ENCRYPTED FILE SYSTEM WITH FUSE

PROJECT SYNOPSIS

SUBMITTED BY

MELVIN VARKEY – AM.EN.P2CSN14004

ROHIT KIRAN – AM.EN.P2CSN14008

Introduction

- What is FUSE?
- File encryption
- Why Encrypt?
- Keys for Encryption and Decryption

What is FUSE ?

- FUSE – **F**ile System in **Use**rspace
  - An operating system mechanism for Unix-like computer operating systems that lets non-privileged users create their own file systems without editing kernel code.
  - Achieved by running file system code in user space while the FUSE module provides only a "bridge" to the actual kernel interfaces.
  - Implemented as a loadable kernel module.

The Encrypted File System (EncFS)

- Provides an encrypted file system in user space.
- Runs without any special permission.
- Uses FUSE library and Linux kernel mode to provide the filesystem interface.
- Provides security
- Two directories involved – The "**source directory**" which holds the encrypted files and "**mountpoint**" which provides the unencrypted view of files.
- Files are encrypted using volume key – stored encrypted in the source directory – password is used to decrypt

Implementation

Fuse Installation and mount

- http://fuse.sourceforge.net/

- ./configure
- make
- make install
- fusexmp.c
- fusexmp /mnt/fuse -d
- fusermount

Source Code

./doc: contains FUSE-related documentation

./include: contains the FUSE API headers, which you need to create a file system. The only one you need now is **fuse.h**.

./lib: holds the source code to create the FUSE libraries that you will be linking with your binaries to create a file system.

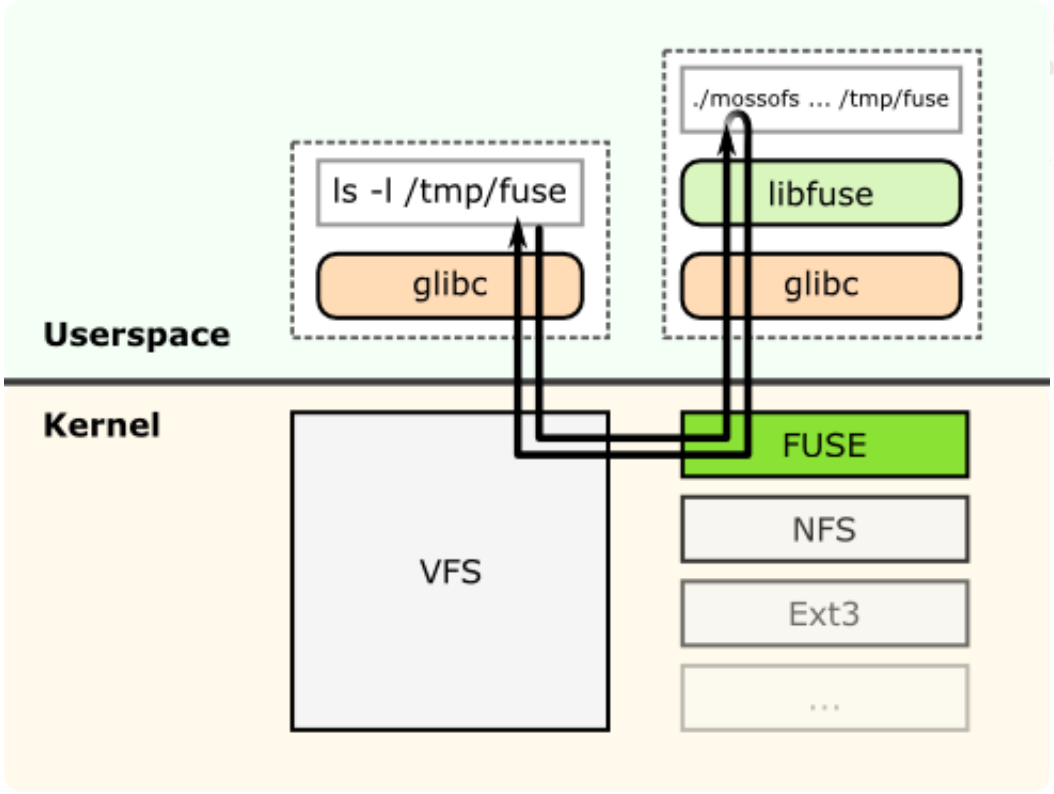
./util: has the source code for the FUSE utility library.

./example: contains samples for your reference.

