**DELHI TECHNOLOGICAL UNIVERSITY**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



**DIS Project Design Document**

**CO – 407**

7th semester (2020 – 2021)

**Submitted to:**  **Submitted by:**

Dr. Divyashikha Sethia Amol Gupta: 2K17/CO/052

COE Department Arnav Kumar: 2K17/CO/074

**Introduction**

Sockets are used to establish communication between processes especially in client server model. Sockets are IP addresses concatenated with port number. When a process at client system wants to communicate and request for information from server, it is assigned a socket consisting of IP address of that system with port number. Port number should be greater than 1024 as port numbers below this are standard.

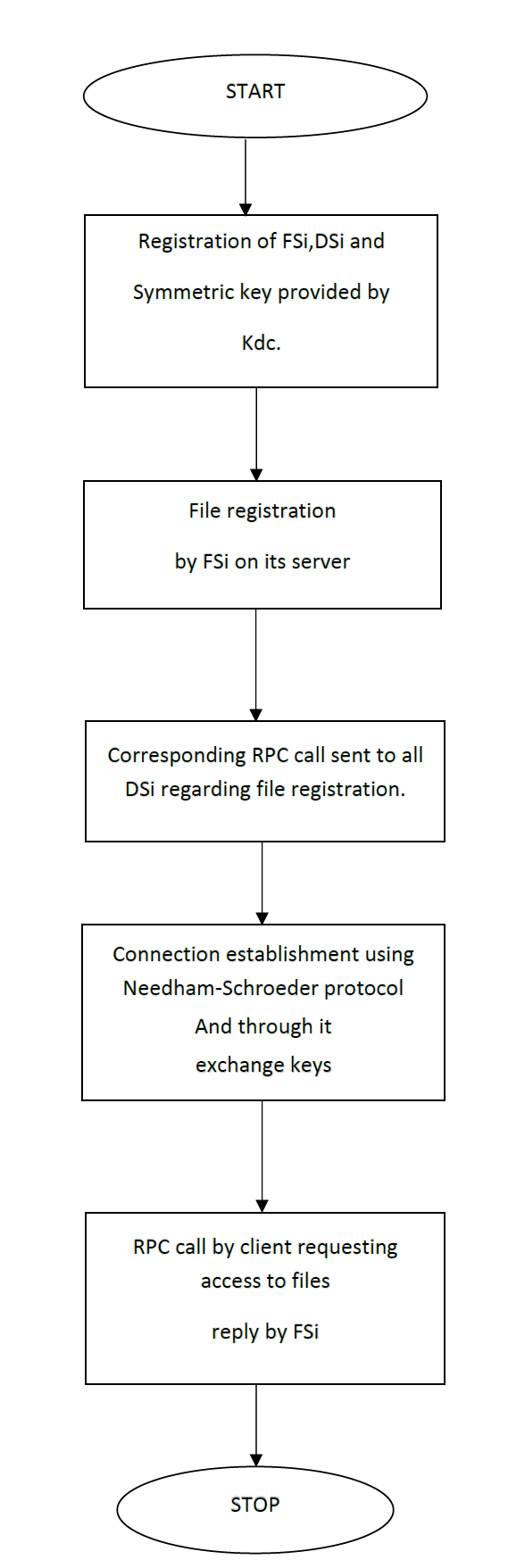
Remote Procedure Calls (RPCs) are used when a program wants to call a procedure located on another system connected over a network. The calling process is suspended and execution of called procedure takes place by passing message containing identifier of the function to execute and the parameters to pass to that function. Information is returned back to the requester in a separate message after execution of the function.

When the client invokes a remote procedure the RPC system calls the appropriate stub passing it the parameters provided to the remote procedure. This stub locates the port on the server and marshalls the parameters. The stub then transmits the message to the server. A similar stub on the server side receives this message and invokes the procedure on the server. Return values are passed back using the same technique.

