**B.E. PROJECT**

**ON**

# Enhancing Security in Cloud Computing

# using DROPS

## Submitted by:

### Himanshu Doley(730/IT/15)

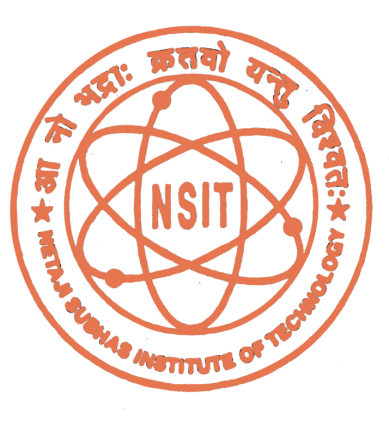
### Karan (734/IT/15)

**Navdeep Redhu (749/IT/15)**

##### In partial fulfillment of B.E. (Information Technology) degree of University of Delhi

**Under the Guidance of**

Ms. Amarjit Malhotra



**DIVISION OF INFORMATION TECHNOLOGY**

### NETAJI SUBHAS INSTITUTE OF TECHNOLOGY

**UNIVERSITY OF DELHI, DELHI**

# ACKNOWLEDGEMENTS

We would like to express our deepest gratitude to our advisor **Ms. Amarjit Malhotra** for giving us the opportunity to work under their able supervision. We are indebted towards the faculty of NSIT, whose guidance and teaching for the past four years have helped us in understanding the theory and practices of Instrumentation and Control Engineering. We also thank our parents and families who always supported us unconditionally through thick and thin. Their belief in our capabilities always pushed us to give our best in this project. Without these invaluable contributions, this project could not have been completed

.

Himanshu Doley Navdeep Redhu Karan

Roll No.730/IT/15 Roll No.749/IT/15 Roll No.734/IT/15

Information Technology Division

Netaji Subhas Institute of Technology (NSIT)

Azad Hind Fauj Marg

Sector-3, Dwarka

PIN – 110078

# DECLARATION

##### This is to certify that the project entitled, “**Enhancing security in cloud computing using DROPS**” by **Himanshu Doley**, **Navdeep Redhu** and **Karan Meena** are record of bonafide work carried out by us, in the Division of Information Technology, Netaji Subhas Institute of Technology, University of Delhi, New Delhi, in partial fulfillment of requirements for the award of the degree of Bachelor of Engineering in Information Technology, University of Delhi in the academic year **2018-2019**.

The results presented in this thesis are original and have not been submitted to any other university in any form for the award of any other degree.

Himanshu Doley Navdeep Redhu Karan

Roll No.730/IT/15 Roll No.749/IT/15 Roll No.734/IT/15

Information Technology Division

Netaji Subhas Institute of Technology (NSIT)

Azad Hind Fauj Marg

Sector-3, Dwarka, New Delhi

PIN – 110078

****

**CERTIFICATE**

This is to certify that the project entitled “**Enhancing security in cloud computing using DROPS**” submitted by **Himanshu Doley(730/IT/15), Navdeep Redhu(749/IT/15) and Karan 734/IT/15)** for the partial fulfillment of Award of Bachelor of Engineering is a bonafide record of the work done by the candidates under my guidance. To the best of my knowledge, this work has not been submitted for the award of any other degree.

Date:

Place:

Signature

**Ms. Amarjit Malhotra**

Assistant Professor, IT

NSIT, New Delhi

**ABSTRACT**

Outsourcing data to a third-party administrative control, as is done in cloud computing, gives rise to security concerns. The data compromise may occur due to attacks by other users and nodes within the cloud. Therefore, high security measures are required to protect data within the cloud. However, the employed security strategy must also take into account the optimization of the data retrieval time. In this project, we propose division and replication of data in the cloud for optimal performance and security (DROPS) that collectively approaches the security and performance issues. In the DROPS methodology, we divide a file into fragments, and replicate the fragmented data over the cloud nodes. Each of the nodes stores only a single fragment of a particular data file that ensures that even in case of a successful attack, no meaningful information is revealed to the attacker.

**TABLE OF CONTENTS**

ACKNOWLEDGEMENTS

DEDICATION

DECLARATION

CERTIFICATE

PLAGIARISM REPORT

ABSTRACT

TABLEOFCONTENTS

i

ii

iii

iv

v

vi

vii

1. INTRODUCTION………………………………………………………………..1 1.1 Cloud computing………….……………………………………………………1 1.1.1 Benefits of cloud computing….……………………………………………..1

1.1.2 Types of cloud computing ……...………………………………………..2

1.1.3 Uses of cloud computing…………………………...…………………….3

1.1.4 Types of cloud services……………………………..……………….........3

1.2 Security concerns in cloud computing..………………………………………...3

1.3 Methods to enhance security..………...………………………………………...8

1.3.1 Data encryption…....…………………………………………….………….....8

1.3.2 Data fragmentation…………………………………………………………....9

1.4 Need for DROPS…………………..……………………………………………..15

1. Project Requirements…………………………………………………………………17

2.1 System Requirements………………………………………………………….…17

2.1.1 Operating System……………………………………………...…………..….17

2.1.2 Graphics Processing Unit…………………………..……………………….17

2.2 Software Requirements……………………………..……..………………………17

2.2.1 Matlab……………………………………………………………………….17

2.2.2 Variables…………………………………….………………………………….18

2.2.3 Matrices and Vectors……………………………...………………………………..18

1. PROJECT APPROACH…………………………………………………………………….21

3.1 Topic Research and Selection…………………………….……………………………21

3.2 Research…………………………………………………….………………………….21

1. DROPS……………………………………………………………..………………………..22

4.1 Introduction …………………………………………………………………………….22

4.2 Concept..…………………………………………………………..……………………23

4.2.1 Data fragmentation……………………………………………..…………………….23

4.2.2 Centrality……………….…………………………………………………………….24

4.2.3 T-Coloring…………………………………………………………………………….25

4.3 System Model……………….……………………………………...……………………25

4.3.1 Notations……….………………………………………………..……………………25

4.4 DROPS…………………………….………………………………..……………………28

4.4.1 Algorithm……….………………………………………………..…………………....25

1. DYNAMIC FILTRATION OF INPUT COMPLEX………………………………………….29

5.1 Working of filter…………………………………………………………………………29

5.2 Filtration of ECG signal ………………………………………………………………...29

5.2.1 MATLAB CODE……………………………………………………………………..30

1. RESULTS……………………………………………………………………………………..34

6.1 Amplitude and phase characteristics of EMG signal…………………………………….34

6.1.1 Amplitude and phase characteristics at 1st stage………………………………………35

6.1.2 Amplitude and phase characteristics at 2nd stage……………………………………..36

6.1.3 MATLAB CODE……………………………………………………………….…….37

6.2 Fixed point precision and double precision……………………………………….……..40

6.2.1 Fixed point precision………………………………………………………………….40

6.2.2 Double precision ……………………………………………………………….……..41

6.3 Output EMG signal ………………………………………………………………..……..42

6.4 Power spectrum …………………………………………………………………………..43

6.4.1 Power spectrum of EMG waveform……………………………………………..…….44

6.4.2 MATLAB CODE……………………………………………………………………...44

6.5 Signal to Noise ratio………………………………………………………………………56

6.5.1 SNR of EMG to ECG………………………………………………………………….57

1. Conclusion………………………………………………………………………………………58
2. References……………………………………………………………………………………….59
3. Bio data………………………………………………………………………………………….60

**1.** **INTRODUCTION**

**1.1** **Cloud computing**

Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. The term is generally used to describe data centers available to many users over the Internet. Large clouds, predominant today, often have functions distributed over multiple locations from central servers. If the connection to the user is relatively close, it may be designated an edge server.

The availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture, and autonomic and utility computing has led to growth in cloud computing.

***1.1.1*** ***Benefits of cloud computing***

i)**Cost**-Cloud computing eliminates the capital expense of buying hardware and software and setting up and running on-site data centres—the racks of servers, the round-the-clock electricity for power and cooling, the IT experts for managing the infrastructure. It adds up fast.

ii)**Speed** - Most cloud computing services are provided self service and on demand, so even vast amounts of computing resources can be provisioned in minutes, typically with just a few mouse clicks, giving businesses a lot of flexibility and taking the pressure off capacity planning.

iii)**Global scale -** The benefits of cloud computing services include the ability to scale elastically. In cloud speak, that means delivering the right amount of IT resources—for example, more or less computing power, storage, bandwidth—right when it is needed and from the right geographic location.

iv)**Productivity -** On-site data centres typically require a lot of “racking and stacking”—hardware setup, software patching, and other time-consuming IT management chores. Cloud computing removes the need for many of these tasks, so IT teams can spend time on achieving more important business goals.

v)**Performance -** The biggest cloud computing services run on a worldwide network of secure data centres, which are regularly upgraded to the latest generation of fast and efficient computing hardware. This offers several benefits over a single corporate data centre, including reduced network latency for applications and greater economies of scale.

vi)**Reliability-**Cloud computing makes data backup, disaster recovery and business continuity easier and less expensive because data can be mirrored at multiple redundant sites on the cloud provider’s network.

***1.1.2*** ***Types of cloud computing***

Not all clouds are the same and not one type of cloud computing is right for everyone. Several different models, types and services have evolved to help offer the right solution for your needs.

First, you need to determine the type of cloud deployment or cloud computing architecture, that your cloud services will be implemented on. There are three different ways to deploy cloud services: on a public cloud, private cloud or hybrid cloud.

i)**Public cloud-**Public clouds are owned and operated by a third-party cloud service providers, which deliver their computing resources like servers and storage over the Internet. Microsoft Azure is an example of a public cloud. With a public cloud, all hardware, software and other supporting infrastructure is owned and managed by the cloud provider. You access these services and manage your account using a web browser.

i)**Private cloud-**A private cloud refers to cloud computing resources used exclusively by a single business or organisation. A private cloud can be physically located on the company’s on-site data center. Some companies also pay third-party service providers to host their private cloud. A private cloud is one in which the services and infrastructure are maintained on a private network.

i)**Hybrid cloud-**Hybrid clouds combine public and private clouds, bound together by technology that allows data and applications to be shared between them. By allowing data and applications to move between private and public clouds, a hybrid cloud gives your business greater flexibility, more deployment options and helps optimise your existing infrastructure, security and compliance.

***1.1.3*** ***Uses of cloud computing***

i)**Create new apps and services -** Quickly build, deploy and scale applications—web, mobile and API—on any platform. Access the resources you need to help meet performance, security and compliance requirements.

ii)**Test and build applications -** Reduce application development cost and time by using cloud infrastructures that can easily be scaled up or down.

iii)**Analyse data -** Unify your data across teams, divisions and locations in the cloud. Then use cloud services, such as machine learning and artificial intelligence, to uncover insights for more informed decisions.

iv)**Store, back up and recover data-**Protect your data more cost-efficiently—and at massive scale—by transferring your data over the Internet to an offsite cloud storage system that is accessible from any location and any device.

v)**Deliver software on demand-**Also known as software as a service (SaaS), on-demand software lets you offer the latest software versions and updates around to customers—anytime they need, anywhere they are.

***1.1.4*** ***Types of cloud services***

Most cloud computing services fall into four broad categories: infrastructure as a service (IaaS), platform as a service (PaaS), server less and software as a service (SaaS). These are sometimes called the cloud computing stack because they build on top of one another. Knowing what they are and how they are different makes it easier to accomplish your business goals.

i)**Infrastructure as a service(Iaas)-**The most basic category of cloud computing services. With IaaS, you rent IT infrastructure—servers and virtual machines (VMs), storage, networks, operating systems—from a cloud provider on a pay-as-you-go basis.

ii)**Platform as a service(Paas)-**Platform as a service refers to cloud computing services that supply an on-demand environment for developing, testing, delivering and managing software applications. PaaS is designed to make it easier for developers to quickly create web or mobile apps, without worrying about setting up or managing the underlying infrastructure of servers, storage, network and databases needed for development.

iii) **Software as a service(Saas)-**Software as a service is a method for delivering software applications over the Internet, on demand and typically on a subscription basis. With SaaS, cloud providers host and manage the software application and underlying infrastructure and handle any maintenance, like software upgrades and security patching. Users connect to the application over the Internet, usually with a web browser on their phone, tablet or PC.

**1.2** **Security concerns in cloud computing**

Cloud computing poses privacy concerns because the service provider can access the data that is in the cloud at any time. It could accidentally or deliberately alter or delete information. According to the Cloud Security Alliance, the top three threats in the cloud are Insecure Interfaces and API's, Data Loss & Leakage, and Hardware Failure—which accounted for 29%, 25% and 10% of all cloud security outages respectively. Together, these form shared technology vulnerabilities.

i)**Data Breaches-**A study conducted by the Ponemon Institute entitled “Man In Cloud Attack” reports that over 50 percent of the IT and security professionals surveyed believed their organization’s security measures to protect data on cloud services are low. This study used nine scenarios, where a data breach had occurred, to determine if that belief was founded in fact.

After evaluating each scenario, the report concluded that overall data breaching was three times more likely to occur for businesses that utilize the cloud than those that don’t. The simple conclusion is that the cloud comes with a unique set of characteristics that make it more vulnerable.

ii)**Hijacking of Accounts -**Attackers now have the ability to use your (or your employees’) login information to remotely access sensitive data stored on the cloud; additionally, attackers can falsify and manipulate information through hijacked credentials.