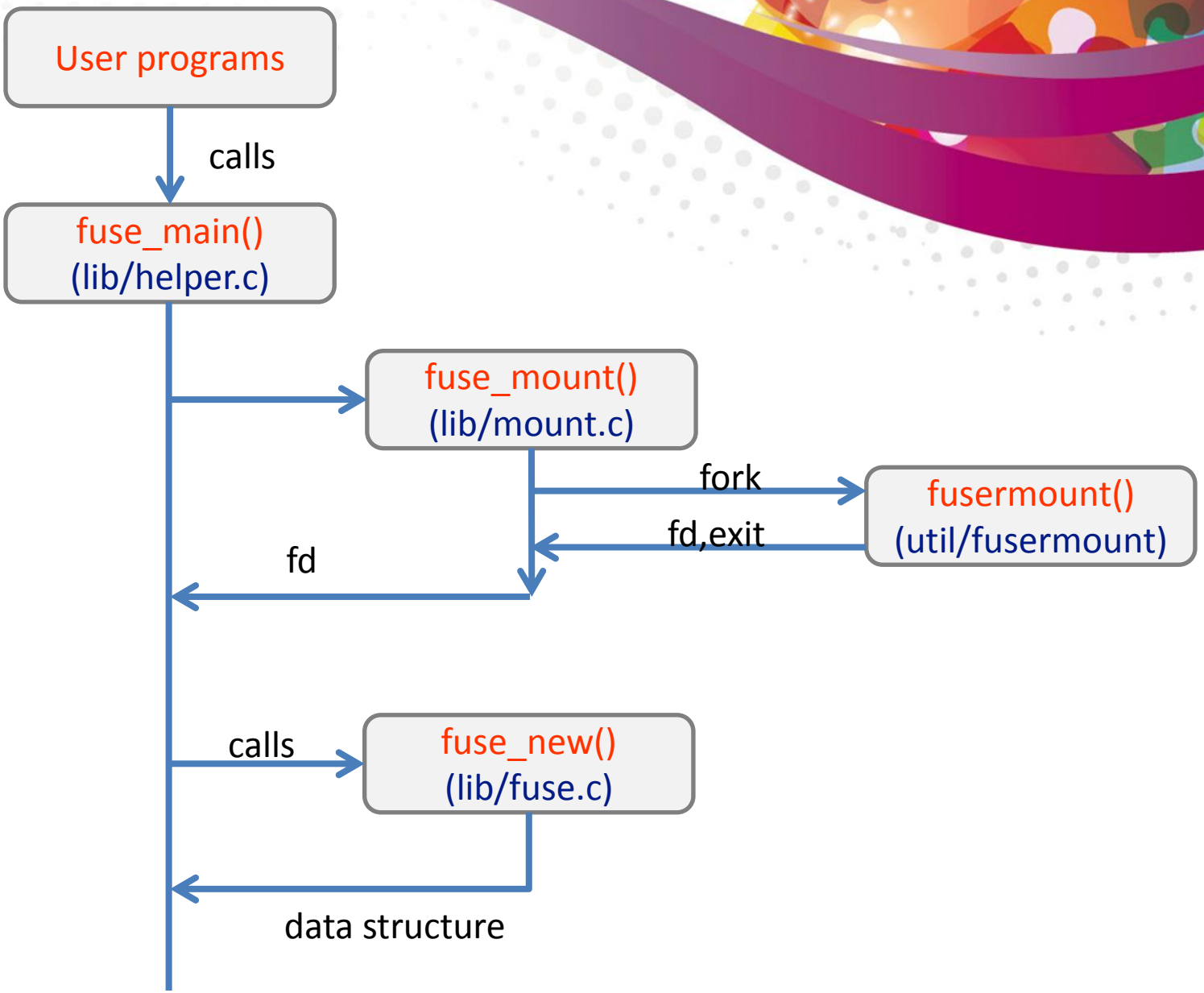
FUSE Structure

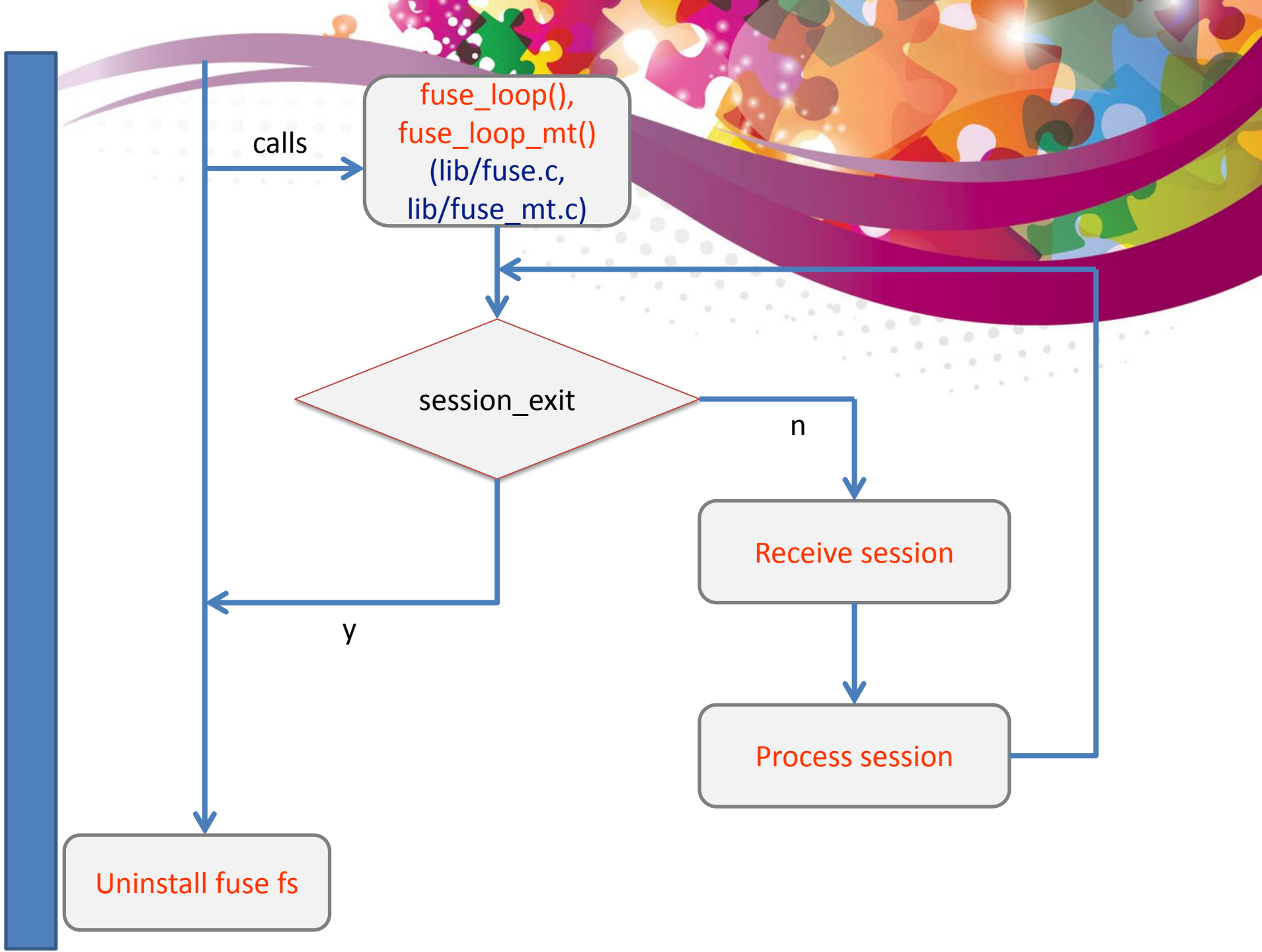
- **FUSE kernel module (fuse.ko)**
  - inode.c, dev.c, control.c, dir.c, file.c
- **LibFUSE module (libfuse.*)**
  - helper.c, fuse_mt.c, fuse.c, fuse_lowlevel.c, fuse_loop.c, fuse_loop_mt.c, fuse_session.c
- **Mount utility (fusermount)**
  - fusermount, mount.fuse.c, mount_util.c, mount.c, mount_bsd.c,

FUSE Structure



The FUSE kernel module and the FUSE library communicate via a special file descriptor which is obtained by opening */dev/fuse*. This file can be opened multiple times, and the obtained file descriptor is passed to the mount syscall, to match up the descriptor with the mounted filesystem.


User programs
calls
fuse_main()
(lib/helper.c)
fuse_mount()
(lib/mount.c)
fork
fusermount()
(util/fusermount)
fd,exit
fd
calls
fuse_new()
(lib/fuse.c)
data structure


calls
fuse_loop(),
fuse_loop_mt()
(lib/fuse.c,
lib/fuse_mt.c)
session_exit
n
y
Receive session
Process session
Uninstall fuse fs

How it Works !!!

FUSE user space file system will link with FUSE library and provides:

- Register file operation methods w/ library
  - struct fuse_operations
  - getattr, mknod, create, read, write, readdir,
  - readlink, getdir, mknod, chmod, etc.
- Mount point and options
- FUSE library calls the mount() system call
  - filesystem type is “fuse”
