Pretty Good Privacy (PGP)
Introduction to Key Management

GnuPG is a free implementation of PGP. This mini-workshop uses GnuPG on FreeBSD. We do not cover encryption or data-signing here, just the business of creating key pairs, sharing public keys, verifying fingerprints and key signing.

Check the GnuPG web page for documentation on the GnuPG package:

  http://www.gnupg.org/

In particular, look at the documentation, and the "Mini HOWTO" which is good background material for the topics in this workshop.

** Public Key Cryptography **

1. Public key crypto uses two related keys -- a secret (or "private") key, which is never shared and which should always be kept in a secure place, and a public key. The public key can be shared with anybody.

2. To encrypt some data so it can only be read by one person, you need that person's public key.

3. To decrypt some data that someone sent you, you need your secret key.

4. To sign some data, you use your secret key.

5. To check a signature on some data, you use the public key of the person who used it.

** Two Precautions Worth Taking **

1. Before you use someone's public key, make sure you trust it. That is, be sure that the public key was not modified in between the owner and you. You can increase your trust by comparing the fingerprint of the key you have with that calculated by the key's owner. You can also gain some measure of trust by checking signatures that might be present on the key.

2. When you are talking to the owner of a public key, either directly in person or via telephone, think about how much trust you have in the identity of that person. Use measures like the reputation of the person amongst other people you trust, matching photo i.d. (e.g. passports) and the person's knowledge of shared experiences in the past to gain a level of trust you are comfortable with.

** Installing GnuPG **

On FreeBSD, GnuPG is included in the ports tree as security/gnupg. You can install it in the usual way:

```
  # cd /usr/ports/security
  # make install && make clean
```

** Creating a Public/Private Key Pair **

```
  $ gpg --gen-key
```

The default values for the type and size of key are usually OK (but feel free to choose larger key sizes if you like -- larger keys require more computer power to work with, but are harder for other people to compromise).

Always set an expiry date, even if it is a long way in the future. Choosing 1 year is reasonable (type "1y").

Type your real name, and your e-mail address when asked to do so. You don't have to type a comment. You can edit any of these before you create the key.

Type a passphrase. It's important to choose something that you can remember but which will be hard for anybody else to guess, just like any password. You will be asked to type it twice.

The GPG software will create a Public/Private key pair using random information it obtains from the system. If it seems to be taking a long time, try to keep the machine busy by using the network or the keyboard, since both those things are used to add randomness to the key generation process.

** Extracting your Public Key **

To extract your public key as text which you can easily cut and paste, or include in an e-mail message, do

```
  $ gpg -a --export <your key id>
```

The key id is a hexadecimal number that will have been displayed after you generated your key. If you know you only have one key which matches your e-mail address, you can use your e-mail address instead of the key id, and everything should work.

To see what public keys you have installed, you can always type

```
  $ gpg --list-keys
```

Once you have extracted your public key, you can send it to other people. Good ways of doing this are e-mail, putting it on a web page, or sending it to a key server.

** Generating your Public Key's Fingerprint

Comparing public keys is difficult, because the keys themselves are usually quite long. A much easier method (and, in practical terms, pretty much as good) is for each person to generate the fingerprint of their copy of a key, and then compare the fingerprints. Fingerprints are short enough to be easily read out over the phone.

You calculate the fingerprint for a local copy of a public key like this:

```
  $ gpg --fingerprint <key id>
```

** Importing Someone Else's Public Key

Once you have obtained a public key, you can import it to your local keyring so that you can use it like this:

```
  $ gpg --import <filename>
```

** Signing a Public Key

If you have a copy of someone else's public key on your keyring and you have decided that you trust it (e.g. by verifying the fingerprint with the key's owner) and you have also decided that you trust the identity of the key's owner (e.g. by checking a passport) you can sign it. This does two things:

1. It helps you remember in the future that you have checked the key, and it is to be trusted.

2. If other people receive a copy of the key with your signature, and they trust you, then they can use your signature to help them decide whether they trust the key. This helps build what is known as "the web of trust".

To sign a key:

```
  $ gpg --sign-key <key id>
```

** More Information

There are many more things you can do with GnuPG than those described in these notes. For more information, see:

  http://www.gnupg.org/

Of course, you can also ask your AfNOG friends on the AfNOG list, or send e-mail to the instructors directly if the GnuPG documents are not clear.

Joe Abley <jabley@ca.afilias.info>
AfNOG, Nairobi, Kenya, 2006