Windows Privacy Tray

A free GUI Front-End for GNU Privacy Guard
WinPT version 1.0.1 released on 30 September 2006
Document version 1.1.1 updated on 18 October 2006

Timo Schulz
Sundar Pillay

Table of Contents

1 GNU Privacy Guard

1.1 What is GNU Privacy Guard?

GNU is a self referencing acronym (GNU'S NOT UNIX) created by Open source enthusiasts. GNU Open source projects are written under GNU General Open source license. Privacy Guard is a application for secure communication & authentication. Privacy Guard follows GNU Open source license to provide free usage of Privacy Guard for commercial, non-commercial and personal use. It is appropriately named as GNU Privacy Guard. It is also called as GNUPG, and sometimes GPG.

1.2 Privacy Standards

GNUPG is based on OPENPGP standards/specifications (RFC2440) for communication privacy. OPENPGP standard allows End users to follow a common secure mechanism to use in secure signing, encryption of data.

1.3 Usage of GNU Privacy Guard

GNUPG allows users to create encryption,decryption keys as a key pair. It also provide method for signature for the messages sent, and verification of signatures when received. A key pair contains a Private key and a Public key. One or more key pairs can be used by an user. These key pairs used in encryption, decryption, signature and verification.

2 Windows Privacy Tray

2.1 A Gui Front-end for GNUPG

GNUPG allows users to manage their keys using commands at command prompt(in Windows). WinPT acts as Front-End Gui for easy management of key pairs and related activities. It saves a lot of time when using GUI Front-End, than typing all commands in the command prompt. WinPT allows flexible execution of commands, with advanced options.

WinPT is a graphical GNUPG front-end which resides in the task bar. It is divided into several, so-called, managers. There is a manager for the key(ring), for files and for smart cards. The aim of the program is to secure email communication and to perform file encryption and to allow an easy and user friendly way for key management.

2.2 Requirements for WinPT

You need at least Windows 98/2K/XP. But Windows XP or better is recommended. The program also works on NT/95/ME but there is no support for these OS versions any longer. Mainly because the OS vendor dropped support and no bug fixes will be provided. So, WinPT may not work on such platforms.

You need to have a working GNUPG 1.4.x installation on the machine you plan to install WinPT. If you do not have GNUPG in your machine, please visit http://www.gnupg.org and download the latest GPG version there. It comes with a graphical installer so there is no need to do the installation manually.

Alternatively, you may use one of the graphical GNUPG installers which are available on the internet.Gpg4Win package includes a set of very useful privacy tools including GNUPG and WinPT along with two German Manual.It is very easy to use with an average size (~4MB). There is also a light version which does not come with German Manual. When installing Gpg4Win, both GNUPG and WinPT can be installed together.

2.3 Installation

Download the latest zip package from http://wald.intevation.org/projects/winpt. Also, download the signature of the package, and verify to make sure that the release is authentic and it is not altered in any way. All files in the zip package need to be unpacked in the same folder, so if you change the folder do not forget to move all files.

To activate the program you just need to start WinPT.exe. The program will start and run appear in the Windows Tray bar. You should now see a little (golden key) icon in the taskbar which indicates that the program is running. If you want to quit the program, right click on the symbol and select "Exit".

2.3.1 Configuration

After the installation much of the default settings no need to be changed. If you prefer a special keyserver, it is probably a good idea to open the keyserver dialog and to set one of the existing keyservers as the default or create a new entry and mark it as the new default. The default keyserver is *subkeys.pgp.net*, which is the best choice for most users.

2.3.2 The GPG Preference Dialog

In this dialog you can change your GPG config and customize its behavior. Please be advised that in most cases there is no need to overwrite the default GPG path settings. There are three different paths available. First, the GPG home directory. The place where the keyrings are stored and also the config files. The second path points directly to the gpg.exe. The third is the

path to the language files, where you usually store your winpt.mo/gpg.mo files. These entries should be only changed when there is a real need, and extra caution is needed because with wrong settings, WinPT will not be able to work any longer!

The second part of the dialog is the "General GPG options" section. Here you can influence the behavior of some commands. If you do not know what they mean, it is safe not to change the values and stick with the default ones. For expert users, it is possible to set the signature class of issued key signatures and to set an expiration date for key signatures or to specify an comment in armor files. The "Encrypt to this key" might be useful for anybody who needs to decrypt mails or any data he sent to a recipient. The field value should contain the key ID of the default key pair.

2.3.3 Preferences

In the WinPT preference dialog, the user can modify and/or disable the default options. For new users it is suggested to leave the default values as they are, except when there are problems related to the hotkeys.

To enable keyring backups, the user can either decide to use the GPG home directory as the backup folder or any other folder. In the latter case, a folder needs to be chosen. The program makes the backup before it terminates and thus it is very important that the keyrings are still accessible at this moment. For example if you use an USB flash drive to store your keyrings, you should unplug it after the the icon disappeared at the task bar. By default the secret keyring will not be backuped, if you wish that the secret keyring should be also backuped, and this usually means the backup folder cannot be accessed by other people, you need to mark "Backup includes secret keyring".

