

# Capítulo 13

**Segurança de dados em redes de computadores.**

## **Redes Privadas Virtuais (VPNs)**

---

Neste capítulo faz-se uma introdução a aplicações e standards que implementam segurança (por encriptação) na comunicação de dados em redes de computadores.

# Chaves de encriptação simétricas e assimétricas

---

- **Chave simétrica**

Uma palavra (ou frase) secreta é utilizada para codificar dados sensíveis. Este processo é reversível: a mesma palavra é utilizada para de-criptar os dados, produzindo o texto original.

O algoritmo mais utilizado ("De facto standard") para encriptação simétrica é o Data Encryption Standard (DES) [IBM 1977]

# Chaves de encriptação simétricas e assimétricas

---

- **Chave assimétrica**

O emissor codifica os dados com a chave pública (acessível livremente). O receptor descodifica os dados com a chave privada (que apenas ele conhece) [Diffie-Helman 1976]

O algoritmo mais utilizado na actualidade é o Rivest-Shamir-Adelman (RSA). É utilizado em várias aplicações: PGP (correio electrónico), SSH (secure shell), HTTPS (secure sockets layer).

A chave assimétrica traz enormes vantagens na Internet porque assegura a confidencialidade na transmissão de dados sem que o emissor precise de saber a chave privada

## Segurança ao nível das aplicações

---

- A segurança ao nível das aplicações é implementada encriptando os dados que vão ser transmitidos tendo previamente (ou simultaneamente) assegurado a identidade do receptor ou do emissor (ou de ambos).
- Vamos falar de duas aplicações e um protocolo que utilizam chaves assimétricas: PGP, SSH, e HTTPS

## Pretty Good Privacy (PGP)

---

PGP é a aplicação mais utilizada no mundo para assinar digitalmente e encriptar correio electrónico.

PGP é utilizado essencialmente com dois objectivos:

1. **encriptação.** O emissor utiliza a chave pública do destinatário para encriptar um texto. Apenas o destinatário pode descodificar o texto utilizando a sua chave privada.

## Pretty Good Privacy (PGP)

---

**2. assinatura digital.** O emissor utiliza a sua chave privada para assinar digitalmente um texto. A assinatura digital é uma sequência de caracteres que assegura a integridade do texto: é uma "checksum" realizada com o auxílio da chave privada.

O receptor está na posse da chave pública do emissor ou vai buscar esta chave a um local público na Internet (servidor de chave publicas por exemplo <http://subkeys.pgp.net>) e realiza agora com a chave pública uma "checksum" no texto recebido.

## Trust chain model

---

Como é que alguém prova que "aquela" é a sua chave pública? A chave pública associada a uma determinada conta de email (exemplo [jbastos@ualg.pt](mailto:jbastos@ualg.pt)) pertence à pessoa que a regista no servidor de chaves públicas...

No entanto, a chave pública pode ser assinada digitalmente por uma ou mais pessoas ou identidades (por exemplo Verisign).

Este é o *trust chain model* que já conhecemos da banca (confiamos mais num cheque visado onde o banco assina por cima da assinatura do dono do cheque, do que num cheque simples...)

## Configuração do GnuPG

---

A aplicação GnuPG é a versão open source da aplicação PGP (comercial.)

– Para criar o par {chave pública, chave privada} executa o comando  
`$gpg --gen-key`

Vais encontrar a chave pública e a chave privada em

`~/ .gnupg/secring.gpg`

`~/ .gnupg/pubring.gpg`

A chave privada deve manter-se secreta (não deve ser lida por ninguém, apenas pelo seu dono). A chave pública deve ser colocada num servidor de chaves públicas na Internet (por exemplo <http://subkeys.pgp.net>).

## Configuração do GnuPG (2)

---

- Podes importar para o ficheiro pubring.gpg chaves públicas de outras pessoas com o comando

```
$gpg --import chave_publica_de_outra_pessoa.gpg
```

- Para assinares digitalmente um texto com a tua chave privada executa o comando

```
$gpg --clearsign texto.txt
```

## Exemplo: texto assinado digitalmente

---

```
-----BEGIN PGP SIGNED MESSAGE-----
```

```
Hash: SHA1
```

```
exemplo de um texto com assinatura digital
```

```
-----BEGIN PGP SIGNATURE-----
```

```
Version: PGPfreeware 5.5.3i for non-commercial use
```

```
<http://www.pgpi.com>
```

```
iQA/AwUBRXcDbamRjZWNO/rjEQLzFgCeNeRejrF9byibSG34/pCeGZaBYC0AoN46  
UGEthD1XVm4JGfBujGn9PeON  
=FUH3
```

```
-----END PGP SIGNATURE-----
```

