systems, Elsevier*, Vol.28, No.3, 2012, pp: 583-592.

[2] Lifei Wei ,HaojinZhu,ZhenfuCao,XiaoleiDong,WeiweiJia,YunluChen,AthanasiosV.Vasilakos, “Security and Privacy for storage and computation in cloud computing”, Information Sciences , Elsevier,Vol.258,2013,pp:371-386.

[3] S. Subashini. , V. Kavitha., “A survey on security issues in service delivery models of cloud computing,” Journal of Network and Computer Applications, Elsevier , Vol.34, No.1, 2010,pp :1-11.

[4] Xiao X and Xiao Y, “Security and privacy in cloud computing”, IEEE Communications Surveys and Tutorials”, Volume.15, Number.2, 2013, pp.843-859.

[5] Hans P. Reiser., “Byzantine Fault Tolerance for the Cloud,” IBM Zurich Research, 2011.

[6] G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner,Z. N. J. Peterson, and D. X. Song, “Provable Data Possession at untrusted stores,” in ACM Conference on Computer and Communications Security, ACM, 2007, pp:598–609.

[7] C.Erway,A.Kupcu,C.Papamanthou,andR.Tamassia, “Dynamic Provable Data Possession”,Proc.16thACM Conference on Computer and Communication Security(CSS ’09),2009, pp:213-222.

[8] Yan Zhu,HongxinHu,”Cooperative Provable Data Possession for Integrity Verification in Multi-cloud storage”, IEEE Transactions on Parallel and Distributed Systems,Vol.23, No.12, 2011, pp: 1–14.

[9] Zhang J, Tang W, Mao J, “Efficient public verification proof of retrievability scheme in cloud”, Cluster Computing, Volume.17, Number.4, 2014, pp.1401-1411.

[10] Qian Wang, Cong Wang, KuiRen ,Wenjing Lou and Jin Li, “ Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing”, IEEE Transactions on Parallel and Distributed Systems,Vol.22,No.5,2011, pp: 847-859.

[11] Kan Yang, XiaohuaJia , “An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing”, IEEE Transactions on Parallel and Distributed Systems, Vol.24, No.9, 2013,pp:1717-1726.

[12] Hong Liu, HuanshengNing ,QingxuXiong, and Laurence T.Yang, “Shared Authority Based Privacy-Preserving Authentication Protocol in Cloud Computing”, IEEE Transactions on Parallel and Distributed Systems,Vol.26,No.1,2015,pp:241-251.

[13] Yang Tang, Patrick P.C.Lee,JohnC.S.Lui and Radia Perlman, “Secure Overlay Cloud Storage with Access Control and Assure Deletion”, IEEE Transactions On Dependable and Secure Computing, Vol.9,No.6,2012,pp: 903-916.

[14] Po-Wen Chi and Chin-Laung Lei, “Audit-Free Cloud Storage via Deniable Attribute-based Encryption”, IEEE Transactions on Cloud Computing, No.99, 2015,pp:1-14.

[15] Hsiao-Ying Lin and Wen-Guey, “A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding”, IEEE Transactions on Parallel and Distributed Systems,Vol.23, No.6, 2012, pp:995-1003.

[16] Mazhar Ali, RevathiDhamotharan, ErajKhan,SameeU.Khan, Athanasios V.Vasilakos,Keqin Li and Albert Y.Zomaya, “SeDaSC: Secure Data Sharing in Clouds”. IEEE Systems Journal, No.99, 2015, pp:1-10.

[17] Amol A.Dhumal, Dr. Sanjay Jadhav, “Confidentiality-Conserving Multi-Keyword Ranked Search above Encrypted Cloud Data”, International conference on Communication, Computing and Virtualization ,Procedia Computer Science, ScienceDirect, Vol.79, 2016,pp:845-851.

[18] Ning Cao, Cong Wang,MingLi,Kui Ren and Wenking Lou, “Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data”, IEEE Transactions on Parallel and Distributed Systems,Vol.25,No.1, 2014, pp:222-233.

[19] Smitha Sundareswaran, Anna C. Squicciarini, and Dan Lin, “Ensuring Distributed Accountability for Data Sharing in the Cloud”, IEEE Transactions on Dependable and Secure Computing, Vol. 9, No. 4, 2012, pp:556-568.

[20] Thomas Brindha and RamaswamyShaji, “An Instance Communication Channel Key Organizer Model for Enhancing Security in Cloud Computing”, The International Arab Journal of Information Technology ,Vol.13, No.5,2016,pp:509-516.

[21] Mazhar Ali, Kashif Bilal, Samee U.Khan, BharadwaiVeeravalli, Keqin Li, and AlbertY.Zomaya, “DROPS: Division and Replication of Data in Cloud for Optimal Performance and Security”, IEEE Transactions on Cloud Computing, No.99, 2015,pp:1-14.

[22] Raj, R.G. and Balakrishnan, V. “A Model For Determining The Degree Of Contradictions In Information” Malaysian Journal of Computer Science, Vol. 24(3): September 2011. pp 160-167.

[23] Baojiang Cui, Zheli Liu and Lingyu Wang , “Key-aggregate Searchable Encryption(KASE) for Group Data Sharing via Cloud Storage”, IEEE Transactions on Computers,Vol.65,No.8, 2016,pp:2374-2385.

[24] Atiqa Qazi, Ram Gopal Raj, Muhammad Tahir, Mahwish Waheed, Saif Ur Rehman Khan, and Ajit Abraham, “A Preliminary Investigation of User Perception and Behavioral Intention for Different Review Types: Customers and Designers Perspective,” The Scientific World Journal, vol. 2014, Article ID 872929, 8 pages, 2014. doi:10.1155/2014/872929.

[25] BoyangWang,Baochun Li and Hui Li, “Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud”, IEEE Transactions on Cloud Computing,Vol.2,No.1, 2014,pp:43-56.

[26] G. Wanga, Q. Liu, J. Wu and M. Guo, “Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers”, Computers and Security, Elsevier, Vol.30, No.5, 2011, pp:320-331.

[27] Seung-Hyun Seo,MohamedNabeel,XiaoyuDing,and Elisa Bertino, “An Efficient Certificate Encryption for Secure Data Sharing in Public Clouds”, IEEE Transactions on Knowledge and Data Engineering,Vol.26,No.9, 2014,pp:2107-2119.

[28] Xuefeng Liu, Yuqing Zhang, BoyangWang,andJingbo Yan, “Mona :Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud”, IEEE Transactions on Parallel and Distributed Systems,Vol.24,No.6, 2013,pp:1182-1191.

[29] Zhongma Zhu and Rui Jiang, “A Secure Anti-Collusion Data Sharing Scheme for Dynamic Groups in the Cloud”, IEEE Transactions on Parallel and Distributed Systems, Vol.27, No.1,2016, pp:40-50.

[30] Dongyoung Koo, JunbeomHur, Hyunsoo Yoon, “Secure and efficient data retrieval over encrypted data using attribute-based encryption in cloud storage”, Computers & Electrical Engineering,ScienceDirect,Vol.39,No.1, 2013,pp:34-46.

[31] Jiadi Yu, Peng Lu ,GuangtaoXue, “ Towards Secure Multikeyword Top-k Retrieval over Encrypted Cloud Data”, IEEE Transactions on Dependable and Secure Computing,Vol.10,No.4, 2013,pp:239-250.

[32] Thomas Brindha ,RamaswamySwarnammalShaji, “ A secure transaction of cloud data using conditional source trust attributes encryption mechanism”, Soft Computing, Springer, DOI: 10.1007/s00500-016-2405-6, 2016.