

Practical Applications of Public-key Cryptography: Securing Email Communications with PGP

By pagvac (Adrian Pastor), Petko Petkov and Rabia Barakat - December 2004

When it comes to asymmetric cryptography the most popular and widely used application that comes to anyone's mind is PGP. PGP stands for "Pretty Good Privacy" and is the standard public key cryptography application used today.

In the examples of this project we chose to use PGP Desktop. The reason for this choice is that PGP Desktop is easier to use than other text-based versions of PGP such as gnuPG. PGP Desktop provides us with a very intuitive GUI accessible from the Windows Start Menu (see Figure 1), the PGP taskbar icon (see Figure 2), and from Windows explorer (shell integration). So from now on, every time we mention PGP, we will be referring to the *PGP Desktop* version.

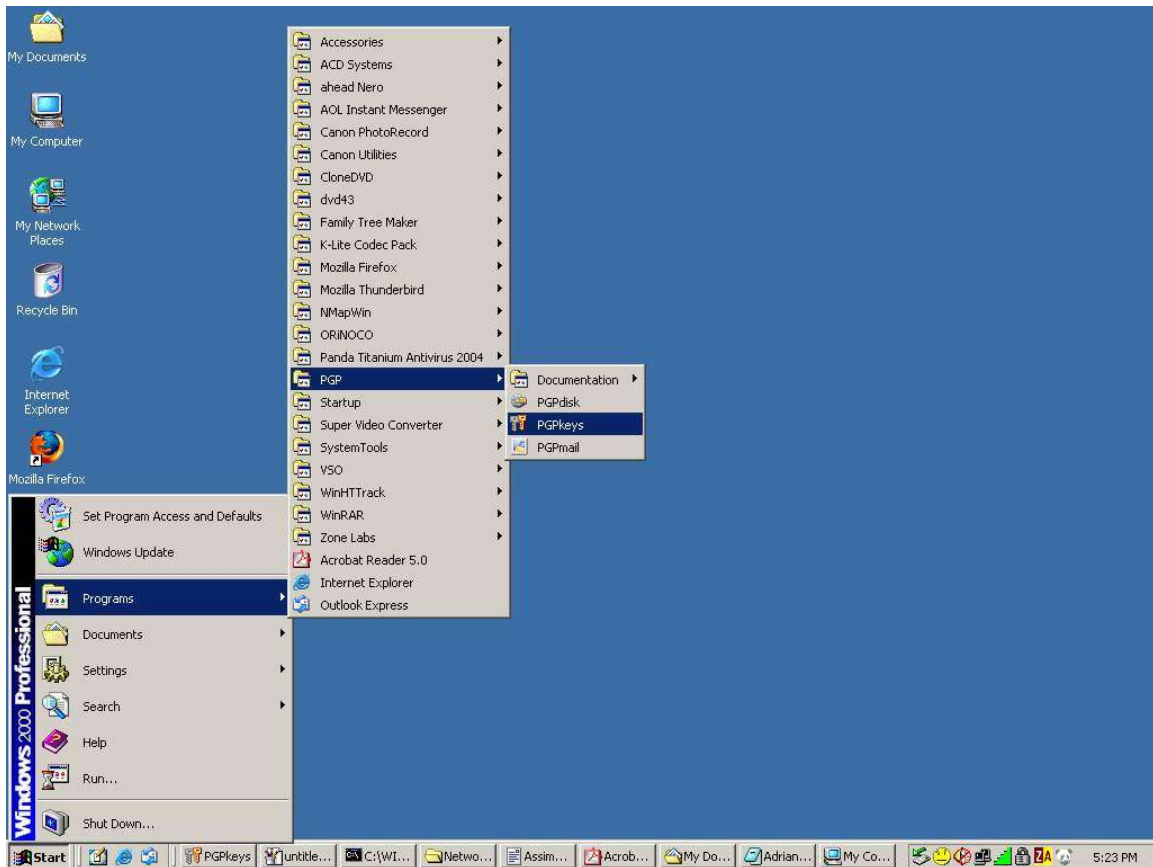


Figure 1

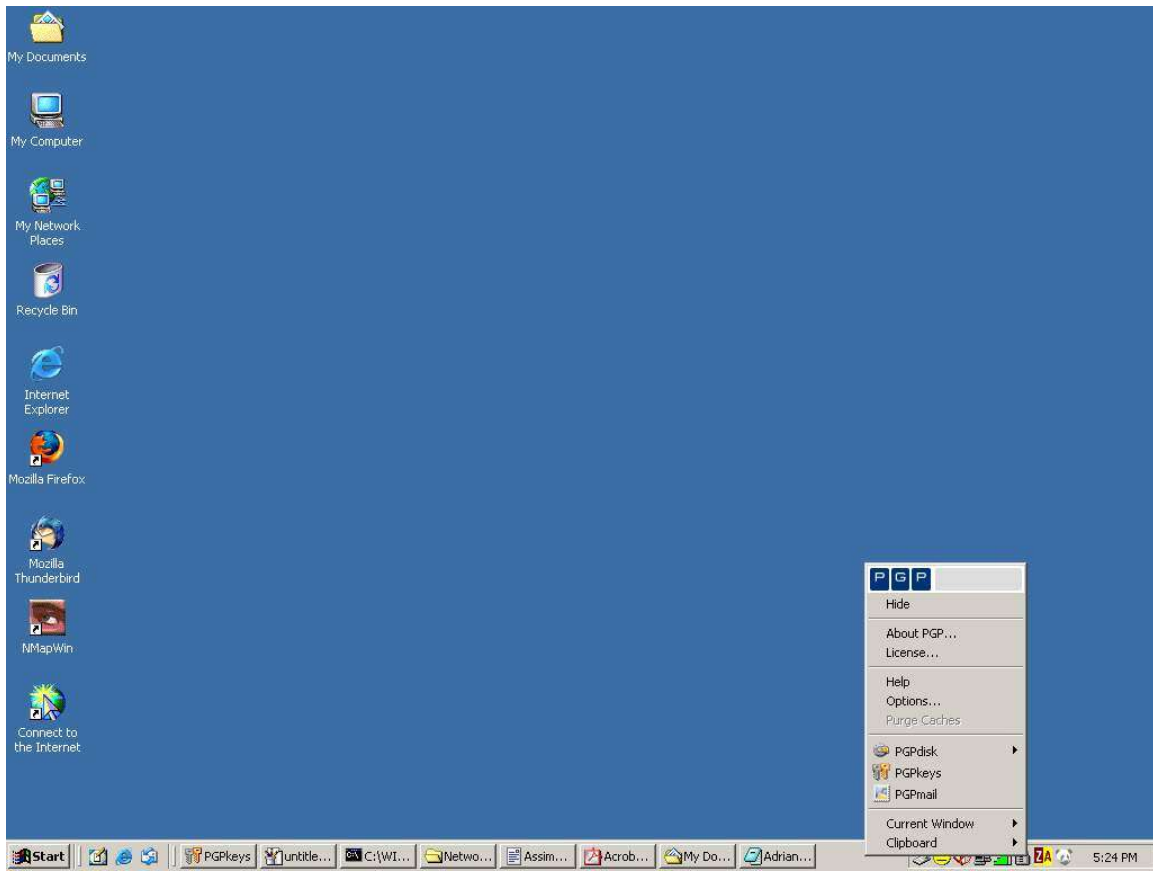


Figure 2

PGP allows us to secure our data in different ways. We can encrypt our emails, store devices (harddrives, floppy disks, flash drives, etc...), ICQ chat sessions, and truly wipe out files from our harddrive (Windows by itself does not do this when we “delete” a file). PGP also allows us to manage public keys by offering a public keyserver (idap://keyserver.pgp.com). Both public and private keys are saved in what PGP calls “keyrings” which are just files. The main components of PGP Desktop are the following:

- PGPkeys: used to create our keypairs (private key and public key), share our public key/s with other individuals (by uploading them to a keyserver) and add other individuals' public keys to our keyring
- PGPmail: used to encrypt our emails (by using the recipient's public key) and decrypt other people's emails sent to us (by using our private key)
- PGPdisk: used to encrypt and decrypt portions of our harddrive
- Documentation: this includes information on both cryptography, and PGP Desktop (see reference 1).

After we install PGP Desktop, we need to give some information about ourselves. These are our “full name” and our “organization name” (in case we use PGP within an organization/company). The last field required is “license number”. The license number is ONLY required in the case that PGP is intended for commercial use. In our case we do not need to enter any license key since we will be using PGP for personal use.

License key numbers need to be purchased from www.pgp.com. There are two reasons why we would want to purchase a license key. The first reason is that using PGP for commercial use without a license key is illegal since it violates the license agreement (*see reference 2*). The second reason is that when no license key is entered, PGP becomes “PGP Freeware”. PGP Freeware is not only free but is also a limited (restricted) version of PGP. In other words, we will not be able to use all the features without owning a license key!

