Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

# Using GPL Software For Email and File Encryption

Because privacy is important, the security of information is sometimes legally required, and internet communication often does not provide this necessary security inherently. Email encryption and file encryption can provide a higher level of security for internet communication, but too often providers of proprietary encryption technology and related services like the PGP Corporation's PGP encryption software are too expensive to fit the budgets of people or organizations involved. I have researched the viability of usi...

Copyright SANS Institute
Author Retains Full Rights

AD

**Using GPL Software For Email and File Encryption**
**Version 1.4b Option 2**
**May 12, 2003**

## Abstract

Because privacy is important, the security of information is sometimes legally required, and internet communication often does not provide this necessary security inherently. Email encryption and file encryption can provide a higher level of security for internet communication, but too often providers of proprietary encryption technology and related services like the PGP Corporation's PGP encryption software are too expensive to fit the budgets of people or organizations involved.

I have researched the viability of using GNUPG, the GNU Privacy Guard software, for file encryption and email encryption for use within my organization. GNUPG is free and open-source software, so the only foreseeable contraindication was usability: is it easy enough to install, use, and manage to deploy and support for several thousand users. With the help of two other pieces of open-source software, the GDATA plug-in for Microsoft Outlook™ and GNUPG Shell Extensions, it did prove itself to be a worthy solution, will save us the $70-$80 per seat licensing of PGP, and can be deployed to an unlimited number of users.

## Before Snapshot

Before implementing GNUPG[1], my organization relied on a single employee, who had purchased his own personal PGP software client, in order to send encrypted files.
Any time anyone in the organization needed to encrypt email or files, they had to contact the one individual with a valid license to meet their need. When this employee ceased working in my organization and no funds were available to purchase PGP software, and we were desperate to find alternatives. Out of this desperation, I began researching GNUPG. To understand the importance of our mission in finding a way to encrypt information, I will further explain the function of my organization and how it uses encryption to eliminate risks associated with the sharing of sensitive information.

Our organization uses MS Outlook™ as its email client and an independent agency maintains its exchange server. Nearly every user desktop runs some version of Microsoft Windows. Most of them are either Windows NT 4.0 or Windows 2000 Professional. The internal technology staff, including myself, maintains all of the other file servers and end-user desktops for the organization. The internal technology staff also manages the installation and use of email clients for local and remote users. Presently, we consist of 3 Windows NT 4.0 domains spread across 230 locations throughout the state. Our internal technology staff also assists 15 other administrators who maintain 4 other independent domains for our organization. An installation guide was needed so staff at remote locations could install the encryption software. Our staff

[1] http://www.gnupg.org

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

does not at this time have administrative rights in those 4 NT Domains, so we need a common document that our entire organization can use as a standard.

The organization I work for is a government agency that provides statewide health-related services and as a result, has access to case-sensitive data. This includes personal information on people with sexually transmitted diseases like HIV/AIDS, as well as other sensitive data related to birth records and drug control. The organization also has a need for encryption in other areas. One of which involves the circumstances in which it acts as a "middleman" between a pharmaceutical manufacturer and an agency with locations across the state that dispense the pharmaceutical products. Other agencies within the organization need the ability to communicate securely with outside medical sources. Because of the sensitive nature of the information, the communication that is sent and received must be encrypted. Our organization also needs to be able to transmit email and files between our various divisions that need to be encrypted. Files on common file servers are very vulnerable to being read or tampered with, so by encrypting those sensitive files we can add another layer of security to our organization. Email is sent in clear text unless it is encrypted, so now that we can have this ability (the ability to send encrypted email) for free our organization is finding new ways of doing business.

## **During Snapshot**

I experienced no problems running GNUPG from a Windows command prompt with my initial installment. I enjoyed and made use of the tutorial, GNU Privacy Guard - a User's Perspective[2]. This tutorial really helped with my understanding of the entire public/private key process. Like most people that I talk to about encryption, I had a hard time understanding the public/private key process. The concept of giving away my public key or putting it on a server that someone could download it and change it, seemed dangerous to me, but after reading the tutorial I was able to put all of the pieces of the puzzle together. I spent a lot of time learning that I use the recipient's public key to encrypt things to send to them. What helped me the most was realizing that the private key always does the decrypting. I was confused that people could somehow decode or hack things with my public key.

One major benefit of using GNUPG is that it is compatible with the PGP standard. This means that we can communicate with users all over the world that have software that is compatible with the PGP standard. There are both commercial and freely available applications that are compatible with the PGP standard so there are little millions of users who use this standard. PGP has been in existence for over ten years so it really has a proven track record.

Most of my users would not be able to use the command-line version of the product and initially I was not aware of any plug-in for Microsoft Outlook™, therefore, I was uncertain as to whether GNUPG would meet our needs. GNUPG in and of itself does not have

[2] http://www.chromebob.com/proj/gpg-tutorial.html

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

the ability to integrate with email clients. So in order to be able to encrypt email I had to find a program that would work with both GNUPG and Microsoft Outlook. While perusing the GNUPG website I did find a plug-in that would work for Microsoft Outlook™. Microsoft Outlook is by far our organization's most used application and having the ability to encrypt email through that application really fills a void. I then downloaded the G-Data GNUPG MS Outlook™ Plug-in[3] and installed it on my test machine.

The G-Data plug-in provided the option of installing GNUPG and a key manager along with the outlook plug-in. The key manager that came with the plug-in did not function well. It would not import keys, including my private key which is pretty necessary to the encryption/decryption process. However, it worked great for encrypting and decrypting email if I used the command line interface to import my keys first. Even after importing keys in this manner, the G-Data Plug-in didn't show these imported keys properly or at all in its graphical interface.

After many attempts, I discovered that I could install the plug-in without the key manager for email encryption functionality and install GNUPG and GNUPG Shell Extension[4] separately for a key manager and for file encryption functionality. The feature that I liked best about the Shell Extension software was the fact that after its installation, I could right-click on any file that I wanted to encrypt and choose GNUPG > encrypt from a drop-down menu. That kind of easy use and integration with the normal look and feel of Windows operation was exactly what would make this project successful.

The GNUPG Shell application was much more successful at managing all of my keys, (private key, public key and any public keys that I import) and provided an easy mechanism to do file encryption. However, I still wanted to use the G-Data Plug-in to encrypt and digitally sign email. Using GNUPG to digitally sign email adds another layer of authenticity to email correspondence, in order to sign an email you must use your private key and pass phrase and the recipient must have your public key in order to verify the signature. The reason this adds security to the email is that the pass phrase has to be used, so if someone were to spoof your email address chances are they would not have your private key and pass phrase and would be unable to forge your digital signature.

From the onset, I experienced no problems getting the three programs to work independently, it was when I decided to use them together that I discovered problems making them share the same imported key and trust databases. One reason for this difficulty is that the G-Data plug-in defaulted to install its own version of gpg.exe (gpg.exe is the actual binary file that GNUPG installs) in it's own specific program directory. The G-Data plug-in also wanted to keep all of the GNUPG configuration information such as public and private keys, and it's trust database in the same specific

[3] http://www3.gdata.de/gpg/index.html
[4] http://web.utanet.at/ascherst/gpgsx/

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

program directory. This became a problem when we decided we wanted to use the newest release of GNUPG, but its default configuration looked for all of the above mentioned binaries and configuration files in a different folder, c:\gnupg. I fumbled around for days trying to sync up copies of GNUPG information in their two respective folders. I then decided that I needed to choose one folder for everything. My first attempt at having everything in one folder failed miserably. I installed GNUPG and allowed it do the defaults and I then installed the G-Data plug-in and told it to install in c:\gnupg instead of it's default location in c:\Program Files\GnuPGExch. With this configuration, the plug-in would not work at all nor would it allow me to uninstall it.

My next plan was to install the G-Data Plug-in first and then install GNUPG into the G-Data program folder. This failed as well, but I discovered a key piece of information that eventually helped me find a final solution: There is a registry file included in the GNUPG distribution that one is supposed to enter into the registry during installation. It includes the path to the location of the GNUPG binary. To make GNUPG install successfully in the G-Data program directory, I had to first manually edit the path in the registry file before installing it. Instances of c:\gnupg needed to be changed to c:\Program Files\GnuPGExch.

