

This documentation is for gpgmailencrypt 2.x

1 What is gpgmailencrypt

gpgmailencrypt can encrypt e-mails.

It supports

- * PGP/Inline

- * PGP/Mime

- * SMime

It can be used normally as a script doing everything on command line or in daemon mode, where gpgmailencrypt acts as an encrypting smtp server.

It takes emails and if a encryption key exists for this user it will return the e-mail encrypted to another e-mail server.

The encryption method can be selected per user.

1.1 Prerequisites

The following software needs to be installed

- python3.x
- gnupg (I recommend version 2)
- openssl

2 Installation

2.1 General

1. Copy the file „gpgmailencrypt.py“ into the directory /usr/local/bin and make the file executable via

```
sudo chmod 755 /usr/local/bin/gpgmailencrypt.py
```

2. Create a default configuration file via

```
gpgmailencrypt.py -x > ~/gpgmailencrypt.conf
```

3. Copy the configuration file into the directory /etc

```
sudo cp ~/gpgmailencrypt.conf /etc
```

2.2 Daemon

1. Copy the file gpgmailencrypt.init into the directory /etc/init.d via

```
sudo cp gpgmailencrypt.init /etc/init.d/gpgmailencrypt
```

2. Create a user (gpg-mailencrypt) under which the daemon should run

```
sudo adduser gpg-mailencrypt
```

3. You can set the user in the file /etc/default/gpgmailencrypt. It should contain

```
USER="gpg-mailencrypt"
DIR="/usr/local/bin"
```

3 Configuration /etc/gpgmailencrypt.conf

3.1 General

```
[default]
prefered_encryption = pgpinline           # valid values are
'pgpinline','pgpmime' or 'smime'
add_header = no                          # adds a X-GPGMailencrypt header to
the mail
domains =                               # comma separated list of domain
names, that should be encrypted, empty is all
spamsubject =***SPAM                    # Spam recognition string, spam will
not be encrypted
output=mail                             # valid values are 'mail'or 'stdout'
locale=en                                # DE|EN|ES|FR|IT|NL|PL|PT|RU|SE'

[gpg]
keyhome = /var/lib/gpgmailencrypt/.gnupg      # home directory of public
pgpkeyring
gpgcommand = /usr/bin/gpg2
allowgpgcomment = yes                      # allow a comment string in the GPG
file

[logging]
log=none                                  # valid values are 'none', 'syslog', 'file'
or 'stderr'
file = /tmp/gpgmailencrypt.log
debug = no

[mailserver]
host = 127.0.0.1                          #smtp host
port = 25                                  #smtp port
authenticate = False                       #user must authenticate
smtpcredential =/etc/gpgmailencrypt.cfg      #file that keeps user and
password information
                                           #file format 'user=password'

[encryptionmap]
user@domain.com = PGPMIME                 #PGPMIME | PGPINLINE | SMIME | NONE
```

gpgmailencrypt documentation

```
[usermap]
#user_nokey@domain.com = user_key@otherdomain.com

[smime]
keyhome = ~/.smime                #home directory of S/MIME public key
files
opensslcommand = /usr/bin/openssl
defaultcipher = DES3              #DES3|AES128|AES192|AES256
extractkey= no                    #automatically scan emails and extract
smime public keys to 'keyextractdir'
keyextractdir=~/.smime/extract

[smimeuser]
smime.user@domain.com = user.pem[,cipher] #public S/MIME key file [,used cipher,
see defaultcipher]

[daemon]
host = 127.0.0.1                  #smtp host
port = 10025                      #smtp port
smtps = False                    #use smtps encryption
sslkeyfile = /etc/gpgsmtp.key     #the x509 certificate key file
sslcertfile = /etc/gpgsmtp.crt    #the x509 certificate cert file
authenticate = False             #users must authenticate
smtppasswords = /etc/gpgmailencrypt.pw #file that includes users and
passwords                        #file format 'user=password'
statistics=1                     #how often per day should statistical
data be logged (0=none) max is 24
```