The main two features that are disabled in the Freeware version of PGP are “PGPdisk” and “PGP email plugins”. This means that without a license (this is our case) we will only be able to perform two different tasks: manage our keyring and encrypt/decrypt email. Therefore, in our study of PGP we will focus on securing email communications.

The first three tasks a new PGP user must learn to do are the following:

- Create a keypair
- Upload public key/s to a keyserver
- Download other people's public keys from a keyserver

In order to create our keypair (public and private key) we need to access PGPkeys. In our case we chose to access it from the start menu:

- Start/Programs/PGP/PGPkeys
- (*see Figure 1*)

For the purpose of illustrating our example of encrypting/decrypting email we will use two imaginary characters: Monserrate and Adrian. Monserrate will be the sender and Adrian the recipient. So Monserrate will encrypt an email by finding Adrian's public key from a keyserver. Then Adrian will decrypt the received email by using his corresponding private key. Please remember that *only Adrian's private key will decrypt the data that was encrypted with its correspondent public key*.

The problem with this scenario is that Adrian doesn't have any way of verifying that it was truly Monserrate who used his public key to send him an email and not someone else claiming to be Monserrate! In order for Adrian to verify that it was Monserrate who sent him the email, Monserrate will need to sign the email with her private key. This is what we call a digital signature (*see reference 3*). This signature can only be verified with its correspondent public key and cannot be modified or copied to a different document.

The following are the steps required for Monserrate to send an encrypted and signed email to Adrian:

1. Create keypair
2. Download Adrian's public key from keyserver and add it to her keyring
3. Encrypt email using Adrian's public key
4. Sign message using her private key
5. Send email

If we still don't have any keyring configured then the PGPkeys screen will not show any keys (*see Figure 3*). The PGPkeys screen is in charge of showing our keypair and public keys that are part of our keyring.

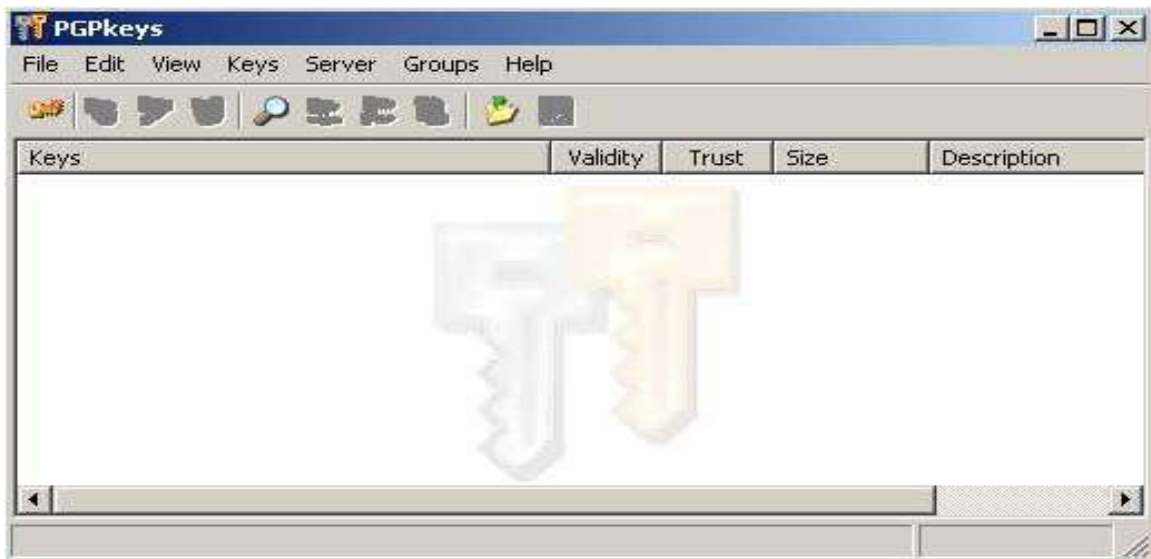


Figure 3

After opening PGPkeys we must proceed with the creation of our keypair. For that we need to click on "Keys/New Key..." and then follow the wizard. This wizard will ask us our full name and email address (*see Figure 4*).