2.4 WinPT and Personal Firewalls

Because the program uses a global hook to remember the last active current window, it might be possible that Firewalls warn that the process contains a global hook which is a potential security risk. In some cases, there might be even a warning that key logging is possible. This is a false alarm because the hook provided by the program, a CTB (Computer Based Training) hook, can be only used to save handles of newly created windows, or windows which are activated or in case of a focus change. Details can be found in the source code of the program or additional information about the CTB hook at msdn.microsoft.com

To provide access to keyservers and to download HTTP keys, the program needs to be able to make outbound connections to the following ports: 80 (http), 11371 (keyserver)

2.4.1 Using a HTTP Proxy

If you are behind a firewall and you have no chance make a connection to a keyserver, maybe because of a policy, you can use a http proxy for outbound connections. Open the Keyserver dialog and click on the button "Change Proxy". A new dialog opens where you can enter the proxy specific host name and ports. If the proxy requires authentication, you also have to provide your user name and your password. Please bear in mind that only a base64 authentication is supported and no other proxy types (SOCKS for example) can be used.

3 Privacy Guard Concepts and Functions

3.1 Single password method versus Key Pair method

Early days of Computing, people use single password to encrypt and shared the same password to the person who needs to decrypt. Now, we use complex big password containing 1024 or 2048 randomly generated characters. We call this type of big password as Key. Since we can't remember this Key easily, and also we can't type this Key quickly, We use Passphrase (like a password) to apply the full Key. GNUPG takes your Passphrase, and verify it with the Key. If the passphrase is eligible to access the Key, then GNUPG application allows to use it.

These Keys are generated in Pair. These are complementary keys, which means if one key encrypt another key is able to decrypt, and vice versa. And these keys act in one way only, which means, if Public key is used to encrypt the message, then only private key can decrypt that message. In the same way, if a Private key is used to encrypt, it cannot decrypt the message; only Public Key can decrypt.

A Key pair includes one Public Key and one Private key. A person have to keep the Private key (also called Secrete key) under control and protect with the password. Public key is used to encrypt data(including email) by others to the owner of the Public key.

After encryption by the Public key, the data can be decrypted by the Owner of the Public key by using the Private key. Once encrypted by the Public key, the sender cannot decrypt the data. Only the Private key which was created as a Pair along with Public key can decrypt.

3.2 Key rings, Export, Armor

GNUPG creates keys and stores them in the small files. Private Keys (also called Secret Keys) are kept in secring.gpg file. Public Keys are kept in pubring.gpg file. Trustdb is Your trust rating for all Keys including Public Keys of others which you collect. GNUPG also has a gpf.conf file. These Key rings (secring.gpg & pubring.gpg) along with trustdb.gpg kept separately for each user.

From the Key ring, we need to export the Public Key for distribution. The exported key will have .asc to show it has ASCII coded. To ensure, the exported key can be safely communicated thru email or displayed on the web, it is converted to ASCII with format, and it is called armored key

3.3 Finger Print

Finger Print is similar to checksum. Finger Print of the Public Key is created by using SHA1 method. SHA1 method is a hash method which has better data integrity that the checksum.

The fingerprint of the key is hexadecimal (160-bit) sequence divided into 10 groups of 4 hex digits as per the OPENPGP Standard. You can get the fingerprint of a key by opening the key property dialog. There you can mark the fingerprint and copy it to the clipboard. The fingerprint of a key can be compared to human fingerprints, it is unique for each key.

```
Example: 1D75 8108 5BC9 D9FB E78B  2078 ED46 81C9 BF3D F9B4
```

It is a good idea to publish your fingerprint wherever possible. For example via a business card or your website.

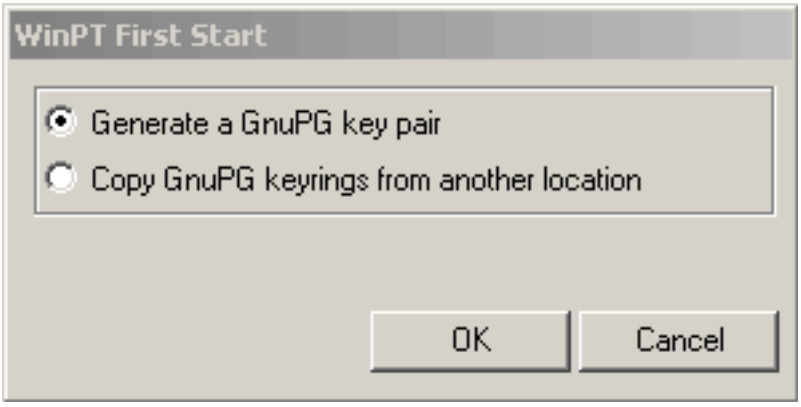
3.4 Ownertrust

The ownertrust is how do you rate Public keys of somebody else (other than your Personal keys). For example, if you know that Bob is really the owner of the key, you should sign it. But he is also known to sign other keys without checking the identity of the other key owner. Values

for the ownertrust are 1) Don't Know 2) Don't Trust 3) Marginal 4) Full and thus you should probably use an ownertrust value like "Marginal". But this is a personal decision and stored in a separate file and never exported with the public keys. For further information, please take a look into the GNU Privacy Handbook. Just a last word on your Personal Key Pairs, they are automatically marked as "Ultimate" because the key belongs to you and you trust it implicit.