Other methods of hijacking include scripting bugs and reused passwords, which allow attackers to easily and often without detection steal credentials.

iii)**Malware Injection-**Malware injections are scripts or code embedded into cloud services that act as “valid instances” and run as SaaS to cloud servers. This means that malicious code can be injected into cloud services and viewed as part of the software or service that is running within the cloud servers themselves.

iv)**Data loss-**Data on cloud services can be lost through a malicious attack, natural disaster, or a data wipe by the service provider. Losing vital information can be devastating to businesses that don’t have a recovery plan. Amazon is an example of an organization that suffered data loss by permanently destroying many of its own customers’ data in 2011.

**1.3** **Methods to enhance security**

***1.3.1*** ***Data encryption***

iii)**Identity-based Encryption-**This technique mainly helps in certificate management and management of public key for the public key infrastructure. Outsourcing computation here is put to the identity-based encryption while the same scheme for the server has also been proposed.

iii)**Attribute-based Encryption-**This type of encryption, in general, has numerous forms. The most common one among them is cipher text attribute based and key attribute based encryption. Some researchers have also suggested a new attribute based encryption method with the hierarchical name attribute based encryption.

***1.3.2*** ***Data fragmentation***

The security of a large-scale system, such as cloud depends on the security of the system as a whole and the security of individual nodes. A successful intrusion into a single node may have severe consequences, not only for data and applications on the victim node, but also for the other nodes. The data on the victim node may be revealed fully because of the presence of the whole file.

The amount of compromised data can be reduced by making fragments of a data file and storing them on separate nodes

**1.4 Need for DROPS**

In a cloud environment, a file in its totality, stored at a node leads to a single point of failure. A successful attack on a node might put the data confidentiality or integrity, or both at risk.

Security and replication are essential for a large-scale system, such as cloud, as both are utilized to provide services to the end user.

In the DROPS methodology, we propose not to store the entire file at a single node. The DROPS methodology fragments the file and makes use of the cloud for replication.

The fragments are distributed such that no node in a cloud holds more than a single fragment, so that even a successful attack on the node leaks no significant information.

In the DROPS methodology, user sends the data file to cloud. The cloud manager system (a user facing server in the cloud that entertains user’s requests) upon receiving the file performs: (a) fragmentation, (b) first cycle of nodes selection and stores one fragment over each of the selected node, and (c) second cycle of nodes selection for fragments replication. The cloud manager keeps record of the fragment placement and is assumed to be a secure entity.

**2. PROJECT REQUIREMENT**

**2.1 SOFTWARE REQUIREMENTS SPECIFICATION**

Software Requirements Specification (SRS) is the starting point of the software development activity. Little importance was given to this phases in the early days of software development. The emphasis was first on coding and then shifted to design.

As systems grew more complex, it become evident that the goal of the entire system cannot be easily comprehended. Hence need for the requirements analysis phase arose. Now, for large software systems, requirements analysis is perhaps the most difficult activity and also the most error prone.

Some of the difficulty is due to the scope of this phase. The software project is imitated by the client needs. In the beginning these needs are in the minds of various people in the client organization. The requirement analyst has to identify the requirements by tacking to these people and understanding there needs. In situations where the software is to automated a currently manuals process, most of the needs can be understood by observing the current practice.

The SRS is a means of translating the ideas in the minds of the clients (the output) into formal document (the output of the requirements phase). Thus the output of the phase is a set of formally specified requirements, which hopefully are complete and consistent, while the input has none of these properties.

**2.2 Functional Requirement**

**Select Input Audio File**

**Select Output Audio File**

**Embedded Data**

**Select Key**

**Embedded Audio File**

**Select Encrypt Audio File**

**Extract Audio File**

**View Output**

**Select Key**

**Enter New Text**

**2.3 Performance Requirements**

The project must the end user requirements. Accuracy and fast must be imposed in the project.

The project is development as easy as possible for the sake of end user. The project has to be developed with view of satisfying the future requirements and future enhancement.

The tool has been finally implemented satisfying the needs specified by the company. As per the performance is concerned this system said is performing

This processing as well as tine taken to generate well reports where also even when large amount of data was used. The system is designed in such a way that even when large amount of data used for processing there would less performance degradation.

**2.4 Interface Requirements**

**2.4.1 Hardware Interface**

The stranded input device like keyboard and mouse are to get input. The output will be generated and display in the monitor. The reports can also be exported to a SQL-server document are text file. The stranded printer in used to take outputs.

**2.4.2 Software Interface**

The design part and interface id done the front end ASP.Net and SQL server as a backend of the project.

**2.5 Operational Requirements**

The database or databases that are being failed over to the stand by server cannot be used for anything else. But databases on the standby server not being used for failover can still be used normally.

When it comes time for actual fail over,you much one of two things to make your application work either rename the standby server the same name as the failed production server(and the IP address),or re-point your user’s applications to new standby server.in some cases,neither of this option is practical.

**2.6 Resource Requirements**

**2.6.1 Software Specification:-**

OPERATING SYSTEM : WINDOW XP & ABOVE

FRONT END: Microsoft visual studio .NET 2010, HTML, CSS

CODING LANGUAGE: C#, .NET

**2.6.2 Hardware Specification:-**

SYSTEM : intel pentium or above

HARD DISK : 40 GB

FLOPPY DRIVE : 1.44 MB

MONITOR : 15 VGA colour monitor

MOUSE : Logitech.

RAM : 128 MB or above

KEYBOARD : 110 keys enhanced.

**2.7 Security Requirements**

Web application are available via network access, it is a difficult. If not possible, to limit the population of the end-user who may access the applications? In order to product sensitive connect and provide secure mode be implemented throughout the infrastructure that the supports web application and within the application itself.

Web Application have become heavy integrated with critical corporate and database.

E-commerce application extracts and then store sensitive customer information.

**2.8 Design Requirements**

To create project, add base masters and masters to the project, assign behaviors to the master, create and assign behavior sets, and then apply, test and validate those behaviors. It also shows how to create and build a stencil to hold the shapes.

**2.9 Quality and Reliability Requirements**

A software component that is developed for reuse would be correct and would contain no defects. In reality, formal verification is not carried out routinely, and defects can add to occur.However,with each reuse,,defects are found eliminated, and a components qualify improve as a result. Over time the components virtually defect free.

Software reliability is defined in statistcal term as “the probability of faultier-free operation of a computer program in a specified environment for specified tine”. The software quality and reliability, failure is nonconformance to software requirements. Failure can be only anything or catastrophic. one failure can be corrected within seconds while another requirements week even mouths to correct. Complicating the issue even further, the correction of the one failure may in fact result in the introduction of the errors that ultimately result in other failure.

**3.PROJECT APPROACH**

**3.1 Topic Research and Selection**

We read a number of research papers, University websites and held discussions with our faculty in-charge to identify the topic where there was scope for research and implementation; Decided to work on DROPS (Division and replication of data in cloud for optimal performance and security), because of our interest in the field and scope for solving a world-issue.

