

[Foswiki](#) > [Sysadmin Web](#) > [CmiBrasilTech](#) > [GnuPGpt \(13 Sep 2007, AlsteR\)](#)

## Manual de Criptografia

Este é um tutorial de criptografia usando o GNU Privacy Guard (GPG). Para você entender legal o que está escrito aqui é recomendável alguma experiência com edição de textos no computador, saber navegar na internet e utilizar seu correio eletrônico. Outros conhecimentos de computação são opcionais e muito bem vindos.

Esse texto está dividido em cinco partes:

- [Introdução e conceitos de criptografia baseada em chaveiro](#)
- [Instalando programas de criptografia](#)
- [Usando os programas de criptografia](#)
- [Criptografia e internet: e-mail, conexão segura e proteção de software](#)
- [Protegendo seu par de chaves: criptografando seu disco rígido](#)

A leitura da primeira parte é indispensável para compreender o resto do texto. As outras seções podem ser lidas independentemente.

## Índice

[Manual de Criptografia](#)

[Índice](#)

[1. Introdução e conceitos de criptografia](#)

[O que é a criptografia:](#)

[Como funciona](#)

[Exemplo de chave pública](#)

[Exemplo de chave privada](#)

[Exemplo de mensagem criptografada](#)

[Fingerprint \(Impressão Digital\)](#)

[Assinaturas digitais](#)

[O limite da confiabilidade](#)

[Repositórios de chaves](#)

[Histórico](#)

[2. Instalando programas de criptografia](#)

[Programas de criptografia](#)

[Instalando o GPG](#)

[Modo Texto](#)

[GNU/Linux](#)

[Windows](#)

[Modo Gráfico](#)

[GNU/Linux](#)

[Windows](#)

[3. Usando os programas de criptografia](#)

[Modo Texto](#)

[Como criar seu par de chaves](#)

[Como compartilhar sua chave pública](#)

[Como adicionar uma chave pública de alguém na sua lista](#)

[Listando seu chaveiro](#)

[Como assinar mensagens e arquivos](#)

[Como verificar mensagens assinadas](#)

[Como codificar uma mensagem para alguém](#)

[Como decodificar uma mensagem que enviaram para você](#)

[Verificando Impressões Digitais e Assinando Chaves](#)

[Recebendo sua chave assinada](#)

[Confiando em chaves](#)

[Removendo chaves](#)

[Cancelando um par de chaves](#)

[Outros comandos](#)

[Modo Gráfico](#)

[Linux: usando o GPA](#)

[Criando um novo par de chaves](#)

[Gerenciando chaves públicas](#)

[Encriptar](#)

[Desencriptar](#)

[Assinando arquivos](#)

[Como verificar mensagens assinadas](#)

[Windows: usando o WinPT](#)

[Como gerenciar chaves públicas](#)

[Como importar uma chave pública](#)

[Como exportar uma chavé pública](#)

[Como importar ou exportar uma chave pública de um servidor](#)

[Encriptar textos com Ctrl-c ou Ctrl-x](#)

[Desencriptar textos com Ctrl-c ou Ctrl-x](#)

[Encriptar e desencriptar arquivos](#)

[Windows: usando o GP Gee](#)

#### [4. Criptografia e internet](#)

[Criptografia e correio eletrônico](#)

[Programas de email](#)

[Mozilla Mail / Thunderbird](#)

[Instalando no Mozilla](#)

[Instalando no Thunderbird](#)

[Importando chaves públicas](#)

[Assinando mensagens](#)

[Como verificar mensagens assinadas](#)

[Enviando mensagens encriptadas](#)

[Recebendo mensagens criptografadas](#)

[Usando o KMail](#)

[Criptografia em listas de discussão](#)

[Conexão segura: criptografia na rede](#)

[Criptografia como proteção de software](#)

#### [5. Criptografando seu disco rígido](#)

[No Linux](#)

[No Windows](#)

[Nota sobre Modo Texto e Modo Gráfico](#)  
[Resumão: tabela de consulta rápida](#)  
[Referências](#)  
[Sobre este manual](#)  
[Mais informações](#)

## 1. Introdução e conceitos de criptografia

### O que é a criptografia:

A criptografia é o método de codificar mensagens de modo a garantir que só a pessoa que tenha em mãos o código correto possa lê-la.

*Para que serve a criptografia?* Em resumo, ela serve para

- Proteger informações
- Preservar a privacidade das pessoas
- Permitir a autenticidade de informações (assinatura digital)

Proteger informações significa que você pode escolher quem acessa suas informações. Preservar a privacidade das pessoas quer dizer que quem não estiver autorizado a acessar suas informações não conseguirá decifrá-las. Permitir a autenticidade de informações evita que falem em seu nome coisas que você não disse.

### Como funciona

A criptografia que veremos aqui é baseada num princípio de dificuldade de realizar operações reversíveis. Por exemplo, é muito fácil saber qual é o produto de dois números primos - números que só são divisíveis por 1 e por eles mesmos, como 7, 11, 23. Mas é realmente muito difícil saber, dado um número qualquer, quais são os números primos que eu preciso multiplicar para obter esse número. Ou seja, essas operações matemáticas com números primos são reversíveis - se eu sei quais são os números primos eu sei o produto deles e se eu sei qual é o número eu posso descobrir quais são seus fatores primos e são mais difíceis de se realizar de um sentido do que de outro.

A dificuldade de descobrir quais são os fatores primos de um número aumenta exponencialmente com o tamanho do número. Exponencialmente quer dizer realmente muito. É nisso que a criptografia se apóia.

Imagine um cadeado com duas chaves. Uma somente fecha o cadeado e a outra somente abre. É mais ou menos assim que o GPG funciona. Você criará duas chaves, uma pública e outra privada. A chave pública é a que "fecha" (criptografa) e a chave privada é a que "abre" (decifra) depois que você fornece sua frase-senha.

As pessoas que querem enviar documentos critografados para você devem ter a sua chave pública e **SOMENTE VOCÊ** deve ter sua chave privada e deve saber sua frase-senha. De modo análogo, se você quiser mandar uma mensagem codificada para uma pessoa, você precisa ter a chave pública da mesma.

O esquema de criptografia baseado em par de chaves funciona assim:

```
mensagem original -> chave pública -> mensagem codificada  
mensagem codificada -> chave privada -> mensagem original
```

Essas operações são reversíveis, ou seja, é possível termos, por exemplo

```
mensagem codificada -> chave pública -> mensagem original
```

o que, em outras palavras, significa a "quebra" da mensagem codificada e consequentemente a invasão da sua privacidade, pois alguém pode interceptar a mensagem codificada e junto com sua chave pública determinar qual é a mensagem original. **Calma, não se assuste!** Apesar disso ser perfeitamente possível, devemos lembrar que essas operações são reversíveis mas também são mais fáceis de fazer num sentido do que em outro, isto é, é muito mais fácil fazer:

```
mensagem original -> chave pública -> mensagem codificada
```

do que

```
mensagem codificada -> chave pública -> mensagem original
```

Pra você ter uma idéia, a primeira operação demora tipicamente alguns segundos, dependendo do tamanho da mensagem e do tamanho da chave. Já a segunda demora tipicamente **milhares de anos** (literalmente) e precisa de um poder computacional enorme, de milhares de computadores ou supercomputadores trabalhando juntos. Essa dificuldade de fazer a operação reversa é que possibilita o uso da criptografia como ferramenta de privacidade pessoal, pois na prática é inviável quebrar mensagens criptografadas.

A operação

```
mensagem codificada -> chave privada -> mensagem original
```

também demora apenas alguns segundos para ser realizada.

**Resumindo:** a criptografia tratada neste pequeno manual é baseada em pares de chaves, uma **chave pública** e uma **chave privada**. Você troca sua chave pública com outras pessoas e mantém sua chave privada em segredo. Se você quiser mandar uma mensagem codificada para alguém, é só usar a chave pública dessa pessoa para criar uma mensagem que só ela poderá ler. De mesma forma, se alguém quiser lhe enviar uma mensagem codificada, basta que essa pessoa tenha a sua chave pública e só você será capaz de ler a mensagem, e para isso você deverá usar sua chave privada. **É muito importante que você tenha esse princípio de funcionamento bem claro em sua mente antes de continuar.**

A coleção de sua(s) chave(s) pública(s) e privada(s) e mais as chaves públicas de outras pessoas que você possui é denominada de chaveiro.

## Exemplo de chave pública

Uma chave pública é uma sequência de códigos, e pode ser apresentada como um arquivo de texto comum. Esse arquivo conterá um monte de caracteres malucos. Veja só:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.0.6 (GNU/Linux)
Comment: For info see http://www.gnupg.org

mQGiBD6cuEoRBACQ6QKhCjN2qKHM0ee0dW9w0nnnd9V0eLREreSmMsRD6kSdXnrGLQGrXMCBjelw0KB/vh2mcKn646PmSaq4sC+bLSOMAhME/IaDyDWZNWXo37yYlhDuVD5wZGZNGMwov/bPDvJnjJTSXlo4/glbUNyVU2n9kiFYoDoSGjg1jQcYLwCgye0JtHFSoelA62s+YmypI3KAzkcD/1S7Fim4p5X6HA/mUkFtuExDggba+0KmsmYNfPZ/Q0sPAK850r0zKAbGXAKUqezCuKCZ6FaesbIU5hTcQb2zKmzg1hzmtsWCTrxuj2X8c//yRjDzFjT8KKCYKZWBGQTapGhvlHHq8ZsLlYZRzYyYtZ13a809FWk8G0Aq5/FAbe3KA/4nK+XAZUwXqXzu+xKINMu98zi9QWnVfBA5G132uCyrw0TBG1BV3803QFxGXdBrEnGlz/RU9xjkVnEBWyqri0+lgDXuEZ6pWyx70UJ1S2qPDYKxoLTB68ZWafKUxYP8JYBC06hk6Ztq7Fj kQGHxKMYQ9HCC3l9octyNsux+b/50+rQhU2lsdmIvIFJoYXR0byA8cmhhhdHRvQHJpc2V1c5uZXQ+iFcEEExECABcFAj6cuEoFCwckAwQDFQMCAxYCAQIXgAAKCRDA5viFRw5zwno5AKCGTBDCw9dalpr+SG7MQx00u1iehwCeKuGC1FljdhHcu2CQX/JzoWLb/Qi5AQ0EPpy4TRAЕAKmsLjxi2k/ITXu2kZJ7+SQPftb9yRs8Fx0snmytkUw08HaryPfuoMU51xG8XYL4b1GL6u2J67KJ004R3buwUDmH16RC+nmMNlnaa0zlozsYIuB+r3s18hNLAss1LX0P00b60wnar5VM8yNgVhEZkwBs6VhVlfinYThWmpaxXLU3AAMFA/498Mfl1rs4z6vzkmIGu3Mqy+2CXSA/oCp9zPffLJNM+WUGhpbxkbbsNEzHdTfWPcoHi23k01KjT5CAiyiP30o6g80V+3/WRYqeR4UNT6e87JDZo8kzjnTigI7XoAkqTjhL8pzhzvjbogZAa1LDPg02H//iRaBUjjrGa0Tl9x2c4hGBBgRAgAGBQI+nLhNAoJEMDm+IVHDnPcsrYAoL9sL0bVCteWmkAPFL3b5e/pUFFAAKCzRRY3tPu6sHczFz0cw3Szeden5x5kBog0+nLhKEQQAk0kCoQozdqih5jnnjnVvcDp553fVdHi0RK3kpjLEQ+pEnV56xi0Bq1zAgY3pcDigf74dpnCp+u0j5kmquLAvmmy0jjAITBPYGg8g1mTVl6N+8mJYQ7lQ+cGRmTRjMKL/2zw74zYyU0l5a0P4JW1DclVNp/ZIhWKA6Eho4NY0HGC8AoMnjibRxUqHpQ0trPmJsqSNygM5HA/9UuxYpuKeV+hwP5lJBbbhMQ4IG2vji prJmDXz2f0NLDwCv0Tq9MygGxlwClKnsrwrigmehWnrGyFOYU3EG9sysps4NYc5rbFgk68bo9l/HP/8kYw8xY0/CigmCmVgRkE2qRob5Rx6vGbC5WGUC2MmLWdd2vNPRVpPBtAKufxQG3tygP+JyvlwGVMF6l87vsSiDTLvfM4vUFp1XwQ0Rtd9rgsq8DkwRtQVd/NN0BcRl30axJxpc/0VPcY5FzxAVsqq4jvpRg17hGeqVsse9FCdUtqjw2CsaC0wevGVgHyLMWD/CWAQtOoZ0mbauxY5EBh8SjGEPRwgt5faHLCjbLsfm/+UPq0IVNpbHZpbyBsAaGF0dG8gPHJoYXR0b0ByaXNldXAubmV0PohXBBMRAgAXBQI+nLhKBQsHCgMEAxDAGMWAgECF4AACgkQw0b4hUc0c8J60QCghkwQwsPXWpaa/khuzEMdNLtYnocAnirhgtRZY3YR3LtgkF/yWaFiwf0IuQENBD6cuE0QBACprC48SNpPyE17tpGSe/kkD37W/ckbPBcTrJ5srZFMdvB2q8j37qDF0dcRvF2C+G5Ri+rtieuyiTjuEd27sFA5h9ekQvp5jDZZ2mtM5aM7GCLgfq97NfITSwLLNS19D9Dm+jSJ2q+VTpmjYFYRGZMAb0lYVZX4p2E4VpqWl1y1NwADBQP+PfDH5da70M+r85JiBtzKsvtgtl0gP6Aqfcz3xSyTTPllBoaW8ZG27DRMx3UxVj3KB4tt5NNSo0+QgIsoj99K0oPDlft/1kWKnkeFDU+nv0yQ2aPJMJ4504oC016AJKkyYS/Kc4c7426BmQGjdSwz4Dth//4kWgVI46xmjk5fcn0IRgQYEQIABgUCPpy4TQAKCRDA5viFRw5zwrK2AKC/bCzm1QrXlppADxS92+Xv6VBXwACgs0UWN7T7urB3MxcznMN0s3gz
```

```
ecc=
=1J0v
-----END PGP PUBLIC KEY BLOCK-----
```

Esta é a minha chave pública. Ela é grande e bonita, mas não muito compreensiva. Para falar a verdade, eu não entendo nada do que está escrito lá, com a exceção dos seguintes pedaços:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

Esse primeiro texto vai informar ao leitor que este é o começo da chave pública.

```
Version: GnuPG v1.0.6 (GNU/Linux)
Comment: For info see http://www.gnupg.org
```

Essas duas linhas informam sobre qual programa que criou a chave pública e em qual sistema operacional.

```
-----END PGP PUBLIC KEY BLOCK-----
```

Essa linha avisa ao leitor que a sequência de caracteres malucos da chave pública acabou.

Quando você vai mandar uma mensagem criptografada pra alguém, o programa de criptografia vai usar os caracteres malucos que ficam entre os textos -----BEGIN PGP PUBLIC KEY BLOCK----- e -----END PGP PUBLIC KEY BLOCK----- da chave pública da pessoa pra criar a mensagem codificada.

**OBS:** Não há nenhuma restrição quanto ao nome de arquivo que uma chave pública pode ter. Usualmente é um nome de arquivo com extensão .asc, .key ou qualquer outra coisa. Por exemplo, o arquivo de chave pública do seu amigo Groucho pode ser *groucho.asc*, *groucho.key* ou qualquer outra coisa. Se você quiser conferir se um arquivo contém uma chave pública, basta abri-lo num editor de textos qualquer e ver se o conteúdo do arquivo se parece com o exemplo de chave pública acima.

## Exemplo de chave privada

Uma chave privada também é uma sequência de códigos que também pode ser apresentada como um arquivo de texto comum. Esse arquivo conterá um monte de caracteres malucos. Aqui segue um exemplo fictício de chave privada:

```
-----BEGIN PGP PRIVATE KEY BLOCK-----
Version: GnuPG v1.2.1 (GNU/Linux)

mQGiBD6cuEoRBACQ6QKhCjN2qKHm0ee0dW9w0nnnd9V0eLREreSmMsRD6kSdXnrG
LQGrXMCBjelw0KB/vh2mcKn646PmSaq4sC+bLS0MAhME/IaDyDWZNWXo37yYlhDu
VD5wZGZNGMwov/bPDvjNjJTSXlo4/glbUNyVU2n9kiFYoDoSGjg1jQcYLwCgye0J
tHFSoelA62s+YmypI3KAzkcD/1S7Fim4p5X6HA/mUkFtuExDggba+0KmsmYNfPZ/
Q0sPAK850r0zKAbGXAKUqezCuKCZ6FaesbIU5hTcQb2zKmzg1hzmtsWCTrxuj2X8
```

```
c//yRjDzFjT8KKCYKZWBGQTapGhv1HHq8ZsL1YZRzYyYtZ13a809FWk8G0Aq5/FA
be3KA/4nK+XAZUwXqXzu+xKINMu98zi9QWnVfBA5G132uCyrw0TBG1BV3803QFxG
XdBrEnGlz/RU9xjkVnEBWyqrIo+lGDxuEZ6pWyx70UJ1S2qPDYKxoLTB68ZWafKU
xYP8JYBC06hk6Ztq7FjkQGHxKMYQ9HCC3l9octyNsux+b/5Q+rQhU2lsdm1vIFJo
YXR0byA8cmhhhdHRvQHJpc2V1cC5uZXQ+iFcEEExECABcFAj6cuEoFCwcKAwQDFQMC
vUFp1XwQ0Rtd9rgsq8DkwRtQVd/NN0BcRl3QaxJxpc/0VPCy5FZxAVsqq4jvpRg1
7hGeqVsse9FCdUtqjw2CsaC0wevGVgHyLMWD/CWAQtOoZ0mbauxY5EBh8SjGEPRw
gt5faHLcjblsfm/+UPq0IVNpbHZpbyBSaGF0dG8gPHJ0YXR0b0ByaXNldXAubmV0
NS19D9Dm+jSj2q+VTPMjYFYRGZMAb0lYVZX4p2E4VpqWl1y1NwADBQP+PfDH5da7
0M+r85JiBrtzKsvtgtl0gP6Aqfcz3xSyTTPlBoaW8ZG27DRMx3UxVj3KB4tt5NNS
PohXBBMRAgAXBQI+nLhKBQsHCgMEAxDAGMwAgECF4AACgkQw0b4hUc0c8J60QCg
hkwQwsPXWpaa/khuzEMdNLtYnocAnirhgtRZY3YR3LtgkF/yWaFiwf0IuQENBD6c
uE0QBACprC48SNpPyE17tpGSe/kkD37W/ckbPBcTrJ5srZFMDvB2q8j37qDF0dcR
vF2C+G5Ri+rtieuyiTjuEd27sFA5h9ekQvp5jDZZ2mtM5aM7GCLgfq97NfITSwLL
7JDZo8kzjnTigITXoAkqTJhL8pzhzvjb0GZAaN1LDPg02H//iRaBUjjrGa0Tl9x2
c4hGBBgRAgAGBQI+nLhNAoJEMDm+IVHDnPCs rYAoL9sL0bVCteWmkAPFL3b5e/p
1FljdhHcu2CQX/JZoWLb/Qi5AQ0EPpy4TRAEKmsLjxI2k/ITXu2kZJ7+SQPftb9
ecc=
=1J0v
-----END PGP PRIVATE KEY BLOCK-----
```

Como você deve ter percebido, uma chave privada é uma porção de texto cuja estrutura é igual à da chave pública. As primeiras duas linhas,

```
-----BEGIN PGP PRIVATE KEY BLOCK-----
Version: GnuPG v1.2.1 (GNU/Linux)
```

informam ao programa de criptografia que o texto que vem na sequência é uma chave privada. O programa assumirá que o resto do texto é uma chabe privada até encontrar a linha

```
-----END PGP PRIVATE KEY BLOCK-----
```

## Exemplo de mensagem criptografada

Como exemplo, que tal criptografarmos a seguinte mensagem:

```
Seu pai é careca?
```

O resultado, quando criptografado com minha chave pública, é

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.2.3 (GNU/Linux)

hQE0A3v8xQeh8DSxEAP/Rks1pm5W0yNZTtlJP1FVK602ix872ZMws0s0YZ9oBC+o
```

```
pN9/03jLjlk3hF0v9KJkq3SQxu+E7zSCbuMigAum0QZgV3BMVjZDGKDUN3D8Sgt  
fBju6Y7Vn6wK870WQlayWMAab+t77wQPjKuH9IPYJwsZ6zJK5YIhgILKU0FoKAD  
/RWDMqQua/iswijTh5Stg0+FJuseec22TgGPzZC4D05s9JE7gMI03GvSTGR+re4i  
SaCzrppudx0acsuCcRhR6IkA6lT8fKaZImU/aMev/UgwadGP3XeqiwxPUT+qHg6H  
iYZyMShKAMZzF+PdgflaDYGffgIQBXpxWxjRMufX9LTW0LYBo1mQ6W1XgJG3AY47  
knPyqSKRfuNPwayAGRdZM2yoq/DnwIUGB5cyfiNqkv9lX125uXud99T3mDojYLzr  
N/pWTHTdu5UA/RbBrPaeZ7HX3tdXnhlCzw==  
=Jqqw  
-----END PGP MESSAGE-----
```

E aí, você acha que dá pra decifrar? Só com minha chave privada!

As primeiras duas linhas,

```
-----BEGIN PGP MESSAGE-----  
Version: GnuPG v1.2.3 (GNU/Linux)
```

indicam que o que vem na sequência é uma mensagem criptografada. O fim da mensagem criptografada é assinalada na linha

```
-----END PGP MESSAGE-----
```

Agora vou codificar a mesma mensagem utilizando várias chaves públicas, para que mais pessoas possam lê-la.

```
mensagem original -> várias chaves públicas -> mensagem cifrada para vários destinatários
```

O resultado é

```
-----BEGIN PGP MESSAGE-----  
Version: GnuPG v1.2.3 (GNU/Linux)  
  
hQE0AwXhZ7S+8an6EAP8CwSfq+TjKziYiM4QGwAwNynKDLSkMdw6KeTmSi3XgfP+  
Wf6p6KWGuVmD7ffGMwCBuB+mKpjmxhZ9EyR4JAVpgkMMN2ZCrTcMu4tDZqqvJV/t  
jws1dwg0QaeCdmA6jbxBDuWUsbg0zxY/5URmttCfKGTgI00h73VDZ3HXM4MrWIcE  
AKFaCgiMybhYZLlXqEpqgyJWH54fsXSgRbv2yZE2G+9aNEoLAVji7NfluKGhrosF  
gRCeUFc1pGRnF+Z5+aMj/xK/9DKeqGt97aQbViQ33s/3soqsKBWzd95rKK1ZyBRn  
iIEoEXlc+eDRK+j6EsjR+Ex87fU4gCM9aNPQGkdLZZpJhQE0Azrq0korX+quEAQA  
g0EV5STP2zCgQ5sSljd6D021LXVky5P/4vvgBPUUpH9svMQUF5iJTPc8H/YUo0mW  
Hpmk6nwXDsaWDhyotuiiPHRBd+D1ckCtwsjRZeP0hkU2r0m+9k4kI3Q34XX3jPDI  
YABHRWP0tJaazCU/1PD1rKA0JWolmA3x3i2AEizwLrkD/R+8xBg0UEV+XNJSEXqS  
ooHbKCu9dItIXjHj2kzTpA5F9anT0xZYG+Q9SivWavZVAd8KLcDM7My1s/udjGyp  
KWeSxycHu6rYmdJw1by+eWddKn+2W7iSgna0hjEimZcy53wYmrh5Wzv5vDL0Qd2d  
RSX12GjdqW9Fm0Qvag2q8sIAhQI0A7jcJ6nlDz0bEAf9EPB7HaCBFSovTp/c2GsU
```

```
HEyc3gRQ2RD1XPlnKA/PhE5zTIGCXiRdVLfQuoZ4x0wHng8ppM5xJkIU5oohhysu  
m7YFlkM7rG9kavVAoATDeNfoLXjA7q/Cntl9xyy2NEUU9oQCilrS6JTvJ9UPPY1u  
kW9zS8CNNIAgIQ+rVJZtc1E+34N8x0Rk+4nQz40tY0nE6XMN83bIYfoIxGc0dn80  
PV0z5ej6utcZV2sdTZc0JnX3lXonUhkSCvsDcxRfu0emf0bR2XZMq70rFoKQ1mly  
TQ9AW2mUyEbEtVsFdT8H0AC3M7z/EmnK97nAyk0NYko1RrHgWY/ekhElqxvfzy2I  
6Qf9F9tTLshUy7QdM7WiGoJaxPwTUwlzGw29rnZo36JNKyWW8qnGU5+YH5iKzM4u  
xeA0t12SKcBgzN2ucEdm0xm0A1rYL0IgDLWTYnD/YkmJbSp4xhwKEUEF4juiJdUd  
ZTf1+tF6Q6yIRIFwLy3ES3NX6sA/dxkin2YR0C41VtcGqn84lCj3c4VnMw7LcLtX  
GBP9y76t0bNEXsCYGIsvqjxMXDFh0smQ8v1Loo1qFxBCjuPXwWBWL77814o2rS+Y  
1eH+Q9rv5RajMygDD1afYxKo1mey7yuRbiG02owHDNm6wECCH77bRnxAVRLSWD  
UCsIRMk0qWKApD/M7qIWn4qt19JwAcG4KKr0GdIYswARUoJIMKn+TMJmbA8TxTKo  
J7yrXniv3C9pyeEPqRUWdZqo4szVDKPLz+VMCV4tzSE5iINGBNNXAM9HxqwmAX9  
tqLVyGhs9UVe8RY=  
=cbe0  
-----END PGP MESSAGE-----
```

Percebeu como mensagem criptografada ficou maior? É que agora ela possui uma codificação para cada chave pública.