  - filehandle of /dev/fuse passed as option
- Filesystem calls are passed to FUSE library which invokes associated fuse operation in FUSE filesystem.

Encryption

- FileNode sends read/write requests through FileIO instance
- FileIO instances form chain
- BlockFileIO layer converts requests into block-oriented requests
- AES - Encryption and Decryption
- 128 bit
- Each filesystem uses a randomly generated key
- File name encryption options
- key size
- filesystem block size
- block encryption & stream encryption

Encryption

- aes.h
- aes-crypt.c
- Key and IV
- struct xmp_state
- AES_ENCRYPT 1
- AES_DECRYPT 0
- AES_PASSTHRU -1
- Rounds
- do_crypt(FILE* in, FILE* out, int action, char* key_str);

Conclusion

- FUSE – Incremental Implementation
- Ease of access and security
- Mostly implemented now a days, even on servers
- Advanced Security by User login and File encryption - AES
- EncFS - the simplest software for disk encryption on Linux.
- It is not safe if the adversary has the opportunity to see two or more snapshots of the ciphertext at different times.
- Protects files from malicious modification, but there are serious problems with this feature.

References

- http://fuse.sourceforge.net/
- https://wiki.archlinux.org/index.php/EncFS
- www.askubuntu.com
- http://stackoverflow.com/
- github.com
- http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Thank You