Figure 4

After that, we will be asked a passphrase which is just a combination of words (*see Figure 5*). This passphrase will be the mechanism to ensure that only Monserrate can use her private key. This means that if a malicious user grabbed Monserrate's keyring (which contains her private key) he/she would not be able to use Monserrate's private key without knowing the corresponding passphrase. When entering the passphrase, PGP will indicate the degree of quality. The longer the passphrase is, the more secure it will be.



Figure 5

After entering the passphrase all we need to do is click on "Next" and our key will be generated. If everything was done correctly our new keypair should be added to our keyring and shown in the PGPkeys screen (see Figure 6).

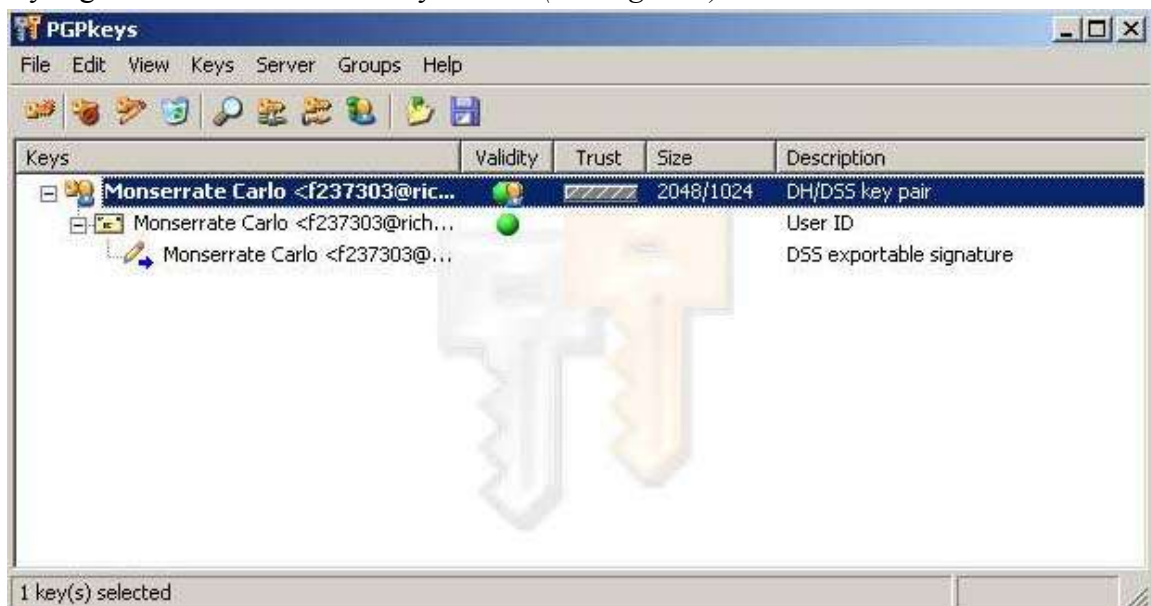


Figure 6

Now that Monserrate has her keypair what she needs next is Adrian's public key so that she can encrypt an email that only Adrian can decrypt. To search for Adrian's public key we need to click on “Server/Search...” and enter either his name or email address (see Figure 7). After finding the right key all we need to do is to add it to Monserrate's keyring by right-clicking on the public key and then clicking on “Import to Local Keyring” (see Figure 8). Now, we should be able to see Adrian's public key on Monserrate's keyring (see Figure 9).

