access

GPG GUIDE FOR SECURE COMMUNICATIONS

October 2011

/// GPG GUIDE FOR SECURE ///// /// COMMUNICATIONS ////////////

For more information, please contact:

soc@accessnow.org

PGP Key ID: 0xF08D380A

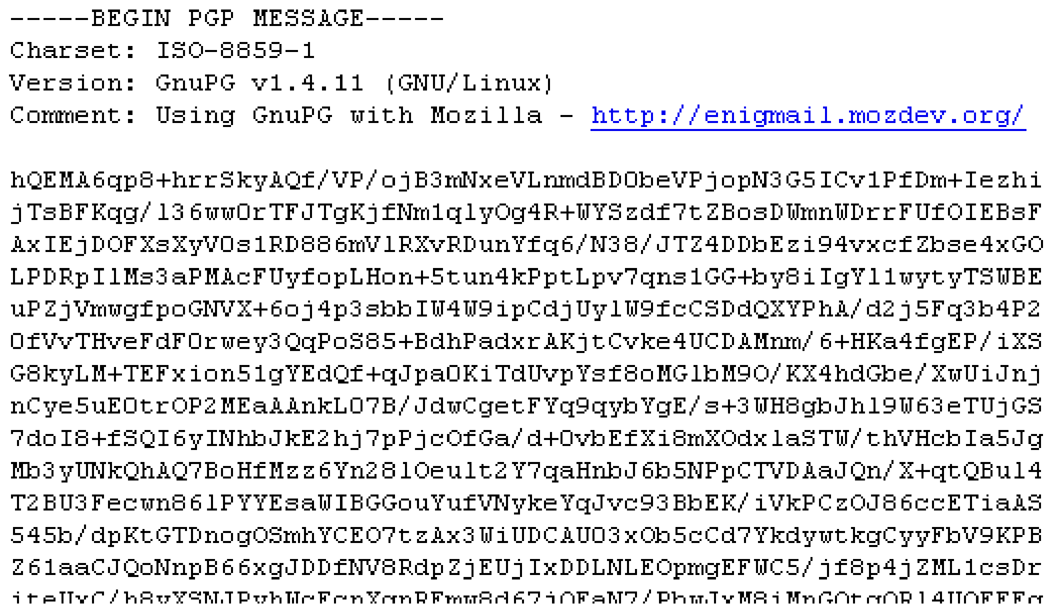
WHY YOU NEED TO USE ENCRYPTION IN GENERAL AND GPG IN PARTICULAR

Encrypting data and communications performs many functions that improve security. First, it ensures that information sent over an insecure network such as the Internet or telephone networks, cannot be read if it is intercepted (confidentiality). Second, it ensures that you have a way to verify that a piece of information that appears to come from a trusted contact has in fact been sent by that contact (authentication) and has not been tampered with (integrity). For human rights workers placed in hostile environments, both functions are vital to protect the lives and safety of not only the workers themselves but those of their contacts, allies, friends, aides, and supporters. It may seem far-fetched that insecure communications could pose a greater risk than physical presence at a rally or crisis point — but discussing plans over insecure communications may make it easy for NGO workers to be targeted when future locations are known.

Over the last five years, many media outlets have reported on the increasing risks for humanitarian workers providing help in crisis situations. Communications are not the only area where attention to security is necessary, but they are an important one.

GPG — for GNU Privacy Guard — is a well-respected, established, thoroughly studied piece of encryption software. Based on PGP (for Pretty Good Privacy), it uses the techniques of public key cryptography (see below in this guide) to provide both confidentiality and authentication. Versions are available for all of the most common desktop/laptop operating systems (including Windows, Mac, and Linux), and versions are either available or being written for Webmail access via the Firefox and Chrome browsers and for the iPhone, iPad, and Android smart mobile devices. Although its primary use is for email, it is also an excellent cryptosystem for other applications such as mailing lists, and files and filesystems.

This guide explains the installation and use of GPG and the basics of public key cryptography.



```
-----BEGIN PGP MESSAGE-----
Charset: ISO-8859-1
Version: GnuPG v1.4.11 (GNU/Linux)
Comment: Using GnuPG with Mozilla - http://enigmail.mozdev.org/

hQEMA6qp8+hrrSkyAQf/VP/ojB3mNxeVLnmdBDObeVPjopN3G5ICv1PfDm+Iezhi
jTsBFKqg/l36wwOrTFJTgKjfNm1qlyOg4R+WYSzdf7tZBosDWmnWDrrFUfOIEBsF
AxIEjDOFXsXyVOs1RD886mVlRXvRDunYfq6/N38/JTZ4DDbEzi94vxcfZbse4xGO
LPDRpIlMs3aPMAcFUyfopLHon+5tun4kPptLpv7qns1GG+by8iIgYl1wytyTSWBE
uPZjVmwgfpoGNVX+6oj4p3sbbIW4W9ipCdjUylW9fcCSDdQXYPhA/d2j5Fq3b4P2
OfVvTHveFdFOrwey3QqPoS85+BdhPadxrAKjtCvke4UCDAMnm/6+HKa4fgEP/iXS
G8kyLM+TEFxion51gYEdQf+qJpaOKiTdUvpYsf8oMGlbM9O/KX4hdGbe/XwUiJnj
nCye5uEOtrOP2MEaAAnkLO7B/JdwCgetFYq9qybYgE/s+3WH8gbJhl9W63eTUjGS
7doI8+fSQI6yINhbJkE2hj7pPjcOfGa/d+OvbEfXi8mXOdxlaSTW/thVHcbIa5Jg
Mb3yUNkQhAQ7BoHfMzz6Yn28lOeult2Y7qaHnbJ6b5NPpCTVDAaJQn/X+qtQBul4
T2BU3Fecwn86lPYYEsaWIBGGouYufVNykeYqJvc93BbEK/iVkPCzOJ86ccETiaAS
545b/dpKtGTDnogOSmhYCEO7tzAx3WiUDCAUO3xOb5cCd7YkdywtkgCyyFbV9KPB
Z61aaCJQoNnpB66xgJDDfNV8RdpZjEUjIxDDLNLEOpmgEFWC5/jf8p4jZML1csDr
iteUxC/h8vXSNJPvhWcFcnXqnREmw8d67jOFaN7/PhwJxM8iMnGOtqOR14UOEEFq
```

Message text after encryption with GPG.

GOOD PRACTICE SUMMARY

Make encrypting all your communications a habit. That way, you won't forget at a moment when it really matters. In addition, if you only encrypt a percentage of your messages, you will make those messages a target for surveillance efforts.

Keep your private key secret. If it is compromised, revoke the key and generate a new one. Keep the passphrase protecting your private key secret and change it and revoke your keys as soon as possible should it become compromised.

Publish your public key as widely as possible, and get it signed by as many people as possible using correct signing procedures (the person signing the key should verify both your identity and the integrity of the key). To this end, whenever you expect to meet the members of other NGOs than your own, take advantage of the meeting to sign each others' keys.

Verify the identity of your correspondents by downloading their public keys from known keyservers and checking the signatures.

