MyPGP
Graphical User Interface for PGP

6.2.2020

1 Prerequisites

MyPGP is based entirely on BouncyCastle for all cryptographic functions: it is merely a user interface. However, see “Elgamal encryption” below.

MyPGP is distributed with last version of BouncyCastle.

2 Files and directories

2.1 HOME directory

When MyPGP starts, it requests to identify a directory in your file system. This is the HOME directory for this execution.

2.2 Key directories

MyPGP uses file folders to organise keys, either public or private.

All keys in HOME are loaded and shown under the branch HOME in the console.

If HOME has sub-folders, these sub-folders are loaded as well, and will show as sub-branches of MYPGP in the console. Sub-folders may nest to any depth.

Adding keys is as simple as placing a copy in the desired folder. Removing keys is as simple as removing from the folder. Keys may repeat in different folders.

The files in the folders above may be simple keys, or key rings such as

- pubring.pkr
- secring.skr

MyPGP skips files and folders starting with an underscore character

_

or ending with any of the following extensions:

.skip

.mypgp

.jar

MyPGP skips the folder and its sub-folders.

MyPGP has no concept of a single key directory. You may have different sets of keys under different directories. You choose the desired set on start.

This allows different directories for different, unrelated, projects.

2.3 database.mypgp

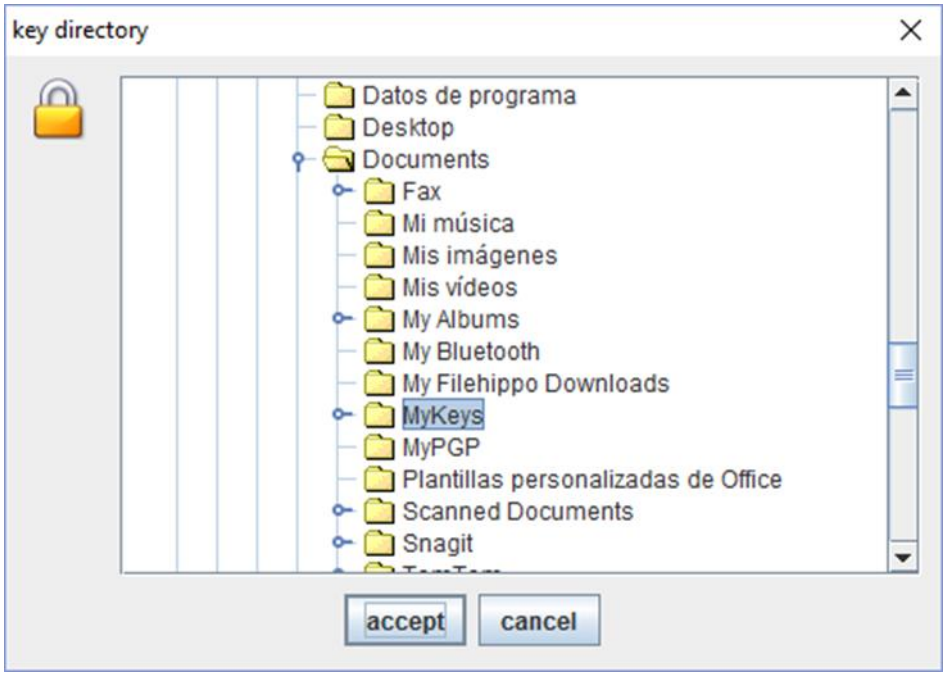
MyPGP uses a file named “database.mypgp” under HOME to store information about

