Manual de Criptografia

Versão 1.0.0

ISO 1312

21 April, 2017

Conteúdo

PDF | EPUB

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.

CAPÍTULO 1

Criptografia com OpenPGP

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.

1.1 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 dificil 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 cave 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 quise 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.

1.2 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

mQGiBD6cuEoRBACQ6QKhCjN2qKHmOeeOdW9wOnnnd9V0eLREreSmMsRD6kSdXnrG
LQGrXMCBjelwOKB/vh2mcKn646PmSaq4sC+bLSOMAhME/IaDyDWZNWXo37yYlhDu
VD5wZGZNGMwov/bPDvjNjJTSXlo4/glbUNyVU2n9kiFYoDoSGjg1jQcYLwCgyeOJ
tHFSoelA62s+YmypI3KAzkcD/1S7Fim4p5X6HA/mUkFtuExDggba+OKmsmYNfPZ/
Q0sPAK85Or0zKAbGXAKUqezCuKCZ6FaesbIU5hTcQb2zKmzg1hzmtsWCTrxuj2X8
c//yRjDzFjT8KKCYKZWBGQTapGhvlHHq8ZsLlYZRzYyYtZ13a809FWk8G0Aq5/FA
be3KA/4nK+XAZUwXqXzu+xKINMu98zi9QWnVfBA5G132uCyrwOTBG1BV3803QFxG
XdBrEnGlz/RU9xjkVnEBWyqriO+lGDXuEZ6pWyx70UJ1S2qPDYKxoLTB68ZWAfKU
xYP8JYBC06hk6Ztq7FjkQGHxKMYQ9HCC3l9octyNsux+b/5Q+rQhU2lsdmlvIFJo
YXR0byA8cmhhdHRvQHJpc2V1cC5uZXQ+iFcEExECABcFAj6cuEoFCwcKAwQDFQMC
AxYCAQIXgAAKCRDA5viFRw5zwno5AKCGTBDCw9dalpr+SG7MQx00u1iehwCeKuGC
1FljdhHcu2CQX/JZoWLB/Qi5AQ0EPpy4TRAEAKmsLjxI2k/ITXu2kZJ7+SQPftb9
```

```
yRs8FxOsnmytkUwO8HaryPfuoMU51xG8XYL4blGL6u2J67KJOO4R3buwUDmH16RC
+nmMNlnaa0zlozsYIuB+r3s18hNLAss1LX0P0Ob6Ownar5VM8yNgVhEZkwBs6VhV
lfinYThWmpaXXLU3AAMFA/498Mfl1rs4z6vzkmIGu3Mqy+2CXSA/oCp9zPfFLJNM
+WUGhpbxkbbsNEzHdTFWPcoHi23k01KjT5CAiyiP30o6g8OV+3/WRYqeR4UNT6e8
7JDZo8kzjnTigI7XoAkqTJhL8pzhzvjboGZAaN1LDPgO2H//iRaBUjjrGaOTl9x2
c4hGBBgRAgAGBQI+nLhNAAoJEMDm+IVHDnPCsrYAoL9sLObVCteWmkAPFL3b5e/p
UFfAAKCzRRY3tPu6sHczFzOcw3SzeDN5x5kBogQ+nLhKEQQAkOkCoQozdqih5jnn
jnVvcDp553fVdHi0RK3kpjLEQ+pEnV56xi0Bq1zAgY3pcDigf74dpnCp+uOj5kmq
uLAvmy0jjAITBPyGg8g1mTVl6N+8mJYQ7lQ+cGRmTRjMKL/2zw74zYyU0l5aOP4J
W1DclVNp/ZIhWKA6Eho4NY0HGC8AoMnjibRxUqHpQOtrPmJsqSNygM5HA/9UuxYp
uKeV+hwP5lJBbbhMQ4IG2vjiprJmDXz2f0NLDwCvOTq9MygGxlwClKnswrigmehW
nrGyFOYU3EG9syps4NYc5rbFgk68bo9l/HP/8kYw8xY0/CigmCmVgRkE2qRob5Rx
6vGbC5WGUc2MmLWdd2vNPRVpPBtAKufxQG3tygP+JyvlwGVMF6l87vsSiDTLvfM4
vUFp1XwQORtd9rgsq8DkwRtQVd/NN0BcRl3QaxJxpc/0VPcY5FZxAVsqq4jvpRg1
7hGeqVsse9FCdUtqjw2CsaC0wevGVgHylMWD/CWAQtOoZOmbauxY5EBh8SjGEPRw
gt5faHLcjbLsfm/+UPq0IVNpbHZpbyBSaGF0dG8gPHJoYXR0b0ByaXNldXAubmV0
PohXBBMRAgAXBQI+nLhKBQsHCgMEAxUDAgMWAgECF4AACgkQwOb4hUcOc8J6OQCg
hkwQwsPXWpaa/khuzEMdNLtYnocAnirhgtRZY3YR3LtgkF/yWaFiwf0IuQENBD6c
uE0QBACprC48SNpPyE17tpGSe/kkD37W/ckbPBcTrJ5srZFMDvB2q8j37qDFOdcR
vF2C+G5Ri+rtieuyiTjuEd27sFA5h9ekQvp5jDZZ2mtM5aM7GCLgfq97NfITSwLL
NS19D9Dm+jsJ2q+VTPMjYFYRGZMAbOlYVZX4p2E4VpqWl1y1NwADBQP+PfDH5da7
OM+r85JiBrtzKsvtgl0gP6Aqfcz3xSyTTPllBoaW8ZG27DRMx3UxVj3KB4tt5NNS
o0+QgIsoj99KOoPDlft/1kWKnkeFDU+nvOyQ2aPJM4504oCO16AJKkyYS/Kc4c74
26BmQGjdSwz4Dth//4kWgVI46xmjk5fcdnOIRgQYEQIABgUCPpy4TQAKCRDA5viF
Rw5zwrK2AKC/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.

