Grupo de Resposta a Incidentes de Segurança – UFRJ

Última modificação: 31/03/04

Usando o GNU Privacy Guard (GnuPG)

GRIS-2004-T-001

Universidade Federal do Rio de Janeiro
Instituto de Matemática
Departamento de Ciência da Computação
Grupo de Resposta a Incidentes de Segurança

Rio de Janeiro, RJ – Brasil

Usando o GNU Privacy Guard (GnuPG)

GRIS-2004-T-001

Breno Guimarães de Oliveira

A versão mais recente deste documento pode ser obtida na página oficial do GRIS

GRIS – Grupo de Resposta a Incidentes de Segurança
CCMN Bloco E 2° andar
Salas: E2000 e E2003
Av. Brigadeiro Trompowski, s/nº
Cidade Universitária - Rio de Janeiro/RJ
CEP: 21949-900
Telefone: +55 (21) 2598-3309

Sumário

1 – Introdução

O objetivo deste documento é explicar o uso do *GNU Privacy Guard* (*GnuPG*), uma ferramenta livre e gratuita para comunicação e armazenamento seguro de dados, que pode ser usada para criptografar dados e criar assinaturas digitais (suporta os algoritmos *DSA*[1], *ElGamal*[2], *RSA*[3], 3DES[4], CAST5[5], Blowfish[6], AES[7] e Twofish[8]). O GnuPG obedece o padrão de Internet OpenPGP proposto no RFC 2440 [9] e, assim, visa ser compatível com a ferramenta comercial PGP[10], criada por Philip R. Zimmermann.

O GnuPG é uma ferramenta para linha de comando e sem interface gráfica de instalação. Ambientes gráficos para ele serão vistos na seção 6 deste documento. Até lá, sempre que for indicado um comando a ser digitado pelo usuário, assumimos que o leitor sabe acessar a linha de comando de seu sistema operacional, acessar, se for o caso, o diretório em que o GnuPG foi instalado (ver próxima seção) e executar o comando.

Existem mais de 30 comandos diversos aceitos no modo de edição do GnuPG, e mais ainda no modo normal. Como o objetivo deste documento não é produzir uma descrição exaustiva de cada funcionalidade do programa, apenas os comandos principais serão demonstrados. Para usufruir melhor do GnuPG, explore (com cautela) os demais comandos, ou leia a documentação oficial do GnuPG. Uma breve descrição de cada comando está incluída no final deste documento.

2 – Instalando o GnuPG

O GnuPG deve ser obtido via Internet em seu sítio oficial, *http://www.gnupg.org*. Ele funciona em inúmeros sistemas operacionais, e veremos com mais detalhes o processo de instalação nos sistemas Windows e GNU/Linux, além de indicações sobre outros sistemas, como o MacOS X, Solaris, BSD, etc.

2.1 – Windows

Antes de iniciar a instalação do GnuPG para o Windows, certifique-se de que possui uma ferramenta para descompactar arquivos ZIP (como o CAM Unzip, disponível em *http://www.camunzip.com*) e outra para verificar a assinatura md5 de arquivos (como o md5sum, disponível em *http://downloads.activestate.com/contrib/md5sum/Windows/md5sum.exe*).

Pegue a versão mais recente do **GnuPG para Windows** na seção "*Download* " da página oficial do mesmo (subseção "*Binaries*"), e anote o valor md5 indicado ao lado do nome do arquivo na página oficial. Quando este documento foi desenvolvido, a versão mais recente do GnuPG era a 1.2.4, e portanto a página oficial do GnuPG mostrava os valores:

```
bb568fe26abbe045d91f95ae0324eab2  gnupg-w32cli-1.2.4.zip
```

Uma vez obtido o GnuPG, use a ferramenta *md5sum* para verificar a integridade do arquivo. Isso garante que você está em posse de uma versão oficial que não foi modificada e, portanto, não deve possuir código malicioso:

Entre no prompt de comando (antigo DOS), cujo atalho provavelmente está em "Iniciar/Programas/ Acessórios/Prompt de comando". Agora entre no diretório em que você colocou o arquivo ZIP do GnuPG que acabou de baixar e o arquivo *md5sum.exe*. Por exemplo, se eles estão no diretório C:\gnupg, digite:

```
C:\> cd gnupg
C:\gnupg>
```

Em seguida, chame o programa md5sum seguido do nome do arquivo do GnuPG (no nosso exemplo, o arquivo chama-se "`gnupg-w32cli-1.2.4.zip`")

```
C:\gnupg> md5sum gnupg-w32cli-1.2.4.zip
```

A seguinte mensagem deve aparecer:

```
bb568fe26abbe045d91f95ae0324eab2 *gnupg-w32cli-1.2.4.zip
```

Como o valor md5 que aparece antes do nome do arquivo é igual àquele da página oficial do GnuPG, o arquivo está correto e podemos prosseguir com a instalação. Caso contrário, não confie no conteúdo desse arquivo: apague-o e tente novamente. Note que, se você estiver instalando uma versão diferente da 1.2.4, o valor do md5 será diferente do deste exemplo. O importante é que ele seja exatamente igual ao que você viu na página oficial do GnuPG.

Agora, descompacte o arquivo baixado no diretório de sua preferência (por exemplo, o próprio C:\gnupg), usando sua ferramenta de descompactação preferida.

Para que o GnuPG funcione em português, entre no diretório onde você o descompactou e renomeie o arquivo "`pt_BR.mo`" para "`gnupg.mo`", pelo Windows Explorer ou diretamente na linha de comando, digitando:

```
C:\gnupg> ren pt_BR.mo gnupg.mo
```

Se desejar, apague os outros arquivos com extensão ".mo", já que eles não serão utilizados.

Finalmente, edite o arquivo "`gnupg-w32.reg`" que se encontra neste diretório (clique com o botão direito do mouse e escolha a opção "Editar"), e troque os caminhos de `HomeDir`, `gpgProgram` e `MODir` para o diretório correto (o que você escolheu para colocar o GnuPG). Por exemplo, se você instalou no diretório `C:\ferramentas\gpg\`, troque as linhas:

```
"HomeDir"="C:\\GnuPG"
"gpgProgram"="C:\\GnuPG\\gpg.exe"
"MODir"="C:\\GnuPG\\Locale"
```

por

```
"HomeDir"="C:\\ferramentas\\gpg"
"gpgProgram"="C:\\ferramentas\\gpg\\gpg.exe"
"MODir"="C:\\ferramentas\\gpg"
```

Observação: Repare que, ao contrário do padrão Windows, nesse arquivo devem ser usadas duas barras ao invés de uma para separar diretórios.

Finalmente, salve as modificações e execute o arquivo "`gnupg-w32.reg`" (duplo clique com o mouse no arquivo), para que essas modificações sejam adicionadas ao registro do Windows (você provavelmente precisará de privilégios de administrador para realizar essa última etapa).

2.2 – GNU/Linux

Pegue a versão mais recente do **GnuPG para GNU/Linux** na seção "*Download* " da página oficial do mesmo, e anote o valor md5 indicado ao lado do nome do arquivo na página oficial. Quando este documento foi desenvolvido, a versão mais recente do GnuPG era a 1.2.4, e portanto a página oficial do GnuPG mostrava os valores:

```
16d0b575812060328f8e677b7f0047cc  gnupg-1.2.4.tar.bz2
```

Uma vez obtido o GnuPG, use a ferramenta *md5sum* para verificar a integridade do arquivo. Isso garante que você está em posse de uma versão oficial que não foi modificada e, portanto, não deve possuir código malicioso:

Entre no diretório em que você colocou o arquivo .bz2 do GnuPG que acabou de baixar. Por exemplo, se ele está no diretório `\tmp\gnupg`, digite:

```
$ cd \tmp\gnupg
```

Em seguida, chame o programa md5sum seguido do nome do arquivo do GnuPG (no nosso exemplo, o arquivo chama-se "`gnupg-1.2.4.tar.bz2`")

```
$ md5sum gnupg-1.2.4.tar.bz2
```

A seguinte mensagem deve aparecer:

```
16d0b575812060328f8e677b7f0047cc  gnupg-1.2.4.tar.bz2
```

Como o valor md5 que aparece antes do nome do arquivo é igual àquele da página oficial do GnuPG, o arquivo está correto e podemos prosseguir com a instalação. Caso contrário, não confie no conteúdo desse arquivo: apague-o e tente novamente. Note que, se você estiver instalando uma versão diferente da 1.2.4, o valor do md5 será diferente do deste exemplo. O importante é que ele seja exatamente igual ao que você viu na página oficial do GnuPG.

Agora, descompacte o arquivo baixado no diretório de sua preferência (por exemplo, o próprio `\tmp\gnupg`), usando a ferramenta `tar`.

```
$ tar xvjf gnupg-1.2.4.tar.bz2
```

Nota: Caso tenha obtido a versão .tar.gz (ao invés da .tar.bz2), troque o parâmetro "j" por "z", no comando acima.

Um diretório chamado "`gnupg-x.y.z`" (onde "`x.y.z`" é o número da versão que você obteve) deve ter sido criado. Para compilar e instalar o programa, digite a seguinte seqüência de comandos (em nosso exemplo, estamos usando a versão 1.2.4. Troque esse número pela versão que estiver instalando):

```
$ cd gnupg-1.2.4
$ ./configure
$ make
$ /bin/su -c "make install"
```

O comando "`su`" acima executará a instalação no sistema, e pedirá a senha do usuário *root*. Se tudo correu bem, o GnuPG foi instalado com sucesso no diretório `/usr/local/bin`, e pode agora ser utilizado por qualquer usuário do sistema. Do contrário, verifique o arquivo README localizado no diretório recém-criado para solução de problemas. O script "configure" também pode ser chamado com parâmetros personalizados de configuração (mais detalhes na documentação oficial).

Para bloquear páginas de memória usadas pelo GnuPG e evitar que elas sejam escritas no disco rígido (operações de *swap* de memória feitas pelo sistema operacional), o que oferece um risco potencial às suas chaves privadas, convém colocar o GnuPG como `setuid root`. Embora o GnuPG abandone os privilégios de `root` assim que a memória é alocada, é importante notar também que programas com `setuid root` que apresentem falhas em seu código podem permitir que usuários maliciosos explorem tais vulnerabilidades e consigam elevar seus privilégios no sistema.

Para saber se o seu sistema operacional suporta bloqueio de memória sem precisar de privilégios de `root`, execute o comando "`gpg`" como usuário comum e verifique se a mensagem: "`gpg: WARNING: using insecure memory!`" **não** aparece. Do contrário, o bloqueio de memória só pode ser feito com `setuid root`. Pondere com cuidado sobre os prós e contras disso explicitados no parágrafo acima e tome sua decisão. Caso queira colocar o GnuPG como `setuid root`, digite o seguinte comando como usuário `root`:

```
# chmod 4755 /usr/local/bin/gpg
```

Caso prefira manter o GnuPG como está e continuar usando páginas de memória que podem ser escritas em disco rígido, pode ser interessante remover esse aviso, adicionando a linha "`no-secmem-warning`" no arquivo `~/.gnupg/gpg.conf`. Essa prática, no entanto, pode passar uma falsa sensação de segurança e a impressão de que o problema foi eliminado, quando na verdade o GnuPG continua usando memória insegura, apenas não avisa mais.

Finalmente, para que o GnuPG funcione em português, defina a variável de ambiente `LANG` com o valor `pt_BR`. Isso é feito de maneiras diferentes dependendo da *shell* que você usa (para saber qual a sua *shell* padrão, digite "`echo $SHELL`"):

```
% setenv LANG pt_BR                (csh)
$ export LANG; LANG=pt_BR          (sh)
$ export LANG=pt_BR                (bash, ksh)
```

Para não ter que fazer isso sempre que for usar o GnuPG, adicione o comando relativo a sua *shell* no seu arquivo *.login* ou *.profile*. através de um editor de textos de sua escolha ou do comando echo, por exemplo:

```
$ echo "export LANG=pt_BR" >> ~/.profile
```

2.3 – Outros

Além dos sistemas GNU/Linux e Windows, o GnuPG oferece pacotes pré-compilados que funcionam em diversas outras plataformas. A versão para Mac OS X deve ser obtida em *http://macgpg.sourceforge.net/*. Pacotes para sistemas operacionais que seguem o padrão POSIX (como o FreeBSD) podem ser encontrados em *http://gnupg.unixsecurity.com.br/*. Existe ainda código-fonte e binários pré-compilados para o RISC OS, em *http://www.sbellon.de/gnupg.html*. Entre nos endereços citados acima para saber mais detalhes sobre o processo de instalação do GnuPG em cada uma dessas plataformas. Não se esqueça de **sempre** verificar a assinatura *md5* do arquivo baixado, para garantir que está usando uma versão confiável.