## Fingerprint (Impressão Digital)

*Tudo bem, alguém me manda sua chave pública, eu a adiciono no meu chaveiro e começo a utilizá-la para encriptar coisas. Mas quem me garante de que a chave que recebi realmente é daquela pessoa?*

Chegamos a um dos pontos mais delicados da criptografia usando par de chaves: o que garante que a chave pertence àquela pessoa? Isso será melhor discutido na seção [O limite da confiabilidade](#).

Existe uma maneira de confirmar a procedência da chave utilizando a **impressão digital** dessa chave pública, que nada mais é do que um número associado àquela chave. A idéia é que você possa confirmar a impressão digital da chave pública com a pessoa proprietária, de preferência ao vivo ou de alguma forma bem segura. É uma das melhores maneiras de se certificar que a chave realmente pertence à pessoa.

Por exemplo, alguém me envia uma chave pública. Eu a adiciono ao meu chaveiro. Quando tiver a oportunidade de encontrar a pessoa ao vivo, troco com ela nossas impressões digitais, isto é, passo pra ela a impressão digital da minha chave pública - que mantendo anotada num papelzinho dentro da minha carteira! - e ela me passa sua impressão, que também estava anotada num papelzinho! Quando chegar em casa, posso confirmar se a chave pública dela que tenho no meu computador tem a mesma impressão digital que me foi passada durante o encontro. Se as duas coincidirem, posso ter quase certeza de que aquela chave realmente pertence à pessoa em questão. Eu digo quase pois nós que usamos criptografia somos muito céticos e paranóicos.

Aqui segue um exemplo de impressão digital de uma chave pública:

```
268F 448F CCD7 AF34 183E 52D8 9BDE 1A08 9E98 BC16
```

Não é nada complicado manter esse conjunto de letras num papelzinho ou impresso no seu cartão de

visitas!

## Assinaturas digitais

Na seção anterior explicamos como a criptografia pode ser utilizada para criptografar e descriptografar informações. O uso da criptografia não se restringe a isso. Com ela, é possível criar assinaturas digitais.

Quando você fecha um contrato, você assina à caneta na linha pontilhada. Espera-se que você tenha lido tudo direitinho e corcorde com o que estava escrito. Em outras palavras, você dá confiança naquele documento e assina em baixo. Sua assinatura é a prova de que você deu sinal verde. Quanto mais complicada, cheia de garranchos e rabiscada for sua assinatura, melhor para você, pois dificilmente alguém conseguirá falsificar sua assinatura.

Com a criptografia baseada em chaveiro a coisa é mais ou menos assim. Bem mais ou menos. É parecida no sentido de que você precisa confiar no que está assinando. Mas diferente no sentido que ninguém conseguirá falsificá-la (a não ser que roubem sua chave privada e sua senha).

A assinatura funciona da seguinte maneira: eu escrevo um texto, por exemplo:

```
Torta na cara, torta no pé, torta onde quiser!
```

Em seguida, utilizo a seguinte operação:

```
mensagem original -> chave privada -> mensagem assinada
```

A mensagem assinada conterá a informação da mensagem original e virá acompanhada de uma porção de texto gerada pela combinação da mensagem original, da minha chave privada e muitas operações matemáticas malucas. Essa porção de texto é a assinatura digital. Se alguém receber essa mensagem e possuir minha chave pública, poderá testar se essa mensagem tem uma assinatura correta de minha chave privada:

```
mensagem assinada -> chave pública -> confirmação da assinatura
```

Se eu assinar a mensagem anterior, a mensagem assinada será

```
-----BEGIN PGP SIGNED MESSAGE-----
```

```
Hash: SHA1
```

```
Torta na cara, torta no pé, torta onde quiser!
```

```
-----BEGIN PGP SIGNATURE-----
```

```
Version: GnuPG v1.2.3 (GNU/Linux)
```

```
iD8DBQFAuSSqvSy1nGtWZ3cRAvbuAJ9CZgA3YXCWCWiHMgtqA1pnjUqnaQCgvrv  
JRwCqz/lUTdhE0j5KzoMvX0=
```

```
=l6xH  
-----END PGP SIGNATURE-----
```

De modo análogo ao caso da mensagem criptografada, as linhas

```
-----BEGIN PGP SIGNED MESSAGE-----  
Hash: SHA1
```

indicam que o que vem a seguir está assinado digitalmente. Os caracteres malucos são a assinatura em si. Um programa que tenha essa assinatura e a chave pública correspondente à chave privada que a assinou pode verificar se a assinatura confere.

Por fim, a linha

```
-----END PGP SIGNATURE-----
```

avisa o programa de criptografia que encerrou a porção de texto que contém a assinatura.

Eu também poderia enviar minha mensagem num arquivo e a assinatura separadamente, para facilitar a leitura da mensagem. Por exemplo, a assinatura correspondente à mensagem

```
Mensagem protegida contra estelionato.
```

Pode ser distribuída num arquivo em separado, cujo conteúdo, se assinado com minha chave pública, será:

```
-----BEGIN PGP SIGNATURE-----  
Version: GnuPG v1.2.3 (GNU/Linux)  
  
iD8DBQBALBiEvSy1nGtWZ3cRAmneAKCzNFcwvmIYUS4/t9x9jeAWTCLcygCfSbpT  
UrUnpqD+GUfovjNiS56dCec=  
=8Czp  
-----END PGP SIGNATURE-----
```

A assinatura digital permite até que você assine sua cópia da chave pública dos seus amigos e amigas. Uma vez que você trocou sua impressão digital com a deles, assinar a sua cópia de uma chave pública de outra pessoa indica que você confia naquela chave pública.

## O limite da confiabilidade

Como foi dito na seção [Fingerprint \(Impressão Digital\)](#), não há nada que garanta a origem das chaves públicas que recebemos e adicionamos ao nosso chaveiro.

O que podemos fazer é, além de recebermos ao vivo as impressões digitais das chaves públicas, dar níveis de confiança a determinadas chaves. Dar um nível de confiança às vezes é a única coisa que

podemos fazer quando não tivermos a oportunidade de trocas as impressões digitais.

Algo muito popular é o uso da rede de confiabilidade, que é mais ou menos baseada no princípio do **amigo de amigo meu é meu amigo**. Por exemplo, Maria tem a chave pública de João e ambos já trocaram suas impressões digitais e assinaram as chaves públicas um do outro. Maria ainda tem a chave pública de Josefina mas nunca poderão trocar suas impressões digitais ao vivo, pois Josefina mora na Cracóvia e lá é muito longe. João, porém, já foi até a Cracóvia, teve a oportunidade de trocar sua impressão digital com Josefina e assinou a chave pública dela. Com isso, Maria poderá ter confiança na chave de Josefina, pois como Maria confia na chave de João e João confia na chave de Josefina, Maria pode também confiar na chave de Josefina, que poderá confiar na chave de Maria.

Criando essa **teia de confiabilidade**, as pessoas podem ter um pouco mais de segurança no uso da criptografia. O importante é confiar não só que as chaves públicas que você possui realmente pertencem aos seus amigos, mas também confiar nos seus amigos, pois eles assinarão chaves que eventualmente você poderá confiar.

## Repositórios de chaves

Existem alguns computadores na internet que atuam como servidores públicos de chaves públicas: eles armazenam chaves públicas, podendo receber e enviar chaves públicas, o que torna mais fácil a busca por chaves, principalmente de pessoas com as quais você ainda não teve contato. De uma maneira semelhante ao uso dos sites de busca, você vai até o servidor de chaves e procura pelo nome da pessoa, seu email, etc, e a partir disso pode baixar e adicionar chaves ao seu chaveiro.

## Histórico

Essa seção é meramente ilustrativa e tem como objetivo dar um tempero à discussão. Sua leitura pode ser dispensada sem que as próximas seções fiquem comprometidas.

O cerne do problema, quando falamos em criptografia, é que o meio onde a mensagem vai se propagar é público, ou seja, qualquer pessoa presente naquele meio pode capturar a mensagem, seja esse meio uma sala, uma mesa de bar, o sistema telefônico ou a internet. Desde o começo, essa arte não teve muito sucesso em criar meios privados para a circulação de mensagens, o que ela conseguiu foi criar sistemas que impediam que a mensagem fosse compreendida caso fosse interceptada durante seu trajeto até o receptor.

O esquema de par de chaves que tratamos aqui, também conhecido como chaves assimétricas, é talvez a descoberta [mais importante no campo da criptografia nos últimos séculos](#), mas foi precedida por outras também muito interessantes e será sucedida por técnicas ainda mais refinadas - como no caso da criptografia quântica.

Aqui seguem algumas referências muito interessantes sobre a aventura de cifrar mensagens através dos tempos:

- [Criptografia na Wikipedia](#)
- [Wikibook sobre criptografia](#)

## 2. Instalando programas de criptografia

## Programas de criptografia

Neste tutorial ensinamos apenas como utilizar o programa [GNU Privacy Guard](#), que é uma ferramenta de criptografia inteiramente baseada em software livre. Existe outra alternativa ao GPG, que é o [Pretty Good Privacy](#), mais popular que o GPG, mas com dois inconvenientes:

**Liberdade e segurança.** A licença do GPG permite que você o estude, use, modifique-o e distribua. A do PGP não. Além disso o PGP usa um algoritmo (método de criptografia) patenteado nos U\$A.

É por isso que no nosso tutorial encorajamos o uso do GPG. Ele é o programa base para a criptografia. Se você for utilizar criptografia no seu programa de email, por exemplo, estará indiretamente utilizando o GPG. Aqui trataremos o uso do GPG no modo texto e no modo gráfico. [Leia nota sobre modo texto e modo gráfico.](#)

## Instalando o GPG

Existem duas opções para usar o GPG: modo texto, onde você digita os comandos manualmente, e o modo gráfico, mais intuitivo e recomendado para quem ainda não tem familiaridade com criptografia.

Todos os programas para o Modo Gráfico são na verdade **extensões** do GPG em modo texto, o que significa que todos os programas de criptografia aqui listados sempre vão utilizar o mesmo chaveiro, ou seja, tanto seu(s) gerenciador(es) de chaveiro como seu(s) programa(s) de email utilizarão sempre as mesmas informações de chave pública e privada.

Nas seções seguintes, damos instruções de como instalar o GPG no modo texto, tanto no Windows quanto no Linux ou MacOSX. Se preferir, você pode pular a parte Modo Texto e ir direto para a seção [Modo Gráfico](#).

### Modo Texto

#### GNU/Linux

Se você usa GNU/Linux ou outro \*NIX (BSD like, etc), provavelmente sua distribuição deve ter um pacote do GPG. A instalação do mesmo depende de qual distribuição de linux você usa. Aqui ensinaremos como instalar nas distribuições mais populares.

Se você usa o [Indymix](#), todos os programas de criptografia que você necessita já estão instalados e você pode pular para a próxima seção!

Se o seu Linux é o [Debian](#) ou compatível (como Kurumin, Knoppix ou Gnoppix), entre na internet, abra um terminal e digite o comando para instalar o GPG:

```
su -c "apt-get update ; apt-get install gnupg"
```

Em seguida, digite a senha de administrador do sistema, caso esta seja pedida. Depois de instalar o GPG, digite no terminal:

```
su -c "chmod 4755 $(which gpg)"
```

Em seguida, digite a senha de administrador do sistema, caso esta seja pedida.

Esse último comando fará com que o GPG tenha permissão para proteger a leitura da parte memória onde ele carregará suas informações secretas e evita a exibição de algumas mensagens de erro.

Se você usa outra distribuição de Linux, procure na documentação específica do seu sistema como fazer isso, ou [baixe](#) os fontes do programa e compile. Se você usa MacOS, tente o [MacGPG](#).

## Windows

Se você usa Windows, meus pêsames.

1. Faça o download da versão do GPG para Windows em: <http://www.gnupg.org/download.html>
2. Descompacte o arquivo que você baixou.
3. Copie os arquivos gpg.exe e gpgv.exe para c:\windows
4. No prompt do DOS digite

```
gpg
```

e deve aparecer algo assim:

```
gpg: c:/gnupg: directory created
gpg: c:/lib/gnupg/options.skl: can't open: No such file or directory
gpg: you have to start GnuPG again, so it can read the new options file
```

5. Digite "gpg" de novo e você verá:

```
gpg: c:/gnupg/secring.gpg: keyring created
gpg: c:/gnupg/pubring.gpg: keyring created
gpg: Go ahead and type your message ...
```

6. Não digite nada! Pressione ctrl-c

## Modo Gráfico

### GNU/Linux

No Linux, nós recomendamos que você use o [GPA: GNU Privacy Assistant](#). Sua instalação depende de qual distribuição de linux você usa. Aqui ensinaremos como instalar nas distribuições mais populares. Se você usa [Indymix](#), todos os programas de criptografia que você necessita já vem instalados e você pode pular para a próxima seção!

Se você usa [Debian](#) ou compatível (como Kurumin, Knoppix ou Gnoppix), basta se conectar à internet, abrir um terminal e dar o comando para sua instalação:

```
su -c "apt-get update ; apt-get install gpa"
```

Em seguida, digite a senha de administrador do sistema, caso esta seja pedida.

Para outras distribuições, consulte a documentação correspondente. Ou, se você preferir, [baixe](#) o código fonte do GPA e compile-o você mesmo.

## Windows

No Windows, você pode usar o Windows Privacy Tools, que é um dos programas mais práticos para criptografia. Nele, você pode codificar tanto arquivos quanto textos copiados com ctrl-c ou ctrl-x.

1. Baixe o WinPT em <http://winpt.sourceforge.net/>
2. Execute esse programa que você baixou. Ele irá instalar e adicionar o WinPT na sua barra de tarefas.
3. Quando você executar essa instalação, a primeira janela que irá aparecer pede para que você selecione qual sua língua preferida para a instalação. Faça isso e clique em "Ok"
4. Em seguida aparecerá uma janela com uma mensagem de boas vindas. Apenas clique em "Next" (ou o equivalente na língua que você escolheu).
5. Aparecerá então a opção para você escolher em qual pasta do seu computador o WinPT será instalado. A pasta padrão é

C:\Arquivos de programas\Windows Privacy Tools

Se você quiser escolher outro local, não há problema: basta selecionar a pasta clicando em "Browse". Depois que você fizer isso, clique em "Next".

6. Agora é tempo de escolher quais pedaços do WinPT você quer instalar. Esses pedaços, ou componentes, são:

- O GPG propriamente dito
- O WinPT
- "Plugins" pra você usar criptografia no seu programa de email
- Acessórios diversos

Você pode simplesmente clicar em "Next" que o WinPT e a maioria desses componentes serão automaticamente instalados.

Se você usa o Eudora para ler email, selecione "WinPT Eudora Plugin" dentro da seção E-Mail Plugins que se encontra na lista de componentes e clique em "Next".

**7.** A próxima etapa que aparecerá na janela de instalação é escolher o nome da pasta que irá aparecer no menu Iniciar do Windows. Digite o que você quiser e clique em "Next".

**8.** Agora você poderá escolher se existirão atalhos para o WinPT no seu desktop, no menu Iniciar, etc, e a língua na qual o programa será exibido. Faça isso e clique em "Next".

**9.** Na janela de instalação aparecerão opções avançadas de instalação. Se você achou tudo isso muito complicado, não se preocupe: apenas clique em "Install".

**10.** Por alguns instantes você verá uma barra de progresso que indicará a quantas anda a instalação. Em seguida, aparecerá a mensagem final da instalação do WinPT. Apenas clique em "Finish".

**11.** A instalação já foi concluída e agora é tempo de configurar o WinPT. No canto da sua barra do Windows aparecerá um ícone em forma de chave - eles estarão provavelmente ao lado do relógio. Esse ícone é o menu principal do WinPT. Aparecerá também uma janela intitulada "Windows Privacy Tray", dizendo que você não tem um par de chaves (pública/privada) criptográficas. Se é a primeira vez que você instala um software desse tipo em seu computador, e portanto você não possui chaves, selecione "Have WinPT to generate a key pair" (Use o WinPT para gerar um par de chaves) e clique em "Ok".

**12.** Aparecerá uma nova janela, intitulada Key Generation. Nos campos correspondentes, digite seu nome, comentário (isso é opcional, e pode ser qualquer coisa, desde uma frase que você goste até algo sobre você, sei lá), seu endereço de email, a data de validade desse par de chaves (escolha uma data clicando no botão "... ou deixe em branco caso você queira que esse par de chaves não tenha data de validade), sua senha e a confirmação da senha (confirmar a senha é só digitá-la novamente). Depois de tudo isso, clique em "Start".

**13.** Aparecerá uma janela maluca na qual são mostrados alguns caracteres. Não se preocupe com isso. É o PGP criando seu par de chaves. Depois disso, aparecerá uma janelinha dizendo que a geração do par foi completada. Clique em "Ok".

**14.** Se aparecer uma janela dizendo que é recomendável você fazer um backup das suas chaves, apenas clique em "Yes" e escolha uma pasta para gravar isso. **Não grave essas informações em pastas que você compartilha com outras pessoas, sejam elas pessoas que usam o mesmo computador que você ou sejam pastas compartilhadas através de programas como o Kazaa.**

**15.** Pronto! Você já pode criptografar suas mensagens. A próxima seção lhe ensinará como usar o WinPT no seu dia-a-dia.

### 3. Usando os programas de criptografia

Esta seção trata do uso diário da criptografia, tanto em modo texto quanto em modo gráfico. Se você ainda não o fez, [leia nota sobre modo texto e modo gráfico](#). Ações como criar e gerenciar chaves, assinar, verificar assinaturas, criptografar e descriptografar serão tratadas adiante.

#### Modo Texto

As instruções de como utilizar o GPG no Modo Texto funcionam tanto se você usa GNU/Linux quanto Windows, MacOSX ou qualquer outro sistema operacional.

##### Como criar seu par de chaves

Abra um terminal. Em seguida:

1. Digite "gpg --gen-key"

2. Nas três primeiras perguntas apenas aperte enter. É seguro usar as opções padrão do GPG:

```
gpg (GnuPG) 1.2.3; Copyright (C) 2003 Free Software Foundation, Inc.  
This program comes with ABSOLUTELY NO WARRANTY.  
This is free software, and you are welcome to redistribute it  
under certain conditions. See the file COPYING for details.
```

Por favor selecione o tipo de chave desejado:

- (1) DSA e ElGamal (padrão)
- (2) DSA (apenas assinatura)
- (5) RSA (apenas assinatura)

Sua opção? 1

O par de chaves DSA terá 1024 bits.

Prestes a gerar novo par de chaves ELG-E.

tamanho mínimo é 768 bits

tamanho padrão é 1024 bits

tamanho máximo sugerido é 2048 bits

Que tamanho de chave você quer? (1024) 1024

O tamanho de chave pedido é 1024 bits

3. Na terceira pergunta você deve escolher por quanto tempo suas chaves serão válidas. Você pode escolher chaves que expiram em poucos dias para usá-las de teste ou chaves que expiram em anos ou que não expiram. Para uma chave que expira em **n** dias, digite **nd** (por exemplo, **10d** para dez dias), para **n** semanas digite **nw** (por exemplo, **2w** para duas semanas), para **n** meses, use **nm** (por exemplo, **5m** para cinco meses), para **n** anos use **ny** (por exemplo, **1y** para hum ano). Se você quer uma chave que não expira, digite **0** (zero). Veja o exemplo:

```
Por favor especifique por quanto tempo a chave deve ser válida.
```

0 = chave não expira

<n> = chave expira em n dias

<n>w = chave expira em n semanas

<n>m = chave expira em n meses

<n>y = chave expira em n anos

A chave é valida por? (0) 6m

Key expira em Dom 15 Ago 2004 22:44:56 BRT

Depois dessa data a chave não mais será válida. É possível, no entanto, alterar a data de validade da chave depois que ela foi criada.

4. Será perguntado se está tudo correto. Se você seguiu tudo direito até aqui pode digitar "y" ou "s" caso você esteja usando uma versão traduzida para o português e apertar ENTER.

Está correto (s/n)? s

**5.** Digite seu nome.

Você precisa de um identificador de usuário para identificar sua chave; o programa constrói o identificador a partir do Nome Completo, Comentário e Endereço Eletrônico desta forma:

"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Nome completo: Truta

**6.** Digite seu e-mail.

Endereço de correio eletrônico: truta@uzma.net

**7.** Não digite um comentário a não ser que você queira ter mais de uma chave como por exemplo uma para usar no trabalho, outra para usar em casa.

Comentário:

**8.** Verifique se tudo que você digitou está certo. Se estiver digite "o" e aperte enter.

Você selecionou este identificador de usuário:

"Truta <truta@uzma.net>"

Muda (N)ome, (C)omentário, (E)ndereço ou (O)k/(S)air? o

**9.** Digite sua frase-senha \***IMPORTANTE**\* sem essa frase-senha você não poderá descriptografar o que lhe enviarem! Por isso use uma FRASE que você não esqueceria, porém que seja **muito difícil** de alguém adivinhar. Use letra minúsculas e maiúsculas, elas serão diferenciadas, pontuação (,.!?) e espaços. Você não verá o que digita.

Você precisa de uma frase secreta para proteger sua chave.

Digite a frase secreta:

**10.** Repita sua frase-senha para confirmar que você não cometeu nenhum erro ao digitá-la. De novo você não verá o que digita.

Repita a frase secreta:

**11.** Se você digitou as duas frases-senha iguais suas chave será gerada. Digite no teclado e move o

mouse aleatoriamente para ajudar o programa a gerar as chaves.

Precisamos gerar muitos bytes aleatórios. É uma boa idéia realizar outra atividade (digitar no teclado, mover o mouse, usar os discos) durante a geração dos números primos; isso dá ao gerador de números aleatórios uma chance melhor de conseguir entropia suficiente.

```
+++++.+++++.+++++.+++++.+++++.+++++.+++++.+++++.+++++.+++++.+
```

chaves pública e privada criadas e assinadas.  
chave marcada como de confiança absoluta

```
pub 1024D/90716386 2004-02-18 Truta <truta@uzma.net>
    Impressão da chave = 27FF 8594 D529 B428 8E7F 9A81 C127 6A77 9071 6386
sub 1024g/7C866836 2004-02-18[expira: 2004-08-16]
```

**12.** Digite gpg --fingerprint e você verá a sua chave e sua "impressão digital" algo do tipo:

```
pub 1024D/90716386 2004-02-18 Truta <truta@uzma.net>
    Impressão da chave = 27FF 8594 D529 B428 8E7F 9A81 C127 6A77 9071 6386
sub 1024g/7C866836 2004-02-18[expira: 2004-08-16]
```

A sua "impressão digital" é um número hexadecimal (na base 16) de 40 dígitos, sendo que os 8 últimos são a identificação da sua chave, o que diferencia a sua chave das outras.

## Como compartilhar sua chave pública

**1.** Para que o GPG seja de alguma utilidade você terá que mandar sua chave pública para aqueles que vão te enviar mensagens criptografadas. Para isso extraia sua chave pública digitando:

```
gpg --export --armor -o chave.asc seu@email
```

Sua chave pública estará no arquivo chave.asc no diretório atual. Dê uma olhada nela. Você também pode exportar chaves públicas de outras pessoas, desde que elas estejam no seu chaveiro. Basta usar o email delas ao invés do [seu@email](mailto:seu@email).

**2.** Existem basicamente dois métodos para que você transmita sua chave pública:

\* você enviar sua chave para uma pessoa (por email, disquete, cd, etc.).

\* você enviar sua chave pública para um servidor e cada pessoa que quiser usá-la baixa a chave do servidor.

Enviar sua chave para um servidor é o meio mais cômodo de compartilhar sua chave com as pessoas que vão enviar mensagens criptografadas para você. Dessa forma você não precisa ficar mandando suas chave pública pra todo mundo.

Para mandar sua chave para um servidor de chaves, use o comando

```
gpg --keyserver servidor.de.chaves --send-keys nome
```

Se você não souber sob quais nomes suas chaves públicas estão registradas, liste-as com o comando

```
gpg --list-keys
```

Por exemplo, se o servidor for keys.indymedia.org e o nome da sua chave for truta, o comando para exportá-la será

```
gpg --keyserver keys.indymedia.org --send-keys truta
```

É importante observar que você pode exportar **qualquer** chave pública do seu chaveiro, que não precisa ser necessariamente sua.