**3.2 Research**

We read research papers, kernels in the internet, eliminated irrelevant ones and selected a few related papers to do an exhaustive study. This is something that hasn’t been done before. We studied about C# (back end), .NET(framework) and SQL (database server), and understood the working of the reward systems.

We introduce Division and Replication of Data in the Cloud for Optimal Performance and Security system (DROPS). It shares client record into fragments and repeats them at dynamic Distributed calculation. In any case, the surety measures measure concern get expanded by the advantages of minimal effort, insignificant administration (from a client’s point of view), and more projecting adaptability accompany. Security is a standout amongst the most pivotal perspective among those forbidding the fare mentum reception of distributed computer science. In cloud computing, the information is stored on third party space which results in a security concerns. The user and lymph gland within cloud may compromise the data. Therefore, to protect data within the cloud senior high school security measures are required. Divide a data file into fragments, and replicate the fragmented data over the cloud lymph gland is done in DROPS methodological analysis. Only a fragment of a particular data file can be stored by each of the node that ensures that no meaningful information is revealed to the aggressor even in case of a successful attack.

**4.DROPS**

**4.1 Introduction**

We present Division and Replication of Data in the cloud for Optimal Performance and Security (DROPS) that judicially fragments user files into pieces and replicates them at strategic locations within the cloud. The division of a file into fragments is performed based on a given user criteria such that the individual fragments do not contain any meaningful information. Each of the cloud nodes (we use the term node to represent computing, storage, physical, and virtual machines) contains a distinct fragment to increase the data security. . A successful attack on a single node must not reveal the locations of other fragments within the cloud. To keep an attacker uncertain about the locations of the file fragments and to further improve the security, we select the nodes in a manner that they are not adjacent and are at certain distance from each other. The node separation is ensured by the means of the T-coloring. To improve data retrieval time, the nodes are selected based on the centrality measures that ensure an improved access time. To further improve the retrieval time, we judicially replicate fragments over the nodes that generate the highest read/ write requests. The selection of the nodes is performed in two phases. In the first phase, the nodes are selected for the initial placement of the fragments based on the centrality measures. In the second phase, the nodes are selected for replication. The working of the DROPS methodology is shown as a high-level work flow .

The proposed DROPS scheme ensures that even in the case of a successful attack, no meaningful information is revealed to the attacker.

We do not rely on traditional cryptographic techniques for data security. The non- cryptographic nature of the proposed scheme makes it faster to perform the required operations (placement and retrieval) on the data.

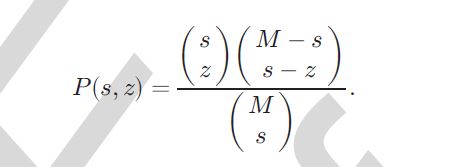
We ensure a controlled replication of the file fragments, where each of the fragments is replicated only once for the purpose of improved security.

A secure and optimal placement of data objects in a distributed system is presented in .An encryption key is divided into n shares and distributed on different sites within the network. The division of a key into n shares is carried out through the (k, n) threshold secret sharing scheme. The network is divided into clusters. The number of replicas and their placement is determined through heuristics. A primary site is selected in each of the clusters that allocates the replicas within the cluster. The scheme presented in combines the replication problem with security and access time improvement. Nevertheless, the scheme focuses only on the security of the encryption key. The data files are not fragmented and are handled as a single file. The DROPS methodology, on the other hand, fragments the file and store the fragments on multiple nodes. Moreover, the DROPS methodology focuses on the security of the data within the cloud computing domain that is not considered in.

**4.2 CONCEPT**

**4.2.1 Data Fragmentation**

The security of a large-scale system, such as cloud depends on the security of the system as a whole and the security of individual nodes. A successful intrusion into a single node may have severe consequences, not only for data and application on the victim node, but also for the other nodes. The data on the victim node may be revealed fully because of the presence of the whole file . A successful intrusion may be a result of some software or administrative vulnerability .In case of homogenous systems, the same flaw can be utilized to target other nodes within the system. The success of an attack on the subsequent nodes will require less effort as compared to the effort on the first node. Comparatively, more effort is required for heterogeneous systems. However, compromising a single file will require the effort to penetrate only a single node. The amount of compromised data can be reduced by making fragments of a data file and storing them on separate nodes. A successful intrusion on a single or few nodes will only pro- vide access to a portion of data that might not be of any significance. Moreover, if an attacker is uncertain about the locations of the fragments, the probability of finding fragments on all of the nodes is very low. Let us consider a cloud with M nodes and a file with z number of fragments. Let s be the number of successful intrusions on distinct nodes, such that s > z. The probability that s number of victim nodes contain all of the z sites storing the file fragments represented by P(s, z).



**4.2.2.Centrality**

The centrality of a node in a graph provides the measure of the relative importance of a node in the network. The objective of improved retrieval time in replication makes the centrality measures more important. There are various centrality measures; for instance, closeness centrality, degree centrality, betweenness centrality, eccentricity centrality, and eigenvector centrality. We only elaborate on the closeness, between- ness, and eccentricity centralities because we are using the aforesaid three centralities in this work. Centrality indices are answers to the question "What characterizes an important vertex?" The answer is given in terms of a real-valued function on the vertices of a graph, where the values produced are expected to provide a ranking which identifies the most important nodes.

The word "importance" has a wide number of meanings, leading to many different definitions of centrality. Two categorization schemes have been proposed. "Importance" can be conceived in relation to a type of flow or transfer across the network. This allows centralities to be classified by the type of flow they consider important. "Importance" can alternatively be conceived as involvement in the cohesiveness of the network. This allows centralities to be classified based on how they measure cohesiveness. Both of these approaches divide centralities in distinct categories. A further conclusion is that a centrality which is appropriate for one category will often "get it wrong" when applied to a different category.

When centralities are categorized by their approach to cohesiveness, it becomes apparent that the majority of centralities inhabit one category. The count of the number of walks starting from a given vertex differs only in how walks are defined and counted. Restricting consideration to this group allows for a soft characterization which places centralities on a spectrum from walks of length one (degree centrality) to infinite walks (eigenvalue centrality). The observation that many centralities share this familial relationships perhaps explains the high rank correlations between these indices.

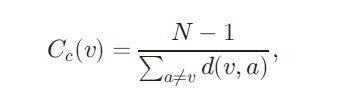
**4.2.2 (a)** **Betweenness Centrality**

The betweenness centrality of a node n is the number of the shortest paths, between other nodes, passing through n.Formally, the betweenness centrality of any node v in a net- work is given.

In a weighted network the links connecting the nodes are no longer treated as binary interactions, but are weighted in proportion to their capacity, influence, frequency, etc., which adds another dimension of heterogeneity within the network beyond the topological effects. A node's strength in a weighted network is given by the sum of the weights of its adjacent edges.