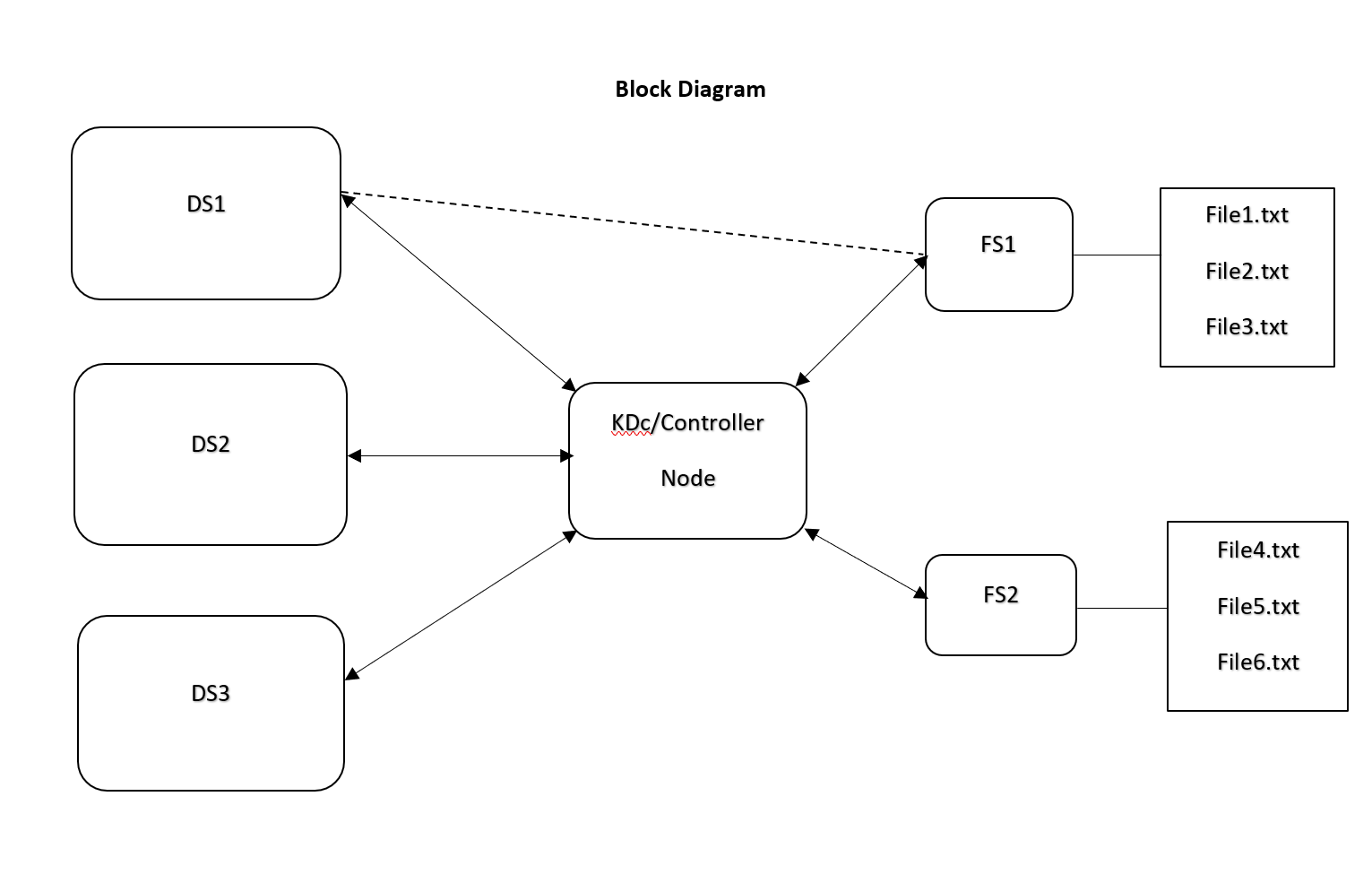
**Problem description**

The purpose of this project is to design a secure file system which allows distributed system nodes DSi to access the remote files stored on the remote file servers FSi in a secure manner using RPCs. This will familiarize us with RPC, File system and security issues in Distributed System.