## Como adicionar uma chave pública de alguém na sua lista

Vamos supor que o arquivo que contém a chave pública de alguém é chama-se truta.asc. Para incluir esta chave na sua lista, basta dar o comando:

```
gpg --import truta.asc
```

Para importar uma chave pública de um servidor, você precisa saber qual é o servidor e qual é o ID da chave. Com isso em mãos, basta digitar

```
gpg --keyserver servidor.de.chaves --recv-keys id-da-chave
```

que a chave pública será importada. Por exemplo, se o servidor que você estiver usando for o keys.indymedia.org, e o ID da chave for B9A88F6F?, seu comando será

```
gpg --keyserver keys.indymedia.org --recv-keys B9A88F6F
```

Se você não tiver o ID da chave que você quer adicionar, primeiro faça uma busca no servidor de chaves. Por exemplo, se eu quiser saber qual é o ID da chave do Pietro, basta que eu dê o comando

```
gpg --keyserver keys.indymedia.org --search-keys pietro@indymedia.org
```

A saída provável será

```
gpg: a procurar por "pietro@indymedia.org" no servidor HKP keys.indymedia.org
Keys 1-1 of 1 for "pietro@indymedia.org"
(1)    Pietro Ferrari <pietro@indymedia.org>
      1024 bit DSA key D75301BF, created 2002-02-23
```

```
Enter number(s), N)ext, or Q)uit >
```

Assim você obtém o ID da chave do Pietro, que é [D75301BF?](#). Aí é só digitar Q para sair da busca e em seguida adicionar a chave, usando o comando

```
gpg --keyserver keys.indymedia.org --recv-keys D75301BF
```

Você poderia, ao invés de procurar pelo email do Pietro, procurar apenas pelo nome dele, usando o comando

```
gpg --keyserver keys.indymedia.org --search-keys pietro
```

É importante ressaltar que você só encontrará a chave desejada desde que a pessoa que você procura deixou a chave naquele servidor.

Depois de adicionar uma chave pública de terceiros, efetue os procedimentos na seção [Verificando Impressões Digitais e Assinando Chaves](#).

## Listando seu chaveiro

Você pode ver todas as chaves do seu chaveiro - incluindo seu par de chaves pública e privada - digitando

```
gpg --list-keys
```

A saída é algo do tipo

```
/users/alice/.gnupg/pubring.gpg
-----
pub 1024D/BB7576AC 1999-06-04 Alice (Judge) <alice@cyb.org>
sub 1024g/78E9A8FA 1999-06-04
```

## Como assinar mensagens e arquivos

Existem muitas maneiras de assinar mensagens ou arquivos. A primeira delas consiste em entrar no GPG para escrever sua mensagem. No seu terminal, digite:

```
gpg --clearsign
```

E entre com sua senha. A opção **clearsign** pede ao GPG para que ele crie uma assinatura utilizando texto comum, isto é, codificado em caracteres ASCII (legíveis ao usuário). Não sabe o que é ASCII ou texto comum? Então veja uma nota [aqui](#).

Depois de entrar com sua senha, o GPG estará esperando para que você escreva sua mensagem.

Escreva sua mensagem - "Testando essa parada!", por exemplo - e após escrevê-la, pule uma linha e digite simultaneamente as teclas **Ctrl** e **D** do seu teclado. Isso fará com que o GPG crie uma assinatura da sua mensagem. O resultado deve ser algo do tipo

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Testando essa parada!
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.2.3 (GNU/Linux)

iD8DBQFAMQPvSy1nGtWZ3cRALd5AJ4sI0ed+/kFmFwARMngu+EF73DMCQCgz59l
9okdJxyGs65FL9SycVmVMNg=
=WC+w
-----END PGP SIGNATURE-----
```

*Certo. Assinei minha mensagem. E agora, o que faço com isso?* Bom, se você quiser mandá-la pra alguém, copie e cole todo esse texto, desde o -----**BEGIN PGP SIGNED MESSAGE**----- até o -----**END PGP SIGNATURE**----- e cole no corpo da sua mensagem de email. Se o destinatário tiver sua chave pública, ele poderá facilmente verificar se a assinatura confere. Mas caso você queira guardar essa mensagem assinada, basta copiar tudo e colar num arquivo.

**ATENÇÃO:** se porventura você alterar essa mensagem, sua respectiva assinatura perderá seu valor. Se você quiser alterar a mensagem, faça e depois assine novamente.

Vejamos agora uma segunda maneira de assinar mensagens. Escreva seu texto num arquivo, por exemplo no **texto.txt**. Em seguida, digite no seu terminal:

```
gpg --clearsign texto.txt
```

Depois de entrar com sua senha, o GPG escreverá a mensagem assinada no arquivo **texto.txt.asc**. Com esse procedimento é possível assinar qualquer tipo de arquivo, e o formato da mensagem assinada será em texto simples (ASCII). Para criar uma assinatura separada da mensagem - a mensagem no arquivo **texto.txt** e apenas a assinatura no arquivo **arquivo.txt.asc** - é só digitar:

```
gpg -a --detach-sig texto.txt
```

As mensagens e assinaturas armazenadas em texto simples ocupam mais espaço do que se estivessem no formato "binário". Se você quiser guardar a assinatura num formato binário, que é pouco amigável para ser visualizada num editor de textos mas que ocupa pouco espaço - basta digitar

```
gpg --sign texto.txt
```

E a mensagem assinada estará no arquivo **texto.txt.gpg**. Além de assinar, a opção **sign** compacta a mensagem, ocupando menos espaço ainda. Essa forma de assinar não é boa para trocar mensagens

com outras pessoas, já que não está num formato legível. Prefira sempre a opção **clearsign**

## Como verificar mensagens assinadas

Uma vez que você tenha recebido uma mensagem ou arquivo assinado, você terá de verificar se a assinatura está correta. Existem várias maneiras de fazer isso.

A maneira mais simples é a que se segue: você recebeu uma mensagem como essa:

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Alow!
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.2.3 (GNU/Linux)

iD8DBQFAMU1TvSy1nGtWZ3cRAtz2AJ41a1dGqGwb0wT+kz4WoFq9/4+RoQCfZH29
0gPrLalgr50rC4gC9Lahb0w=
=d2qf
-----END PGP SIGNATURE-----
```

Para verificar essa assinatura, copie todo esse texto, desde o **-----BEGIN PGP SIGNED MESSAGE-----** até o **-----END PGP SIGNATURE-----**, digite, no seu terminal

```
gpg
```

depois, cole todo o texto e em seguida digite simultaneamente as teclas **Ctrl** e **D** do seu teclado. Se tudo der certo, o GPG detectará que trata-se de uma mensagem assinada e verificará sua validade. No caso da mensagem acima, termos:

```
gpg: Assinatura feita em Seg 16 Fev 2004 20:08:03 BRT usando DSA, ID da chave 6B566777
gpg: Assinatura correta de "Silvio Rhatto <rhattoEMriseup.net>"
```

Se você recebeu uma mensagem assinada num arquivo, por exemplo o **mensagem.txt.asc**, basta digitar

```
gpg --verify mensagem.txt.asc
```

Que a assinatura será verificada. Se você recebeu uma mensagem com a assinatura num arquivo separado - **mensagem.txt** e **mensagem.txt.asc**, por exemplo - digite

```
gpg --verify mensagem.txt.asc mensagem.txt
```

## Como codificar uma mensagem para alguém

Assim como para assinar mensagens, existem várias maneiras de se criptografar uma mensagem. A primeira delas consiste em digital sua mensagem no próprio gpg. Vamos lá. No terminal, digite:

```
gpg -e -a -r email@da.pessoa
```

E em seguida escreva sua mensagem. Quando terminal, pule uma linha e em seguida digite simultaneamente as teclas **Ctrl** e **D** do seu teclado. O GPG então fornecerá a mensagem codificada para o usuário cujo email é [email@da.pessoa](mailto:email@da.pessoa). Agora é só copiar todo o texto, desde **-----BEGIN PGP MESSAGE-----** até **-----END PGP MESSAGE-----** e colar no seu email ou arquivo!

Para uma encriptar e compactar um arquivo, digite:

```
gpg -r nome-do-usuário -e mensagem.txt
```

O arquivo de saída será **mensagem.txt.gpg**

Se você quiser apenas encriptar e que o arquivo de saída possa ser enviado por email - texto comum, isto é, codificado em caracteres ASCII (legíveis ao usuário), digite

```
gpg -r nome-do-usuário -e -a mensagem.txt
```

E o arquivo de saída será **mensagem.txt.asc**

Não sabe o que é ASCII ou texto comum? Então veja uma nota [aqui](#).

## Como decodificar uma mensagem que enviaram para você

A maneira mais simples de desencriptar uma mensagem é colá-la diretamente no GPG. Suponha que você tenha recebido a mensagem

```
-----BEGIN PGP MESSAGE-----  
Version: GnuPG v1.2.3 (GNU/Linux)  
  
hQE0A3v8xQeh8DSxEAP8DGLGac90dZzX7KxRqpkaYuJ/8NVN5Ayht0KZwRogZwwZ  
19g/teTPQLYwgcCLQoDKxsnsX3lBGEzjAGxXme6aqcJetD0sXcULtap99AfpJqI0/  
LCDjxFqNRP45UwnnbafakiVDTj71H6jDi6UMP4c+F/JJL65Q9R0HW08k0B1IM1sD  
/RsQu1pqZl/X8PRZyVdLSwpzGpR5uaxA837f+Zl0+1Fh2DhoiE2AxnLXdwlMlRtK  
SrpqBdWifULoX46E1+D0d128e24K74d+utMux4uk8t9Lb0D5C8RDfShKHoLJIwul  
fhEmtN8bo2Kmg/z8sWnDjW3Ik3opVtsTPNPQQh1J9GV401UBViB6IzhkXhhNZ4ae  
qexbLil0VwJczLa3y3UmwkiXcs92k6thpUCIJjyeRTtpey2LKdVLHLd1o5ti8/or  
nqYoq1eXHcV0ckmfh3Uq8ZAEX6bzSKc  
=g5JE  
-----END PGP MESSAGE-----
```

Tudo que você precisa fazer é digitar

```
gpg
```

no seu terminal e em seguida colar a mensagem. O GPG detectará que trata-se de uma mensagem privada e pedirá pela sua senha, mais ou menos assim:

```
gpg: Vá em frente e digite sua mensagem ...
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.2.3 (GNU/Linux)

hQE0A3v8xQeh8DSxEAP8DGLGac90dZzX7KxRqpkaYuJ/8NVN5AyhtQKZwRogZwwZ
19g/teTPQLYwgcCLQoDKxsnX3lBGEzjAGxXme6aqcJetD0sXcULtap99AfpJqIO/
LCDjxFqNRP45UwnnbafakiVDTj71H6jDi6UMP4c+F/JJL65Q9R0HW08k0B1IM1sD
/RsQu1pqZl/X8PRZyVdLSwpzGpR5uaxA837f+Zl0+1Fh2DhoiE2AxnLXdwlMLRtK
SrpqBdWifULoX46E1+D0d128e24K74d+utMux4uk8t9Lb0D5C8RDfShKHoLJlwul
fhEmtN8bo2Kmg/z8sWnDjW3IK3opVtsTPNPQQh1J9GV40LUBViB6IzhkXhhNZ4ae
qexbLil0VwJczLa3y3UmwkiXcs92k6thpUCIJjyeRTtpey2LKdVLHLdlo5ti8/or
nqYoq1eXHcV0ckmfxH3Uq8ZAEX6bzSKc
=g5JE
-----END PGP MESSAGE-----
Você precisa de uma frase secreta para desbloquear a chave secreta do
usuário: "Silvio Rhatto <rhatto@riseup.net>"
chave de 1024-bit/ELG-E, ID A1F034B1, criada em 2003-06-20 (ID principal da chave 6B566
```

Digite a frase secreta:

Digite sua senha e em seguida pressione simultaneamente as teclas **Ctrl** e **D** do seu teclado que a mensagem secreta irá aparecer!

Se alguém lhe enviou uma **mensagem.txt.asc**, basta dar este comando para descriptá-la e guardar o texto em **mensagem.txt**:

```
gpg -d mensagem.txt.asc > mensagem.txt
```

Esse comando funciona tanto para arquivos criptografados em texto simples quanto em formato "binário".

## Verificando Impressões Digitais e Assinando Chaves

Conforme você viu na seção [O limite da confiabilidade](#), é possível verificar pela impressão digital da chave pública se ela pertence realmente a quem afirma pertencer.

Suponha que alguém envia uma chave pública e você a [adiciona em seu chaveiro](#). Posteriormente você tem a oportunidade de encontrar ao vivo o suposto dono da chave e ele lhe fornece a impressão digital da chave num papel.

Chegando em casa, você decide verificar se a impressão digital bate com a chave pública. No seu

terminal, digite

```
gpg --fingerprint email@da.pessoa
```

onde **email@da.pessoa** é o email da pessoa que você encontrou. A impressão digital da chave será impressa. Confira se ela é idêntica àquela que você tem anotada. Se elas forem iguais, você pode começar a pensar em assinar essa chave pública. Aqui mostrarei o procedimento de assinar uma chave pública e em seguida reproduzirei um trecho do [Guia Foca Linux](#), que por sua vez foi retirado da lista debian-user-portuguese EM lists.debian.org, que trata de modo muito sério a assinatura de chaves.

Digite no seu terminal:

```
gpg --edit-key email@da.pessoa
```

Aparecerão informações sobre essa chave. Digite

```
sign
```

E digite sua senha. Pronto, a chave estará assinada.

*Certo, assinei a chave pública daquela pessoa. E agora, o que faço com isso? Você pode [exportar a chave pública](#) dessa pessoa - que será automaticamente exportada com sua assinatura. Você tanto pode exportá-la num arquivo e enviá-la para essa pessoa quanto mandar essa chave assinada para um servidor de chaves.*

Suponha que Arrelia assinou a chave de Pasqualin, [exportou-a](#) no arquivo **pasqualin.asc** e enviou-a para Pasqualin. Quando Pasqualin [adicionar](#) a chave contida em **pasqualin.asc** no seu chaveiro, a assinatura feita por Arrelia da chave pública de Pasqualin será automaticamente adicionada ao chaveiro de Pasqualin. Agora, sempre que Pasqualin enviar sua chave pública para alguém, a assinatura de Arrelia sempre estará presente.

Um outro método para o intercâmbio de assinaturas utiliza os servidores de chaves. Por exemplo, se a chave pública de Pasqualin estiver armazenada no servidor chaves.privacidade.net, basta que Arrelia assine a chave pública de Pasqualin e [exporte-a](#) para esse servidor para que o servidor adicione a assinatura de Arrelia na sua cópia da chave pública de Pasqualin. Em seguida, basta que Pasqualin [atualize](#) seu chaveiro, de forma que sua própria chave pública seja baixada do servidor chaves.privacidade.net para que a assinatura de Arrelia entre no seu chaveiro.

Aqui segue o texto retirado do [Guia Foca Linux](#):

Trocando assinaturas de chaves digitais

Direitos de republicação cedidos ao domínio público, contanto que o texto seja reproduzido em sua íntegra, sem modificações de quaisquer espécie, e incluindo o título e nome do autor.

1. Assinaturas digitais
2. Chaves digitais e a teia de confiança
3. Trocando assinaturas de chaves digitais com um grupo de pessoas

### 1. Assinaturas digitais

Uma assinatura digital é um número de tamanho razoável (costuma ter de 128 a 160 bits) que representa um bloco bem maior de informação, como uma e-mail.

Pense numa assinatura como se ela fosse uma versão super-comprimida de um texto. Se você muda alguma coisa (por menor que seja) no texto que uma assinatura "assina", essa assinatura se torna inválida: ela não mais representa aquele texto.

Existe uma relação direta entre uma assinatura e informação que ela assina. Se uma das duas for modificada, elas passam a não mais "combinar" uma com a outra. Um programa de computador pode detectar isso, e avisar que a assinatura é "inválida".

Os algoritmos mais usados para criar e verificar assinaturas digitais são o SHA-1, RIPEM160 e MD5. O MD5 não é considerado tão bom quanto os outros dois.

Assinaturas digitais também funcionam com arquivos "binários", ou seja: imagens, som, planilhas de cálculo... e chaves digitais.

### 2. Chaves digitais e a teia de confiança

Chaves digitais são fáceis de falsificar, você só precisa criar uma chave nova no nome de sicrano, por um endereço de e-mail novinho em folha daqueles que você consegue nesses webmail da vida, e pronto. Agora é só espalhar essa chave por aí que os bestas vão usá-la pensando que é de sicrano.

A menos que os "bestas" não sejam tão bestas assim, tenham lido o manual do seu software de criptografia, e saibam usar assinaturas e a teia de confiança para verificar se a tal chave é de sicrano mesmo.

Programas de criptografia (os bons, tipo PGP e GNUpg) usam um sistema de assinaturas nas chaves digitais para detectar e impedir esse tipo de problema: Ao usuário é dado o poder de "assinar" uma chave digital, dizendo "sim, eu tenho certeza que essa chave é de fulano, e que o e-mail de fulano é esse que está na chave".

Note bem as palavras "certeza", e "e-mail". Ao assinar uma chave digital, você está empenhando sua palavra de honra que o nome do dono de verdade daquela chave é o nome que está na chave, e que o endereço de e-mail daquela chave é da pessoa (o "nome") que também está na chave.

Se todo mundo fizer isso direitinho (ou seja, não sair assinando a chave de qualquer um, só porque a outra pessoa pediu por e-mail, ou numa sala de chat), cria-se a chamada teia de confiança.

Numa teia de confiança, você confia na palavra de honra dos outros para tentar verificar se uma chave digital é legítima, ou se é uma "pega-bobo".

Suponha que Marcelo tenha assinado a chave de Cláudia, e que Roberto, que conhece Marcelo pessoalmente e assinou a chave de Marcelo, queira falar com Cláudia.

Roberto sabe que Marcelo leu o manual do programa de criptografia, e que ele não é irresponsável. Assim, ele pode confiar na palavra de honra de Marcelo que aquela chave digital da Cláudia é da Cláudia mesmo, e usar a chave pra combinar um encontro com Cláudia.

Por outro lado, Roberto não conhece Cláudia (ainda), e não sabe que tipo de pessoa ela é. Assim, rapaz prevenido, ele não confia que Cláudia seja uma pessoa responsável que verifica direitinho antes de assinar chaves.

Note que Roberto só confiou na assinatura de Marcelo porque, como ele já tinha assinado a chave de Marcelo, ele sabe que foi Marcelo mesmo quem assinou a chave de Cláudia.

Enrolado? Sim, é um pouco complicado, mas desenhe num papel as flechinhas de quem confia em quem, que você entende rapidinho como funciona.

O uso da assinatura feita por alguém cuja chave você assinou, para validar a chave digital de um terceiro, é um exemplo de uma pequena teia de confiança.

### 3. Trocando assinaturas de chaves digitais com um grupo de pessoas

Lembre-se: ao assinar uma chave digital, você está empenhando sua palavra de honra que toda a informação que você assinou naquela chave é verdadeira até onde você pode verificar, e que você tentou verificar direitinho.

Pense nisso como um juramento: "Eu juro, em nome da minha reputação profissional e pessoal, que o nome e endereços de e-mail nessa chave são

realmente verdadeiros até onde posso verificar, e que fiz uma tentativa real e razoável de verificar essa informação."

Sim, é sério desse jeito mesmo. Você pode ficar muito "queimado" em certos círculos se você assinar uma chave falsa, pensando que é verdadeira: a sua assinatura mal-verificada pode vir a prejudicar outros que confiaram em você.

Bom, já que o assunto é sério, como juntar um grupo de pessoas numa sala, e trocar assinaturas de chaves entre si? Particularmente se são pessoas que você nunca viu antes? Siga o protocolo abaixo, passo a passo, e sem pular ou violar nenhum dos passos.

1 - Reúna todos em uma sala, ou outro local não tumultuado, pressa e bagunça são inimigas da segurança.

2 - Cada um dos presentes deve, então, ir de um em um e:

2.1 - Apresentar-se, mostrando calmamente documentação original (nada de fotocópia) comprovando sua identidade. RG, CPF, passaporte, certidão de nascimento ou casamento, carteira de motorista, cartão de crédito são todos bons exemplos. Só o RG sozinho não é -- tem muito RG falsificado por aí -- mas o RG junto com o cartão de banco já seria suficiente. Se nenhum documento tiver foto, também não é o bastante.

\* Se alguém pedir o documento na mão, para verificar direitinho, não leve pro lado pessoal. Deixe a pessoa verificar até estar satisfeita (mas não descuide do documento). Isso só significa que ela leva muito a sério a responsabilidade de assinar chaves.

2.2 - Entregar um papel com as informações da chave: Nome (QUE OBRIGATORIAMENTE PRECISA SER O MESMO NOME CONSTANTE NOS DOCUMENTOS APRESENTADOS EM 2.1), e-mail, número da chave (keyID), e fingerprint da chave (assinatura digital da chave)

RECIPIENTE DO PAPEL: Se você achar que os documentos que te apresentaram não são prova suficiente, talvez porque o nome não bate com o da chave, ou porque uma foto nos documentos não está parecida com quem mostrou os documentos, marque discretamente no papel, porque você NÃO deve assinar essa chave. Se achar que o outro vai engrossar, não diga para ele que não vai assinar a chave dele.

3 - Pronto. Podem ir embora, porque o resto dos passos deve ser feito com calma, em casa. Lembre-se que você não vai estar efetuando nenhum julgamento moral a respeito de quem você assinar a chave. Você só irá afirmar que a chave de sicrano é realmente aquela, e mais nada.

4 - Para cada uma das chaves que você marcou no papel que "posso assinar":

4.1 - Peça para o seu programa de criptografia mostrar a chave e sua assinatura (fingerprint).

SE: O nome no papel for exatamente igual ao nome na chave (user ID/UID da chave). E: A assinatura no papel for exatamente igual à assinatura na chave (fingerprint). ENTÃO:  
Vá para o passo 4.3.

4.2 - As informações não bateram, por isso você não deve assinar a chave. Se quiser, envie um e-mail avisando que não poderá assinar a chave. Não aceite tentativas de retificação por e-mail ou telefone. Um outro encontro face-à-face, refazendo todos os passos 2.1 e 2.2 é o único jeito de retificar o problema.

4.3 - As informações bateram, o que garante que o \*nome\* está correto. Agora é preciso ter certeza do endereço de e-mail. Para isso, envie uma e-mail \*CIFRADA\* pela chave que você está testando, para o endereço de e-mail constante na chave. Nessa e-mail, coloque uma palavra secreta qualquer e peça para o destinatário te responder dizendo qual a palavra secreta que você escreveu. Use uma palavra diferente para cada chave que estiver testando, e anote no papel daquela chave qual palavra você usou.

4.4 - Se você receber a resposta contendo a palavra secreta correta, você pode assinar a chave. Caso contrário, não assine a chave -- o endereço de e-mail pode ser falso.

Comandos do gpg (GNUpg) correspondentes a cada passo:

2.2 -        gpg --fingerprint <seu nome ou 0xSuakeyID>  
(retorna as informações que devem estar no papel a ser entregue no passo 2.2)

4.1 -        gpg --receive-key <0xKEYID>  
(procura a chave especificada nos keyservers)  
              gpg --sign-key <0xKEYID>  
(assina uma chave)

Assume-se que você sabe cifrar e decifrar mensagens. Caso não saiba, ainda não é hora de querer sair assinando chaves.

O trecho acima descreve talvez o método mais rigoroso que alguém poderia ter para a assinatura de chaves públicas. É lógico que você não precisa ser tão rígido para assinar chaves dos seus amigos ou pessoas que você realmente conhece e confia. Para essas, basta trocarem ao vivo as impressões digitais das chaves públicas e confirmarem seus endereços de email para que ambos possam assinar as chaves públicas.

## Recebendo sua chave assinada

Se alguém assinou sua chave, é conveniente você atualizar sua cópia da sua chave pública para que ela contenha essa nova assinatura. Você pode fazer isso de duas maneiras: importando a chave que a pessoa te enviou pela forma usual, ou seja, utilizando o comando [gpg --import](#) ou, caso ela a tenha enviado para um servidor de chaves, atualizando seu chaveiro de acordo com as últimas modificações de chaves do servidor.

Para essa segunda opção, basta digitar:

```
gpg --refresh-keys --keyserver keys.indymedia.org
```

onde **keys.indymedia.org** é o servidor de chaves para o qual a pessoa mandou a chave assinada. Esse comando fará com que todas as chaves públicas do seu chaveiro - inclusive a sua - sejam atualizadas a partir das chaves públicas existentes no servidor de chaves. Assim, se alguém assinou uma chave e a exportou para o servidor de chaves, esse comando atualizará seu chaveiro substituindo a chave pública antiga pela nova.

É muito interessante dar esse comando periodicamente para atualizar seu chaveiro, independentemente de alguém ter assinado uma chave. As atualizações de chaves podem acontecer sem ninguém te avisar.

## Confiando em chaves

Assinar chaves mostra a outras pessoas que você confia na procedência de determinadas chaves públicas. Mas pode acontecer de você assinar a chave de um amigo seu mas não confiar nas chaves que ele assina. Existe uma maneira de lembrar a você em quais colegas seus você confia quando eles assinam chaves de outras pessoas, que é o chamado **nível de confiabilidade** daquela chave.

Essa informação não é passada a outros usuários. Quando exportada, não existirá diferença nenhuma se a chave pública foi definida por você com alto ou baixo nível de confiabilidade, já que o nível de confiabilidade não tem relação nenhuma com a assinatura de mensagens. Em outras palavras, esse nível apenas ajuda o usuário na hora de decidir se confia ou não em determinada chave que ele não assinou mas que foi indiretamente confiada pela assinatura de alguém confiável.