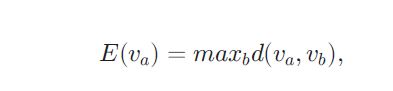
**4.2.1(b) Closeness Centrality**

A node is said to be closer with respect to all of the other nodes within a network, if the sum of the distances from all of the other nodes is lower than the sum of the distances of other candidate nodes from all of the other nodes [24]. The lower the sum of distances from the other nodes, the more central is the node. Formally, the closeness centrality of a node v in a network is defined as:



**4.2.1(c) Eccentricity**

The eccentricity of a node n is the maximum distance to any node from a node n [24]. A node is more central in the net- work, if it is less eccentric. Formally, the eccentricity can be given as:



It may be noted that in our evaluation of the strategies the centrality measures introduced above seem very meaningful and relevant than using simple hop-count kind of metrics.

**4.2.3 T-Coloring**

In graph theory, a T-Coloring of a graph {\displaystyle G=(V,E)}, given the set *T* of nonnegative integers containing 0, is a function {\displaystyle c:V(G)\rightarrow \mathbb {N} } that maps each vertex of G to a positive integer (color) such that {\displaystyle (u,w)\in E(G)\Rightarrow \left|c(u)-c(w)\right|\notin T}. In simple words, the absolute value of the difference between two colors of adjacent vertices must not belong to fixed set *T*. The concept was introduced by William K. Hale. If *T = {0}* it reduces to common vertex coloring. {\displaystyle s\_{i}=\sum \_{j=1}^{N}a\_{ij}w\_{ij}}



The mapping function f assigns a color to a vertex. In simple words, the distance between the colors of the adjacent vertices must not belong to T. Formulated by Hale, the T-coloring problem for channel assignment assigns channels to the nodes, such that the channels are separated by a distance to avoid interference.

**4.3 SYSTEM MODEL**

Consider a cloud that consists ofM nodes, each with its own storage capacity. Let Si represents the name of Ith node and si denotes total storage capacity of Si. The communication time between Si and Sj is the total time of all of the links within a selected path from Si to Sj represented by c(i, j). We consider N number of file fragments such that Ok denotes kth fragment of a file while ok represents the size of kth fragment. Let the total read and write requests from Si for Ok be represented by ri k and wi k, respectively. Let Pk denote the primary node that stores the primary copy of Ok. The replication scheme for Ok denoted by Rk is also stored at Pk. Moreover, every Si contains a two-field record,

**Notations and Their Meanings**

**Symbols**  **Meanings**

M Total number of nodes in the cloud

N Total number of file fragments to be placed

Ok kth fragment of file

ok Size of Ok

Si ith node

si Size of Si

ceni Centrality measure for Si

colSi Color assigned to Si

T A set containing distances by which assignment of fragments must be separated

rik Number of reads for Ok from Si

Rik Aggregate read cost of ri

wik Number of writes for Ok from Si

Wik Aggregate write cost of wik

NNik Nearest neighbor of Si holding Ok

c(i,j) Communication cost between Si and Sj

Pk Primary node for Ok

Rk Replication schema of Ok

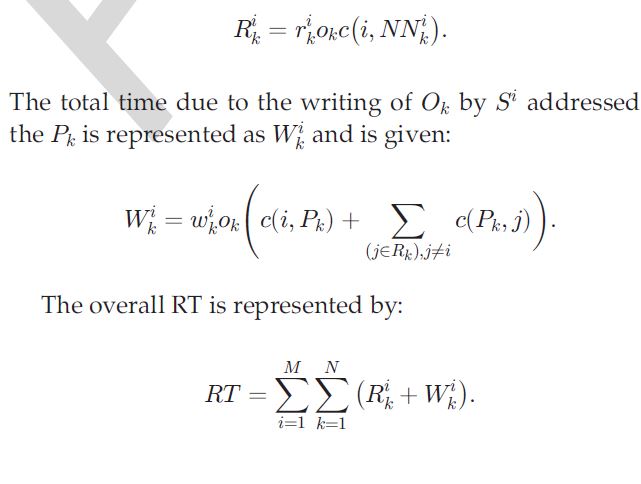
RT Replication time

storing Pk for Ok and NNik that represents the nearest nodestoring Ok. Whenever there is an update in Ok, the updated version is sent to Pk that broadcasts the updated version toall of the nodes in Rk. Let b(i,j) and t(i,j) be the total bandwidth of the link and traffic between sites Si and Sj, respectively. The centrality measure for Si is represented by ceni.Let colSi store the value of assigned color to Si. The colSi can have one out of two values, namely: open\_color and close\_color. The value open\_color represents that the node is available for storing the file fragment. The value close\_color shows that the node cannot store the file fragment.

Let T be a set of integers starting from zero and ending on a prespecified number. If the selected number is three, then T ¼ f0; 1; 2; 3g. The set T is used to restrict the node selection to those nodes that are at hop-distances not belonging to T.

Our aim is to minimize the overall total network transfer time or replication time (RT) or also termed as replication cost (RC). The RT is composed of two factors: (a) time due to read requests and (b) time due to write requests. The total read time of Ok by Si from NNi.

The total time due to the writing due of Ok by Si addressed to the Pk is represented as Wik and is given:

V 

The storage capacity constraint states that a file fragment can only be assigned to a node, if storage capacity of the node is greater or equal to the size of fragment. The bandwidth constraint states that bði; jÞ \_ tði; jÞ8i, 8j. The DROPS methodology assigns the file fragments to the nodes in a cloud that minimizes the RT, subject to capacity and bandwidth constraints.

**4.4 DROPS**

A successful attack on a node might put the data confidentiality or integrity, or both at risk. The aforesaid scenario can occur both in the case of intrusion or accidental errors. In such systems, performance in terms of retrieval time can be enhanced by employing replication strategies. However, replication increases the number of file copies within the cloud. Thereby, increasing the probability of the node holding the file to be a victim of attack as discussed. Security and replication are essential for a large-scale system, such as cloud, as both are utilized to provide services to the end user. Security and replication must be balanced such that one service must not lower the service level of the other.

In the DROPS methodology, we propose not to store the entire file at a single node. The DROPS methodology fragments the file and makes use of the cloud for replication. The fragments are distributed such that no node in a cloud holds more than a single fragment, so that even a successful attack on the node leaks no significant information. The DROPS methodology uses controlled replication where each of the fragments is replicated only once in the cloud to improve the security. Although, the controlled replicationdoes not improve the retrieval time to the level of full-scale replication, it significantly improves the security.

