Project Description

We have programed both server and client (instead of using existing server).

Therefore, we have two packages:

1. miniPro – for the server
2. client – for the secured client
   *will be mentioned from now on as 'server' and 'client' in this document.

The files in 'server' are:

- mainFTPServer.py
- registerServer.py
- server.py

The files in 'client' are:

- AESCipher.py
- client.py
- encryptedClient.py
- encryptFiles.py
- SecuredFTPClient.py

We'll explain now the functionality of every file and it's content.

In client:

client.py stands for all functions that represent project's server's options: register, connect, login, upload fie, download file, delete file, rename file, overwrite file, show files in directory.
Also, it has 'makeKey' and 'padKey' functions for the secured client's secrets for password, encrypting and decrypting, and authentication.

AESCipher.py is python file needed for encrypting files' names.

encryptedClient.py is a wrapping class for 'client.py' with all the functions used in the server, but with encryption, decryption and authentication functions. These are the real functions called by the user. This class uses encryptFiles.py and AESCipher.py.

encryptFiles,py stands for all the encryption 'for real'. In other words, encrypting, decrypting and authentication are done with more functions like reading files and helper functions. All the encoding is done in this class for making the functions in 'encryptedClient.py' clear as possible.

SecuredFTPClient.py stands for the GUI.

In server:

mainFTPServer.py uses 'registerServer.py' and 'server.py' for initializing the server and running it.

registerServer.py creates a socket and listens to it. When the client sends a 'register' command for the server - the command is passed through this socket and a check of user name validity is done, before register command is sent to FTP server.

server.py initializes the FTP server.