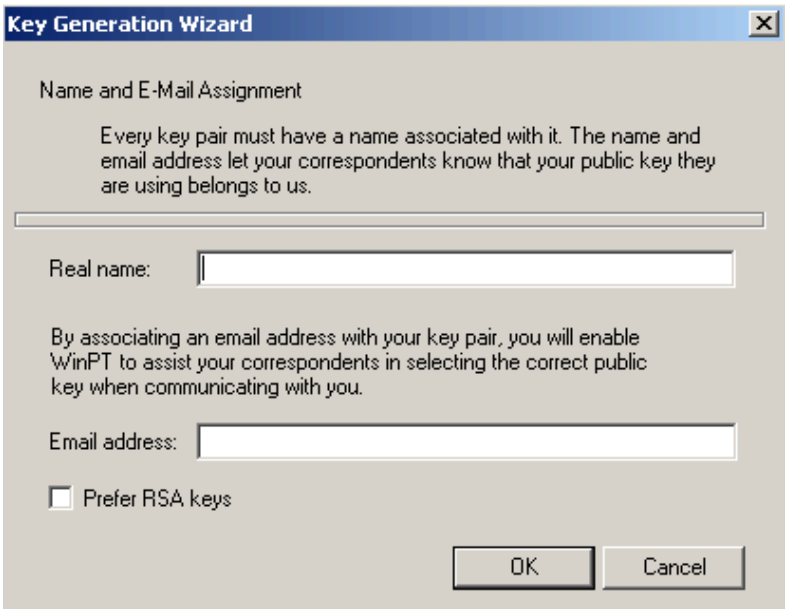
3.5 The First Start

When the program is started the first time, it offers two choices. One is to *generate a GNUPG key pair* and the other is to *copy existing GPG keyrings from another location*



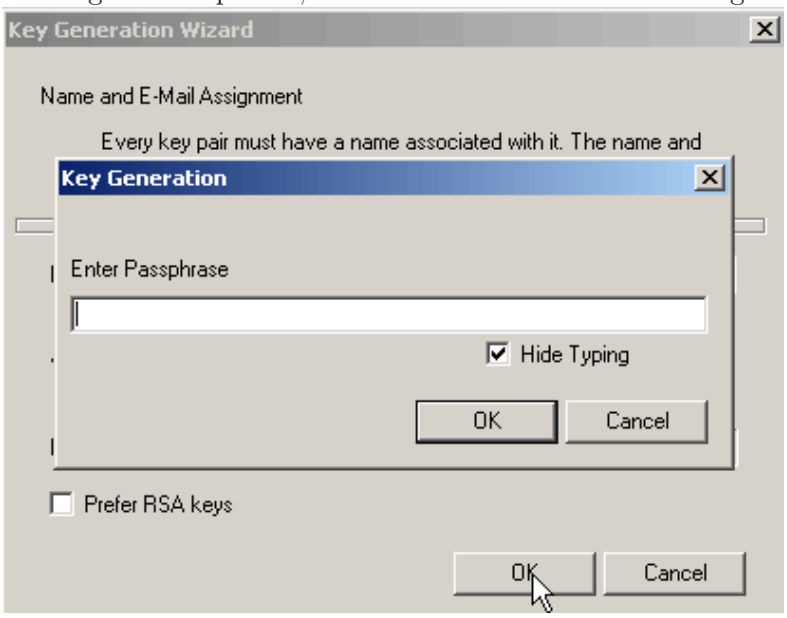
Screen 3.1: **WinPT First Start**

This introduction is starting from scratch, So, assume that the user selects to generate a new key pair, a new dialog will be shown to collect information about **Name**, **Email** and optionally **Comment**. If the user prefer RSA keys, the check box should be marked. If the entered data is OK, WinPT then go to next step to generate key pair using Key Generation Wizard.