Once I made this edit, I was off and running with a reliable way to encrypt and decrypt email through extra buttons added to my MS Outlook™ toolbar by the G-Data plug-in. The G-Data plug-in is really the most crucial application of the three because it is the one that the end user uses most often and it also happened to be the on that would only work in it's default installation path. However, I was only able to generate my private/public key pair and perform other key management tasks from the command prompt at this point. The graphical key manager that the G-Data plug-in installs would not recognize either my private/public key pair or any valid keys that I imported. I knew that this (using the GNUPG program gpg.exe at the command line) would not be an option for my end users, most of them have never used the command prompt and asking that they correctly type in DOS commands with the right spaces and syntax didn't seem to be a good idea. I had to have a viable solution for a graphical key management program.

A working GUI key management utility and an easy method for encrypting files were exactly the functionalities provided by the GNUPG Shell Extensions program. The GNUPG Shell Extensions program also assumed that GNUPG binaries and configuration information would reside in c:\gnupg. It also insisted on keeping one of its own configuration files in c:\gnupg. It gave me error messages and refused to function properly until I created an empty c:\gnupg folder in which it could place its "options" file.

In retrospect, the above-outlined procedure could have had everything working appropriately in the c:\gnupg folder, but because I had already deployed this software to part of my organization with GNUPG installed in the G-Data program folder, I thought it would be more appropriate to standardize and to maintain consistency across the organization. In the future I will learn to test with a smaller user base and not let as many people in on the ground floor of the project. It would have in retrospect been

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

better to have everything pointing to c:\gnupg, but because of my inexperience of moving the public and private keys from folder to folder and computer to computer, I decided to leave thing as they are especially since part of my organization went live with the G-Data plug-in. The G-Data plug-in is by far the most finicky application of the three, every time that I installed it into a directory other than what it wanted for it's default directory I couldn't get it to work. On one occasion I couldn't even get it to uninstall, so that is why I installed the other programs around its needs.

After documenting my successful installation procedure, several co-workers working in information technology at my organization were given a copy of the instructions and asked to test the installation procedure. During this time, my documentation underwent several revisions as a result of feedback from "test" users.

Once we determined that each of the three programs involved was cooperating with the others and that we could successfully deploy these applications to both Windows NT 4.0 and Windows 2000 Professional computers, we then finalized the an installation manual for management and select technology staff to review.

After review by technical staff, I distributed the installation guide to yet another group of "power users" and asked for feedback, which I also incorporated into the installation guide. This uncovered many issues that I hadn't even thought of, one of the most comical was one of our users that wanted to decrypt things on his Blackberry wireless email pager. As of this writing I am still unable to help him. We also got feedback from people that use Microsoft's Outlook Web Access. Microsoft Outlook Web Access is a program from Microsoft that allows users to access their email using any standard web browser. This allows our users to send and receive email from any place that has an internet connection, thereby making their email highly available. Again I am unable at this time to help them finding a client or a way to decrypt their mail when using the Microsoft Outlook Web Access. So we have had to state that we are presently only supporting encryption from the users' primary work computer, or from computers that can run Microsoft Outlook.

The final hurdle was to add a section to the installation guide explaining how to use private keys to encrypt files and encrypt email while moving keys between multiple computers. This seemed necessary for users that used multiple computers and needed encryption functionality on more than one. Importing and exporting keys is not as simple as copying and pasting files. The biggest obstacle that we have to overcome is that most of our users created a new private/public key pair when they installed the three programs on the next computer that they were going to use. This created mass confusion. I had to emphasize that they should not create another private/public key pair when they next installed the software. Three users had sent out two different public keys and had people encrypting things for them with their second public generated key that the user was trying to decrypt with their first generated private key. I had to finally get those users to standardize on the second key pair that they generated and had them try and track down who they had originally emailed their first public to and send

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

those people their second generated public key. Thankfully they hadn't published either of their keys to any key servers.

Key servers are used as a central repository for people to upload their public keys where other users can search for them and download them. One major benefit of using key servers is that if you need to revoke a public key you can publish the revocation to the key server and it then labels the revoked key appropriately so people can easily see it is no longer in use. That way you don't have to email everyone and tell them to stop using the public key that you are revoking.

We decided that since it is not very easy to find many publicly available key servers that we would pick one key server to use for our entire organization. We wanted a key server that would be publicly available and highly available. We decided to use http:pgp.mit.edu, and so far we have been very pleased. The use of this key server helps us to have our keys in one central place that users can get to from anywhere that is connected to the internet and it also means that we don't have to worry about distributing our keys via email or other means.

I added a section to the installation manual explaining how to create a revocation key, following the discovery of some users that they had created multiple private keys. I added this because it is important to create a revocation key at the same time the public/private key pair is created in the event that the pass phrase is forgotten or a key is compromised. Here are some reasons where the revocation key would be necessary: If someone were to be able to obtain a user's private key and be able to guess the user's pass phrase, they could then impersonate that user and thus compromise the entire encryption process for that user. If the user were suspicious that someone had obtained their pass phrase and private key, they could send out their revocation key and also publish the revocation key to any key servers where they had uploaded their public keys. Another scenario would be that if the user forgot their pass phrase. If this happened then the user would then be unable to decrypt anything that had been encrypted with their public key. Again the user could send out or publish their revocation key. They would then have to create another private/public key pair and repeat the process of publishing or distributing their public key. So it is very important to create a revocation key soon after the creation of the private/public key pair. It is very important therefore that users take great care in storing their private keys and that they also keep strong pass phrases. The public key does not need the same care, because it is no good without the corresponding private key. There is no pass phrase generated for the public key. The public key is meant to be distributed so that people can encrypt mail and files to send back to the user.

One of my users asked what at first seemed to be a very good question, "what happens if someone sends out a public key pretending to be me?" I came to the conclusion that since we associate the public key with an email address, they would also have to receive any return email coming to the real user's email address, and it is unlikely that someone would be able to easily do this.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

In the appendix, you will find my installation guide that we have adopted. It has been revised twice and is assumed to always be a work in progress. This installation guide was initially given to two individuals for proof reading and testing. After their comments were gathered and implemented into the guide, a training class was conducted for technology staff in remote locations so that they could independently install and maintain the three programs.

**After Snapshot**

The organization now has a common installation location and an accompanying guide that has been and will continue to be relentlessly tested. Thus far, things are working well.

From a technological point of view, I would standardize my install in the c:\gnupg folder if I could do it over again. However, in order to take the path of least resistance and to eliminate the possibility of interrupting the work of staff, I decided to leave things as recorded in the installation guide. I now realize that I could simply install all three programs in their default locations (GNUPG would install in c:\gnupg, GNUPG Shell would install in c:\program files\gpgsx and G-Data plug-in would be in c:\program files\gnupgexch) I would then only have to point the G-Data Plug-in to look for gpg in c:\gnupg. This would be less confusing and would nullify the need to edit the registry file that is part of the GNUPG program.

As the result of this study, the organization can communicate sensitive information internally and with outside sources such as physicians, hospitals, the National Center for Disease Control and pharmaceutical distributors. For example, when remote locations need pharmaceutical products, they notify my organization of the need. We, in turn, combine all requests into a database and forward this information to the manufacturer via an encrypted file. The manufacturer then decrypts the file and sends the requested products directly to the location from which it was requested. This is both an efficient and secure way of communicating sensitive information. To make things even easier, we have made use of a freely available key server[5] at pgp.mit.edu which allows the organization's staff to upload public keys and make them available to both internal and external sources.

Since the success of this project, the organization's technology staff has gone even further with its uses of encryption such us using it to verify digital signatures from mailing lists (SANS, Microsoft, etc.) and protecting confidential information such as passwords.

It is quite an accomplishment to be able to fulfill the need of the organization to communicate securely through the use of encryption, and to do so at no cost makes it an even greater accomplishment! I am very pleased that I was able to find free software that is compatible with a global standard (PGP). To be able to do this and have it to be easy to use creates a win-win situation for my staff. It has been very

[5] http://pgp.mit.edu

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

exciting and successful. In fact, we receive few telephone calls and/or emails from remote locations requesting assistance due in part to the installation guide that is provided. In respect to this challenge, we have accomplished all of our initial goals and continue to improve related services. Currently, we have trained our technology staff and outside users to install and use encryption and they are doing so successfully with the help of the installation guide. By using a universal encryption standard, we no longer have to depend on just one individual to send encrypted files. We can both encrypt files and send encrypted email and as an added bonus, we can digitally sign files and email.

