Programming Assignment 5:
An Encrypted Filesystem

CSCI 3753 - Operating Systems
University of Colorado at Boulder
Spring 2012
By Andy Sayler and Junho Ahn and Richard Han
Adopted from Assignment by Chris Wailes

Due Date: Friday, April 27th, 2012 11:55pm

1 Assignment Introduction

2 Your Task

2.1 Dependencies

This assignment has severl dependines that must be isnatlled in order for teh rpeovided code to build correctly. On Debian or Ubuntu, start by running `sudo apt-get update` to update your package list. Then run `sudo apt-get install <package(s)>` to install the following packages:

- `fuse-utils`
- `libfuse-dev`
- `openssl`
- `libssl-dev`
- `libssl-doc` (optional)
- `libssl1.0.0` or `libssl0.9.8`
- `attr`
- `attr-dev`

You will need a working Internet connection in order to insure these packages install correctly. Note that you can also specify multiple packages in a single `sudo apt-get install <package(s)>` call. Some packages may have their own dependencies, but `apt-get` will automatically take care of installing these for you.

3 What's Included

We provide some code and examples to help get you started. Feel free to use it as a jumping off point (appropriately cited).

- **Makefile** A GNU Make makefile to build all the code listed here.
- **README** As the title so eloquently instructs: read it. Provides usage instructions and examples for files listed here.
- **fusehello.c** A basic "Hello World" FUSE example. See `README` for usage instructions.
- **fusexmp.c** A basic FUSE mirrored filesystem example that mirrors the root directory (/) and supports most standard operations. See `README` for usage instructions.
- **xattr-util.c** A basic extended attribute manipulation program. Provides an example of proper Linux xattr use. See `README` for usage instructions.
- **aes-crypt-util.c** A basic AES encryption program using the local aes-crypt library (see `aes-crypt.h`) and the OpenSSL EVP API[9]. See `README` for usage instructions.
- **aes-crypt.h** A basic AES file-pointer centric encryption library interface. Implemented in `aes-crypt.c`.
- **aes-crypt.c** A basic AES file-pointer centric encryption library implementation. Uses the OpenSSL EVP API [9].

4 What You Must Provide

When you submit your assignment, you must provide the following as a single archive file:

1. A copy of your Encrypted Filesystem FUSE code
2. A copy of any supporting code used by your filesystem
3. A makefile that builds any necessary code
4. A README explaining how to build and run your code

5 Grading

40% of you grade will be based on implementing a filesystem that meets the following criteria. You will be expected to provide functional proof of the following criteria during your grading session. The rubric below shows grading criteria:

- **+10 points:** Filesystem properly mirrors target directory specified at mount time.
- **+10 points:** Filesystem uses extended attributes to differentiate between encrypted and unencrypted files.
- **+10 points:** Filesystem can transparently read and write securely encrypted files
- **+10 points:** Filesystem can transparently read and update unencrypted files

If your code does not build or run without errors, you will not receive any credit on the objective portion (40%) of your assignment.

If your code generates warnings when building under gcc on the VM using `-Wall` and `-Wextra` you will be penalized 1 point per warning. In addition, to receive full credit your submission must:

- Meet all requirements elicited in this document
- Code must adhere to good coding practices.
- Code must be submitted to Moodle prior to due date.

The other 60% of your grade will be determined via your grading interview where you will be expected to explain your work and answer questions regarding it and any concepts related to this assignment.

6 Obtaining Code

The starting code for this assignment is available on the Moodle and on github. If you would like practice using a version control system, consider forking the code from github. Using the github code is not a requirement, but it will help to insure that you stay up to date with any updates or changes to the supplied codebase. It is also good practice for the kind of development one might expect to do in a professional environment. And since your github code can be easily shared, it can be a good way to show off your coding skills to potential employers and other programmers.

Github code may be forked from the project page here:
`https://github.com/asayler/CU-CS3753-2012-PA5`.

7 Resources

Refer to your textbook and class notes on the Moodle for an overview of filesystems.

If you require a good C language reference, consult K&R[6]. If you need an updated C99 reference see Harbison & Steele[5].

The Internet[12] is also a good resource for finding information related to solving this assignment.

You may wish to consult the man pages for the following items, as they will be useful and/or required to complete this assignment. Note that the first argument to the "man" command is the chapter, insuring that you access the appropriate version of each man page. See `man man` for more information. Not all of these man pages are installed be default. Install the previously discussed dependencies or consult an online man page repository if you can not locate a specific man page on your system.

- `man 1 make`
- `man 1 fusermount`
- `man 5 attr`
- `man 2 setxattr`
- `man 2 getxattr`
- `man 2 listxattr`

- `man 2 removexattr`
- `man 3 EVP`
- `man 3 EVP_CipherUpdate`
- `man 3 crypto`
- Many of the system calls used in the FUSE examples also have man pages

In addition, you may find a number of the references in the bibliography helpful.

References

[1] freedesktop.org. *Guidelines for extended attributes.* `http://www.freedesktop.org/wiki/CommonExtendedAttributes`.

[2] FUSE. *Filesystems in Userspace.* `http://fuse.sourceforge.net/`.

[3] FUSE. *Fuse Doxygen API Reference.* `http://fuse.sourceforge.net/doxygen/index.html`.

[4] FUSE. *Fuse Wiki.* `http://sourceforge.net/apps/mediawiki/fuse/index.php?title=Main_Page`.

[5] Harbison, Samuel and Steele, Guy. *C: A Reference Manual.* Fifth Edition: 2002. Prentice Hall: New Jersey.

[6] Kernighan, Brian and Dennis, Ritchie. *The C Programming Language.* Second Edition: 1988. Prentice Hall: New Jersey.

[7] OpenSSL. *Cryptography and SSL/TLS Toolkit.* `http://www.openssl.org/`.

[8] OpenSSL. *OpenSSL Documents.* `http://www.openssl.org/docs/`.

[9] OpenSSL. *OpenSSL EVP Documentation.* `http://www.openssl.org/docs/crypto/EVP_EncryptInit.html`.

[10] Pillai, Saju. *Openssl AES encryption example.* Decemper 9th, 2008. `http://saju.net.in/blog/?p=36`.

[11] Pfeiffer, Joseph. *Writing a FUSE Filesystem: a Tutorial.* January 10th, 2011. `http://www.cs.nmsu.edu/~pfeiffer/fuse-tutorial/`.

[12] Stevens, Ted. *Speech on Net Neutrality Bill.* 2006. `http://youtu.be/f99PcP0aFNE`.