Nome:_____ Nº:_____
Nome:_____ Nº:_____

Ficha Prática nº 3

Conceitos de Sistemas Informáticos: 2005/06

Ligue a sua máquina e introduza o login **diguest** e a senha **diguest**. Escolha a opção **Linux fedora 4 <net>** e espere que o sistema operativo seja carregado. Entre no login **diguest** com a senha **diguest** e depois faça **startx** para activar o ambiente gráfico X Window.

A seguir são apresentadas uma série de questões. Uma vez respondidas, o aluno deverá devolver a ficha ao docente da disciplina a fim de ser avaliada.

O GnuPG (ou o comercial PGP) é um conhecido pacote de *software* que permite que duas entidades comuniquem entre si por correio electrónico de forma segura, através de criptografia de chave assimétrica (ou pública). Isto significa que:

- a mensagem chegou ao destino inalterada
- a mensagem apenas pode ser lido pelo destinatário desejado
- o destinatário pode verificar a autenticidade do remetente

Geração de um par de chaves (pública e privada)

O PGP oferece várias opções para gerar uma chave assimétrica, tais como o algoritmo e o comprimento da chave. Uma chave de 1024 bits é suficiente para usos gerais, como por exemplo, enviar um mail assinado digitalmente. Uma chave de 2048 bits é recomendada para situações mais críticas, como, por exemplo, enviar a chave que vai cifrar simetricamente a mensagem. Quanto maior é o comprimento da chave, maior é a segurança oferecido pelo sistema criptográfico; em contrapartida mais morosos se tornam os processos de cifragem e decifragem.

Execute na sua directoria-base:
(na primeira vez que é executado, algumas versões do gpg criam apenas a directoria .gnupg, na directoria base do utilizador, onde residirão todos os ficheiros necessários ao GPG. Neste caso torna-se necessário correr o comando uma segunda vez para aí colocar os ficheiros com as chaves geradas.)

gpg --gen-key

e siga os passos apresentados, aceitando as opções indicadas por defeito. Faça **Real name: diguest**.
(Para alimentar o gerador de números aleatórios convêm mover o cursor do rato durante a fase da criação das chaves).

Numa determinada fase deste processo é pedida uma boa *passphrase*, que serve para cifrar a chave privada a guardar no disco do seu PC. Porque razão este procedimento é muito recomendável para a chave privada, mas dispensável para a chave pública?

Uma vez gerada a chave assimétrica, vamos exportá-la para o ficheiro key.txt.
Liste primeiramente as chaves criadas, alojadas na *keyring* pubring.gpg. (**Nota**: a chave DSA é usada apenas para as assinaturas e a chave subordinada El Gammal é usada para a encriptação)

gpg --list-keys

e depois exporte a sua chave (neste caso o *key_ID* é *diguest*):
gpg --armor --export key_ID > /tmp/key.txt

Diga qual a componente da chave (pública ou privada) que é exportada, e qual o interesse em se realizar esta operação?

Faça *su* – e depois, como root (senha diguest), crie uma conta no seu PC através dos seguintes comandos: *adduser newguest*, atribua-lhe uma senha usando o comando *passwd newguest*, Active o serviço de email com *service sendmail start*. Finalize com *exit*.

Abra um novo terminal X no seu ambiente de trabalho e faça: *su -l newguest*
Repetindo os passos já descritos, crie uma chave assimétrica para newguest (Real name: newguest).
Mantenha este terminal e o do diguest abertos até ao final deste módulo.

Assinatura de uma mensagem

A operação mais comum será provavelmente assinar mensagens, a fim do emissor se autenticar perante o destinatário, e o emissor não poder repudiar a sua origem (uma chave pode também ser assinada, antes de ser distribuída, para certificar a sua proveniência).

No terminal do diguest, crie um pequeno ficheiro de texto (mensagem.txt) com um editor à sua escolha, e depois execute o seguinte comando (esta operação usa a *default key*):

gpg --clearsign < mensagem.txt > assinado.txt

Observe o ficheiro assinado.txt e descreva sucintamente as operações efectuadas sobre a mensagem que conduziram à sua assinatura.

Mande por email o ficheiro assinado.txt ao utilizador newguest (como o destinatário encontra-se na mesma máquina que o remetente da mensagem, basta o login como endereço de mail)

mail newguest < assinado.txt

No terminal do newguest, guarde a mensagem transmitida usando a opção **w** do mail, a fim de gravá-la sem o cabeçalho do MAIL (o que acontece se for gravada com a opção **s**):

***w** [nº mesg] signed.txt*

e depois abandone a aplicação mail digitando **q**

Antes de verificar a assinatura é preciso primeiramente possuir a chave pública do remetente. Obtenha o ficheiro /tmp/key.txt e faça:

gpg --import /tmp/key.txt

Verifique, com *gpg -- list-keys*, se a chave foi correctamente importada para a *keyring* das chaves públicas do newguest. Seguidamente execute:

gpg --edit-key diguest

Auxiliando-se do man gpg (ver opções de --edit-key) identifique os diferentes níveis de confiança que se poderão atribuir a uma chave pública. Diga ainda quais os níveis de confiança (*trust*) que newguest atribuiu à chave pública do diguest e à sua própria chave pública?