I spent approximately three weeks working on this solution. It has been both extremely rewarding and extremely frustrating. I have tested the GNUPG program on a two different operating systems (Red Hat Linux and FreeBSD) and have found it to be extremely portable and functional. It actually works better on *NIX operating systems and in some cases GNUPG is even installed by default. I have exported my private/public key pair to a Mini compact CD that fits in my shirt pocket. I also have all three of the programs that we use on this CD, for both the Windows and Linux/ FreeBSD platforms. Now within minutes I can install software and my private/public key pair on a variety of different operating systems. This makes me extremely portable. My next step is to explore the feasibility of using a plug-in that will work with a web mail server. This would make encryption truly available from anywhere.

I am pleased that we are able to provide a free solution that meets the needs of the organization and as an added bonus has not caused an excessive support burden on our staff.

© SANS Institute 2003, Author retains full rights

## References


1. "The Gnu Privacy Guard", URL: http://www.gnupg.org
2. "G-Data Microsoft Outlook Plug-in", URL:http://www3.gdata.de/gpg/index.html
3. "GPG Shell Extensions", URL: http://web.utanet.at/ascherst/gpgsx/
4. "Gpg Tutorial", URL: http://www.chromebob.com/proj/gpg-tutorial.html
5. "MIT PGP Key Server", URL: http://pgp.mit.edu


© SANS Institute 2003, Author retains full rights.

## Appendix A

Encryption Software Install Guide
Revision 1.1

This installation guide assumes that you have network connectivity to \\p1\install to install the following programs:

1. GNUPG
2. G-Data MS Outlook Encryption Plug-in
3. GPG Shell GNUPG GUI for file encryption.

Otherwise you will need to download each program from the following locations:

1. http://www.gnupg.org
2. http://www3.gdata.de/gpg/index.html
3. http://web.utanet.at/ascherst/gpgsx/

Other assumptions are that you are authorized to install these programs and that you have local administrator rights to the machine where you are installing these programs. It is crucial that you follow the steps in the order listed in the installation guide. If you don't install in the order the guide details, then the programs will not work together and it will also create an unnecessary support burden for our staff.

This guide is still very much a work in progress, if you have any comments or corrections please email SERVICE REQUEST and we will address the issues as soon as possible.

Where we deem necessary we will revise the installation guide, increment the version number and publish it via email.

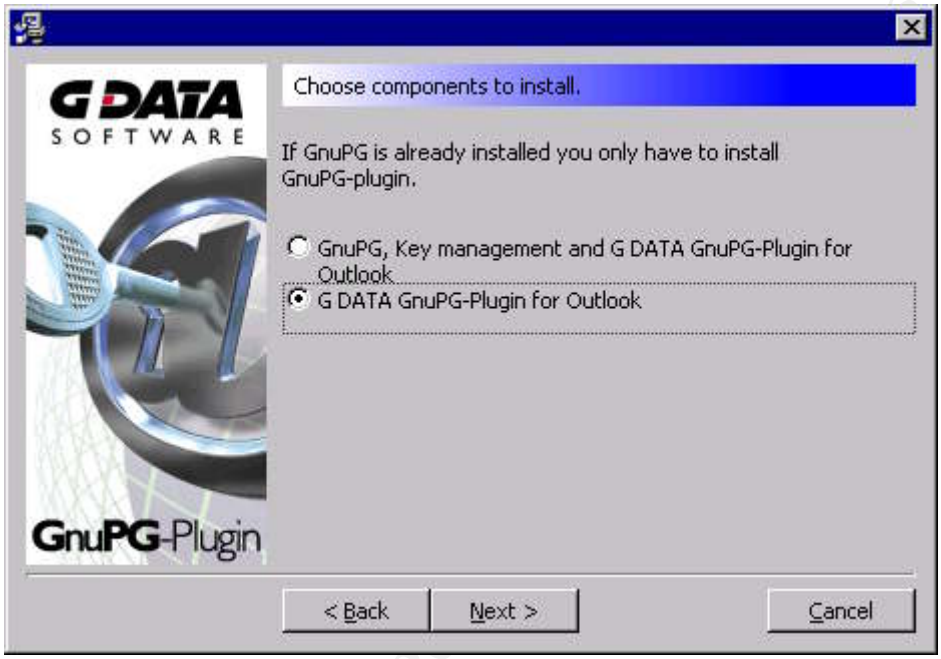
© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Installing the GNUPG Outlook Plug-in

1. The GNUPG outlook plug-in is found in our install Repository. Double click on the executable and the following screen should appear. Click next.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

2. This step is crucial. We will install GNUPG separately and as well as a key manager. Make sure that you select the bottom option that only installs the G-DATA GnuPG-Plug-in for Outlook. Click next after selecting the bottom G-Data option.




© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

3. Click finish

© SANS Institute 2003, Aut

Installing GNUPG

1. Now that the plug-in is installed we can install the newest version of GNUPG in the same folder in which we installed the G-Data Plug. First unzip the GnuPG archive to a temporary directory, then copy all of the files into the c:\Program files\GnuPGExch\

2. A registry entry needs to be made to tell windows where to find gpg.exe. To do this you can edit the gnupg-w32.reg file included in the GNUPG archive so it points to the correct location instead of c:\gnupg. This will ensure that gpg, the G-Data plug-in, and later the GPG Shell extension are all using the same gpg executable, keys and configuration data

Here is a sample of what the file gnupg-w32.reg should look like:

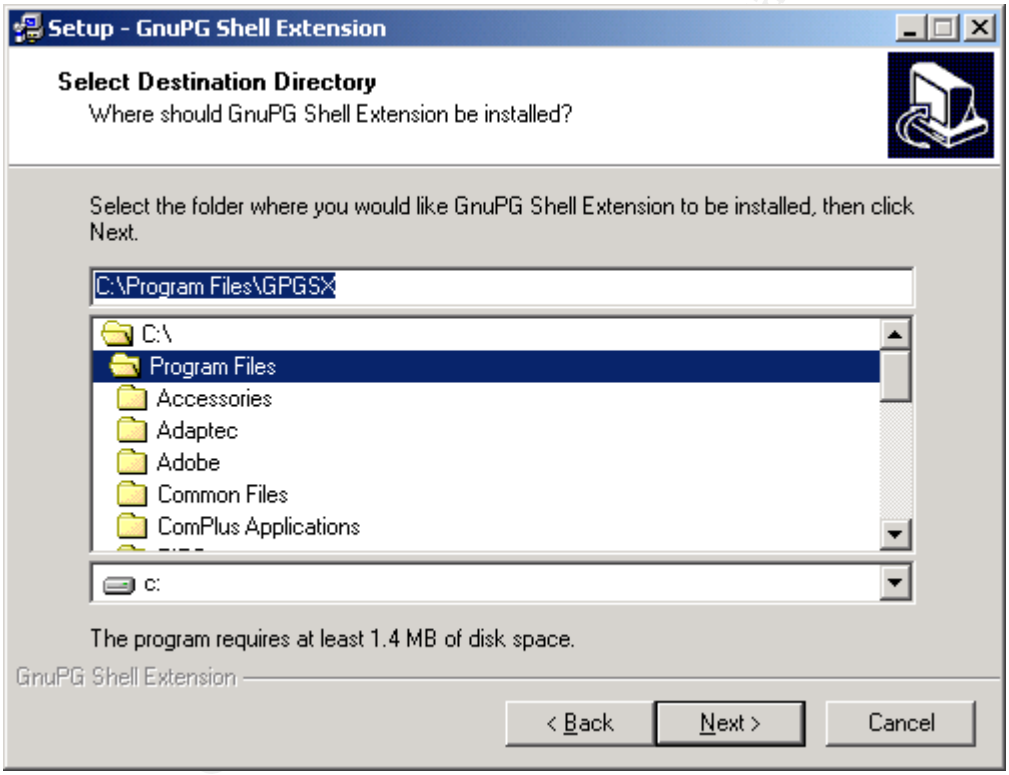
```
REGEDIT4

[HKEY_CURRENT_USER\Software\GNU\GNUPG]
"HomeDir"="C:\\Program Files\\GnuPGExch"
"gpgProgram"="C:\\Program Files\\GnuPGExch\\gpg.exe"

[HKEY_CURRENT_USER\Control Panel\Mingw32\NLS]
"MODir"="C:\\Program Files\\GnuPGExch\\Locale"
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

3. After you have the appropriate changes to gnupg-w32.reg, you can save and double click on it to complete the registry entries.

© SANS Institute 2003, Author retains full rights.

Installing GPG Shell Key Manager

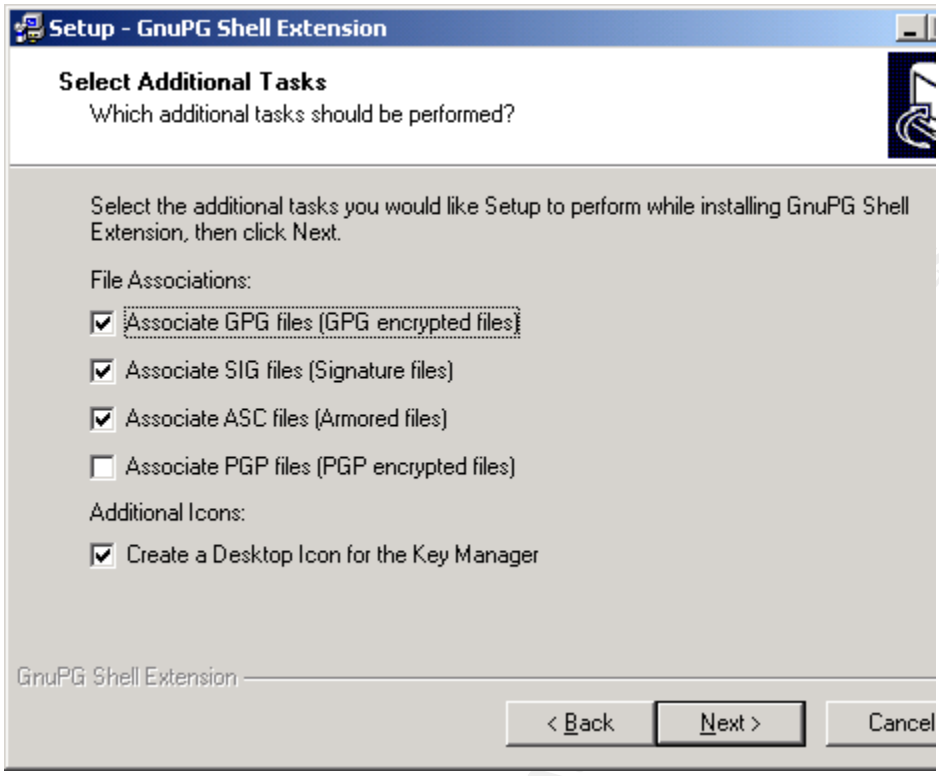
1. Now we can install GPG Shell extensions for key management and file encryption. First double click on the executable.




2. The default program directory is okay here, so just click next.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

3. Make sure that you select  create a desktop icon of the key manager option.

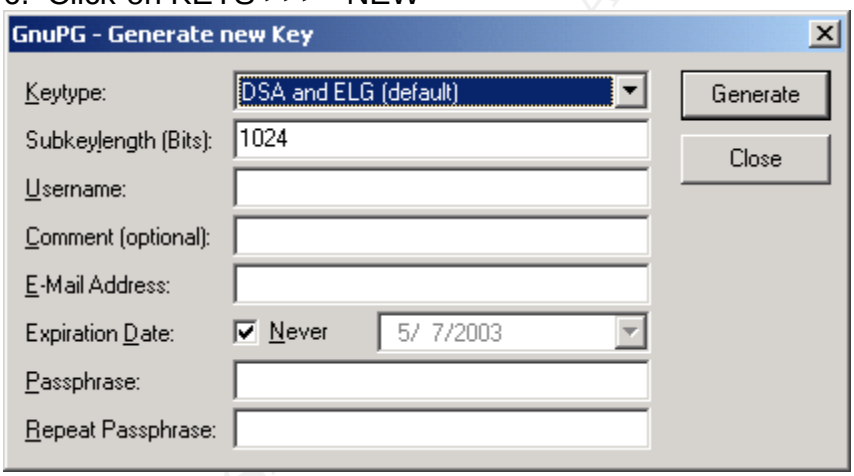



4. Click install then at the last dialog box, click finish

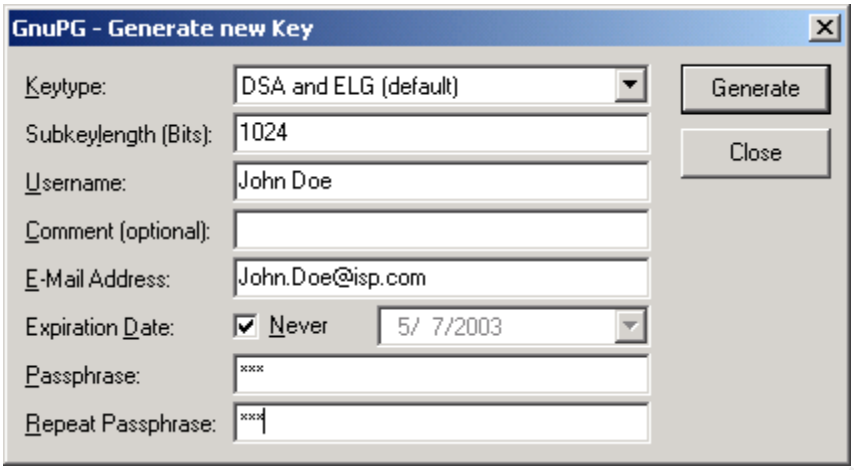
© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

5. Now double click on the Frog and create your key pair…

6. Click on KEYS >>> NEW




Author retains full rights



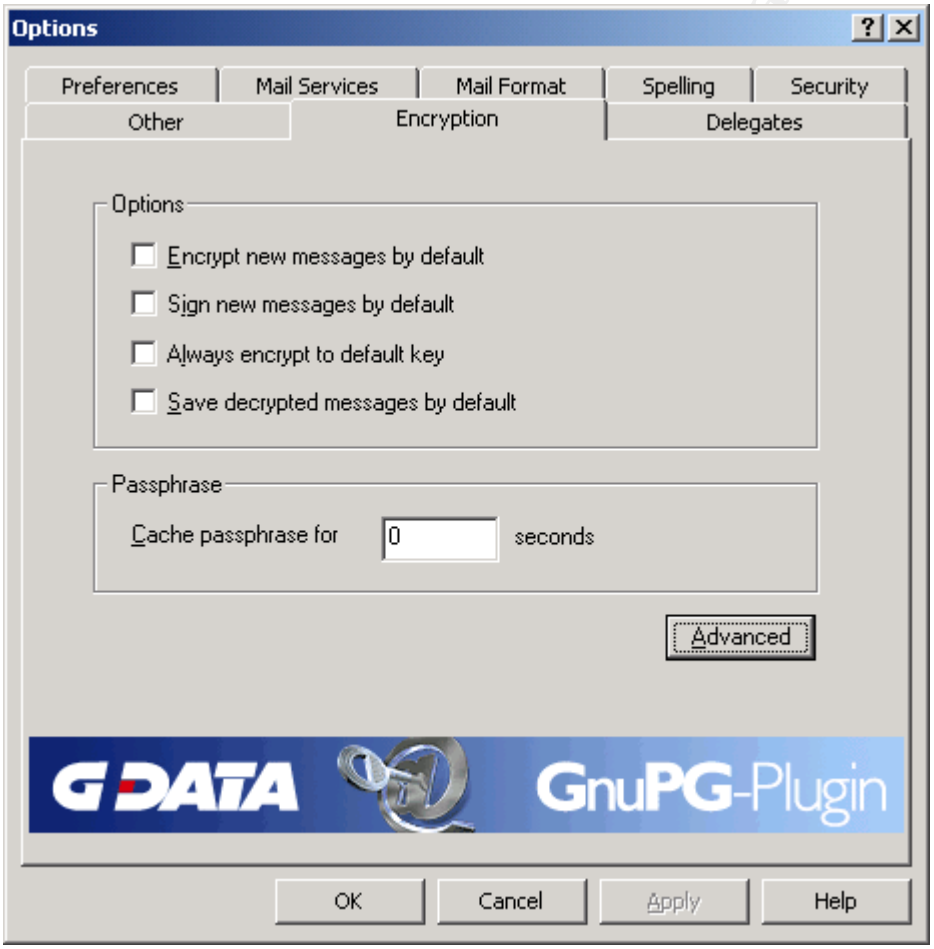

7. Fill in the appropriate information. Name, Email and pass phrase are required fields. Make sure that your Pass Phrase is COMPLEX and that you DON'T forget it!!! We will cover creating a revocation key and how to back up your keys AND how to travel with your keys later. YOUR Pass phrase and secret key must be protected, take great effort to make sure that no one sees your pass phrase!! Click on Generate, this will generate your Private and Public Key Pair! It asks to backup the key pair. I suggest that you skip this step at the present time .

© SANS Institute 2003, Author retains full rights

Final GNUPG Outlook Plug-in Configurations.

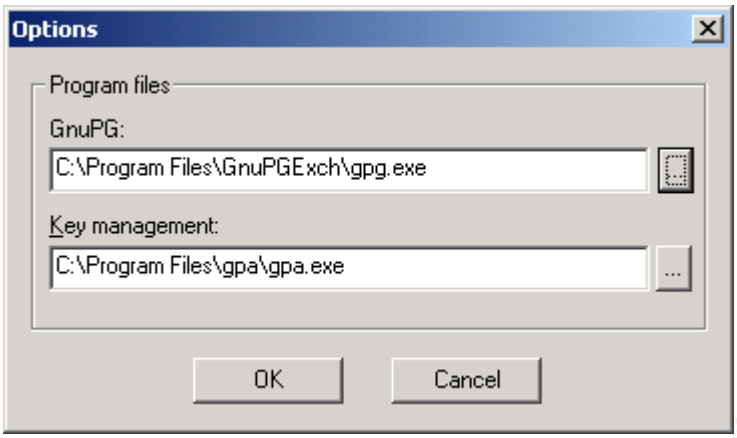
Before we start outlook and send some encrypted email, let's make sure that we are pointing to the correct path.