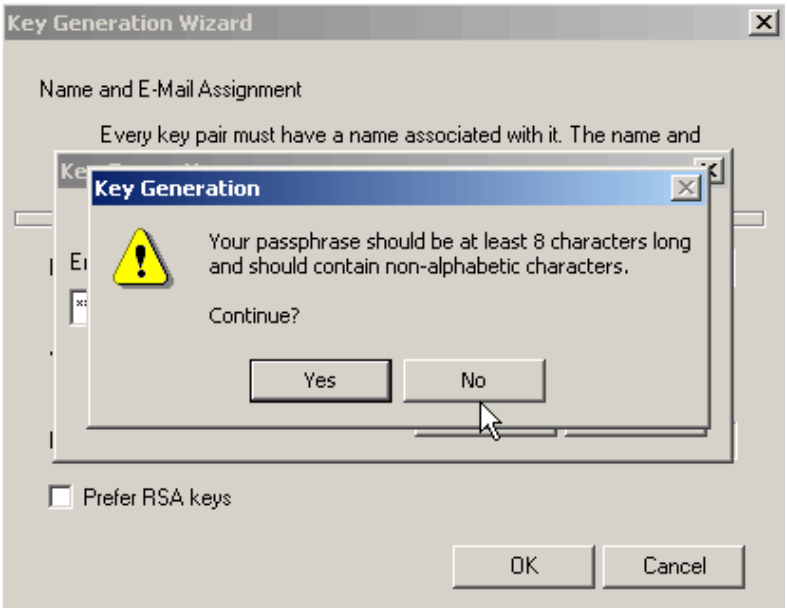
Screen 3.2: **Key Generation Wizard**

You have to enter a good Passphrase; use more than 8 characters or digits



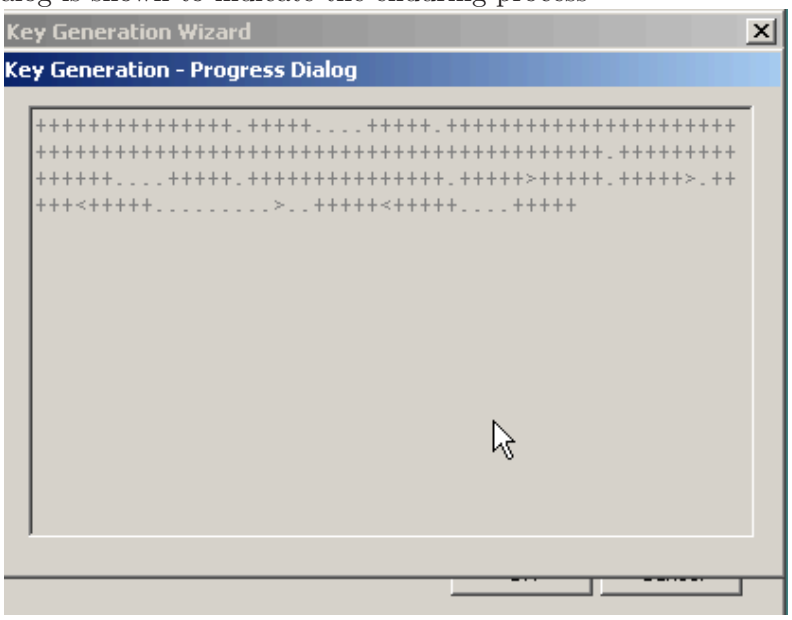
Screen 3.3: **Enter Passphrase**

After entering passphrase, it will ask whether to continue.



Screen 3.4: **Continue Key Generation**

A progress dialog is shown to indicate the enduring process



Screen 3.5: **Key Generation progress...**

When the generation of the keypair is done, WinPT offers the chance to backup the existing keyrings. This is definitely an important decision because if the keyring will get corrupted or lost, there is no way to recover the encrypted data. That is why it is also important to store the backup, at least of the secret keyring, at a **safe** place.

3.5.1 Use existing Keyrings and/or Keys

If you already have a valid OpenPGP key pair then the program will copy your existing keyrings to the new home directory. Please bear in mind that you need to set the ownertrust manually for each imported key. You can skip this step if you exported the ownertrust manually to a file, but because this is a step for experienced users it is not described here. The most important step is, to set your own key to ultimate ownertrust after import.

If you have other OpenPGP programs and you wish to use the keys from this application, it is a good idea to select all keys you want to use and to export them into a single file. Then open the WinPT Key Manager and drag the file into the Key Manager window.

3.6 The Passphrase for the Secret Key

First a short explanation what passphrase is. A passphrase is like a password but usually longer, maybe a sentence, which can consists of any 7-bit ASCII characters. It is used to protect your secret key and thus it is very import to chose a secure passphrase. If your computer, and thus the secret key, were stolen and an attacker can guess your passphrase he is able to decrypt all your data and to create signatures in your name! A good passphrase is difficult to guess but easy to remember and should be at least 10 characters long. An easy way to generate a strong passphrase is to use a sentence only you know but you can easily remind and then take the first letter of each word, plus some special characters and maybe even some intentionally made spelling mistakes.

`Example - A easy to remember sentence:`
`Row - row - row your boat, gently down the stream`
`Passphrase:` **R - r - ryb,gdts**

Never write down your passphrase or share it among other people!

3.6.1 Adding a new Photographic ID

With this function you can add a photo to your public. It will be displayed in the key property dialog.

You just need to select a JPEG file which contains the photo and enter your passphrase and confirm with OK. Please read the note in the dialog carefully to make sure the photo has a proper size (file, height and weight).

4 The Web of Trust

http://www.gnupg.orgprovide a useful information for understanding how GNUPG works. Web of Trust is, how we use the GNUPG key trust mechanism in practice

The certification scheme of OpenPGP is not based on a hierarchical or centralized certification approach. Instead it uses a combination of ownertrust and direct key certification. In the OpenPGP standard each End User can rate the level of trust on others's Public Keys.

When you receive a Public key of another person how do you know that the key is created by the person and for that person? Best way is both of you meet and exchange public keys. Since both of you meet, you can verify each other GNUPG finger print identification if needed.

In case if you are not able to meet the new person, but you know people who already trust, and those trusted people already trust the new person, atleast you have some confidence.

Here is an example with the imaginary persons called Alice, Bob, Carol and Dave.

Alice knows Bob and checked the fingerprint of Bob's key when he met him personally. Thus she knows that the key really belongs to its owner and he trusts Bob to certify other keys. Then she issued a signature on Bob's key. Bob knows Carol and also checked her identity. Then he signed her key. Alice does not know Carol, but he knows Bob and Bob trusts Carol. Because Alice trusts Bob, at a level she decided before, he also trusts Carol. Dave is isolated and does not know anybody for the mentioned reasons, thus he is not in the Web of Trust(WoT). Another very important point is, that the signer can decide, after the certification, how much he trusts the key owner to certify other keys.

It is very important to check the identify of a key owner. Mostly this is done by comparing the fingerprint, which were submitted by phone or written down at a personal meeting, with the fingerprint of the key in the keyring. Please bear in mind that anybody can create a key with an email address and a specific name. Thus it is not recommend to sign keys without doing this check before!

It is a good idea to publish your fingerprint wherever possible. For example on your business card or your website.

5 Key Server and Key Management

5.0.1 Sending a Key to the Keyserver

After you generated a new key pair, it is a good idea to send your key to the keyserver to make it available for other users. If you issue a signature, the key ID is part of the signature and people can automatically retrieve your key when they try to verify the signature.

Actually, the action is performed in the Key Manager and not in the keyserver dialog. Just open the Key Manager, select the key you want to send right-click on it and chose "Send to Keyserver" in the popup menu. Then a message box with the result is shown.

