Cifragem assimétrica – GnuPG

1 Introdução

Nos sistemas de cifra de chave pública cada utilizador possui um par de chaves que consiste numa chave privada e numa chave pública. Como o próprio nome indica, a chave privada deve-se manter secreta e nunca deve ser revelada. A chave pública pode ser distribuída para qualquer utilizador com quem pretendemos comunicar. O emissor cifra a informação com a chave pública do receptor e após a cifra só a chave privada do receptor pode decifrar a informação.

Este protocolo resolve o problema de troca de chaves dos sistemas de cifra simétrica. Não é necessário o emissor e o receptor acordarem entre si qual a chave a usar, basta apenas que o emissor obtenha uma cópia da chave pública do receptor. Além disso qualquer pessoa que queira comunicar com o receptor usa sempre a mesma chave pública. Por isso bastam *n* pares de chaves para que *n* pessoas possam comunicar de forma privada entre si.

Tanto na cifragem simétrica como na de chave pública a segurança reside nas chaves. Por isso o tamanho da chave é um bom indicador da segurança, mas não podemos comparar directamente o tamanho da chave de um sistema de chave simétrica com um de chave pública. Num ataque de força bruta a um sistema de chave simétrica com 128 bits existem 2^{128} chaves possíveis. Para um sistema de chave pública com 512 bits, o atacante tem de factorizar um número com 512 bits (até 155 dígitos decimais). Ou seja, a computação a efectuar é diferente, por isso actualmente uma chave simétrica de 128 bits é suficiente, mas numa chave pública, devido aos algoritmos de factorização existentes, deve-se usar uma chave pública de 1024 bits.

2 GnuPG

O GnuPG (http://www.gnupg.org) é uma ferramenta para comunicação e armazenamento de ficheiros de forma segura. A segurança do GnuPG assenta em vários sistemas de criptografia:

- Cifra simétrica
- Cifra de chave pública
- e *hashing*

2.1 Gerar chaves

Para podermos começar a usar o GnuPG temos que criar um par de chaves. O comando para a sua criação é o seguinte (pode ser necessário executar este comando mais do que uma vez se os ficheiros de configuração ainda não existirem):

```
[aluno@lcalinux aluno]$ gpg --gen-key
Please select what kind of key you want:
   (1) DSA and ElGamal (default)
   (2) DSA (sign only)
   (5) RSA (sign only)
Your selection?
```

O GnuPG é capaz de gerar vários tipos de chaves, mas uma chave primária deve ser capaz de produzir assinaturas digitais. Por isso a opção 1 é a que deve ser usada, onde são geradas 2 pares de chaves, um DSA para fazer assinaturas e outro (ElGamal) para cifrar. As outras opções geram apenas chaves para criar assinaturas.

O próximo passo é escolher o tamanho da chave. O GnuPG exigem que as chaves tenham no mínimo 768 bits. O valor por omissão é 2048 bits. Quanto maior for a chave mais segura ela será contra ataques do tipo força bruta, mas também as operações de cifragem e decifragem demorarão mais tempo.

Depois é necessário especificar a validade da chave:

```
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0)
```

É pedido também uma identificação do utilizador, é esta identificação que permite associar uma chave a uma pessoa. Nesta fase é criada apenas uma identificação do utilizador, mas é possível acrescentar mais ID's se pretendermos utilizar a chave em contextos diferentes (por exemplo como estudante e como trabalhador, em que os endereços de e-mail são diferentes).

```
You need a User-ID to identify your key; the software constructs the
user id from Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name:
```

Por fim é pedido uma frase senha para proteger a chave privada.

2.2 Gerar um certificado de revogação

Após a criação das chaves deve-se gerar um certificado de revogação. Se o utilizador se esquecer da sua frase senha, ou se a sua chave privado foi comprometida ou perdida, o

certificado de revogação deve ser publicado para alertar os utilizadores que a sua chave pública não deve ser usada. Com este certificado é possível verificar as assinaturas digitais feitas anteriormente, mas não permite a cifragem. O comando é o seguinte:

```
[aluno@lcalinux .gnupg]$ gpg --output revoke.asc --gen-revoke mfrade

sec  1024D/99B3E03A 2002-10-22   Miguel Frade <mfrade@estg.iplei.pt>

Create a revocation certificate for this key?
```

Em que "revoke.asc" é o ficheiro onde vai ser guardado o certificado e "mfrade" deve ser o identificador do utilizador para o qual pretendemos gerar o certificado.

2.3 Troca de chaves

Para comunicar é necessário trocar chaves públicas. Para listar as chaves que um utilizador possui:

```
[aluno@lcalinux .gnupg]$ gpg --list-keys
/home/aluno/.gnupg/pubring.gpg
------------------------------
pub  1024D/99B3E03A 2002-10-22 Miguel Frade <mfrade@estg.iplei.pt>
sub  1024g/5A8EBE3E 2002-10-22

pub  1024D/C89F8593 2002-10-22 Aluno (Teste do GnuPG) <aluno@estg.iplei.pt>
sub  1024g/213D934E 2002-10-22
```

Para trocarmos a nossa chave temos de a extrair primeiro:

```
[aluno@lcalinux aluno]$ gpg --output aluno.gpg --export aluno
```