3.2 PGP specific configuration

```
[gpg]
keyhome = /var/lib/gpgmailencrypt/.gnupg # home directory of public
gpgkeyring
gpgcommand = /usr/bin/gpg2
allowgpgcomment = yes             # allow a comment string in the GPG
file
```

3.3 SMIME specific configuration

```
[smime]
keyhome = ~/.smime                #home directory of S/MIME public key
files
opensslcommand = /usr/bin/openssl
defaultcipher = DES3              #DES3|AES128|AES192|AES256
extractkey= no                    #automatically scan emails and extract
smime public keys to 'keyextractdir'
keyextractdir=~/.smime/extract

[smimeuser]
smime.user@domain.com = user.pem[,cipher] #public S/MIME key file [,used cipher,
see defaultcipher]
```

3.4 Daemon specific configuration

```
[daemon]
host = 127.0.0.1           #smtp host
port = 10025               #smtp port
smtps = False             #use smtps encryption
sslkeyfile = '/etc/gpgsmtp.key' #the x509 certificate key file
sslcertfile = '/etc/gpgsmtp.crt' #the x509 certificate cert file
authenticate = False      #users must authenticate
smtp passwords = '/etc/gpgmailencrypt.pw' ;use smtps encryption
```

The gpgmailencrypt.pw has the following format:

```
user1=password1
user2=password2
```

Don't forget to make the file readable only for the gpgmailencrypt user!

The x509 certificate files can be created see:

```
https://www.e-rave.nl/create-a-self-signed-ssl-key-for-postfi
```

For the authenticate option see chapter 6.3.

4 Key Management

The following commands have to be used as the user, that is running gpgmailencrypt. Remember that in daemon mode this user is 'gpg-mailencrypt'. So for daemon mode you first have to change the user

```
sudo su - gpg-mailencrypt
```

4.1 PGP

Add a PGP key to the public key ring

```
gpg --import publickey.gpg
```

4.2 SMIME

Smime keys are stored in the directory ~/.smime per default. You have to create it if it does not exist. Each key is stored in a single file in pem-format.

Usually you get the smime.key file in a different format. To convert it use

```
openssl pkcs7 -print_certs -inform DER -in smime.p7s -out smime.pem
```

Let's say you get the smime.p7s from agentj@mib.

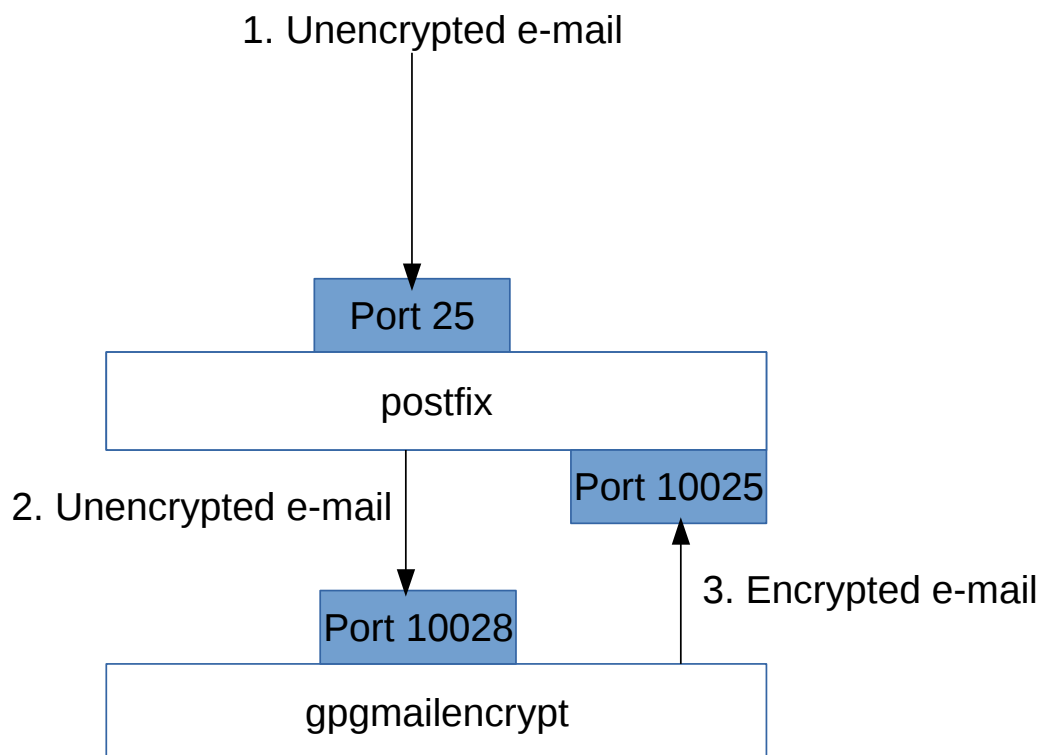
Instead of 'smime.pem" you should use a unique name for the file and copy it in ~/.smime/

```
cp smime.pem ~/.smime/agentj@mib.pem
```

