Assignment 4 README

Team:

William Ritson (writson@ucsc.edu)
Jacob Janowski (jnjanows@ucsc.edu)
Payton Schwarz (peschwar@ucsc.edu)
Bryan McCoid (bmccoid@ucsc.edu)
Kfir Dolev (kdolev@ucsc.edu)

Important Note! - Our kernel config is named PAUL
You can build it with `sudo make buildkernel KERNCONF=PAUL`

1. Abstract

We offer a method to test our project, as well a description of how to use each of its components

2. Overview

1. Test Script
2. CryptoFS Specs
3. Protectfile Specs
4. Setkey/Getkey Specs

1. Test Script

We provide a shell script to test the functionality of the cryptofs file system. The script should verify the following properties

1. Protectfile should encrypt files when run with -d
2. Protectfile should decrypt previously encrypted files when run with -e
3. Protectfile should raise an error if trying to encrypt an encrypted file (eg a file with sticky bit set to true) or decrypt an encrypted file.
4. Cryptofs should read encrypted files, so long as their sticky bit is true
5. Cryptofs should read plaintext files, so long as their sticky bit is false
6. Cryptofs should encrypt any data written to a file whose sticky bit is set to true
7. Cryptofs should not encrypt any data written to a file with a sticky bit set to zero

Running the Test

Step 1:Navigate to the test directory

- `cd <repopath>/asgn4`

Step 2: Execute the test script as root

- On user account - `sudo sh test.sh`

- On root account - `sh test.sh`

Step 3: Clean generated folders (optional)

- Do this you want to clean up folders generated by test (if you want to run test.sh multiple times, be sure to clean in between)
- Run `sudo sh cleantest.sh`

2. Cryptofs

Mount CryptoFS - `mount -t cryptofs <dir1> <dir2>`

3. Protectile

Located in the *asgn4/aes* directory.

Build - `make`
Encrypt file - `sudo protectfile -e <key1> <key2> <filename>`
Decrypt file - `sudo protectfile -d <key1> <key2> <filename>`

Where <key1> and <key2> are 32 bit integers.

4. Setkey/Getkey

User program in the asgn4 directory

Build - `cc setkey.c -o setkey`
User Program - `setkey <key1> <key2>`
System call - `syscall(548, <key1> <key2>)`

Where <key1> and <key2> are 32 bit integers.