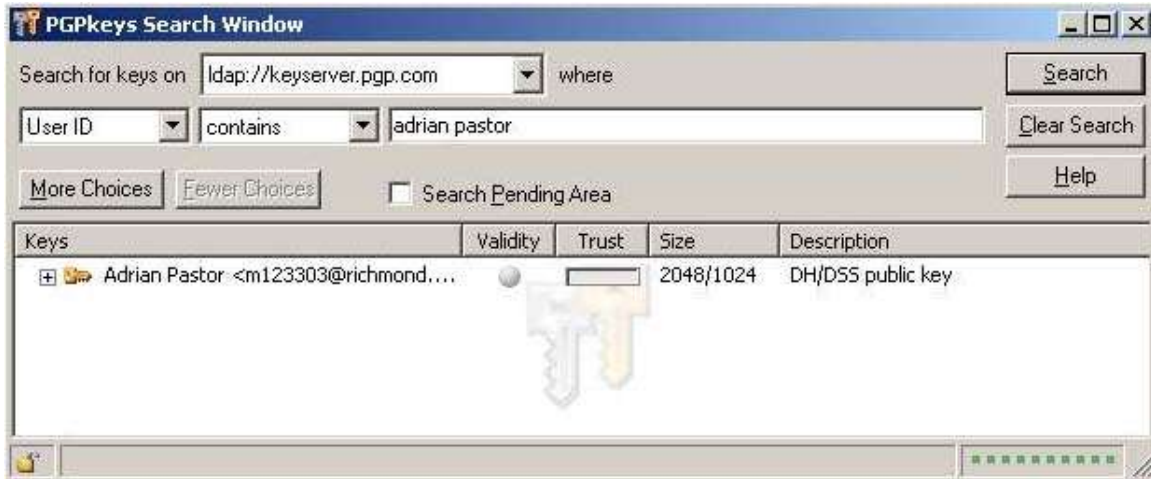


Figure 7

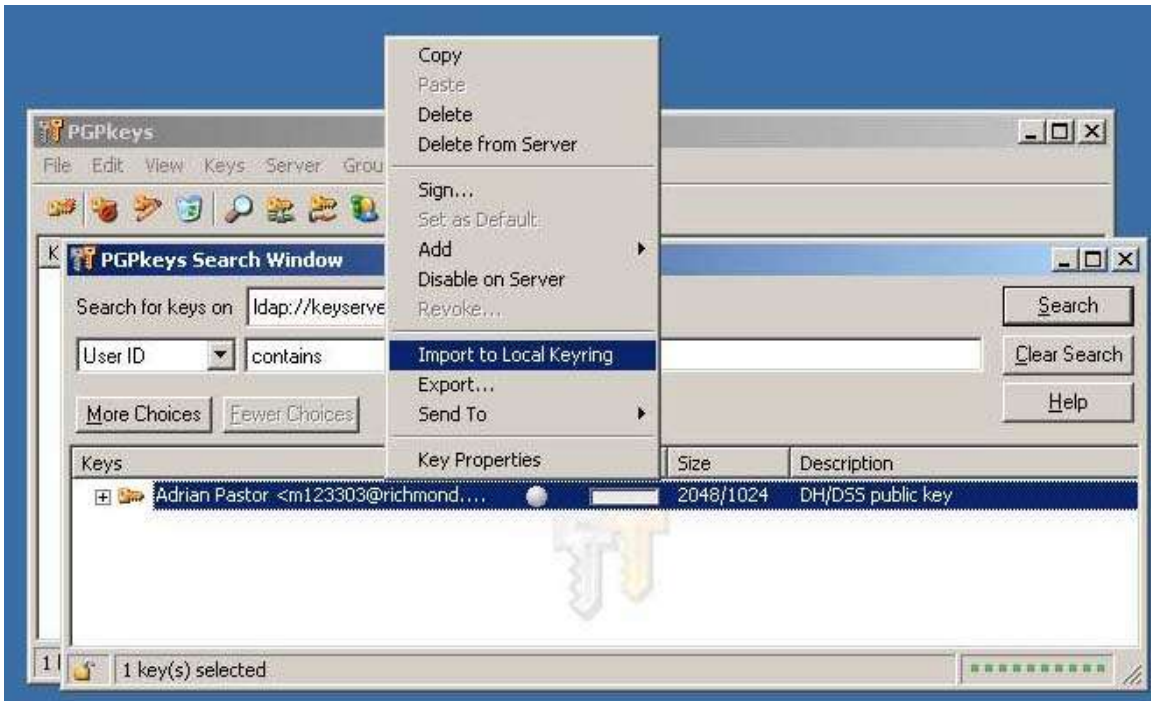


Figure 8

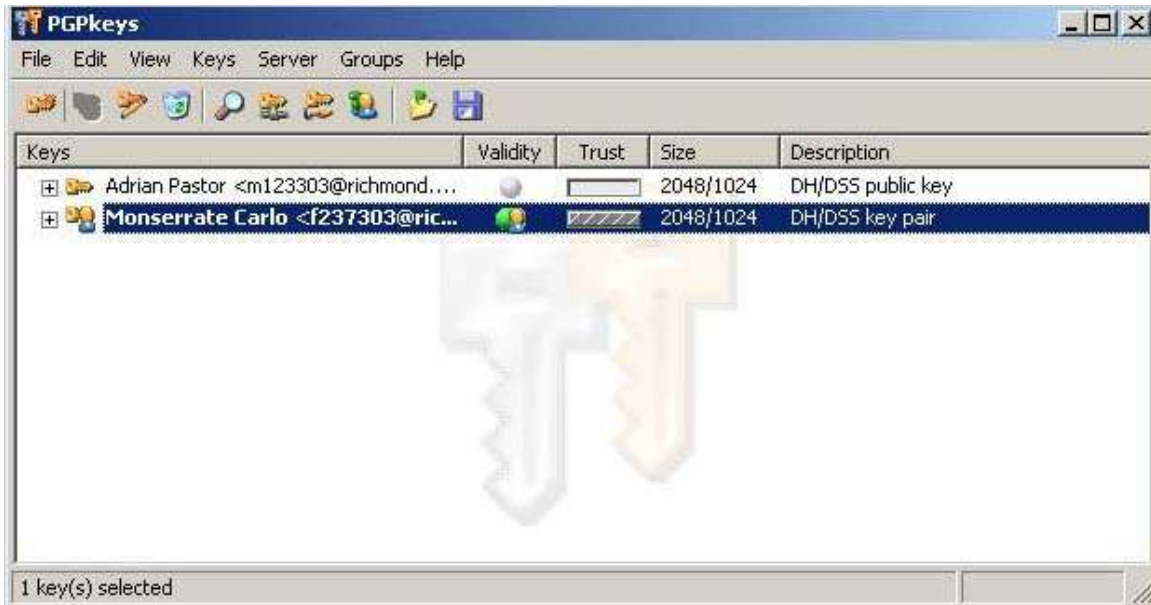


Figure 9

Finally all we have to do left is encrypt and sign the email it by selecting the window where the message is contained and clicking on the “PGP taskbar icon/Current Window/Encrypt & Sign” (see Figure 10). After that, we will need to choose Adrian as the recipient and enter Monserate's passphrase in order to sign the message with her private key. As a result, we will have an encrypted message that only Adrian will be able to decrypt with his private key (see Figure 11).