For this user you need also an entry in /etc/gpgmailencrypt.conf

```
[smimeuser]  
agentj@mib = agentj@mib.pem
```

5 Integrating gpgmailencrypt in postfix



Install and configure gpgmailencrypt as daemon.

/etc/gpgmailencrypt.conf

```
[mailserver]  
host = 127.0.0.1  
port = 10025  
[daemon]
```

gpgmailencrypt documentation

```
host = 127.0.0.1
port = 10028
```

/etc/postfix/main.cf

```
content_filter=gpgmailencrypt:[127.0.0.1]:10028
```

/etc/postfix/master.cf

```
localhost:10025 inet n - n - - smtpd
    -o content_filter=
    -o mynetworks=127.0.0.0/8
    -o receive_override_options=no_unknown_recipient_checks
    -o smtpd_recipient_restrictions=permit_mynetworks,reject_unauth_destination
    -o smtpd_authorized_xforward_hosts=127.0.0.0/8

gpgmailencrypt unix - - n - 2 smtp
    -o smtp_data_done_timeout=1800
```

5.1 Using authentication

For using the authentication add the following to gpgmailencrypt section in master.cf

```
-o smtp_sasl_auth_enable=yes
-o smtp_sasl_password_maps=hash:/etc/postfix/gpgmailencrypt_passwd
```

With /etc/postfix/gpgmailencrypt_passwd having the following structure

```
localhost user:password
```

Then use the following commands

```
sudo chmod 640 /etc/postfix/gpgmailencrypt_passwd
sudo postmap /etc/postfix/gpgmailencrypt_passwd
```

5.2 Using smtps

To use the gpgmailencrypt smtps feature with postfix 2.x you need to install stunnel (in Ubuntu the package is called stunnel4)

Create the file /etc/stunnel/gpgmailencrypt.conf

```
[gpgmailencrypt-smtps]
accept = 10000
```

