Bionic Buffalo Tech Note #82:

Hints for Using PGP

last revised Wednesday 28 October 1998

OVERVIEW

This document gives some miscellaneous hints for using PGP. It is not a substitute for other PGP documentation. It is a quick introduction for anyone who wants it. We hope this will make it easier to use PGP.

There are two main uses of PGP:

- *authentication* - PGP will verify the authorship of digitally-signed documents or files
- *encryption* - PGP will scramble data so that only the intended recipients can unscramble it

A PGP user first creates a key pair. The key pair has two parts: a public key and a private key. The public key should be distributed widely to anyone who wants to communicate with the PGP user. The private key is kept a secret, and remains known only to the PGP user who created the key pair.

PGP uses a pass-phrase to keep the private key secret. The pass-phrase is not the same as the private key itself. The pass-phrase is used to encrypt (scramble) the private key so the private key cannot be read from the user's disk.

This is how authentication (document signing) works:

1. The author of the document uses his private key to create a signature. The signature can be given out with the document.
2. A reader can use the matching public key to verify it was signed using the private key.

To sign a document, you need your own private key. To verify a signature, you need the public key of the signer.

This is how encryption works:

1. The author of the document encrypts (scrambles) the document using the public keys of anyone who should be able to read the document.
2. Each reader must use his matching private key to decrypt (unscramble) the document.

To encrypt a document, you need the public key of the recipient. To decrypt a document which was encrypted for you, you need your own private key.

PGP can also sign and encrypt documents at the same time.

RESOURCES

On its web site *www.tatanka.com*, Bionic Buffalo maintains several links to other web sites with PGP information. Also, any major portal or search engine can be used to find other PGP web sites. Please use our links page, or use the search engines and portals, if you need more information.

CONTENT TYPES

Internet e-mail and web pages define a *content-type* for different kinds of files. By knowing the content type of a file, a program can know how to interpret the file. The web server tells the browser the content types of files used by the browser.

Most servers determine the content type based on the ending of the file name. For example, files which end in *.htm* or *.html* are hypertext markup language files, and files which end in *.pdf* are portable document format files. There are standard names for these types: hypertext files are called *"text/html"*, and portable document format files are called *"application/pdf"*.

When browsers read the files from disk, they also use file name endings to determine the content types. However, when browsers connect to a web server, they receive the type string (such as *"text/html"* or *"application/pdf"*) directly from the server, and do not look at the file name.

There are three content types defined for PGP files:

- *"application/pgp-encrypted"* files usually have names which end in *.pgp* or *.asc*
- *"application/pgp-keys"* files usually have names which end in *.skr* or *.pkr*
- *"application/pgp-signature"* files usually have names which end in *.sig*

When the web server and browser are configured properly, and when a user clicks on a PGP file on a web page, then the browser will use the PGP software to interpret the file. However, this usually doesn't happen for two reasons:

1. Most servers do not recognize the three PGP content types listed above. When a server does not recogize a content type, it tells the browser the content type is *"application/octet-stream"*, and the browser asks the user what to do or tries to save the file to disk. To fix this, **the server must be configured to recognize PGP content types**.

2. When PGP is installed, the installation program usually registers the file name endings such as *.pgp* or *.sig*. However, the installation usually does not set the content types for PGP files. To fix this, **the browser must be configured to recognize the content types of PGP files**.

If your server or browser are not set up correctly, consult their documentation for configuration instructions. Usually, searching the documentation for the phrases *"MIME type"* or *"content type"* will find the instructions.

PGP-ENCODED E-MAIL

There are three ways to encode encrypted information in internet mail:

1. Include the information in the body of the mail.

2. Use PGP MIME encoding.

3. Use S/MIME encoding.

In the first method, the mail software does not know or care what is in the message. A PGP message sent this way begins with text which looks approximately like the following:

```
-----BEGIN PGP MESSAGE-----
Version: PGP for Personal Privacy 5.5.3
qANQR1DBwk4D9D9lKbCpivMQC/wNPAsoiGqXz975xm7eeFre7jJbGOxSn+K4a+H3
Hm1WiPkHya7bWAVxDoT5LBKuM3HN2SB0+s34ygyZNS+XU0/X3CGSkXILvn1Z3mkT
xsrk9oXWU6lkglIa0oerUsGkUkAOuxQG3ee+IyG2bpxelFyicFT3XoRPkNyyEpi9
```

and ends like this:

```
GtAjjeAvKeUzM1LsZ8orHAMPUwTB3slUgFpWdvPcxhI88ytB5tPJcGMuiZMrs3Qq
/roVR+eP0uwQ7qA1fChD0zgpmkryzR4bsYOz1Ddof4FPxO+YDD0NVV6N67mk1ca4
E7N35Ov3m4s10ZRO
=2KRZ
-----END PGP MESSAGE-----
```

Some mailers (such as Qualcomm's Eudora) have a button to extract such messages from the body of the mail, and send them to the PGP program for decryption. With other mailers, you can use select, copy, and paste to get the message into PGP.

The original standard for encoding PGP messages into mail was PGP MIME. Qualcomm's Eudora and Microsoft's Outlook will recognize a PGP MIME message, and know to send it to the PGP program for interpretation. Other mailers (including Netscape) don't recognize the PGP MIME format, and show the message as two parts: an "encrypted" part and a part with content type *"application/octet-stream"*. The second part is the message itself. The best way we know how to deal with this is to save the second part as a file with extension *.asc*, and then open the file separately using PGP.

The newest standard for encoding secure information is S/MIME. Unfortunately, the current versions of PGP software don't support S/MIME applications. We recommend avoiding S/MIME when using PGP. Otherwise, most other software will not be able to read your messages.

RSA KEYS

Older versions of PGP used RSA keys, while the new versions prefer DH/DSS keys. These two kinds of keys are incompatible. In summary,

- old versions of PGP can use RSA keys, and can generate RSA key pairs
- new versions of PGP can use RSA or DH/DSS keys, and can generate DH/DSS key pairs, but cannot generate RSA key pairs

(However, PGP sells a newer, "business" version of PGP which can generate both kinds of key pairs.)

Especially because some users outside the United States rely on RSA keys, it is sometimes important to be able to create an RSA key pair. If you cannot buy the business version in your country (due to export restrictions or whatever), you can create an RSA key pair using an older version (such as version 2.6), then import the RSA key pair into the newer PGP keyring.

There are many versions of PGP, most of them "unofficial", but many consider the unofficial versions to be better than the commercial versions for various reasons. Whether or not this is true, remember that additional features such as gigantic RSA keys cannot be understood by all other versions of the software. If you generate jumbo-size keys, you may limit your ability to communicate with others. It may be wise to have another key pair, which is smaller, for use with those who do not support the larger keys.

KEY LENGTH

Key sizes are usually expressed in bits. For instance, you might have a 1024-bit RSA key, or a 56-bit DES key.

Not all forms of encryption are equally difficult to break, nor do they have the same weaknesses. In particular, an RSA or DH/DSS key of a given size is may be weaker than a different kind of key of a much smaller size. Therefore, comparison of key sizes among different encryption methods is not usually meaningful.

A PGP message actually uses two different encryption methods and two different keys of different sizes. The content of the message usually is encrypted using a randomly-generated 128-bit key and the IDEA encryption algorithm. Then the 128-bit key itself is encrypted for each recipient using that recipient's public key. The actual strength of the content's encryption is therefore based on a 128-bit IDEA key, and not on the public/private key pair.