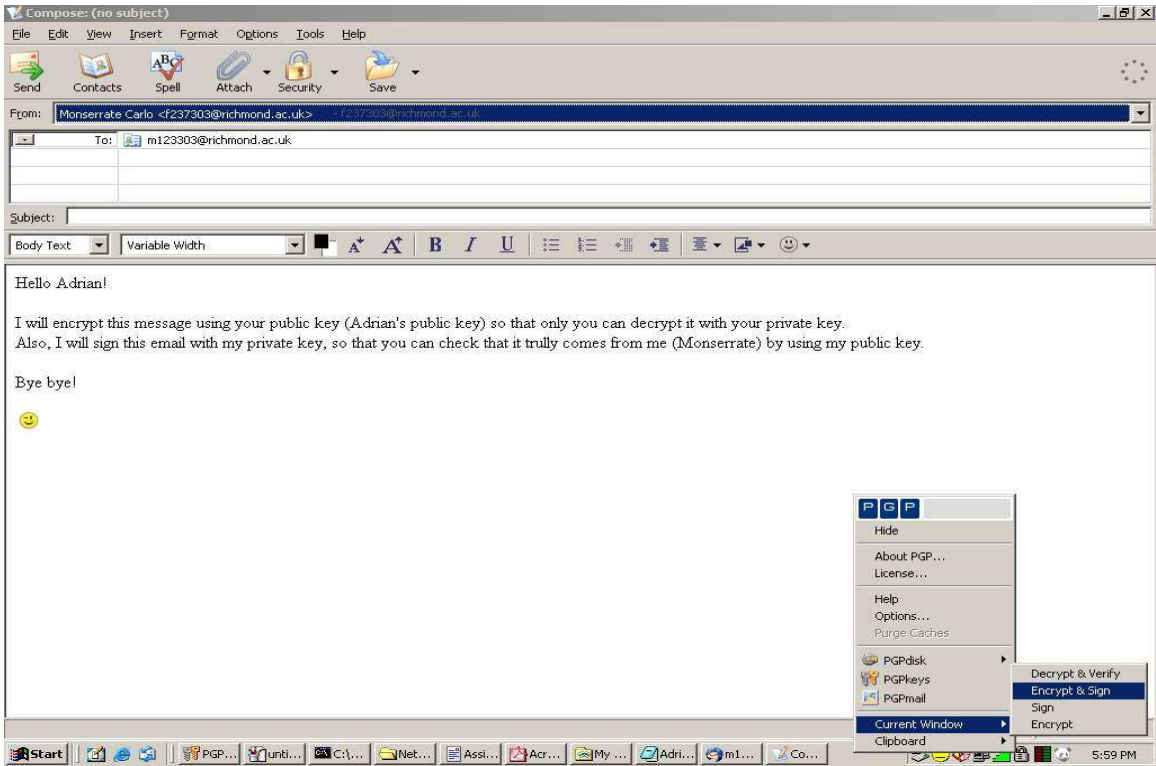


Figure 10

