

Segurança em Computação

Trabalho Individual 3

Adriano Tosetto - 15104099

30 de abril de 2019

1 Criação do Certificado

Antes de tudo foi necessário baixar o GPG. Para isso, foi executado o comando abaixo:

```
sudo apt-get install gpg
```

Após instalado, é necessário executar o comando:

```
gpg --gen-key
```

O GPG vai pedir algumas informações, como email e nome, então as chaves estarão disponíveis no local.

1.1 Backup da Private Key

Para realizar o backup da chave privada, o seguinte comando pode ser utilizado:

```
gpg --export-secret-keys KEYID > Desktop/private-key.asc
```

Onde o KEYID é identificador da chave.

1.2 Publicação da chave pública

Para mandar a chave para um servidor, o seguinte comando é necessário:

```
gpg --send-keys --keyserver <server> KEYID
```

onde **server** é o servidor para ser enviado, no meu caso foi <https://www.rnp.br>. KEYID é o id da chave. Também é possível mandar a chave diretamente no site do RNP clicando [Aqui](#).

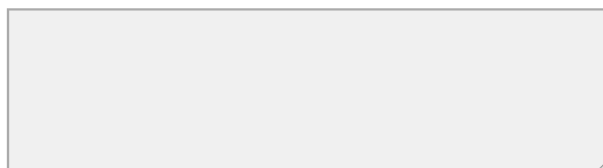
Figura 1: É possível copiar e colar a chave recém criada diretamente para o site do RNP

Submissão de chaves

AVISO IMPORTANTE: Tenha em mente que uma vez que uma chave é enviada para um servidor, esta será distribuída para outros servidores no mundo rapidamente, então tenha certeza do que está fazendo antes de submeter uma chave.

A **única** forma de remover uma chave pública do servidor é enviando um certificado de revogação, que irá requerer acesso à sua chave privada.

Cole a sua chave publica PGP aqui:



Limpar

Enviar

2 Revogação de um Certificado

Uma boa prática para certificados GPG é ter um **certificado de revogação**, caso o usuário perca sua chave privada. Para criar um, são necessários os seguintes comandos:

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

onde KEYID é o id da chave e revoke.asc é o certificado de revogação. É possível e recomendado mover esse certificado para outra máquina. Agora, para revogar o certificado, é necessário mais o seguinte comando:

```
gpg --import revoke.asc
```

E por fim

```
gpg --keyserver <server> --send-keys KEYID
```

Onde KEYID é o id da chave e **server** é keyserver.cais.rnp.br no meu caso.

Figura 2: Exemplo do meu certificado revogado

```
pub 3072R/4F5484F6 2019-04-30 *** KEY REVOKED *** [not verified]
Adriano Tosetto <adrianotosetto33@gmail.com>
```

3 Assinaturas de Certificados e Revogação das Mesmas

3.1 Assinar o certificado de um terceiro

Antes de assinar o certificado de alguém, é necessário adicionar a chave dessa pessoa no anel de chaves. Para tal:

```
gpg --recv-keys KEYID
```

O KEYID é o id de chave da outra pessoa.

Para assinar essa chave, o seguinte comando é necessário:

```
gpg --sign-key KEYID
```

Onde KEYID é o id da chave da pessoa.

Agora é necessário enviar o certificado assinado para o servidor realizando o seguinte comando:

```
gpg --keyserver <server> --send-keys KEYID
```

Onde KEYID é o id da chave da pessoa e <server> é o servidor para onde se está mandando o certificado assinado.

Figura 3: Eu assinei o certificado do aluno Gustavo Olegário

```
pub 2048R/0F7EFC5D 2019-04-27
Fingerprint=1860 394F 6819 38D4 3500 69CC D173 9888 0F7E FC5D uid Gustavo Olegario <gustavo-olegario@hotmail.com>
sig sig3 0F7EFC5D 2019-04-27 2021-04-26 [selfsig]
sig sig 1F0CC988 2019-04-27 2021-04-26 Adriano Tosetto <adriano.rafael19@hotmail.com>sub 2048R/7574F573 2019-04-27
sig sbind 0F7EFC5D 2019-04-27 2021-04-26 [sub 4096R/7298483C 2019-04-27]
sig sbind 0F7EFC5D 2019-04-27 [ ]
```

3.2 Revogar a assinatura

Para revogar a assinatura, utiliza-se o seguinte comando:

```
gpg --edit-key KEYID
```

Onde KEYID é o id da chave. Logo em seguida abre um terminal e é preciso entrar com o seguinte comando:

```
revsig
```

Ele vai pedir algumas informações (como o porquê da revogação) e só ir seguindo os passo. É preciso salvar a ação com:

```
save
```

Por fim, é preciso reenviar a chave para o servidor:

```
gpg --keyserver <server> --send-keys KEYID
```

Onde <server> é servidor (keyserver.cais.rnp.br, no meu caso) e KEYID é o id da chave.