BASIC INSTALLATION

You will follow the same basic installation steps on all platforms:

» Download GPG for your platform plus any additional software you may need;

» Unpack and install GPG and the additional software;

» Run GPG and generate a key pair (one public, one private);

» Select a passphrase to protect your private key, which should be easy to remember and convenient to type but hard to guess;

» Set any additional options available in your software installation;

» Export your public key in ASCII armored format and get someone who can verify your identity to sign it to authenticate it as yours;

» Upload your signed public key to one or more keyservers;

» Make a note of your key's expiration date and set a reminder to extend or regenerate it in advance;

» Create a revocation certificate and store it somewhere absolutely safe so you can upload it immediately if your private key is compromised (for example, the device it's stored on is lost or stolen) or you forget your passphrase. Always remove it from any device where you store your private keys.

See the end of this guide for installation instructions for specific platforms.

GPG IN USE

Note for Windows users: Some of the instructions below require you to open a command line. If you have installed GPG's Kleopatra module or another graphical front end, you can perform these operations via that route.

In XP, you can do this by clicking on the Start menu, choosing Run, and typing in

```
command.com
```

Alternatively, choose Programs | Accessories | Command Prompt.

As installed, Vista and 7 do not include the Run option on the Start menu; you will need to enable it. To do so, click on the Start button and choose Properties, then the Start Menu tab, and then Customize. You will get a new window with a lengthy list of options. Scroll down to find Run Command, and check the box to enable it. Click OK, then Apply. Run should now appear on the Start menu and you can now type in

```
command.com
```

as in XP.

Note for Mac OS X users: Some of the instructions below require you to open a command line. To do this, go to Applications | Utilities | Terminal.

Once you have opened a command line (also known as a "console") the commands are in the same on all platforms.

CHOOSING A PASSPHRASE

The choice of a good, hard-to-crack passphrase is crucial because it is the weakest link in the GPG security system. Anyone with your passphrase and your private key can gain access to all your communications and – even worse – can impersonate you to all your trusted contacts. Passphrases should be easy to type (because you will be using it a lot), easy to remember, and hard to guess. While it's generally acceptable to reuse passwords on sites of low importance (for example, the registration systems on news sites, your private key is the most sensitive piece of data you will ever be asked to protect. You should choose a passphrase that is unique and you should guard it with exceptional care. If there is ever any chance that your passphrase has been compromised, you should revoke your keys as soon as possible. Then, as soon as you can get to a secure situation (trusted network, trusted system, trusted keyboard), generate new ones with a new passphrase.

For your passphrase, do not use your name, your NGO's name, any portion of your or your NGO's street address, or the names of your pets, spouse, children, schools, or any other piece of information about you that has been published on the Net (even in a supposedly private area such as a locked Facebook page). Also do not use single words you can find in a dictionary of any language; these are easily cracked.

Many textbooks will tell you the safest passwords are those that have been randomly generated or conform to rules specifying a minimum length and at least one capital letter and one number. Passwords created by these methods are hard to remember accurately and hard to type; many people, faced with having to remember even just one random 22-character string will write it down, a very insecure practice.

A more usable strategy is to create a memorable sentence; this will naturally have capitals, spaces, and punctuation and be very hard to guess.

If your passphrase is ever compromised (that is, becomes known to another person), revoke your key, generate a new one, secure it with a completely new passphrase, and then issue warnings as appropriate given that you now must assume that all the communications sent and received using your old key are no longer secure.

If you ever need to change your passphrase – for example, because you find yours hard to type or you want to make it more secure.

Open a command line and type the command

```
gpg --edit-key <mykey>
```

replacing <mykey> with either your key's numbered ID or a portion of the email address associated with it.

At the command prompt type

```
passwd
```

You will be prompted to enter your old passphrase and then to type your new passphrase twice (once to enter, once to confirm). At the command prompt type

```
save
```

Remember to update any backups or additional installations after you do this so they all are configured with the new passphrase.

EMAIL

You should make encrypting all your email a habit, for two reasons. First, if you encrypt only sensitive messages, those messages will immediately stand out as targets for investigation to anyone monitoring your communications. Second, if you encrypt selectively there is a good chance that at some point you will forget to encrypt something significant.

Fortunately, setting your email software to encrypt all your email can be as simple as ticking a box on an options screen.

For Thunderbird (Windows and GNU/Linux), you will need to download and install the Enigmail plug-in. Once you have done this, Thunderbird will add an OpenPGP menu from which you can set preferences and operate most GPG functions. You will need to use GPG's Kleopatra module, a command line, or another Windows front end to generate, edit, or revoke keys or to change your passphrase.

For Outlook (Windows) — version 2003 SP2 or later or 2007 only — you'll need the GpgOL plug-in, which Gpg4win will install for you. The next time you open Outlook, Extras | Options should have a GpgOL tab where you can configure GPG. You should choose to enable S/MIME support and turn on encryption and message signing by default.

WEBMAIL

Until the projects to create PGP/GPG plug-ins for Firefox and Chrome release their first versions, there is no good option for using GPG encryption with services that can only be operated via a Web browser.

If the system allows secured connections via TLS with IMAP and/or POP3 (as Gmail and Hotmail do), however, all your interaction with your email can take place on your local machine in a standard mail client (such as Thunderbird). In such a set-up, you will download all your email to your machine to decrypt and read, and then write and encrypt the replies before uploading them to send. You will not be able to read the encrypted messages or verify the signatures from your Web browser.

For a Webmail system that does not allow such a set-up, it might be possible to create a filter that automatically forwards the email to a second account that can support PGP/GPG. It's hard to see the benefits of such a set-up (since your keys are tied to your email address). As many of these services also block forwarding, we would recommend you choose a different service for encrypted mail.

FILES AND FILESYSTEMS

GPG can be used to encrypt selected individual files or whole filesystems.

On any platform, use the command

```
gpg -c <filename>
```

You will be prompted to type a password twice (once to enter, once to confirm). This password does not have to be the passphrase that protects your secret keys. Do not lose the password or you will not be able to retrieve the file's contents. Delete the unencrypted version of the file.

To decrypt the file use the command

```
gpg <filename.gpg>
```

You will be prompted to enter the password you used when you encrypted the file.

In Windows, either right-click on one or more filenames in Windows Explorer and choose Encrypt from the resulting pop-up menu or open Kleopatra, choose File | Encrypt, and then browse to the correct files from the dialog box. You'll find the decrypt command on the same menus.

If you wish to encrypt your entire hard drive, although you can do this with GPG it's not really the purpose the software was primarily designed for. We would suggest instead using Truecrypt (www.truecrypt.org) or, on a Mac, the built-in hard disk encryption.

ENCRYPTING FILES FOR CONFIDENTIALITY

One of the beauties of GPG is that you can encrypt a file so that only the person you designate can open and read it. To do this, you will encrypt the file using the intended recipient's public key, typing in this command and replacing Bob with a portion of the recipient's email address and my-file.txt with the file you wish to encrypt:

```
gpg --recipient <Bob> --encrypt <my-file.txt>
```

If you want to paste the entire file into a plain text email to send it to the recipient, you will need to add the ASCII armor option:

```
gpg --recipient Bob --armor --output your-file.asc
--encrypt my-file.txt
```

Bob will decrypt the file by using this command:

```
gpg --decrypt <the-file.gpg>
```

MAILING LISTS

The GPG cryptosystem can support both general-purpose, one-to-one email and mailing lists. To run an encrypted mailing list, you will need to install a GPG-enabled listserver. Available implementations include Schleuder (http://schleuder2.nadir.org/), GPG-Ezmlm (http://www.synacklabs.net/projects/crypt-ml/), and RedIRIS. It is also possible to modify the GNU Mailman (www.list.org) listserver to provide GPG capability (http://non-gnu.uvt.nl/mailman-pgp-smime/).