- key aliases
- key lists

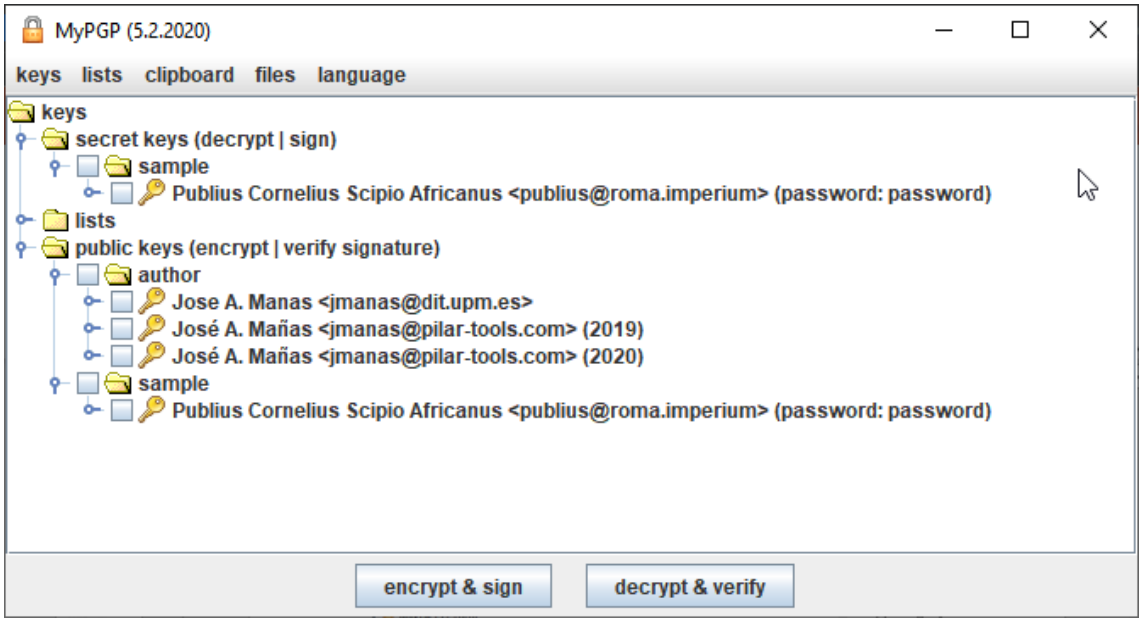
The file is plain text, and you may edit it; but it usually written by MyPGP using commands from the user interface.

3 Start up

Select keys directory:



4 Main panel



4.1 Usage example: encrypt a file

1. Select one or more recipient keys
2. Browse and select the file to encrypt (one or more)
   files >> encrypt

Alternatively, you may drag and drop a file from a file explorer to the ENCRYPT & SIGN button.

The console reports on the success or failure of the operation

4.2 Usage example: sign a file

1. Select one signing key
2. Browse and select one or more files to sign
   files >> sign

Alternatively, you drag and drop a file from a file explorer onto the ENCRYPT & SIGN button.

Console logs the operation

4.3 Usage example: encrypt and sign a message

As in the previous examples, but choosing simultaneously secret keys for signature and public keys for encryption

5 Menu keys

new key

To create a new key. Please, fill the screen with the required information.

A signing key is always generated. Encrypting sub-keys are optional.

It may take a while. Wait.

Two files are created under the selected directory

- public key: email_pub.asc
- secret key: email_sec.asc

Please note that while MyPGP requires a password, its robustness is not checked. It is on your own to use adequate passwords or passphrases.

alias

The user that generates the key, decides the name that appears in it. Others cannot modify that name. If we want to have another name, we can assign a local alias.

Select a key and click keys / alias.

copy

MyPGP copies name and fingerprint of selected key(s) to the clipboard.

export

MyPGP generates files containing the selected public and secret keys.

For public keys, simply an X_pub.asc file is generated in the selected directory.
For secret keys, it is required to facilitate the protecting password and enter a new password. It can be used to change the password of a secret key. One X_sec.asc file is generated in the selected directory.

Be careful about the destination directory. If the new key is under MyPGP HOME directory, it will be loaded on next execution. If the same key is loaded twice with different passwords, MyPGP will use the last loaded one, and it may cause some confusion.

refresh

Reloads keys from HOME.

files & keys

Presents a record of which keys are loaded from each file. Please note that rings from other pgp software may enclose many keys.

6 Menu lists

add key(s) to list(s)

Select one or more lists, and one or more keys. All the selected keys are added to all the selected lists.

remove key(s) from list(s)

Select one or more lists, and one or more keys. All the selected keys are removed from all the selected lists.

new list

Create a new list.

remove list(s)

Remove a list. Keys in the list are removed only from the list.