## Configuração do GnuPG (3)

---

– Para verificares a autenticidade de um texto com uma assinatura digital executa o comando

```
$gpg --verify texto.txt
```

```
$gpg --verify --keyserver wwwkeys.pgp.net --  
honor_http_proxy texto.txt
```

– Para encriptar ou de-encriptar um ficheiro utiliza o comando

```
$gpg --encrypt ficheiro
```

```
$gpg --decrypt ficheiro
```

## Secure Sockets Layer (SSL)

---

- O protocolo SSL (inventado pela Netscape) funciona num nível intermédio entre a camada de transporte (TCP) e a camada das aplicações. É utilizado para garantir transmissões seguras de dados em vários serviços. O mais popular é o serviço HTTPS (HTTP em cima de SSL).
- O cliente (browser) garante a identidade do servidor HTTP através da chave pública deste que tem na sua posse. Alternativamente, o servidor pode enviar "on the fly" (no momento) a sua chave pública ao cliente.
- O algoritmo mais utilizado é o RSA.

## Secure Sockets Layer (SSL) (2)

---

- O cliente gera uma chave secreta única (chave simetrica) para cada transacção. Encripta-a com a chave pública do servidor HTTP e envia-a para o servidor.
- A partir deste momento o cliente encripta todos dados que envia ao servidor com a chave secreta. A resposta do servidor também vem encriptada com a mesma chave secreta. O algoritmo mais utilizado para gerar esta chave simétrica é o DES.

NOTA IMPORTANTE: a chave pública do servidor tem que vir assinada digitalmente por uma entidade idónea (Certificate Authority) que garanta a sua autenticidade (exemplos Verisign, Thawte, ...).

Chaves públicas auto-assinadas (self-signed) não garantem coisa nenhuma!

## Criação de um certificado para o servidor HTTPS utilizando openssl

- vamos criar um "self-signed certificate", isto é vamos ser nós próprios a "Certificate Authority" que certifica a autenticidade da nossa chave pública...
- O comando seguinte cria o par {chave pública, chave privada} para a Certificate Authority

```
#openssl genrsa -des3 1024 > server.key
```

O ficheiro server.key contém agora o par de chaves da Certificate Authority. O algoritmo utilizado foi o RSA.

## Criação de um certificado para o servidor HTTPS utilizando openssl

– O comando seguinte gera um certificado "x509" para o servidor HTTPS assinado com a chave pública da Certificate Authority válido por 365 dias

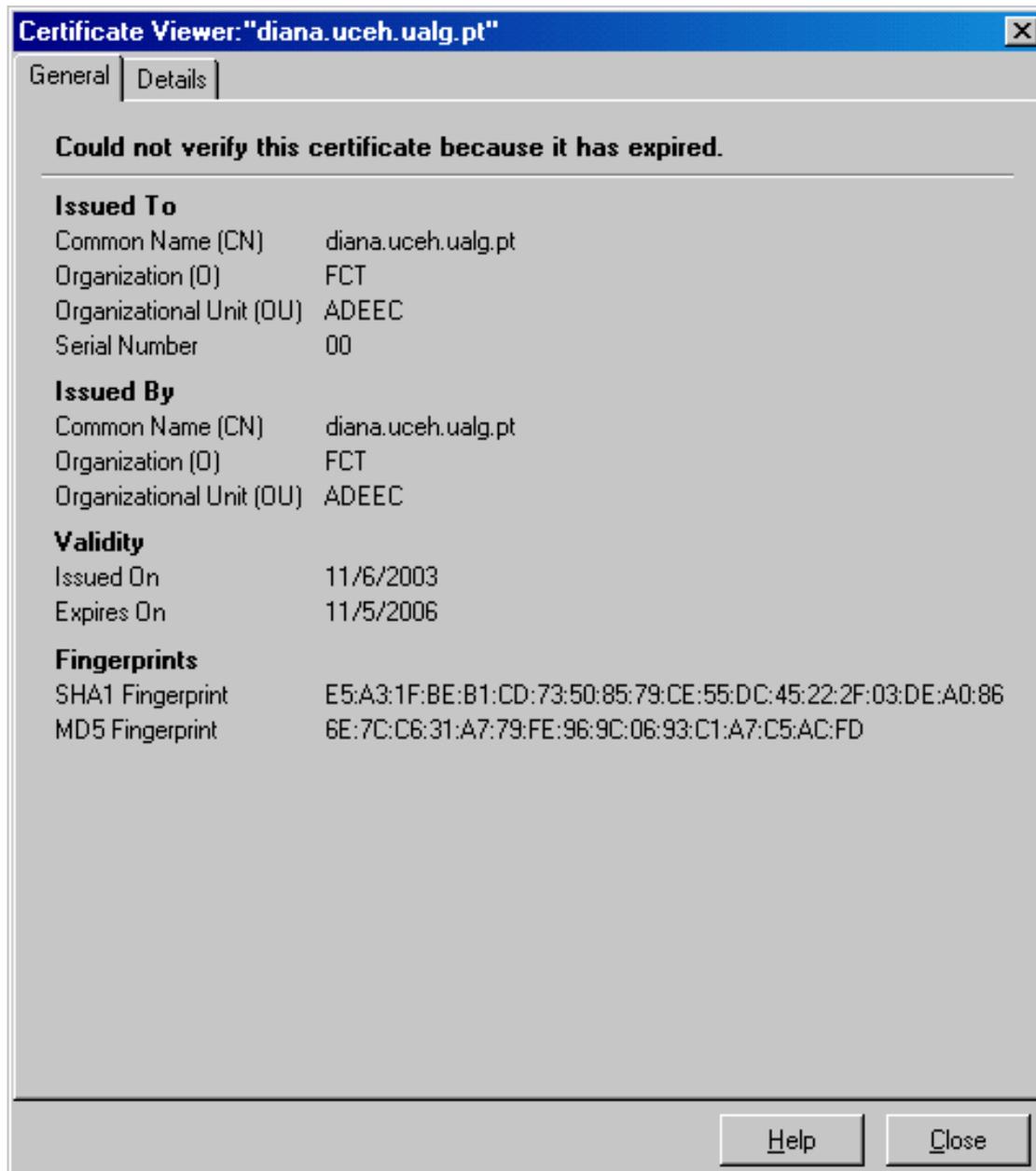
```
#openssl req -new -key server.key -x509 -days 365 -  
out server.crt
```