1. Click on Tools
2. Click on Options
3. Click on Encryption tab




4. Click the advanced button

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.




5. Make sure that the GnuPG is pointing to C:\Program Files\GnuPGExch\gpg.exe and then restart Outlook.

© SANS Institute 2003, Author retains full rights

The Encryption Process

In order to send anything encrypted to someone we first need his or her Public Key. How do we get it? Here are two ways:

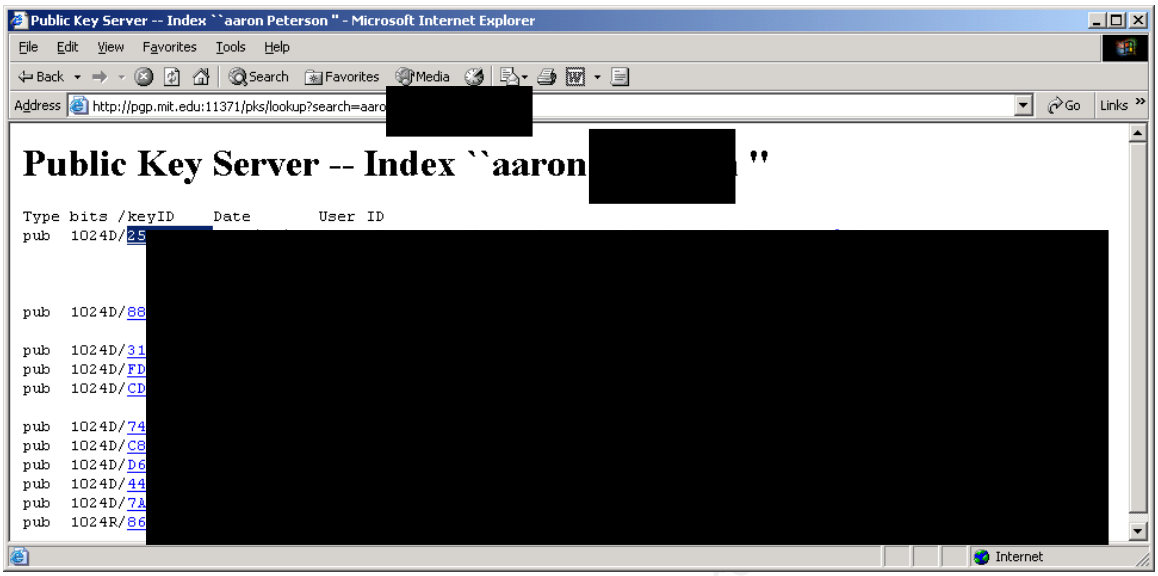
a. They could email it to us
b. We can download it.

Let's explore option b and download my friend Aaron's Key from a public key server.
1. Point your browser to http://pgp.mit.edu and search for Aaron. When you look for other people's keys you will probably want to further limit the search results by searching for a specific email address of the person.

2. Click on the "Do the search! Button"

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

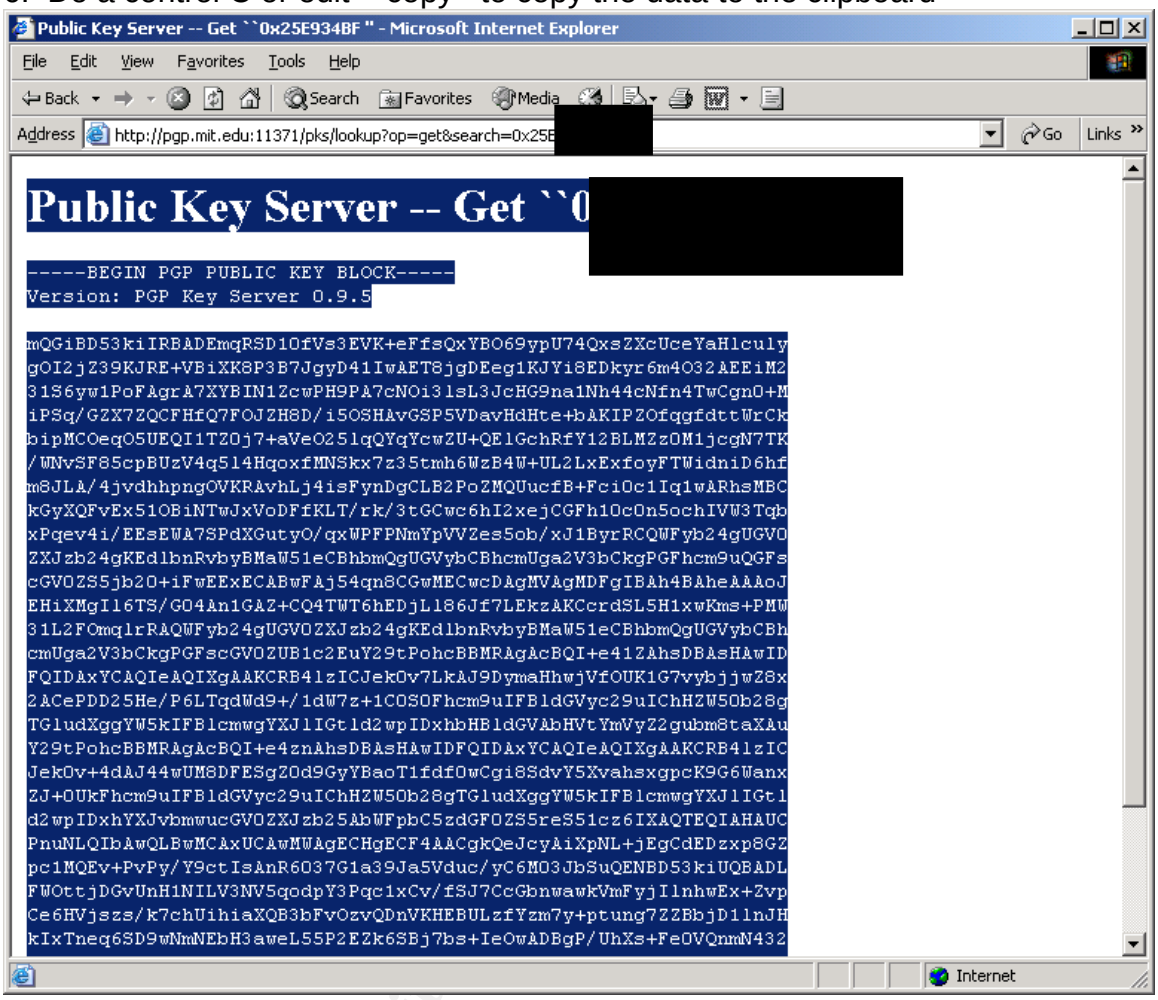
3. Pick the right person…. The key we are looking for is 25[redacted]F




© SANS Institute 2003, Author retains full rights.

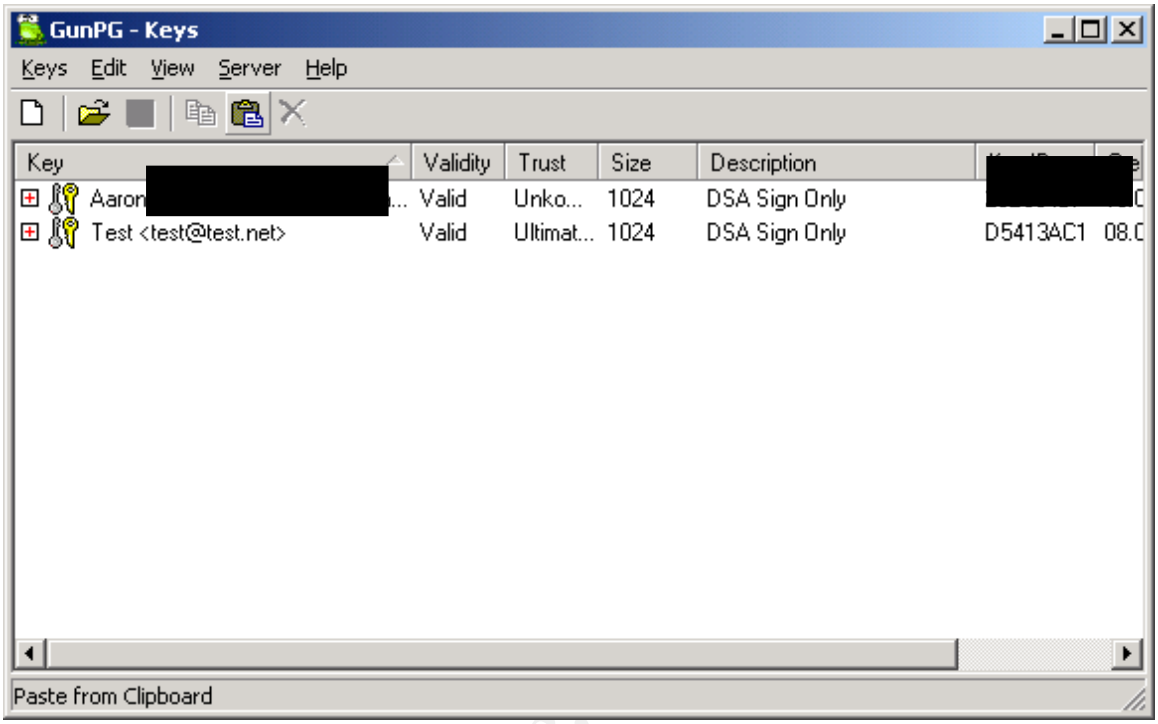
© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

4. Click on the link for his key 25E9???????
5. Do a control A or edit > select all to select all of the key that is listed
6. Do a control C or edit > copy to copy the data to the clipboard



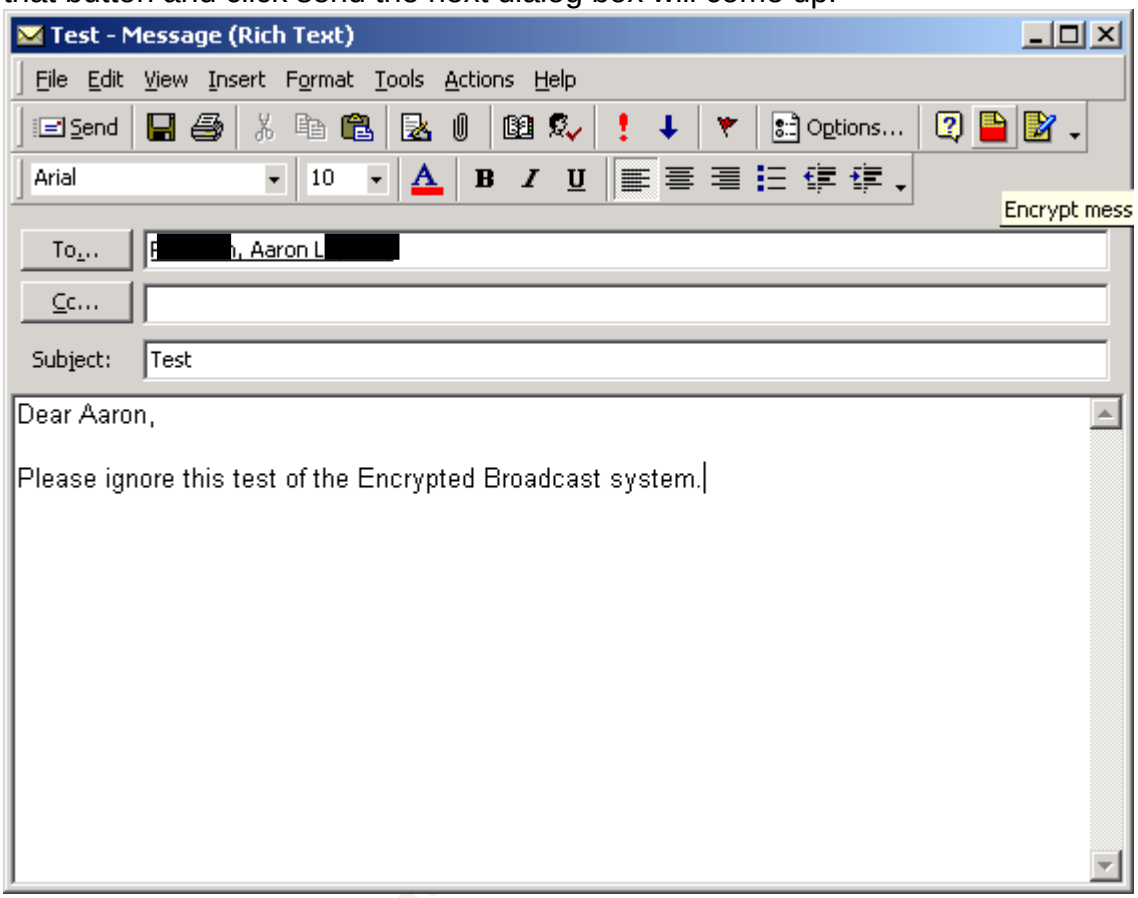

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

7. Now we will paste his key into our Key Manager (aka the Frog), click on the clipboard shortcut and it will add his public key to our Keyring



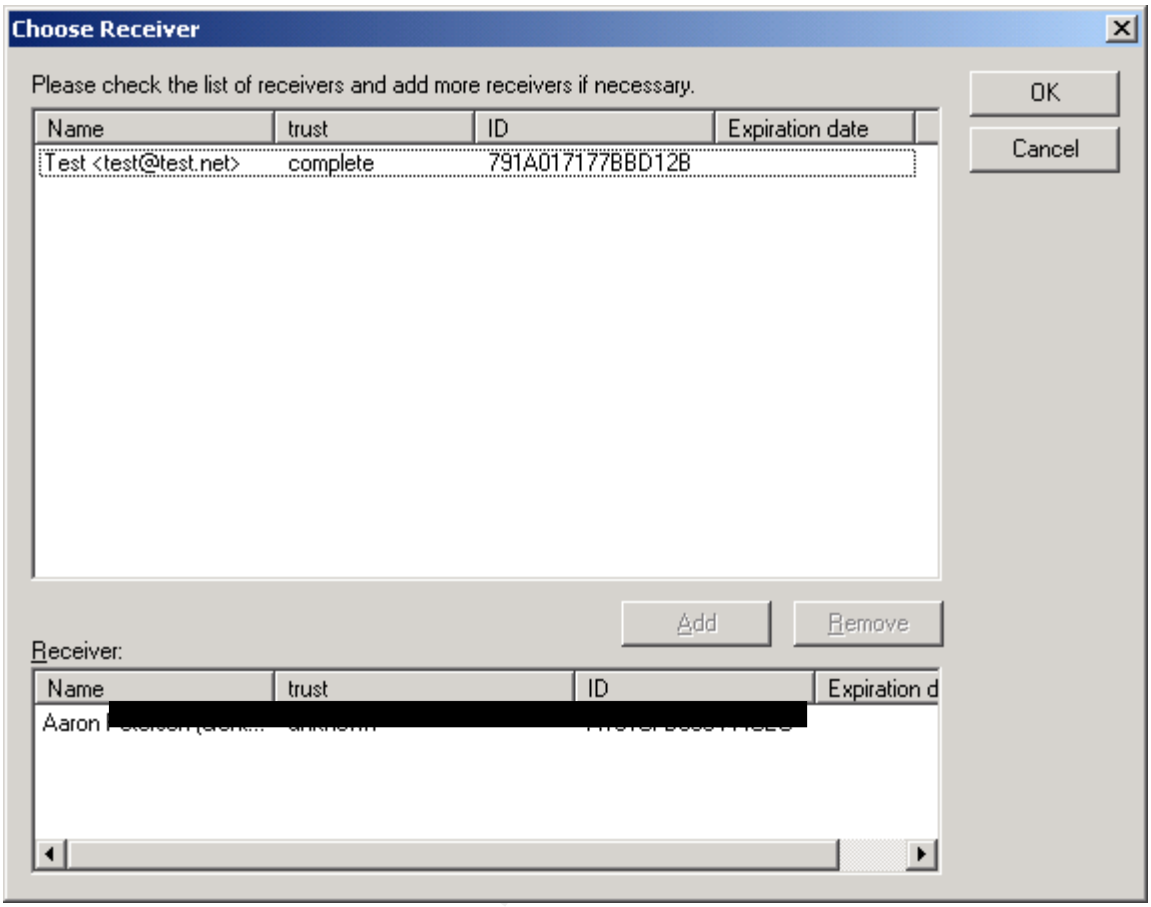

© SANS Institute 2003,

8. Start Outlook (one caveat that I have found is that you have to restart outlook in order to see NEW keys that you have added to your keyring) and create a new message as you normally would. Notice the Yellow and red Icon to the right of the options button, click that and it will encrypt the message before sending it. The other button with the pencil is for you to use to digitally sign your email. So once you click that button and click send the next dialog box will come up.




© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

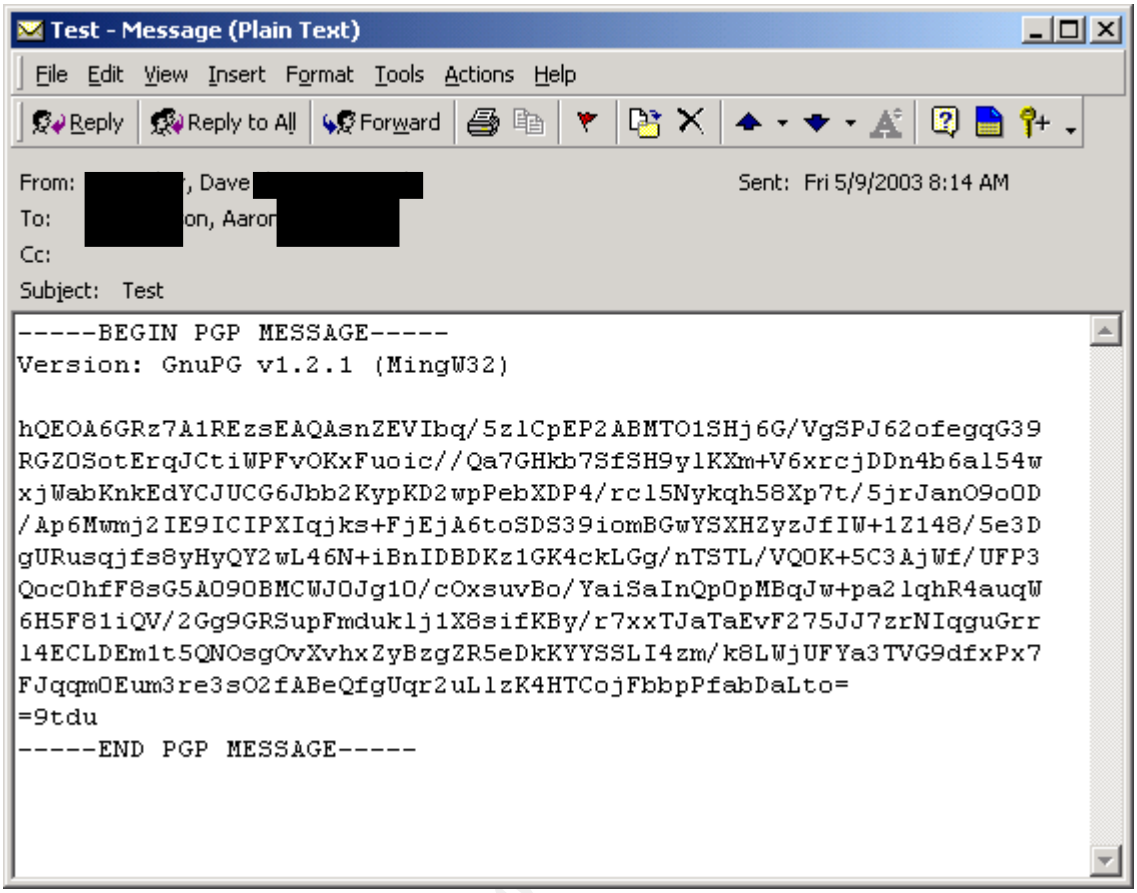
9. Now just double click on the key that you want to use to encrypt the message and say ok.




10. Click ok

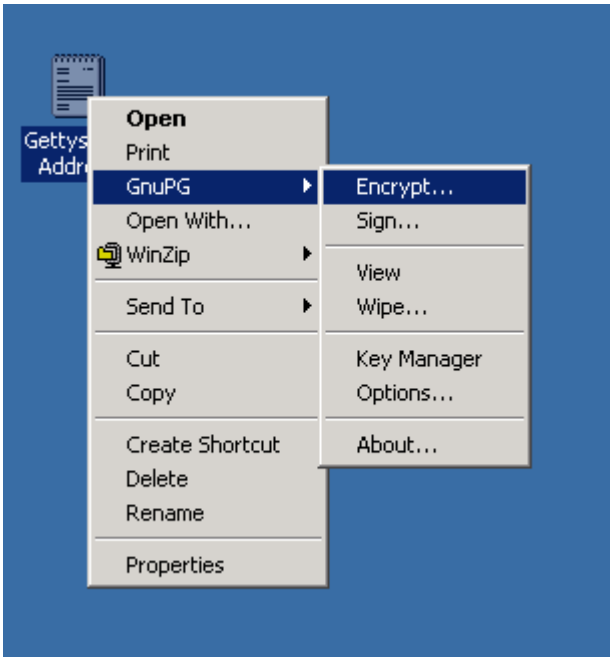
© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

If you look in sent items you will see that the message has become encrypted.




© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

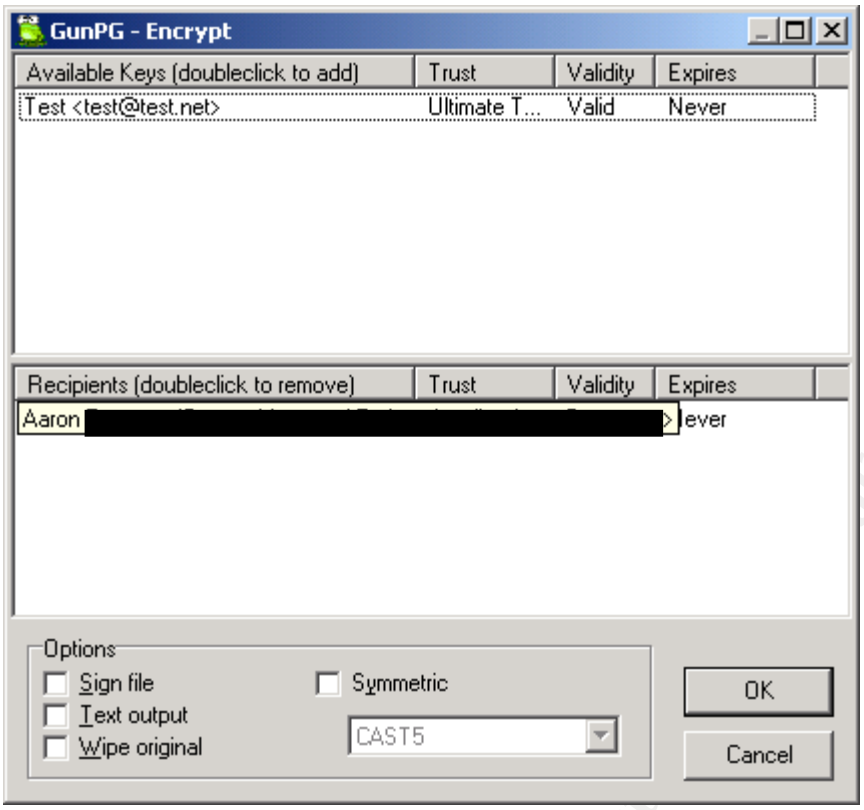
11. Now let's encrypt a file to send to our friend.. in this case we will right click on the Gettysburg Address on our desktop. We will select the GNUPG option, and follow the arrow over and left click on encrypt.



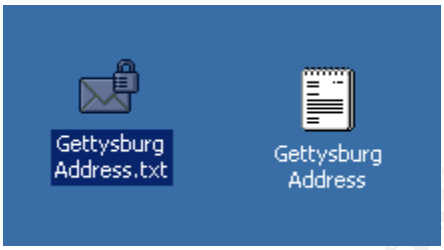

When we select the encrypt option it brings up the following dialog box::

Be patient while the GPG Shell program loads.. once it loads double click on the key you want to use to encrypt the file

© SANS Institute 2003, Author retains full rights

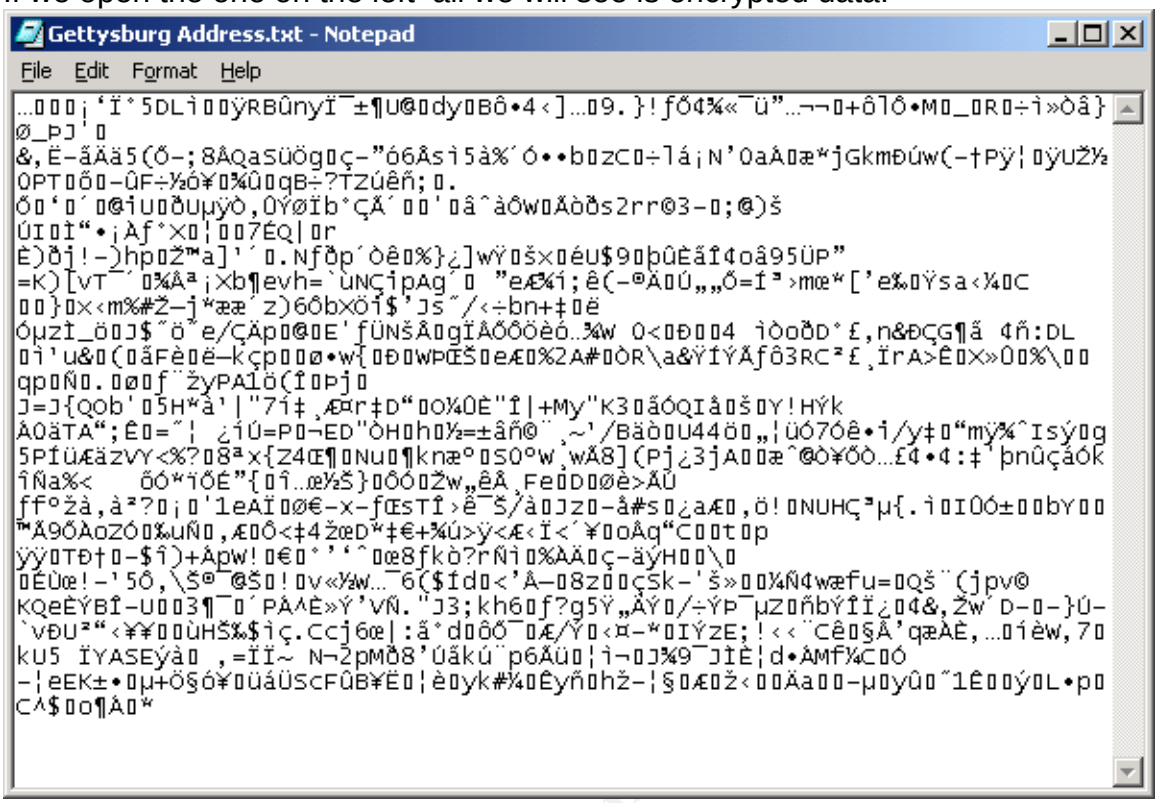



12. Select OK and it will create the following encrypted file with a different icon.




© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

If we open the one on the left  all we will see is encrypted data.




However our friend will be able to use his private key and pass phrase to decrypt and open the file.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

## Moving your Private KEY to another computer

As I mentioned earlier, take great care in maintaining your private encryption key, if someone gets it and your pass phrase then they can impersonate you and decrypt all of your mail and your files. Having said all of that, install the three applications that we use following the training manual until you get to the point where you create your private/public key pair. Do not create another public/private key pair! Use the GPG Shell program to export your private and public keys.