MANAGING KEYS

SECURITY

Keep your private key secure. Nobody but you should have access to it.

GENERATING KEYS

When you generate your first keypair — one public, one private — you should take the default options. However, it's helpful to know a little more about how these keys work and what your overall choices are. To understand those, it's helpful to know a little about cryptography in general.

For its strength — that is, the difficulty of breaking the code — a cryptosystem relies on three elements: 1) the robustness of the algorithm, or method, by which the cleartext is encrypted; 2) the length of the key the algorithm uses to encrypt the message; and 3) the user's care in operating the system correctly. For a physical world analogy, the algorithm is the type of lock — cylinder, for example, or lever — and the key is the key that opens a particular lock. Assuming that the algorithm is resistant to cracking — and the algorithms GPG uses have resisted cracking attempts by experts — the longer the key the longer it will take to decode a particular message by repeated attempts at guessing the key ("brute force"). The reason has to do with probabilities: the longer the key the less likely it is that a brute force attempt will hit on the right one.

There is a trade-off, however: the longer the key the longer encrypting and decrypting take. GPG will not allow you to create a key less than 768 bits long. For the moment, 1024 bits is a reasonable compromise; the safe length will increase as readily available processing power continues to increase.

The command to generate a key (see under "GPG In Use" for instructions on opening a command prompt) is

```
gpg --gen-key
```

GPG will offer choices of several types of keys - you should accept the default, which is DSA and ElGamal. What happens when you do is that GPG generates not one but two keypairs, of which one is the primary (or master) keypair you'll use for making signatures and second one is a subordinate keypair, or subkey, used only for encrypting messages and other data. You can add more subkeys at a later date if you need to.

For more on adding subkeys and extra user IDs, see the GNU Privacy Guard manual, in the section titled "Adding and deleting key components."

EXCHANGING KEYS

When you connect to a public keyserver to upload your key or download or verify a contact's key - and you should – that connection (via the hkp protocol) is in the clear. This opens the way to two risks: first, that your network operator can monitor your traffic to see who you intend to communicate with; second, that a man-in-the-middle attack could substitute a false key for the real one. The risk of this type of attack is therefore something you should be aware of.

Wherever possible, confirm with your contacts that you have their genuine keys by comparing the fingerprints.

This is a problem even if your installation of Thunderbird (using Enigmail) is set to use a secure proxy, such as Tor (www.torproject.org. One workaround is to use a local http proxy, for example Polipo (http://www.pps.jussieu.fr/~jch/software/polipo/). Once you've downloaded and installed it, add these two lines to your Thunderbird configuration:

```
socksParentProxy = "localhost:9050"
socksProxyType = socks5
```

You then need to add this option to the GPG command line (or set it in Enigmail's advanced preferences):

```
--keyserver-options http-proxy=http://
localhost:8123
```

KEYS AND EXPIRATION DATES

When you generate your keys, GPG sets an expiry date. Make a note of that date and make sure to keep track of it so you can extend or regenerate your keys in good time. To do this, open a command prompt and type

```
gpg --edit-key <mykey>
```

Replace <mykey> with either the key's numbered ID or a portion of your email address. GPG should find the key, list its parameters, and give you a new command prompt.

Type

```
expire
```

and it will prompt you for the details of the new expiry date.

If you miss the date and your keys expire it is still possible to extend them but the procedure is much more complicated.

GPG will offer you the option of setting your keys to never expire. However, this is not recommended, for a variety of reasons. Setting a near-term expiry date can, for example, act as a backup in a situation where you are unable to upload the revocation certificate.

You will need to renew each subkey separately, and subkeys created since your original installation will have different expiration dates.

In this example, from the GNU Privacy Guard manual, Chloe has two user IDs, one public key, and three subkeys. The primary key and one of the subkeys will never expire, but two of the subkeys have expiration dates:

```
chloe% gpg --edit-key chloe@cyb.org
Secret key is available.
pub  1024D/26B6AAE1  created: 1999-06-15 expires: never
trust: -/u
sub  2048g/0CF8CB7A  created: 1999-06-15 expires: never
sub  1792G/08224617  created: 1999-06-15 expires: 2002-06-14
sub   960D/B1F423E7  created: 1999-06-15 expires: 2002-06-14
(1)  Chloe (Jester) <chloe@cyb.org>
(2)  Chloe (Plebian) <chloe@tel.net>
Command>
```

Type

```
toggle
```

at the command prompt to see a similar information display, but for your private keys.

REVOKING KEYS

If a computer or smart phone (or any other device such as a backup disk or USB stick) containing your private key is lost or stolen, you should revoke your key as soon as possible and generate new ones. Assuming you followed the instructions above and created a revocation certificate when you installed GPG, all you need to do now is upload it to the public keyservers. If you didn't, assuming you still have access to your private key, to revoke a key, use the command

```
gpg --gen-revoke <mykey>
```

You will be offered the opportunity to provide a reason (such as "key compromised") and will have to type in your passphrase. For <mykey> type in either the Key ID number or some part of the user ID that identifies the keypair.

In return, you will be given an ASCII-armored block which you upload to the public keyservers. If you are generating the revocation certificate at the time of installation, keep the certificate safe somewhere — if necessary, print it out and store it in a safe deposit box — as anyone can publish the revocation certificate, after which your key cannot be used for new communications.

After revocation, you can still use your key to decrypt past communications. However, you should (as much as is possible) notify your past correspondents that your key has been revoked and point them to the location where they can retrieve your new public key.

Guard the revocation certificate with great care (because anyone can upload it to a public key server and render your key useless) and store it separately from your private key so that they are not compromised simultaneously. Good storage options include a safely stored print-out (it's short), CD, or USB drive. Delete the revocation certificate from the GPG folder of the machine on which your keys were generated.

You can also revoke subkeys individually using the "revkey" command. Instructions on how to do this and some considerations to keep in mind can be found in the "Revoking key components" section of the GNU Privacy Guard manual.

GPG FEATURES

CONFIDENTIALITY

Confidentiality ensures that only the intended recipient can read the contents of the message. Confidentiality is achieved in GPG by the following process.

Alice writes an email to Bob on her workstation. Also locally, Alice's workstation's GPG installation randomly generates a one-time symmetric key and uses it to encrypt the message using a fast symmetric key algorithm. GPG then encrypts the symmetric key is encrypted using Bob's public key and added to the message. This sounds convoluted, but the point is to combine the benefits of the public key infrastructure (PKI) with the speed of symmetric cryptography: the second cryptographic part of the process is computationally much more intense, but since the symmetric key is small this is manageable. When Bob receives the message he uses his private key to decrypt the symmetric key and then uses the symmetric key to decrypt the message body. As all the encryption and decryption takes place locally on the originating and destination computers no middle man is required to keep the message or key secret (and therefore no man-in-the-middle attack is possible).

