**COMO GERAR UMA CHAVE PUBLICA/PRIVADA COM GPG**

[root@fedora22 ~]# **gpg --gen-key** //gera uma chave publica/privada (chave assimétrica)

gpg (GnuPG) 1.4.20; Copyright (C) 2015 Free Software Foundation, Inc.

This is free software: you are free to change and redistribute it.

There is NO WARRANTY, to the extent permitted by law.

Por favor selecione o tipo de chave desejado:

(1) RSA and RSA (default)

(2) DSA and Elgamal

(3) DSA (apenas assinatura)

(4) RSA (apenas assinatura)

Sua opção? **1**

RSA keys may be between 1024 and 4096 bits long.

What keysize do you want? (2048) 1024

O tamanho de chave pedido é 1024 bits

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

0 = chave não expira

<n> = chave expira em n dias

<n>w = chave expira em n semanas

<n>m = chave expira em n meses

<n>y = chave expira em n anos

A chave é valida por? **(0)**

Key does not expire at all

Is this correct? (y/N) **y**

You need a user ID to identify your key; the software constructs the user ID

from the Real Name, Comment and Email Address in this form:

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

Nome completo: **kenny alves** //nome será visto publicamente

Endereço de correio eletrônico: alvkennedy@gmail.com

Comentário: **testando**  //comentário será visto publicamente

Você selecionou este identificador de usuário:

"kenny alves (testando) <alvkennedy@gmail.com>"

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

Você precisa de uma frase secreta para proteger sua chave.

a frase secreta não foi repetida corretamente; tente outra vez.

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.

Não há bytes aleatórios suficientes. Por favor, faça algum outro trabalho

para que o sistema possa coletar mais entropia!

(São necessários mais 271 bytes)

...........+++++

..........+++++

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.

.+++++

..+++++

gpg: key F5DAF877 marked as ultimately trusted

chaves pública e privada criadas e assinadas.

gpg: a verificar a base de dados de confiança

gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model

gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u

pub 1024R/F5DAF877 2016-11-15

Key fingerprint = B3C2 F6E0 EFFD A9AC 5FB6 884B 6ED8 F946 F5DA F877

uid **kenny alves (testando) <alvkennedy@gmail.com>**

sub 1024R/3BA2975E 2016-11-15

**COMO LISTAR AS CHAVES EXISTENTES**

/root/.gnupg/pubring.gpg

------------------------

pub 1024R/F5DAF877 2016-11-15

**uid kenny alves (testando) <alvkennedy@gmail.com>**

sub 1024R/3BA2975E 2016-11-15

**EXPORTAR A CHAVE PÚBLICA A PARTIR DA CHAVE PRIVADA**

[root@fedora22 ~]# gpg --export kenny alves > kennyalves.pub //exporta chave publica

[root@fedora22 ~]# ls

anaconda-ks.cfg Desktop kennyalves.pub

**IMPORTANDO UMA CHAVE PÚBLICA AO CHAVEIRO**

[root@fedora22 ~]# gpg --import kennyalves.pub

gpg: key F5DAF877: "kenny alves (testando) <alvkennedy@gmail.com>" not changed

gpg: Número total processado: 1

gpg: não modificados: 1

**CRIANDO UM ARQUIVO CRIPTOGRAFADO**

gpg --out **NomeDoArquivoCriptografado** --recipient “**kenny alves**” --encrypt **ArquivoDeTextoQueSeraCriptografado.txt**

**DECRIPTOGRAFANDO UM ARQUIVO**

gpg --out **NomeDeSaidaDoArquivoDecriptografado** --decrypt **NomeDoArquivoCriptografadoRecebido**