Para alterar o nível de confiabilidade, digite:

```
gpg --edit-key email@da.pessoa
```

Aparecerão informações sobre aquela chave. Em seguida, digite

```
trust
```

Aparecerá um menu de opções:

```
Por favor decida quanto confia neste utilizador para  
verificar correctamente as chaves de outros utilizadores  
(vendo passaportes, verificando impressões digitais...)?
```

```
1 = Não sei  
2 = Eu NÃO confio  
3 = Confio moderadamente  
4 = Confio plenamente  
5 = Confio de forma total  
m = voltar ao menu principal
```

```
Sua decisão?
```

Agora é só escolher o nível de confiabilidade desejada.

Formalmente, uma chave pública é considerada válida apenas se ela for assinada por você. Mas usando o conceito de Teia de Confiabilidade, o GPG é bem flexível em considerar uma chave válida, por exemplo, se:

- Ela foi assinada por você **ou**
- Ela foi assinada por alguém que você confia totalmente **ou**
- Ela foi assinada por três chaves que você confia moderadamente **ou**
- Se existe um caminho entre você e a chave pelo qual todas as chaves estão assinadas. João assinou a chave de Raimundo, que assinou a chave de Maria, cuja chave você assinou; esse caminho permite que o GPG considere válida a chave de João, **sem que você precise assiná-la**. Normalmente o número de pessoas nessa corrente, para que a chave torne-se válida, não pode ser maior que cinco.

**Você não precisa decorar esse esquema!** Uma vez que você seleciona o nível de confiabilidade de uma chave, o GPG automaticamente recalcula a validade de todas as chaves do seu chaveiro, usando um método um pouco mais sofisticado do que o exemplificado acima.

## Removendo chaves

Se você quiser remover a chave pública de alguém, use o comando

```
gpg --delete-key email@da.pessoa
```

onde **email@da.pessoa** é o email da pessoa cuja chave você quer apagar. Agora, se você quiser remover um par de chaves (pública e privada), use o comando:

```
gpg --delete-secret-and-public-key seu@email
```

onde **seu@email** é o seu email. Exemplo:

```
gpg --delete-secret-and-public-key truta@uzma.net
gpg (GnuPG) 1.2.3; Copyright (C) 2003 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

sec 1024D/90716386 2004-02-18 Truta <truta@uzma.net>

Deletar esta chave do chaveiro? s
Esta é uma chave secreta! - realmente deletar? s
pub 1024D/90716386 2004-02-18 Truta <truta@uzma.net>

Deletar esta chave do chaveiro? s
```

**Mas tome cuidado:** uma vez que você apagou seu par de chaves, não há mais como recuperá-las, ler mensagens criptografadas para você ou assinar mensagens. **Lembre-se de revogar sua chave pública antes de cancelá-la (veja como na próxima seção).**

## Cancelando um par de chaves

Se você quiser cancelar um par de chaves, por qualquer motivo - alguém roubou sua chave secreta e sua senha, por exemplo - você usará o comando para revogar sua chave.

O comando

```
gpg --output revoke.asc --gen-revoke ID
```

revogará minha chave cuja identificação é **ID** (que pode ser tanto o nome do par de chaves, como o seu número ou o email correspondente) e gerará um certificado de revogamento no arquivo **revoke.asc**. Esse certificado serve para ser enviado a quem tiver minha chave pública para que saibam que cancelei meu par de chaves. É uma espécie de *assinatura de cancelamento*.

## Outros comandos

Para maiores informações sobre como usar o gpg em modo texto, consulte as [Referências](#) ou então digite no seu terminal

```
man gpg
```

## Modo Gráfico

### Linux: usando o GPA

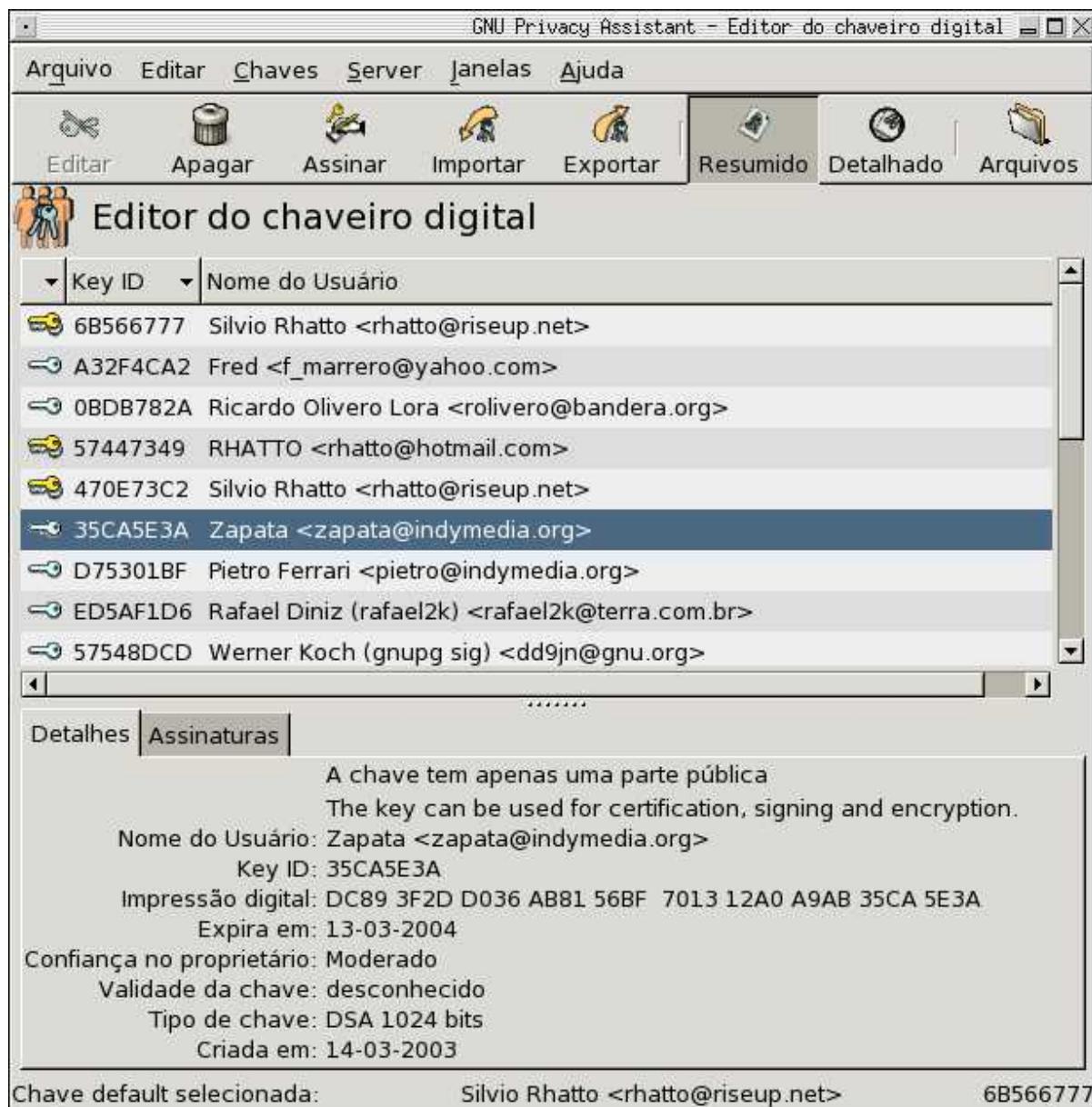
O **GPA**, ou GNU Privacy Assistant (Assistente de Privacidade da [GNU](#)) é basicamente um **gerenciador** de chaveiro. Um gerenciador de chaveiro é a ferramenta com a qual você pode controlar as cópias de chaves públicas e as chaves privadas que você possui, podendo apagar, assinar, mudar confiabilidade e criar novos pares de chaves, dentre outras coisas. O GPA permite ainda que você assine, verifique assinaturas, criptografe e descriptografe arquivos. O GPA não é feito para você utilizar criptografia no seu email, mas ele é útil mesmo assim porque os programas de email que utilizam criptografia não tem capacidade de gerenciar chaves e eles apenas cuidam de assinar, verificar assinaturas, criptografar e descriptografar mensagens de email. Por isso, é recomendável que, além de usar um cliente de email com suporte à criptografia (saiba mais sobre isso na seção [Criptografia e correio eletrônico](#)), você também use o GPA para administrar suas chaves. Uma outra alternativa a GPA é o [SeaHorse](#), que não será abrangido neste manual.

Para instalar o GPA, consulte a seção [Instalando o GPA](#). Se você já instalou ou no seu sistema o GPA já vem instalado, basta que você abra um terminal e digite, **como usuário comum** (isto é, sem estar como administradores do sistema),

```
gpa
```

ou então acesse o GPA a partir do menu do seu sistema.

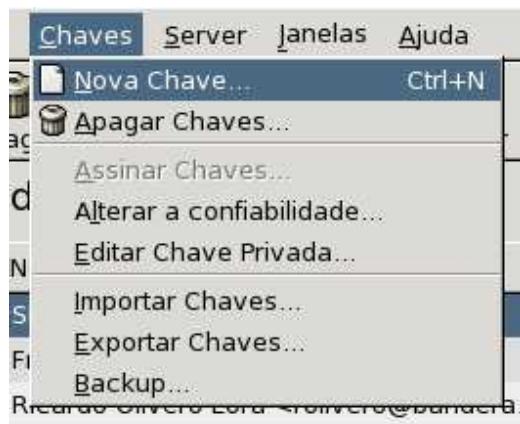
A figura a seguir mostra a página principal do GPA:



Gerenciador do chaveiro digital: a janela principal do GPA.

## Criando um novo par de chaves

Logo que você entrar no GPA pela primeira vez, o programa poderá avisar que você ainda não tem uma chave privada e perguntará se você quer gerá-la agora ou mais tarde. Se você não tem uma chave privada, clique em "Gerar chave agora". Caso ele não faça isso, vá no menu **Chaves** e clique em \*Nova Chave...\*.



Em seguida, digite seu nome e clique em "Avançar". Faça o mesmo para o seu email e opcionalmente para o comentário. Digite sua senha e confirme-a, tomando cuidado para não escolher uma senha muito óbvia. De que adianta uma poderosa ferramenta de criptografia se a sua senha é uma porcaria?

Por favor indique o seu nome completo.

Seu nome fará parte da nova chave para permitir aos outros identificar mais facilmente as chaves.

Seu nome:

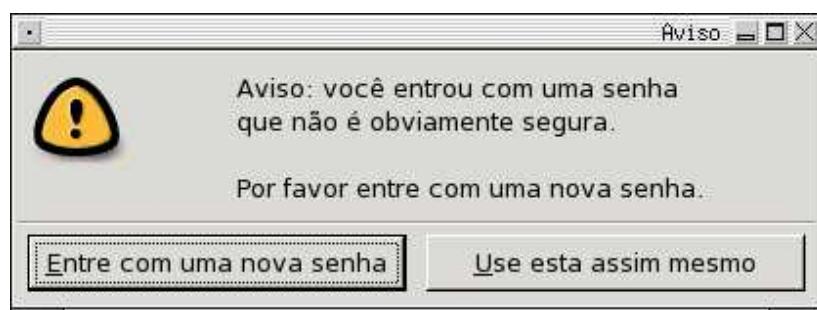
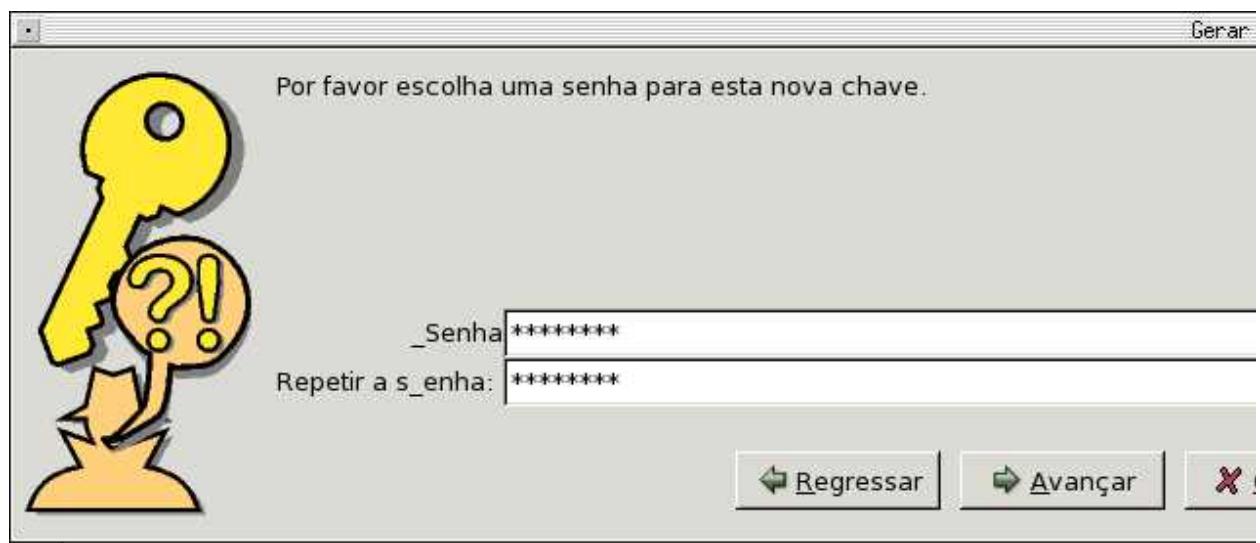
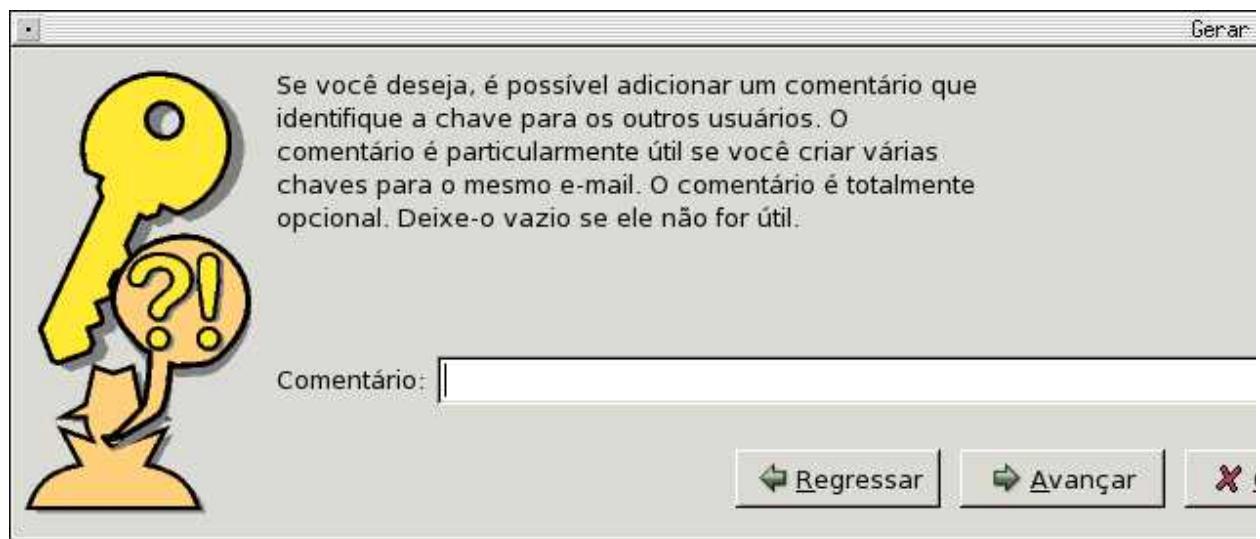


Por favor indique o seu e-mail.

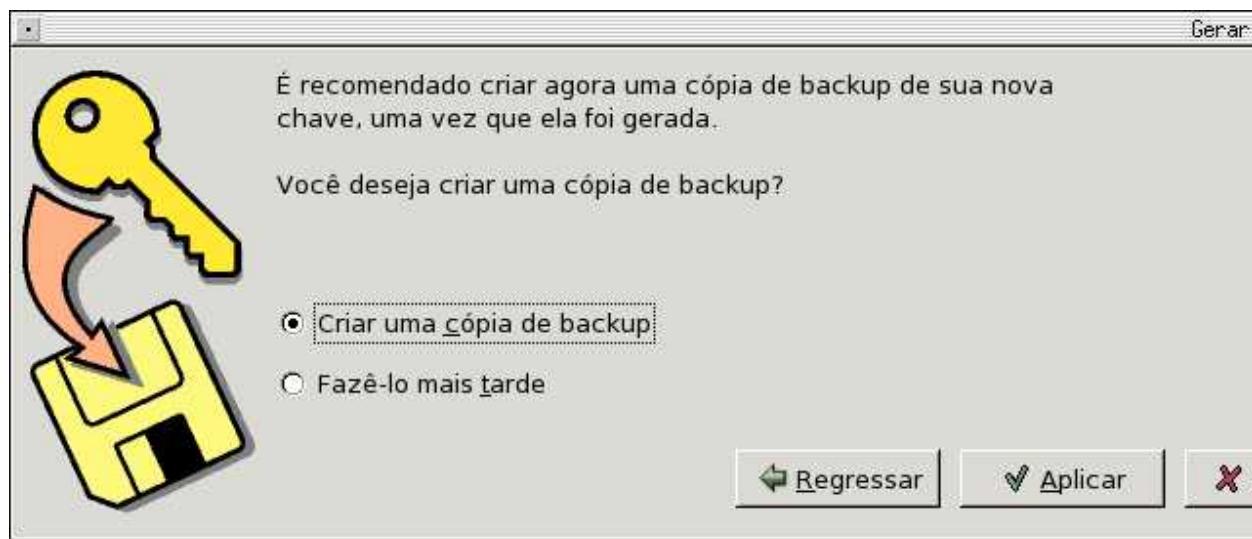
Seu endereço eletrônico ferá parte da nova chave para permitir às outras pessoas a identificação mais fácil das chaves. Se você tem vários e-mails, será possível adicionar outros endereços mais tarde.

Seu e-mail:





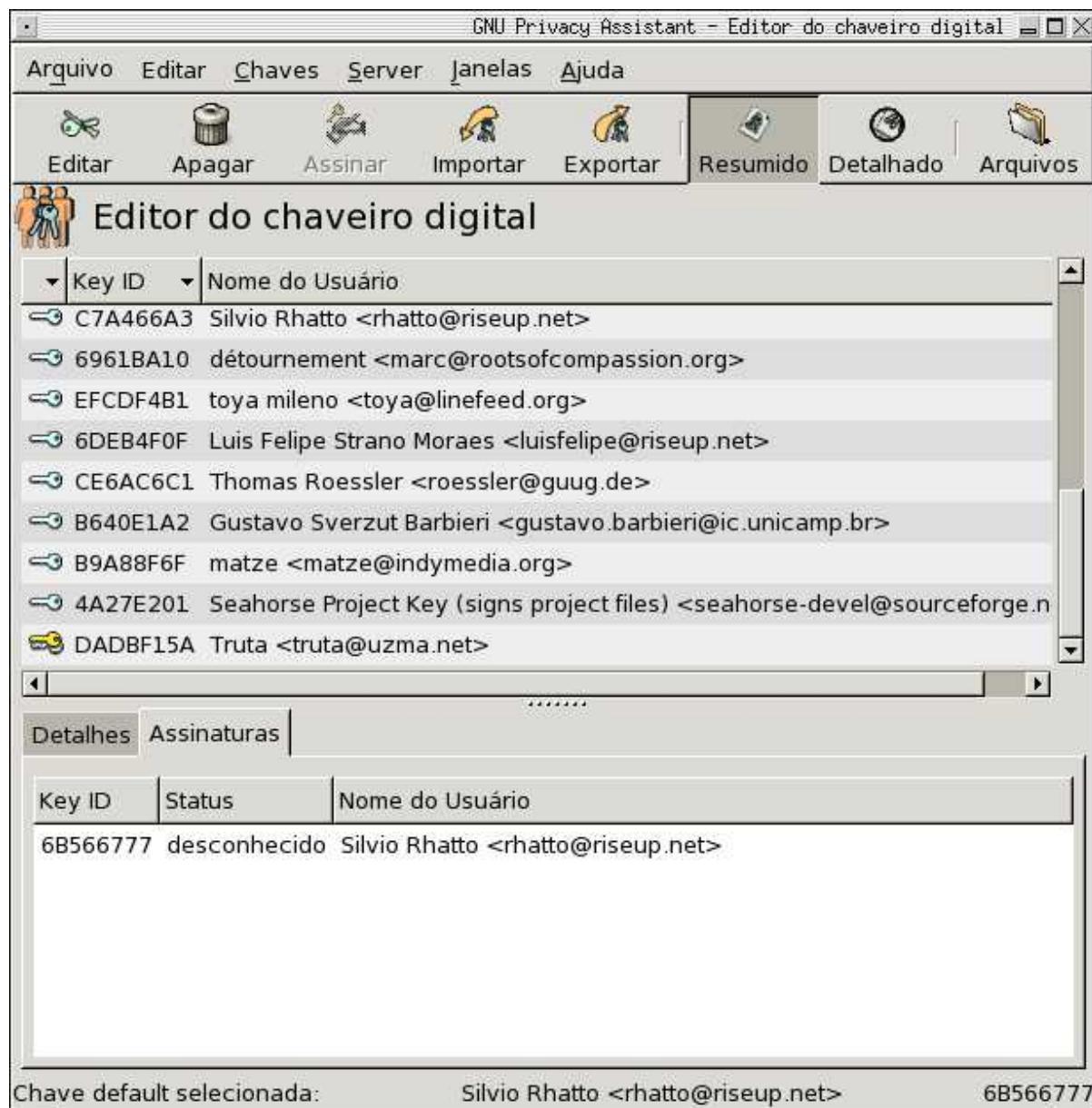
Se quiser, peça para que seja criada uma cópia de backup, algo que só é recomendado caso você queira ter um chaveiro num outro computador ou quando você pretende reinstalar seu sistema. Depois disso, demorará um certo tempo para sua chave ser criada. Agora é só começar a brincadeira.