AUTHENTICATION

Authentication is a function by which Alice can verify that multiple emails from "Bob" are in fact from "Bob", whether or not she knows "Bob"'s true, real-world identity. Note that GPG has no way of binding a particular key to a particular real-world identity; it can only verify that each email was encrypted with the same private key, which is presumed to be the same individual. Authentication is achieved in GPG by the following process.

Bob creates the message he wants to send Alice, and uses a one-way cryptographic function to generate a "hash" (see glossary) value, or digest, of the message. He then encrypts the hash value using his private key. Alice (or anyone else) can verify Bob's signature by using the sender's public key to decrypt the hash value, and then running the email content though the same mathematical one-way function to produce a second hash and then comparing the two. If the hashes match Alice can be sure Bob sent exactly that message because Bob's public key was able to decrypt the hash value and the hash value proves that the signature belongs to this particular message.

IDENTIFICATION

Identification means being able to verify that messages to or from a particular email address are going to or from a specific identifiable real-world human being. Identification is achieved in GPG by one of two processes.

The first is the web of trust created by the practice of signing public keys. If enough people have signed a particular public key, the binding between a particular public key and a particular email address becomes quite strong. For this to work Alice must upload her public key to a public keyserver and take the time and care to build up the web of trust around herself by proving her identity to as many others as possible and getting them to cryptographically acknowledge that by signing her public key. We recommend that NGO members take advantage of every meeting to verify and sign each other's keys.

The alternative is for Alice and Bob to meet in person, and inspect each other's identification documents, then use the GPG cryptosystem to verify that each individual is in possession of the private keys that cryptographically match the public keys known to be associated with the email address in question.

INTEGRITY

Integrity means being able to verify that the content of an unencrypted email from a sender has not been modified in transit by a third party. Integrity is achieved in GPG by the following the same process as above for authentication, which simultaneously proves that the message content, the unique input to the one-way mathematical hash function, cannot have been modified in transit. Otherwise, the two hashes would not match.

NON-REPUDIATION

Non-repudiation means a sender cannot later deny authorship of a message or making a statement or promise therein. Non-repudiation is achieved in GPG by the following process.

Alice is trying to claim that she did not send a message Bob believes came from her. Bob verifies the message signature with her public key and checks that the public key is valid for that particular message. If both those conditions are met, then Alice must be the source since only she could have successfully signed that particular message coming from that particular email address using that specific private/public key pair.

TRUST

The public key infrastructure (PKI) created by individuals using the GPG cryptosystem confers trust by making it explicit to third parties that one or more individuals within the system trust(s) another individual. What follows explains the processes in GPG of conferring and inferring trust.

If Alice trusts Bob and Carol trusts Alice, Alice can confer trust upon Bob so that Carol, by extension, also trusts him. The process begins when Bob uploads his public key to a public keyserver. Alice first verifies Bob's identity. She may do this by meeting him in person and checking their identity documents, but often her verification will be based on a long history of interaction between the two individuals. Next, Alice uses her private key to sign Bob's public key and then uploads the signed key to the keyserver. Now, anyone who cares to check can prove, using Alice's public GPG key, that she signed Bob's public key, which implies that Alice trusts Bob.

For Carol to infer that Bob is likely to be who he claims to be and someone she can trust, Bob first must have uploaded his public key to a public keyserver and have taken the time and care to build up the web of trust around his public key by encouraging others to sign his public key and upload it back to the keyserver. Carol can infer some measure of trustworthiness just from the GPG cryptosystem before Bob reveals anything to her in a message: Carol can examine the list of who has conferred trust to the individual by signing his public key. She can verify the authenticity of the conferred trust, but she still must decide whether these relationships are enough to give her confidence that Bob is really who he says he is. This is not just a game of numbers; much of Carol's eventual decision whether or not to trust Bob will rely on the identities of the people who have signed his key. In many cases this will be a personal decision; alternatively or in addition, Carol can use trust metrics built into the GPG cryptosystem to get a numerical rating of the system's confidence that Bob's identity is as claimed.

CRYPTOGRAPHIC PROPERTY	ENCRYPT / DECRYPT	SIGN / VERIFY	PUBLIC SERVER
AUTHENTICATION	—	YES	—
INTEGRITY	—	YES	—
NON-REPUDIATION	—	YES	—
CONFIDENTIALITY	YES	—	—
CONFER TRUST	YES*	YES	YES
INFER TRUST	YES*	YES	YES

** While it's not absolutely essential to use encryption in the process of signing someone's public key or to communicate with someone you should use it as a step to ensure that the person in control of the email account is the same as the person with the passphrase protecting the GPG private key associated with that email address.*

SECURITY POLICIES

EMAIL

People think of their email as private conversations between themselves and one or more other individuals. Yet they rarely take steps to enforce that privacy, and the first question usually asked in relation to GPG is, "Why encrypt email?".

The fact is that email as generally implemented today is the digital equivalent of postcards. It has been widely held among security experts since the early 1990s that Internet society should use encryption by default; encrypting an email message is the virtual equivalent of what we do in the world of physical mail when we enclose letters in envelopes.

To many users, email protocols seem direct: the sender's client connects to its outgoing SMTP (simple Mail Transfer Protocol) server, that server connects directly to the recipient's incoming mail server, and the receiver's client connects directly to that same incoming mail server. That's just three connections, all direct, right? Wrong. The fact is that in order to make those three transfers the mail traffic will typically be routed to and pass through perhaps a dozen or more computers. Any person with administrative access to any of those computers can read the email in transit – as long as email messages are, as they typically are, passed around in clear text. Encryption ensures that even if messages are intercepted by any of those dozens of administrators the contents can't be read. It is generally held by security experts that encryption should have been made the de facto standard many years ago. Because it's not, you will have to help lead the way to make it so not only to benefit yourself and your organization but to benefit all Internet users.

BCC

BCC stands for "blind carbon copy"; it enables someone to be sent a copy of an email message without the other recipients' knowledge.

The only time the BCC field should be used is when sending an announcement-style email to a large list of people. In that situation, using BCC rather than a simple CC (or "carbon copy") preserves the privacy of each recipient from all the other people on the list an d anyone else who may be able to access the mailing list archives. With public lists, it's an important practice to ensure that list members are not added to spammers' lists.

In any other situation it's considered poor etiquette to use the BCC field, especially when encryption is used to secure a conversation. The participants in an email thread should be able to assume that the email is being transparently sent to only the visible, explicitly stated recipients. This is particularly true for an encrypted thread, as the act of encrypting the thread sends a strong message that the conversation is intended only for the eyes of known recipients and that the emails are to be kept hidden from the rest of the world. BCCing someone else opens up what should be a very private conversation to another party and is a betrayal of the principle of transparency and of the trust of the participants in the thread. If you are caught by the participants — for example, if the person who has been BCC'd responds to the thread — you will have damaged your relationship with them.