3 – Criando chaves

O GnuPG utiliza um método de criptografia conhecido como chave assimétrica, em que duas chaves são criadas: a primeira, pública, serve para que qualquer um codifique mensagens e arquivos de modo que apenas você possa decodificar (de fato, uma vez codificada, nem mesmo o remetente pode reverter o processo); a segunda, privada, deve ser mantida em absoluto sigilo e serve para que você decodifique mensagens criptografadas com sua chave pública equivalente. Por isso, esse método também é conhecido como criptografia de chave pública.

Diversos servidores de chaves existem espalhados pela Internet para facilitar a distribuição de suas chaves públicas, dentre eles *http://pgp.mit.edu/* e *http://www.keyserver.net/*. Servidores de chaves são sincronizados entre si constantemente, então em geral basta submeter sua chave a apenas um deles, se desejar utilizá-los. Existem inúmeras formas de divulgar sua chave pública como veremos neste documento, e essa é apenas uma delas. Por tratar-se de um método particular, não entraremos em detalhes sobre como divulgar sua chave em servidores. Informações mais específicas podem ser encontradas nos endereços citados acima.

Para gerar um par de chaves no GnuPG, você precisará digitar o comando "`gpg --gen-key`". Note que, se o diretório onde o GnuPG foi instalado não estiver na sua variável de caminhos (PATH),

você precisará acessá-lo para que o comando tenha efeito. Caso o comando tenha sido bem sucedido, o GnuPG iniciará o processo de criação de nova chave, com a seguinte mensagem:

```
gpg (GnuPG) 1.2.4; 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)
   (4) RSA (apenas assinatura)
Sua opção?
```

Nesse ponto você deve escolher o tipo de algoritmo criptográfico que será usado em seu novo par de chaves. Como as opções 2 e 4 servem apenas para assinatura (As opções 3 e 5 são omitidas pelo programa pois não se aplicam a essa etapa), escolha a 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)
```

O GnuPG determina 1024 bits para o DSA, e cabe a você decidir quanto escolherá para o ElGamal. Por questões de segurança, o tamanho mínimo permitido é de 768 bits. Aqui, quanto maior o tamanho da chave menor a probabilidade de alguém conseguir descobrir seu valor por força bruta (o bom e velho tentativa-e-erro). O valor máximo permitido são 4096 bits, mas o máximo sugerido são 2048 bits, já que valores maiores levam muito tempo para serem computados. Para usuários comuns que não guardam segredos de Estado ou os números da próxima loteria acumulada, o padrão 1024 é suficiente, e foi escolhido em nosso exemplo. Faça sua escolha e siga para a próxima etapa.

```
O tamanho de chave pedido é 1024 bits
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)
```

Agora, escolha o período de validade da sua chave. Uma boa prática de segurança é trocar de chave – e conseqüentemente, de senha – a cada três meses. Isso porque, caso algo dê errado e sua chave privada seja capturada por terceiros, eles não terão tempo o suficiente para quebrá-la por força bruta mesmo com computadores dedicados, que só conseguiriam completar a tarefa quando o valor da chave não é mais o mesmo – afinal de contas, ela expirou e você criou uma nova.

Mais uma vez, se você deseja apenas manter sua privacidade e não é responsável por nenhum dado sigiloso como informações internas de sua empresa, é mais cômodo – embora não tão seguro –

desativar a validade de seu par de chaves. Como mostra a sintaxe, escreva um número qualquer seguido de nada para dias, "w" para semanas, "m" para meses ou "y" para anos, ou zero, para que a chave não expire. Se quiser que sua chave seja válida pelos, digamos, próximos seis meses, digite "6m". Em nosso exemplo, digitamos "0", para que a chave não expire. Feito isso, uma confirmação de sua escolha aparecerá. Caso esteja certo de sua decisão, confirme. Senão, repita esse passo.

```
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: João Ninguém
Endereço de correio eletrônico: jninguem@exemplo.com.br
Comentário: gerente
Você está usando o conjunto de caracteres `iso-8859-1'.
Você selecionou este identificador de usuário:
    "João Ninguém (gerente) <jninguem@exemplo.com.br>"
Muda (N)ome, (C)omentário, (E)ndereço ou (O)k/(S)air?
```

Chegou a hora de criar seu identificador de usuário, a partir de algumas informações suas. Isso é feito para que o GnuPG crie uma etiqueta para seu par de chaves, fazendo com que ele seja facilmente distinguível por você no meio de outras chaves (sim, você pode usar o mesmo GnuPG para manipular tantas chaves quanto quiser). Portanto, preencha o questionário que aparecerá na tela. Você precisará incluir seu nome completo, endereço de correio eletrônico e algum comentário opcional. O nome real precisa ter pelo menos cinco caracteres, e se não quiser colocar comentários basta deixar o campo "comentário" em branco. Não se preocupe se errar algum dado, pois o GnuPG pedirá confirmação e permitirá que você modifique quaisquer campos antes de criar o identificador, como mostra nosso exemplo acima. Verifique se seus dados estão corretos e prossiga para a próxima etapa.

```
Você precisa de uma frase secreta para proteger sua chave.

Digite a frase secreta:
```

Chegamos à fase final do processo de criação de chaves, onde você deve digitar uma senha para sua chave. Essa proteção extra auxiliará a manter secreta sua chave privada, e pode incluir espaços, o que significa que você pode colocar até uma pequena frase como senha. Note que os caracteres digitados não aparecem, nem mesmo asteriscos, para evitar que pessoas saibam quantos caracteres sua senha tem. Evite, no entanto, escolher uma senha muito grande (por exemplo, mais de 30 caracteres), já que você precisará digitá-la sempre que precisar decodificar ou assinar um arquivo. Escolha com muito cuidado uma senha que não seja fácil de adivinhar, como palavras no dicionário e dados facilmente verificáveis (nome do cônjuge, data de aniversário dos filhos, CPF, etc.). Feito isso, ele pedirá confirmação na mesma linha, exibindo a mensagem:

```
Repita a frase secreta:
```

Digite novamente a mesma senha e pronto, seu par de chaves está prestes a ser criado. Nessa próxima tela, o GnuPG gerará números primos que serão usados na criação de sua chave. Para

garantir que os números aleatórios usados nessa etapa serão dificilmente reproduzíveis, faça como indicado e realize outras tarefas enquanto ele estiver trabalhando, como movimentar o mouse ou abrir arquivos.

```
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.
+++++.++++++++++.+++++.++++++++++++++.++++++++++++++++++++++++.++++++++++++
++++++++..+++++++++++++++++++++++++..+++++.+++++..+++++.++++++++++++++.....
..+++++
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/D6B70989 2004-01-25 João Ninguém (gerente) <jninguem@exemplo.com.br>
    Impressão da chave = 0A0F 1A7F AD0A 8850 4281  4213 55A5 D604 0C52 CCFA
sub 1024g/37A4B0DB 2004-01-25
```

Nota: No Windows, caso um erro tenha ocorrido durante a etapa final de criação de chaves, remova o arquivo "`gnupg.mo`" e tente novamente. O GnuPG ficará em inglês, mas deve funcionar.

Ao final do processo, três importantes arquivos foram criados: "`pubring.gpg`", o arquivo-chaveiro onde ficarão as chaves públicas de todos aqueles com quem você queira se comunicar via GnuPG, junto com sua própria chave pública, que deve ser passada a todos que queiram comunicar-se com você; "`secring.gpg`", o arquivo-chaveiro contendo sua(s) chave(s) privada(s); e "`trustdb.gpg`", com informações sobre o nível de confiabilidade de cada chave pública armazenada.

Convém criar uma cópia de segurança de seus arquivos-chaveiro, já que de nada adianta lembrar-se de sua senha quando um problema qualquer no computador apagou sua chave privada. Para isso, basta copiar os arquivos `pubring.gpg`, `secring.gpg` e `trustdb.gpg` citados acima para uma mídia segura (como um *CD-R*, por exemplo), e guarde-a com cuidado para que ninguém além de você tenha acesso a ela.

Embora não faça muito sentido ter mais de uma chave privada, é sempre possível executar novamente o comando "`gpg --gen-key`" para fazê-lo. Note, no entanto, que caso isso seja feito, talvez seja necessário adicionar o parâmetro "`--local-user`" (ou simplesmente "`-u`") seguido de qualquer campo do identificador de usuário (como o endereço de correio eletrônico) para que o GnuPG direcione seus comandos à chave certa, em todos os comandos em que é feito uso da chave privada.

3.1 – Criando um certificado de revogação

Após a criação de uma chave privada, é importantíssimo que seja criado um certificado de revogação para a mesma. Esse certificado permitirá a você anular sua chave caso esqueça a senha ou alguém consiga descobri-la. Para isso, digite o comando "`gpg --gen-revoke id`", onde *id* é o número de identificação ou qualquer campo do identificador de usuário (como o endereço de correio eletrônico) da chave desejada. O GnuPG vai fazer algumas perguntas e gerar então o certificado, exibindo-o na tela, para que você anote seu valor. Caso prefira que ele coloque o certificado em um arquivo de texto, redirecione a saída do GnuPG para um arquivo qualquer, adicionando "`--output`" (ou "`-o`") ao comando acima, que fica "`gpg --output nomedoarquivo --gen-revoke id`".

Em nosso exemplo, digitamos o comando "`gpg --gen-revoke jninguem`" (como em nossa lista não há ninguém com nome de conexão – o nome que vem antes do símbolo de "@" no endereço de correio eletrônico – igual ao nosso, não precisamos digitar o resto do email). O GnuPG exibirá detalhes sobre a chave privada do usuário, como mostrado abaixo:

```
sec  1024D/D6B70989 2004-01-25   João Ninguém (gerente) <jninguem@exemplo.com.br>

      Create a revocation certificate for this key? y

      Please select the reason for the revocation:
        0 = Nenhum motivo especificado
        1 = A chave foi comprometida
        2 = A chave foi substituída
        3 = A chave já não é utilizada
        Q = Cancel
       (Probably you want to select 1 here)
      Sua decisão? 0
```

Você precisa especificar um motivo para o certificado ter sido criado, ou digitar "*Q*" para cancelar toda a operação. Como acabamos de criar a chave e não sabemos qual seria o motivo de uma possível futura revogação, escolhemos a opção "*0*". Ele pedirá que você coloque uma descrição opcional do anulamento, que pode ter quantas linhas desejar (tenha em mente, no entanto, que o tamanho do certificado cresce de acordo com o tamanho da descrição opcional), e deve ser terminado com uma linha em branco. Em nosso exemplo, como o certificado está sendo criado antecipadamente, não colocamos descrição alguma.

```
      Enter an optional description; end it with an empty line:
      >
      Reason for revocation: Nenhum motivo especificado
      (No description given)
      Is this okay? y

      Você precisa de uma frase secreta para desbloquear a chave secreta do
      usuário: "João Ninguém (gerente) <jninguem@exemplo.com.br>"
      chave de 1024-bit/DSA, ID D6B70989, criada em 2004-01-25

      Digite a frase secreta:
```

Naturalmente, você só pode criar um certificado de revogação para uma conta a qual possui a senha. Isso evita que outras pessoas criem falsos certificados em seu nome e cancelem suas chaves.

```
ASCII armored output forced.
Revocation certificate criada.

