

Procedures to build crypto libraries in Minix

In this document, we give step-by-step instructions on how to create a crypto library (*libcrypt.a*), and compile/link/build/run applications using the newly-built library in Minix.

Step 1: Get the files needed:

1. Download the *libcrypt.tar* file to your host machine from <http://www.cis.syr.edu/~wedu/seed/Labs/Files/libcrypt.tar>
2. Upload the *libcrypt.tar* file to your Minix machine, and put it in the directory of */usr/tmp*. You can use *ftp* to upload the *libcrypt.tar* file.
3. Login to your Minix machine, and do the following:

```
# cd /usr/tmp
# tar xvf libcrypt.tar
```

Now, in this directory (*/usr/tmp*), there should be two directories: *libcrypt*, and *demo*, and one file: *README*

In the default directory:

README: explanation of the contents of this package

In *libcrypt/* directory:

md5.h: header file for the md5 algorithm
md5.c: function implementation of the md5 algorithm
aes.h: header file for the aes algorithm
aes.c: function implementation of aes algorithm
sha256.h: header file for the sha256 algorithm
sha256.c: function implementation of sha256 algorithm
hmac_md5.c: function implementation of hmac_md5 algorithm
Makefile: the makefile used to build the library

In *demo/* directory:

hmc_md5_demo.c: the program to demonstrate the usage of hmac_md5
aes_demo.c: the program to demonstrate to use of aes algorithm

In the following steps, we assume our current directory is */usr/tmp*.

Step 2: Create the crypto library of our own:

We name the crypto library that we will create as **libcrypt.a**. Follow the procedures below:

1. Copy the header files to the */usr/include* directory, using the following command:

```
# cp libcrypt/*.h /usr/include
```
2. Create a sub-directory under */usr/src/lib* called *crypt*:

```
# mkdir /usr/src/lib/crypt
```
3. Copy the function implementation files, as well as the *Makefile*, to */usr/src/lib/crypt* directory:

```
# cp libcrypt/*.c /usr/src/lib/crypt
# cp libcrypt/Makefile /usr/src/lib/crypt
```

4. We need to modify the *Makefile* in the directory of */usr/src/lib*. Follow the instructions below to do the modification:

- a. `# cd /usr/src/lib`
- b. `# vi Makefile` // Or you can use “mined”, which is another editor in Minix
- c. In *all:* part, add
`cd crypt && $(MAKE)`
- d. In *install_i86* part, find the appropriate place, add
`$(LIB)/libcrypt.a \`
- e. Still in *install_i86* part, find the appropriate place, add
`$(LIB)/libcrypt.a: libcrypt.a`
`[TAB key]install -c -o bin $? $@`
- f. In *install_i386* part, find the appropriate place, add
`$(LIB386)/libcrypt.a \`
- g. Still in *install_i386* part, find the appropriate place, add
`$(LIB386)/libcrypt.a: libcrypt.a`
`[TAB key]install -c -o bin $? $@`
- h. Save the changes made to the *Makefile*, and exit

Note:

- `[TAB key]` is the white spaces created by pressing the “TAB” key on your keyboard. You need to follow the format exactly in creating / modifying the *Makefile*, otherwise it will not work.
 - The easiest way to do this is *copy & paste*, and then modifies the corresponding file names (if you are afraid of making mistakes).
 - More information can be found by checking the file */usr/src/lib/ansi/Makefile*
5. Build the library (*libcrypt.a*) using the following commands:
- ```
make all
make install
// After this step, you will find that libcrypt.a is in the directory /usr/src/lib
```
6. We need to modify the *descr* in the directory of */usr/lib*. Follow the instructions below to do the modification:
- a. `# cd /usr/lib`
  - b. `# vi descr` // Or you can use “mined”, which is another editor in Minix
  - c. Find the definition of *libs*, and at the end of the sentence (after `$A/$ARCH/libe.a`), add `$A/$ARCH/libcrypt.a`
  - d. Save and exit
7. Reboot the system:
- ```
# reboot
```

Step 3: Compile and link the demo programs

1. Compile the *aes_demo.c* and *hmc_md5_demo.c* programs

```
# cd /usr/tmp/demo  
# cc aes_demo.c -o aes_demo  
# cc hmc_md5_demo.c -o hmc_md5_demo
```

2. Run the *aes_demo* and *hmac_md5_demo* program:

```
# ./aes_demo  
# ./hmc_md5_demo
```