International Journal of Advance Research in Science and Engineering
Volume No 06, Special Issue No. (02), September 2017, ICITTESE-17
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Comparison between 2PC, D2PC and SD2PC Algorithm for Transactional Security in Cloud

Shilpi Chaubey[1],Sonali Katahlia[2],Priyanka Sao[3]

Assistant Professor, CSE Dept, Columbia Institute of Engg & Tech, Raipur, India

ABSTRACT

Cloud computing is a reflection in view of the thought of pooling physical assets and displaying them as a virtual asset. It is another model for provisioning assets, for arranging applications, and for stage autonomous client access to administrations. Many are on the social networking sites where it is required to give some personal as well as professional information of one. This information is somewhat and somewhere stored on to the database in the form of data. These data should be secured and information should not be loss in one or the other way. In this project the main work is to protect the data of users. The data which are transacted from the cloud storage area should also be done in more safely manner without giving harm to anyone. Transactional Security should be maintained for cloud transaction in order to protect the transactions. In this paper we are comparing different algorithm i.e. 2PC algorithm, D2PC algorithm and SD2PC algorithm and showing the result regarding the different algorithm in which it has been examined that Secured Dynamic Two Phase Commit (SD2PC) algorithm is the best one used for providing the security of the cloud transaction using the best providing the security key which is generated at the time of sharing. This algorithm is used for providing security and also it is time optimal algorithm. By the use of this algorithm the transactional security is maintained. The user who knows the security key can only access the files and can download it In this thesis new algorithm Secured Dynamic Two Phase Commit (SD2PC) algorithm is proposed which is the modified version of the Dynamic Two Phase Commit (D2PC) algorithm which is used to maintain the security for cloud transaction activity.

Keyword: cloud computing, data storage, 2PC algorithm, D2PC algorithm, cloud transaction.

I. INTRODUCTION

Cloud computing is utilized for the administrations and applications which keep running on an appropriated system utilizing virtualized assets and got to by basic Web conventions and systems administration guidelines. It is separated on the premise of idea where assets are virtual and boundless and that subtle elements of the physical frameworks on which programming runs are inattentive from the client.

The bigger sizes of cloud computing frameworks are empowered on account of step by step utilization of the Web and the developing of diverse substantial administration organizations. Cloud computing makes the long-held long for utility figuring conceivable with a pay-as-you-go, endlessly adaptable, all around accessible framework. With cloud computing, we can begin little and turn out to be enormous quick. That is the reason distributed computing is progressive, regardless of the fact that the innovation it is based on is transformative.

WHAT IS CLOUD COMPUTING?

Cloud computing takes the innovation, administrations, and applications that are like those on the Web and transform them into a self-administration utility. The utilization of "cloud" makes reference to the two vital ideas:

- **Abstraction:** Cloud computing modified works the subtle elements of framework usage from clients and designers. Applications keep running on physical frameworks that aren't determined, information is put away in areas that are obscure, organization of frameworks is outsourced to others, and access by clients is omnipresent.
- **Virtualization:** Cloud computing virtualizes frameworks by pooling and sharing assets. Frameworks and capacity can be provisioned as required from a unified foundation, expenses are evaluated on a metered premise, multi-tenure is empowered, and assets are adaptable with agility.

To help clear up how cloud computing has changed the way of business framework arrangement, consider these three cases:

- **Google:** In the most recent decade, Google has assembled an overall system of server farms to administration its web search tool. In doing as such Google has caught a significant segment of the world's promoting income. That income has empowered Google to offer free programming to clients in light of that base and has changed the business for client confronting programming. This is the fantastic Software as an administration.

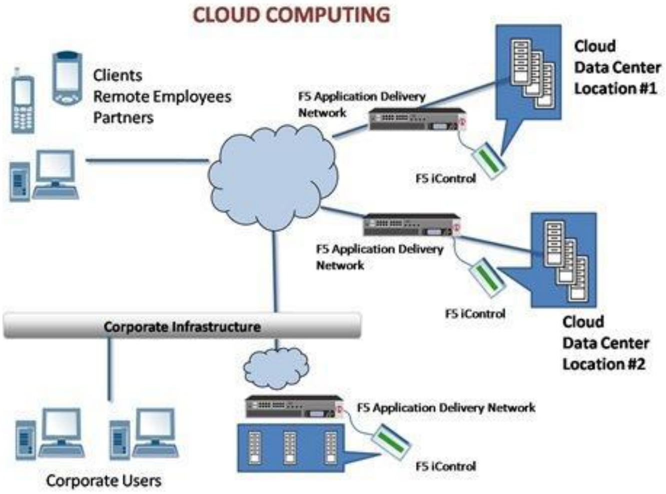

Fig 1: Cloud Computing

Azure Platform: By difference, Microsoft is making the Azure Platform. It empowers .NET Framework applications to keep running over the Internet as a substitute stage for Microsoft designer programming running on desktops.

- **Amazon Web Services:** One of the best cloud-based organizations is Amazon Web Administrations, which is an Infrastructure as a Service offering that gives you a chance to lease virtual PCs all alone foundation.