Please move it to a medium which you can hide away; if Mallory gets
access to this certificate he can use it to make your key unusable.
It is smart to print this certificate and store it away, just in case
your media become unreadable.  But have some caution:  The print system of
your machine might store the data and make it available to others!
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.2.4 (GNU/Linux)
Comment: A revocation certificate should follow

iEkEIBECAAkFAkAgJRQCHQAACgkQ4eh/uNFtIsb/AACdExQURD4q2XADi7gnl8TB
2NGJekcAoLZQ5Z6GYsclPmyU3V52Op+zEw2j
=ww2x
-----END PGP PUBLIC KEY BLOCK-----
```

Seu certificado é tudo que está entre as linhas tracejadas (começando com "`-----`"), e deve sempre incluir essas linhas delimitadoras.

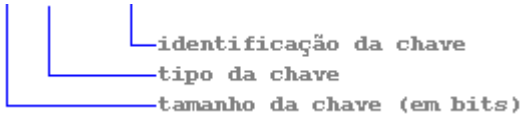
O mesmo procedimento poderia ter sido feito em nosso exemplo redirecionando a saída do GnuPG para um arquivo chamado, digamos, *anulachave.asc*, caso tivéssemos digitado: "`gpg --output anulachave.asc --revoke-key jninguem`". Terminado o procedimento acima, o arquivo de texto conteria o certificado de revogação, que pode então ser copiado para uma mídia segura ou impresso. O certificado é relativamente pequeno, então é recomendado que ele seja impresso e guardado **em local seguro**. Do contrário, qualquer um com acesso ao seu certificado poderá publicá-lo e, assim, anular a validade de sua chave pública. Vale lembrar aos mais precavidos que, ao imprimir o certificado, a impressora pode armazenar temporariamente o mesmo em uma área não protegida do computador, que pode ser monitorada por um atacante com acesso ao seu sistema.

Veja o final da seção seguinte para saber como revogar uma chave.

4 – Manipulando suas chaves

Para listar as chaves armazenadas, digite o comando "`gpg --list-keys`". Em nosso exemplo, a seguinte resposta apareceu:

```
------------------------
pub  1024D/D6B70989 2004-01-25 João Ninguém (gerente) <jninguem@exemplo.com.br>
sub  1024g/37A4B0DB 2004-01-25
```



Antes de mais nada, vamos entender o que acabou de ser mostrado. Apenas as chaves públicas são mostradas. A primeira coluna indica o tipo de chave: a palavra `pub` indica uma chave pública principal para assinaturas, enquanto a palavra `sub` indica uma chave pública subordinada. A segunda coluna indica a quantidade de bits da chave, seguido por seu tipo e seu número de identificação em hexadecimal. O código do tipo varia de acordo com a seguinte tabela:

código	significado
D	chave DSA
g	chave ElGamal somente para codificar
G	chave ElGamal para codificar e assinar
R	chave RSA

A terceira coluna nos mostra a data de criação da chave, na forma *ano-mês-dia*. Por fim, a última coluna exibe o identificador do usuário.

Essa lista pode ficar muito grande à medida que você vai adicionando chaves públicas de pessoas com quem deseja comunicar-se de forma segura. Para ver os detalhes de apenas uma chave, entre com o comando "`gpg --list-key id`", onde *id* é o número de identificação ou qualquer campo do identificador de usuário (como o endereço de correio eletrônico) da chave que deseja visualizar.

Para saber informações mais específicas sobre as chaves de um usuário (e editá-las, caso possua a senha), digite o comando "`gpg --edit-key id`". Em nosso exemplo, digitamos "`gpg --edit-key jninguem`". O GnuPG nos retornou o seguinte:

```
Chave secreta disponível.

pub  1024D/D6B70989  criada: 2004-01-25 expira: never      confiança: u/u
sub  1024g/37A4B0DB  criada: 2004-01-25 expira: never
(1). João Ninguém (gerente) <jninguem@exemplo.com.br>

Comando>
```

A primeira linha nos diz que a chave privada desse usuário está disponível, de modo que mensagens criptografadas com a chave pública do usuário "*João Ninguém*" podem ser decodificadas usando esse arquivo-chaveiro (se a senha for fornecida, naturalmente). As demais linhas estão divididas em colunas, como explicado acima, só que aqui os identificadores de usuários aparecem apenas depois da listagem de chaves. Com o comando "`--edit-key`", o GnuPG entra em modo de edição, e portanto a última linha espera que você digite um comando qualquer do GnuPG. Como a chave secreta está disponível para esse id, podemos digitar o comando "`toggle`" para ver os componentes privados do par de chaves:

```
Comando> toggle

sec  1024D/D6B70989  criada: 2004-01-25 expira: never
ssb  1024g/37A4B0DB  criada: 2004-01-25 expira: never
(1)  João Ninguém (gerente) <jninguem@exemplo.com.br>

Comando>
```

As linhas exibidas estão divididas em colunas de modo similar ao explicado acima, só que agora a primeira coluna exibe valores diferentes, indicando dois novos tipos de chave: `sec` para a chave privada principal para assinaturas, e `ssb` para chaves privadas subordinadas. Para retornar à visualização dos componentes públicos, basta digitar o comando "`toggle`" novamente. Sempre que desejar ver essas informações (com ou sem o comando "`toggle`"), basta digitar o comando "list", ou simplesmente apertar a tecla `[Enter]`.

Modificações feitas em modo de edição somente serão gravadas após a finalização do programa pelo usuário. Isso é feito através dos comandos "`quit`", que pergunta ao usuário se ele deseja ou não gravar suas alterações antes de sair do programa, ou "`save`", que grava todas as modificações feitas e finaliza o programa.

Podemos executar também o comando "`check`", para verificar a integridade das chaves:

```
Comando> check
uid  João Ninguém (gerente) <jninguem@exemplo.com.br>
sig!       D6B70989 2004-01-25   [auto-assinatura]

Comando>
```

O comando "`check`" lista as assinaturas de cada par público armazenado. Como pode ser observado, a auto-assinatura da chave em nosso exemplo (exibida na linha "`sig!`" acima) é a chave principal para assinaturas "`D6B70989`". Isso significa que essa chave está ligada diretamente à chave pública associada e, portanto, pode ser considerada válida. Mais informações sobre assinaturas digitais na seção 5 deste documento.

Como foi visto, seu par de chaves contém uma chave principal para assinaturas e uma chave subordinada. O mesmo usuário pode ter tantas chaves subordinadas e tantos identificadores de usuário quanto achar conveniente. Diferentes identificadores de usuário são especialmente úteis quando você precisa de mais de uma identidade como, por exemplo, uma para o trabalho e outra para assuntos pessoais. Chaves subordinadas adicionais também são úteis uma vez que podem ser usadas somente para codificar e, se danificadas, podem ser substituídas sem a necessidade de recertificação (já que a chave principal não foi comprometida).

Para adicionar um novo identificador de usuário ligado ao mesmo par público/privado de chaves, execute o comando "`adduid`" quando estiver editando o usuário desejado (através do comando "`gpg --edit-key id`" visto acima). Em nosso exemplo, vamos adicionar uma nova *uid* para o usuário João Ninguém:

```
Comando> adduid
Nome completo: José da Silva
Endereço de correio eletrônico: zesilva@exemplo.com.br
Comentário:
Você está usando o conjunto de caracteres `iso-8859-1'.
Você selecionou este identificador de usuário:
    "José da Silva <zesilva@exemplo.com.br>"

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

```
Você precisa de uma frase secreta para desbloquear a chave secreta do
usuário: "João Ninguém (gerente) <jninguem@exemplo.com.br>"
```

Nesse ponto você precisará digitar a senha do usuário para o qual deseja criar o identificador adicional. Isso garante que ninguém criará novos identificadores para um usuário sem a permissão do mesmo.

```
chave de 1024-bit/DSA, ID D6B70989, criada em 2004-02-03

pub  1024D/D6B70989  criada: 2004-01-25 expira: never      confiança: u/u
sub  1024g/37A4B0DB  criada: 2004-01-25 expira: never
(1). João Ninguém (gerente) <jninguem@exemplo.com.br>
(2)  José da Silva <zesilva@exemplo.com.br>

Comando>
```

O novo identificador para o usuário foi criado e exibido pelo GnuPG. Note que ambos estão vinculados à mesma chave principal e, portanto, ou são a mesma pessoa ou dividem o mesmo arquivo-chaveiro, o que raramente é uma prática recomendada (para criar novos usuários, cada qual com sua chave privada, digite "`gpg --gen-key`", como visto na seção 3). Para criar uma nova chave subordinada à chave principal do usuário selecionado, quer para criptografar ou para assinar arquivos, basta executar o comando "`addkey`". Repare que, assim como o "`adduid`", as perguntas feitas pelo GnuPG são muito similares àquelas de quando seu usuário foi criado. Você vai perceber, no entanto, que agora as opções de tipo de chave são diferentes.

```
Comando> addkey
Key is protected.

Você precisa de uma frase secreta para desbloquear a chave secreta do
usuário: "João Ninguém (gerente) <jninguem@exemplo.com.br>"
chave de 1024-bit/DSA, ID D6B70989, criada em 2004-01-25

Por favor selecione o tipo de chave desejado:
   (2) DSA (apenas assinatura)
   (3) ElGamal (apenas criptografia)
   (4) RSA (apenas assinatura)
   (5) RSA (apenas criptografia)
Sua opção? 5
Que tamanho de chave você quer? (1024)
O tamanho de chave pedido é 1024 bits
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)
Key does not expire at all
Está correto (s/n)? s
Realmente criar? s
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.
+++++
.....+++++

pub  1024D/D6B70989  criada: 2004-01-25 expira: never      confiança: u/u
sub  1024g/37A4B0DB  criada: 2004-01-25 expira: never
sub  1024R/798B9609  criada: 2004-02-06 expira: never
(1). João Ninguém (gerente) <jninguem@exemplo.com.br>
(2)  José da Silva <zesilva@exemplo.com.br>

Comando>
```

O GnuPG oferece uma série de comandos para edição de chaves e usuários, e muitas vezes precisamos indicar exatamente qual subchave (chave secundária) ou identificador de usuário queremos manipular. Isso é feito através dos comandos "`key`" e "`uid`", que marcam e desmarcam subchaves e identificadores de usuários, respectivamente, bastando digitar o comando seguido pelo índice da subchave ou do identificador. Em nosso exemplo, para marcar a chave secundária que acabamos de criar, digitamos o comando abaixo:

```
Comando> key 2

pub  1024D/D6B70989  criada: 2004-01-25 expira: never      confiança: u/u
sub  1024g/37A4B0DB  criada: 2004-01-25 expira: never
sub* 1024R/798B9609  criada: 2004-02-06 expira: never
(1). João Ninguém (gerente) <jninguem@exemplo.com.br>
(2)  José da Silva <zesilva@exemplo.com.br>
```

Repare que um asterisco ("*****") apareceu ao lado da palavra "sub" da chave secundária escolhida, marcando-a. Para desmarcar, basta repetir o comando. Note ainda que o comando "`key`" marca e desmarca apenas as chaves secundárias, uma vez que cada usuário pode ter apenas uma chave de decodificação (a chave principal).

Para apagar uma chave secundária, marque-a como demonstrado no exemplo acima e, em seguida, digite o comando "`delkey`":

```
Comando> delkey
Você realmente quer remover esta chave? s

pub  1024D/D6B70989  criada: 2004-01-25 expira: never      confiança: u/u
sub  1024g/37A4B0DB  criada: 2004-01-25 expira: never
(1). João Ninguém (gerente) <jninguem@exemplo.com.br>
(2)  José da Silva <zesilva@exemplo.com.br>
```

Apenas chaves secundárias podem ser apagadas. Caso queira desativar a chave principal, basta digitar o comando "`disable`".

```
Comando> disable
```

Ao listar as chaves do usuário novamente (pelo comando "list" ou apertando a tecla [Enter]), a seguinte mensagem será exibida:

```
Comando> list

pub  1024D/D6B70989  criada: 2004-01-25 expira: never      confiança: u/d
*** Esta chave foi desativada
sub  1024g/37A4B0DB  criada: 2004-01-25 expira: never
(1). João Ninguém (gerente) <jninguem@exemplo.com.br>
(2)  José da Silva <zesilva@exemplo.com.br>
```

Repare que, na coluna "confiança", a letra "d" indica que a chave está desativada. Um aviso do GnuPG também é exibido nessa situação. Atente, no entanto, para o fato de que as modificações só entram em efeito após a finalização do programa (comandos "quit" ou "save", vistos anteriormente). Para reativar a chave, basta digitar o comando "enable".

Analogamente ao comando "key", o comando "uid" marca o identificador de usuário indicado pelo índice:

```
Comando> uid 2

pub  1024D/D6B70989  criada: 2004-01-25 expira: never      confiança: u/u
sub  1024g/37A4B0DB  criada: 2004-01-25 expira: never
(1). João Ninguém (gerente) <jninguem@exemplo.com.br>
(2)* José da Silva <zesilva@exemplo.com.br>
```

Ao contrário do comando "key", qualquer identificador de usuário pode ser selecionado.

