

Networking and Security

Description:

You will create a "Dropbox/Sugarsync/GoogleDrive/MS/Apple..."-like system to securely store and share files "in the cloud" (really on a remote server). Users will select unencrypted (plain) files (of any type) from a menu. Then that file will be encrypted and then sent to a server (service). Users will be able to selectively, or as a group, retrieve (and decrypt) those files later.

You should provide both the client side and server side of your implementation.

Hint:

C clients may look like this:

```
sockfd = socket(AF_INET, SOCK_STREAM, 0);
address.sin_family = AF_INET;
address.sin_addr.s_addr = inet_addr("127.0.0.1");
address.sin_port = htons(9734);
len = sizeof(address);
result = connect(sockfd, (struct sockaddr *)&address, len);
write(sockfd, &ch, 1);
read(sockfd, &ch, 1);
printf("char from server = %c\n", ch);
close(sockfd);
exit(0);
```

And servers like this:

```
server_sockfd = socket(AF_INET, SOCK_STREAM, 0);
server_address.sin_family = AF_INET;
server_address.sin_addr.s_addr = htonl(INADDR_ANY);
server_address.sin_port = htons(9734);
server_len = sizeof(server_address);
bind(server_sockfd, (struct sockaddr *)&server_address, server_len);
listen(server_sockfd, 5);
while(1) {
    printf("server waiting\n");
    client_sockfd = accept(server_sockfd,
        (struct sockaddr *)&client_address, &client_len);
    read(client_sockfd, &ch, 1);
    write(client_sockfd, &ch, 1);
    close(client_sockfd);
}
```

More Hints:

In Python, there are several libraries (depends on Python version you use.)

There are encrypt/decrypt libraries for all languages (AES, DES, etc)

Consider using "websockets", Java, Python or PHP

Bonus: Use a public server (AWS, Google, etc) for hosting, browser (client) interface, Authentication (user,pwd), Certificates, Digital Signing, quotas

<http://www.linuxjournal.com/content/tech-tip-really-simple-http-server-python>
<https://docs.python.org/2/library/simplehttpserver.html>

<https://fragments.turtlmeat.com/javawebserver.php>

<https://astaxie.gitbooks.io/build-web-application-with-golang/en/03.2.html>
<https://tutorialedge.net/post/golang/creating-simple-web-server-with-golang/>

<https://www.dlitz.net/software/pycrypto/>
<http://python-guide-pt-br.readthedocs.io/en/latest/scenarios/crypto/>

<https://golang.org/pkg/crypto/>