1.3 Exemplo de chave privada

Uma chave privada tambem 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)

mQGiBD6cuEoRBACQ6QKhCjN2qKHmOeeOdW9wOnnnd9V0eLREreSmMsRD6kSdXnrG
LQGrXMCBjelwOKB/vh2mcKn646PmSaq4sC+bLSOMAhME/IaDyDWZNWXo37yYlhDu
VD5wZGZNGMwov/bPDvjNjJTSXlo4/glbUNyVU2n9kiFYoDoSGjg1jQcYLwCgyeOJ
tHFSoelA62s+YmypI3KAzkcD/1S7Fim4p5X6HA/mUkFtuExDggba+OKmsmYNfPZ/
Q0sPAK85Or0zKAbGXAKUqezCuKCZ6FaesbIU5hTcQb2zKmzg1hzmtsWCTrxuj2X8
c//yRjDzFjT8KKCYKZWBGQTapGhvlHHq8ZsLlYZRzYyYtZ13a809FWk8G0Aq5/FA
be3KA/4nK+XAZUwXqXzu+xKINMu98zi9QWnVfBA5G132uCyrwOTBG1BV3803QFxG
XdBrEnGlz/RU9xjkVnEBWyqriO+lGDXuEZ6pWyx70UJ1S2qPDYKxoLTB68ZWAfKU
xYP8JYBC06hk6Ztq7FjkQGHxKMYQ9HCC3l9octyNsux+b/5Q+rQhU2lsdmlvIFJo
YXR0byA8cmhhdHRvQHJpc2V1cC5uZXQ+iFcEExECABcFAj6cuEoFCwcKAwQDFQMC
vUFp1XwQORtd9rgsq8DkwRtQVd/NN0BcRl3QaxJxpc/0VPcY5FZxAVsqq4jvpRg1
7hGeqVsse9FCdUtqjw2CsaC0wevGVgHylMWD/CWAQtOoZOmbauxY5EBh8SjGEPRw
gt5faHLcjbLsfm/+UPq0IVNpbHZpbyBSaGF0dG8gPHJoYXR0b0ByaXNldXAubmV0
NS19D9Dm+jsJ2q+VTPMjYFYRGZMAbOlYVZX4p2E4VpqWl1y1NwADBQP+PfDH5da7
OM+r85JiBrtzKsvtgl0gP6Aqfcz3xSyTTPllBoaW8ZG27DRMx3UxVj3KB4tt5NNS
PohXBBMRAgAXBQI+nLhKBQsHCgMEAxUDAgMWAgECF4AACgkQwOb4hUcOc8J6OQCg
hkwQwsPXWpaa/khuzEMdNLtYnocAnirhgtRZY3YR3LtgkF/yWaFiwf0IuQENBD6c
uE0QBACprC48SNpPyE17tpGSe/kkD37W/ckbPBcTrJ5srZFMDvB2q8j37qDFOdcR
vF2C+G5Ri+rtieuyiTjuEd27sFA5h9ekQvp5jDZZ2mtM5aM7GCLgfq97NfITSwLL
7JDZo8kzjnTigI7XoAkqTJhL8pzhzvjboGZAaN1LDPgO2H//iRaBUjjrGaOTl9x2
c4hGBBgRAgAGBQI+nLhNAAoJEMDm+IVHDnPCsrYAoL9sLObVCteWmkAPFL3b5e/p
1FljdhHcu2CQX/JZoWLB/Qi5AQ0EPpy4TRAEAKmsLjxI2k/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)