7 Menu clipboard

Select some text on any screen. Cut it to the clipboard (ctrl-C, usually). Operate on the clipboard and paste the result where appropriate (ctrl-V, usually).

Public keys. Used for encryption. Select one or more public keys. You can select

- one by one, or
- select one or more lists, or
- select one or more [sub-]directories.

The operation is performed with all the keys selected, directly or indirectly.

Secret keys. Used for signing. You will be prompted for the secret password.

view

Opens one screen to display the contents of the clipboard.

encrypt

The clipboard is encrypted for all the selected public keys.

sign

The clipboard is signed with the selected secret key.

encrypt & sign

The clipboard is signed with the selected secret key, and encrypted for all the selected public keys.

decrypt & verify

The clipboard has information on whether it is signed, encrypted, or both. In order to verify a signature, if present, the user only observes the outcome. In order to decrypt, the user will be prompted for a password.

8 Menu files

Select one or more files with the explorer. The operations will be applied to each of the selected files.

Public keys. Used for encryption. Select one or more public keys. You can select

- one by one, or
- select one or more lists, or
- select one or more [sub-]directories.

The operation is performed with all the keys selected, directly or indirectly.

Secret keys. Used for signing. You will be prompted for the secret password.

encrypt

The file is encrypted for all the selected public keys.

sign

The file is signed with the selected secret key.

encrypt & sign

The file is signed with the selected secret key, and encrypted for all the selected public keys.

decrypt & verify

The file has information on whether it is signed, encrypted, or both. In order to verify a signature, if present, the user only observes the outcome. In order to decrypt, the user will be prompted for a password.

secure delete

The file is removed securely. First, it is rewritten with alternating 0's and 1's. Then, it is removed from the directory.

9 Menu language

Choose language for the interface.

10 Other topics

10.1 Elgamal encryption

Elgamal encryption uses a random secret in a cyclic group G of order q with generator g. Generating such a group may take quite a while (from several minutes to hours). Though secure, it may not be very convenient.

MyPGP offers 3 options:

IETF

MyPGP uses one of the known groups standardized by IETF

- RFC 5996 Internet Key Exchange Protocol Version 2 (IKEv2)
- RFC 3526 More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)

This option is very fast. The groups are known to be secure.

These groups are hardcoded in MyPGP.

bits	strength	DH Group	RFC
1024	~80	2	4306
1536		5	3526
2048	~112	14	3526
3072	~128	15	3526
4096		16	3526

GnuPG

GnuPG uses several small prime numbers. The generation of random prime numbers is based on the Lim and Lee algorithm to create practically safe primes. This algorithm creates a pool of smaller primes, select a few of them to create candidate primes of the form

$$p = 2 * p_0 * p_1 * ... * p_n + 1,$$

tests the candidate for primality and permutes the pool until a prime has been found.

This option is fast enough to be practical.

This algorithm is implemented by MyPGP using BouncyCastle to generate prime numbers and to check primality.

Standard

The classical approach is to find a secure prime p (size selected by the user). To discover it, MyPGP follows the standard procedure

1. choose a random number q; compute p = 2q+1 and repeat until p is prime.
2. select a generator g

Step 1 consumes a hell of time. Step 2 is fast.

The implementation of this algorithm is copied from BouncyCastle into MyPGP to add support for user cancelation.

10.2 Elliptic curves (EC)

MyPGP supports the following curves:

bits	NIST FIPS 186-4	SEC	RFC6637
192	P-192	secp192r1	
224	P-224	secp224r1	
256	P-256	secp256r1	must
384	P-384	secp384r1	may
521	P-521	secp521r1	should

Be aware that elliptic curves are only supported by a limited number of implementations of PGP.

10.3 Strength (bit length)

Source: http://www.keylength.com/

	RSA	DSA Elgamal	ECDSA ECDH	symmetric keys	comment
2010	1024	1024	160	80	should be avoided
2010-2030	2048	2048	224	112	ok for normal use
> 2030	3072	3072	256	128	ok for normal use
>> 2030	7680	7680	384	192	long term
>>> 2030	15360	15360	512	256	long term