About the Script

The script I named *AutoEncrypt* is like a dropbox program, that automatically encrypts the contents of the folder, and places it in a directory in the same location as the script is located. The unencrypted version is then deleted for safety. When decryption is needed, only the correct passphrase will allow the decryption to take place. While there's more information below and in accompanying documents, it's worth noting here that I chose to replicate the names (including extension) of the files so as to not cause suspicion with .gpg extensions.

Upon initialization, the script will generate a key for the purpose of encryption using RSA 3072. After a specified time interval, the script will pull files in the *AutoEncrypt* directory, make encrypted copies, then delete the originals. To decrypt, a user must know the password set at the initialization stage else nothing will happen.

Purpose

To provide a convenient and easy method of encrypting documents and files. Instead of having to use the GPG commands constantly, a user need only place the files into the AutoEncrypt directory

Justification

The more inconvenient a process is, the more likely a human being will become complacent in their security. While the average home user will not need this and will be satisfied with just their windows login as security, larger companies with liability and security expectations from clients/Governments cannot ignore the need for tight security. This software acts as a last stand, where if an attacker was able to break through security and steal the documents, they'd still face the monolithic task of decrypting them. The idea of this script is to give administrators a way for non-tech professionals to comply with security policies of the company, or an easy way for home users to secure their sensitive documents easily.

Potential Users

I imagine this tool being used by system administrators, as some knowledge of linux commands is still required to set-up and use decryption capabilities. The act of encryption though is automated. A more diligent personal user may also use this with elementary knowledge of linux commands.

How to Use the Script

Upon running the script for the first time, a user must set the passphrase for the key pairing, choose where they want the directory for files that need encryption and then another directory for decrypted files to be sent to. Running the script again will take one to the menu where the user can find about their preferences, choose to decrypt one specific file, or decrypt everything. There also exists an option to reset the program, which will decrypt all the files residing in the program (of course after the passphrase is entered), and move it to the decryption folder initially set by the user. Afterwards, the data file for the program and the encryption key is deleted. For safety, if you initialize with the same directory locations, the decrypted directory will still hold the decrypted files, however if they needed to be encrypted in the first place, they should be re-encrypted as soon as possible.

Additionally running the script with “-f“ is to force the program to immediately encrypt the contents of the *AutoEncrypt* directory, without having to wait for the allotted time period to pass.

Running with “-i” is to initialize the program, however this will not work if it has already been initialized, in which case the user must reset the program via the menu.

Running the program with “-a” is for cron jobs, which will also encrypt the contents of *AutoEncrypt* and give a small alert in the top right of the screen to confirm to the user that the directory has been encrypted. In the event of a reset, without re-initializing the user will be notified that the encryption failed because the program needs initialization by a similar alert.