A chave é exportada em formato binário, mas se pretendermos enviá-la por e-mail, sem ser num ficheiro anexo, este formato pode trazer problemas. Por isso podemos usar a opção "--armor" para gerar apenas caracteres legíveis:

```
[aluno@lcalinux aluno]$ gpg --output aluno.gpg --armor --export aluno
[aluno@lcalinux aluno]$ cat aluno.gpg

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.2.0 (GNU/Linux)

mQGiBD21fRYRBACSbjKVd216zPmQDW7c08G+mt9U60nPKQvPAgoRf+iFarp0l2MQ
5FmU4H/IpTmdnU2WWXZpnXxz/JtKj9fVyJRO4HE77pTtgyn4V2c3BhcYjl90DMU9
1dG5Vdh9NHaij6AhKq19ekAzfzT8fsP2cHkWbZFUUVUmPDX/OlIIk4G80wCgp2/n
 (...)
cpluTaWIRgQYEQIABgUCPbV9HAAKCRBVw0aZmbPgOigqAJ0e3NESDLZ0xZRvxPmt
I66StOFpnACfTti60Hv8eZnjhqcQW0nR+wzPfZg=
=FVcY
-----END PGP PUBLIC KEY BLOCK-----
```

Para importar uma chave:

```
[aluno@lcalinux aluno]$ gpg --import aluno.gpg
```

Depois de importar uma chave deve validá-la. A validação consiste em verificar a autenticidade da chave, para isso deve executar **gpg --edit-key <ID>**. O que nos leva a uma linha de comando onde podemos executar algumas tarefas (experimente digitar "help").

```
[aluno@lcalinux aluno]$ gpg --edit-key aluno

gpg: checking the trustdb
gpg: checking at depth 0 signed=1 ot(-/q/n/m/f/u)=0/0/0/0/0/1
gpg: checking at depth 1 signed=0 ot(-/q/n/m/f/u)=1/0/0/0/0/0
pub  1024D/C89F8593  created: 2002-10-22 expires: never      trust: -/f
sub  1024g/213D934E  created: 2002-10-22 expires: never
(1). Aluno (Teste do GnuPG) <aluno@estg.iplei.pt>

Command> fpr
pub  1024D/C89F8593 2002-10-22 Aluno (Teste do GnuPG) <aluno@estg.iplei.pt>
 Primary key fingerprint: 6A33 F71B 1179 B5C0 7726  E3C7 5E6B E8F9 C89F 8593
```

Aí o comando "fpr" imprime o "fingerprint" da chave importada que deve ser confirmada, por um meio seguro, se coincide com o "fingerprint" obtido pelo dono da chave. Após a verificação do "fingerprint" da chave, devemos assiná-la (comando **sign**), que é o mesmo que dizer devemos validar a chave. Existem mais formas de validar chaves que pode consultar no manual do GnuPG.

2.4 Cifrar documentos

Se pretendermos cifrar um documento para enviar a alguém, podemos usar o seguinte comando:

```
[aluno@lcalinux aluno]$ gpg --armor --encrypt --recipient aluno ola.txt
```

De onde resultará o ficheiro ola.txt.asc cifrado com a chave pública de "aluno". Para que se possa ler o ficheiro é necessário possuir a chave privada e executar o seguinte comando:

```
gpg --output ola.txt --decrypt ola.txt.asc
```

Se pretendemos guardar um ficheiro de forma segura, um backup que apenas possa ser lido por nós, podemos usar a cifragem simétrica:

```
gpg --symmetric ola.txt
```

2.5 Assinar documentos

Por vezes é necessário garantir a integridade e a autenticação de um ficheiro sem o cifrar. Um exemplo comum desta situação é a distribuição na internet de aplicações para o utilizador instalar. Além de ser necessário haver uma garantia de que o ficheiro foi descarregado correctamente, também é preciso ter garantias que a aplicação não foi publicada por outra entidade (por exemplo um hacker). Por isso é útil assinar os documentos e guardar a assinatura num ficheiro separado. Para criar a assinatura e verificá-la devem-se usar os seguintes comandos:

```
gpg --armor --output ~/doc2.sig --detach-sig ficheiro.txt
gpg --verify doc2.sig ficheiro.txt
```

3 PGP

O PGP (Pretty Good Privacy - http://www.pgp.com) foi criado por Phil Zimmermann em 1991. Inicialmente o PGP era freeware, mas com o passar dos anos acabou por ser comprado e a versão mais recente já é comercial. Embora continue a haver uma versão freeware do PGP esta é muito limitada (o GnuPG é uma versão freeware da GNU do PGP). Esta aplicação tem a vantagem de permitir a sua integração directa com o Outlook, não sendo necessário o utilizador dominar as opções da linha de comando como acontece com o GnuPG.

4 Exercícios

1. gerar um par de chaves
2. exportar a chave para um ficheiro, em formato binário e ascii (verifique as diferenças, como por exemplo o tamanho do ficheiro)
3. importar chaves públicas dos colegas
4. validar as chaves publicas
5. Autenticação e segurança de ficheiros:
    a. criar um ficheiro com o comando: ls /etc/* > ls.txt
    b. assinar o ficheiro ls.txt
    c. crie o hash sha-1 num ficheiro separado sha1.txt e assine o ficheiro
    d. quais as diferenças entre as operações da alínea *b* e *c*?
6. Partilha de informação
    a. Partilhe com um colega os ficheiros ls.txt, ls.txt.asc e sha1.txt
    b. valide a informação