Pretty Good Privacy version 1.0 - Installation Guide

How to Install PGP

First, PGP only runs on MSDOS machines at this time; if you have something else, you will need to be patient (or get the source and help port the program to new environments; see the Copyleft notice).

PGP is distributed in a compressed archive format, which keeps all the relevent files grouped together, and also saves disk space and transmission time. The current version, 1.0, is archived with the PKZIP utility, and the PGP executable binary release system is in a file named PGP10.ZIP. This contains the executable program, the user documentation, and a few keys and signatures. There is also a second file available containing the C and assembly source code, called PGP10SRC.ZIP; unless you are a programmer interested in cryptography, it is probably of little interest to you. It may or may not be available from the source from which you get PGP10.ZIP; if not, and you want it, see the Licensing and Distribution section of the PGP User's Guide.

You will need PKUNZIP version 1.1 or later to uncompress and split the PGP10.ZIP archive file into individual files. PKUNZIP is shareware and is widely available on MSDOS machines (the only machines on which PGP runs now anyway).

Create a directory for the PGP files. For this description, let's use the directory C:\PGP as an example, but you should substitute your own disk and directory name if you use something different. Type these commands to make the new directory:

```
   c:
   md \pgp
   cd \pgp
```

Uncompress the distribution file PGP10.ZIP to the directory. For this example, we will assume the file is on floppy drive A - if not, substitute your own file location.

```
   pkunzip a:pgp10
```

The next step is recommended but not required. It can help protect you from a later virus infection or other destruction of the PGP program. Put a blank floppy on drive A (in this example) and copy everything you just uncompressed onto it - then WRITE PROTECT the floppy disk. You can use that copy later (even without PKUNZIP handy) to check to see if your copy of PGP is still trustworthy.

```
   copy *.* a: /v                    (with a blank floppy in A:)
```

Now write protect the floppy.

The next step is to validate the PGP program, using PGP itself. This will detect virus infections or any other tampering of the PGP software. For a full explanation of why this is important, read the PGP User's Guide, with particular attention to the sections on Vulnerablities, Viruses and Trojan Horses, Key Management, and Signed Key Server Certificates.

If this self-validation test fails, you definitely have a problem - either the program PGP.EXE or the PGP.CTX file have been damaged (perhaps tampered with, perhaps virus infected, or perhaps just corrupted in transit), or else the program is unmodified but does not work on your hardware. We know of no MSDOS machines on which it doesn't work, but it is always possible. If it does check out OK, you have just run PGP successfully! Unfortunately, this does not in itself prove that the program is unmodified. If someone deliberately tampered with the program before you got it, they could have substituted a bogus key for Philip Zimmermann and their own fake signature file PGP.CTX as well, or modified the program to give false assurances. Your only real safeguard is to get your first copy of PGP from a reliable source. See the list of sources in the PGP User's Guide.

But for now, let's assume that you have a good copy of PGP for the following self test:

```
   pgp pgp.ctx pgp.exe
```

PGP should report a good signature from Philip R. Zimmermann on the PGP.EXE executable program file, which indicates your copy of PGP software has no virus infection and has not been tampered with.

Any time that you want to reassure yourself that PGP has not been corrupted by virus or by any other means, you can repeat this test using the write protected floppy disk version (don't remove the write protect!), thusly:

```
   SET PGPPATH=A:\
   a:pgp a:pgp.ctx pgp.exe
```

Setting the Environment

Next, you can set an MSDOS "environment variable" to let PGP know where to find its special files, in case you use it from other than the default PGP directory. Use your favorite text editor to add the following lines to your AUTOEXEC.BAT file (usually on your C: drive):

```
   SET PGPPATH=C:\PGP
   SET PATH=C:\PGP;%PATH%
```

Substitute your own directory name if different from "C:\PGP".

Generating Your First Key

One of the first things you will want to do to really use PGP (other than to test itself) is to generate your own key. This is described in more detail in the "RSA Key Generation" section of PGP User's Guide. Remember that your key becomes something like your written signature or your bank card code number or even a house key - keep it secret and keep it secure! Use a good, unguessable pass phrase and remember it. Right after you generate a key, put it on your key rings and copy your secret keyring (KEYRING.SEC) to a blank floppy and write protect the floppy.

Validating Future Releases of PGP

Now we get to an important new feature of this technology. Once you have a good copy of this first release, you can use it to check the validity of future releases! We will use the same key to sign future releases. If you acquire something which claims to be PGP 1.1 (or 2.3 or whatever), be sure to check it using the write protected PGP.EXE and KEYRING.PUB on the write protected floppy. Say you just got PGP11.ZIP and uncompressed it to directory C:\PGPNEW. The directory should now contain a new PGP.EXE and a PGP.CTX, among other files. Put the write protected floppy in drive A and enter:

```
   SET PGPPATH=A:\
   a:pgp pgp.ctx pgp.exe
```

If the write protected copy of PGP from the floppy, using the KEYRING.PUB from the same floppy, validates the new version of PGP, you can be pretty sure that it really is from me and has not been tampered with. (There could still be a problem if a virus which was specifically targeted at PGP was already resident in your computer memory when you did the test; you might want to reboot first and use any anti-virus programs you trust first).