Email encryption with Pine and GPG

Pine is a nice but powerful console mail client, and just like GPG, it is available on many Linux and Unix based computers. It is not hard to make Pine read and send GPG-encrypted or -signed messages. Several tools are available that glue gpg and pine together, and I use Pine Privacy Guard (PPG). This is a small Perl script that does all the hard interfacing work.

Installation and configuration

- Install *Pine Privacy Guard* on your computer:

```
wget http://quantumlab.net/pine_privacy_guard/pinepg-1.02.tgz
tar -C /usr/local -zxvf pinepg-1.02.tgz
cd /usr/local/
chown -R root:root /usr/local/pinepg-1.02
ln -s pinepg-1.02 pinepg
chmod 755 /usr/local/pinepg/pine_privacy_guard.pl
```

  If you install PPG in another place than `/usr/local/pinepg`, make certain that you adapt the pathnames that are used further down in the Pine configuration.

- Uncomment the line `default-recipient-self` in your `~/.gnupg/gpg.conf` file to encrypt to self so you can are able read your own encrypted *sent* messages later on:

```
#default-key 0xYourKeyID
default-recipient-self
```

- Configure Pine's display-filters and sending-filters
  either edit your `~/.pinerc` directly and make sure that the relevant bits look like this:

```
# This variable takes a list of programs that message text is piped
into
# after MIME decoding, prior to display.
display-filters=_LEADING("-----BEGIN PGP MESSAGE-----")_
/usr/local/pinepg/decrypt _RESULTFILE_ _DATAFILE_ _PREPENDKEY_,
        _LEADING("-----BEGIN PGP SIGNED MESSAGE-----")_
/usr/local/pinepg/verify _TMPFILE_ _RESULTFILE_

# This defines a program that message text is piped into before MIME
# encoding, prior to sending
sending-filters=/usr/local/pinepg/clearsign _RESULTFILE_ _DATAFILE_
_PREPENDKEY_,
        /usr/local/pinepg/encrypt _RECIPIENTS_ _RESULTFILE_ _DATAFILE_
_PREPENDKEY_
```

  or configure this from within Pine: (**S**etup > **C**onfig > **W**hereis > `display-filters` and add the two values shown above. Same for `sending-filters`)

- (Optionally - looks cool) add these custom X-Headers to each email you send with pine:

```
# Add these customized headers (and possible default values) when
composing
customized-hdrs=X-GPG-PUBLIC-KEY:
http://pgp.mit.edu:11371/pks/lookup?op=get&search=0xA75CBDA0,
        X-GPG-FINGERPRINT: F2CE 1B92 EE1F 2C0C E97E 581E 5E56 AAAF A75C
BDA0
```

  You can edit `~/.pinerc` directly (look for the `customized-hdrs=` line) or configure from within pine (**S**etup > **C**onfig > **W**hereis > `customized-hdrs`)
  Of course, the example GPG fingerprint and key-URL are mine, and you should substitute your own.

Working with GPG in Pine

Sending email

You use Pine as usual, so composing a new email is no different than before. The fun starts when you press <CTRL>-X to send it. Upon pressing <CTRL>-X instead of sending the message, Pine will ask which filter you want to use. You can choose 1 of 3 options (and cycle through them using <CTRL>-P and <CTRL>-N):

1. Unfiltered; send email without using any cryptography.
2. Encrypt the email (PinePG will also sign it by default).
3. Clear-sign the email.

After entering your choice, the email is sent.

During your Pine session, the first time you have to use GPG to read or send a mail message, you will be prompted for your GPG passphrase. After that PinePG will securely remember your passphrase for the remainder of that session, so that you won't have to enter it again.

Receiving email

If you open any email that contains a GPG signed or encrypted message, the pinepg filter is automatically invoked. You are prompted for your GPG passphrase, so that the message can be decrypted. The caracters that you type, will not be displayed on the screen.
The output of GnuPG is displayed. It will show whether or not the encrypted text was successfully decrypted, if the GPG signature is valid if it was signed, and any other relevant information:

```
gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information
gpg: Signature made Fri Mar 17 03:19:07 2006 PST using DSA key ID A75CBDA0
gpg: Good signature from "Eric Hameleers <alien slackware.com>"
gpg:                 aka "Eric Hameleers <alien sox.homeip.net>"
```

Press E to continue.
The decrypted message is displayed as a normal message.

- *Note:* The plaintext is not saved to the mailbox.
- *Note:* The above warning about *insecure memory* is caused by the fact that I am not running GPG as root, and the `/usr/bin/gpg` binary is not setuid root. If you're bothered by the warning, implement either one of these solutions:
  - run

    ```
    chmod +s /usr/bin/gpg
    ```

    to fix the use of insecure memory. or

  - if you can't or don't want to install gpg setuid(root), then you can add the commandline parameter

    ```
    --no-secmem-warning
    ```

    to the gpg command, or put the line

    ```
    no-secmem-warning
    ```

    in your configuration file `~/.gnupg/options` or `~/.gnupg/gpg.conf`. This will disable the warning message.

PGP/Mime

Pine Privacy Guard does not handle PGP/Mime encrypted emails. I found a couple of links with possible solutions, but have not yet looked too deeply into these.

- Reading PGP/Mime messages with Pine
- pgp-mime-handler