These new capacities empower applications to be composed and conveyed with insignificant cost and to be quickly scaled and made accessible worldwide as business conditions grant. This is genuinely a progressive change in the way venture processing is made and sent [2].

TWO PHASE COMMIT PROTOCOLS

in transaction processing, databases, and computer networking, the **two-phase commit protocol** (**2PC**) is a type of atomic commitment protocol (ACP). It is a distributed algorithm that coordinates all the processes that participate in a distributed atomic transaction on whether to *commit* or *abort* (*roll back*) the transaction (it is a specialized type of consensus protocol). The protocol achieves its goal even in many cases of temporary system failure (involving process, network node, communication, etc. failures), and is thus widely used. However, it is not resilient to all possible failure configurations, and in rare cases, user (e.g., a system's administrator) intervention is needed to remedy an outcome. To accommodate recovery from failure (automatic in most cases) the protocol's participants use logging of the protocol's states. Log records, which are typically slow to generate but survive failures, are used by the protocol's recovery procedures. Many protocol variants exist that primarily differ in logging strategies and recovery mechanisms. Though usually intended to be used infrequently, recovery procedures compose a substantial portion of the protocol, due to many possible failure scenarios to be considered and supported by the protocol.

DISTRIBUTED TRANSACTIONAL PROTOCOLS

The exchange is the conveyed sort of exchange in which distinctive frameworks are joined to one another with a specific end goal to perform the disseminated exchange. A circulated exchange is a kind of exchange with two or more drew in system has. By and large, has give assets, and an exchange administrator is in charge of creating and taking care of the exchange. Like some other exchange, an appropriated exchange ought to incorporate each of the four ACID properties (atomicity, consistency, isolation, given the way of the work, atomicity is essential to guarantee a win big or bust result for the operations group (unit of work). Databases are standard value-based assets, and exchanges as a rule reach out to a little number of such databases. In such cases, a disperses exchange may be seen as a database exchange that ought to be synchronized between different partaking databases distributed between different physical areas. The detachment property introduces a remarkable snag for multi-database exchanges.

DYNAMIC TWO PHASE COMMIT (D2PC) PROTOCOL

The Dynamic two-phase commit protocol is a variation of Tree 2PC with no foreordained organizer. Understanding messages are sent by every endless supply of the exchange (getting to be prepared). The organizer is resolved progressively by a hustling assertion of messages at the hub where the messages impact. Messages may impact either at an exchange tree hub, or at an edge. In the recent case one of the two edge's hubs is chosen as a facilitator. D2PC is time ideal (among all the examples of a particular exchange tree, and any particular Tree 2PC protocol execution; all occasions have the same tree; every occasion has an alternate hub as organizer): it submits the facilitator and every associate in least conceivable time, permitting a brief arrival of bolted assets.

D2PC is an adjustment of T2PC where the CC is progressively dictated by dashing READY (YES vote) messages, on an each exchange premise, as opposed to being settled, foreordained. For any given exchange D2PC emulates some example of T2PC. We later see that this occasion is ideal (in the arrangement of all cases for a same exchange) in the accompanying sense: It executes the confer choice (i.e., finishes stage 1 of 2PC) in least time. It confers every member in least time.

A pioneer, the organizer TM exists for every exchange to arrange 2PC for it, commonly the TM of the facilitator database. Nonetheless, the facilitator part can be exchanged to another TM for execution or dependability reasons. Instead of trading 2PC messages among themselves, the members trade the messages with their individual TMs. The important TMs convey among themselves to execute the 2PC convention construction above, "speaking to" the particular members, for ending that exchange. With this building design the convention is completely conveyed (does not require any focal handling segment or information structure), and scales up with number of system hubs (system size) successfully [17].

II. RELATED WORK

(Iskander M.K. et al, 2014) given the concept for maintaining consistency in the cloud data transaction. In distributed transactional database systems deployed over cloud servers, entities cooperate to form proofs of authorizations that are justified by collections of certified credentials. These proofs and credentials may be evaluated and collected over extended time periods under the risk of having the underlying authorization policies or the user credentials being in inconsistent states. In this paper, we highlight the criticality of the problem. Accordingly, they propose several increasingly stringent levels of policy consistency constraints, and present different enforcement approaches to guarantee the trustworthiness of transactions executing on cloud servers, by using Two-Phase Validation commit protocol which is the enhancement of the Two- Phase-commit protocol. They identified several consistency problems that can arise during cloud-hosted transaction processing using weak consistency models, particularly if policy-based authorization systems are used to enforce access controls [10].

(Das S. et al) Different paper has many different policies for maintaining transactional consistency in the cloud. The concept of ElasTraS gives the idea of elasticity and scalability of the data storage in a cloud environment and with the limitation to a single partition provides the guarantee of ACID property for transactions. The ElasTras designing is done because of the elasticity and scalability on reassigning the dynamic partition which depends on the load of the system. Moreover, ElasTraS can dynamically partition the database for applications where the designer does not want to statically partition the database, but scalability and elasticity does not support the flexible transactions. Depending on these requirements of modern applications, there are many processes of formalizing the various forms of transactions that can be efficiently executed by ElasTraS, while having minimal impact of application design and preserving the 3- tier architecture of web-servers, application servers, with ElasTraS replacing the database servers in the cloud, thus providing a high degree of elasticity and flexibility throughout the entire architecture [6].