1. Change into this folder
c:\Program Files\GnuPGExch\
2. execute these commands:
3. gpg --import < a:\secring.gpg
4. gpg --import < a:\pubring.gpg

This will import your secret key and your public keyring.

## Creating a Revocation Key

Now that you are living dangerously by wanting gpg to travel around with you, you need to create a revocation key so that if your key gets compromised you can issue a revocation.

Make sure that you are in this folder:
c:\Program Files\GnuPGExch\
Issue this command where test= the user id that you gave your key.
1. gpg --gen-revoke test

HERE IS a SAMPLE of the Procedure…….
C:\Program Files\GnuPGExch>gpg --gen-revoke test

sec 1024D/D5413AC1 2003-05-08 Test <test@test.net>

Create a revocation certificate for this key? Y
Please select the reason for the revocation:
0 = No reason specified
1 = Key has been compromised
2 = Key is superseded
3 = Key is no longer used

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
  Q = Cancel
(Probably you want to select 1 here)
Your decision?
Your decision? 3
Enter an optional description; end it with an empty line:
> It's gone
>
Reason for revocation: Key is no longer used
It's gone
Is this okay? y

You need a passphrase to unlock the secret key for
user: "Test <test@test.net>"
1024-bit DSA key, ID D5413AC1, created 2003-05-08

ASCII armored output forced.
Revocation certificate created.

Please move it to a medium which you can hide away; if Mallory gets
access to this certificate he can use it to make your key unusable.
It is smart to print this certificate and store it away, just in case
your media become unreadable.  But have some caution:  The print system of
your machine might store the data in a temporary location where it could be accessible
by others.

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.2.1 (MingW32)
Comment: A revocation certificate should follow

iFIEIBECABIFAj6+30ULHQNJdCdzIGdvbmUACgkQuct/W9VBOsEZbQCeNLbsa557
LdG7EB4GE76iwdP190MAmwV845PaMUHhStHTfIDMunNc/s41
=fNMy
-----END PGP PUBLIC KEY BLOCK-----