hQEOA3v8xQeh8DSxEAP/Rks1pm5WOyNZTtlJP1FVK602ix872ZMwsOsOYZ9oBC+o
pN9/03jLjlk3hF0v9KJkq3SQxu+E7zSCbuMigAum0QZgV3BMVjZDGKLDUN3D8Sgt
fBju6Y7Vn6wK87OWQlayWMAab+t77wQPjKuH9IPYJwsZ6zJJK5YIhgIlKU0FoKAD
/RWDmqQua/iswijTh5StgO+FJuseec22TgGPzZC4D05s9JE7gMIO3GvSTGR+re4i
```

```
SaCzrppudxOacsuCcRhR6IkA6lT8fKaZImU/aMev/UgwadGP3XeqiwxPUt+qHg6H
iYZyMShKAMZzF+PdgflaDYGffgIQBXpxWxjRMufX9LTW0lYBo1mQ6W1XgJG3AY47
knPyqSKRfuNPwayAGRdZM2yoq/DnwIUGB5cyfiNqkV9lXl25uXud99T3mDojYLzr
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)

hQEOAwXhZ7S+8an6EAP8CwSfq+TjKziYiM4QGwAwNynKDLSkMdw6KeTmSi3XgfP+
Wf6p6KWGuVmD7ffGMwCBuB+mKpjmXhZ9EyR4JAVpgkMMN2ZCrTcMu4tDZqqvJV/t
jws1dwg0QaeCdmA6jbxBDuWUsbgOzxY/5URmttCfKGTgIOOh73VDZ3HXM4MrWIcE
AKFaCgiMybhYZLlXqEpqgyJWH54fsXSgRbv2yZE2G+9aNEolAVji7NfluKGhrosF
gRCEuFc1pGRnF+Z5+aMj/xK/9DKeqGt97aQbViQ33s/3soqskBWzd95rKK1ZyBRn
iIEoEXlc+eDRk+j6EsjR+Ex87fU4gCM9aNPQGkdLZZpJhQEOAzrq0korX+quEAQA
gOEV5STP2zCgQ5sSljd6D02lLXVky5P/4vvgBPUUpH9svMQUF5iJTPc8H/YUoOmW
Hpmk6nwXDsAwDhyoTuiiPHRBd+D1ckCtwsjRZePOhkU2r0m+9k4kI3Q34XX3jPDI
YABHRWP0tJaazCU/1PD1rKA0JWoLmA3x3i2AEizwLrkD/R+8xBgOUEV+XNJSEXqS
ooHbKCu9dItIXjHj2kzTpA5F9anT0xZYG+Q9SivWavZVAd8KLcDM7My1s/udjGyp
KWeSxcyHu6rYmdJw1by+eWddKn+2W7iSgnaOhjEimZcy53wYmrh5Wzv5vDL0Qd2d
RSX12GjdqW9FmOQvag2q8sIAhQIOA7jcJ6nlDzObEAf9EPB7HaCBFSovTp/c2GsU
HEyc3gRQ2RD1XPlnKA/PhE5zTIGCXiRdVLfQuoZ4xOwHng8ppM5xJkIU5oohhysu
m7YFlkM7rG9kavVAoATDeNfoLXjA7q/Cntl9xyy2NEUU9oQCilrS6JTvJ9UPPY1U
kW9zS8CNNIAgIQ+rVJZtc1E+34N8xORk+4nQz4QtY0nE6XMN83bIYfoIxGc0dn8O
PVOz5ej6utcZV2sdTZc0JnX3lXonUhkSCvsDcxRfUOemfObR2XZMq7OrFoKQ1mly
TQ9AW2mUyEbEtVsFdT8HOAC3M7z/EmnK97nAykONYko1RrHgWY/ekhElqxfvzy2I
6Qf9F9tTLshUy7QdM7WiGoJaxPwTUwlzGw29rnZo36JNKyWW8qnGU5+YH5iKzM4u
xeAOt12SKcBgzN2ucEdmOxm0A1rYL0IgDLWTYnD/YkmJbSp4xhwKEUEF4juiJdUd
ZTf1+tF6Q6yIRIFwLy3ES3NX6sA/dxkiN2YROC41VtcGqn84lCj3c4VnMw7LcLtX
GBP9y76tObNEXsCYGIsVqjxMXDFh0smQ8v1Loo1qFxBCjuPXWWBWL77814o2rS+Y
1eH+Q9rv5RajMygDD1afYxKoI1mey7yuRbiG02owHDNMs6wECCH77bRnxAVRLSWD
UCsIRMk0qWKApD/M7qIWn4qtl9JWAcG4KKrOGdIYswARUoJIMKn+TMJmbA8TxTKo
J7yrXniv3C9pyeEPqRUWdZqo4szVDKPGLz+VMCV4tzSE5iINGBNNXAM9HxqwmAX9
tqLVyGhs9UVe8RY=
=cbe0
-----END PGP MESSAGE-----
```

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

1.4 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 mantenho 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! A experiência mostra que é mais eficaz imprimir do que copiar manualmente, uma vez que é muito errar na escrita dos caracteres. Mas se você imprimir, confira o fingerprint impresso para evitar uma impressora mal intencionada :P

1.5 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)

iD8DBQFAuSSqvSy1nGtWZ3cRAvbuAJ9CZgA3YXCWCWiHMgtqA1pnjUqnaQCgvvrv
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.

1.6 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.

1.7 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.

1.8 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 criculação 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 (versão em inglês)
- Wikibook sobre criptografia (inglês)

CAPÍTULO 2

Instalando 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. Aqui trataremos o uso do GPG no modo texto e no modo gráfico. Leia nota sobre modo texto e modo gráfico.

2.1 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.

2.2 Modo Texto no 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"
```