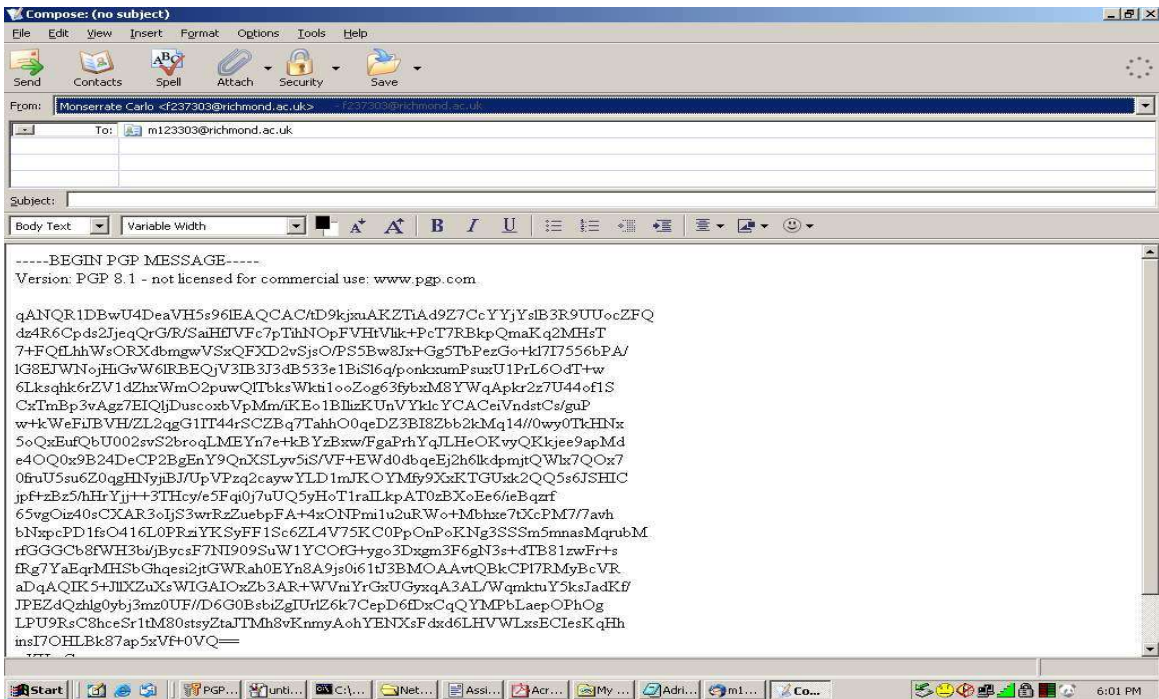
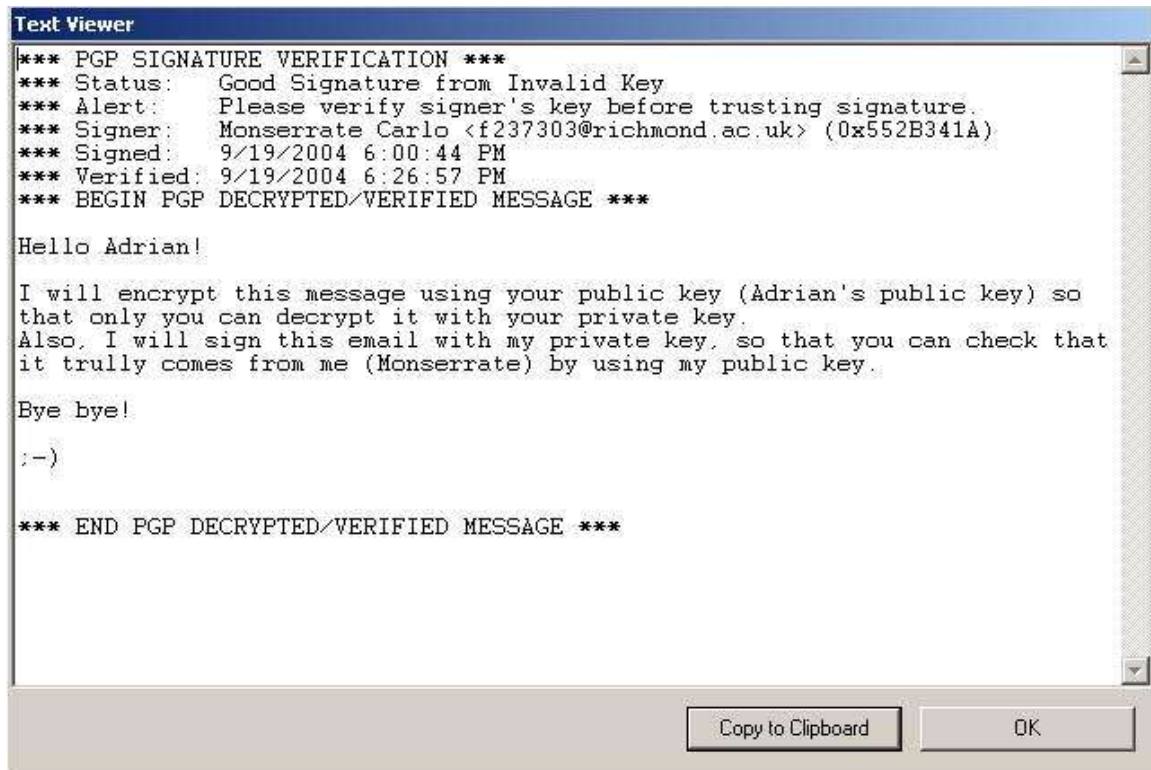