Figura 4: Eu revoguei a assinatura do certificado do aluno Gustavo Olegário

```
pub 2048R/0F7EFC5D 2019-04-27
    Fingerprint=1860 394F 6B19 3804 3500 69CC D173 98B8 0F7E FC5D uid Gustavo Olegario <gustavo-olegario@hotmail.com>
sig sig3 0F7EFC5D 2019-04-27 _____ 2021-04-26 [selfsig]
sig sig 1F0CC988 2019-04-27 _____ Adriano Tosetto <adriano.rafael10@hotmail.com>
sig revok 1F0CC988 2019-04-27 _____ Adriano Tosetto <adriano.rafael10@hotmail.com>sub 2048R/7574F573 2019-04-27
sig sbind 0F7EFC5D 2019-04-27 _____ 2021-04-26 [sub 4096R/7298483C 2019-04-27]
sig sbind 0F7EFC5D 2019-04-27 _____ [ ]
```

4 Anel de Chaves Privadas

São as chaves do usuário que são usadas para aplicações GPG. Por exemplo, encriptar um arquivo irá usar um **subkey** específica, assinar um documento usará outra **subkey**, elas ficam no anel de chaves privadas.

5 Assinatura Local e em Servidor

Sem um servidor de certificados, se um usuário A assina o certificado de um usuário B, o usuário A deve mandar o certificado de B assinado para o usuário B. Quando B recebe seu certificado assinado por A, ele precisa passar para todos o certificado atualizado.

Com o uso de servidores, o usuário apenas assina o certificado de B e manda para o servidor e todos os outros usuários podem atualizar o certificado de B, agora assinado por A, apenas dando fetch diretamente do servidor.

6 Banco de Dados de Confiabilidade

Banco de dados de confiabilidade armazena as informações de confiança que o usuário tem sobre outras chaves. Para mudar o nível de confiança para uma dada chave, execute:

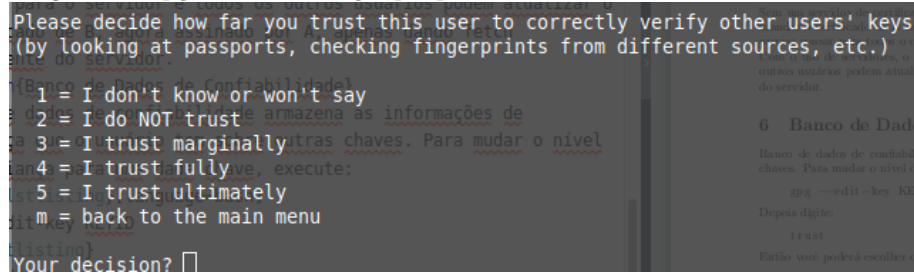
```
gpg --edit-key KEYID
```

Depois digite:

```
trust
```

Então você poderá escolher o nível de confiança.

Figura 5: Opções de confiança que para a chave do aluno **João Paulo Tiz**



7 Sub-chaves

Há mais de uma chave privada no Anel. A primeira é a *Master Key* e sua principal função é identificar o usuário. É ela que é usada para assinar o nome e email do usuário no certificado. As outras chaves no anel são as *subkeys*. Elas são usadas para encriptar e assinar dados reais. A *Master Key* assina as *subkeys* para mostrar que elas pertencem ao usuário.

A ideia desse esquema é fazer com que o gerenciamento de chaves se torne mais fácil. Com ele é possível substituir as *subkeys* e outro ponto é que a *Master Key* fica muito menos exposta.

8 Certificado GPG

Para enviar uma imagem basta:

```
gpg --edit-key KEYID
```

Na edição de chaves, digite:

```
addphoto
```

em seguida, digite o caminho do arquivo:

```
/home/user/dir/dir/foto_XXX.jpg
```

Salve a mudança:

```
save
```

Agora é preciso reenviar a chave:

```
gpg --keyserver <server> --send-keys KEYID
```

No meu caso, server foi o do RNP.

