`CryptoFS` Security Note

footoredo

June 26, 2018

1 Basic structure

```
.cfs
  / keys
      / 793220291197.key
      / dfc7027894e1.key
        ......
  / structure.sec
  / contents
      / 1494eade37fa.key
      / 9be1f33a5b86.key
        ......
```

2 How to obtain the master key?

STEP 1 Retrive the unique identity of the device `$DEVID`.

STEP 2 Ask for user passpharse `$PASS`.

STEP 3 Compute `$KEY = hashsum($DEVID + $PASS)`.

STEP 4 Compute `$ID = hashsum($KEY + $PASS)`.

STEP 5 Find the key file `keys/$ID[:12].key` and decrypt it using `$KEY`.

3 What's in the decrypted key file?

PART 1 Symmetric key `$SIMKEY`

PART 2 Public-key encryption key-pair

KEY 1 Public key `$PUBKEY`

KEY 2 Private key `$PRIKEY`

4 `.sec` file

A `.sec` file is the ecrypted version of the original file combined with digital signature to check its integerity. It can be decrypt into a corresponding `.raw` file.

PART 1 Digital signature over hashsum of encrypted content (using `$PUBKEY` and `$PRIKEY`).

PART 2 Encrypted content (using `$SIMKEY|$SALT`).

Why use salt? If not, two file with same content will have same encrypted content.

5 `structure.sec`

This file stores the directory structure of all original files. It is intended for implementation of `ls` command and operation validity check. Furthermore, it also stores the `$SALT` for each file, which is needed in the section below, and the `stat` struct.

Why use salt? Without salt, one can easily verify if a certain file exists. He just need to check if there is a file named `‘hashsum("/foo/bar")[:12]‘.sec` under the folder `contents`.

6 Where to find a file?

STEP 1 Assume the dir for the file is `$DIR`. First of all check if it is valid in `structure.sec`.

STEP 2 If it is valid, we can retrive `$SALT` of this file. This file's identity can be computed in `$ID = hashsum($DIR + $SALT)`.

STEP 3 Find the corresponding `.sec` file `contents/$ID[:12].sec` and decrypt it.

7 Implementation with `fuse`

7.1 `mount`

1. At startup, first check if `.cfs` folder exists in the given mount point.
   - If found, query for the `$PASS` and load keys.
   - If not, query for the `$PASS` as well and generate new keys.
2. After keys are restored/generated, decrypt/create `structure.raw` file.

7.2 `umount`

1. Store `structure.raw` to `structure.sec`.
2. Store keys.

7.3 `open`

1. Check if the file exists.
2. Decrypt corresponding `.sec` file into `.raw` file.

7.4 `read/write`

Just operate on the decrypted `.raw` file.

7.5 `close`

Encrypt `.raw` file into `.sec`.