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.

2.3 Modo Gráfico no 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:

```
sudo apt-get update ; sudo 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.

CAPÍTULO 3

Usando os GPG em modo texto

Esta seção trata do uso diário da criptografia, tanto em modo texto quando 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.

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.

3.1 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:

   (a) DSA e ElGamal (padrão)

   (b) DSA (apenas assinatura)

   (c) 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 = chave expira em n dias w = chave expira em n semanas m = chave expira em n meses 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 mova o mouse aleatóriamente 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]

    (a) 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.

3.2 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.

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.

3.3 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 chava 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.

3.4 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
```

3.5 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 clerasign 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)
```

```
iD8DBQFAMQPyvSy1nGtWZ3cRAld5AJ4sIOed+/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

3.6 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
0gPrLalgr5OrC4gC9LahbOw=
=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
```

3.7 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@daSTOPSPAM.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.

3.8 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)

hQEOA3v8xQeh8DSxEAP8DGLGac9OdZzX7KxRqpkaYuJ/8NVN5AyhtQKZwRogZwwZ
19g/teTPQLYwgcCLQoDKxsnX3lBGEzjAGxXme6aqcJetD0sXcULtap99AfpJqIO/
lCDjxFqNRP45UwnnbafakiVDTj71H6jDi6UMP4c+F/JJL65Q9ROHW08kOB1IM1sD
```

```
/RsQu1pqZl/X8PRZyVdLSwpzGpR5uaxA837f+Zl0+1Fh2DhoiE2AxnLXdwlMlRtK
SrpqBdWifULoX46E1+DOd128e24K74d+utMux4uk8t9Lb0D5C8RDfShKHoLJIwul
fhEmtN8bo2Kmg/z8sWnDjW3Ik3opVtsTPNPQQh1J9GV40lUBViB6IzhkXhhNZ4ae
qexbLilOVwJczLa3y3UmwkiXcs92k6thpUCIJjyeRTtpey2LKdVLHLd1o5ti8/or
nqYoq1eXHcVOckmfxH3Uq8ZAEX6bzSKc
=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)

hQEOA3v8xQeh8DSxEAP8DGLGac9OdZzX7KxRqpkaYuJ/8NVN5AyhtQKZwRogZwwZ
19g/teTPQLYwgcCLQoDKxsnX3lBGEzjAGxXme6aqcJetD0sXcULtap99AfpJqIO/
lCDjxFqNRP45UwnnbafakiVDTj71H6jDi6UMP4c+F/JJL65Q9ROHW08kOB1IM1sD
/RsQu1pqZl/X8PRZyVdLSwpzGpR5uaxA837f+Zl0+1Fh2DhoiE2AxnLXdwlMlRtK
SrpqBdWifULoX46E1+DOd128e24K74d+utMux4uk8t9Lb0D5C8RDfShKHoLJIwul
fhEmtN8bo2Kmg/z8sWnDjW3Ik3opVtsTPNPQQh1J9GV40lUBViB6IzhkXhhNZ4ae
qexbLilOVwJczLa3y3UmwkiXcs92k6thpUCIJjyeRTtpey2LKdVLHLd1o5ti8/or
nqYoq1eXHcVOckmfxH3Uq8ZAEX6bzSKc
=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 6B566777)

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 desencriptá-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".

3.9 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 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 esta        4.2 -  As informações não bateram, por isso você não deve ass
            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.

3.10 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 modificaçõs 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.

3.11 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 ente, 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.

Voce 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.

3.12 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 criptografas para você ou assinar mensagens. Lembre-se de revogar sua chave pública antes de cancelá-la (veja como na próxima seção).

3.13 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.

3.14 Outros comandos

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

```
man gpg
```

3.15 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`
- 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 --search-keys email-ou-nome`
- Receber chaves de um servidor: `gpg --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`

CAPÍTULO 4

Criptografia e internet

4.1 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 comprenderá 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.govSTOPSPAM.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.

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 mesagem 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 mensage 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@destinatarioSTOPSPAM.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@destinatarioSTOPSPAM.br, ficticio@ficticio.comSTOPSPAM.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.

4.2 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 comprenderá 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 seguira. Vejamos como funciona:

Quando você usa um nagevador 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.

`Inserir imagem: cacert.jpg`

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.

`Inserir imagem: cacert2.jpg`

O processo da conexão segura via web é o seguinte: suponha que você esteja visitiando o site 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.

`Inserir imagem: https.jpg`

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 smile

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.

`Inserir imagem: https2.jpg`

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 portando a conexão segura pode estar comprometida.