Para definir um identificador como principal (necessário quando há mais de um identificador de usuário), selecione um identificador e entre com o comando "primary". Após digitar a senha, a nova associação será feita e um pequeno ponto (".") na listagem indicará o novo usuário principal. Quando o identificador principal está marcado, o pequeno ponto é sobreposto pelo asterisco que indica a seleção. Para verificar se o ponto está lá, basta entrar com o comando "uid" e desmarcá-lo, como demonstrado no exemplo abaixo, em que trocamos o identificador principal de "João Ninguém" para "José da Silva". O uid 2 foi marcado no exemplo acima, e abaixo o definimos como primário e, em seguida, desmarcamos o uid e verificamos a listagem atualizada:

```
Comando> primary
Comando> uid 2
Comando> list

pub  1024D/D6B70989  criada: 2004-01-25 expira: never      confiança: u/u
sub  1024g/37A4B0DB  criada: 2004-01-25 expira: never
(1)  João Ninguém (gerente) <jninguem@exemplo.com.br>
(2). José da Silva <zesilva@exemplo.com.br>

Comando>
```

Para apagar um usuário, marque-o como mostrado acima e digite o comando "deluid":

```
Comando> uid 2
Comando> deluid
Realmente remover este ID de usuário? s

pub  1024D/D6B70989  criada: 2004-01-25 expira: never      confiança: u/u
sub  1024g/37A4B0DB  criada: 2004-01-25 expira: never
(1)  João Ninguém (gerente) <jninguem@exemplo.com.br>

Comando>
```

Duas importantes questões precisam ser notadas: ao remover um usuário marcado como principal – como fizemos no exemplo acima – é preciso definir outro como principal, a menos que reste apenas um identificador de usuário (ainda assim, a definição deste como usuário principal é uma boa prática). A segunda questão é mais óbvia: não é possível ficar sem identificadores, pois cada par de chaves precisa estar associado a pelo menos um identificador de usuário.

É possível ainda modificar a data de validade de uma chave. Isso é feito através do comando "`expire`". Quando nenhuma chave secundária está selecionada (comando "key" visto anteriormente), o comando modifica a expiração da chave primária. Do contrário, é considerada a chave selecionada. Vejamos o que acontece ao modificarmos a data de validade da chave primária de nosso exemplo:

```
Comando> expire
Modificando a data de validade para uma chave primária.
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) 2y
```

Acima, definimos que a chave terá validade de dois anos, contados à partir deste momento.

```
Chave expira em Qui 26 Jan 2006 22:28:41 BRST
Está correto (s/n)? s

Você precisa de uma frase secreta para desbloquear a chave secreta do
usuário: "João Ninguém (gerente) <jninguem@exemplo.com.br>"
chave de 1024-bit/DSA, ID D6B70989, criada em 2004-01-25

pub  1024D/D6B70989  criada: 2004-01-25 expira: 2006-01-26 confiança: u/u
sub  1024g/37A4B0DB  criada: 2004-01-25 expira: never
(1). João Ninguém (gerente) <jninguem@exemplo.com.br>

Comando>
```

Note que a data de expiração da chave secundária não foi afetada.

Vejamos agora o que deve ser feito para anular a validade de suas chaves. Uma vez criado um certificado de revogação para a mesma, como visto na seção 3.1, basta importá-lo como faria com uma chave normal (veja seção 3.3) e distribuir novamente sua chave pública nos servidores de chaves, ou

simplesmente publicar seu certificado de revogação de modo que os outros o importem em seus arquivos-chaveiro, anulando sua chave da mesma forma. Isso indicará a todos que esta chave não deve ser mais usada. Uma vez feita a importação, ao editarmos a chave veremos que o campo "confiança" está indicando que a chave foi anulada:

```
pub  1024D/D6B70989  criada: 2004-01-25 expira: 2006-01-26 confiança: -/r
sub  1024g/37A4B0DB  criada: 2004-01-25 expira: never
(1). João Ninguém (gerente) <jninguem@exemplo.com.br>
```

Note que o certificado é aplicado a sua chave pública, então você não precisa da chave privada para usá-lo. Por isso, é muito importante que o mesmo seja guardado em local seguro até que seja necessário (esperamos que nunca), do contrário qualquer um de posse do mesmo poderá anular sua chave.

Caso você ainda tenha acesso a sua chave privada, é possível revogar chaves secundárias, identificadores de usuário e até mesmo assinaturas a qualquer momento. Para isso, você precisa acessar seu par de chaves em modo de edição pelo comando "`gpg --edit-key id`" visto anteriormente. Atente para o fato de que, ao contrário do comando "`gpg --gen-revoke`", nesse caso o anulamento é feito imediatamente. Portanto, pense com cuidado antes de utilizar tais comandos.

Para revogar uma chave secundária, selecione-a e em seguida digite o comando "`revkey`":

```
pub  1024D/D6B70989  criada: 2004-01-25 expira: 2006-01-26 confiança: u/u
sub  1024g/37A4B0DB  criada: 2004-01-25 expira: never
(1). João Ninguém <jninguem@exemplo.com.br>

Comando> key 1

pub  1024D/D6B70989  criada: 2004-01-25 expira: 2006-01-26 confiança: u/u
sub* 1024g/37A4B0DB  criada: 2004-01-25 expira: never
(1). João Ninguém <jninguem@exemplo.com.br>

Comando> revkey
Você realmente quer revogar esta chave? s
Please select the reason for the revocation:
  0 = Nenhum motivo especificado
  1 = A chave foi comprometida
  2 = A chave foi substituída
  3 = A chave já não é utilizada
  Q = Cancel
Sua decisão? 3
Enter an optional description; end it with an empty line:
> Não quero mais usar chaves menores do que 2048 bits
>
Reason for revocation: A chave já não é utilizada
Não quero mais usar chaves menores do que 2048 bits
Is this okay? y

Você precisa de uma frase secreta para desbloquear a chave secreta do
usuário: "João Ninguém <jninguem@exemplo.com.br>"
chave de 1024-bit/DSA, ID D6B70989, criada em 2004-01-25
```

```
pub  1024D/D6B70989  criada: 2004-01-25 expira: 2006-01-26 confiança: u/u
sub  1024g/37A4B0DB  criada: 2004-01-25 expira: never
rev! subchave foi revogada: 2004-01-28
(1). João Ninguém <jninguem@exemplo.com.br>
```

Na listagem de suas chaves, uma linha de aviso abaixo da chave secundária anulada indicará a situação da mesma, incluindo a data em que o procedimento foi tomado.

Revogar identificadores de usuário pode parecer sem sentido, mas é necessário quando quaisquer dados incluídos no mesmo são modificados, como o comentário ou o endereço de correio eletrônico. Para revogar um identificador de usuário, selecione-o e entre com o comando "`revuid`":

```
Comando> uid 1

pub  1024D/D6B70989  criada: 2004-01-25 expira: 2006-01-26 confiança: u/u
sub  1024g/37A4B0DB  criada: 2004-01-25 expira: never
rev! subchave foi revogada: 2004-01-28
(1)* João Ninguém <jninguem@exemplo.com.br>

Comando> revuid
Realmente revocar este ID de utilizador? s
Please select the reason for the revocation:
  0 = Nenhum motivo especificado
  4 = O identificador do utilizador já não é válido
  Q = Cancel
(Probably you want to select 4 here)
Sua decisão? 4
Enter an optional description; end it with an empty line:
> não sou mais gerente.
>
Reason for revocation: O identificador do utilizador já não é válido
não sou mais gerente.
Is this okay? y

Você precisa de uma frase secreta para desbloquear a chave secreta do
usuário: "João Ninguém (gerente) <jninguem@exemplo.com.br>"
chave de 1024-bit/DSA, ID D6B70989, criada em 2004-01-25

pub  1024D/D6B70989  criada: 2004-01-25 expira: 2006-01-26 confiança: u/u
sub  1024g/37A4B0DB  criada: 2004-01-25 expira: never
rev! subchave foi revogada: 2004-01-28
(1). [revoked] João Ninguém (gerente) <jninguem@exemplo.com.br>
```

Um aviso na linha do identificador de usuário indica que houve a revogação. Repare que, em ambos os casos acima, a chave principal em si não foi anulada, e tanto sua parte pública quanto privada podem ser utilizadas normalmente. Note ainda que as chaves secundárias e identificadores de usuários não foram apagados, apenas marcados como inválidos. A atualização de seu arquivo-chaveiro dessa forma (revogando chaves ao invés de apagando-as) permite que você continue a decodificar mensagens codificadas com sua chave pública antes da revogação, mas ao mesmo tempo avisa os demais sobre a nova situação, de modo que aos poucos todos estarão usando sua nova chave.

Apagar chaves secundárias e identificadores de usuários implica em não poder validar (e, portanto, decodificar) arquivos codificados ou assinados com as mesmas.

É possível ainda revogar assinaturas, prático quando se perde a confiança em uma chave pública previamente assinada. Ao digitar o comando "`revsig`", o GnuPG irá circular por todas as chaves assinadas pelo usuário (incluindo a própria chave) para que o mesmo indique quais deseja revogar.

Manter seus arquivos-chaveiro atualizados após importações e exportações (vistas na próxima seção) também é importante. Para apagar chaves públicas de seu arquivo-chaveiro, digite o comando "`gpg --delete-key id`", onde *id* é o número de identificação ou qualquer campo do identificador de usuário (como o endereço de correio eletrônico) da chave desejada. Caso o usuário definido possua chave privada nessa conta, o GnuPG exibirá um aviso e pedirá que você remova antes a chave privada.

Para apagar chaves privadas de seu arquivo-chaveiro, digite "`gpg --delete-secret-key id`". Atenção ao fazê-lo, pois uma vez removida a chave privada, o usuário desta não poderá decodificar nenhum arquivo codificado com a chave pública equivalente, a menos que possua uma cópia de reserva da chave privada (veja acima como exportar chaves privadas) em outro lugar.

5 – Exportando e importando chaves

Talvez um dos aspectos mais importantes do GnuPG seja a troca de chaves entre pessoas. Para que outros possam criptografar mensagens para você, é necessário que tenham acesso a sua chave pública. Esse procedimento é conhecido como *exportar* sua chave pública. Uma vez em posse dessa, o destinatário *importa* a chave para dentro de seu arquivo-chaveiro. Idealmente, cada chave deveria ser entregue ao destinatário em mãos, mas na maioria dos casos ela é enviada por correio eletrônico, colocada em servidores de chaves ou publicada em páginas da Internet.

Para exportar sua chave pública em modo binário e gravá-la em um arquivo compreendido pelo GnuPG, entre com o comando "`gpg --output minhachave.gpg --export id`", onde *id* é o número de identificação ou qualquer campo do identificador de usuário (como o endereço de correio eletrônico) da chave desejada, e *minhachave.gpg* é o nome do arquivo em que deseja gravar sua chave pública. Para exportar a chave de nosso usuário-exemplo para o arquivo `joaozinho.gpg`, digitamos:

"**`gpg --output joaozinho.gpg --export jninguem`**"