RAISING AWARENESS — USING THE GPG FINGERPRINT

Because the entire public key is unwieldy to use as an email signature or on a business card, GPG provides a function to generate a more compact version known as a "fingerprint". To generate a GPG fingerprint, go to a command prompt in the directory where GPG is installed and type

```
gpg --fingerprint <keyID>
```

replacing <keyID> with either the key's number or a portion of the email address associated with it.

Use your fingerprint as much as possible: include it on your business card and use it as a standard signature on your unencrypted email. Doing so raises awareness among the people you interact with that GPG is important and available, and helps encourage wider use. Having the fingerprint handy on business cards also assists with key signings.

USING ENCRYPTION

SENSITIVE INFORMATION

Always encrypt sensitive information you send via email. If any of the intended recipients does not currently have PGP/GPG installed encourage them to do so and assist them through the process.

REQUESTING ENCRYPTION

If someone sends you sensitive information in the clear, politely request that they send any further information encrypted. If they don't know how, help them through the process.

THREAD ENCRYPTION

When a group of correspondents share in an extended email discussion, all should ensure that forwards and replies are also sent encrypted to all participants.

ATTACHMENTS

Make sure the attachments you send with encrypted email are also encrypted.

There are two methods of doing this. Option one: attach the file to the email and then encrypt the entire email, attachment and all, using GPG MIME. Option two: first encrypt the file on its own using the command line or a GPG utility and then attach it to the email and encrypt the email body. (If the email contains no sensitive information, you could send the email body unencrypted, but our recommendation is that all email should be encrypted as a standard practice.) The first option is not supported by all email clients, and in practice you are most likely to use option two.

SUBJECT LINES

GPG does not encrypt traffic data (see below), which includes the identities of sender and recipient as well as an email's subject line. You will need to pay special attention to the crafting of subject lines to ensure that they do not reveal any sensitive information regarding the contents of the email while still commanding the recipient's attention.

TRAFFIC DATA

The content of a given message is often less revealing than the information about the message (its metadata): who sent it, who received it, when, at which locations, and how often these correspondents communicate. This type of data, typically included in email message headers (and also on the logs kept by Internet service providers of the Web and other addresses subscribers visit) is known as “traffic data”, and many countries, most notably in the EU, have passed laws requiring that this type of data be retained for periods from many months to many years for law enforcement and security purposes.

Consider, for example, a simple message: *Get some tomatoes.*

Evaluating the import of such a message is impossible without knowing something about the relationship between the correspondents. If they communicate dozens of times a day, perhaps they’re a married couple. If one is a known trouble-maker at public events, perhaps they’re arming up for their next outing. If they’re the managers of a grocery store, the message may be commercially sensitive.

Keep in mind, therefore, that although GPG protects the content of messages, it does nothing to protect traffic data. To protect traffic data and guard anonymity, you need a tool like Tor (http://www.torproject.org), which is specifically designed for this purpose.

VERIFYING DOWNLOADS

Many sites make available a hash that you can use to check that downloaded files have not been tampered with (such as by adding spyware or other malicious software).

On all platforms, at a command prompt you can type

```
gpg --verify <filename>
```

GPG will compare the signatures and will report whether they match.

In Windows, you have two additional choices. Either right-click on the filename in Windows Explorer to get a pop-up menu and choose GpgEX | Verify or open Kleopatra and choose Decrypt | Verify from the File menu. Either way you get the same graphical interface; check the appropriate boxes and click on Decrypt/Verify to proceed. The next screen will report whether the signatures match.

For more detail on verifying downloads, see GnuPG’s page on integrity checking: http://www.gnupg.org/download/integrity_check.html.

KEY MANAGEMENT, SIGNING, AND SECURITY

AUTHENTICATION

Sign your messages when it is important for the recipients to be sure the information has come from you.

INTEGRITY, URL LINKS, AND ATTACHMENTS

Sign a message when it is important to be able to prove that the information in the message has not been altered in transit. Such situations might include, for example, sending emails that include URLs to external resources or sending attachments.

In the case of URLs, it is important to ensure that they have not been altered in transit to point to a different resource that delivers a malicious payload.

The risk with attachments is that a different attachment carrying a malicious payload could be substituted by a third party while the message is in transit. This is true even if the attachments are encrypted, as there is nothing stopping a malicious third party from encrypting the substitute attachment with the recipient’s public key. Therefore, the sender must either sign the attachment and also attach the signature to the email, or bundle the entire email and attachment together using MIME and then sign the entire encapsulated mail.

NON-REPUDIATION

If you believe that it may be important in the future that a particular promise made by email cannot be withdrawn, ask the sender to use GPG to sign the message in question.

TRUST NETWORK

VERIFYING IDENTITY

Whenever you meet someone known to use PGP/GPG, take advantage of the occasion to add an identity verification and keysigning session to the meeting agenda with the aim of enhancing the value of GPG’s web of trust. People who have a long deeply trusted history between them, whether they are able to meet in person or not, should also sign each other’s keys, since they have no need to physically verify each other’s identity.

CONFERRING TRUST RATINGS

At keysigning event, it also increases the value of the web of trust if you rate the level of trust you have in the individuals whose keys you’re signing. This rating should include not just the confidence you have that they are who they claim to be, but also how much you trust them in general. GPG supplies a mechanism for doing this. Call up the GPG key editor by typing

```
gpg –key-edit <keyID>
```

replacing **<keyID>** with a portion of the email address associated with the key whose rating you want to edit.

At the command prompt type

```
trust
```

GPG will offer you a menu of choices you can use to modify the person's trust rating.

PERSONAL KEYRING EXPANSION

Get in the habit of routinely checking to see if a new contact has a PGP/GPG key associated with their email address/es. If a valid public key exists for the contact, add it to your personal keyring for future use.

CHECKING TRUST

Whenever you communicate with a member of the GPG cryptosystem for the first time, always check the person's inferred trust. To be useful the web of trust has to be used.

ABOUT PUBLIC KEY CRYPTOGRAPHY

Public key cryptography is a well-established method of securing spontaneous communications between strangers. Although the same idea was developed in secret in the mid 1970s at Britain's GCHQ, its invention is generally credited to Whitfield Diffie, Martin Hellman, and Ralph Merkle, who published the first descriptions of the technique in 1976. In 1978, Ron Rivest, Adi Shamir, and Leonard Adleman published the first algorithm, known as RSA, to support signing as well as encryption. RSA Data Security, now a division of EMC, was the company set up to exploit the RSA algorithm commercially. PGP was an early implementation of RSA, although both it and GPG support other algorithms as well.

Public key cryptography relies on the use of a complementary key pair rather than a single ("symmetric") key. Each of the pair can decrypt material encrypted by the other; each can encrypt material such that it can only be decrypted by the other. One of the keys is kept secret, known only to its owner. The other is public, and published as widely as possible.

Using this system, when Bob wants to send Alice information that only she can read, he encrypts it using her public key (which he can download from a keyserver or even copy out of a book); she will use her private key to decrypt and read it. Similarly, when Alice wants to prove to Bob that she wrote a particular message, if she encrypts it with her private key, using her public key to decrypt it proves to Bob that she was the author. There are many variants on this structure to suit specific situations: Bob could, for example, encrypt the message with both his private key and Alice's public key so that the message was authenticated as coming from him and only Alice can read it.