5.0.2 Add, Delete or Edit a Keyserver Entry

The keyserver dialog allow to change the existing keyserver entries, to delete them or to add new entries. Just right click on a selected item and a popup menu will be shown with ("Edit", "Remove" and "New").

5.0.3 Retrieve a key by its email address

If you only know the email address from your partner, you can enter it instead of the key ID. It is unlikely but possible that there are more keys with the same address. In this situation, WinPT will warn you that multiple keys were imported. The difference to the search function is, that the keys were directly fetched and not displayed as a key result list.

```
Example:

pattern: name_of_friend@gmx.net

[Receive]
```

5.1 Keyserver Access

An easy way to retrieve keys is the keyserver. You can think of it like a huge database with a lot of keys as its content. It is possible to search keys by a pattern, a keyid or even a fingerprint. WinPT allows to access different kind of keyservers. For example LDAP, HKP, Finger and HTTP. But the focus will be set on HKP because this is the common case.

In some situations WinPT asks the user whether to retrieve keys automatically. One example is the signature verification when the key that issued the signature was not found in the keyring.

The main keyserver dialog allows to fetch one or more keys directly or to search for a given pattern.

5.1.1 Retrieve a key by Key ID

The best way to fetch a key from the server is by the key ID. Just enter the key ID, it is a good idea to prefix it with 0x, and click the "Receive" button.

```
Example:

pattern: 0xBF3DF9B4

[Receive]
```

5.1.2 Search for a key by pattern

If you want to communicate with a new mail partner and you are not sure about the key ID, it can be useful to search for his email address. This address is considered as quite unique. Not all keyserver support this query mode, so if you get an error please use subkeys.pgp.net.

```
Example:

pattern: winpt@windows-privacy-tray.com

[Search]
```

Now a dialog is opened with a list of all keys which matched the search string. If the name and the email address is known, the matching key should be selected and "Receive" should be clicked. Then the key will be downloaded and added to your keyring. Now you can encrypt data with this key, for example an email.

5.2 Using the Clipboard

WinPT does not depend on a special mail client interface. It uses the clipboard to encrypt and/or sign data. For the examples, let's assume that you want to write a new mail or that you received a mail protected by GNUPG.

5.2.1 The Clipboard Editor

This dialog allows it to modify the clipboard contents directly and/or to display the contents of the clipboard. It is also possible to load a text file into the clipboard or store the contents into a file. For the convenience, the dialog also allows to encrypt and/or decrypt clipboard data.

5.2.2 Encrypt Data in the Clipboard

Just copy the text from the mailer window into the clipboard. This is usually done by CTRL+C, make sure you really selected all portions of the text. Then right-click on the tray icon and select Clipboard->Encryption. Now a dialog is shown to select the recipients. This means you need to select all keys which should be able to decrypt the mail. Confirm with "OK". GNUPG now encrypts the data with the selected recipients. At the end a message box with the result is shown. Now the clipboard should contain the encrypted data. Just paste it into the mailer window. The output should contain a header and a footer "BEGIN PGP MESSAGE" and "END PGP MESSAGE.

5.2.3 Decrypt/Verify Data from the Clipboard

The most common case is probably that you got a signed email and now you want to verify it. For this procedure, you have to copy the entire signature in the clipboard. The easiest way is to use CTRL+A and CTRL+C, then all available text will be copied. WinPT (GNUPG) is smart enough to figure out the signature related data. Now go to the taskbar, display the popup menu and select Clipboard->Decrypt/Verify. Now a new dialog, the verify dialog, should be available on screen with all information about the signature. For example who is the signer, when was it signed how much do you try this key and what was signed and most important, the status of it (is the signature good or BAD). A special case is when you don't have the public key to verify the signature, if this happens WinPT offers to download the key from the default keyserver. If the key was not found, the procedure is aborted because without the key the sig cannot bed checked.

5.2.4 Sign the Clipboard

We assume that text that shall be signed is already in the clipboard. If not, select the text you want to sign and copy with via CTRL+C in the clipboard. Now go to the taskbar and open the popup menu, Clipboard->Sign. If you just have one secret key, the passphrase dialog will be automatically shown. All you need is to enter your passphrase and confirm. In case of more available secret keys, a list with all keys is shown and you can select which key shall be used for signing. The output is always a cleartext signature which is in text format. Do not try to sign binary clipboard data, the result would be unpredictable and not readable by human beings.

5.3 The Current Window Support

Compared to the clipboard mode, the CWS mode has some advantages. Let us assume that you want to extract text from an editor window. With the CWS mode, the program automatically tries to focus the window to select the text and to copy it to the clipboard and execute the selected command (Sign, Encrypt, Decrypt) and pastes back the GPG data to the window. No manual user interaction is needed. Except this different behaviour, it is very likewise to the clipboard mode and thus we do not describe each command again.

5.4 The Key Manager

This part of the program is probably most important for many users. It contains function to manage your keyring and to perform actions which are required and/or useful in the OpenPGP environment.

5.4.1 Tips

- If you want to import quickly a key from a into the keyring, just drag and drop the file into the Key Manager window. Then the import procedure will be automatically started.
- Key which were fetched from keyservers often contain a lot of, maybe obsolete, self signatures, if you want to get rid of them you can use the Key Edit->Clean feature. Just start the edit dialog and select the clean command. That's it.
- The keyserver dialog does not allow to import a key directly via an URL, as an alternative you may use the "Import HTTP..." feature in the Key Manager. With it you can directly fetch keys from the web (Example: `http://www.users.my-isp.de/~joe/gpg-keys.asc`).
- To customize the parameters of the generated key, you can use the expert key generation. It allows you to set the public key algorithm and/or the size of the key directly.
- Most of the list view based dialogs allow to use the right mouse button, to show popup menus with available commands.

