System design

# 1. Requirements

- Server is able to:

+ Sync file with client with encryption

+ Have a mechanism to prevent DDos

+ Detect new file stored

+ Have a file sending mechanism using queue like MQTT

+ Store encrypted file

+ Exchange new AES key everyday

+ Have simple GUI providing the functions:

+ Decrypt stored files to view

+ Delete decrypted files

+ restore previous working state after restarting

- Client is able to:

+ Sync file with server with encryption

+ Store encrypted file

+ Exchange new AES key everyday

# 2. Modules’ function in server

## 2.1. Transmission

- Send file

- Receive file

- Put file to queue if cannot send

- Send key

- Receive key

- Form packet following format

- Hash utility

## 2.2. ED

## 2.3. StorageModule

# 3. Modules’ function in server

# 4. Technology description

## 4.1. Message exchanging protocol: Protocol 3Vil

|  |  |
| --- | --- |
| **Byte** | **Description** |
| 0 | Flag |
| 1->4 | Size of payload |
| 5.. | Payload |

### 4.1.1. Flag byte description

|  |  |
| --- | --- |
| **Bit** | **Description** |
| 0 | If there are successive incoming packets |
| 1,2,3,4 | Packet type |

### 4.1.2. Packet type

|  |  |
| --- | --- |
| **Bit value** | **Description** |
| 0000 | Send data |
| 0001 | Send hash |
| 0010 | Send public key |
| 0011 | Send asymmetric key |
| 0100 | Response hash permitted |
| 0101 | Response hash prohibited |
| 0110 | Request public key from server |
| 1111 | Response Internal Err |