Verifique a assinatura fazendo:

gpg --verify signed.txt

Tendo em conta o que observou na alínea anterior, interprete a informação apresentada no terminal.

Gestão de chaves públicas

O PGP adoptou um modelo de confiança de chaves públicas designado *web of trust*. Contrariamente ao S/MIME, não existe aqui uma autoridade certificadora central. Cada utilizador certifica, com um determinado nível de confiança, as chaves públicas dos outros, assinando-as com a sua chave privada. Estes certificados são armazenados juntamente com as chaves nas *keyrings*. Caberá depois ao utilizador final aceitar, ou não, a chave pública com o nível de confiança que lhe foi associada.

O utilizador newguest possui uma chave pública que supostamente pertence a diguest.
A fim de se certificar disto, o newguest pode comparar a *fingerprint* da chave importada com a da chave original. Caso sejam iguais, então o newguest tem realmente a chave pública do diguest e pode certificá-la assinando-a com a sua chave privada.
Para verificar as *fingerprints* faça em ambos terminais:

gpg --fingerprint diguest ou *gpg --edit-key diguest fpr*

Se forem iguais, newguest pode agora autenticar a chave importada, assinando-a com a sua chave privada:

gpg --edit-key diguest sign (responda com y, *passphrase* correcta, *save*)
Verifique novamente a assinatura. Qual a diferença fundamental entre a informação agora apresentada acerca da validade da assinatura e a que foi mostrada na verificação anterior. Descreva sumariamente as operações ocorridas na validação desta assinatura.

Experimente alterar uma letra do texto da mensagem presente em *signed.txt* e depois volte a verificar a validade da assinatura. O que pode concluir sobre as garantias oferecidas pela colocação de uma assinatura digital num documento informático.

Envelope digital

Para além da assinatura, pode também ser conveniente enviar a mensagem cifrada (simetricamente). Chama-se envelope digital ao conjunto enviado formado pela mensagem cifrada e pela chave de sessão simétrica, esta cifrada assimetricamente.

Como diguest, experimente criar um envelope digital e enviá-lo por mail ao newguest. Para tal *deve primeiramente importar as chaves do newguest* e depois fazer:

gpg -r recipientID --armor --sign --encrypt < mensagem.txt > cifrado.txt

Diga porque é necessário especificar o parâmetro *recipientID*, e que tipo de chave ele representa/identifica?

Compare o conteúdo de cifrado.txt com o de assinatura.txt. Apesar de ambos estarem assinados, diga porque razão não observa, em cifrado.txt, a parte respeitante à assinatura (sugestão: consulte o anexo).

Faça agora:

gpg -r recipientID --armor --sign --encrypt < mensagem.txt > encriptado.txt

Compare encriptado.txt com cifrado.txt (*diff encriptado.txt cifrado.txt*) e, sabendo que a mensagem cifrada é a mesma, diga porque razão são diferentes os seus conteúdos (sugestão: consulte o anexo).

Envie por *mail*, ao newguest, o envelope criado e depois decifre e verifique automaticamente o documento recebido fazendo *gpg --decrypt cifrado.txt*. Qual o tipo de informação apresentado após esta operação? Teria sido possível ao diguest realizar esta operação? Porquê?

O argumento *armor*, usado ao longo deste módulo, permite obter ficheiros de texto contendo apenas caracteres com código ASCII inferior a 128. Que vantagem daí advém no envio desses ficheiros por mail?

ANEXO

2.2 Confidentiality (*extraído do RFC 1991*)

PGP provides confidentiality by encrypting messages to be transmitted or data files to be stored locally using conventional encryption. In PGP, each conventional key is used only once. That is, a new key is generated as a random 128-bit number for each message. Since it is to be used only once, the session key is bound to the message and transmitted with it. To protect the key, it is encrypted with the receiver's public key. The sequence is as follows:

-the sender creates a message
-the sending PGP generates a random number to be used as a session key for this message only
-the sending PGP encrypts the message using the session key
-the session key is encrypted using the recipient's public key and prepended to the encrypted message
-the receiving PGP decrypts the session key using the recipient's private key
-the receiving PGP decrypts the message using the session key

Both digital signature and confidentiality services may be applied to the same message. First, a signature is generated for the message and prepended to the message. Then, the message plus signature is encrypted using a conventional session key. Finally, the session key is encrypted using public-key encryption and prepended to the encrypted block.