3 How Gpg4win works

The special feature of Gpg4win and its underlying **"Public Key" method** is that anyone can and should understand it. There is nothing secretive about it - it is not even very difficult to understand.

The use of individual Gpg4win program components is very simple, even though the way it works is actually quite complicated. This section will explain how Gpg4win works - not in all details, but enough to explain the principles behind this software. Once you are familiar with the principles, you will have considerable trust in the security offered by Gpg4win.

At the end of this book, in Chapter 24, you can also open the remaining secrets surrounding "Public Key" cryptography and discover why it is not possible to break messages encrypted with Gpg4win using current state of technology.

Lord of the keyrings

Anyone wishing to secure something valuable locks it away - with a key. Even better is a key that is unique and is kept in a safe location.

If the key should ever fall into the wrong hands, the valuables are no longer secure. Their security stands and falls with the security and uniqueness of the key. Therefore the key must be at least as well protected as the valuables themselves. To ensure that it cannot be copied, the exact characteristics of the key must also be kept secret.

Secret keys are nothing new in cryptography: it has always been that keys were hidden to protect the secrecy of the messages. Making this process very secure is very cumbersome and also prone to errors.

The basic problem with the "ordinary" secret transmission of messages is that the same key is used for both encryption and decryption, and that both the sender as well as recipient must be familiar with this secret key. For this reason, these types of encryption systems are also called **"symmetric encryption"**.

This results in a fairly paradoxical situation: Before we can use this method to communicate a secret (an encrypted message), we must have also communicated another secret in advance: the key. And that is exactly the problem, namely the constantly occuring issue of always having to exchange keys while ensuring that they are not intercepted by third parties.

In contrast - and not including the secret key - Gpg4win works with another key that is fully accessible and public. It is also described as a "public key" encryption system.

This may sound contradictory, but it is not. The clue: It is no longer necessary to exchange a secret key. To the contrary: The secret key can never be exchanged! The only key that can be passed on is the public key (in the public certificate) - which anyone can know.

That means that when you use Gpg4win, you are actually using a pair of keys - a secret and a second public key. Both key components are inextricably connected with a complex mathematical formula. Based on current scientific and technical knowledge, it is not possible to calculate one key component using the

other, and it is therefore impossible to break the method.

Section 24 explains why that is.

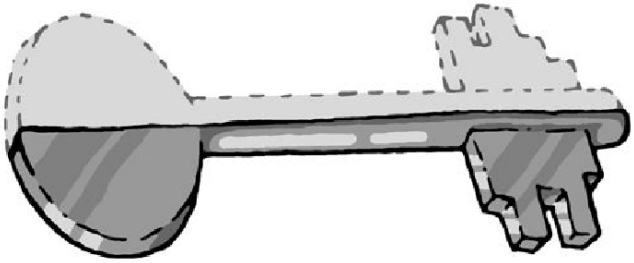
The principle behind public key encryption

The **secret** or **private key** must be kept secret.

The **public key** should be as accessible to the general public as much as possible.

Both key components have very different functions:

The secret key component **decrypts** messages.



The public key component **encrypts** messages.

The public mail strongbox

This small exercise is used to explain the difference between the "public key" encryption system and symmetric encryption ("non-public key" method) ...

The "secret key method" works like this:

Imagine that you have installed a mail strongbox in front of your house, which you want to use to send secret messages.

The strongbox has a lock for which there is only one single key. No one can put anything into or take it out of the box without this key. This way, your secret messages are pretty secure.

Since there is only one key, the person you are corresponding with must have the same key that you have in order to open and lock the mail strongbox, and to deposit a secret message.

You have to give this key to that person via a secret route.

They can only open the strongbox and read the secret message once they have the secret key.

Therefore everything hinges on this one key: If a third party knows the key, it is the end of the secret messages. Therefore you and the person you are corresponding with **must exchange the key in a manner that is as secret** as the message itself.

But actually - you might just as well give them the secret message when you are giving them the key...