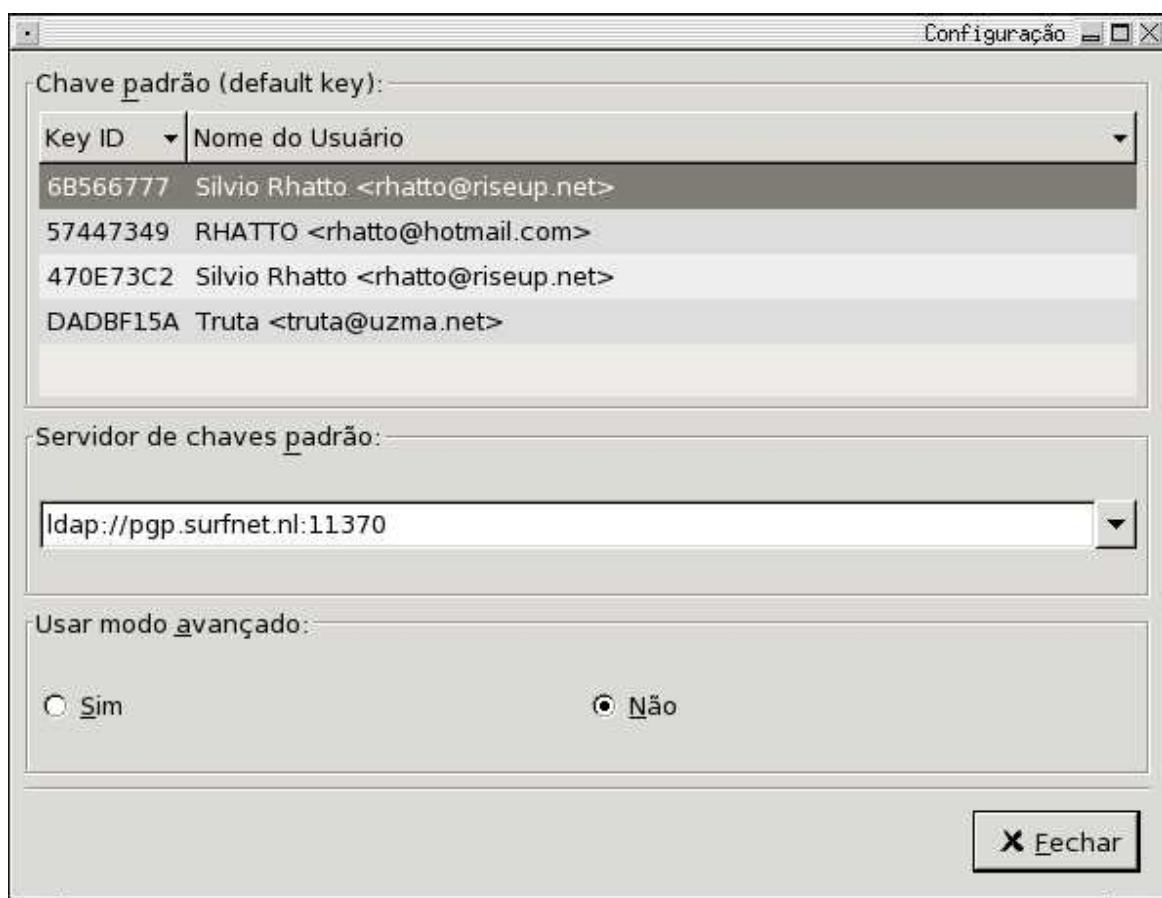
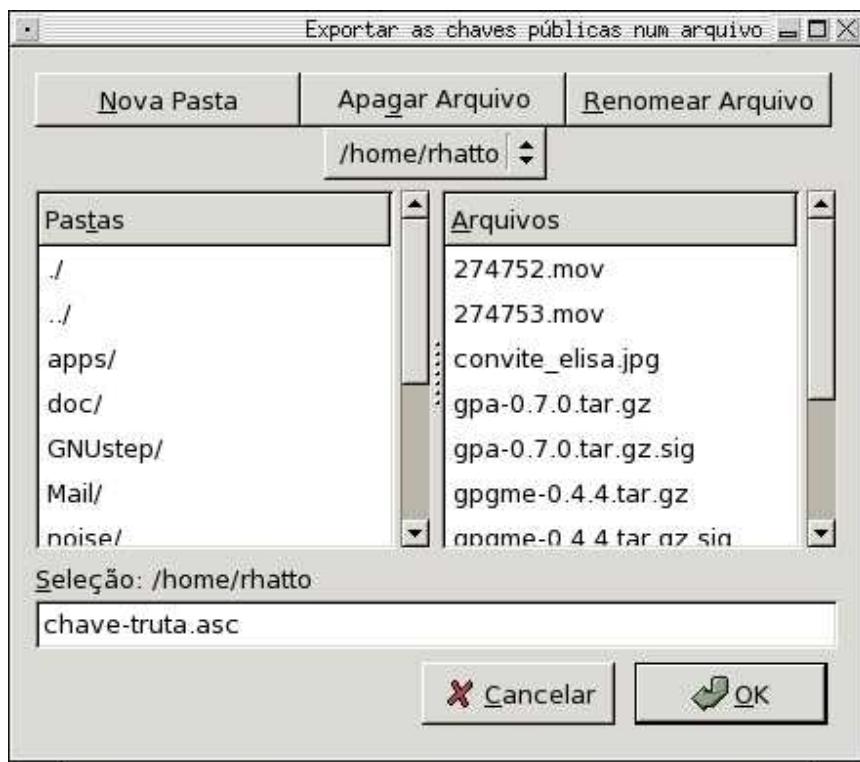


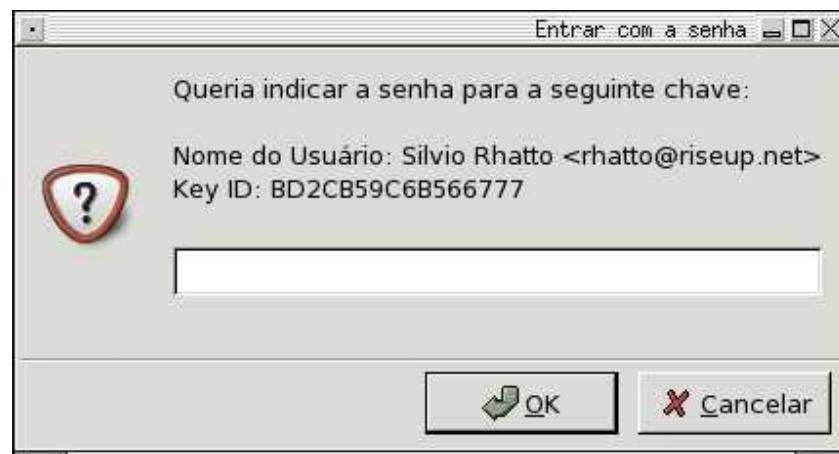
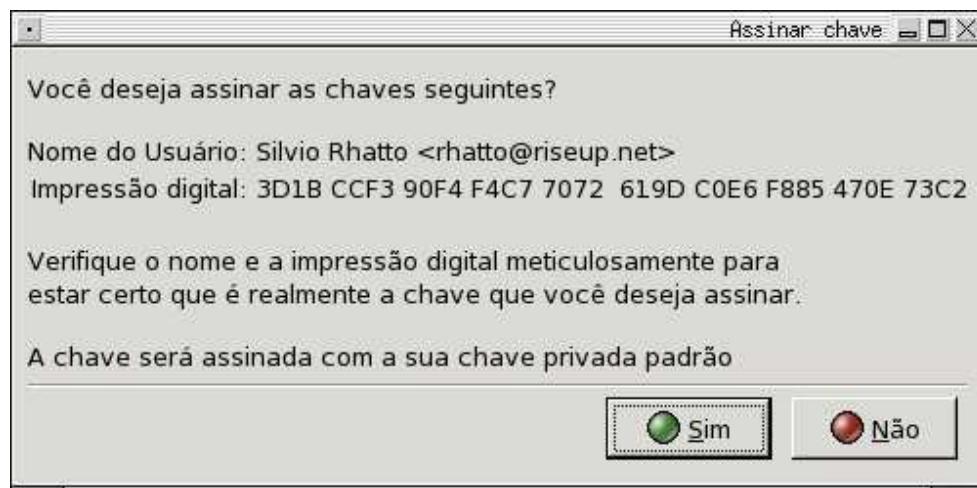
## Gerenciando chaves públicas

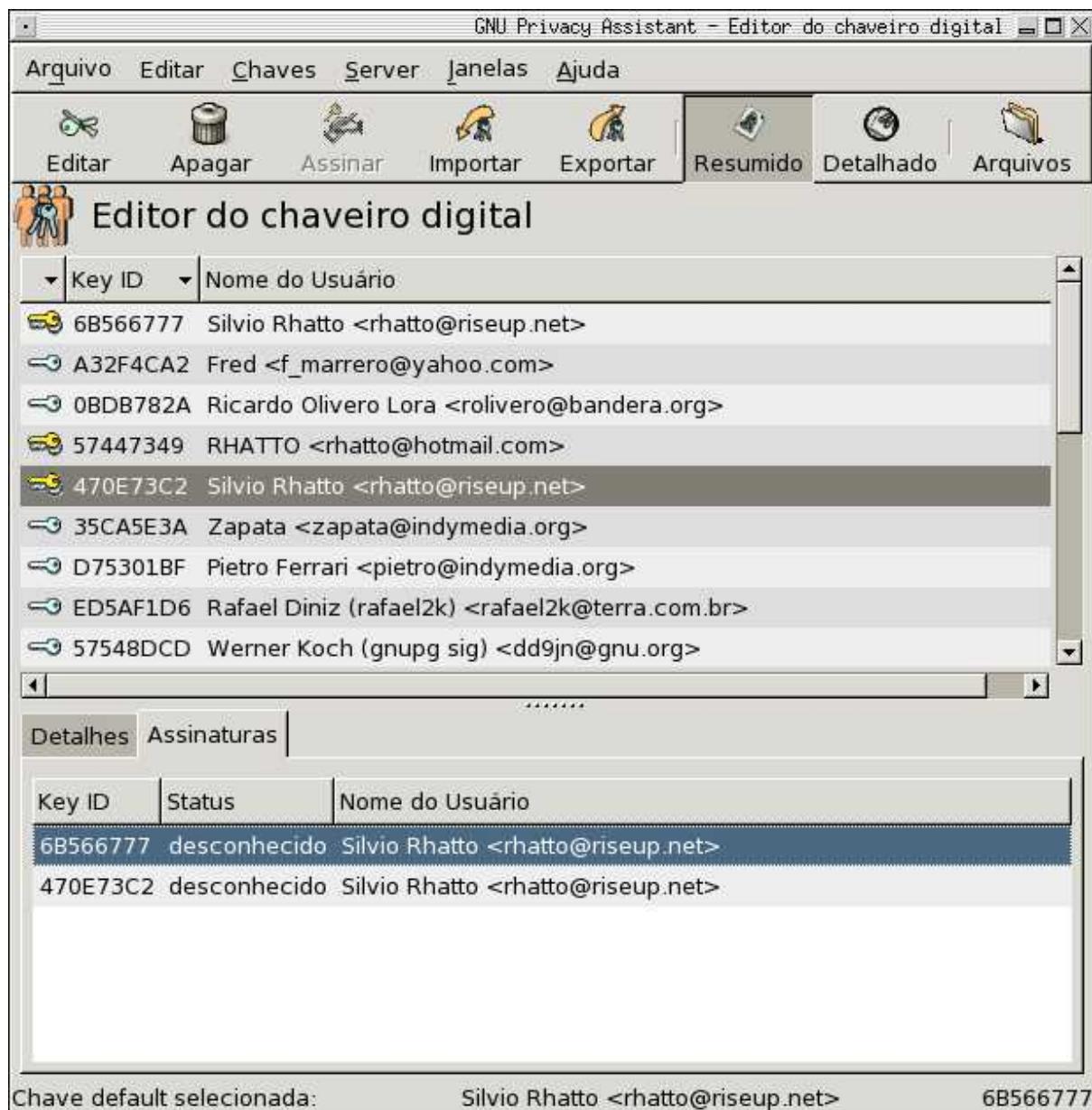
Para adicionar uma chave pública ao seu catálogo, basta clicar no botão "Importar". Aqui você pode tanto usar o ID da chave quanto usar um arquivo de chave pública que você tenha recebido. Para exportar chaves, o procedimento é análogo. Uma vez que você exportou uma chave pública, é só anexar o arquivo correspondente nas suas mensagens de email para distribuir a todos sua chave pública.

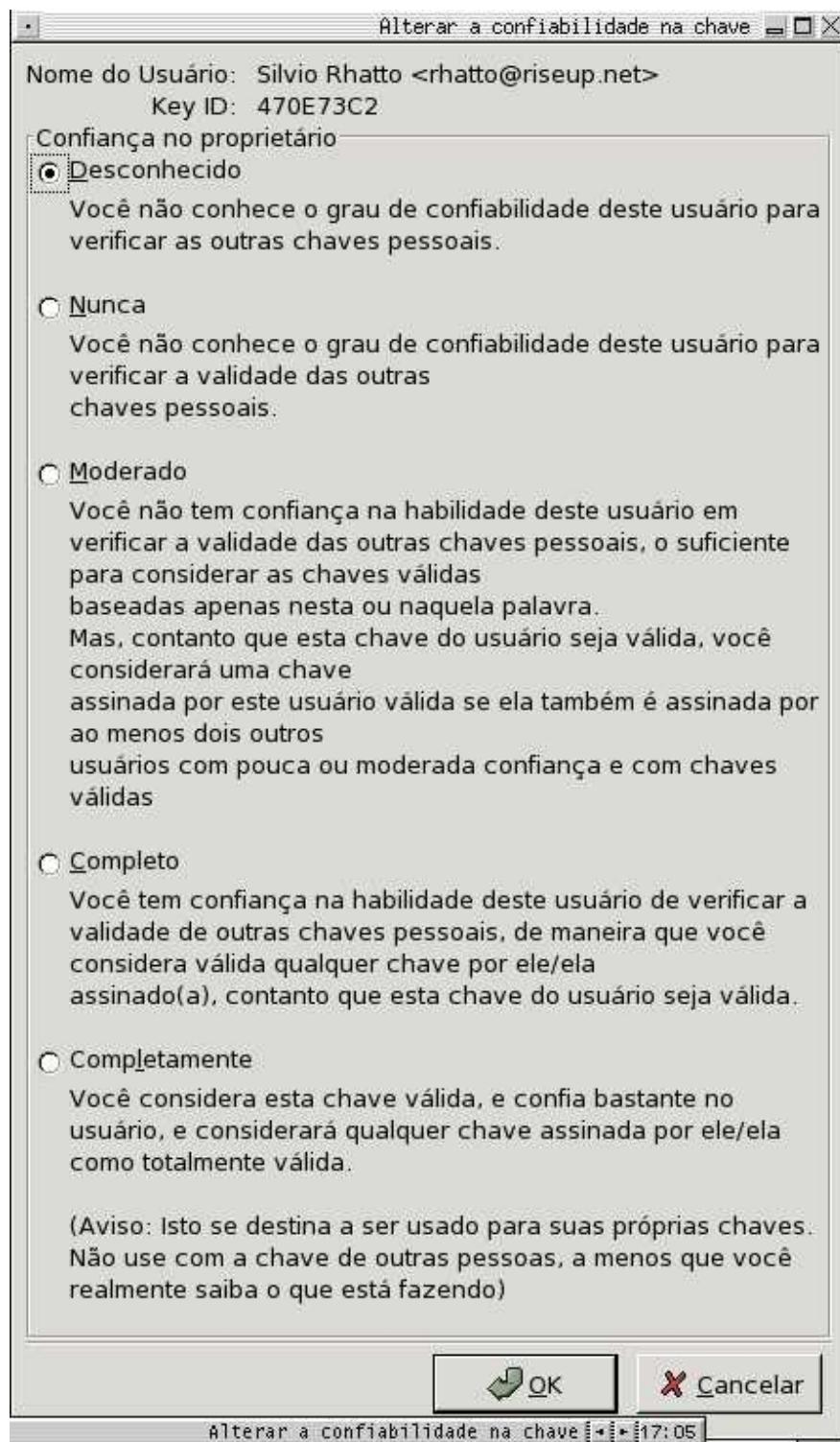


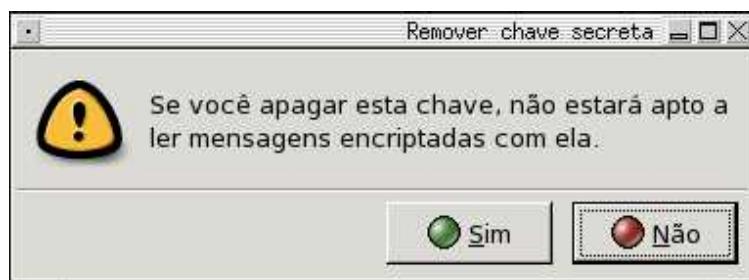
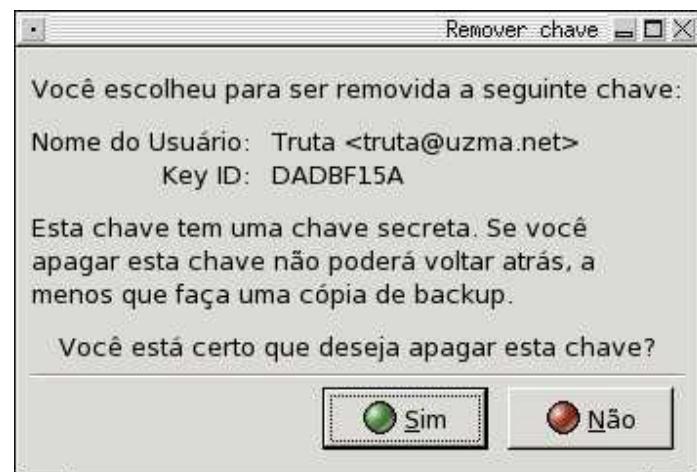






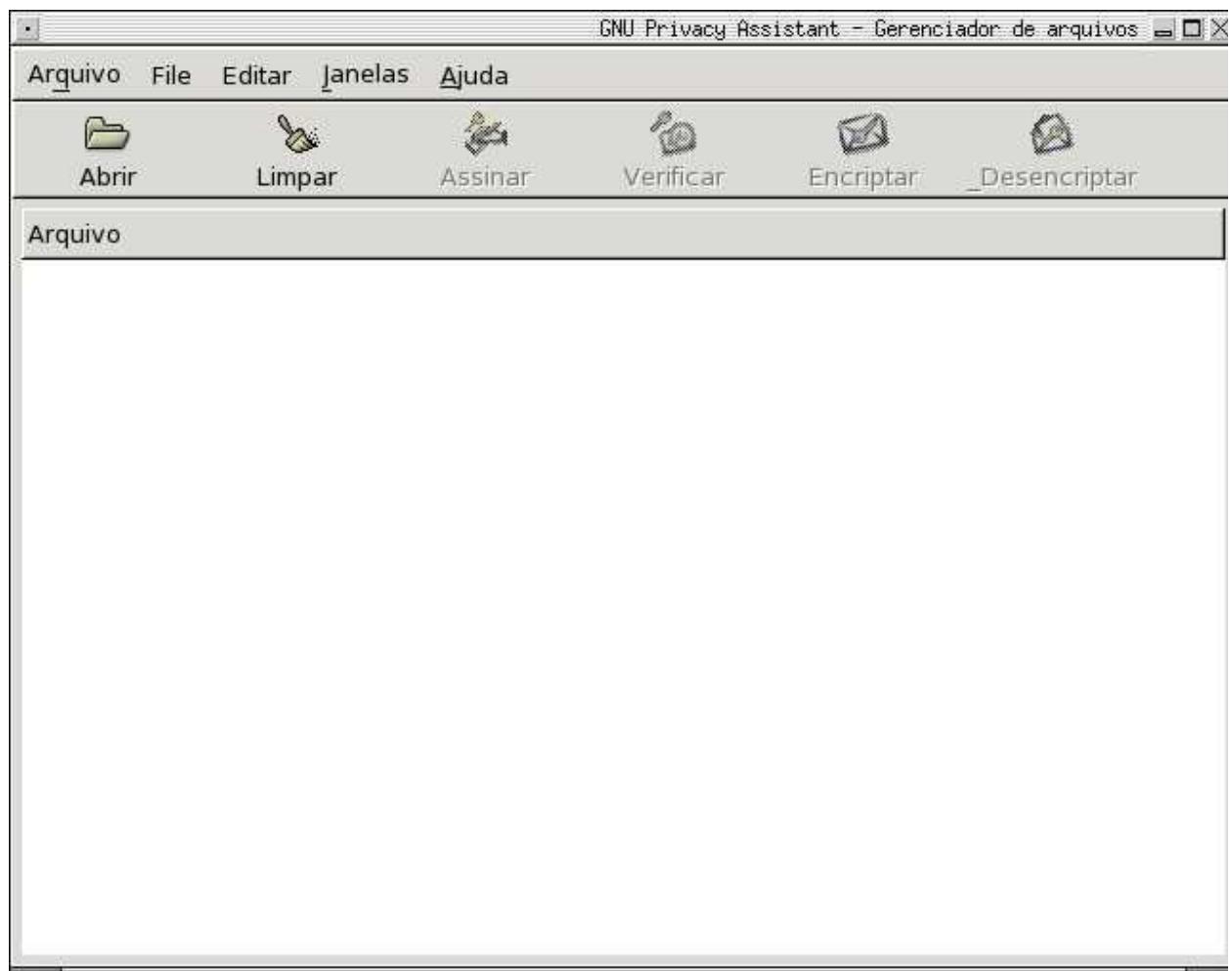




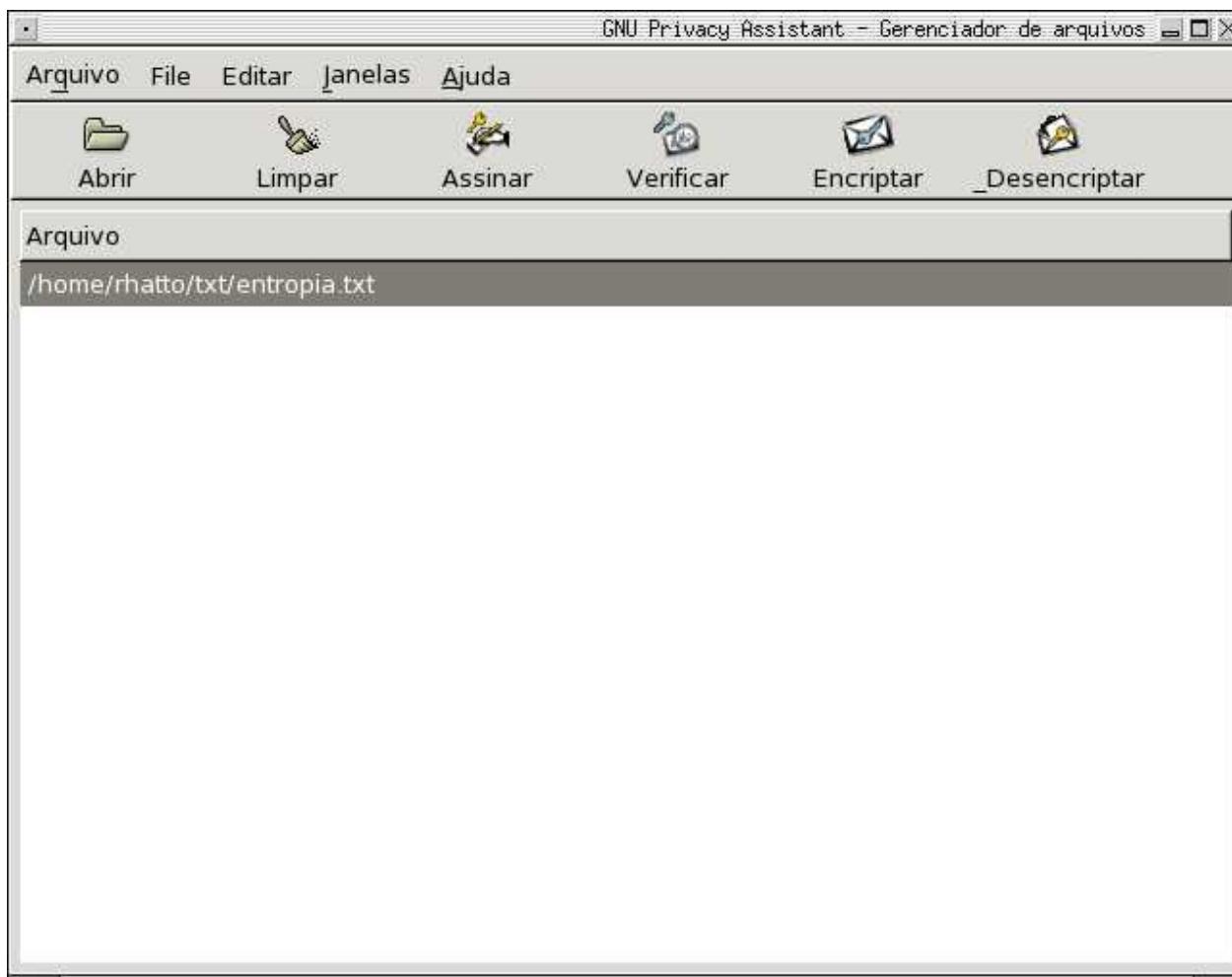


## Encriptar

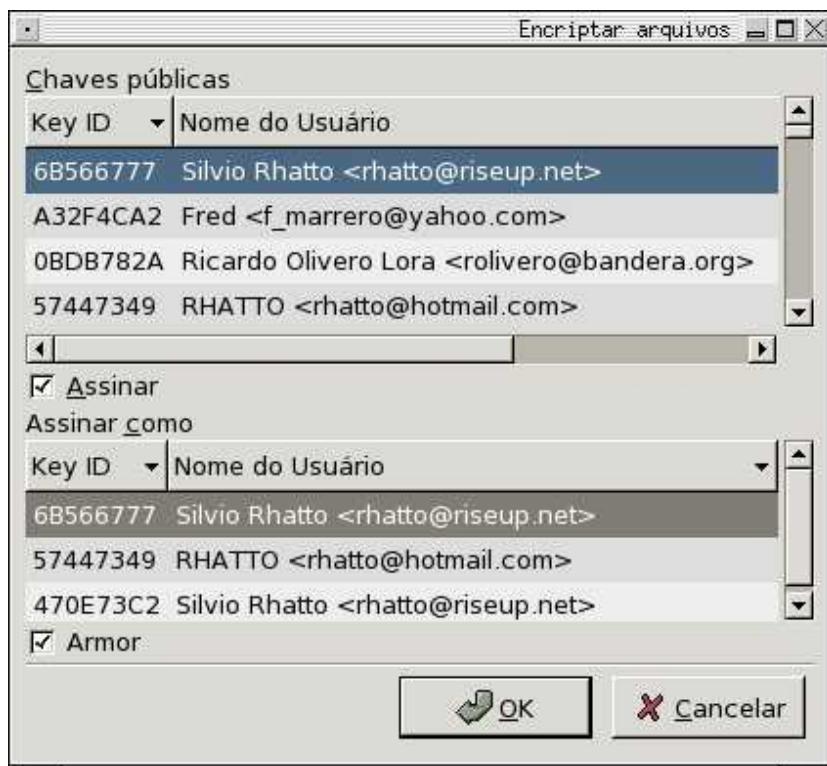
Certo. Agora vamos ao que interessa. Ecriptar e desencriptar. Vamos testar com um arquivo qualquer. Entre num editor de textos e crie um arquivo com o nome entropia.txt. No GPA, clique no botão "Arquivos". Aparecerá o Gerenciador de Arquivos.