Figure 11

Finally, all Adrian will need to do when he receives the email from Monserrate is decrypt it with his private key and verify the Monserrate's signature. In order to do this, we need to select the window that contains the encrypted message received from Monserrate and click on "PGP Desktop taskbar icon/Current Window/Decrypt & Verify".

In order to decrypt the message, Adrian needs his private key. So, he will be prompted to enter his passphrase (remember that the passphrase is always needed when using the private key). The verification of the signature is done automatically by PGP by checking if the digital signature matches its corresponding public key (Monserrate's public key). In the end, we get our precious plaintext message (*see Figure 12*).



```
Text Viewer
*** PGP SIGNATURE VERIFICATION ***
*** Status:    Good Signature from Invalid Key
*** Alert:     Please verify signer's key before trusting signature.
*** Signer:    Monserrate Carlo <f237303@richmond.ac.uk> (0x552B341A)
*** Signed:    9/19/2004 6:00:44 PM
*** Verified:  9/19/2004 6:26:57 PM
*** BEGIN PGP DECRYPTED/VERIFIED MESSAGE ***

Hello Adrian!

I will encrypt this message using your public key (Adrian's public key) so
that only you can decrypt it with your private key.
Also, I will sign this email with my private key, so that you can check that
it trully comes from me (Monserrate) by using my public key.

Bye bye!

;-)

*** END PGP DECRYPTED/VERIFIED MESSAGE ***

Copy to Clipboard  OK
```

Figure 12

As we have seen, public-key cryptography is extremely useful for securing our communications in our daily life and maintaining our right to privacy. Thanks to great tools like PGP Desktop, anyone who understands the basic principals of asymmetric cryptography can secure his data quite easily by using a very straight-forward GUI.

References

1. PGP/Help/About PGP
2. PGP® Desktop for Windows User's Guide (*found in the documentation of PGP® Desktop 8.1*). Page 7. June 2004.
3. *An Introduction to Cryptography* (*found in the documentation of PGP® Desktop 8.1*). Page 17. June 2004.

Glossary

- Key ring: component used to keep all your public and private keys.
- .pkr: PGP public key ring file. It is used to store our recipients' public keys
- .skr: PGP private key ring. It is used to store our private key/s. This file should be kept in a secure place and should not be lost.
- GUI: Graphical User Interface
- Passphrase: combination of words used to maintain exclusive access your private key
- Plaintext: readable message (non-cypher message)

Useful Links

PGP Freeware

- <http://www.pgp.com/downloads/freeware/index.html>

Purchase and Download FAQ

- <http://www.pgp.com/storefaqs.html>

Installation FAQ

- <http://www.pgp.com/purchasefaqs.html>

Renewal FAQ

- <http://www.pgp.com/renewalfaq.html>