The RSA algorithm was patented in 1978, but it wasn't until the early 1990s that computing power was cheap and widespread enough for it to become a realistic possibility for everyday use by individuals. During the early 1990s many policy battles were fought between digital rights activists and governments used to thinking of encryption as a military weapon. Government policy makers favored such ideas as key escrow (requiring individuals to deposit copies of their private keys in a government database) and import/export restrictions on the basis that widespread use of strong cryptography would make it hard for law enforcement and security services to protect the public against organized crime and other threats. The growth of electronic commerce, however, made the peacetime uses of strong cryptography too significant to persist with such policies, and the restrictions were removed in most countries.

THE WEB OF TRUST

We use webs or networks of trust all the time in everyday life without consciously thinking about it. You buy a car from a particular dealership based on a friend's recommendation. You buy fresh food from a restaurant you have never seen or heard of before because you trust the health authorities to have inspected the kitchen. More generally, you presume that someone who has set up a shop has entered a lease agreement with a landlord and can be held to account by the police if you have reason to complain about the goods you have been sold.

A web of trust is a mechanism that allows strangers to trust each other by providing some representation of trust to be applied to them by third parties. In the digital world, eBay is a trust network that uses the feedback rating system to allow its users to give each other confidence in trading with the site's large population of unknown, unseen strangers. The aggregated feedback history of an eBay user who has complete many successful transactions that have attracted consistently high ratings gives us a high degree of confidence that it is safe to deal with this person.

This simple explanation hides the fact that even in the physical world the reality is often a little more grey. A used car salesman may have a reputation for ripping people off, but it may not be clear whether the law is definitively being broken or whether the salesman is just extremely good at the job of selling you something that in the final analysis does not meet your requirements. It is within these grey areas that a web of trust can really come into its own.

Trust networks only work if the underlying mechanism is also trusted. We must be able to be certain that the entity we are dealing with is accurately identified. It also means that an entity cannot deny having participated, once committed to a message or a transaction. A system like eBay is a secure and closed environment, owned by a single proprietary corporation that is generally benevolent to its userbase. The Internet at large, however, is an open community, and in that context we must rely on other mechanisms to achieve the same level of confidence.

Public key cryptography is the answer to providing webs of trust in open communities. By design, public key cryptography allows participants to digitally sign their communications and transactions so we can be sure we know that person A really is person A even if we have no way to link "person A" to a real-world identity. Cryptographic signing gives us the quality of non-repudiation, meaning a participant cannot deny having created a message or transaction if the signature on that transaction matches theirs. The PGP (Pretty Good Privacy) and GPG (GNU Privacy Guard) email encryption systems allow the building of open community webs-of-trust.

ABOUT GPG

GNU Privacy Guard (GPG) is a free, open source software implementation of public key cryptography that is developed and licensed by the Free Software Foundation's GNU software project. It is an alternative to the PGP (for "Pretty Good Privacy") suite of cryptographic software. GPG complies with RFC 4880, which specifies IETF standards for OpenPGP, and it is interoperable with PGP and other OpenPGP-compliant systems. The first version of GPG was released in September 1999.

In the interests of speed and efficiency GPG uses not only public key cryptography but also a number of other cryptographic mechanisms such as symmetric keys, hash functions, and so on. This multi-faceted approach makes it a flexible cryptosystem with many uses. It is optimally suited for securing email, but it can also be used to secure mailing lists, selected files, and whole filesystems, as well as communications over other protocols (for example, instant messaging, via plug-ins for popular clients such as Jabber).

The most important guarantee of the resistance to cryptanalysis of a particular piece of cryptographic software is that it has been thoroughly studied and inspected – peer-reviewed – by the cryptographic community. PGP and GPG have so far successfully withstood this process (for 20 and 12 years respectively) and are considered robust and secure.

ADDITIONAL RESOURCES

GNU Privacy Guard Manual:

Gpg4win Compendium (PDF): http://wald.intevation.org/frs/download.php/775/gpg4win-compendium-en-3.0.0-beta1.pdf

//

GLOSSARY

Hash function: A cryptographic hash, or digest, is a one-way mathematical function that takes a block of data and returns a fixed-size bit string. It is used to prove that the contents of a message have not changed because even a slight change to the message will change the hash, often quite dramatically; the message cannot be reconstructed from the hash; and no two different messages generate the same hash. Well-known hash functions include MD4, MD5, SHA-1 and SHA-2.

Keyserver: A keyserver stores copies of public keys so that anyone may download a copy and use it to authenticate messages appearing to come from a particular person. When you generate a new keypair, you should upload your public key, and you should update it as you get it signed by those who can verify your identity and tie it more closely to your public key.

Non-repudiation: A significant characteristic of public key cryptography is that when a message has been signed by a particular person's private key (signaled by being able to decrypt that message using that person's public key) the person cannot later deny authorship.

PKI (Public key infrastructure): The ecosystem of keyservers and certificates that manage and authenticate public keys.

Public key cryptography: By 1976, the advent of cheap computing and the dawn of computer networks created a clear need for a technique to enable spontaneous secure communications between strangers. The solution, publicly attributed to Whitfield Diffie, Martin Hellman, and Ralph Merkle but separately privately invented at Britain's GCHQ, replaces the single symmetric key with a simultaneously generated keypair, of which one is kept private and the other is public (and published as widely as possible),

Signature: In public key cryptography, a signature to an email message authenticates the message and robustly identifies the sender. The signature is created by using a one-way cryptographic function to create a digital hash of the message and then using the sender's private key to encrypt the hash. When a message is signed by this means the recipient can use the sender's public key to decrypt the message (proving the identity of the sender) and then run the same cryptographic function over the message to generate a second hash to compare with the decrypted one (proving the message has not been tampered with).

Symmetric key cryptography: The dominant form of cryptography until the 1980s, symmetric key cryptography uses a single key to encrypt and decrypt communications. The drawback is that two strangers wishing to communicate securely must first distribute copies of the key to each other. In GPG, for speed reasons, the session keys used to encrypt whole messages are symmetric, and the public/private key pair are used to exchange session keys and for signing.

Web of trust: Until public key cryptography was invented, most trust was hierarchical and relied on authorities. A bank, for example, might issue a letter of credit a stranger could present to prospective business partners in a foreign land to show that he was trustworthy. GPG builds instead on the "six degrees" idea and assumes that people trust people they know. Accordingly, when you generate your public key, you ask people you know or who can verify both your identity and the key itself to sign your key. As the number of people who have signed your key grows, your key becomes more trustworthy to the entire network, partly through numbers (just like eBay reputations) and partly because some of the people who have signed your key will be personally or professionally known to those who seek to use it. The resulting chain of trust grows into a network very like the World Wide Web itself, hence web of trust.

DETAILED INSTALLATION INSTRUCTIONS, BY PLATFORM

WINDOWS

The Gpg4win installation package includes modules to integrate GPG with a number of email clients. Our focus here is on Thunderbird and Outlook because almost everyone will be using one of these two.

INSTALLING GPG4WIN

Download Gpg4win from http://www.gpg4win.org/download.html and save the file in a convenient location.