In the DROPS methodology, user sends the data file to cloud. The cloud manager system (a user facing server in

the cloud that entertains user’s requests) upon receiving the file performs:

(a) fragmentation,

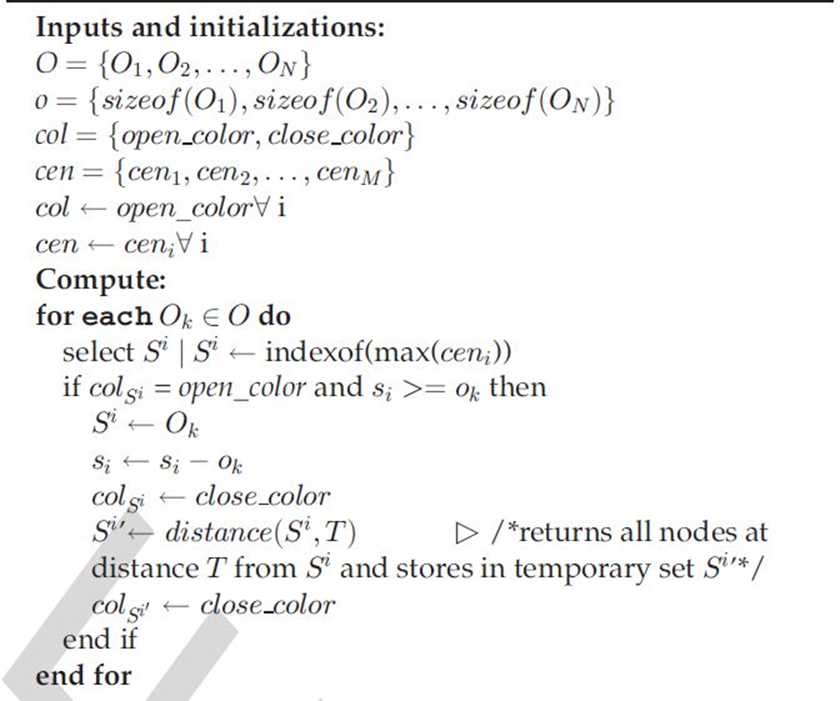
(b) first cycle of nodes selection and stores one fragment over each of the selected node, and

(c) second cycle of nodes selection for fragments replication. The cloud manager keeps record of the fragment placement and is assumed to be secure identity.

The fragmentation threshold of the data file is specified to be generated by the file owner. The file owner can specify

the fragmentation threshold in terms of either percentage or the number and size of different fragments. The percentage fragmentation threshold, for instance, can dictate that each fragment will be of 5 percent size of the total size of the file. Alternatively, the owner may generate a separate file containing information about the fragment number and size, for instance, fragment 1 of size 5,000 Bytes, fragment 2 of size 8,749 Bytes. We argue that the owner of the file is the best candidate to generate fragmentation threshold. The owner can best split the file such that each fragment does not contain significant amount of information as the owner is cognizant of all the facts pertaining to the data. The default percentage fragmentation threshold can be made a part of the service level agreement (SLA), if the user does not specify the fragmentation threshold while uploading the data file.

We primarily focus the storage system security in this work with an assumption that the communication channel between user and the cloud is secure.



Once the file is split into fragments, the DROPS methodology selects the cloud nodes for fragment placement. The selection is made by keeping an equal focus on both security and performance in terms of the access time. We choose the nodes that are most central to the cloud network to provide better access time. For the aforesaid purpose, the DROPS methodology uses the concept of centrality to reduce access time.

We implement DROPS with three centrality measures, namely:

1. Betweenness ,
2. closeness , and

(c) eccentricity centrality.

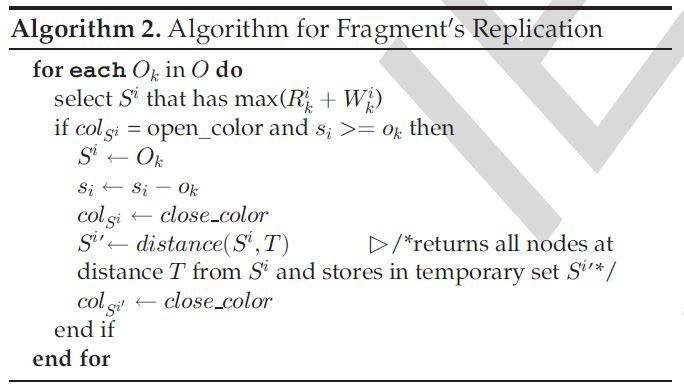
However, if all of the fragments are placed on the nodes based on the descending order of centrality, then there is a possibility that adjacent nodes are selected for fragment placement. Such a placement can provide clues to an attacker as to where other fragments might be present, reducing the security level of the data. To deal with the security

aspects of placing fragments, we use the concept of T-coloring that was originally used for the channel assignment problems.

We generate a non-negative random number and build the set T starting from zero to the generated random number. The set T is used to restrict the node selection to those nodes that are at hop-distances not belonging to T. For the said purpose, we assign colors to the nodes, such that, initially, all of the nodes are given the open\_color. Once a fragment is placed on the node, all of the nodes within the neighbourhood at a distance belonging to T are assigned close\_color. In the aforesaid process, we lose some of the central nodes that may increase the retrieval time but we achieve a higher security level. If somehow the intruder compromises a node and obtains a fragment, then the location of the other fragments cannot be determined. The attacker can only keep on guessing the location of the other fragments.

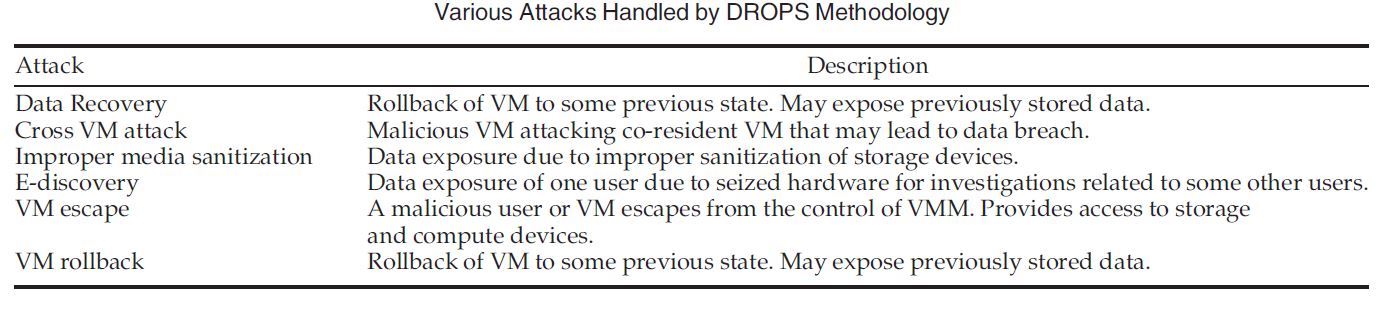
The probability of a successful coordinated attack is extremely minute. The process is repeated until all of the fragments are placed at the nodes. Algorithm 1 represents the fragment placement methodology.

