

Final Report

Wenjie Bai

1 submitted source files

purdec.c, purenc.c readme, finalreport.pdf makefile

2 work accomplished

file encryption/decryption in local mode; secure file transmission in distant mode.

3 code layout

./purenc

1. parse the arguments;
2. prompt password;
3. call `localmode()` or `distantmode()` depending on arguments.
4. if in local mode, initialize `gcrypt` handler `gcry_cipher_hd_t`, read input file contents, write encrypted contents into output file.
5. if in distant mode, generate salt value using in key derivation, initialization vector used in AES256 encryption, send filename, salt and initialization vector, mac key in phase 1; send encrypted contents in phase 2.

./purdec

1. parse the arguments;
2. prompt password;
3. call `localmode()` or `distantmode()` depending on arguments.
4. if in local mode, initialize `gcrypt` handler `gcry_cipher_hd_t`, read input file contents, write decrypted contents into output file.

5. if in distant mode, receive filename, salt value using in key derivation, initialization vector used in AES256 encryption, hmac key in phase 1; receive encrypted contents in phase 2 and write decrypted contents into output file.
6. receive hmac tag;
7. verify hmac tag.

4 program usage

To install libgcrypt, run `sudo apt-get install -y libgcrypt-dev`

local mode:

```
./purenc filename -l
./purdec -l filename
```

distant (network) mode:

```
./purdec -d port
./purenc filename -d ip:port
```

5 questions

Q: There will be a particular decision you'll need to make for dealing with PBKDF2. What extra input does it require and why is this used? How does your program deal with it?

PBKDF2 requires both password and salt as input, there is function called `generateRandom()` in `purenc.c` which is used to generate salt and initialization vector in distant mode. Note that in local mode, fixed salt is used since decryption side has no way of knowing the salt.

Q: Number of hours spent on the project : about 24.

6 results

6.1 local mode

Local mode encryption is shown in Fig. 1.

1. remove hello.txt.pur;
2. encrypt hello.txt;
3. copy hello.txt to check.txt;
4. remove hello.txt.

Local mode decryption is shown in Fig. 2

- [illegible]

6.2 distant mode

4

1. copy `hello.txt` to `check.txt`;
2. remove `hello.txt`.
3. receive encrypted file and decrypt it into `hello.txt`;
4. `diff -s hello.txt check.txt`. Those files are identical.

[illegible]

Figure 3: distant mode