(Vamsikrishna V. et al, 2014) have given the authorization based secure data transactions in cloud computing which identify multiple consistencies issue that can emerge during cloud host transaction process by using week consistency model. He used the algorithm for maintaining the security in different transactions. They identify multiple consistencies issue that can emerge during cloud host transaction process by using week consistency model especially that policy based authority systems are use to apply access control and elaborated a change of lightweight proof enforcement and consistency model defer, punctual, incremental and continue proofs with the view are global consistency then can apply increase the strong protection with the minimum run time overheads. In this paper we focus on the felt finding of the problem we are defining the normal understanding trusted transaction when we are dealing with the proof of authorization in cloud computing and users can obtain there computation and storage to servers and it is also called cloud. Cloud can make available for use different like application ex (Google apps, what's app) a large amount of data stored in clouds it's highly quick to detect security and privacy. It is more important problems in cloud computing. The users must authenticate itself before initialization of any transaction [21].

(Hashizume K. et al, 2013) has given have presented security issues for cloud models: IaaS, PaaS, and IaaS, which vary depending on the model. As described in this paper, storage, virtualization, and networks are the biggest security concerns in Cloud Computing. Virtualization which allows multiple users to share a physical server is one of the major concerns for cloud users.. Cloud Computing is a relatively new concept that presents a good number of benefits for its users; however, it also raises some security problems which may slow down its use. Cloud Computing is a flexible, cost-effective, and proven delivery platform for providing business or consumer IT services over the Internet. However, cloud Computing presents an added level of risk.

(Wei Z. et al, 2011) proposes CloudTPS is a scalable transaction manager which guarantees full ACID properties for multi-item transactions issued by Web applications, even in the presence of server failures and network partitions. The work done mainly depend on the two main families of scalable data layers: Bigtable and SimpleDB. This work depends on some of the simple ideas. CloudTPS gives linear scalability for web applications typically access only a few partitions in any of their transactions. Although in the presence of network partitions and server failure also CloudTPS supports for the full ACID properties. Failure recovery results in the temporary dropping in throughput and aborted transactions. Network partition recovery may possibly cause temporary unavailability of CloudTPS, as high availability is explicitly chosen to maintain strong consistency. LTM memory overflow is prevented by the memory management mechanism [22].

III. METHODOLOGY

This is a desktop application in which different cloud transaction processes are taking place one after the other. Here cloud environment is being prepared and with use of SAAS model called as Software as a service the work is implemented. There is the build-up of server and cloud environment for making the cloud transaction to be in efficient and right manner. There is the use of Wamp server which is giving full control on to the client side and application is developed in the PHP language and Mysql is being used as the backend. There is the cloud transaction processing between the server and the client. These cloud transaction are being secured by using the Secured Dynamic Two Phase Commit (D2PC) algorithm which is the extended algorithm of Dynamic Two Phase Commit algorithm. The algorithm which is used in this project will provide the security to the cloud transactions. The different architectural arrangements and the performance in the application are as follows:

3.1 CLIENT SIDE ARCHITECTURAL OVERVIEW

The centre of the arrangement is the server. The server gives a safe web interface through which organization control every last bit of its assets, permitting approved clients to empower. The server stores client records in standard document framework designs and can utilized most endeavor record frameworks. This application gives the capacity to get to, control and secure information in the endeavor; it likewise conveys the customer evaluation experience clients expect on desktops, tablets and cell telephones.

It likewise gives the action to standard webDAV customers to get to its records; empowering clients to keep on utilizing principles based efficiency instruments to interoperate consistently with it. The server stores client records in standard record framework designs and can utilize most endeavor document frameworks. Here one can mount the document framework on the server, the framework can utilize it. Further, it can likewise utilize S3 and Swift based article stores or consistent entryways – it is record framework and capacity septic.

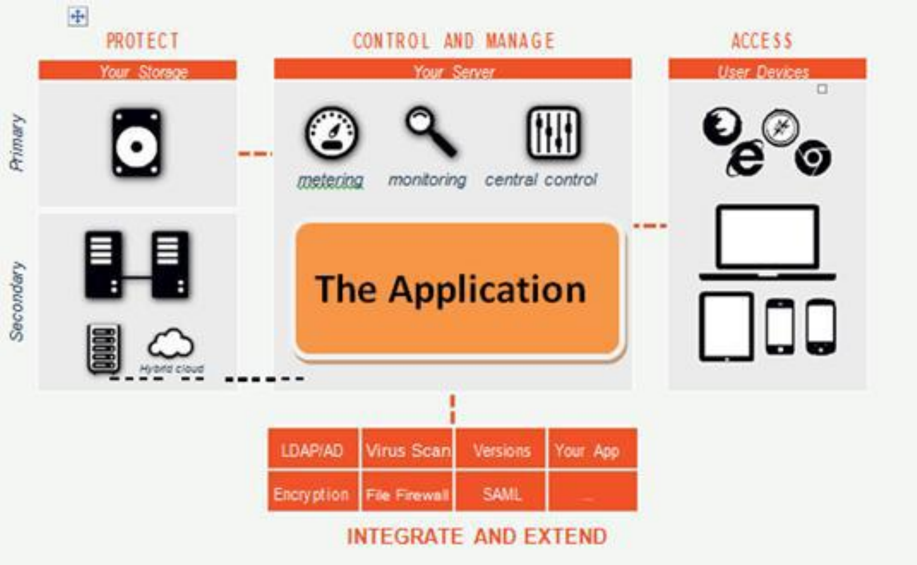

Fig 2: Client Side Architectural Overview

Phase 1: Protect Storage:

It consists of the different storage which has primary storage and secondary storage devices. From this data will transfer form Database into the server.