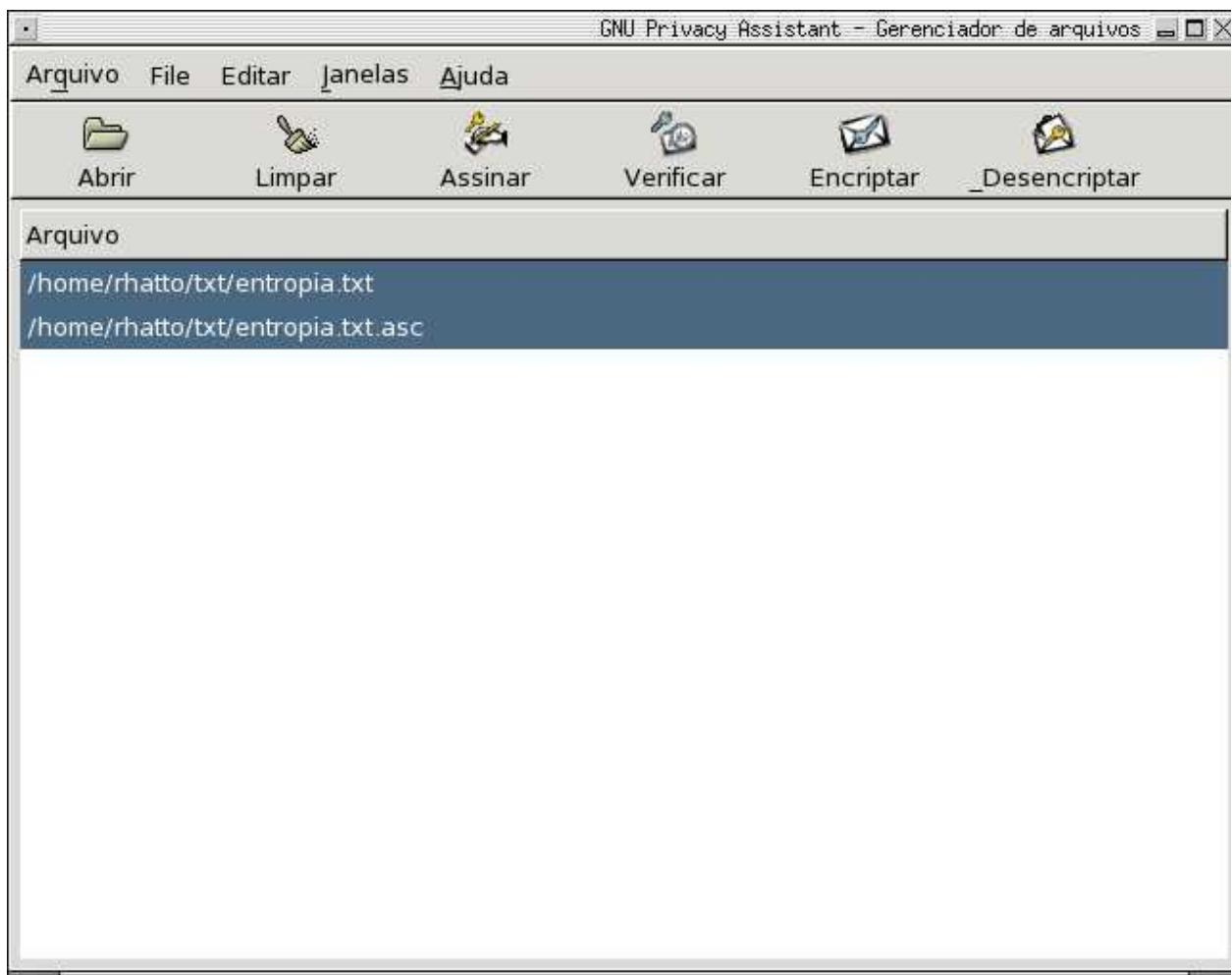
Nessa nova janela, clique no botão "Abrir" e selecione o arquivo entropia.txt.



Clique no botão "Encriptar", selecione seu nome na lista superior e dê um OK. Opcionalmente você pode assinar esse arquivo, e então você terá que digitar sua senha. Se você for mandar mensagens encriptadas por email, não esqueça de selecionar a opção **Armor**, que fará com que a encriptação saia em caracteres legíveis e que podem ser enviados por email.



Pronto, o arquivo com nome entropia.txt.gpg ou entropia.txt.asc, caso você tenha selecionado a opção **Armor**, foi criado e contém a mensagem do arquivo secreto.txt codificada. Experimente abrir o entropia.txt.gpg ou o entropia.txt.asc num editor de textos comum: você verá apenas um monte de caracteres muito estranhos. Sua mensagem agora está protegida e só você pode lê-la. Não esqueça de deletar o arquivo entropia.txt e manter só o entropia.txt.asc ou o entropia.txt.gpg.



A seguir, mostro um exemplo de encriptação onde usei a opção **Armor**:

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.2.1 (GNU/Linux)

hQE0A3v8xQeh8DSxEAP9E4wGxbz+EgftZ4cLMVF7H0Ka7vdWzRGvWzXJKLX6qvs4
QUT/biiKGhCEaoBl4hzs6qyW+B1u2Y35fIWV0gT6ATEce5QdSHFGhjNU7pZVCgKZ
bYc1w9w/dHx/3ZDvSiEX9cF6FEpKIubUH9wofU++PNgtK90p2rr8RTNDruG0ejgd
+wabESliF0rc7LGwAu3pK0CQ+DAJALXdNb001wo5EfYAHdJz61P6Zn8Ea47sggE4
H5hKhRstY9UjrQXFmxbfqxT6L8r2xSxmwkESgxjBb11+MPzK++rgsSFh7m7rV0pV
KKMiVSzqq7479TjDWDOp9BaSwimstUU0rNlJ1fuRpPHP0uoB4dtK2w0TQW2NzSun
HkPlrQsnUwQn89Ki9NwZM3WrdDcI18DPpenVl6//yH6dzaJ6kmcNZCGHh8nb0sTl
0rUBy0MiqiirBkEbu/u9AEen2kCKN8Dq63GsoW94r0IBEZBbjkjwEnr4Ul8UvLtLs
a27oJ9/0TEokaFuT5G4qsI4r9NaBbgnmQ0wW2WmHcnCakSmY9Ad8z+WCQ0B5pNSs
5HAJu6ochbIFEH6tr1RGQGkTugUwtZ5RZLL28FYw9znfP0Qm4IJAdsTTwMIhakkH
AMdpR0ewrsj0GQhxZsA4W1Tusy9AZhMMMMMMVk01hnny0V05wr00jNedWm1MVHMeB
hH/CSzv8Y5g7cC9sRDIN5tiHQynJ3d7Yg1RE4Gava8pgzmXvudh0Rewgftp2rgpX
MYox3sm13cwb0xYm2lmDTZQ3FqjNc/fEsHHFUmuW5nswqm7+Eq0GagpbBW68BjvR
KqDG9QKilStzADVb3JuEcvGnXFjNYpDoVD5+qKYM4Ie0jiexA681yyoVFbkKE98R
Cc2ZNMMqy7BISAnYLhJqsyQ0r9Ib+y+nRz6CK6dF8L0Q4j5hpBpAYgg+hk9Ew89p
85C4x9tPjFLaW3UHaJ40TXM8rNAkVt60quunks/2lzFzKv8VFZvEo8cv0XCQ2GowN
```

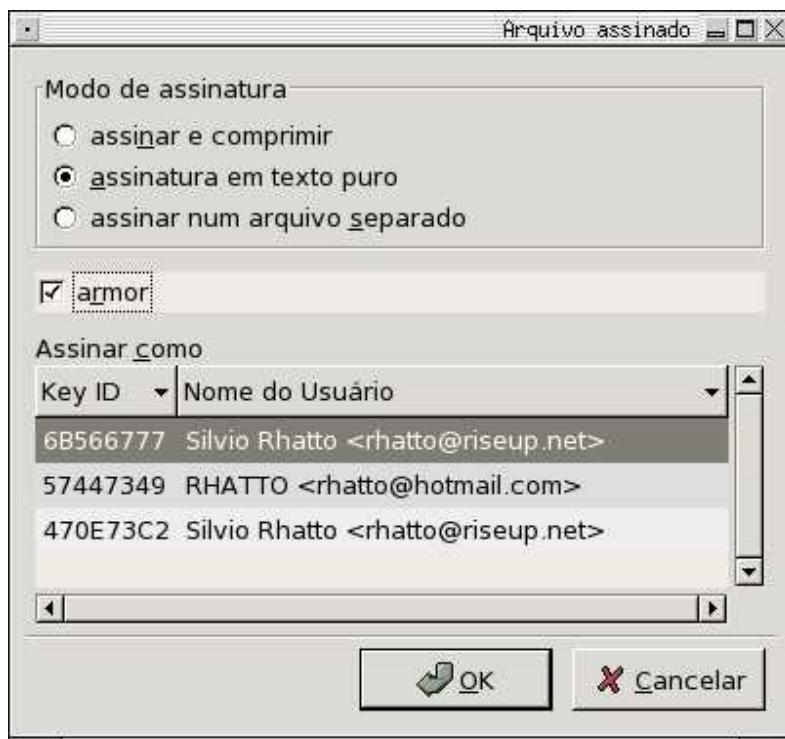
```
QBaJ/EyJ6qFXY0DYJgGkQSsR9RWFFD1Ah9zw98g60dRCzIzz00VdSD8pp6tPtfcl  
ABNKz4/QplbrSPKQFKmEm5fNhEn6G00+m4oXr02+qn6YH31Nk6NuJNx60ZtgZMZ  
ZBWruu9ubusko/NWnhVjBpcmxPJy3g6BrdxG3b0DA40QPfNoryB2sNZU000wf4++  
8Um0j8kSQz3sDDSKzSFYHXdD+9/rGhAF9dVbZK6Zi6lSsNhBGrrdWsIIsvk949WN  
p0pRBu7aRDwPEkDSvJeicoNdv/uvCreUTZACopLtNxLE52rHG90ku2WAq0E6Tcr3  
La/akik6zLR60smoMole5ero9T0d0X94XZ0j9K6JiEBW7c0NXNGz58/SP7MFGPAZ  
QWV6UGUmHQIWk+3nJ73NVRqXdIKVk7s60RpE1Qz5gv+n03P4BbskfuP0xoRljUux  
yc299i2dx6InuPsL+dbAx fqzYPVj8AAhAzXpGr9oVzY/uPYtKk0QyXUb6Zqr02z  
YyrAIDCGKyN6SbEvM+dLNfkEptFUrtpAVcv6f9AunsBYjdUJ02SK1DbpTvTv7k9+  
Szl+cqnJ2F7k5LW6L3klgr1rzfefstd7c16FFAbyclXTYX+QdBC4/o0sNKgYaqqV  
3jlH7ZUHFdqiAWS0MLF8PB2AREARWgrgmC9smEE7ILPANmrGPZRGYHMwLRvJlijg  
WYD4nKFAoJ7eIJQCPhmqvQ5wBr3DLetXwZXSRv1CgQWhkek1HE/ZwvC0i4oFuc3H  
03mlCWV54otm2UD0/vy5m7ECVt/y9qBORdZMv92pu6bmp8uND0736deMPVViISwh  
nmiRMwVlIgnHHu1KKhkVGxWnwkrBbvHhVOJn0DiXNV7QRivnhw2Z0BXZTs2Mu5bS  
  