gpgmailencrypt documentation

```
client = yes  
connect = localhost:10028
```

And change `/etc/default/stunnel4`

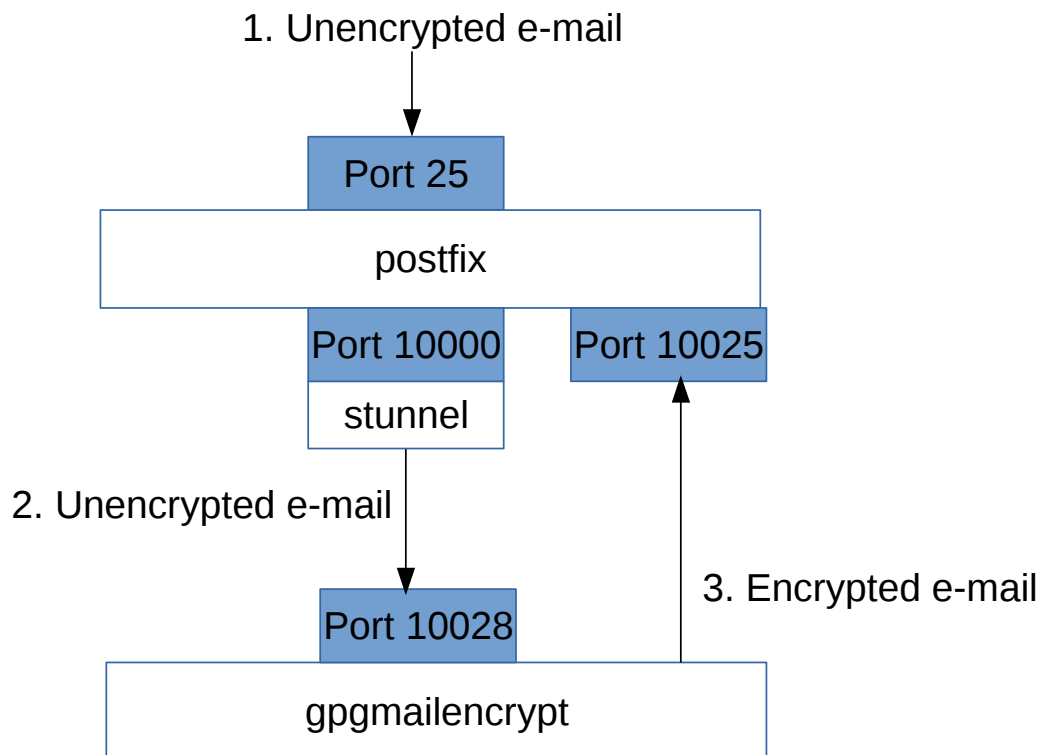
```
ENABLED=1
```

Then start stunnel with

```
/etc/init.d/stunnel4 start
```

`/etc/postfix/main.cf` should be changed to

```
content_filter=gpgmailencrypt:[127.0.0.1]:10000
```



6 The admin console

6.1 The admin user – the hen and egg problem

First you have to create a normal user (which only the admin user can do), then you can give him administration rights.

So for the very first time you have to start the following python program as the user, that runs the gpgmailencrypt daemon. Replace “admin1” and “secret” with the values you like

```
#!/usr/bin/python3
import gpgmailencrypt
user="admin1"
password="secret"
with gpgmailencrypt.gme() as g:
    g.adm_set_user(user,password)
```

Now the user is created.

Now change the /etc/gpgmailencrypt.conf to the following value

```
[daemon]
admins=admin1                #comma separated list of admins, that can
use the admin console
```

Now restart the daemon, that's it.

6.2 Using the admin console

Start

```
gme_admin.py localhost 10025
```

Replace 'localhost' and the port number with the values of your daemon

Now login as an admin user. You will see the following screen

```
Try to connect to localhost:10025 ...
gpgmailencrypt admin console
=====
User: admin1
Password:
OK
Authentication successful.
Welcome. Enter 'HELP' for a list of commands
> _
```

The following commands are available

flush	tries to re-send deferred emails
-------	----------------------------------

gpgmailencrypt documentation

debug true/false	sets the debug mode
deluser	deletes a user example: 'deluser john'
help	Shows all available commands
quit	leave the console
reload	reloads the configuration file
resetstatistics	sets all statistic values to 0
setuser	adds a new user or changes the password for an existing user example: 'setuser john johnspassword'
statistics	print statistic information
users	print users

6.3 Non-admin users

Users, that don't have the admin privilege can be used to login to the server, when the daemon need authentication, set with

/etc/gpgmailencrypt.conf

```
[daemon]
authenticate = True                #users must authenticate
```