Phase 2: Control and Manage Server:

Here server will work in form of the interface which performs metering, monitoring and also have the central control i.e. control to the database as well as application. Here application will work as intermediate for giving the cloud transaction activities performed by the admin or user.

Phase 3: Access User Devices:

Now the values which are filled into the application can be accessed from the different accessible devices such as Computer, Laptop and Tablets.

Along these lines, it empowers to ensure the records as one needed. It lives up to expectations flawlessly with the majority of the current devices and utilities, from standard reinforcements and interruption discovery, to log chiefs and Data Loss Prevention (DLP) arrangements. It can likewise actuate the included encryption module to give an included layer of encryption very still for client record.

3.2 Server Side Architectural Overview

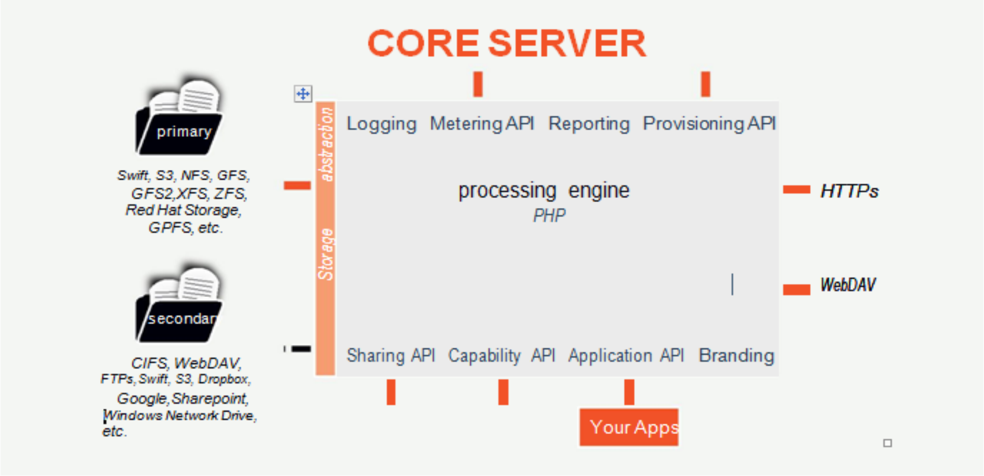

Fig 3: Server Side Architectural Overview

Phase 1: At its centre, it is a PHP web application running on top of Apache on Windows. This PHP application deals with each part of the application, from client administration to modules, record sharing and stockpiling.

Phase 2: Connected to the PHP application is a database where it stores client data, client shared record points of interest, module application states, and the document store (an execution quickening agent).

Phase 3: It gets to the database through a deliberation layer, empowering backing for MySQL. Complete web server logging is given by means of web server logs, and client and framework logs are given in a different log, or can be coordinated to a framework log document.

Phase 4: It incorporates an assortment of open APIs for coordinating with different frameworks. These include:

•**Activity** – gives a RSS channel to convey all exercises connected with a client's records, for example, sharing movement, and overhauled, renamed, erased and evacuated documents

•**Applications** – the most capable API, empowering clients to extend it out of the container, to incorporate with existing base and frameworks, and to make new module applications. Cases of this API being used incorporate the custom confirmation back finishes, music and feature spilling applications, a URL abbreviated application and a picture sneak peak application.

•**Capability** – offers data about the introduced self-abilities, so that the application and outsider applications can inquiry for the empowered elements and module applications.

•**External provisioning** – gives the capacity to include and uproot clients remotely, and empowers administrator to question metering data about its capacity utilization and portion.

•**Sharing** – gives the capacity to outside applications, for example, the versatile application, to share documents from remote gadgets.

•**Branding** – a disentangled instrument for marking servers.

Notwithstanding conveying the centre of the application, its server additionally incorporates the web interface, which gives a control focus to designing, overseeing and observing the framework. Its entry additionally gives end clients devices for controlling access to their documents and organizers. Individual utilizing this application are situated up in the framework as clients, administrators, or both. Administrators can add, enable, and disable the system settings according to one wish in framework includes through the settings menu; they can include and evacuate clients and gatherings; and they can oversee different framework settings and administrative undertakings (relocation and reinforcement, for instance).Clients get to the web interface to use and deal with their documents, and to set granular authorizations on records and organizers imparted to others on the framework.