c+3wi65SSYNwAcVeCQ0X6Q==  
=hG0M  
-----END PGP MESSAGE-----
```

Duvido que você desvende o que está escrito nela!

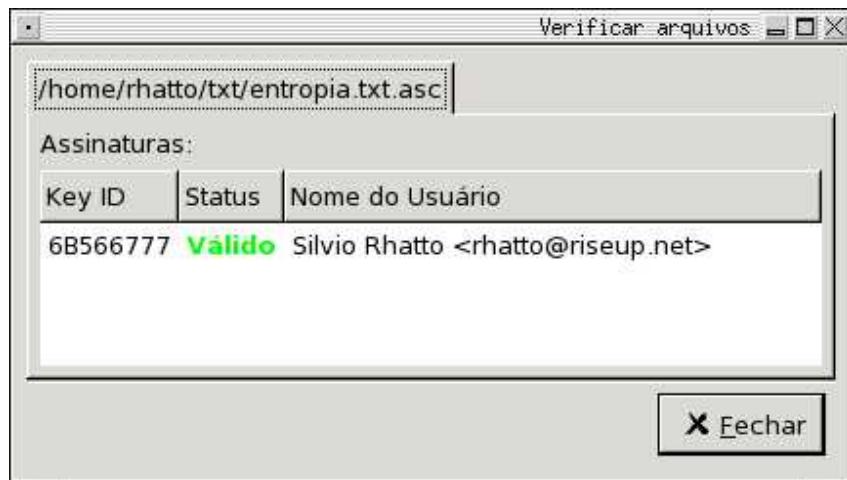
### Descriptar

Para descriptar, selecione o arquivo criptografado e clique em "Descriptar". Você terá que digitar sua senha e então o arquivo será decodificado. Lembre-se sempre: você só pode decodificar arquivos que foram codificados usando sua chave pública e o mesmo vale para seus amigos: eles só conseguirão decodificar mensagens que foram codificadas com as chaves públicas deles.

### Assinando arquivos



## Como verificar mensagens assinadas



## Windows: usando o WinPT

Se você instalou o WinPT de acordo com a seção [Instalando o WinPT](#), tudo fica muito, mas muito fácil. O WinPT deve aparecer como um ícone em forma de chave no canto da sua barra do Windows, provavelmente ao lado do relógio. Se o ícone não estiver lá, basta abrir o menu Iniciar, ir em "Programas" e achar o atalho para o WinPT.

Trataremos aqui apenas das tarefas mais básicas: como encriptar e desencriptar arquivos, como encriptar e desencriptar texto que você recortou (ctrl-x) ou copiou (ctrl-c) e como gerenciar suas chaves e as chaves públicas de outras pessoas.

## Como gerenciar chaves públicas

Para você mandar uma mensagem criptografada para uma pessoa, é necessário que você tenha a

chave pública dela. Para cada pessoa, uma chave pública.

Existem basicamente dois métodos para que você transmita sua chave pública:

- Você enviar sua chave para uma pessoa (por email, disquete, cd, etc.).
- Você enviar sua chave pública para um servidor e cada pessoa que quiser usá-la baixa a chave do servidor.

Enviar sua chave para um servidor é o meio mais cômodo de compartilhar sua chave com as pessoas que vão enviar mensagens criptografadas para você. Dessa forma você não precisa ficar mandando suas chave pública pra todo mundo.

Isso tudo também vale para você obter chaves públicas de outras pessoas: você pode pegar a chave pública de uma pessoa através do servidor de chaves (desde que a pessoa tenha disponibilizado sua chave em servidores) ou recebendo a chave da pessoa via email, disquete, cd, etc.

No WinPT, assim como nos outros programas de criptografia, há um Gerenciador de Chaves, uma espécie de catálogo de endereços, mas que, ao invés de armazenar só o nome e o endereço de email da pessoa, ele guarda também a chave pública das pessoas.

Para abrir o gerenciador de chaves, clique com o botão direito sobre o ícone do WinPT que fica no canto da sua barra do Windows (o ícone se parece com uma chave). Um menu com várias opções deve aparecer. Clique em "Key Manager" (Gerenciador de Chaves).

### Como importar uma chave pública

No **Key Manager**, vá no menu **Key** e depois em **Import...** Aparecerá uma janela para seleção do arquivo. Escolha o arquivo que contém a chave pública da pessoa e clique em **Ok**. Uma janela de confirmação, contendo as informações mais importantes dessa chave pública, vai aparecer. Clique em **Import** e depois em **Ok**. Se a nova chave pública não apareceu, não se preocupe: feche e abra de novo a janela do **Key Manager**.

### Como exportar uma chave pública

No **Key Manager**, selecione a chave que você quer exportar e vá no menu **Key** e depois em **Export**. Uma janela vai abrir para que você escolha o nome do arquivo que conterá a chave pública e o local onde ele será gravado. Pronto.

### Como importar ou exportar uma chave pública de um servidor

Para importar uma chave pública que esteja armazenada em algum servidor na internet, abra o **Key Manager** e clique sobre o menu **Keyserver**. Aparecerá uma janela com um campo de busca. Digite o email da pessoa ou o fingerprint (impressão digital) da chave pública dela. Se a busca retornar sucesso, é só adicionar essa nova chave em sua lista.

Para exportar sua chave pública para um servidor, vá no **Key Manager** e clique sobre ela com o botão direito. Isso abrirá um menu. Vá em **Send to Keyserver** e selecione o servidor para o qual você quer enviar sua chave.

### Encriptar textos com Ctrl-c ou Ctrl-x

Primeiro, digite num editor o texto que você quer criptografar. Em seguida, selecione o texto com o mouse e digite ctrl-c. Com o botão direito do mouse, clique sobre o ícone do WinPT que fica no canto da sua barra do Windows (o ícone se parece com uma chave). Um menu com várias opções deve aparecer. Clique em "Clipboard" e depois em "Encrypt".

### Descriptar textos com Ctrl-c ou Ctrl-x

Com o botão direito do mouse, clique sobre o ícone do WinPT que fica no canto da sua barra do Windows (o ícone se parece com uma chave). Um menu com várias opções deve aparecer. Clique em "Clipboard" e depois em "Decrypt".

### Encriptar e descriptar arquivos

Para saber encriptar arquivos, é necessário que você saiba o que é um arquivo, como copiar e mover arquivos usando o Windows Explorer e saber o que é a extensão de um arquivo. Se você não souber essas coisas, pule esse seção, pois um mal uso dela pode prejudicar sua privacidade. Por exemplo, imagine se você confundir e mandar pra alguém a mensagem original ao invés da mensagem criptografada. Esse tipo de coisa é muito comum quando não se domina bem a manipulação de arquivos.

Com o botão direito do mouse, clique sobre o ícone do WinPT que fica no canto da sua barra do Windows (o ícone se parece com uma chave). Um menu com várias opções deve aparecer. Clique em "File manager" (gerenciador de arquivos).

Uma janela com uma lista em branco vai aparecer. É o gerenciador de arquivos do WinPT. Nele, você pode criptografar e descriptografar arquivos. Antes de efetuar qualquer uma dessas operações, você precisa adicionar os arquivos à lista do gerenciador. Existem duas maneiras de se fazer isso.

- A primeira delas é arrastar o ícone do arquivo desejado até a janela do gerenciador.
- A outra maneira é ir no menu "File" (arquivo) do gerenciador e depois em "Open" (abrir). Aí é só adicionar o arquivo desejado.

### Windows: usando o GPGee

- <http://gpgee.excelcia.org/>

## 4. Criptografia e internet

### Criptografia e correio eletrônico

O uso mais frequente da criptografia é no envio e recebimento de emails. Uma vez que os pacotes de informação são transmitidas de servidor em servidor pela internet até chegar no computador de destino, qualquer pessoa pode monitorar esses pacotes e obter seu conteúdo. Utilizando a criptografia assegura que apenas o destinatário compreenderá o conteúdo da mensagem.

No caso do correio eletrônico as coisas são ainda um pouco mais complicadas, pois não existe dificuldade nenhuma em enviar emails falsos. Isso não é uma vulnerabilidade, mas sim uma características do serviço de email. Qualquer pessoa pode acessar um servidor de email e redigir uma mensagem em nome do Presidente da República. Basta que você configure seu programa de email

para que envie suas mensagens como [presidente@brasil.gov.br](mailto:presidente@brasil.gov.br). Isso é possível devido à própria arquitetura do sistema de email existente na internet.

Além disso, um email passa por muitos servidores até chegar ao seu destino. Se a mensagem não estiver criptografada, ela pode ser facilmente interceptada por terceiros. Isso que estou falando não são falhas do sistema de email, mas sim características deles. Vejamos com um pouco mais de detalhe como tudo isso funciona. Se preferir, pule para a [próxima seção](#).

Na internet existem três tipos de programas de email: os programas do tipo MTA (Mail Transport Agent), os MDA (Mail Delivery Agent) e ou MUA (Mail User Agent).

Os Mail Transport Agents (ou Agentes de Transporte de Emails) são aqueles programas que enviam a mensagem de email para o servidor de destino, que pode ser, por exemplo, o servidor de emails do provedor de internet usado pelo destinatário da mensagem. Por causa do protocolo que os MTAs utilizam, eles são mais conhecidos como servidores SMTP (Simple Mail Transfer Protocol).

Já os MDAs (Agentes de Entrega de Email) são os programas que enviam a mensagem até o usuário de destino, e para que este usuário possa receber a mensagem ele deve utilizar um programa do tipo MUA, que são os programas de email propriamente ditos. Os servidores de entrega de email também são conhecidos como servidores POP ou IMAP. Já os clientes de email (MUAs) podem ser tanto um site de Webmail como os programas do tipo Mail (MacOSX), Mozilla Mail, etc.

Parece complicado?

O caminho que uma mensagem de correio eletrônico percorre é o seguinte: o remetente da mensagem utiliza seu respectivo cliente de email (MUA) para enviar a mensagem até o seu programa MTA, que pode ser o servidor SMTP do seu provedor ou até mesmo ser um programa rodando no computador do remetente. Esse MTA por sua vez envia essa mensagem até o MTA do usuário de destino. Quando o destinatário verificar suas novas mensagens, basta que ele utilize seu cliente de email (MUA) para se conectar até seu servidor POP ou IMAP (MDA) e receber suas mensagens.

Por exemplo, suponha que um usuário do provedor remetente.br deseja enviar uma mensagem para [destinatario@destinatario.br](mailto:destinatario@destinatario.br). O remetente envia o email para o servidor de SMTP do provedor remetente.br, que por sua vez manda a mensagem para o SMTP de destinatario.br. Agora é só o destinatário se conectar ao programa POP, IMAP ou Webmail do provedor destinatario.br para receber a mensagem.

remetente >	smtp.remetente.br >	smtp.destinatario.br >	pop3.destinatario.br >	destinatário
(mua)	(mta)	(mta)	(mda)	(mua)

Em todo esse processo não existe verificação do email de origem da mensagem, isto é, uma vez que o remetente se conectou em smtp.remetente.br ele pode escolher qualquer email de origem. Ele pode escrever a mensagem como [destinatario@destinatario.br](mailto:destinatario@destinatario.br), [ficticio@ficticio.com.br](mailto:ficticio@ficticio.com.br) (não precisa nem ser um email válido). Isso acontece porque normalmente os servidores STMP só requerem usuário e senha para serem utilizados e eles não fazem nenhuma verificação de destinatário. Às vezes esses servidores estão desconfigurados e não precisam nem de senha para serem utilizados, sendo então muito usados para SPAM (mala direta). Um outro detalhe importante é que qualquer pessoa pode rodar seu próprio programa de SMTP e com isso enviar mensagens em nome de outras pessoas.

Já os servidores tipo MDA (pop3, imap, webmail) requerem usuário senha para fornecerem o acesso à uma conta de email, o que impede as pessoas de receberem o email de outras pessoas.

Se você quiser testar o quanto o sistema de correio eletrônico é frágil nesse aspecto da verificação de endereços, experimente utilizar uma ferramenta de email anônimo, como o <http://anony.co.uk/> ou o <http://email.bonloup.org/>. Basta preencher o formulário, escolhendo inclusive o remetente da mensagem, para enviar um email anônimo.

Como na internet todas as informações passam de servidor em servidor até chegar no seu destino, as mensagens de email são naturalmente "interceptadas" por outros servidores. Se essa mensagem passar por algum servidor controlado por pessoas maliciosas, as mensagens de email podem ser lidas com muita facilidade.

Resumindo, qualquer pessoa pode escrever um email com o endereço de email de outra pessoa, mas dificilmente conseguirá receber o email de outras pessoas. No entanto, é **possível** que mensagens de email sejam interceptadas enquanto estiverem indo para o seu destino, e se ela não estiver corretamente criptografada qualquer pessoa pode interpretá-la.

Assim, além de criptografar suas as mensagens, é muito útil assinar com sua chave pública as mensagens de email que você envia. Assim as pessoas só confiarão nas mensagens enviadas com o seu endereço se elas estiverem assinadas com sua chave privada. Por isso, se você não usar criptografia, você não terá como provar que

- As mensagens de email que você não enviou mas que alguém enviou em seu nome não são realmente suas
- As mensagens que você enviou em seu nome são realmente suas

Se você adotar a assinatura de mensagens como padrão, você consegue facilmente provas os dois itens anteriores.

## Programas de email

Nessa seção abordaremos os seguintes programas de email que suportam criptografia,

- Mozilla Mail / Thunderbird
- KMail
- Ximian Evolution

e ainda como utilizar criptografia em Webmail. Se você utiliza Windows, os dois únicos clientes de email que trataremos aqui são o Mozilla Mail e o Thunderbird, pois são softwares livres e tem um bom suporte à criptografia. Considere seriamente trocar de cliente caso você use Outlook, Eudora ou similar.

## Mozilla Mail / Thunderbird

Se você usa Linux, Windows ou MacOS, não deixe de usar o [Mozilla](#). O Mozilla não é apenas um navegador (browser) pra internet de código aberto, mas é **O Navegador**. É uma questão de ética usar o Mozilla. Além disso, o Mozilla possui um programa de email muito bom que tem amplo suporte a criptografia.

A seguir temos instruções de como utilizar criptografia com o Mozilla e com suas versões alternativas (Thunderbird e Firefox).

## Instalando no Mozilla

**1** - Se você ainda não instalou o GPG, [faça-o](#).

**2** - Se você ainda não usar o Mozilla, faça o download dele em <http://www.mozilla.org/releases/> e instale.

**3** - Se você ainda não configurou o Mozilla Mail, vamos lá! Com o Mozilla aberto, vá no menu **Window** e depois em **Mail & newsgroups**. A janela do Mozilla Mail deve aparecer. Se não apareceu é porque você não fez uma boa instalação do Mozilla. Volte ao item **1** e comece tudo de novo.

**4** - Junto com a do Mozilla Mail, vai aparecer uma outra janela pedindo suas configurações de email. Digite-as da mesma forma como você faria num outro cliente de email.

**5** - Depois de configurar sua conta de email, instale o software de criptografia. Para isso, abra o Mozilla como superusuário e vá até a página <http://enigmail.mozdev.org/download.html>

**6** - Na seção **Express Install**, clique no botão **Install**.

**7** - Aparecerá o procedimento padrão pra instalar coisas no Mozilla. É só ir prosseguindo que a instalação é feita sem problemas. Se aparecer algum erro do tipo **Permission Denied** é porque você não executou o Mozilla como superusuário.

**8** - Abra o Mozilla Mail mais uma vez como superusuário e feche em seguida. Não me pergunte porque tem que ser assim, apenas faça.

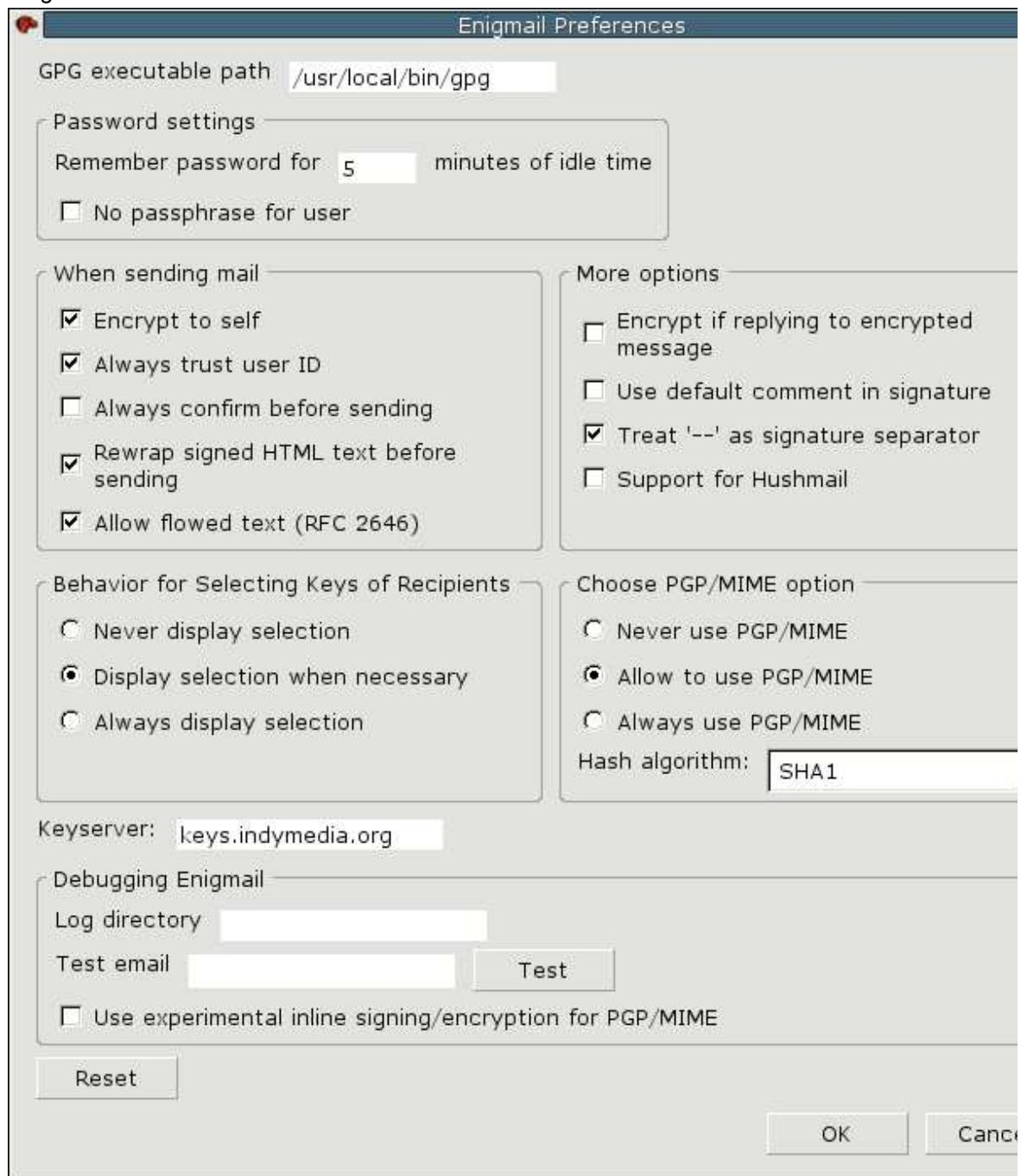
**9** - Abra o Mozilla Mail novamente, mas agora com o seu usuário. No menu deverá aparecer o item **Enigmail**. Se ele estiver, beleza. Senão, a instalação não foi bem sucedida. Recomece 😞

**10** - Vá nesse menu (Enigmail) e depois em **Preferences** (preferências) e configure o Enigmail. O mais importante aqui é dizer ao Enigmail onde está o GPG (no Linux, pode ser /usr/bin/gpg ou /usr/local/bin/gpg e no Windows pode ser c:\windows\gpg.exe ou na pasta na qual você fez a instalação).

- Enigmail:

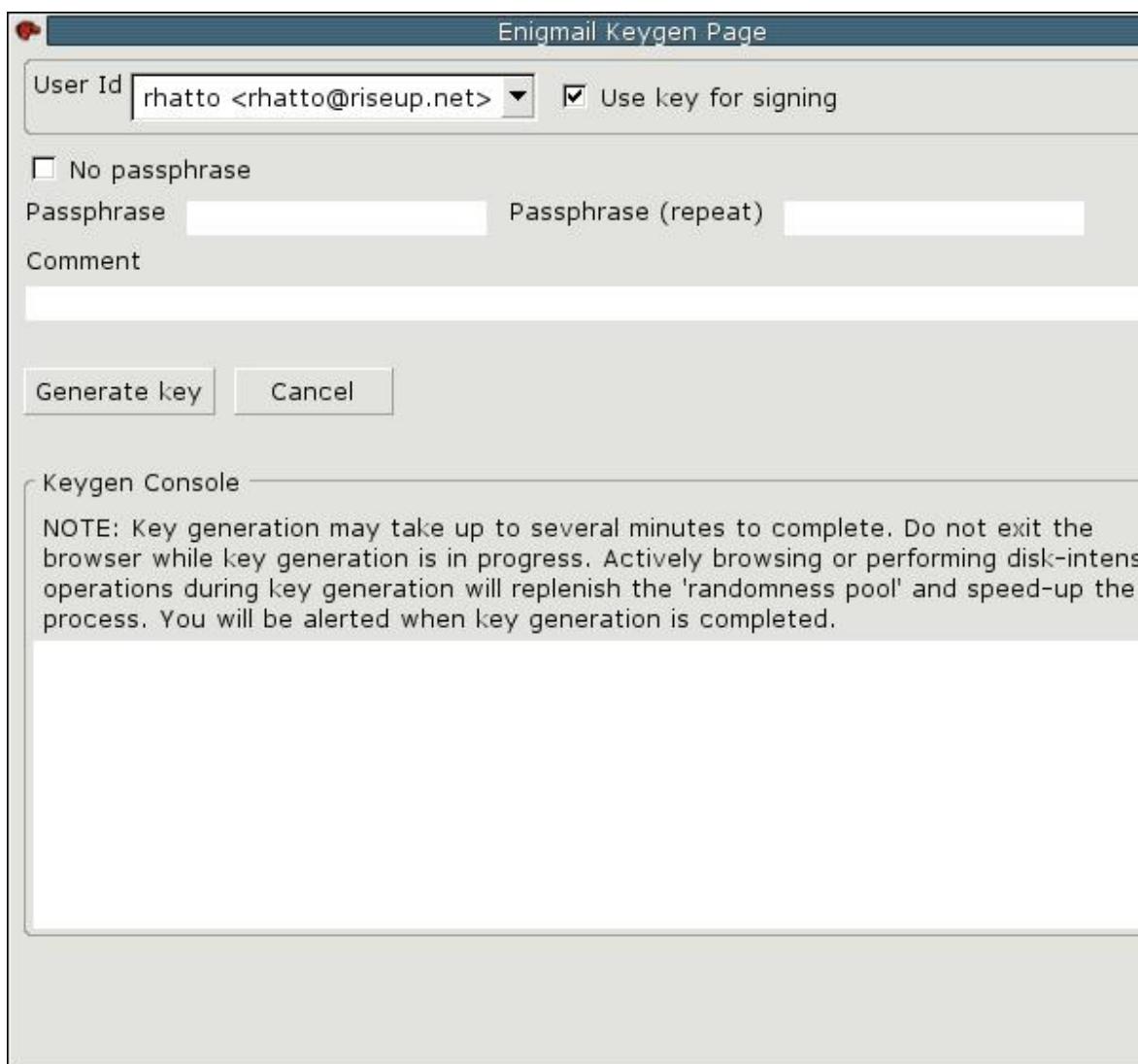


- Enigmail:



**11** - Em seguida chega a hora de escolher seu par de chaves que o Enigmail utilizará. Aqui você tem três opções: gerar um par de chaves pelo próprio Enigmail, gerar utilizando outro programa de criptografia e utilizá-lo no Enigmail ou então usar um par de chaves que você já tem. Gerar diretamente pelo Enigmail é a forma mais simples e pode ser feita de acordo com as duas figuras abaixo.

- Enigmail:



- Enigmail:



Se você gerou suas chaves num outro programa ou já possuía um par de chaves, basta ir no menu **Account Settings** do Mozilla Mail e em seguida em **OpenPGP Security** e no campo **Use specific PGP Key** digite o **ID** da sua **chave pública** já existente.

### Instalando no Thunderbird

O Thunderbird é uma versão do Mozilla Mail mais econômica em termos de custo computacional, isto é, é um programa que roda melhor do que o Mozilla Mail em computadores mais lentos. O Thunderbird

pode ser baixado e instalado a partir [daqui](#). Em conjunto com o Thunderbird, é recomendável que você utilize o Firefox como navegador Web. O Firefox é uma versão leve do Mozilla e pode ser baixado a partir [daqui](#).

A instalação do Enigmail no Thunderbird é similar à do Mozilla Mail, porém a menor integração entre o navegador e o leitor de emails faz com que algumas coisas tenham que ser feitas "na mão":

**1** - Se você ainda não instalou o GPG, [faça-o](#).

**2** - Se você ainda não usar o Thunderbird e o Firefox, faça o download deles em <http://www.mozilla.org/products/thunderbird/> e <http://mozilla.org/products/firefox/> e em seguida instale-os.

**3** - Se você ainda não configurou o Thunderbird, vamos lá! Com ele aberto, você deverá ver uma opção do tipo **Create a new account** (Criar Nova Conta). Clique ali e configure suas opções de email, da maneira como é feita em qualquer outro cliente de correio. Em seguida, feche o Thunderbird.

**4** - Agora é importante que você saiba com certeza qual é a versão do seu Thunderbird, já que para cada Thunderbird há uma versão correspondente do Enigmail.

**5** - Depois de configurar sua conta de email, instale o software de criptografia. Para isso, abra o Firefox e vá até a página <http://enigmail.mozdev.org/download.html>

Lá existe uma tabela contendo o endereço para baixar o Enigmail para cada versão do Thunderbird. Você precisará instalar dois arquivos, um para o **enigmail** e outro para o **enigmime**. Por exemplo, se você instalou o Thunderbird 0.6 e você esteja usando o Linux, vá até a linha que começa com **Thunderbird 0.6** e baixe os arquivos indicados, que no caso são o [enigmail-0.84.0.xpi](#) e o [enigmime-0.84.0-linux.xpi](#).