**Block diagram and flow chart**



Flowchart



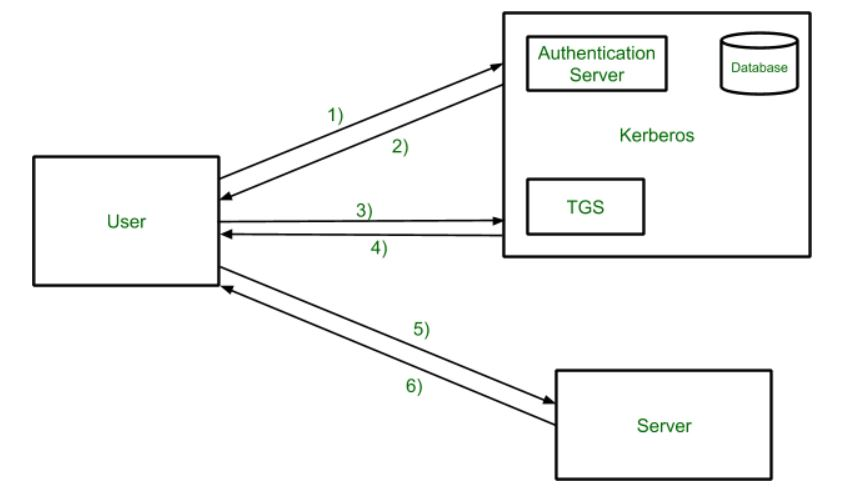
Block diagram

**Protocol**

We will be using the Kerberos protocol which is explained below.

Kerberos is an authentication protocol for client/server applications. It provides a centralized authentication server whose function is to authenticate users to servers and servers to users. In Kerberos Authentication server and database is used for client authentication. Kerberos runs as a third-party trusted server known as the Key Distribution Center (KDC). Each user and service on the network is a principal.

The KDC supplies tickets and generates temporary session keys that allow a user to securely authenticate to a service. The KDC stores all the secret symmetric keys for users and services. There are two servers within the KDC, the Authentication Server (AS) and the Ticket Granting Server (TGS).



Kerberos overview

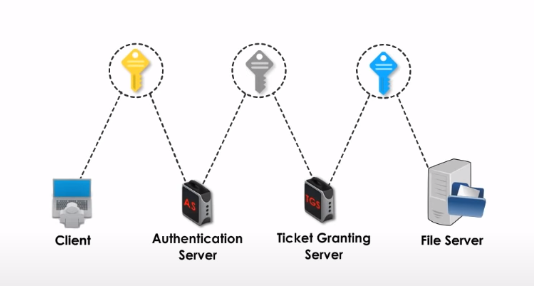
The main components of Kerberos are:

* Authentication Server (AS): The AS server confirms that a known user is making an access request and issues the ticket for TGS.
* Database: AS retrieves the client’s password from the database based on userID.
* Ticket Granting Server (TGS): The TGS confirms that a user is making an access request to a known service and issues service tickets.

The use of these components is explained in detail in the steps followed in Kerberos given below.

Steps followed in Kerberos

1. The client sends a request to the AS which is encrypted using a password.
2. When the AS gets the client’s request, it retrieves the client’s password from the database based on userID and uses his password as a key to decrypt his request. The password is a shared secret key between the AS and the client. After verifying the client, AS sends back a ticket called Ticket Granting Ticket (TGT) encrypted with another secret key.
3. After the client gets the TGT, he sends this ticket to the TGS along with his request.
4. When the TGS gets the TGT, it decrypts the ticket with the secret key shared with AS. Then TGS issues the client a token which is encrypted with another secret key. This third secret key is shared between the TGS and the file server.
5. The client sends the token to the file server.
6. When the file server gets the token, it decrypts the token with the secret key shared with TGS. The file server allows the client to use its resources for certain period of time according to the token.



Key sharing in Kerberos

**Platforms:** Visual Studio Code, Python Environment, xmlrpc module, socket module.