O GnuPG criará o arquivo `joaozinho.gpg` contendo a chave. Esse arquivo pode ser então copiado e distribuído para todas as pessoas com quem deseja se comunicar de forma segura. Na maioria dos casos, no entanto, a exportação da chave em modo binário pode ser inconveniente, especialmente quando o usuário deseja publicar sua chave via página na Internet ou correio eletrônico. Para contornar esse problema, o GnuPG oferece a opção "`--armor`" (ou simplesmente "`-a`"), que exibe a chave em modo texto (ASCII), para que seja copiada e publicada facilmente em qualquer documento de texto, e ainda assim corretamente interpretada pelo GnuPG como uma chave. Para exibir sua chave em modo texto diretamente na tela do terminal, digite o comando "`gpg --armor --export id`". Em nosso exemplo, vamos exportar a chave do usuário "João Ninguém":

```
gpg --armor --export jninguem

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.2.4 (GNU/Linux)

mQGiBEAe68MRBACrzQGMPRT7bbk3Voa5zYlqnTzHMcyyjsdnok7gYnO6VIFo9NyD
UdwOfdyixEKInc+MYe8kgyttRRe+3xNZ+NJlzxAI5VZ7OdnYBhzUaAGv0UMfnCei
ywEjokABVHr0YY3Pvfn7rbMLymNnPJaVatioiLA7WT3yRYDhXliECNDr6wCg242M
g1u0s8MW+s1Pgh4m8JoktX8EAIz2L3J0fZJtlbW2tp327p9LmGqNYQnfC8PnKWMv
MVn3h2jbhrryCFCao5uks+D/BKxaRGeOx7Xb4JSNElbCxWhupwBVcBVgzpyuugZt
7bkOfPqe1akftEWBYhRsH1pTQ5rywtwovl1EkDKGd3FCB6ZysObPHvNpgHAj0+2o
LMLrA/9dEVsnkY1wpaBd70aC0cNKYC2oORIhaOs8/8dTvVoHPBSmjcWFQGldCYCl
WaVUn3/fLOxISrfFYMGFa+Pz82RNfHB9zixv4U0NJYXx4QIEY1lgGIBzJShNcCT1
MpRLlSwj/QC2qDz8fzEH5Hv4N4tFz28d6zs/P6fpalp6adKXr7QySm/Dhm8gTmlu
Z3XCgm0gKGdlcmVudGUpIDxqbmluZ3VlbUBleGVtcGxvLmNvbS5icj6IXgQTEQIA
HgUCQCgneQIbAwYLCQgHAwIDFQIDAxYCAQIeAQIXgAAKCRBrRxzG1rcJiRDkAKDB
LtjSozlq6Qa8UbIlfpVeUsZogACgteLaLjtHpQCAHynMblCv+lIybPK5AQ0EQB7r
xRAEANypAyqWusrhwxnTWSaHFKXkShyi65xe/E9kP8KBMmKd4N9i5Z2oZVbGxDYK
Aoxx4DFwJ9Y63acQVAdVi6arf62yZ3E0RG9DIM2X4aPuNOacv5F0SuYGPp1Dv4GQ
9KTzoPSZx+iXtL4E40tTV4RUQOp9jpnvEzfbdUIQnGU5xQkHAAMHBADCks+FFPly
CfFvPi92nBQb4S8LUsJDBp/WYQepvBUJZXwCEr25p+p22lwUnk5++bETFocELYS7
ariCCMvAcEQLogZdT51yw+jZrY/xEXziovYCS1wetjwkQfDuS0g7BYRuWvQYS2lH
jtRbtD4PkJUREA+zkVRQVNRwT9X+cwJDvYhJBBgRAgAJBQJAHuvFAhsMAAoJEGtH
HMbWtwmJIGMAoLj8QqKQfkH3YIcNuMNQ1muXStepAJ4rZWthNSJFAjpM4SKlXieZ
+3jzYQ==
=ujlP
-----END PGP PUBLIC KEY BLOCK-----
```

A chave pública é tudo que está entre as linhas tracejadas (começando com "-----"), e deve sempre incluir essas linhas delimitadoras ao ser reproduzida. Agora que a chave foi exportada, você pode copiá-la e colá-la em uma mensagem de correio eletrônico, publicar na Internet, etc. É sempre possível vê-la novamente repetindo o comando no GnuPG, mas caso prefira guardá-la em arquivo, incremente o comando com o parâmetro "--output" visto acima, que redireciona a saída do GnuPG para um arquivo de sua escolha. O comando ficaria, portanto:
"`gpg --output arquivo.txt --armor --export id`".

Tome cuidado com essa prática, porém, pois arquivos de texto podem ser alterados por usuários maliciosos. De fato, mesmo a publicação de sua chave pública via Internet, correio eletrônico ou qualquer meio intermediário de baixa confiança não oferece segurança, já que estão sujeitas a ataques similares. Veja o exemplo abaixo para um maior esclarecimento do problema:

Suponha que Alice quer se comunicar de forma segura com Bruno, mas Claudia quer saber tudo que Alice diz. Bruno coloca sua chave pública em um arquivo (ou página na Internet, ou mensagem de correio eletrônico), só que Claudia consegue acesso ao arquivo e troca a chave pública de Bruno por sua própria chave pública ***antes*** *que Alice consiga pegá-la. Alice finalmente pega a chave que acredita ser de Bruno, mas está pegando na verdade a chave de Claudia. Alice usa essa chave para codificar todas as mensagens privadas que envia para Bruno. Claudia intercepta cada mensagem, decodifica (pois só ela tem a chave privada relativa), lê, e só então a codifica novamente, usando a chave real de Bruno, e encaminha a mensagem para este, sem que ele ou Alice desconfiem do que está acontecendo.*

Naturalmente, trata-se de um golpe complexo, então se você não tem razão para crer que algo similar possa acontecer em seu caso, a publicação via correio eletrônico ou página na Internet pode ser um bom método – contanto que esteja ciente dos riscos. No final desta seção é explicado como a verificação da impressão digital da chave pode minimizar esse problema.

É possível ainda exportar sua chave privada, útil para cópias de reserva ou para transferir seu uso do GnuPG para outra máquina sem que seja necessário criar uma nova chave. O comando é igual ao acima, só que ao invés de usar o parâmetro "`--export`", use "`--export-secret-key`". Muito cuidado para não divulgar sua chave privada! Verifique sempre se as linhas delimitadoras da chave indicam: `-----BEGIN PGP **PUBLIC** KEY BLOCK-----` ou `-----BEGIN PGP **PRIVATE** KEY BLOCK-----`.

Importar chaves públicas de pessoas com quem deseja se comunicar é ainda mais fácil do que exportar. Basta digitar o comando:

`gpg --import *nomedoarquivo*`

onde *nomedoarquivo* é o nome do arquivo que contém a chave pública desejada, podendo ser tanto o arquivo binário gerado pelo GnuPG como um arquivo qualquer contendo a chave em modo texto (incluindo as linhas tracejadas delimitadoras, como mencionado acima). O arquivo não precisa sequer conter somente a chave, desde que ela esteja escrita integralmente. Isso torna especialmente fácil a importação de chaves enviadas por correio eletrônico, em que basta gravar a mensagem em um arquivo de textos e executar o comando acima.

Embora o valor da chave em si seja de difícil verificação devido ao seu tamanho, o problema de confiabilidade descrito acima pode ser contornado comparando o valor da impressão digital da mesma com o verdadeiro dono da chave. Impressões digitais são únicas para cada chave, e pequenas o suficiente para que a confirmação possa ser feita pessoalmente, por telefone, ou quaisquer outros meios, desde que você consiga certificar-se de que está de fato conversando com o dono da chave (não se esqueça que *emails* podem ser forjados).

Para ver a impressão digital de qualquer chave importada, ou a sua própria, digite o comando "`gpg --fingerprint *id*`", onde *id* é o número de identificação ou qualquer campo do identificador de usuário (como o endereço de correio eletrônico) da chave desejada. O mesmo resultado pode ser obtido digitando "`gpg --edit-key *id*`" e, em seguida, digitando o comando "`fpr`". Caso deseje ver a impressão digital de todas as chaves armazenadas em seu arquivo-chaveiro, digite apenas "`gpg --fingerprint`", sem especificar o *id*.

Em nosso exemplo, ao digitarmos "**`gpg --fingerprint jninguem`**", o GnuPG retornou o seguinte:

```
pub 1024D/D6B70989 2004-01-25 João Ninguém (gerente) <jninguem@exemplo.com.br>
    Impressão da chave = 0A0F 1A7F AD0A 8850 4281 4213 55A5 D604 0C52 CCFA
sub 1024g/37A4B0DB 2004-01-25
```

A impressão digital da chave, nesse caso, é "`0A0F 1A7F AD0A 8850 4281 4213 55A5 D604 0C52 CCFA`". Se a impressão obtida após a importação for igual a do proprietário legítimo da chave, você pode ter certeza de que sua cópia está correta. Chaves importadas devem ser **sempre** validadas.

Após essa verificação, você pode assinar a nova chave com sua chave privada, efetivando a validação da mesma, como mostrado na seção 5 deste documento. Assinar chaves verificadas é uma boa prática de segurança, assim como o uso apenas de chaves públicas assinadas por você. Dessa forma, ainda que seu arquivo-chaveiro tenha sido modificado indevidamente, você verá que as chaves adulteradas não estão assinadas – afinal, apenas você tem a senha que permite assinaturas com sua chave privada.

6 – Codificando e decodificando arquivos

Como visto no início da seção 3, o GnuPG utiliza um método de criptografia por chave assimétrica. Ao codificar arquivos, você precisa da chave pública da pessoa a quem deseja enviá-los. Essa pode ser até mesmo sua própria chave pública, caso deseje guardar seus arquivos de modo que apenas você consiga abri-los. Uma vez codificado com a chave pública, apenas a pessoa com a chave privada relativa – e a senha de acesso – poderá reverter o processo. Fazendo uma analogia, é como se cada pessoa disponibilizasse um cofre aberto (a chave pública) para quem desejasse. Qualquer um pode colocar documentos e objetos e fechar o cofre (criptografar usando a chave pública). Uma vez fechado, no entanto, apenas o dono legítimo do cofre sabe a combinação exata e pode recuperar o que foi colocado lá dentro (decodificar usando a chave privada).

Para codificar um arquivo, portanto, certifique-se de que a chave pública do destinatário foi devidamente importada como visto na seção 3.3 e, em seguida, entre com o comando:

```
gpg --output arquivosecreto --encrypt --recipient id arquivooriginal
```

onde *arquivosecreto* é o nome desejado para o arquivo depois de ter sido codificado, *id* é o número de identificação ou qualquer campo do identificador de usuário (como o endereço de correio eletrônico) da chave desejada, e *arquivooriginal* é o nome do arquivo original a ser codificado. O parâmetro "`--encrypt`" (ou simplesmente "`-e`") indica que desejamos codificar um arquivo, enquanto que o parâmetro "`--recipient`" (ou simplesmente "`-r`") deve sempre preceder o id da chave pública desejada para codificar o arquivo. Em nosso exemplo, suponha que o usuário João queira criptografar o arquivo *fofocas.txt* para sua irmã Maria. Maria lhe entrega sua chave pública e João a importa em seu programa. Para codificar o arquivo, ele digita: "`gpg --output fofocas.gpg -encrypt --recipient maria fofocas.txt`". O arquivo *fofocas.gpg* será criado no diretório atual, e pode ser enviado.

Para mensagens enviadas por correio eletrônico, pode ser mais prático codificá-las em modo texto, adicionando a opção "`--armor`" ao comando de codificação descrito acima:

```
gpg --armor --output arquivosecreto --encrypt --recipient id arquivooriginal
```

O arquivo gerado pode ser aberto em qualquer editor de textos e contém a mensagem em formato ASCII, similar ao mostrado abaixo:

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.2.4 (GNU/Linux)

hQEOA6W3xuw3pLDbEAQAgD2TAeG8aauIoafqypNW44dkvDe/RPtjY9AhYqF6SOvd
jSLbAbO24D2QF7fBx4PHbl6omywGuggknVvbuCVb5dXTuos4IPmvBNcR/Xg66P2b
VxTYTe78tiPqnVO/O6VNfbNf5eVRfBSQkSptq0XqRrA3EKiL7O9GP0vpr5OUZQ==
=WpTg
-----END PGP MESSAGE-----
```

É claro que, quanto maior o arquivo ou mensagem, maior o tamanho do conteúdo criptografado entre as linhas tracejadas.

Outra funcionalidade oferecida pelo GnuPG é a possibilidade de codificar o mesmo arquivo para vários destinatários. Suponha que você deseja enviar um arquivo sigiloso para três pessoas diferentes, cada um com sua respectiva chave pública. Codificar o mesmo arquivo três vezes, cada vez usando uma chave pública diferente, é uma tarefa extremamente ineficiente. Ao invés disso, você pode codificar o arquivo usando as três chaves públicas ao mesmo tempo. Dessa forma, qualquer um dos destinatários poderá decodificar o arquivo corretamente, e ninguém mais. Para tal, basta adicionar ao seu comando de codificação (visto acima) tantos "`--recipient id`" quanto desejar. Pode ser interessante incluir sua própria chave pública sempre que for codificar arquivos para os outros, de modo que você também possa abri-los. Em nosso exemplo, para que João pudesse também decodificar o arquivo codificado para Maria, bastaria que ele incluísse seu próprio `id` ao comando anterior, que ficaria:

```
gpg --output fofocas.gpg --encrypt --recipient maria --recipient jninguem fofocas.txt
```

Para decodificar arquivos criptografados com sua chave pública, basta digitar:

```
gpg --output arquivooriginal --decrypt arquivosecreto
```

onde *`arquivosecreto`* é o nome do arquivo previamente codificado, e *`arquivooriginal`* é o nome desejado para o arquivo após ter sido decodificado. É importante avisar ao destinatário qual o tipo de arquivo, para que ele possa renomeá-lo de acordo. O parâmetro "`--decrypt`" (ou simplesmente "`-d`") indica que desejamos decodificar um arquivo. Em nosso exemplo, Maria deveria executar o comando "`gpg --output fofocas.txt --decrypt fofocas.gpg`" para ver o conteúdo do arquivo enviado por João. O nome escolhido por Maria como arquivo de saída não precisa ser necessariamente igual ao original. De fato, ela pode até omitir o mesmo, para que o GnuPG exiba o conteúdo diretamente na tela (útil para ver rapidamente curtas mensagens de texto, ou usar essa saída para alimentar outros programas).

Além da criptografia por chave pública (assimétrica), o GnuPG permite a codificação de arquivos com criptografia simétrica, em que você não precisa de suas chaves, e fornece uma senha especial para decodificação no inicio do processo de codificação. Note que essa é uma senha à parte e não está em nada relacionada com a senha de sua chave privada. Portanto, use sempre uma senha diferente da sua senha de chave privada. A codificação simétrica é especialmente útil para criptografar arquivos quando ainda não sabemos a chave pública do destinatário, ou quando desejamos simplesmente codificar arquivos locais, em que a senha só precisa ser conhecida pelo próprio autor. Para codificar arquivos com criptografia simétrica, digite o comando:

```
gpg --output arquivosecreto --symmetric arquivooriginal
```

O parâmetro "`--symmetric`" (ou simplesmente "`-c`") indica que o usuário deseja codificar o arquivo usando criptografia simétrica. Evite ao máximo divulgar senhas em texto puro via canais inseguros de comunicação como telefone ou correio eletrônico. Esse é o calcanhar de Aquiles da criptografia simétrica.

7 – Assinaturas

Mais importante ainda do que enviar mensagens criptografadas, o GnuPG oferece a possibilidade de assinar arquivos. Assinaturas digitais validam a integridade do arquivo, procedência e data e hora em que a mesma foi feita. O procedimento é similar ao da codificação, mas com chaves invertidas: a chave privada do próprio remetente é usada para assinar o documento, e o destinatário pode então conferir a assinatura usando a chave pública do primeiro, disponível para ele. Como somente o verdadeiro dono da chave pública tem acesso a chave privada, podemos afirmar que – se a assinatura conferir – o arquivo foi realmente enviado por ele (ou alguém mais tem acesso a sua chave privada, indicando que a mesma não é mais confiável), além de não ter sido modificado desde sua assinatura, realizada exatamente na data e hora indicada na máquina local do usuário. Caso qualquer modificação tenha sido feita ao arquivo, a verificação da assinatura acusará erro. O método de assinaturas digitais é importante para garantir a integridade de mensagens de correio eletrônico, por exemplo, e é feito com o parâmetro "`--sign`" (podendo ser abreviado para "`-s`"). Sua sintaxe é:

```
gpg --output arquivoassinado --sign arquivooriginal
```

Em nosso exemplo, caso quiséssemos assinar o documento *fofocas.txt*, bastaria digitar:

```
gpg --output fofocassinada.sig --sign fofocas.txt
```

O GnuPG indica qual chave será usada e pede confirmação de sua senha. Em seguida, gera o arquivo assinado com o nome desejado (no exemplo, *fofocassinada.sig*). Se o parâmetro "`--output`" for omitido, o GnuPG criará um arquivo com o mesmo nome do original, seguido da extensão "*.sig*". É possível criar chaves específicas para assinaturas, como indicado na seção 3.2 deste documento. O arquivo original é comprimido antes de ser assinado, e gravado em modo binário. Em muitos casos, como no envio de correio eletrônico, pode ser mais prático gerar o arquivo assinado em modo texto, sem qualquer compressão. Para isso, basta trocar o parâmetro "`--sign`" por "`--clearsign`", como mostra o exemplo abaixo:

```
gpg --output fofocas.sig --clearsign fofocas.txt
```

Caso o parâmetro "`--output`" tenha sido omitido, o GnuPG criará um arquivo com o mesmo nome do original, seguido da extensão "*.asc*". Veja o conteúdo do arquivo assinado, resultado de nosso comando exemplo:

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

Essa é uma mensagem exemplo.

Note que ela NÃO esta codificada, e usuários
maliciosos poderão visualizá-la, embora não possam
modificá-la sem que a assinatura perca sua validade.

-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.2.4 (GNU/Linux)

iD8DBQFALD7Xa0ccxta3CYkRAsR9AJ0YENYATxOYtfHGUcCvNOJlozbxHQCgqYJY
Xpf5uAEWOfrQqD5GLecl1mg=
=G0Qm
-----END PGP SIGNATURE-----
```

O conteúdo do arquivo está em texto puro e incluso dentro dos delimitadores do GnuPG, que exibe também o algoritmo de assinatura utilizado em conjunto com sua chave (no caso, o `SHA1` [11]). Ao final do arquivo, a assinatura propriamente dita é exibida.

É possível ainda assinar um arquivo colocando a assinatura em um arquivo separado, mantendo assim o original em sua íntegra. Isso é ideal para quando o arquivo foi enviado antes da assinatura, ou quando não desejamos modificar o arquivo e este não está em modo texto. Para fazer a assinatura separada, troque os parâmetros "`--sign`" ou "`--clearsign`" por "`--detach-sig`", como indicado:

gpg --output *arquivoassinatura* --detach-sig *arquivooriginal*

Nesse caso, *arquivoassinatura* conterá apenas a assinatura (em modo binário) do arquivo *arquivooriginal*, que não foi modificado. Para que a assinatura seja criada em modo texto, ideal para transmissão via correio eletrônico, adicione o parâmetro "`--armor`" ao início do comando. Caso o parâmetro "`--output`" indicando o arquivo de assinatura seja omitido, este será criado automaticamente, com o nome igual ao do arquivo original, mas com a extensão *.sig* (para assinatura em modo binário) ou *.asc* (para assinatura em modo texto).

Para verificar a assinatura de um arquivo, basta digitar:

gpg --verify *arquivoassinado*

ou

gpg --verify *arquivoassinatura arquivooriginal*

caso a assinatura tenha sido gravada em um arquivo separado.

O GnuPG procurará em seu arquivo-chaveiro por uma chave pública relativa ao arquivo assinado. Caso não seja encontrada nenhuma equivalência, exibirá uma mensagem parecida com a que se segue:

```
gpg: Assinatura feita em 02/13/04 18:34:01  usando chave DSA ID D6B70989
gpg: Impossível verificar assinatura: chave pública não encontrada
```

Em nosso exemplo, assim que o destinatário da mensagem importar a chave pública de João em seu arquivo-chaveiro (explicado na seção 3.3), a mensagem se transformará em:

```
gpg: Assinatura feita em 02/13/04 18:34:01  usando chave DSA ID D6B70989
gpg: Assinatura correta de "João Ninguém (gerente) <jninguem@exemplo.com.br>"
```

Caso o GnuPG exiba a mensagem "*Impossível verificar assinatura: chave pública não encontrada*" e você tenha a chave pública do remetente devidamente importada em seu arquivo chaveiro, então ou o remetente está usando uma chave diferente (mais velha ou mais nova), ou alguém tentou personificá-lo, e o arquivo deve ser considerado inseguro e inválido.

Chaves também podem ser assinadas, para garantir que seu arquivo-chaveiro não será alterado sem seu consentimento. Verifique se a chave pública de um destinatário foi previamente assinada por você antes de codificar quaisquer dados com ela, mas tome muito cuidado para somente assinar chaves quando tiver absoluta certeza de que o dono da chave é realmente quem ele afirma ser. Isso é feito através de comparação da impressão digital da chave pública com o dono legítimo pessoalmente, por telefone, ou quaisquer meios em que se tenha certeza de estar falando com o dono legítimo da mesma, como foi visto na seção 3.3. Para assinar uma chave, entre no modo de edição da chave (visto na seção 3.2) e entre com o comando "`sign`". No exemplo abaixo, João vai assinar a chave pública que recebeu e importou de Maria, depois de ter comparado impressões digitais com ela:

```
gpg --edit-key maria

pub  1024D/E0C81642  criada: 2004-02-13 expira: never      confiança: -/-
sub  1024g/CFD21E5A  criada: 2004-02-13 expira: never
(1). Maria Ninguém (irmã) <marianinguem@exemplo.com.br>

Comando> sign

pub  1024D/E0C81642  criada: 2004-02-13 expira: never      confiança: -/-
Impressão da chave primária: C7A5 30B6 4E31 A217 DC53 7807 9097 5D8F E0C8 1642

     Maria Ninguém (irmã) <marianinguem@exemplo.com.br>

Com que cuidado verificou que chave que está prestes a assinar pertence
a pessoa correta? Se não sabe o que responder, escolha "0".

   (0) Não vou responder. (padrão)
   (1) Não verifiquei.
   (2) Verifiquei por alto.
   (3) Verifiquei com bastante cuidado.

Sua escolha? (entre com '?' para mais informações): 3
```

Indique neste momento o grau de cuidado tomado por você ao verificar a validade da chave. Lembramos mais uma vez a importância de se verificar com bastante cuidado se a chave a ser assinada pertence realmente a pessoa certa, e por isso marcamos a opção *3*. O GnuPG pedirá outra

confirmação, sua senha, e então finalmente colocará sua assinatura na chave escolhida, como mostrado abaixo.

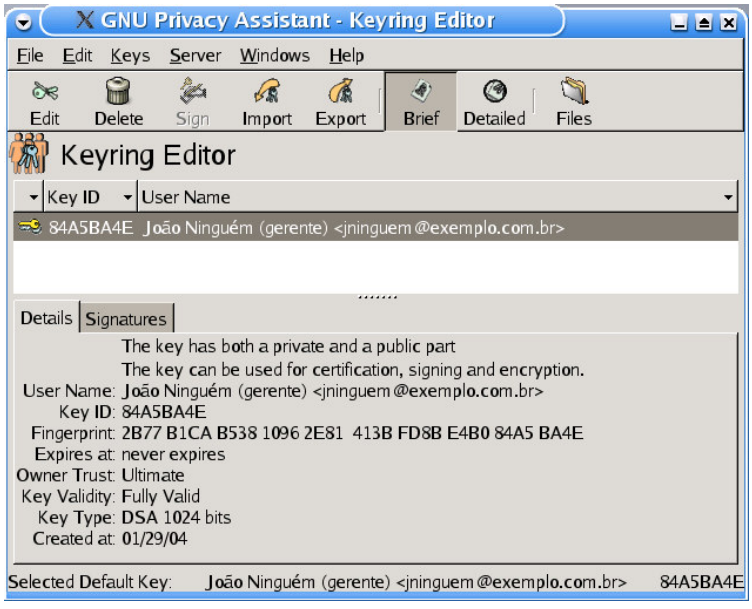
```
Você tem certeza de que quer assinar esta chave com
sua chave: "João Ninguém (gerente) <jninguem@exemplo.com.br>" (D6B70989)

Verifiquei esta chave com muito cuidado.

Realmente assinar? s

Você precisa de uma frase secreta para desbloquear a chave secreta do
usuário: "João Ninguém (gerente) <jninguem@exemplo.com.br>"
chave de 1024-bit/DSA, ID D6B70989, criada em 2004-01-25
```

8 – Interfaces gráficas para o GnuPG

Visando facilitar o acesso às funcionalidades do GnuPG, diversas interfaces gráficas foram criadas para o mesmo, permitindo que usuários gerenciem suas chaves públicas e privadas, codifiquem, decodifiquem e assinem arquivos com o clique de um botão. Nessa seção, veremos o procedimento de instalação de algumas das principais interfaces gráficas gratuitas para o GnuPG, disponíveis em diversas plataformas. Informações detalhadas sobre o uso de cada ferramenta vão além do escopo deste documento.

8.1 – GPA

Plataforma: GNU/Linux
Endereço: *http://www.gnupg.org/(en)/related_software/gpa/index.html*



O *GNU Privacy Assistant* (GPA) é a interface gráfica "oficial" do GnuPG (veja figura ao lado), desenvolvida pelo próprio projeto GnuPG. Com ela, qualquer usuário obtém acesso rápido e fácil à funcionalidade do GnuPG, e pode codificar e decodificar arquivos, gerenciar chaves, entre outros.