```
Inserir imagem: certificate.jpg
```

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.

CAPÍTULO 5

Listas criptografadas

A infra-estrutura de comunicação da maioria dos projetos é precária em termos de privacidade e segurança. Os servidores quem armazena todos os arquivos de todas as listas, por exemplo, em geral não possuem seu disco criptografado. Assim, tudo que já trafegou por ele poderia ser facilmente acessado.

Em momentos de necessidade deve existir um canal seguro para trocar mensagens e as pessoas devem estar preparadas para usá-lo. Para ajudar nisso, existem gerenciadores de listas criptografadas.

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:

- schleuder: atualmente o software mais recomendado para listas de discussão pela sua capacidade de remoção de muitos metadados das mensagens e também por funcionar como um re-enviador (remailer) de mensagens.
- 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, além de possuir uma interface administrativa por email.

Outros softwares semelhantes, porém não recomendados, podem ser encontrados na documentação do Firma.

5.1 Outras vantagens

- Removem boa parte dos metadados das mensagens.
- Opcionalmente podem ofuscar o remetente nos metadados dos emails. Assim, mesmo que alguém capture as mensagens recebidas no servidor, nem sempre (já que isso depende de muitos fatores) conseguirá descobrir quem as enviou.

5.2 Dicas gerais

Para qualquer um destes softwares, valem as seguintes dicas:

- NÂO SE ESQUEÇAM: mandem apenas mensagens criptografadas e assinadas, senão a lista não aceita a mensagem e vocês correm o risco de mandar uma mensagem não-criptografada pela internet.
- Quando for iniciar uma nova thread de mensagens, escolha assuntos sem sentido ou pouco informativos, porque o assunto das mensagens é a única coisa que não tem jeito de estar criptografado. Não é apenas uma limitação dos atuais softwares de listas criptografadas, mas interente aos padrões de email.

- Pra quem usa Thunderbird: verifique se suas mensagens assinadas e/ou criptografadas estão com PGP/MIME: no menu OpenPGP da janela principal do Thunderbird, existe a opção "Preferências". Clicando nela, aparecerá uma janela com diversas abas. Uma dessas abas contém opções de PGP/MIME: vá até essa aba e certifique-se que seu Thunderbird está configurado para usar sempre PGP/MIME. O PGP/MIME é um esquema de enviar a mensagem criptografada e a assinatura como anexos. Fica mais bonito :)
- Como configurar o Thundebird para que sempre que você escreva uma mensagem para esta lista ela seja automaticamente criptografada, sem que vocês precise selecionar isso no menu OpenPGP da janela de composição. Isso é interessante porque pode acontecer de alguém mandar uma mensagem para a lista e esquecer de criptografar. A mensagem não vai entrar na lista, mas trafegará pela internet sem estar criptografada. Vá em OpenPGP / Edit Per-Recipient Rules / Add. Daí na janela que abrir você seleciona os emails afetados por essa regra, que no caso seria o email da lista; depois seleciona as chaves associadas a essa regra, ou seja, a chave da lista; e por último, na parte "Defaults for...", você define qual a opção padrão para "Signing", "Encrypting" e "PGP/MIME", sendo as opções "Always", "Never" e "Yes, if selected in Message Composition".