A chave pública do servidor HTTPS, no formato x509, encontra-se agora no ficheiro server.crt.

## Certificado x509

---

```
diana:/etc/apache/ssl.crt# cat server.crt
-----BEGIN CERTIFICATE-----
MIIDDzCCAnigAwIBAgIBADANBgkqhkiG9w0BAQQFADBpMQswCQYDVQQGEwJQVDEQ
MA4GA1UECBMHQUxHQVJWRTEENMAsgA1UEBxMERkFSTzEMMAoGA1UEChMDRkNUMQ4w
DAYDVQQLEwVBREVFQzEhMBkGA1UEAxMSZG1hbmEudWNlaC51YWxnLnB0MB4XDTAz
MTEwNjE4MjExNFoXDTA2MTEwNTE4MjExNFowaTELMAkGA1UEBhMCUFQxEDAQOBgNV
BAGTB0FMR0FSVkuXDTALBgNVBAcTBEZBUk8xDDAKBgNVBAoTA0ZDVDEOMAwGA1UE
CxMFQURFRUMxGzAZBgNVBAMTEmRyYXZlLnVjZWgudWFsZy5wdDCBnzANBgkqhkiG
9w0BAQEFAAOBjQAwGyKCyYEA04drgfh50AAfMFODcu8uo9h56JX/skdxE6pLfyyI
5K9h736XQi4MwBFslthYq+HkkYR3XvLZsv6v8GmShDG5rqWK4v8GPGfn8VIVMfLj
m0PpTh69qOrAL7tipCq/gXvIGHjDEqIoIKYgS7bJr8Ey/z39MtEK2hwl2ppyU0g/
XQsCAwEAAaOBxjCBwzAdBgNVHQ4EFgQUUGYeQGQZQjk+bqfTzd/bwnIw8kqQwgZMG
A1UdIwSBizCBiIAUGYeQGQZQjk+bqfTzd/bwnIw8kqShbaRrMGkxCzAJBgNVBAYT
AlBUMRAwDgYDVQQIEwdbTEdBULZFMQ0wCwYDVQQHEwRGQVJPMQwwCgYDVQQKEwNG
Q1QxDjAMBgNVBAsTBUFERUVDMRswGQYDVQQDEwJkaWFuYS51Y2VoLnVhbGcucHSC
AQAwDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQQFAAOBgQCkBNL/fzt/atvGEIKM
0HhbTqPZVz+thqJBBMAKXqvVSi5b62nQ2pOTY5y6pvlENCMXbVAVJ3CepB+CTDli
/iBpa5B/WFNsOo5x01vCZ5nRe82D7g/UPtE2vzjFmyalhseHvrF2+QKNmBlTDO/J
57ki1R0djWkTcSzSJHop1SoenA==
-----END CERTIFICATE-----
```



## Secure Shell (SSH)

---

O serviço SSH proporciona uma alternativa segura (encriptação) aos serviços clássicos telnet e ftp.