Double-click on the file to run it.

Gpg4win unpacks for installation like any other ordinary Windows software. You will be asked to choose a language and then you will see a splash screen that describes the software followed by a license agreement.

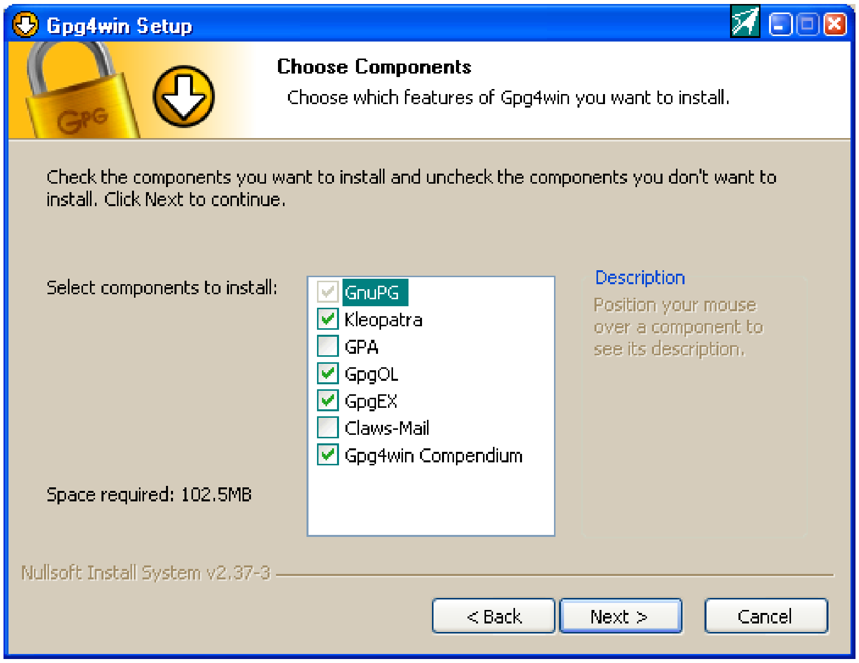
You will then be asked to choose which components you want to install. GnuPG is required. By default, the program is set up to install:

- Kleopatra, a module that manages keys and certificates;
- GpgOL, which integrates Gpg4win with Outlook 2003/2007;
- GpgEX, which integrates Gpg4win with Windows Explorer so you can encrypt a file from within an Explorer window;
- Gpg4win Compendium, the software manual and an excellent resource.

Also offered are:

- GPA, an alternative to Kleopatra to manage keys and certificates;
- Claws-Mail, an alternative email client that integrates well with Gpg4win.

If you intend to use Gpg4win with Outlook, and you are running Outlook 2003 SP2 or later or 2007 under XP, 32-bit Vista, or 32-bit Windows 7, you should ensure that the box is checked to install GpgOL, which integrates GPG into Outlook for you. GpgOL will not run on 64-bit versions of Vista and Windows 7.



Choose Components

Choose which features of Gpg4win you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

☑ GnuPG
☑ Kleopatra
□ GPA
☑ GpgOL
☑ GpgEX
□ Claws-Mail
☑ Gpg4win Compendium

Description

Position your mouse over a component to see its description.

Space required: 102.5MB

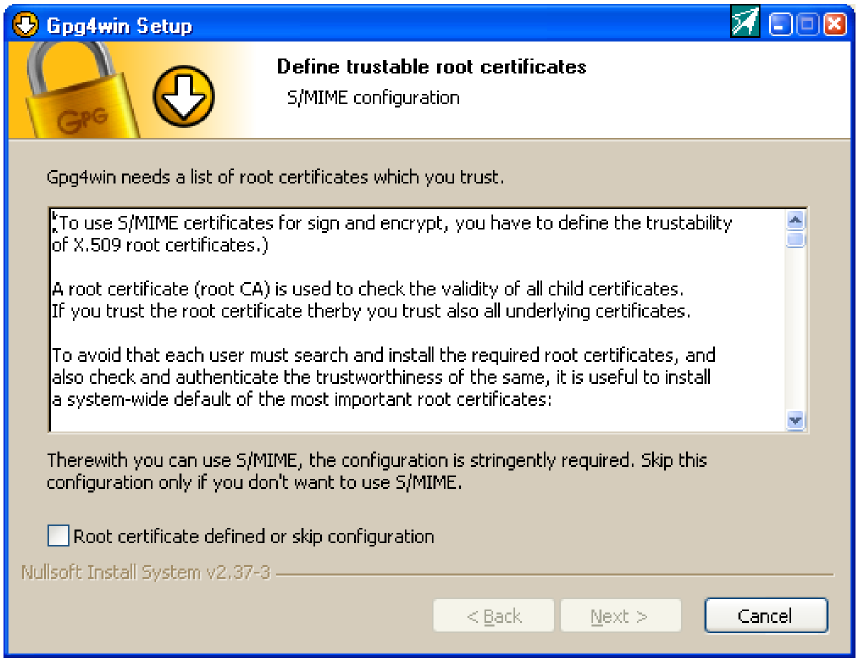
By default the software installs itself into

C:\Program Files\GNU\GnuPG

but you can choose any other directory you like. You will be asked whether to create a desktop icon, a Start menu item, and/or a quick launch bar item, and then installation will proceed.

The software will also create the directory C:\Documents and Settings\<your username>\Application Data\gnupg. This is where your and your correspondents' keys will be stored, and this is the directory to back up if you want to keep a copy of your keyring. When installation is complete, Gpg4win will ask you to define trustable root certificates. You only need to do this if you want to use S/MIME. For the moment, check the box to skip this configuration.

(See next page.)



Define trustable root certificates

S/MIME configuration

Gpg4win needs a list of root certificates which you trust.

To use S/MIME certificates for sign and encrypt, you have to define the trustability of X.509 root certificates.)

A root certificate (root CA) is used to check the validity of all child certificates. If you trust the root certificate therby you trust also all underlying certificates.

To avoid that each user must search and install the required root certificates, and also check and authenticate the trustworthiness of the same, it is useful to install a system-wide default of the most important root certificates:

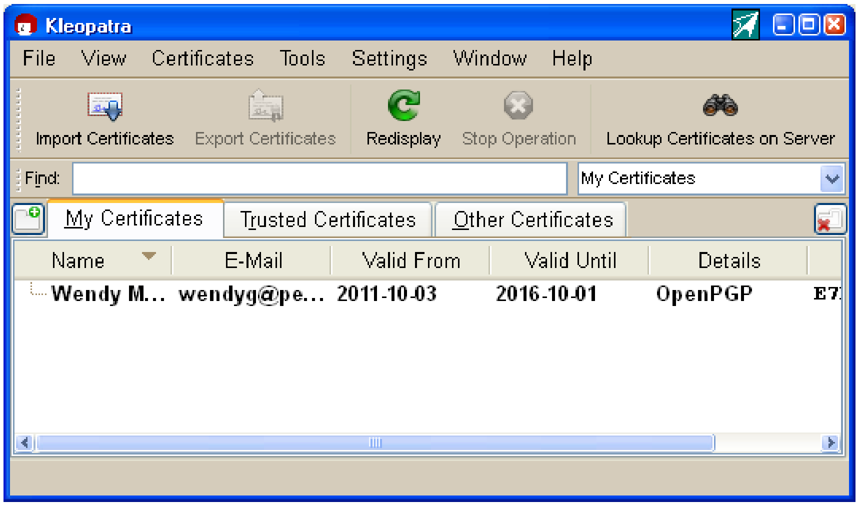
Therewith you can use S/MIME, the configuration is stringently required. Skip this configuration only if you don't want to use S/MIME.