3.3 METHODOLOGY USED

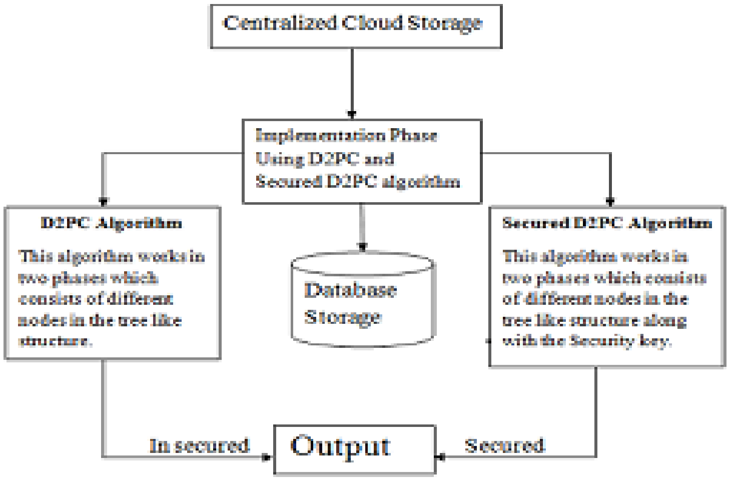
Centralized Cloud Storage
Implementation Phase Using D2PC and Secured D2PC algorithm
D2PC Algorithm
This algorithm works in two phases which consists of different nodes in the tree like structure.
Database Storage
Secured D2PC Algorithm
This algorithm works in two phases which consists of different nodes in the tree like structure along with the Security key.
In secured
Output
Secured

Fig 4: Methodology used

STEP 1: Setting of Centralized Cloud Storage: Cloud environment is prepared by using SaaS i.e. Software as a Service model which is a personal cloud storage also known as the mobile cloud storage. This is the subset of Public Cloud Storage which is used for storing ones data giving access to data from anyplace and anywhere.

STEP 2: Implementation Phase Using D2PC and Secured D2PC algorithm: With the help of Wamp server PHP is used for making the application which is a window based application. In this there is a creation of admin which performs different cloud transaction and makes new users in which sharing processes are taking place between different users. Here, two algorithms are used one is D2PC and another is Secured D2PC algorithm. In D2PC algorithm is used for performing simple cloud transaction processes which not secured. Secured D2PC algorithm is used which consists of the security key and cloud transactions are performed in more secured way using the security key. Algorithms are as follows:

Algorithm 3.1: 2PC Algorithm

Phase I:

Step 1: Each node sends PREPARE message to its neighbor, if required

Step 2: Node in the prepared state that has gotten a READY message from every one of its neighbours sends a READY message and enters the prepared state.

Step 3: Node in the prepared state that has gets a READY message from every one of its neighbors enters the prepared state and is characterized to be a CC.

PhaseII:

Step 1: A CC settles on the choice to commit and enters the committed state. At that point it sends a COMMIT message to every one of its neighbors.

Step 2: A non leaf node that has gotten a COMMIT message from a neighbor sends a COMMIT Message to each other neighbor, commits, and enters the committed state.

Step 3: A leaf node that gets a COMMIT message, commits, sends a COMMIT_ACK to its neighbor, and enters the forgotten state.

Step 4: A (nonleaf) node (in the committed state) that has gotten a COMMIT_ACK message from each to which it has sent a COMMIT, sends a COMMIT_ACK to the node from which it has gotten the COMMIT message and enters the forgotten state.

Step 5: A CC that has gotten COMMIT_ACK from every one of its neighbors enters the forgotten state.

In D2PC algorithm the CC is dynamically determined by racing READY (YES vote) messages, on a per transaction basis, rather than being fixed, predetermined. For any given transaction D2PC mimics some instance of T2PC. It is seen that this instance is optimal (in the set of all instances for a same transaction) in the following sense: It executes the commit decision in minimum time. It commits each participant in minimum time.

Algorithm 3.2: 2PC Algorithm

Phase I: Commit request phase or voting phase:

Step 1: The coordinator sends a **query to commit** message to all cohorts and waits until it has received a reply from all cohorts.

Step 2: The cohorts execute the transaction up to the point where they will be asked to commit. They each write an entry to their undo log and an entry to their redo log.

Step 3: Each cohort replies with an **agreement** message (cohort votes **Yes** to commit), if the cohort's actions succeeded, or an **abort** message (cohort votes **No**, not to commit), if the cohort experiences a failure that will make it impossible to commit

Phase II:Commit phase or Completion phase:

a) Success:

If the coordinator received an **agreement** message from *all* cohorts during the commit-request phase:

Step 1: The coordinator sends a **commit** message to all the cohorts.

Step 2: Each cohort completes the operation, and releases all the locks and resources held during the transaction.

Step 3: Each cohort sends an **acknowledgment** to the coordinator.

Step 4: The coordinator completes the transaction when all acknowledgments have been received

b) Failure:

If *any* cohort votes **No** during the commit-request phase (or the coordinator's timeout **expires**):

Step 1: The coordinator sends a **rollback** message to all the cohorts

Step 2: Each cohort undoes the transaction using the undo log, and releases the resources and locks held during the transaction.

Step 3: Each cohort sends an **acknowledgement** to the coordinator

Step 4: The coordinator undoes the transaction when all acknowledgements have been received.