**O Serviço SSH funciona de uma forma em tudo semelhante aos serviços que utilizam o SSL:**

- O cliente liga-se à porta 22 do servidor e solicita a chave pública deste. Também pode (opcional) enviar ao servidor a sua chave pública.
- O cliente gera uma chave secreta válida para a sessão (chave simétrica), encripta-a com a chave pública do servidor e envia-a ao servidor
- O servidor de-encripta a chave secreta, encripta-a novamente com a chave pública do cliente e envia-a ao cliente para confirmação
- Autenticados mutuamente, cliente e servidor trocam entre si dados encriptados com a chave secreta válida para a sessão.

## Configuração de SSH (1)

---

– As opções de configuração do servidor de SSH e do cliente de SSH encontram-se em

```
/etc/ssh/sshd_config  
/etc/ssh/ssh_config
```

Na fase de instalação do servidor, este gera automaticamente o par de chaves do servidor em

```
/etc/ssh/ssh_host_key  
/etc/ssh/ssh_host_key.pub
```

## Configuração de SSH (2)

---

- O par de chaves do cliente é criado pelo próprio utilizador com o comando

```
$ssh-keygen
```

e encontram-se em

```
~/.ssh/identity
```

```
~/.ssh/identity.pub
```

## Redes Privadas Virtuais (VPNs)

---

Até agora vimos **aplicações** e **serviços** que encriptam os dados que são transmitidos.

Numa rede privada virtual (VPN) **é todo o pacote IP que é encriptado**

**Pacote original (dentro da rede privada)**



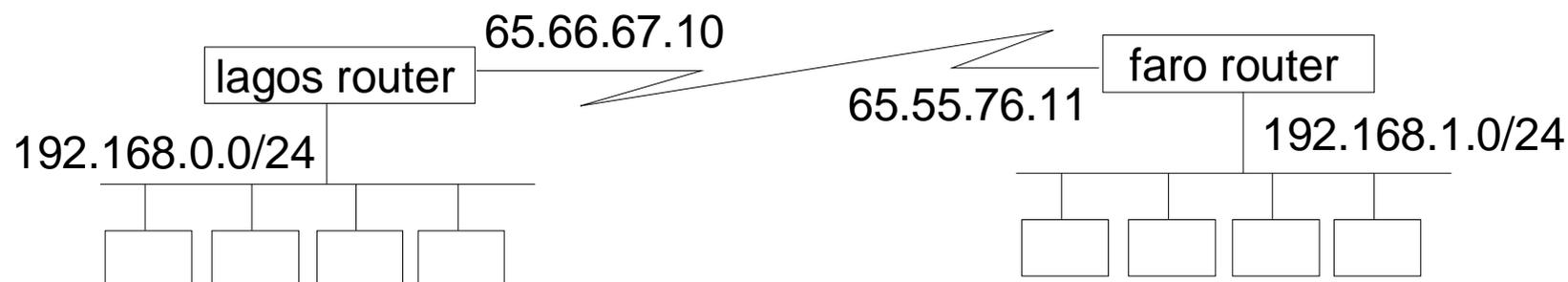
**Pacote na Internet**



## Configuração de uma rede privada virtual (VPN) com o software FreeSWAN

---

Como exemplo prático vamos supor que temos duas agências da mesma empresa, uma em Faro e outra em Lagos, que comunicam entre si através de uma rede privada virtual:



## Configuração de uma rede privada virtual (VPN) com o software FreeSWAN (2)

---

– Os routers tem que estar configurados para fazer o encaminhamento de pacotes

```
#echo 1 > /proc/sys/net/ipv4/forward
```

e as suas tabelas de routing estão actualizadas com a informação como chegar à rede privada no outro lado do "túnel". Por exemplo no lagos router:

```
lagos#route add -net 192.168.1.0 netmask  
255.255.255.0 gw 65.55.76.11
```

## Configuração de uma rede privada virtual (VPN) com o software FreeSWAN (3)

---

– Os routers configuram a rede VPN através de dois ficheiros

1) `/etc/ipsec.secrets`.

Este ficheiro contem a chave pública do router no outro extremo do túnel ou uma chave secreta previamente partilhada.

A chave simétrica pode ser criada com o comando

```
#ipsec ranbits 254
```

O par {chave pública, chave privada) pode ser criado com o comando

```
#ipsec rsasigkey 2048
```

## Configuração de uma rede privada virtual (VPN) com o software FreeSWAN (4)

---

2) `/etc/ipsec.conf`.

Este ficheiro contem as configurações específicas da ligação:

- identificação das interfaces
- identificação das redes privadas
- identificação dos routers
- método de autenticação