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: May 21st, 2013

# Upcoming SANS Training

**Click Here for a full list of all Upcoming SANS Events by Location**

| | | | |
|---|---|---|---|
| SANS Malaysia @ MCMC 2013 | Cyberjaya, MY | Jun 03, 2013 - Jun 08, 2013 | Live Event |
| SANS Pen Test Berlin 2013 | Berlin, DE | Jun 03, 2013 - Jun 08, 2013 | Live Event |
| Industial Control Systems Security Training - Houston | Houston, TXUS | Jun 10, 2013 - Jun 15, 2013 | Live Event |
| Security Impact of IPv6 Summit 2013 | Washington, DCUS | Jun 14, 2013 - Jun 16, 2013 | Live Event |
| SANSFIRE 2013 | Washington, DCUS | Jun 14, 2013 - Jun 22, 2013 | Live Event |
| SANS Canberra 2013 | Canberra, AU | Jul 01, 2013 - Jul 13, 2013 | Live Event |
| Digital Forensics & Incident Response Summit 2013 | Austin, TXUS | Jul 09, 2013 - Jul 16, 2013 | Live Event |
| SANS London Summer 2013 | London, GB | Jul 09, 2013 - Jul 16, 2013 | Live Event |
| SANS Rocky Mountain 2013 | Denver, COUS | Jul 14, 2013 - Jul 20, 2013 | Live Event |
| SANS Mumbai 2013 | Mumbai, IN | Jul 22, 2013 - Jul 27, 2013 | Live Event |
| SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification | Washington, DCUS | Jul 22, 2013 - Jul 26, 2013 | Live Event |
| SANS San Francisco 2013 | San Francisco, CAUS | Jul 29, 2013 - Aug 03, 2013 | Live Event |
| SANS SEC 560: Network Penetration Testing @ Bangalore 2013 | Bangalore, IN | Aug 05, 2013 - Aug 10, 2013 | Live Event |
| SANS Boston 2013 | Boston, MAUS | Aug 05, 2013 - Aug 10, 2013 | Live Event |
| Critical Security Controls Summit | Washington, DCUS | Aug 12, 2013 - Aug 18, 2013 | Live Event |
| Industrial Control Systems Security Training - DC | Washington, DCUS | Aug 12, 2013 - Aug 16, 2013 | Live Event |
| SANS Thailand 2013 | Bangkok, TH | Aug 19, 2013 - Aug 31, 2013 | Live Event |
| SANS Virginia Beach 2013 | Virginia Beach, VAUS | Aug 19, 2013 - Aug 30, 2013 | Live Event |
| Mobile Device Security Summit 2013 | OnlineCAUS | May 30, 2013 - Jun 06, 2013 | Live Event |
| SANS OnDemand | Books & MP3s OnlyUS | Anytime | Self Paced |