Se você usa Windows ou MacOS, não se preocupe. O site do Enigmail detectará qual é o sistema operacional que você está rodando e de acordo com essa informação listará os tipos de Enigmail que você pode baixar. Lembre-se apenas que é indispensável que você baixe os arquivos **enigmail** e **enigmime** correspondentes à sua versão do Thunderbird.

**6** - Abra o Thunderbird como superusuário, para que você tenha poderes para instalar o Enigmail. Vá no menu **Tools**, depois em **Options** e **Extensions** e **Install New Extensions**. Lá você pode instalar o Enigmail, bastando para isso selecionar os arquivos enigmail-0.84.0.xpi e enigmime-0.84.0-linux.xpi baixados previamente conforme as instruções do item anterior.

Aparecerá o procedimento padrão pra instalar coisas no Mozilla. É só ir prosseguindo que a instalação é feita sem problemas. Se aparecer algum erro do tipo **Permission Denied** é porque você não executou o Mozilla como superusuário.

**7** - Abra o Thunderbird mais uma vez como superusuário e feche em seguida. Não me pergunte porque tem que ser assim, apenas faça 😊

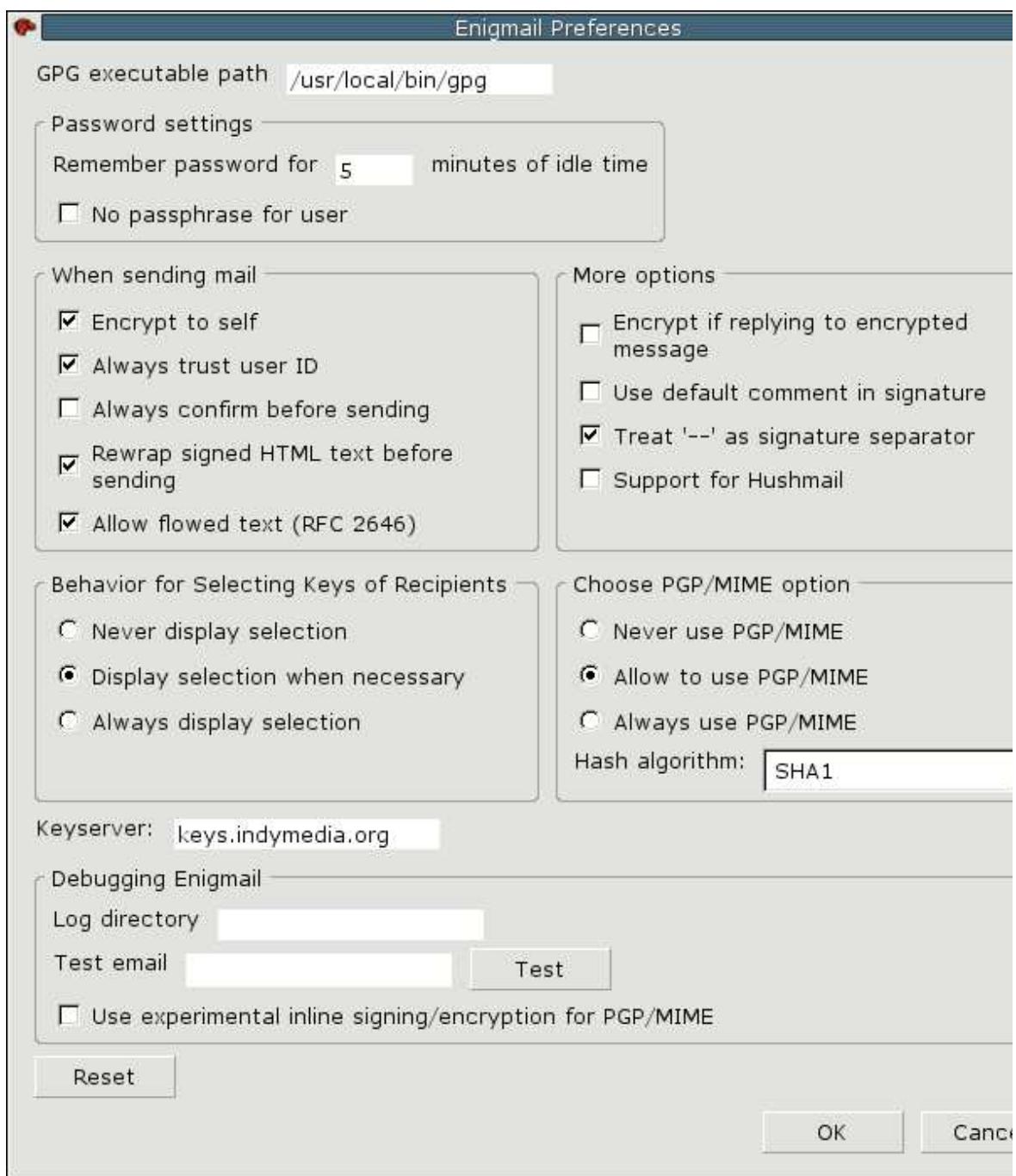
**8** - Abra o Thunderbird novamente, mas agora com o seu usuário. No menu deverá aparecer o item **Enigmail**. Se ele estiver, beleza. Senão, a instalação não foi bem sucedida. Recomece 😊

**9** - Vá nesse menu (Enigmail) e depois em **Preferences** (preferências) e configure o Enigmail. O mais importante aqui é dizer ao Enigmail onde está o GPG (no Linux, pode ser /usr/bin/gpg ou /usr/local/bin/gpg e no Windows pode ser c:\windows\gpg.exe ou na pasta na qual você fez a instalação).

- Enigmail:

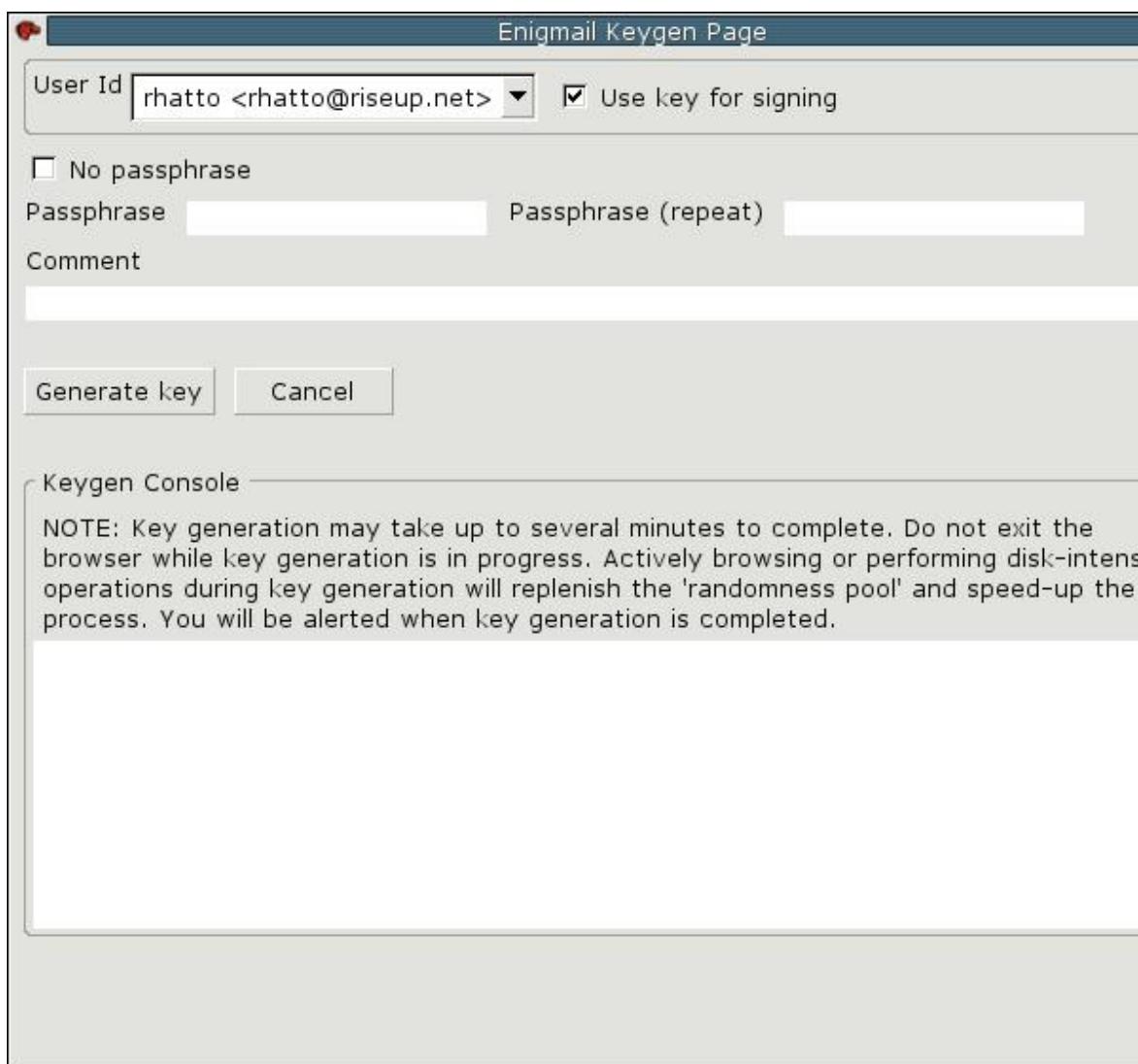


- Enigmail:



**10** - Em seguida chega a hora de escolher seu par de chaves que o Enigmail utilizará. Aqui você tem três opções: gerar um par de chaves pelo próprio Enigmail, gerar utilizando outro programa de criptografia e utilizá-lo no Enigmail ou então usar um par de chaves que você já tem. Gerar diretamente pelo Enigmail é a forma mais simples e pode ser feita de acordo com as duas figuras abaixo.

- Enigmail:



- Enigmail:



Se você gerou suas chaves num outro programa ou já possuía um par de chaves, basta ir no menu **Account Settings** do Mozilla Mail e em seguida em **OpenPGP Security** e no campo **Use specific PGP Key** digite o **ID da sua chave pública** já existente.

### Como verificar mensagens assinadas

- Enigmail:

**Enigmail:** Good signature from Silvio Rhatto <rhatto@riseup.net>, Key Id 0x6B5667

## Enviando mensagens encriptadas

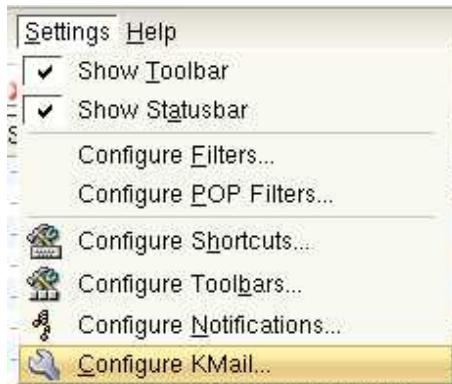
Para fazer isso é só compor uma mensagem normalmente e enviá-la apertando o botão **EnigSend**.

## Recebendo mensagens criptografadas

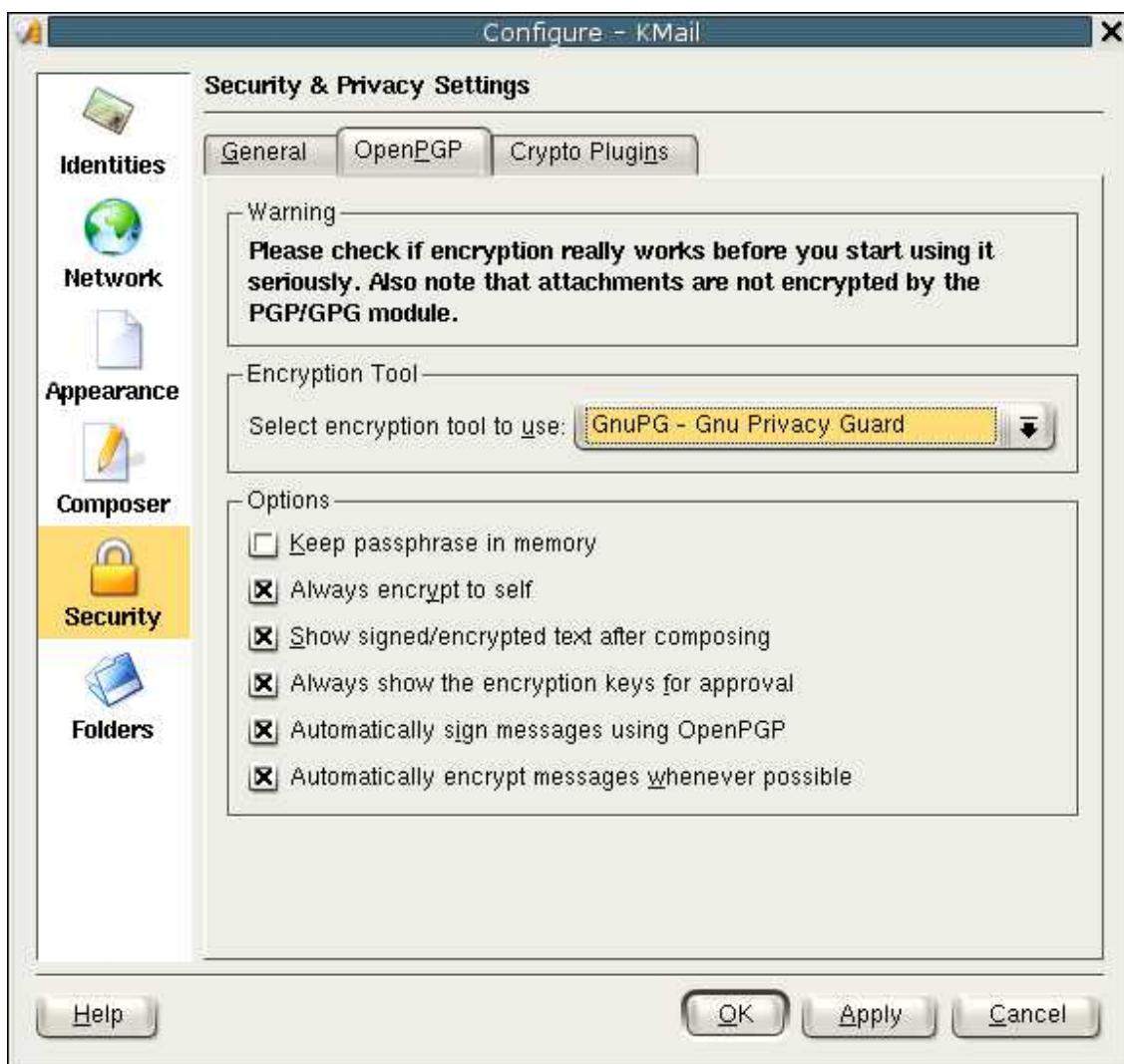
O Enigmail está configurado para automaticamente decodificar mensagens que você receber pelo Mozilla Mail.

## Usando o KMail

- KMail:



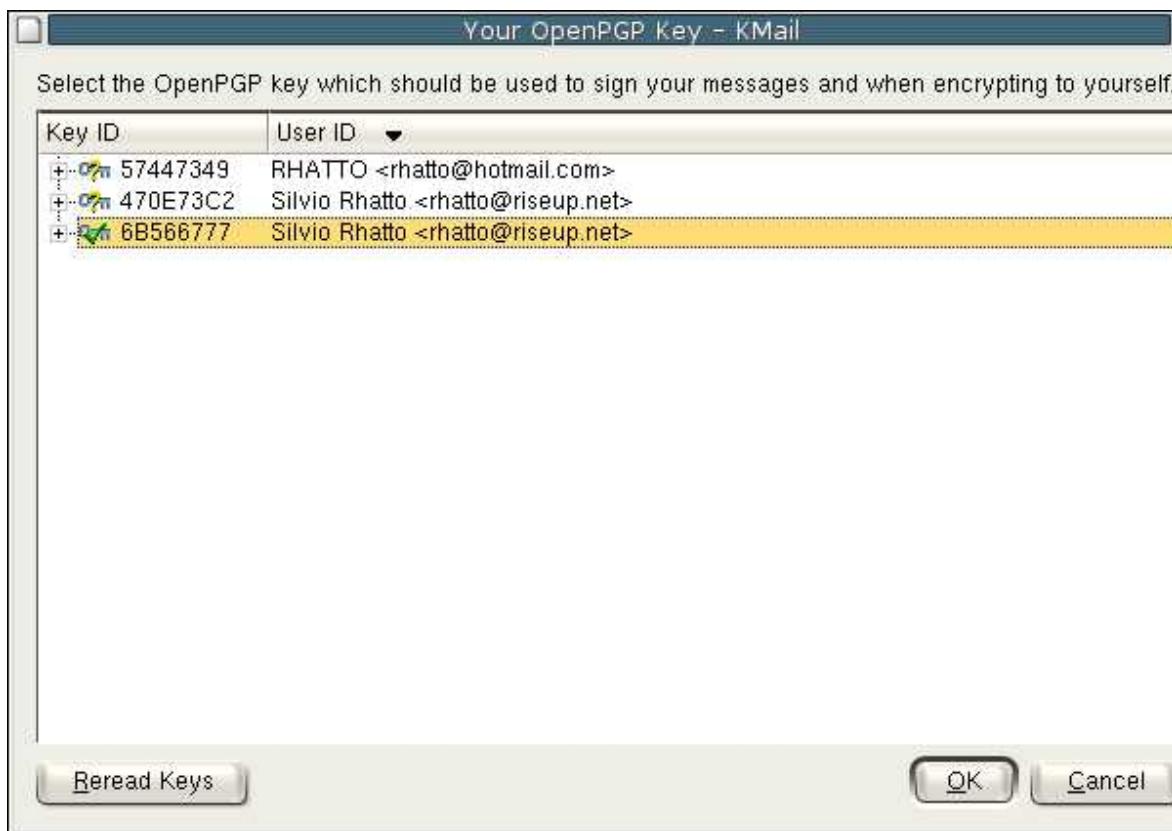
- KMail:



- KMail:



- KMail:



- KMail:



## Criptografia em listas de discussão

Existem alguns softwares que já permitem o uso de criptografia em pequenas listas de discussão por email, funcionando da seguinte forma: a lista possui uma chave pública e a chave pública de todos os assinantes. Quando alguém quiser mandar uma mensagem criptografada para a lista, basta criptografá-la apenas para a chave pública da lista que a lista irá descriptografá-la, criptografar e enviar individualmente para cada assinante usando sua respectiva chave pública.

Os seguintes softwares de lista de discussão suportam criptografia padrão [OpenPGP](#):

- [firma](#): é um programa pequeno e eficiente gerenciador de listas criptografadas; ele foi desenvolvido para ignorar mensagens enviadas à lista que não estejam criptografadas e assinadas
- [mailman](#): o mailman é um software de listas que possui suporte parcial à criptografia através

dos patches (remendos) [NAH6](#) e [GPG-Mailman](#)

## Conexão segura: criptografia na rede

Vou repetir o que escrevi numa seção anterior: Uma vez que os pacotes de informação são transmitidas de servidor em servidor pela internet até chegar no computador de destino, qualquer pessoa pode monitorar esses pacotes e obter seu conteúdo. Utilizando a criptografia assegura que apenas o destinatário compreenderá o conteúdo da mensagem.

Se você estiver visitando um site e em algum momento precisar entrar com uma senha ou qualquer outra informação a ser enviada via formulário, terceiros podem interceptar essa informação e se ela não estiver criptografada, pode ser interpretada por qualquer um.

Para contornar esse problema foram criados diversos protocolos que utilizam criptografia pela internet em tempo real. Por exemplo, quando navegamos na Web nosso navegador utiliza o protocolo HTTP ([HiperText Transfer Protocol](#) - Protocolo de Transferência de Hipertexto), que não suporta nenhum tipo de criptografia. Já o HTTPS (Secure HTTP - HTTP Seguro) foi feito para navegarmos na web de forma um pouco mais segura. Vejamos como funciona:

Quando você usa um navegador web ([Mozilla](#), por exemplo) e se conecta num site utilizando o protocolo de conexão segura (HTTPS), seu navegador e o servidor do site trocam automaticamente suas respectivas chaves públicas e então é iniciada uma transmissão de informações criptografadas.

**ATENÇÃO:** seu navegador não vai enviar ao servidor a chave pública que você criou, mas sim uma chave própria criada automaticamente pelo seu navegador de tal forma que você não precisa de nenhum programa adicional de criptografia (nem mesmo o GPG). Tudo isso é feito de forma praticamente transparente ao usuário.

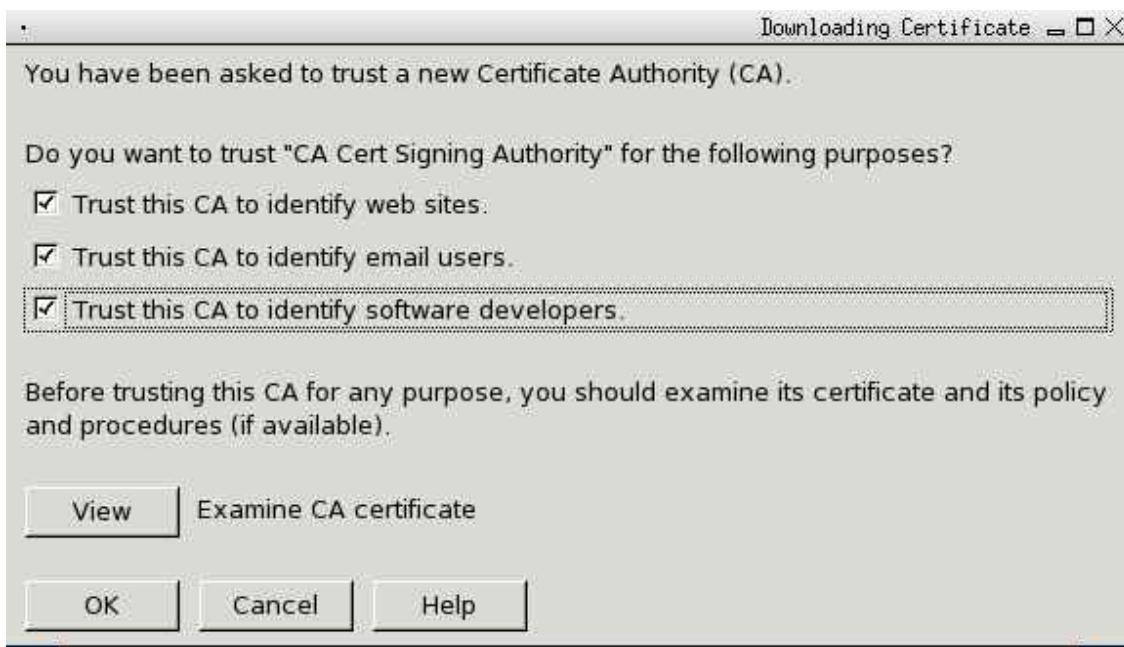
A pergunta natural é: **como podemos confiar que o site que estamos acessando realmente é o site que ele diz ser; em outras palavras, existe um jeito de confiarmos na chave pública do site?**

De uma forma parecida como quando trocamos a impressão digital de nossa chave pública com a de outras pessoas, o site com conexão segura enviará para o seu navegador um **certificado de autenticidade**, que é uma espécie de assinatura da chave pública do site emitida por uma **autoridade certificadora**. Uma autoridade certificadora é qualquer pessoa, organização ou empresa que "assine" certificado de autenticidade. A Autoridade Cerificadora (ou Certificate Authority) mais conhecida é a [CAcert.org](#), uma organização sem fins lucrativos que valida tais certificados.

Tudo que o usuário precisa fazer é

- Confiar na autoridade certificadora
- Instalar o certificado da autoridade certificadora
- Confiar que o certificado instalado é o certificado verdadeiro dessa autoridade, bastando para isso que você verifique a impressão digital desse certificado.

Por exemplo, você pode tentar baixar o arquivo <http://www.cacert.org/cacert.crt>, que contém o certificado principal da Cacert.org. Na maioria dos navegadores, abrirá uma janela perguntando se você aceita a Cacert como uma autoridade certificadora.

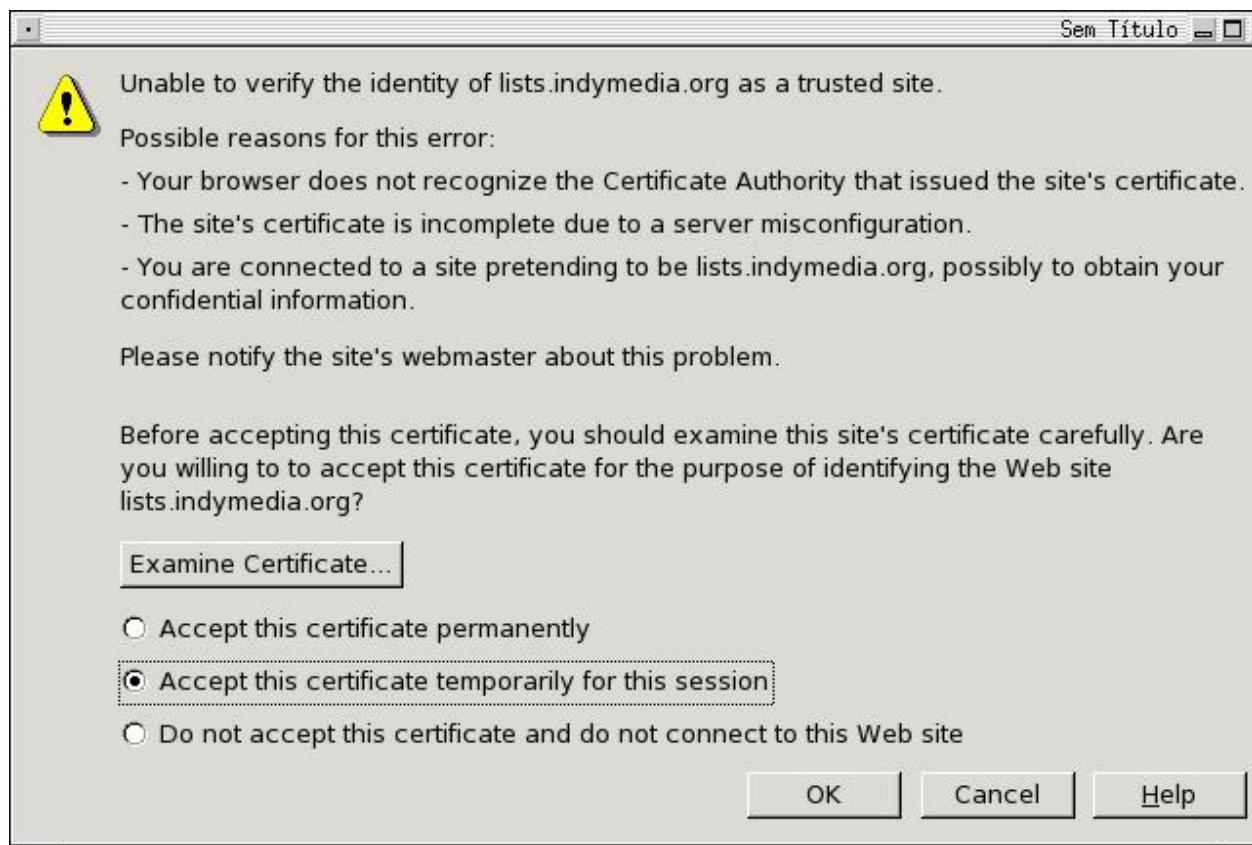


Se você tem dúvidas para aceitar, veja os detalhes do certificado, conferindo se **fingerprints** do certificado corresponde àqueles que a autoridade certificadora divulga em seu site.



O processo da conexão segura via web é o seguinte: suponha que você esteja visitando o site [lists.indymedia.org](http://lists.indymedia.org). Se você quiser fazer isso via conexão segura, você digitará <https://lists.indymedia.org> ao invés de <http://lists.indymedia.org> (note a diferença de https ao invés de http).

Na primeira vez que você acessar um site via conexão segura, caso você não tenha instalado o certificado da autoridade certificadora que validou o certificado desse site, seu navegador lhe informará a respeito e perguntará se você quer mesmo assim aceitar o certificado desse site, conforme mostra a figura abaixo.



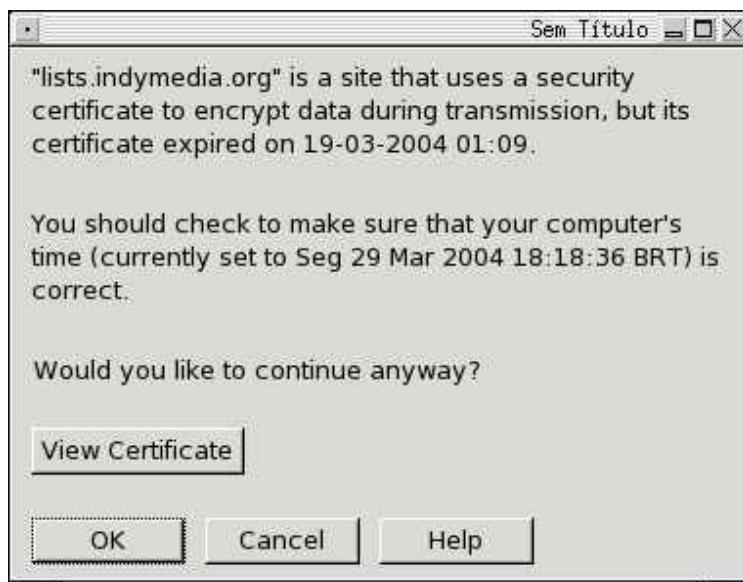
Agora você tem três opções:

- Confiar nesse certificado e clicar em **Ok** ou
- Ir até o site da Autoridade Certificadora que emitiu o certificado do site, instalar o certificado da autoridade e em seguida voltar para o site que você estava tentando acessar via conexão segura ou então
- Desistir de acessar o site 😊

Se clicarmos em **Examine certificate...** (Examinar certificado...), aparecerá uma janela como mostra a figura abaixo. Nela, as impressões digitais do certificado do site <http://lists.indymedia.org> estão em **SHA1 Fingerprint** e **MD5 Fingerprint**. O campo **Issued to** (emitido para) informa para qual site e organização o certificado, enquanto que o campo **Issued by** (emitido por) diz qual foi a autoridade certificadora que emitiu esse certificado. O campo **Validity** (validade) mostra qual é o prazo de validade do certificado.



Quando passa o prazo de validade, o certificado expira e o navegador não mais o reconhecerá como válido e ao acessar o site aparecerá uma janela mais ou menos como a abaixo, avisando que o certificado expirou e portanto a conexão segura pode estar comprometida.



Da mesma forma como é possível navegar pela web com conexão segura, também é possível utilizar outros serviços na internet, como **ftp** (transmissão de arquivos) e **irc** (bate papo), de forma criptografada.

## Criptografia como proteção de software

A criptografia permite que qualquer tipo de arquivo seja assinado. Se você é desenvolvedor de algum software, é muito conveniente assinar os arquivos dos programas que você escreve, pois assim você garante aos usuários que os programas que você escreve estão livres de vírus ou rotinas que podem comprometer a privacidade e o sistema dos usuários.

Tudo que o usuário do precisa fazer é, além de baixar o código ou o seu programa já pronto, baixar a chave pública do programador e também um arquivo contendo a assinatura do programa feita pelo programador e verificá-la como se estivesse verificando a assinatura de um arquivo comum.

## 5. Criptografando seu disco rígido

Uma vez que o seu chaveiro fica armazenado no seu disco rígido, é possível que ele seja copiado por terceiros. Além disso, você pode querer que dados no seu disco sejam criptografados e descriptografados de forma transparente, isto é, sem que você perceba.

Atualmente existem várias implementações de criptografia em disco rígido, tanto em plataforma Linux quanto Windows. Futuras versões deste manual conterão explicações mais detalhadas sobre cada uma delas. Por enquanto, fique com as seguintes referências:

### No Linux

No GNU/Linux, existem as seguintes implementações:

- [dm-crypt](#): a nova tecnologia de criptografia em discos do linux
- [loop-aes](#): uma alternativa às implementações oficiais do linux
- [LUKS - Linux Unified Key Setup](#): um possível novo padrão para criptografia em disco no GNU/Linux; baseado na especificação [TBS1](#) e possui suporte ao dm-crypt.

- [eCryptfs](#): esquema bem prático de criptografia em disco, é o padrão mais recente desta lista.

Estas são implementações um pouco mais antigas, algumas já em desuso ou com o desenvolvimento interrompido:

- [Phonebook](#): implementação da chamada *deniable encryption*, que é a possibilidade do usuário revelar apenas alguns pedaços da informação criptografada caso ele seja intimidado a fazê-lo (por exemplo, no caso de tortura); isso é feito através de um esquema de *camadas*, cada uma delas contendo seu próprio conteúdo criptografado.
- [cryptoloop](#): a tecnologia ainda em uso mas já obsoleta
- [CryptoFS](#): utiliza o [Linux Userland FileSystem](#) para criptografia; um diretório fica com os arquivos criptografados e o acesso aos dados é feito através do ponto de montagem.
- [PPDD](#): cria um dispositivo que criptografa automaticamente os dados numa partição, conceito semelhante ao dm-crypt, mas neste caso não utiliza o device-mapper.
- [CFS](#): sistema de arquivos criptografado que utiliza o NFS como interface.
- [EncFS](#): sistema de arquivos criptografado em nível de usuário, utilizando o [FUSE](#).
- [TCFS](#): Transparent CFS (outra implementação do CFS).
- [StegFS](#): um sistema de arquivos esteganográfico.

Cada uma delas tem suas [vantagens e desvantagens](#). Para o sistema ficar transparente, existe o [pam\\_mount](#), que monta seu sistema de arquivos criptografado automaticamente após você entrar com seu usuário e desmonta logo que você sai.

## No Windows

No Windows, existe o [TrueCrypt](#), um programa que permite você criptografar arquivos comuns e discos inteiros e tratá-los transparentemente como se fossem discos comuns.

## Nota sobre Modo Texto e Modo Gráfico

Existem dois modos de interação entre o usuário e o computador, o modo texto e o modo gráfico. No modo texto o usuário interage através de comandos fornecidos via teclado, o mouse quase não é usado e a tela do computador contém apenas caracteres, sem a possibilidade de visualizar imagens.

Já o Modo Gráfico possibilita que grande parte da interação do usuário com o computador seja feita com mouse, via botões e outros objetos.

O Modo Gráfico é muito mais intuitivo e simples de usar, porém no caso da criptografia os programas em modos gráfico têm muito menos recursos que o GPG no modo texto. Por isso, é interessante que o usuário tenha noções tanto de usar o modo texto quanto o modo gráfico. Mas se você estiver com pressa ou tem preguiça de aprender os maravilhosos comandos do modo texto (também conhecido como **console**), leia ao menos as seções deste manual sobre programas no modo gráfico.

## Resumão: tabela de consulta rápida

O GPG no modo texto apresenta muitos comandos e frequentemente nos esquecemos dos parâmetros e da ordem pela qual eles precisam ser passados ao programa.

- Criar par de chaves: **gpg --gen-key**
- Compartilhar chave pública: **gpg --export --armor -o chave.asc [email@do.usuario](mailto:email@do.usuario)**

- Enviar chaves a um servidor: **gpg --keyserver servidor.de.chaves --send-keys nome-da-chave**
- Listar chaves do seu chaveiro: **gpg --list-keys**
- Importar chaves: **gpg --import nome-do- arquivo**
- Procurar chave num servidor: **gpg --keyserver keys.indymedia.org --search-keys email-ou-nome**
- Receber chaves de um servidor: **gpg --keyserver servidor.de.chaves --recv-keys id-da-chave**
- Assinar em texto simples: **gpg --clearsign nome-de- arquivo (opcional)**
- Verificar assinatura: **gpg --verify nome-do- arquivo**
- Criptografar mensagem: **gpg -e -a -r nome-ou-mail**
- Criptografar em arquivo: **gpg -r nome-ou-email -e -a nome-do- arquivo**
- Descriptografar: **gpg -d nome-do- arquivo**
- Ver impressão digital: **gpg --fingerprint nome-ou-email**
- Atualizar chaves públicas de um servidor: **gpg --refresh-keys --keyserver servidor-de-chaves**

## Referências

Esse Guia foi escrito para conter tudo o que uma pessoa precisa para usar a criptografia no dia-a-dia. Contudo, algumas coisas ficaram de fora ou pouco aprofundadas. Aqui encontram-se alguns guias com maior detalhamento.

- Uma ótima referência em português para criptografia no Linux (Modo Texto apenas) encontra-se no [Guia Foca Linux](#), de Gleydson Mazioli da Silva.
- [A Practical Introduction to GPG in Windows](#), de Brendan Kidwell.
- [Using multiple subkeys in GPG](#), de Adrian von Bidder

Do próprio sítio do GNU Privacy Guard podemos destacar (muita coisa só em inglês):

- [The GNU Privacy Handbook](#), de John Michael Ashley.
- [Perguntas mais frequentes sobre o GNU PG](#), documento mantido por David D. Scribner.
- [GNU PG Mini-HOWTO](#), por Brenno J.S.A.A.F. de Winter, Michael Fischer v. Mollard e Arjen Baart.
- [GPG no MacOSX](#).
- [Ritual de assinatura de chaves no Debian](#)

## Sobre este manual

Este manual foi escrito por Rhatto (rhatto[-@!]riseup.net) e é baseado numa pequena referência de como instalar o GPG por Pietro Bastardi (pietro[-@!]bastardi.net).



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 2.0 Brazil License](#).

## Mais informações

Se esse manual não estiver esclarecedor e você ainda tem dúvidas ou dificuldades para instalar ou usar algum desses programas, não hesite em enviar um email para o Coletivo Técnico do CMI:  
[cmi-brasil-tech@lists.indymedia.org](mailto:cmi-brasil-tech@lists.indymedia.org)

Que a força esteja com você!

-- [PietroFerrari](#) - 04 Sep 2002  
-- [LuiS](#) - 02 Apr 2004 (Pequenas correções ortográficas)  
-- [SilvioRhatto](#) - 12 Nov 2006

- Set ALLOWTOPICCHANGE = [SilvioRhatto](#), [LuiS](#)

Topic revision: r94 - 13 Sep 2007 - 20:41:49 - [AlsteR](#)

Sysadmin.GnuPGpt moved from Sysadmin.GnuPG-IMCpt on 12 Apr 2005 - 09:38 by [BertAgaz](#) - [put it back](#)

Copyright &© by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding Foswiki? [Send feedback](#)