5.4.2 Create a Revocation Certificate

It is very important to do this step early as possible. With this certificate, you can revoke your entire key. The reason for this can be for example, that your key is no longer used or even compromised. After you generated the revocation cert, you should move it to a secure place because anybody who gets access to it, can render your key unusable.

Just right-click on your key and select "Revoke Cert". If you do this step directly after key generation, there is no need to change the default values. Just select a file name and enter the passphrase. The program issues a warning which should be read carefully.

5.4.3 Adding a new Secondary Key

For most users the existing keys in the key pair are enough and no extra key is needed. But there are some exceptions.

- The primary key has no secondary key and the primary key is not able to encrypt data. In this case it can be a good idea to add a secondary encryption key.
- A lot of people use secondary encryption keys with an expiration date. Usually the key is valid for 1-2 years. After the key is expired, a new key is needed in order to encrypt data.

What kind of public key algorithm should be selected is a matter of taste. RSA and ElGamal are both capable for encryption. For most users it's a good idea to let the program chose the key size (in bits). The default settings should be secure enough for most purposes.

5.4.4 Adding a new User ID

If you got a new email account, it's probably a good idea to add these new account to your key also. For example:

A new account was registered at gmail.com (john.doo@gmail.com). Then you should create a new user ID with the following fields:

```
name: John Doo

email: john.doo@gmail.com

comment: (optional)
```

Now email programs are able to associate this address with your key when somebody wants to send you a protected mail to this account.

5.4.5 Adding a new Designated Revoker

If you want to allow another key to revoke your own key, this might be useful if you lost your secret or a similar situation, you can use this function to add a designated revoker to your key.

All you need to do is to select the key you want to add as a designated revoker. But please bear in mind that this procedure cannot be undone and that this person really has the power to make your public key unusable. You really should trust the selected key, in case it is not a key owned by yourself.

5.4.6 Export a Public Key

There are several reasons why to export a public key and there are also several ways to do it. If you want to send the key directly to a mail recipient, you can select the key, right-click, and select "Send Key to Mail Recipient". As an alternative, you can also export it to the clipboard or to a file. To export a key to the clipboard, you can select "Copy key to Clipboard" in the popup menu of the selected key. To export it to a file, you need to select the menu "Key" and then "Export...". The program will automatically suggest a name for the output.

5.4.7 Export your Secret Key

This command should be used with caution because it exports your secret key. Please bear in mind that you should never export your key to a place where it can be accessed by others. An USB stick or a likewise mobile storage device should be used for the export.

5.4.8 Import a Public Key

Similar to the key import, the import of a key can be done in several ways. Let's assume you got a mail with an OpenPGP key included as inline text. Then you can use the current window feature and "Decrypt/Verify" to import the key. Alternative you also may use the clipboard. To achieve this, you first need to select the entire key (CTRL+A) and then copy it to the clipboard (CTRL+C), then use the Key Manager (Edit->Paste) to import it. If the key is stored as an attachment, or you want to import a key from a file in general, just drag the file and drop it into the Key Manager window or use "Key" -> "Import...".

5.4.9 Sign a Public Key

If you verified that a key really belongs to its owner, you should sign the key to integrate it into your Web of Trust and also to mark the key as valid in your keyring. Do not sign a key you just got via email with the request to sign it. Anybody can create a key with your (or better ANY) name, these information are no hint to whom the key really belongs. You can check a key by meeting or calling the key owner and verify the key fingerprint of the key with the one published by the key owner. Additional checks should be to watch at his driver license or the identity card to make sure that name of the key matches the name of the key owner. After this procedure is done, you can open the Key Manager, select the right key and either use the context menu "Sign Key" or use the toolbar button.

The next dialog will summarize the key information and some additional options. For example if the signature should be local or exportable. Local means the signature will be stripped if you export the key and no one else except you can use it to calculate the validity. If you mark the signature exportable, any other user can see and use it. Now you can select the key you want to use to sign and enter the passphrase. Confirm with "OK" and the key will be signed. Now the validity of the new key is "Full". It is probably a good idea to set the ownertrust of the key. For a detailed description, See [section OwnerTrust], page 4.

5.4.10 List Signatures

This dialog contains a list of all signatures of the selected key. The basic dialog, the tree based version, just shows signatures when the issuer key is in the public keyring. A double click opens the signature property dialog which contains detailed description about the selected signature. A dialog which is useful for people who wants to get all information about the key signatures, can click on the "Edit.." button.

5.4.11 Copy Key Information to the Clipboard

Often it is useful to copy parts of the user ID to the clipboard. One example is that you want to send an email to the key owner or that you want to search the key by the email address or you want to copy the fingerprint to the clipboard to paste it somewhere else. This command is available in the popup menu (right click).

5.4.12 Delete one or more Keys

To delete a key, or more than one key, you just need to select the keys in the Key Manager and either select "Delete" or use the toolbar button. Be careful if you delete a key pair, because you will not be able to decrypt and/or sign data any longer. In any case you should have a backup of your key pair at a safe place.

5.4.13 Re-verify Signatures

After you refreshed or imported a lot of new keys, either from a file or the keyserver, it is a good idea to re-verify the signature in the keyring. This speeds up listing operations.