Para compilá-lo e instalá-lo, você precisará antes compilar e instalar os pacotes do `libgpg-error` e `gpgme` (versão maior ou igual a 0.4.3), nessa ordem. Todos os três podem ser obtidos na página oficial do GnuPG. O procedimento é igual para os três:

```
$ tar zxvf ferramenta.tar.gz
$ cd ferramenta
$ ./configure
$ make
$ /bin/su -c "make install"
```

No exemplo acima, troque a palavra "ferramenta" pelo nome do arquivo que está instalando (por exemplo, "`libgpg-error-0.6`", "`gpgme-0.4.3`" e "`gpa-0.7.0`"). Finalmente, antes de usar o GPA, você precisará reiniciar o sistema ou simplesmente executar o comando `ldconfig` como usuário *root*:

```
# ldconfig
```

Agora qualquer usuário do sistema pode executar o GPA para administrar suas chaves, abrindo uma janela de terminal no ambiente gráfico e executando o comando:

```
$ gpa
```

Ou adicionando um atalho para o mesmo na barra de tarefas ou área de trabalho de seu ambiente gráfico favorito.

Nota: O GPA precisa também do GTK (Gimp Tool Kit, disponível em *http://www.gtk.org/*), mas o mais provável é que ele já esteja instalado em seu sistema. Caso o GPA reclame a inexistência ou desatualização do GTK durante a compilação, instale o GTK e tente novamente.

8.2 – WinPT

Plataforma: Windows
Endereço: *http://winpt.sourceforge.net/en/*

O *Windows Privacy Tools* (WinPT) é uma coleção de pequenos aplicativos para facilitar a codificação, decodificação e assinatura de dados, compatível com o GnuPG. Entre suas ferramentas encontram-se um gerenciador de chaves, um gerenciador de arquivos codificados, plugins para criptografar e assinar arquivos diretamente do Windows Explorer (veja figura ao lado) e mensagens nos clientes de email Eudora e Outlook Express, além de permitir os mesmos procedimentos para qualquer texto que se encontre na área de transferência do sistema, o que permite integração com qualquer programa que exiba textos, como outros clientes de email e seu editor de textos favorito.

Para instalá-lo, pegue o arquivo na seção "`download`" do sítio oficial, verifique sua integridade através do md5 obtido na página (exatamente como fez para o GnuPG na seção 2) e execute-o. Durante o processo de instalação, o WinPT vai exibir a mensagem "`Select the folder in which you want to store your key pairs`". Nesse campo, indique o diretório onde armazenou suas chaves (ver seção 3).

Antes de executá-lo, repare que uma versão mais antiga do GnuPG foi instalada junto com o WinPT. Para corrigir esse problema, copie os arquivos do GnuPG mais recente que você está utilizando (localizado no diretório escolhido na seção 2.1) no diretório "`C:\Arquivos de programas\Windows Privacy Tools\GnuPG`" (ou no caminho que você escolheu para a instalação do WinPT), substituindo os arquivos antigos existentes lá.

Ao ser executado, o WinPT coloca um ícone na barra do Windows, como o mostrado ao lado, de onde todas as suas opções podem ser selecionadas.

9 – Programas de correio eletrônico para o GnuPG

Uma das maiores funcionalidades do GnuPG é a possibilidade de troca de mensagens de correio eletrônico (email) criptografado. No entanto, usuários que precisam enviar e receber constantemente emails criptografados podem achar cansativa a rotina de codificação e decodificação manual. Por isso, inúmeras ferramentas de email incluem suporte automatizado ao GnuPG, tornando seu uso fácil e rápido. Veremos agora rapidamente algumas ferramentas de correio eletrônico gratuitas e compatíveis com o GnuPG, para diversas plataformas.

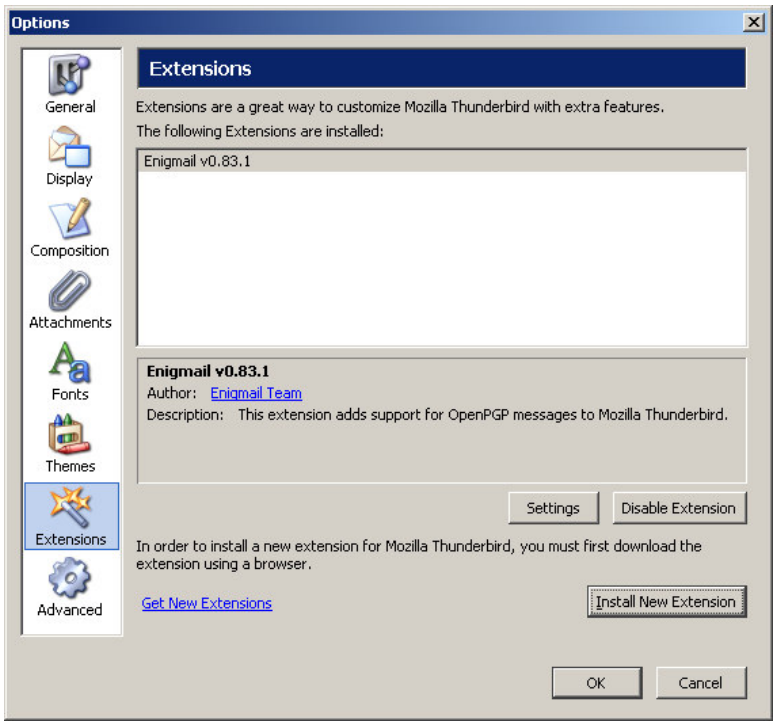
9.1 – Mozilla / Thunderbird

Plataformas: Windows; GNU/Linux; MacOS X
Endereço: *http://www.mozilla.org*

Mozilla é um poderoso navegador de Internet, multiplataforma e com cliente de email embutido. Thunderbird é a versão isolada com apenas o cliente de email. Eles não vêm com suporte nativo ao GnuPG, mas possuem um plugin chamado Enigmail, que pode ser obtido em *http://enigmail.mozdev.org/*.

Usuários de GNU/Linux devem fazer a instalação certificando-se de que possuem privilégios de *root*, e que o diretório `/usr` está montado com permissão para leitura e escrita.

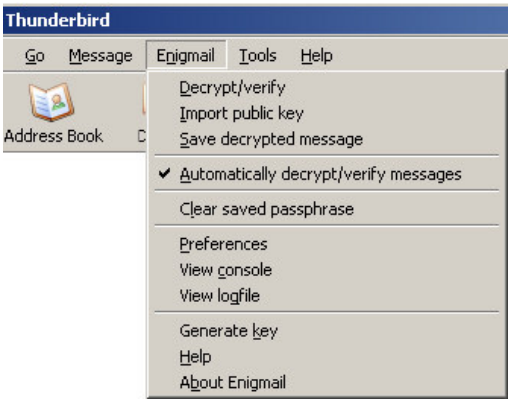
Para instalação no Mozilla, entre com o mesmo na seção "download" do sítio do Enigmail, e clique no botão "Install" específico da sua versão do Mozilla (percorra a página até encontrar sua versão).



Já para o Thunderbird, entre na seção "thunderbird" do sítio do Enigmail e pegue o arquivo "`enigmail-latest.xpi`", junto com o arquivo "`enigmime-latest`" específico de sua versão, como indicado no sítio do Enigmail (se estiver usando o GNU/Linux, por exemplo, o arquivo será "`enigmime-latest-linux.xpi`"). Agora, abra o Thunderbird e entre em "Tools"→ "Options". Na seção "Extensions", clique no botão "Install New Extension" (veja figura ao lado) e escolha o arquivo "`enigmail-latest.xpi`" que acabou de baixar. Repita a operação para o arquivo "enigmime" específico de seu

sistema e reinicie o Thunderbird. Uma nova opção no menu principal deve aparecer, como mostra a figura abaixo, de onde toda a funcionalidade do Enigmail pode ser configurada.

Em sistemas GNU/Linux, o primeiro reinício do programa após a ativação do Enigmail pelo *root* também precisa ser feito nessa conta. Posteriores inicializações não precisam ter privilégios de *root*.



Se o programa travar imediatamente após ter sido instalado o Enigmail, apenas reinicie-o. Esse problema infelizmente ocorre devido ao cache XUL e costuma parar após a segunda ou terceira reinicialização do programa. Caso o problema persista, você pode como último recurso apagar o arquivo "`XUL.mfasl`" (GNU/Linux, Mac OS X) ou "`XUL.mfl`" (Windows), localizado no diretório do seu perfil de usuário e reiniciar o programa.

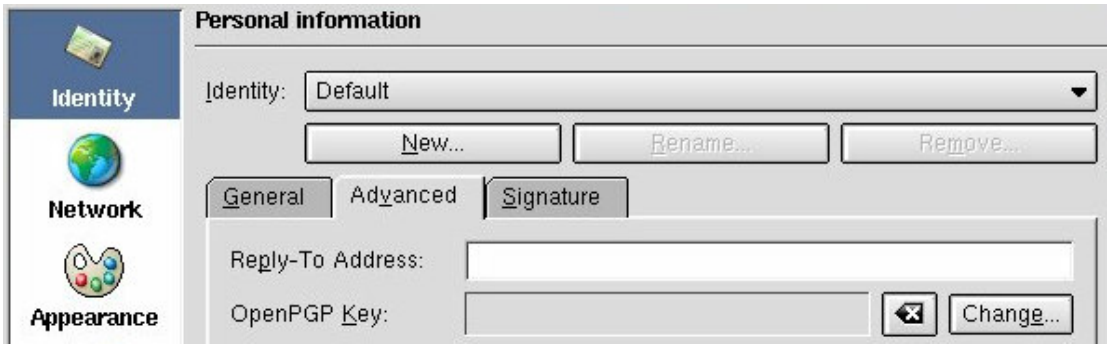
Antes de poder desfrutar do Enigmail, é preciso indicar onde está o GnuPG. Duas coisas podem ser feitas: colocar o caminho do GnuPG na sua variável de caminho (PATH), ou indicar o caminho explicitamente entrando na opção "Preferences" do menu do Enigmail e preenchendo o campo "GPG executable path" com o caminho em que o GnuPG foi instalado (por exemplo, `C:\ferramentas\gpg\gpg.exe`). Para verificar se o Enigmail localizou o executável do GnuPG, entre na opção "About Enigmail" do menu.

Finalmente, para ativar o uso do GnuPG em determinada conta e definir configurações padrão para assinatura e codificação, entre (caso esteja usando o Mozilla) em "`Edit`"→"`Mail & Newsgroups Account Settings`" na janela do "Mail & Newsgroups" ou (caso esteja usando o Thunderbird) em "`Tools`"→ "`Account Settings`" e entre no item "`OpenPGP Security`" da conta que deseja configurar. Repare que agora o menu da janela de composição de novas mensagens conta com novos itens, específicos do Enigmail, como mostra a figura ao lado.

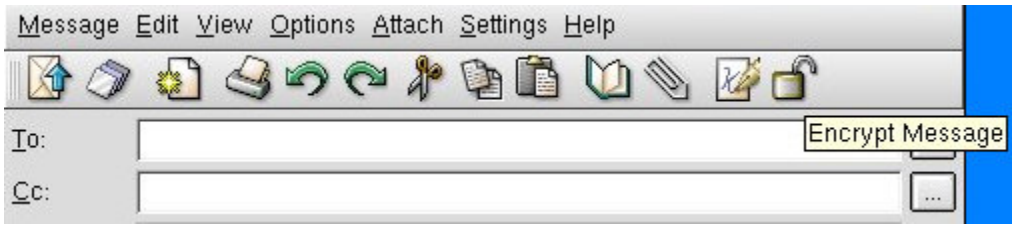
9.2 – Kmail

Plataforma: GNU/Linux
Endereço: *http://kmail.kde.org/*

Kmail é o cliente de email padrão da interface gráfica KDE, e possui suporte integrado ao GnuPG. Para ativar o uso do GnuPG, entre na configuração do Kmail e adicione sua chave no campo "`OpenPGP Key`", na aba "`Advanced`" da seção "`Identity`", como mostra a figura abaixo:



Note que, ao enviar mensagens, elas por padrão não são criptografadas. Para isso, marque os botões "`Sign Message`" e "`Encrypt Message`" na barra de ferramentas do Kmail (os dois botões mais à direita na figura abaixo).



9.3 – Pine

Plataforma: GNU/Linux; Windows, MacOS X; BeOS; *BSD; OS/2; OpenVMS...
Endereço: *http://www.washington.edu/pine/*

Pine é um famoso cliente de correio eletrônico que funciona em linha de comando. Feito inicialmente para usuários inexperientes, suporta hoje em dia uma série de opções e configurações avançadas. Uma de suas principais vantagens é, por funcionar em linha de comando, poder ser executado remotamente (cuidado para fazê-lo somente por canais seguros – como o ssh – do contrário de nada adiantará codificar a mensagem, pois alguém pode tê-la lido antes mesmo de você codificá-la). Embora somente as versões para UNIX e Windows sejam suportadas pelos criadores do Pine, distribuições do mesmo para plataformas como o GNU/Linux e tantas outras já estão consolidadas.