□ Root certificate defined or skip configuration

Nullsoft Install System v2.37-3

This completes basic installation.

You will now need to generate a keypair. You can do this through Kleopatra, which, depending on your choices during installation, should appear on your desktop or in your new Start menu group. If not, navigate to the directory where you installed GPG and double-click on Kleopatra.exe. To generate your keypair, click on File | New | Certificate, and Kleopatra will guide you through the process.



Name	E-Mail	Valid From	Valid Until	Details	
Wendy M...	wendyg@pe...	2011-10-03	2016-10-01	OpenPGP	E7

Choose your passphrase carefully! (As explained above.)

Also generate your revocation certificate at this time, copy it into a secure location, and remove it from the directory where your keys are stored.

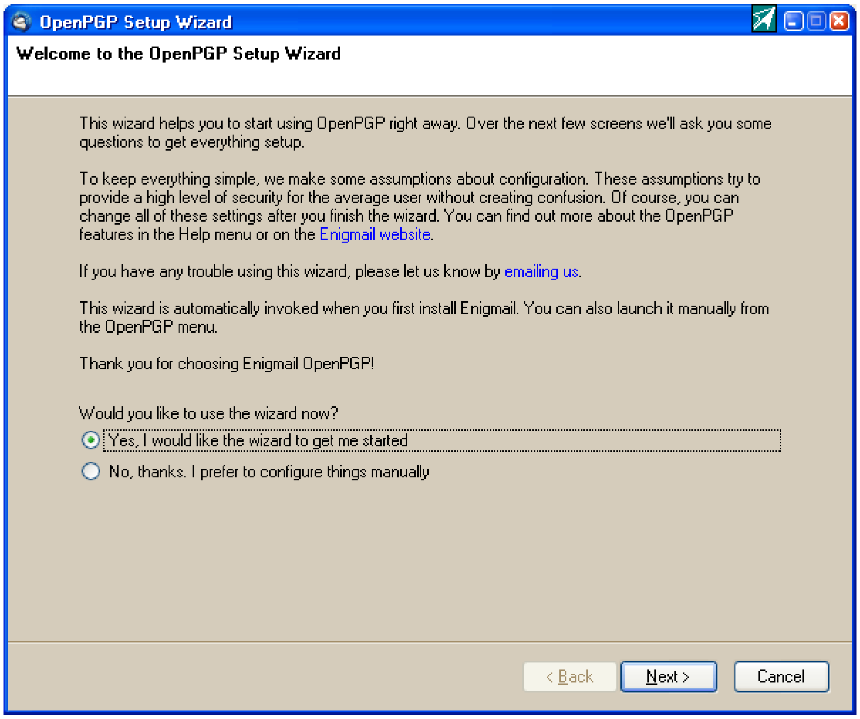
Your email software will take it from here.

WINDOWS — THUNDERBIRD

Download Enigmail from http://enigmail.mozdev.org/download/index.php.html. Make sure you retrieve the correct version for the version of Thunderbird you are using (you can check this in Thunderbird by choosing the About option from the Help menu).

From inside Thunderbird, go to the Tools menu and choose Add-ons. Click on Install in the lower left corner, navigate to the location where you saved the Enigmail add-on, and select it. Thunderbird will do the rest. After you restart Thunderbird, you will find a new menu on the toolbar called OpenPGP that includes all the necessary functions for operating GPG: when and whether to encrypt email, how long to keep your passphrase active before you need to retype it, and access to public keyservers to retrieve your correspondents' keys and upload your own. You can even sign other people's public keys and upload them through the Thunderbird Enigmail OpenPGP menu.

From the OpenPGP menu choose the Setup Wizard to help you through configuring Enigmail.



Welcome to the OpenPGP Setup Wizard

This wizard helps you to start using OpenPGP right away. Over the next few screens we'll ask you some questions to get everything setup.

To keep everything simple, we make some assumptions about configuration. These assumptions try to provide a high level of security for the average user without creating confusion. Of course, you can change all of these settings after you finish the wizard. You can find out more about the OpenPGP features in the Help menu or on the Enigmail website.

If you have any trouble using this wizard, please let us know by emailing us.

This wizard is automatically invoked when you first install Enigmail. You can also launch it manually from the OpenPGP menu.

Thank you for choosing Enigmail OpenPGP!

Would you like to use the wizard now?

● Yes, I would like the wizard to get me started

○ No, thanks. I prefer to configure things manually

WINDOWS – OUTLOOK

If you checked the GpgOL box during installation, Gpg4win will have installed the plug-in so that the next time you open Outlook there should be a new menu labeled GpgOL. However: GpgOL works only with Outlook 2003 and 2007. It does not work with older or newer (2010) versions of Outlook. Because Outlook is proprietary software, integrating GPG is not quite as seamless as it is with other open source software.

However, this may be more of a problem for developers than for users. When you click on the GpgOL options tab, you will see boxes to check to enable S/MIME support (check this), whether to encrypt new messages by default (yes), and whether to sign new messages by default (yes). You can change these settings on a per-message basis if you need to alter them later.

We would recommend leaving the option to show messages in HTML unchecked (though you may find it makes some messages messy to read), and also leaving unchecked the option to present encrypted email as an attachment.

You may need to make some changes in how you use Outlook.

- Do NOT use Microsoft Word to compose messages.
- Do NOT compose and send HTML messages (any formatting you apply may be lost in the process of encryption and decryption).

You can check these settings by going to the Email formatting tab on the Options menu. Set the message format to “Text only”.

UBUNTU (AND MOST OTHER POPULAR LINUX DISTROS) WITH THUNDERBIRD

GPG should be already installed, so you just need to install Thunderbird and follow the same instructions as for “Windows - Thunderbird”

MAC OS X

http://gpgtools.org/

Other graphical front ends for Macs are available at http://www.gnupg.org/related_software/frontends.html#mac.

ANDROID

You can download Android Privacy Guard (APG) and the K-9 email client from the market. Go to http://thialfihar.org/projects/apg/ to learn how to use them.

IOS (IPAD, IPHONE)

Decrypt only for (intact) iPhone - oPenGP Lite - http://pgp.wiredpig.us/2010/pgpgpg-on-iphone/

Access is an international NGO that promotes open access to the internet as a means to free, full, and safe participation in society and the realization of human rights. Founded in the wake of the 2009 Iranian post-election crackdown, Access works to build the technical capacity of digital activists and civil society groups, provide thought leadership and pragmatic policy recommendations to actors in the private and public sectors, and mobilize its global movement of citizens to campaign for digital rights.

For more information, please visit **https://www.accessnow.org**
or email **soc@accessnow.org (PGP Key ID: 0xF08D380A)**