In addition to placing the fragments on the central nodes, we also perform a controlled replication to increase the data availability, reliability, and improve data retrieval time. We place the fragment on the node that provides the decreased access cost with an objective to improve retrieval time for accessing the fragments for reconstruction of original file. While replicating the fragment, the separation of fragments as explained in the placement technique through T-coloring, is also taken care off. In case of a large number of fragments or small number of nodes, it is also possible that some of the fragments are left without being replicated because of the T-coloring. As discussed previously, T-coloring prohibits to store the fragment in neighborhood of a node storing a fragment, resulting in the elimination of a number of nodes to be used for storage. In such a case, only for the remaining fragments, the nodes that are not holding any fragment are selected for storage randomly. The replication strategy is presented in Algorithm 2.

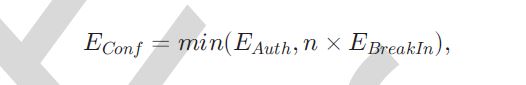


the cloud manager collects all the fragments from the nodes and reassemble them into a

single file. Afterwards, the file is sent to the user.



The number of compromised nodes must be greater than n because each of the compromised node may not give fragment in the DROPS methodology as the nodes are separated based on the T-coloring. Alternatively, an attacker has to compromise the authentication system of cloud [23]. The effort required by an attacker to compromise a node (in systems dealing with fragments/shares of data) is given in as:



where EConf is the effort required to compromise the confidentiality, EAuth is the effort required to compromise authentication, and EBreakIn is the effort required to com- promise a single node. Our focus in this paper is on the security of the data in the cloud and we do not take into account the security of the authentication system. Therefore, we can say that to obtain n fragments, the effort of an attacker increases by a factor of n. Moreover, in case of the DROPS methodology, the attacker must correctly guess the nodes storing fragments of file. Therefore, in the worst case scenario, the set of nodes compromised by the attacker will contain all of the nodes storing the file fragments.

we observe that the probability of the worst case to be successful is very low. The probability that some of the machines (average case) storing the file fragments will be selected is high in comparison to the worst case probability. However, the compromised fragments will not be enough to reconstruct the whole data. In terms of the probability, the worst, average, and best cases are dependent on the number of nodes storing fragments that are selected for an attack.

The DROPS methodology can handle the attacks in which attacker gets hold of user data by avoiding or disrupting security defenses. Table presents some of the attacks that are handled by the DROPS methodology. The presented attacks are cloud specific that stem from clouds core technologies. Table also provides a brief description of the attacks. It is noteworthy that even in case of successful attacks (that are mentioned), the DROPS methodology ensures that the attacker gets only a fragment of file as DROPS methodology stores only a single fragment on the node. Moreover, the successful attack has to be on the node that stores the fragment.

**5.STEGANOGRAPHY**

**5.1 INTRODUCTION**

**Stenography** is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity.

The word *steganography* is of Greek origin and means “concealed writing” from Greek words *steganos* meaning “covered or protected”, and *graphein* meaning “writing”.

The advantage of steganography over [cryptography](https://en.wikipedia.org/wiki/Cryptography) alone is that the intended secret message does not attract attention to itself as an object of scrutiny. Plainly visible encrypted messages, no matter how unbreakable they are, arouse interest and may in themselves be incriminating in countries in which [encryption](https://en.wikipedia.org/wiki/Encryption) is illegal.[[2]](https://en.wikipedia.org/wiki/Steganography#cite_note-2)

Whereas cryptography is the practice of protecting the contents of a message alone, steganography is concerned both with concealing the fact that a secret message is being sent and its contents.

Steganography includes the concealment of information within computer files. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size. For example, a sender might start with an innocuous image file and adjust the color of every hundredth [pixel](https://en.wikipedia.org/wiki/Pixel) to correspond to a letter in the alphabet. The change is so subtle that someone who is not specifically looking for it is unlikely to notice the change.

**5.2 HISOTRY OF STEGANOGRAPHY**:

* The first recorded uses of steganography can be traced back to 440 BC when Herodotus mentions two examples of steganography in his Histories.
* Demaratus sent a warning about a forthcoming attack to Greece by writing it directly on the wooden backing of a wax tablet before applying its beeswax surface.
* Wax tablets were in common use then as reusable writing surfaces, sometimes used for shorthand.
* Ancient Chinese wrote messages on fine silk, which was then crunched into a tiny ball and covered in wax. The messenger then swallowed the ball of wax.
* Special “inks” were important steganographic tools even during Second World War.
* During Second World War a technique was developed to shrink photographically a page of text into a dot less than one millimeter in diameter, and then hide this microdot in an apparently innocuous letter. (The first microdot has been spotted by FBI in 1941.)

**5.3 SYSTEM ANALYSIS (steganography)**

In this section discussed about data flow diagram, Entity relationship diagram. these things are represented as diagrams with proper notation.

**5.3.1 Data Flow Diagram**

The data flow diagram is one of the most improvement tools used by the system analyst DeMacro (1978) Nad Gand Sarson (1979) popularized the use if the data flow diagram as modeling tools through their structured system analysis methodologies.

A data flow diagram should be the first tool used by system analyst to model system components. These components are the system processes; the data used by this processes and external entities that interact with the system and the information flows in the system.

There are four kinds of system components

**5.3.2. Process**

Process show what system does. Each process has one or more data inputs and produce one or more data output, Circles in a data flow diagram represent process. Each process has unique name and number. This name and number appear inside the circle that represents the processes in a data flow diagram.

This process is represented as circle

**5.3.2. Data Stores:**

File or data store is depositary of data. They contain data that is retained in the system. Processes can enter the data into a data store or retrieve data from the data store. Each data store is represented by thin line in the data flow diagram and each data store has a unique name.

The data store is represented in form of a line

**5.3.3 External Entities:**

External entities are outside the system but they either supply input data into the system or use the system output, they are entities which the designer has no control. Square or rectangle may represent external entities that supply data into a system or some times called sources. External entities that use the system data are sometimes called sinks.

**5.3.4 Data Flows:**

Dataflow model the passage of data in the system and are represented lines joining system components. An arrow indicates the direction of the flow and the line labeled by the name of the data flow.