Algorithm 3.1: Secured D2PC Algorithm

Phase 1:

Step 1: Each node sends PREPARE message to its neighbour, if required

Step 2: Node in the prepared state that has gotten a READY message from every one of its neighbours sends a READY message and enters the prepared state.

Step 3: Node in the prepared state that has gets a READY message from every one of its neighbours enters the prepared state and is characterized to be a CC.

Step 4: Security key is generated and sends to the CC.

Phase II:

Step 1: A CC on accepting the security key now settles on the choice to commit and enters the committed state. At that point it sends a COMMIT message to every one of its neighbours.

Step 2: A non leaf node that has gotten a COMMIT message from a neighbour sends a COMMIT message to each other neighbour Commits and enters to the Committed state.

Step 3: A leaf node that gets a COMMIT message, Commits, sends a security key and secured COMMIT_ACK to its neighbour, and enters to forgotten state.

Step 4: A non leaf node in the committed state that has gotten a COMMIT_ACK message, from each to which it has sent a COMMIT, sends a security key and secured COMMIT_ACK to the node form which it has gotten the COMMIT message and enters the forgotten state.

Step 5: A CC that has gotten secured COMMIT_ACK from every one of its neighbours enters the forgotten state.

In Secured D2PC algorithm used for making the secured cloud transaction processes which involves the security key for sharing and downloading of the documents in the cloud environment. Secured D2PC works more efficiently than D2PC providing the security to the transaction processes.

The key is generated during the transaction processes like at the time of sharing and downloading of files by the admin and users both. The key which is generated can be of any type like alphabetical, numeric and alpha numeric. This algorithm makes the transaction processes more secured and safe giving the better result.

On comparing all the three algorithms it has been found that SD2PC algorithm is the best because it takes lesser time to share or download any files.

3.4 Processing Steps in the application:

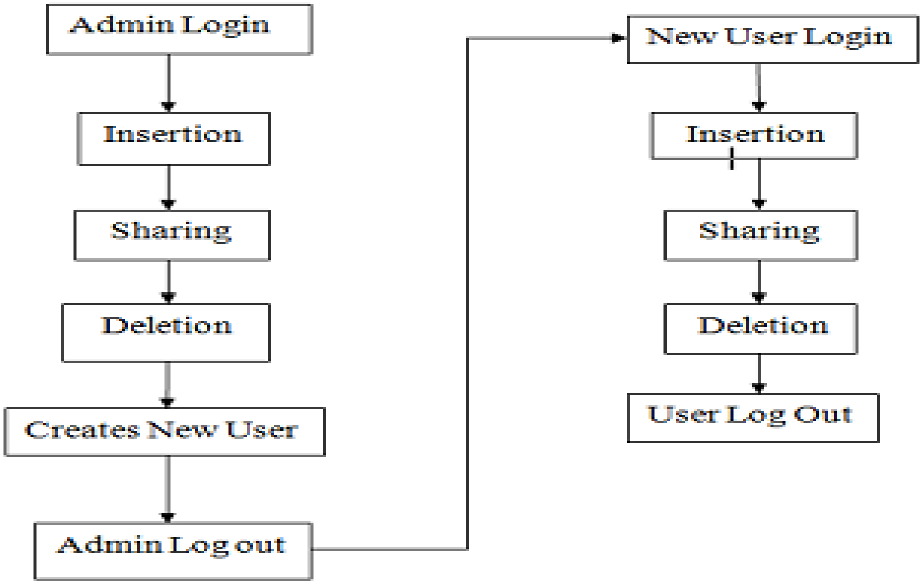

Fig 5: Flow Chart for Processing Steps

Admin Login Process: Admin by using its login id and password makes the login attempt which is the authorization process and given only to the authorized person for doing the log in attempt.

Insertion by admin: Here admin inserts or uploads different types of files such as doc files, pdf files and picture files. If insertion is success then files are inserted or uploaded to the system if some sort of failure occurs files are not inserted.

Sharing process by admin: Now admin can make sharing of files to different users. For sharing there is an option for giving authentication for further processes of the different files. It means that if admin authenticate to users for editing, changing, sharing and deleting of files then only user can perform the edit, share, change and delete processes further. If sharing process is done successfully then files are shared to different users and if any error occurs then files are not shared successfully.

Deletion by admin: If admin wants to delete some value form the application can do it from the delete option. If deletion is done correctly then its success and if any deleting process goes wrong then deletion will not be completed successfully.

Admin Creates new user: Now admin either makes new user who are declared to be the authorized person using the application for performing different transaction processing or admin itself starts the transaction processing.

Admin Log out: Now admin will make the logout attempt form its logout option and the process of steps in the application for the admin processes are done.

New User Login: Now authorized user by using their login id and password makes the login attempt into the application.

Insertion by User: On logging in user if wants to insert then it insert some values or any files from the insert or uploading option and if process is successfully done then values are inserted and if any error occurs then values are not inserted.

Sharing process by user: User can make sharing of files to different users as well as to admin. For sharing there is an option for giving authentication for further processes of the different files. It means that if user wants that another user are able to edit, change, share and delete the files it can give the permission for further processing otherwise not. If sharing process is done successfully then files are shared to different users and if any error occurs then files are not shared successfully. **Deletion by user:** If user wants to delete some value form the application can do it from the delete option. If deletion is done correctly then its success and if any deleting process goes wrong then deletion will not be completed successfully. **User Log out:** Now user will make the logout attempt form its logout option and the process of steps in the application for the user processes are done.