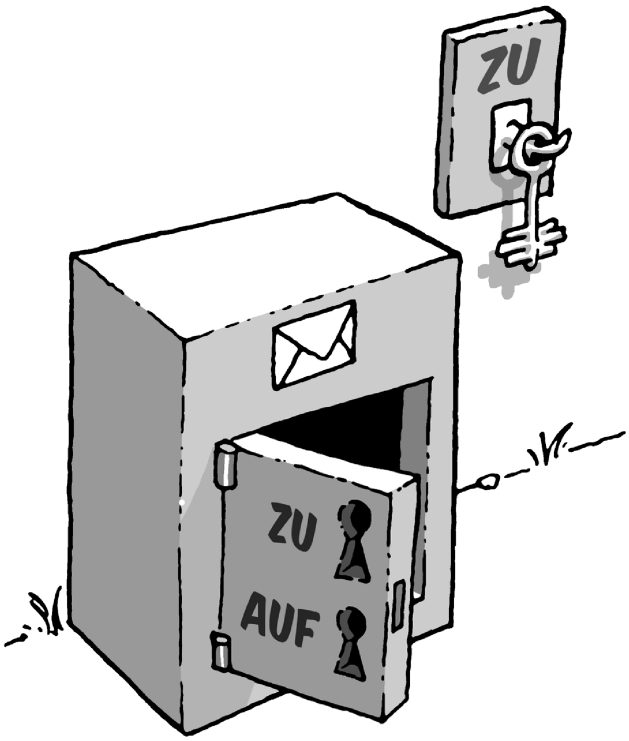
How this applies to e-mail encryption: Around the world, all participants would have to have secret keys and exchange these keys in secret before they can send secret messages per e-mail.

So we might as well forget about this option ...

Now the "public key" method

You once again install a mail strongbox in front of your house. But unlike the strongbox in the first example, this one is always open. On the box hangs a key - which is visible to everyone - and which can be used by anyone to lock the strongbox (asymetric encryption method).

Locking, but not opening: that is the difference!



This key is yours and - as you might have guessed - it is your public key.

If someone wants to leave you a secret message, they put it in the strongbox and lock it with your public key. Anyone can do this, since the key is available to everyone.

No one else can open the strongbox and read the message. Even the person that has locked the message in the strongbox cannot unlock it again, e.g. in order to change the message.

This is because the public half of the key can only be used for locking purposes.

The strongbox can only be opened with one single key: your own secret and private part of the key.

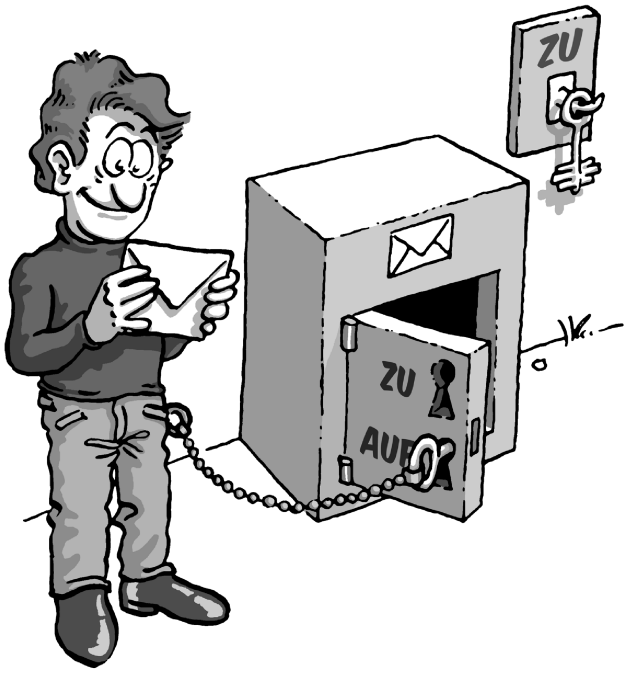
Getting back to how this applies to e-mail encryption: Anyone can encrypt an e-mail for you.

To do this, they do not need a secret key; quite the opposite, they only need a totally non-secret , "public" key. Only one key can be used to decrypt the e-mail, namely your private and secret key.

You can also play this scenario another way:

If you want to send someone a secret message, you use their mail strongbox with their own public and freely available key.

To do this, you do not need to personally know the person you are writing to, or have to speak to them, because their public key is always accessible, everywhere. One you have placed your message in the strongbox and locked it with the recipient's key, the message is not accessible to anyone, including you. Only the recipient can open the strongbox with his private key and read the message.



But what did we really gain: There is still a secret key!

However, this is quite different from the "non-public key" method: You are the only one who knows and uses your secret key. The key is never forwarded to a third party - it is not necessary to transfer keys in

secret, nor is it advised.

Nothing must be passed between sender and recipient in secret - whether a secret agreement or a secret code.

And that is exactly the crux of the matter: All symmetric encryption methods can be broken because a third party has the opportunity to obtain the key while the key is being exchanged.

This risk does not apply here, because there is no exchange of secret keys; rather, it can only be found in one and very secure location: your own keyring - your own memory.

This modern encryption method which uses a non-secret and public key, as well as a secret and private key part is also described as "asymmetric encryption".