5.4.14 Refresh one or more Public Keys from the Keyserver

From time to time it can be useful to refresh keys from the keyring. The reason for this is, that the key might contain new subkeys, user IDs and or new signatures. It is also possible that the expiration date of a key has been updated or other preferences were changed. And maybe even the worst case, that a key has been compromised and is now revoked. If you want to update a single key, select it and right click on it. Then select the item "Refresh from the Keyserver" in the popup menu. If you do not select any key, the Key Manager assumes that you want to refresh all keys in the keyring. Please bear in mind that this can be a lengthy process if you have a lot of keys in your keyring.

5.5 The File Manager

5.5.1 Introduction

The File Manager is no replacement for an Explorer Extension. If you secure your files frequently and you want to do this fast and easy, I suggest to install GPGee. It is a program which integrates itself into the explorer and provide menu entries in the context menu of files and directory. But the File Manager can be very useful if you just want to decrypt and/or encrypt some files without additional programs. You can find the File Manager via the symbol in the taskbar, right click and then "File Manager".

5.5.2 An Overview of the GUI

First there are different ways to add (open) files in the Key Manager. The easiest way is to use drag and drop to add files into the File Manager. Just drag a file from the explorer and drop it into the File Manager window. The second way is to use File->Open. A dialog opens which is common for all "File Open" operations in most Windows application. Now you can select one or more files and confirm. The files will be automatically added to the File Manager window. The main window consists of a listview with three rows.

The first row is the status of the file. It can be "ENCRYPTED", "SIGNED", "PUBKEY", "SECKEY", "SIG" or "UNKNOWN". Dependent on the file status, the File Manager offers different choices. For example "SIG" enables the verify options in the (popup) menu. "UNKNOWN" is the default for all plaintext files. The second row is the file name. And the last row is the status of the operation. It can be either "", "SUCCESS" or "FAILED". An empty status means no operation was started yet. FAILED indicates that the GNUPG operation failed. In this case an error message was issued before.

We assume that user wants to encrypt C:\My Ideas\GPG GUI.txt. Drag the file from the Explorer and drop it into the open File Manager, the main window. The file will be added and recognized as "UNKNOWN". Now we select the file and right click, a popup menu is shown and we select "Encrypt". An new dialog is opened which looks similar to the Clipboard Encryption dialog. Just select the recipients and confirm. In contrast to clipboard encryption, file encryption offers some more extra options. They are described later. And hour glass will be shown as long as GNUPG takes to encrypt the file. When the procedure is done, the third row should be change to "SUCCESS" and the first row to "ENCRYPTED".

5.5.3 The Key Edit Dialog

For the average GNUPG user, the popup menu of the Key Manager contains all command to manage your keys. For example to add a key/userid/revoker/photo, just right click on the click and select the command from the "Add" submenu. But for advanced users, this dialog contain a lot of extra commands to customize your key.

The main dialog contains a list of all keys in the first list view box and all user IDs in the second list view box. The help button gives you a short hint about each command and what it does. For example you can set the primary user ID via the "primary" command or with "deluid" you can delete the selected user ID. Please always bear in mind, that most keyserver are not

able to remove user IDs in its database so if another user fetch your 'updated' key from the keyserver the user ID might be still part of the key. If you want to make an user ID unusable, you should revoke it. This is also possible with this dialog.

5.5.4 Update your Preferences in the Key Manager

To avoid that the user needs detour to select the taskbar icon, click on it, etc., all preferences can be changed in the Key Manager via the Edit->Preferences... menu.

5.5.5 General Options

Now we describe the general options which are possible in some File Manager dialogs.

- Text Output When this option is checked, the output will be encoded in ASCII armor. This can be useful if the file should be transfered via email. The size of the output file is larger than the usual binary output.
- Wipe Original If this option is checked, the original file will be deleted after successful encryption. This can be useful if data should not be available in plaintext any longer on a machine.

6 WinPT Development and Support

6.1 Development Setup

Basic understanding of Windows C-compiler and knowledge how to use the tools and the Win32 API. There is no need to use MS-Visual C, you can use Ming-W32 (gcc) and a free IDE to hack some code. The default building environment is a mingw32 hosted on Linux and it produces W32 executables.

If you plan to contribute some code or to work on an item from the TODO file, please contact me first to make sure no one else is working on it and that and we can discuss the details.

6.1.1 Getting the Source of the Program

As free software, according to the GNU General Public License, WinPT also offers the source code for the program. It can be used for reviews, to compile your own binary and/or to modify and/or redistribute it or just to learn how it works. The source is available at the same place you downloaded the binary. If not, you should contact the author of the site. The entire program can be build with free software; the default environment is a cross-compiler hosted on a Linux box. All you need is the mingw32 packages, a working autoconf environment and the libs WinPT depends on (currently gpgme and libgpg-error). It is also possible to build the binary with cygwin/mingw32 on Windows but this environment is not actively supported and probably needs adjustment of the source.

6.1.2 Native Language Support

The program has the ability to select different languages to provide dialogs and error messages in the native language of the user. Currently German, Japanese, Portuguese (Brazil) and Slovak. When WinPT has been installed via a graphical installer, for example Gpg4Win, the language was automatically selected based on the locale Windows environment. If the stand-alone binary was downloaded, WinPT offers at the first start to select a language, based on the .mo file it founded in the current directory. Otherwise the user needs to perform the following steps. The WinPT ZIP archive contains various .mo files (de.mo, jp.mo, sk.mo) and the user needs to find his native language, if available and rename the file to "winpt.mo". For example, if the user prefers German, "de.mo" -> "winpt.mo". Now the user needs to save the locale dir, where the winpt.mo is stored, in the GNUPG preference dialog.