- Para enviar mensagens totalmente válidas a uma lista do schleuder, é necessário criptografá-las com a chave pública da lista e assiná-las com a chave privada do/a rementente, do contrário a lista reclama ou, dependendo da configuração, pode até recusar a mensagem.
- Lembre-se: criptografia não é maquiagem ou coisa chique, mas sim uma ferramenta importantíssima. Pode ser difícil usá-la no começo, mas todos deveriam fazer um esforço.

5.3 Schleuder

Vantagens do schleuder:

- É possível configurar a lista para apenas enviar mensagens cifradas.
- Possui capacidade de remailing (repasse de mensagens), permitindo que a lista criptografada proteja o remetente de uma mensagem ao re-enviá-la para terceiros.

5.4 Firma

Vantagens do firma:

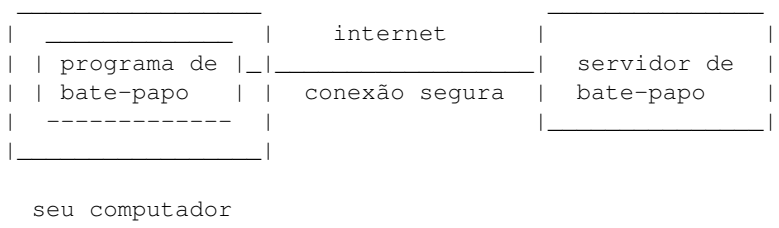
- Não mantém mensagens descriptografadas armazenadas no servidor em nenhum momento.
- Interface administrativa amigável.

Bate-papo com Conexão Segura (SSL)

Conforme diz a seção Criptografia e Internet, a internet é uma rede na qual as informações podem ser facilmente interceptadas. Isso quer dizer que usando um sistema de bate-papo comum é possível que mesmo em conversas privadas alguém escute o seu diálogo.

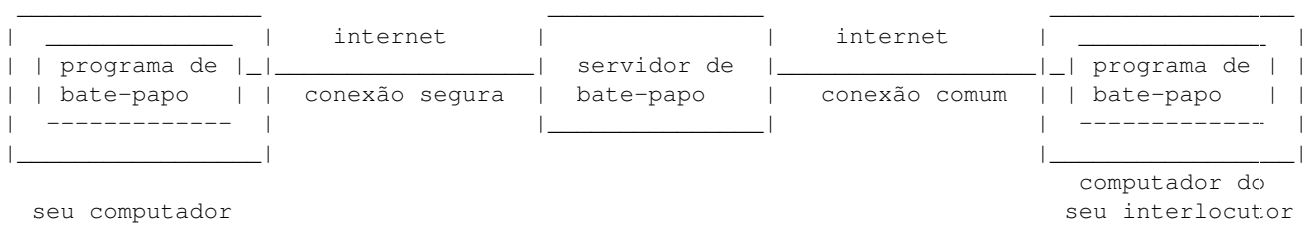
Para evitar esse tipo de coisa e preservar sua identidade, é possível utilizar a tecnologia SSL (Secure Sockets Layer), também conhecida como conexão segura.

A conexão segura utiliza criptografia para conectar você ao servidor. Tudo o que é enviado ao servidor e dele até você só poderá ser interpretado por ambas as partes. Seria algo como esse desenho:



As mensagens, antes de chegar ao servidor ou do servidor até você, passam por vários computadores e nesse caminho ainda poderá ser interceptada, mas só poderá ser interpretada se possuir uma das chaves privadas – a do seu computador, no caso das mensagens enviadas pelo servidor, ou a do servidor, no caso das mensagens enviadas pelo seu computador.

Se você usa conexão segura e está conversando no bate-papo com alguém que não usa conexão segura, então em as mensagens entre vocês só estarão criptografadas na metade do caminho entre você e seu interlocutor:



Para que a conversa seja realmente segura é indispensável que todas as partes envolvidas numa conversa utilizem a conexão segura.

Uma outra vantagem desse tipo de conexão é o mascaramento do IP dos usuários, que é número que identifica cada computador que está na internet. Se alguém sabe o seu número IP, é bem possível que ela saiba identificar qual em qual país está o seu computador e qual é o provedor de acesso.

Por isso, é interessante que o seu IP não esteja disponível para aqueles que frequentam o mesmo bate-papo que você, e é isso que a conexão segura faz: todos os usuários com conexão segura aparentam estar utilizando o próprio servidor

de bate-papo como seu computador pessoal.

Configurando uma conexão segura no bate-papo do CMI

Método simples

A forma mais simples de se conectar por conexão segura no bate-papo do CMI é através do endereço https://irc.indymedia.org/, mas esta forma não é totalmente segura, ainda sim sendo melhor que uma conexão comum. A seguir damos instruções para as formas mais seguras de conexão.

Usuários de GNU/Linux

Há várias formas de fazer uma conexão segura. A mais simples é usar um cliente de IRC com SSL habilitado. A seguir alguns exemplos de clientes com tal suporte:

- Irssi
- Xchat

Para todos eles, o seguinte comando deve funcionar:

```
/server -SSL irc.indymedia.org 994
```

Para o Xchat digite:

```
/sslserver irc.indymedia.org 994
```

No presente momento, o suporte ao Xchat está em avançado estágio, conseguindo habilitação em diferentes distribuições de GNU/Linux e outros Sistemas Operacionais. O método ideal é selecionar as caixinhas 'Usar SSL' e 'Aceitar certificado inválido' na tela 'Lista de Servidores'. Além disso na caixa Servidores adicionar 'irc.indymedia.org/994'. Quando está conectando você poderá ver informações sobre o certificado SSL seguido da habitual informação de conexão do IRC. Verifique que você aparece como irc@127.0.0STOPSPAM.1, para ter certeza que está conectado atravéz de conexão segura.

6.1 Stunnel

O outro, e talvez mais flexível meio de conectar é usando 'stunnel' em conjunto com o cliente de IRC da sua escolha. O código fonte, assim como binários para outros Sistemas Operacionais pode ser obtido em http://www.stunnel.org/download/. Se você está usando Debian GNU/Linux, você pode simplesmente digitar 'apt-get install stunnel'.

O primeiro passo é criar o túnel seguro entre seu computador e o irc.indymedia.org. Voce pode ajustar isso na porta que você quiser. Há algumas questões para manter em mente aqui:

- Usuários diferentes do root não podem atar-se para portas abaixo da 1024
- Muitos firewalls podem bloquear pacotes em portas abaixo de 1024
- O usuário que estiver executando stunnel precisa ter permissão de escrita numa pasta onde ele deixará um arquivo contendo o número do processo (pid) que o stunnel estiver rodando (normalmente /var/run)

Para o propósito deste tutorial nós vamos usar a porta local 6994. Ela é acessivel para os usuários, e assumindo que você tenha acesso para escrever o arquivo pid, você pode se sair bem. Se tiver problemas, mas tem acesso ao root disponível, o que você tem a fazer é rodar stunnel como root. Neste caso sinta-se livre para escolher qualquer porta que você quiser.

Assumimos o seguinte comando:

```
stunnel -c -d 6994 -r irc.indymedia.org:994
```

Isto avisa ao stunnel para ouvir na porta 6994 do seu computador local, e encaminhar a conexão para a porta 994 no irc.indymedia.org. Digite

```
stunnel -h
```

para uma lista completa de descrições destes sinais.

Você deve ter agora ter um túnel seguro habilitado. Para ter certeza, olhe o processo rodando no plano de fundo. Você pode também ver algo similar ao seguinte no seu sistema de log (normalmente /var/log/syslog):

```
Mar 24 21:13:48 yourhost stunnel[11990]: Using 'irc.indymedia.org.994' as tcpwrapper service name
Mar 24 21:13:48 yourhost stunnel[11990]: stunnel 3.22 on i586-pc-linux-gnu PTHREAD+LIBWRAP with
<nop>OpenSSL? 0.9.6c 21 dec 2001 Mar 24 21:13:48 yourhost stunnel[11991]: FD_SETSIZE=1024, file
ulimit=1024 -> 500 clients allowed
```

A parte difícil do processo está completa. Agora tudo que você tem que fazer é conectar do lado local do seu túnel com seu cliente de IRC. Não importa o cliente que você escolha, isto consiste em conectar-se ao localhost atravéz da porta que você escolheu (no nosso caso 6994). Para a maioria dos clientes de IRC o seguinte comando bastará :'/server localhost 6994'. Você deverá conectar-se normalmente.

Quando você faz essa conexão, o stunnel deve reportar algo como isto em seu sistema de logs:

```
Mar 24 21:51:30 yourhost stunnel[12073]: irc.indymedia.org.994 connected from 127.0.0.1:2780.
```

A partir deste ponto você deve estar conectado ao servidor de IRC, e pode curtir comunicação segura!

6.2 No Windows

Baixe o cliente em http://www.xchat.org / http://silverex.org (tenha certeza de estar baixando a versão Windows) e instale-o. Use o comando

```
/sslserver irc.indymedia.org 994
```

para se conectar através de SSL.

6.3 Sobre

Trecho extraído do Guia de Bate-papo seguro do CMI Brasil.

CAPÍTULO 7

Criptografia no sistema de arquivos

Uma vez que o seu chaveiro fica guardado no seu disco rígido ou outro dispositivo de armazenamento, 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:

7.1 No Linux

No GNU/Linux, existem as seguintes implementações:

- LUKS - Linux Unified Key Setup: um possível novo padrão para criptografia em disco no GNU/Linux; baseado na especificação TKS1 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:

- loop-aes: uma alternativa às implementações oficiais do Linux
- 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 este caso não utiliza o device-mapper.
- CFS: sistema de arquivos criptografado que utiliza o NFS como interface.
- TCFS: Transparent CFS (outra implementação do CFS).
- EncFS: sistema de arquivos criptografado em nível de usuário, utilizando o FUSE. Possui diversos problemas de segurança.
- 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.

CAPÍTULO 8

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 tem 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.

CAPÍTULO 9

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.
- OpenPGP Best Practices.
- Manual de privacidade GNU.

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 <http://www.gnupg.org/(en)/documentation/faqs.html>‘_.
- ‘**GNU PG Mini-HOWTO, por Brenno J.S.A.A.F. de Winter, Michael Fischer v. Mollard e Arjen** Baart <http://webber.dewinter.com/gnupg_howto/english/GPGMiniHowto.html>‘_.
- GPG no MacOSX.
- Ritual de assinatura de chaves no Debian.

CAPÍTULO 10

Baixando

Você pode baixar este guia usando

```
git clone --recursive https://git.fluxo.info/crypto
```

CAPÍTULO 11

Sobre

Este manual não tem sido atualizado o quanto deveria. Apesar de ser um bom roteiro introdutório sobre criptografias padrão OpenPGP e TLS, ele não cobre aspectos importantes como anonimato e novos protocolos de mensageria. Assim, consulte documentação adicional como o Guia de Autodefesa Digital.

Este guia é baseado num antigo manual de criptografia disponível originalmente na documentação do Indymedia, originalmente escrito por Pietro Bastardi (pietro em bastardi.net) e radicalmente modificado por Silvio Rhatto (rhatto em riseup.net). Algumas modificações também foram adicionadas por Luis (luis em riseup.net).

Mantido por ISO 1312.