STEP 3: Database Storage: The transaction processes which are being done got to store in the database which is MySQL database using in this project. Here admin can manage the database using the SQL queries.

STEP 4: Output: When the transactions are performed using D2PC algorithm then the result will be the insecure output. Securities to the transactions are not provided using the simple D2PC algorithm. After the application of the Secured D2PC algorithm the transaction processes becomes secured and security keys are used for the purpose of the secured output. Security is maintained in this step using the Secured D2PC algorithm on implementing the security key.

IV. RESULT AND DISCUSSION

This project will guarantee for the trustworthiness of transactions executing on the cloud server. Different cloud transactions are carried on the server which are completed properly in less time and are secured. Security of the transaction is being provided relaxing one regarding to the safety of the files and the users data. Security has been provided to the different transactions which are running on the server. On comparing the 2PC, D2PC algorithm and Secured D2PC it has been found that the proposed algorithm is more efficient while sharing of the data as in SD2PC algorithm there is the use of security key which guarantee the secure sharing of the files and documents into the cloud environment. Security key has been generated during the sharing and downloading of the data. The proposed algorithm is found to be more efficient as here key has been used for the sharing purposes.

4.1 Comparison graph on Sharing of files between 2PC, D2PC and SD2PC algorithm On comparing the readings taken by using 2PC, D2PC algorithm with SD2PC algorithm it is found that the time taken to share the files by using SD2PC algorithm is less. This files sharing using SD2PC is more secure on comparison with sharing of files using D2PC algorithm. It has been found that the new algorithm which is used for sharing the files is more efficient on comparing with the older algorithm. Security is maintained using the key which is generated at time of sharing.

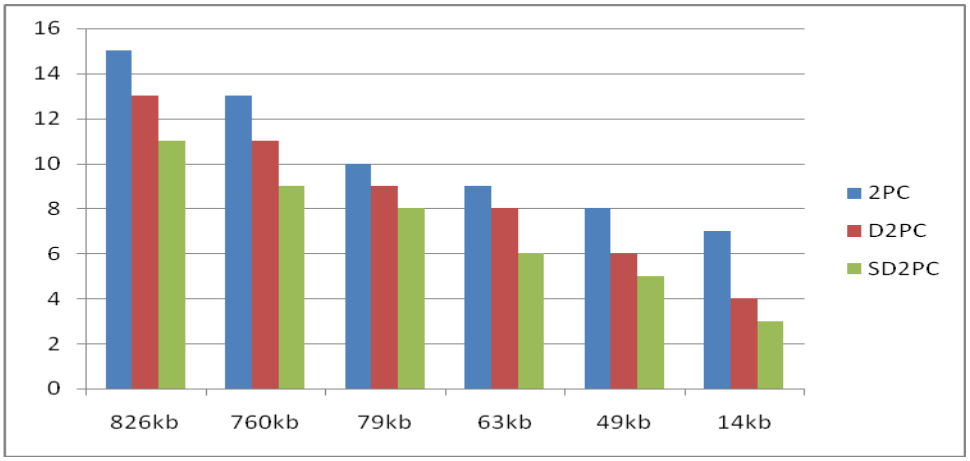

Fig 6: Comparison graph on Sharing of files between 2PC, D2PC and SD2PC algorithm

4.2 Comparison graph on Downloading of files between 2PC, D2PC and SD2PC algorithm On comparing both the readings taken by using D2PC algorithm with SD2PC algorithm it is found that the time taken to download the files by using SD2PC algorithm is less. It has been found that the new algorithm SD2PC which is used for downloading the files is more efficient on comparing with the older algorithm. Security is maintained using the key which is generated at time of downloading.

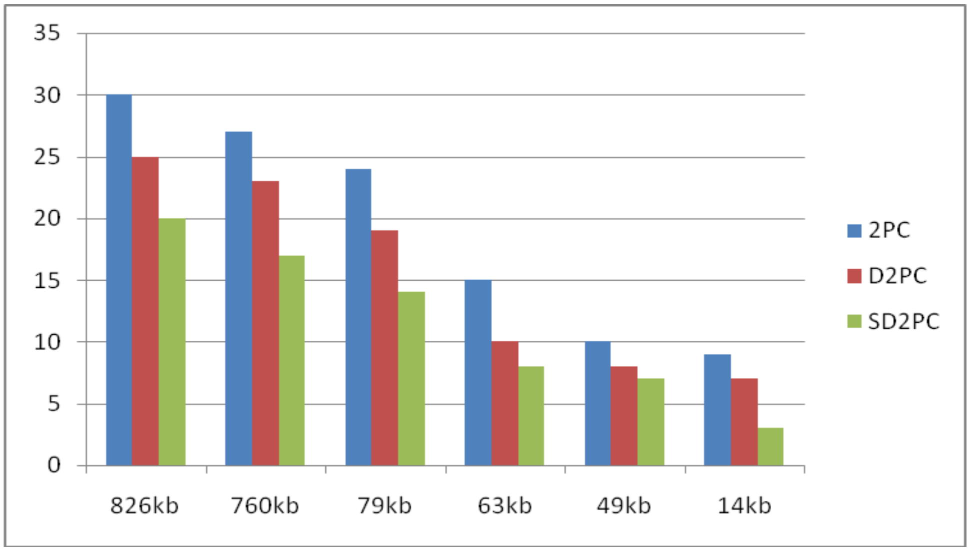

Fig 7: Comparison graph on Downloading of files between 2PC, D2PC and SD2PC algorithm

