```
Key does not expire at all
Is this correct (y/n)?                y
A User-ID identifies your key; the software constructs the user id
from Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"
Real name:      Craig Hunt
Email address:        craig.hunt@wrotethebook.com
Comment:
You selected this USER-ID:
    "Craig Hunt <craig.hunt@wrotethebook.com>"
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?                   o
You need a Passphrase to protect your secret key.
Type the passphrase:          Fateful lightening
Repeat:     Fateful lightening
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
+++++.+++++.+++++.++++++++++++++++++++.+++++.++++++++++++++++++++++++++.++++++++++.
++++++++++++++++++++.++++++++++++++++++++++++++++++++++++++>.+++++.....................
.......+++++^^^
public and secret key created and signed.
```

The `--gen-key` option asks several questions. However, the questions are simple and the initial key generation needs to be done only once. First, gpg asks what kind of key you want. What it is really asking is whether you want to use the keys for digital signatures, for encryption, or for both digital signatures and encryption. Choose (1), which is the default. This creates both types of keys so that you're prepared for any encryption task. Next it asks how long the keys should be; the longer the key, the more difficult it is to generate and crack. The default is 1024 bits, which is plenty long for any realistic gpg application. gpg asks for your name, email address, and, optionally, a comment. It uses this information to identify your keys in the key databases. Finally, it asks for a passphrase that will be used to identify you when you access your secret key.

gpg uses two key databases: one for secret keys and one for public keys. gpg calls these databases "keyrings." The database of secret keys is *secring.gpg* and the database of public keys is *pubring.gpg*. Both public and private keys are used when we encrypt and then decrypt a file. The following example shows the encryption process:

```
$ cat test.txt
This is a test file.
$ gpg --recipient craig.hunt@wrotethebook.com --encrypt test.txt
gpg: Warning: using insecure memory!
$ cat test.txt.gpg
´¥   → º ü ¿  2J  ë ¥;Î¬[  Ø# LÍü" é  ´ÉDÓ Sì P-EÜ ® ¸¿Õ!7 ñ6 ÍÓèî
¢È ó$2[  9øÂÌï@E¬wY $2´6 $B« ´6ÿk_¬ø1ÑOÔBî ¤íy¿[CyöU6®&V¯¤ TWn2        ¡Ó ˚ßx
 ñÒñnT5ª ¥[uü¥ÊÀ2 ,hæq"?ºï´J\Põ ö/o?¨ÒTeBáâÛÙ»˚5oNB= å}/Ô@Némstû
$ rm test.txt
```