O Pine em si não oferece suporte ao GnuPG. No entanto, dada a popularidade de ambos os programas, diversas soluções de integração foram implementadas, como o programa "`PinePGP`" (disponível em *http://www.megaloman.com/~hany/software/pinepgp*) e o script "`Pine Privacy Guard`" (disponível em *http://quantumlab.net/pine_privacy_guard/*). Veremos à seguir uma breve explicação sobre a instalação e uso do último:

O Pine Privacy Guard (PinePG) é um pequeno script em Perl que permite aos usuários do Pine codificar, decodificar, assinar e verificar assinaturas em mensagens de correio eletrônico. Seus pré-requisitos são: *Perl 5*, *Pine 4.10*, *GnuPG 1.0.5*, *shellutils 2.0*, ou versões mais recentes dos mesmos, e por isso veremos aqui apenas a instalação em sistemas GNU/Linux (embora praticamente o mesmo procedimento funcione no UNIX, *BSD, MacOS X entre outros, sistemas como o Windows podem ter mais dificuldade para preencher tais requisitos). É provável que seu sistema já possua todas essas

ferramentas. Do contrário, elas são facilmente encontradas na Internet. Para instalá-lo, baixe o arquivo da página oficial do mesmo e expanda-o em um subdiretório seu para uso pessoal, ou em um diretório genérico (como `/usr/local/pinepg`) para que todos os usuários possam utilizá-lo. Quando este documento foi desenvolvido, o PinePG estava na versão 1.02, então entramos no diretório escolhido onde o arquivo "`pinepg-1.02.tgz`" foi baixado e digitamos:

```
$ tar -zxvf pinepg-1.02
```

Feito isso, procure o arquivo `pine_privacy_guard.pl`, e edite-o. A primeira linha deste arquivo deve conter o caminho completo para o programa Perl, precedido pelos caracteres "**#!**". Verifique se esse caminho está correto para o seu sistema (o comando "`which perl`" deve informar o caminho do Perl em seu sistema), assim como os caminhos das variáveis listadas no trecho "`Set full paths to files`" desse arquivo. Finalmente, garanta que o script pode ser lido por todos digitando:

```
$ chmod 0755 pine_privacy_guard.pl
```

Agora vamos às modificações dentro do Pine. Entre em:

(M)ain Menu → (S)etup → (C)onfig → (W)hereIs

E digite "`display-filters`", sem as aspas, no campo "`Word to find`". Para alterar os filtros de exibição e envio do Pine, coloque as seguintes configurações na tela exibida (nas linhas a seguir, substitua "`/usr/local/pinepg`" pelo diretório em que instalou o PinePG, caso diferente):

```
display-filters = _LEADING("-----BEGIN PGP MESSAGE-----")_
 /usr/local/pinepg/decrypt _RESULTFILE_ _DATAFILE_ _PREPENDKEY_,
_LEADING("-----BEGIN PGP SIGNED MESSAGE-----")_ /usr/local/pinepg/verify
_TMPFILE_ _RESULTFILE_

sending-filters = /usr/local/pinepg/clearsign _RESULTFILE_ _DATAFILE_
_PREPENDKEY_, /usr/local/pinepg/encrypt _RECIPIENTS_ _RESULTFILE_ _DATAFILE_
_PREPENDKEY_
```

Pronto. Agora, ao apertar a combinação de teclas "`ctrl-x`" sempre que for enviar uma mensagem, o Pine perguntará qual filtro deseja ser usado. Escolha 1 (sem filtro) para enviar mensagens sem codificação, 2 para codificar e assinar a mensagem, ou 3 para apenas assinar o email e enviá-lo em modo texto. Naturalmente, o PinePG só funcionará caso existam sua chave privada (para assinaturas) e a chave pública dos destinatários (para codificar). Mensagens enviadas para você serão decodificadas automaticamente. É possível ainda criar pseudônimos, adicionando-os ao arquivo "`.pinepg_aliases`", que deve ficar no seu diretório principal ($HOME). A sintaxe desse arquivo é simples:

```
email1 emailcomchave1 emailcomchave2 ...
```

O PinePG codificará todas as mensagens enviadas para "`email1`" com as chaves públicas de "`emailcomchave1`", "`emailcomchave2`", etc. Isso é especialmente útil por dois motivos: Se houver um endereço de correio eletrônico único para uma lista de usuários, cada qual com sua chave pública,

você pode adicionar uma linha indicando o PinePG a codificar emails enviados para essa lista com as chaves de todos os usuários, por exemplo:

```
turma@exemplo.com.br monica@exemplo.com.br cebolinha@exemplo.com.br  cascao@exemplo.com.br
```

A linha acima fará com que todos os emails enviados por você para "`turma@exemplo.com.br`" sejam codificados com as chaves em seu arquivo-chaveiro cujo identificador de usuário contenha os endereços "`monica@exemplo.com.br`", "`cebolinha@exemplo.com.br`" e "`cascao@exemplo.com.br`".

O arquivo de pseudônimos é útil também para enviar mensagens a uma pessoa cujo endereço de correio eletrônico é diferente do listado no identificador de usuário da mesma. Nosso usuário-exemplo possui o email "`jninguem@exemplo.com.br`" listado em seu identificador de usuário, mas gosta de receber mensagens pelo endereço "`zesilva@exemplo.com.br`", também pertencente a ele. Então, para enviar uma mensagem para "`zesilva@exemplo.com.br`" mas com a chave identificada pelo email "`jninguem@exemplo.com.br`", basta adicionar ao arquivo de pseudônimos a linha:

```
zesilva@exemplo.com.br jninguem@exemplo.com.br
```

indicando que mensagens para "`zesilva@exemplo.com.br`" devem ser codificadas com a chave pública de "`jninguem@exemplo.com.br`".

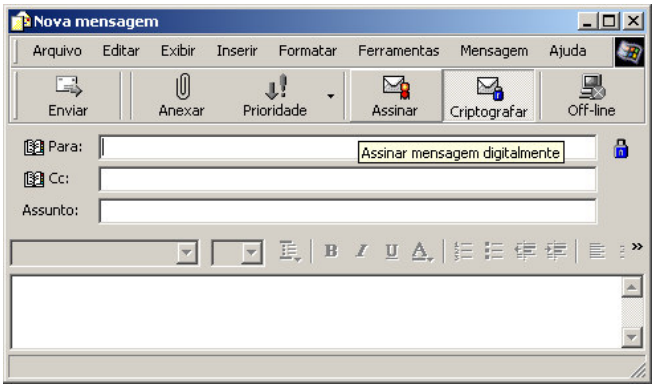
Outros detalhes também podem ser configurados, mas fogem ao escopo deste documento. Mais informações podem ser obtidas na página oficial do script.

Observação: Apenas um dos métodos de integração do Pine com o GnuPG foi citado, mas isso não significa que os demais sejam piores (ou melhores). Estes foram omitidos simplesmente por falta de espaço, já que não pretendemos fazer uma lista exaustiva. O Pine Privacy Guard foi escolhido aleatoriamente entre as opções conhecidas.

9.4 – Eudora/Outlook Express

Plataforma: Windows
Endereço: *http://www.eudora.com/* ; *http://www.microsoft.com/windows/ie/*

Os clientes de email Eudora e Outlook Express não possuem suporte direto ao GnuPG. No entanto, a ferramenta WinPT (ver seção 6.2) inclui plugins para integração com ambos. Após a instalação do WinPT com os plugins, novos botões deverão aparecer na barra de ferramentas, como mostra a figura abaixo:



10 – Guia de referência rápida do GnuPG

Criação, edição e remoção:

gpg --gen-key	cria um novo par de chaves
gpg --output *nomedoarquivo* --gen-revoke *id*	cria certificado de revogação para *id* e armazena no arquivo *nomedoarquivo*
gpg --edit-key *id*	entra em modo de edição para chaves de id
gpg --delete-key *id*	apaga chave pública de *id*
gpg --delete-secret-key *id*	apaga chave privada de *id*

Exportação e importação de chaves públicas:

gpg --armor --export *id*	exibe a chave pública de *id*
gpg --output *nomedoarquivo* --export *id*	grava chave pública de *id* no arquivo *nomedoarquivo*
gpg --armor --output *nomedoarquivo* --export *id*	idem ao descrito acima, mas grava em modo texto
gpg --import *nomedoarquivo*	importa chave contida no arquivo *nomedoarquivo*
gpg --fingerprint *id*	exibe a impressão digital da chave pública de *id*

Codificando e decodificando:

gpg --output *arq.gpg* --encrypt --recipient *id arq*	codifica arquivo *arq* em *arq.gpg*, usando chave de *id*
gpg --armor -o *arq.gpg* --encrypt --recipient *id arq*	idem ao descrito acima, mas grava em modo texto
gpg --output *arq* --decrypt *arq.gpg*	decodifica arquivo *arq.gpg* em *arq*
gpg --output *arq.gpg* --symmetric *arq*	codifica simetricamente *arq* em *arq.gpg*

Assinaturas:

(todos os arquivos-saída citados aqui podem ser modificados através do parâmetro "`--output arquivo`")

gpg --sign *arq*	assina *arq* e grava com compressão em *arq.sig*
gpg --clearsign *arq*	assina *arq* e grava em modo texto em *arq.asc*
gpg --detach-sig *arq*	grava assinatura de *arq* em modo binário em *arq.sig*
gpg --armor --detach-sig *arq*	grava assinatura de *arq* em modo texto em *arq.asc*
gpg --verify *arquivoassinado*	verifica assinatura de *arquivoassinado*
gpg --verify *arquivoassinatura arquivooriginal*	idem, mas para assinaturas em arquivos separados

Abreviações compreendidas pelo GnuPG para parâmetros mais usados:

-a	*--armor*
-o	*--output*
-d	*--decrypt*
-e	*--encrypt*
-r	*--recipient*
-c	*--symmetric*
-s	*--sign*
-b	*--detach-sign*

10.1 – Principais Comandos do Modo de Edição:

quit	Finaliza o modo de edição, perguntando se deve gravar alterações
save	Grava as alterações feitas e finaliza o modo de edição
help	Exibe listagem de todos os comandos do modo edição
fpr	Exibe impressão digital das chaves desse usuário
list	Lista chaves e identificadores de usuário
[enter]	Pressionar essa tecla em uma linha vazia é igual ao comando "list"
uid *n*	Marca identificador de usuário de índice *n*
key *n*	Marca chave secundária de índice *n*
adduid	Adiciona novo identificador de usuário
deluid	Remove identificador de usuário marcado
addkey	Adiciona nova chave secundária
delkey	Remove chave secundária marcada
check	Lista assinatura de todas as chaves desse usuário
sign	Assina a chave marcada
expire	Modifica a data de validade da chave secundária marcada ou da primária
primary	Define um identificador de usuário como primário
toggle	Troca o modo de exibição entre partes pública e privada das chaves
passwd	Modifica a senha desse usuário
revkey	Revoga chave selecionada
revuid	Revoga identificador de usuário selecionado
revsig	Varre assinaturas perguntando quais desejam ser revogadas
enable	Ativa uma chave desativada
disable	Desativa uma chave
showpref	Exibe as preferências
setpref	Configura as preferências

Referências:

[1] – DSA – *http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf*
[2] – ElGamal – *http://www.nullify.org/docs/elgamal.pdf*
[3] – RSA – *http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/*
[4] – 3DES – *http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf*
[5] – CAST5 – *http://www.ietf.org/rfc/rfc2144.txt*
[6] – Blowfish – *http://www.schneier.com/blowfish.html*
[7] – AES – *http://csrc.nist.gov/CryptoToolkit/aes/*
[8] – Twofish – *http://www.schneier.com/twofish.html*
[9] – *RFC 2440: OpenPGP Message Format* – *http://www.ietf.org/rfc/rfc2440.txt*
[10] – PGP: Pretty Good Privacy – *http://www.pgpi.org/*
[11] – SHA1 – *http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf*

Bibliografia:

Ashley, M. (maintainer) – GNU Privacy Handbook, Free Software Foundation
Kidwell, B. – Practical Introduction to GNU Privacy Guard in Windows
Baart, A. – GNU Privacy Guard Mini Howto
Manuais e documentação extra do GNU Privacy Guard incluídas na ferramenta
Documentação das interfaces gráficas para o GnuPG, incluída nas páginas oficiais supracitadas
Documentação dos clientes de correio eletrônico, incluída nas páginas oficiais supracitadas