V.CONCLUSION

Although cloud computing and its services are getting popular day by day but there is always the misconception in the users mind regarding the security of their stored data and different work which users are performing in the cloud environment. Cloud is the vast area where data are stores and the security is also to be maintained for these data. In this project different work has been done in order to protect the security of the data and transactional security is also to be maintained. Following are the different work which is done in this project to maintain the transaction security. They are:

- Using Secured Dynamic Two Phase Commit (SD2PC) algorithm to remove the inconsistency problem for the cloud transaction activity.
- Implemented Secured D2PC algorithm for maintaining the transactional security in the cloud environment by using the security key during the time of different transactional processes.
- Different working processes of the application have been shown by the different screen shots which are showing the working of the different transaction processes.
- Plotted the graph in comparison with the file size and time taken in order to perform the different cloud transactions.
- On using secured D2PC algorithm the transaction processes becomes more fast and secured and misconception regarding the security of data has been removed to a greater extent.

REFERENCES

[1] Abadi D., [2009], ―Data Management in the Cloud: Limitations an Opportunities,‖ IEEE, pp. 3-12.

[2] Barrie S., [2011], ―Cloud Computing Bible‖ , ―Wiley Publishing‖ ,pp 24-34.

[3] Chang F. et al, [2006], ―Bigtable: A Distributed Storage System for Structured Data,‖ Proc. Seventh USENIX Symp

[4] Cooper B.F. et al[2008], ―PNUTS: Yahoo!'s Hosted Data Serving Platform,‖ Proc. VLDB Endowmen, pp. 1277-1288.

[5] DeCandia G. et al [2007], ―Dynamo: Amazons Highly Available Key- Value Store,‖ Proc. 21st ACM SIGOPS Symp. Operating Systems Principles (SOSP).

[6] Das.S et al, [2009] ,―Elastras: An Elastic Transactional Data Store in the Cloud,‖ Proc. Conf. Hot Topics in Cloud Computing (USENIX HotCloud).

[7] Hashizume K. et al, [2013] ―An analysis of security issues for cloud computing‖ , Springer.

[8] Harauz J. et al, [2009], ―Data Security in the world of Cloud Computing ‖ , IEEE,Co Published by the IEEE Computer and Reliability Societies, pp. 61-64

[9] Iskander M. et al,[2014], ―Balancing Performance, Accuracy, and Precision for Secure Cloud Transactions‖ ,IEEE

[10] Kraska T. et al [2009], ―Consistency Rationing in the Cloud:Pay only when it matters‖ ,ACM.

[11] Kale S. et al [2014], ―Supporting Security and Consistency for Distributed, Concurrent Access to Cloud Databases: A Review‖ , International Journal of Current Engineering and Technology.

[12] Lakshman A. et al, [2010], ―Cassandra- A Decentralized Structured Storage System,‖ ACM, pp. 35-40.

[13] Lee A. et al, [2006], ―Safety and Consistency in Policy Based Authorization Systems‖ ,ACM. Lombardi F. et al [2012], "Secure virtualization for cloud computing.

[14] Landis C. et al, ―Cloud Computing Made Easy‖ , Version 0.3, Virtual global, pp. 30-36.

[15] Marrinescu D., [2013] ―Cloud Computing: Theory and Practice‖ , ―Elsevier‖ , Chapter 2, pp 21-28. [16] Raz Y., [2012], ―The Dynamic Two Phase Commitment (D2PC) Protocol‖ , EMC Corporation,pp 162-169.

[17] Ranjith D. et al, [2013], ―Identify Security Using Authentication and Authorization in Cloud Computing‖ , International Journal of Computer and Organizational Trends.

[18] Rakesh A. et al, [2014], ―Secured Consistent Transactions‖ , International Journal of Research in Engineering and Advanced Technology, Volume 2, Issue 2, pp.1-7.

[19] Shraer A. et al, [2010], ―Venus: Verification for Untrusted Cloud Storage‖ , ACM, pp.89 93.

[20] Vamsikrishna V. et al,[2014] ―Authorization Based Secure Data Transaction in CloudComputing‖ ,International Journal of Engineering Trends and Technology (IJETT) .

[21] Vouk M.A., [2008], ―Cloud Computing- Issues, Research and Implementations‖ , Journal of Computing and Information Technology, pp. 235-246.

[22] Wei Z. et al, [2009], ―CloudTPS: Scalable transaction for Web Applications in the Cloud‖ IEEE.

[23] Weihai Y. et al [2004], ―A Dynamic Two-Phase Commit Protocol for Self-Adapting Services‖ , Proc. of the IEEE International Conference on Services Computing.