6.1.3 A short Note about Cryptographic Issues

WinPT itself does not perform any real encryption, signing or decryption. Instead it uses GNUPG as the backend program which provides all kind of cryptographic code to perform the needed operations.

The default values WinPT uses for key sizes, should be sufficient for personal and commercial security for the next years. If you are concerned about the default values, you can always use the expert key generation to make your own decision. GNUPG also provides default values for symmetric cipher preferences. By default, the AES (Advanced Encryption Standard) is used which provides a very good security. You can manually modify your key preferences, this includes cipher, hash, and compression but usually this is not necessary and also can do harm if you use algorithms which are not very widespread among other OpenPGP programs.

6.2 Free Support and Commercial Support

Currently the core WinPT crew has only one developer, Timo Schulz. So, it will take some time to respond to forum messages, and mails. But as a free software project, you are welcome to participate in the WinPT community as a developer or as an end user.There are several ways to help the project. For example you could provide (or work on) the existing documentation or write new docs. You could translate WinPT into a new language or maintain an existing language file. Of course it is also possible to contribute code or to become part of the WinPT developer crew.

If you need commercial support for WinPT or GNUPG in general, please contact g10 Code GmbH http://www.g10code.com

6.2.1 Reporting a Problem (Bug) or a Feature Request

If there is any problem, that includes crashes or the handling, go to please first check the forum at http://wald.intevation.org/projects/winpt to see if someone else reported and/or wrote about the issue. It is possible that the issue is already solved/answered in the forum. Plus all other users can benefit of it because maybe another person has the same problem and then he can check the forum and will find the answer.

Feature requests can be submitted at the same site in a different tab (Tracker->Feature Request). There is no guarantee that the request will be implemented in the next version. The reason is, that other issues might be more important or that the request must be first discussed with other developers. But each request will be considered.

For the case that you found a bug, it is very important to provide much details as possible to allow the developers to track down the problem and to fix it easily. Please do not forgot to be precise as possible and the best idea is to provide a step-by-step text to reproduce the problem.

6.2.2 Problem with the Program or an unexpected behaviour

First let me say that it is very important always to use the newest version. Each new version contains bug fixes and might also fix usability issues. This is also valid for GNUPG, WinPT checks that the minimum GNUPG version is available but even so it is important and often useful to have the newest GNUPG version if this is possible.

But sometimes the problem is not the software itself, but the software which was involved to transfer the data. Here are some examples of what could happen:

- The downloaded file could be broken (FTP ascii->binary issue) and thus WinPT is unable to verify the signature. In this case you should download the file again.

- A mailer broke the signature because the line endings were altered or the mail text was wrapped after the signature was issued. There is no solution to this problem, except to use a smart Mail Client.

- A public key (file or clipboard) will not be recognized but the data should definitely contain one or more keys. Sometimes line endings are messed up or white spaces were removed. In this case GNUPG/WinPT is not able to detect when the data begins and the header section starts. You can use the clipboard editor to see if the ascii armor is broken. If this happened, the file must be repaired manually or should be sent again.

- WinPT reports that the key could not be imported because of missing self signature or a likewise message. To make sure that the receiver can really verify the key belongs to its owner, the key carries a self signature which can be checked by anybody. Some PGP 2.6 version do not issue this self signature and some other PGP versions might be also able to suppress its generation. Such a key cannot be used, even if the import were forced. The solution to this problem is easy but sometimes not possible. Ask the key issuer to self sign his key and to upload it to the keyserver or send it again. But sometimes companies have a policy and thus newly generated keys are not self signed. I do not know what to do in this case except for asking if it would be possible to sign a copy of the key.

- You received a message from a user which uses PGP and WinPT/GNUPG will not be able to decrypt it. First let me say that this should happen very seldom with newer (PGP >= 7) versions of PGP. The reason could be, that IDEA has been used. A patented Cipher which is not included in GNUPG. GNUPG will not be able to decrypt the data because it has been ciphered with IDEA. There is no solution for this problem, except to use the IDEA plug-in. But be advised that the IDEA algorithm is only free for private use and NOT for commercial mails.

Another problem could be, that your files cannot be automatically decrypted by the receiver (who uses PGP) because the file extension of it is .GPG. You can solve this problem by changing the default extension in the WinPT preferences from .GPG to .PGP.

To minimize the change of problems when you communicate with a PGP user, you can add "pgp8" or "pgp7" to your gpg.conf. This can be done via the Key Manager ->Edit->Preferences...->GPG Config Preferences.

7 About this document

This document source is text file created in Texinfo format. So, the source file will have .texi extension. Texinfo format is the default documentation format for GNU projects. From one source file - which includes texinfo format commands and contents - different types of outputs (like *info, html, pdf, dvi, ps*) can be created. However, source file for this document is mainly created for PDF output only, and it does not have texinfo commands to produce appropriate *info* or *html* documents.

7.1 Special commands

The following TEX commands are used in this document, and you can't find them in Texinfo manual.

`@par @allowbreak`

These commands are used to avoid vertical centralising of images which creates unwanted vertical space.

7.2 Customizing the texinfo.tex

When printing the Screen caption, it always appeared as left aligned and there is no option to center the caption. So, the texinfo.tex has been modified to print the caption always in the center.

7.3 feedback

This document is for newbie. So, obviously it won't contain advanced topics. Please keep this in mind, while you provide your feedback.