**5.3.5 UML Diagram:**

**Select Input File**

**Select Output File**

**Embedded Data**

**Select Key**

**Embedded file**

**Select encrypt file**

**Enter New Text**

**Select Key**

**Extract File**

**View Output**

**Data Flow Diagram**

**Select Input Audio File**

**Select Output Audio File**

**Embedded Data**

**Select Key**

**Embedded Audio File**

**Select Encrypt Audio File**

**Extract Audio File**

**View Output**

**Select Key**

**Enter New Text**

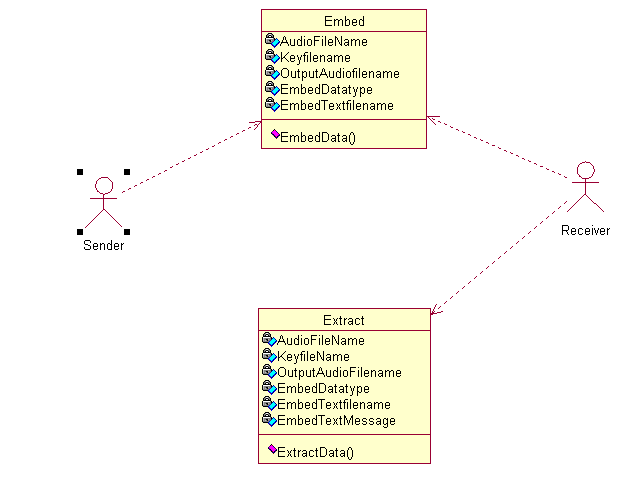
**USECASE DIAGRAM1: (EMBEDDING MODULE):**

sender

**USECASE DIAGRAM2: (EXTRACTION MODULE):**

Receiver

**CLASS DIAGRAM:**



**Software development life cycle:**

Authorization

Certificate

Students

Training

Problem Analysis:

TEXT HIDING IN FILE

Front-end Asp.net

Back-end: SQL Server

Code behind: c#.net

Networking Design:

Internet Explorer 6.0

Coding

Validationnnn

Testing

Document

Reports

Certificate

Complete Certificate

Programmer Training Certificate

Authorization Signatory

Learning Process

DB

1. **BACK END DESIGN**

**6.1 Introduction**

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2005 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2005 Meta Data Services. References to the component

Now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Service

SQL-SERVER database consist of six type of objects, They are,

1. TABLE 2. QUERY

3. FORM 4. REPORT 5. MACRO

**TABLE:**

A database is a collection of data about a specific topic.

**VIEWS OF TABLE:**

We can work with a table in two types, 1. Design View

2. Datasheet View

**Design View**

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold. **Datasheet View**

To add, edit or analyses the data itself we work in tables datasheet view mode.

**QUERY:**

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dataset (if you edit it) or a snapshot (it cannot be edited).Each time we run query, we get latest information in the dataset. Access either displays the dataset or snapshot for us to view or perform an action on it, such as deleting or updating.

**FORMS:**

A form is used to view and edit information in the database record by record .A form displays only the information we want to see in the way we want to see it. Forms use the familiar controls such as textboxes and checkboxes. This makes viewing and entering data easy.

**Views of Form:**

We can work with forms in several primarily there are two views, They are,

**1. Design View**

**2. Form View**

**Design View**

To build or modify the structure of a form, we work in forms design view. We can add control to the form that are bound to fields in a table or query, includes textboxes, option buttons, graphs and pictures.

**Form View**

The form view which display the whole design of the form.

**REPORT:**

A report is used to vies and print information from the database. The report can ground records into many levels and compute totals and average by checking values from many records at once. Also the report is attractive and distinctive because we have control over the size and appearance of it.

**MACRO:**

A macro is a set of actions. Each action in macros does something. Such as opening a form or printing a report .We write macros to automate the common tasks the work easy and save the time.

**MODULE:**

Modules are units of code written in access basic language. We can write and use module to automate and customize the database in very sophisticated ways.It is a personal computer based RDBMS. This provides most of the features available in the high-end RDBMS products like Oracle, Sybase, and Ingress etc.

**6.2 INTERFACE DESIGN**

The ODBC (Open Database Connectivity) interface is a pure .NET to

execute SQl statement. The ODBC provides a set classes and interfaces that can be used by developers to write database applications. Basic ODBC interactions in its simplest form, can be broken down into four steps:

1. Open a connection to the database.

2. Execute a SQL statement

3. Process the result

4. Close the connection to the database

**6.3 TABLE AND DATABASE DESIGN:**

**Admin Login Table**

**6.3.1 Normalization:**

Normalization is the process of strutting relational database schema such that most ambiguity is removed. The stage of normalization are referred to as forms and progress from the least restrictive(first normal form)through the most restrictive(Fifth normal form), generally , most database designers do not attempt to implement anything higher then normal form of Boyce code Normal Form.

**6.3.1.1FIRST NORMAL FORM**:

A relation is said to be in First normal form (INF) if and each attributed of the relation is atomic. More simply, to be INF, each column must contain only a single value and each now contain in the same column.

**6.3.1.2 SECOND NORMAL FORM:**

In the Second normal Form, a relation must first fulfill the requirement to be in first Normal Form. Additional, each donkey attribute in the relation must be functionality dependent upon the primary key**.**

**6.3**

**.1.3 THIRD NORMAL FORM:**

A table is said to be in third normal form and every non key attribute is functionality dependent only on the primary key. This normalization process is applied to this system and the normalized tables are given in the above section.

**REFERENCE**

1.Aleksander Hudic, Shareeful Islam, Peter Kieseberg, Edgar R. Weippl. Data   confidentiality using fragmentation in cloud computing. Int. J. Communication Networks   and Distributed Systems (2012)

2.Yu-Chau liu , Yu-Tau Ma, Hai-Su Zhang , De-Yi Li,  Gui-Sheng Chen. Data for T-Coloring in Cloud. Int. J. For automation and computing .(2011)

3.Arvind Kumar, Km. Pooja. Stegonography-A data hiding technique.(2010). Int. J. For Computer Applications

4 Vijay Kumar Sharma , Dr. Devesh Km , Dr. Pratistha Mathur.  A Study of Steganography based data hiding technique.(2015)

5. Varsha ,Amit Wadhwa , Swati Gupta. Study of Security Issues In cloud. Int. J. for Computer Science and Mobile Computing .(2015)

6.Mazhar Ali, Samee U. Khan, Keqin Li. Division and replication of data in cloud for Optimal Performance  and Security.(2012)

**CONCLUSION**

Outsourcing data to a third-party administrative control, as is done in cloud computing, gives rise to security concerns. The data compromise may occur due to attacks by other users and nodes within the cloud. Therefore, high security measures are required to protect data within the cloud. However, the employed security strategy must also take into account the optimization of the data retrieval time. In this project, we have proposed division and replication of data in the cloud for optimal performance and security (DROPS) that collectively approaches the security and performance issues. In the DROPS methodology, we have divided a file into fragments, and replicate the fragmented data over the cloud nodes. Each of the nodes stores only a single fragment of a particular data file that ensures that even in case of a successful attack, no meaningful information is revealed to the attacker. The higher level of security with slight performance overhead was observed.

**BIO DATA**

## Students involved in the project team include:

## Himanshu Doley 730/IT/15, himanshu8mile@gmail.com

## Navdeep Redhu 749/IT/15, navdeepr1234@gmail.com

## Karan 734/IT/15, meena.karan9k@gmail.com