Figura 6: Foto enviada para o servidor associada com o meu certificado



Para verificar se a foto está associada com a chave:

```
gpg --list-options show-photos --list-keys
```

Figura 7: Como mostrado no print, minha chave tem uma foto minha associada

```
pub rsa3072 2019-04-27 [SC] [expires: 2021-04-26]  
13FAE943F8EE30985621125C52A4FF6D1F0CC9B8  
uid [ultimate] Adriano Tosetto <adriano.rafaell0@hotmail.com>  
uidn{figure}[H] [ultimate] [jpeg image of size 57894]  
suberi rsa3072 2019-04-27 [E] [expires: 2021-04-26]  
antioq(Foto enviada para o servidor associada com o meu
```

9 Envio de Arquivo Cifrados com GPG

Para encriptar usando o GPG, primeiro é preciso importar a chave para o local. No meu caso, usarei a chave pública do aluno *João Paulo Tiz* com o comando:

```
gpg --keyserver keyserver.cais.rnp.br --recv DCAE898A
```

Para encriptar:

```
gpg --output doc.gpg --encrypt --recipient DCAE898A doc
```

Figura 8: Tela do computador do Tiz após decifrar o arquivo cifrado que eu mandei

```
iptiz@dijkstra 3-gpg $ gpg --output doc --decrypt doc.gpg
gpg: encrypted with 2048-bit RSA key, ID 79D4F940C7144E8, created 2019-04-29
"João Paulo Taylor Ienczak Zanette (Forgot last password :P For studying purposes.) <jpaulotiz@gmail.com>"
iptiz@dijkstra 3-gpg $ cat doc
extremamente sigiloso
iptiz@dijkstra 3-gpg $
```

Figura 9: Tela do meu computador após eu decifrar uma mensagem que o Tiz me enviou usando minha chave pública

```
tosetto@tosetto-Inspiron-3437:~$ gpg --output Desktop/doc.txt --decrypt Desktop/for-tosetto-only.gpg
gpg: encrypted with 3072-bit RSA key, ID 97AB19DBFC46CC8F, created 2019-04-27
"Adriano Tosetto <adriano.rafaell10@hotmail.com>"
tosetto@tosetto-Inspiron-3437:~$ cat Desktop/doc.txt
Mensagem especial para você. Não divulgar!
```

10 Assinatura Anexada & Assinatura Separada

Para os dois casos, é necessário possuir as chaves de quem se está querendo verificar a assinatura em seu anel local. Eu vou verificar um documento assinado pelo aluno **João Paulo Tiz**, então é necessário atualizar as chaves dele no meu anel. Dessa forma, utiliza-se o comando:

```
gpg recv-keys KEYID
```

Onde KEYID é o id da chave do **Tiz** (DCAE898A).

10.1 Assinatura Anexada

Para assinar um documento com assinatura anexada basta:

```
gpg --encrypt --sign --local-user 1FOCC9B8 --armor -r jpaulotiz@gmail.com
→ para_assinar.txt
```

Para verificar, basta:

```
gpg --decrypt for-tosetto.txt.asc
```

Eu mandei o arquivo encriptado e assinado por mim: *para_assinar.txt.asc* para o **João Paulo Tiz** e ele me mandou o arquivo assinado e encriptado por ele: *for-tosetto.txt.asc*

Figura 10: Resultado da verificação de assinatura que o **João Paulo Tiz** me mandou

```
tosetto@tosetto-Inspiron-3437:~/Desktop$ gpg --decrypt for-tosetto.txt.asc
gpg: encrypted with 3072-bit RSA key, ID 97AB19DBFC46CC8F, created 2019-04-27
mandei) "Adriano Tosetto <adriano.rafaell10@hotmail.com>"
Oi Tosetto, the 12calitiz decifra meu doc. (peço)
gpg: Signature made Tue 30 Apr 2019 05:36:42 PM -03
gpg: using RSA key BC50CCDA2D2DBAD0302EFEAD1DFE185BDCAE898A da chave
gpg: Good signature from "João Paulo Taylor Ienczak Zanette (Forgot last password :P For studying purposes.) <jpaulotiz@gmail.com>" [uncertain]
gpg: WARNING: This key is not certified with a trusted signature!
gpg: a do meu co There is no indication that the signature belongs to the owner.
Primary key fingerprint: BC50 CCDA 2D2D BAD0 302E FEAD 1DFE 185B DCAE 898A
```

10.2 Assinatura Separada

Para fazer o mesmo com a assinatura separada, basta trocar a flag *-sign* por *-detach-sign*. O comando ficará assim:

```
gpg --encrypt --detach-sign --local-user 1FOCC9B8 --armor -r jpaulotiz@gmail.com  
  ➔ assinatura_separada.txt
```