

TP: Chaves de cifra, certificados e o PGP

Índice

Notas iniciais	1
Antes de começar	2
Objectivos	4
Exercícios	4
Gestão de chaves	4
Opção PGP (pelo menos um dos alunos deve seguir esta alternativa)	4
Opção X509 (pelo menos um dos alunos deve seguir esta alternativa)	6
Enviar e receber mensagens seguras	9
Proteger documentos locais	12

Notas iniciais

1. Os conceitos apresentados nesta ficha de trabalho, assim como os exercícios propostos, complementam o material facultado na componente teórica e não deve ser utilizado sem um claro entendimento dos conteúdos aí apresentados. **Para efeito da apresentação de resultados, deverá registar as suas observações num *logbook*, no mínimo referentes às tarefas assinalados a vermelho.** A avaliação será feita com base no *logbook* e por isso o mesmo deve ser claro e objetivo.
2. Na realização deste trabalho necessita apenas de um computador pessoal com ligação à Internet, uma implementação do OpenPGP, como seja o PGP¹ (ou o GnuPG, ou ainda o GPG4Win), em alternativa/complemento um gestor de certificados X509 (já incluído na maioria dos Sistemas Operativos, ou em aplicações, como o browser Firefox e o cliente de e-mail Thunderbird) e um cliente de e-mail devidamente instalado. A instalação do software é uma tarefa simples, não sendo aprofundada nas tarefas propostas nesta ficha.
NOTA: Existe uma grande semelhança ao **nível funcional** entre a última versão pública do PGP (versão 8.0.2), a versão *trial* do PGP (agora fornecida pela Symantec, mas com a larga maioria das funções desabilitadas, a não ser que seja introduzida uma licença válida 😊) e o

¹ O PGP (Pretty Good Privacy) foi criado por [Philip Zimmermann](#), tendo sido publicado na Internet, pela primeira vez, em 1991. Depois de várias evoluções o PGP acabou por ser adquirido pela Symantec. Deixou de ser um produto de código aberto, mas vários projetos paralelos continuam a suportar a implementação dos mesmos protocolos e técnicas, sendo de realçar duas iniciativas em particular: OpenPGP e GnuPG.

GnuPG, apesar de, naturalmente, os respetivos interfaces serem diferentes; este tutorial baseia-se na utilização do PGP 10.2.0, a executar em ambiente Windows, mas é perfeitamente possível (e até mesmo aconselhável) utilizar o GnuPG. Neste caso será necessário ter o cuidado de adaptar as instruções ao interface do GnuPG.


As diferentes versões **freeware do OpenPGP** estão disponíveis neste [link](#)

A versão *trial* do **PGP** está disponível no site da Symantec - estamos interessados no “PGP Desktop E-mail”, sendo necessário fazer um registo para descarregar o mesmo; as limitações funcionais da versão *trial* não comprometem a realização deste trabalho.

O **Kleopatra** é um exemplo de um gestor de certificados que satisfaz todos os requisitos do trabalho e que poderá usar, em vez do PGP.


O **GnuPG** está disponível em <http://www.gnupg.org/>

O **GPG4Win** está disponível em <http://www.gpg4win.org/>


3. No contexto das definições e trabalhos seguintes, o conceito de “chave” e de “certificado” associado não estão claramente diferenciados, por ser esse o entendimento que a documentação do PGP assume. No entanto, quando nos referirmos ao X509, a diferença é relevante, como veremos.
4. A utilização do símbolo  denota afirmações que podem fazer parte de uma política de segurança.

Antes de começar


1. Cada utilizador deve dispor, pelo menos, de um “**par de chaves**” (essa será a primeira tarefa que lhe irá ser solicitada, mais à frente). Este par de chaves é composto por uma **chave pública** (a disponibilizar publicamente) e uma **chave privada** (a guardar cuidadosamente). No caso do OpenPGP, localmente existem dois importantes ficheiros, designados por **keyrings**. Um deles – `pubring.pkr` – armazena todas as chaves públicas de utilizadores para quem pretende enviar mensagens de forma segura, o outro – `secring.skr` – armazena a(s) chave(s) privada(s). Estes ficheiros estão armazenados de uma forma cifrada, no espaço de trabalho do utilizador, (...)\Documents\PGP, no Windows). Mas tenha em atenção que outras aplicações podem implementar este armazenamento de forma diferente, sendo de esperar que, em todas elas, exista um mecanismo de importação/exportação, que permite facilmente transferir um par de chaves de uma para outra.

 A utilização de mais do que um par de chaves justifica-se quando se pretende utilizar mais do que uma assinatura - podemos querer utilizar uma assinatura pessoal e uma outra assinatura institucional, apenas com alguns elementos de identificação comuns.


2. Ao gerar um par de chaves, na realidade poderá estar a associar várias chaves: uma chave privada ("Mestra") para assinaturas; uma segunda chave, ou melhor uma **subchave, para cifrar**; e uma ou mais subchaves adicionais, para assinar, cifrar ou assinar/cifrar, que poderão ser revogadas, sem comprometer a chave "Mestra".

 Este procedimento permite, por exemplo, manter as assinaturas de chaves públicas válidas durante um largo período de tempo e modificar a subchave de cifra e/ou de assinatura de documentos regularmente (talvez períodos de um ano). Este pode ser um procedimento de segurança muito útil.

3. **Additional Decryption Keys (ADKs)** são chaves de decifra adicionais, que permitirão aos responsáveis da segurança de uma organização decifrar mensagens cifradas para a chave pública associada à ADK. Na prática, é uma segunda chave que pode decifrar.

 Em princípio, estas chaves só serão utilizadas em casos de extrema necessidade, que devem ser especificados cuidadosamente, sob pena de estar a criar uma vulnerabilidade muito relevante!

4. *Corporate Signing Key* é uma chave pública atribuída a uma organização e na qual todos os utilizadores, relacionados de alguma forma com essa organização, podem confiar.

 Todas as chaves assinadas pela correspondente **Corporate Key** (chave privada) podem ser assumidas como válidas – enquanto as não assinadas devem ser assumidas com bastante precaução.

5. Validação de uma chave: quando importamos uma cópia de uma chave pública de “alguém”, podemos adicioná-la ao nosso *keyring* (ou infraestrutura equivalente). Mas antes de a usar para cifrar alguma mensagem temos que proceder à respetiva **validação** (determinar se a identificação existente na chave pública corresponde à pessoa física com quem nos queremos relacionar):
- Se a chave foi entregue pessoalmente, é válida;
 - Se foi entregue por e-mail, ou obtida a partir de um servidor, ou mesmo uma CA:
 - Se vem assinada por alguém ou alguma entidade em quem confiamos, é válida;
 - Caso contrário descartamo-la, ou contactamos a pessoa física para pedir-lhe a “impressão digital” da sua chave pública, comparando-a com a que consta na chave que temos em nosso poder, como mostra a Figura 1. Se a verificação tiver sucesso a chave é válida.
 - O resultado do processo de validação deve ser registado na chave pública alvo (se for possível); se for um certificado PGP, deve ainda ser enviado para uma base de dados por forma a dar conhecimento do nosso “parecer” (esta é a essência do modelo *Web of Trust*);
 - Em caso de dúvidas sobre a validação, deve ser registada a indicação “Marginalmente Válida” e em caso de não validação deve ser registada a indicação “Inválida”.

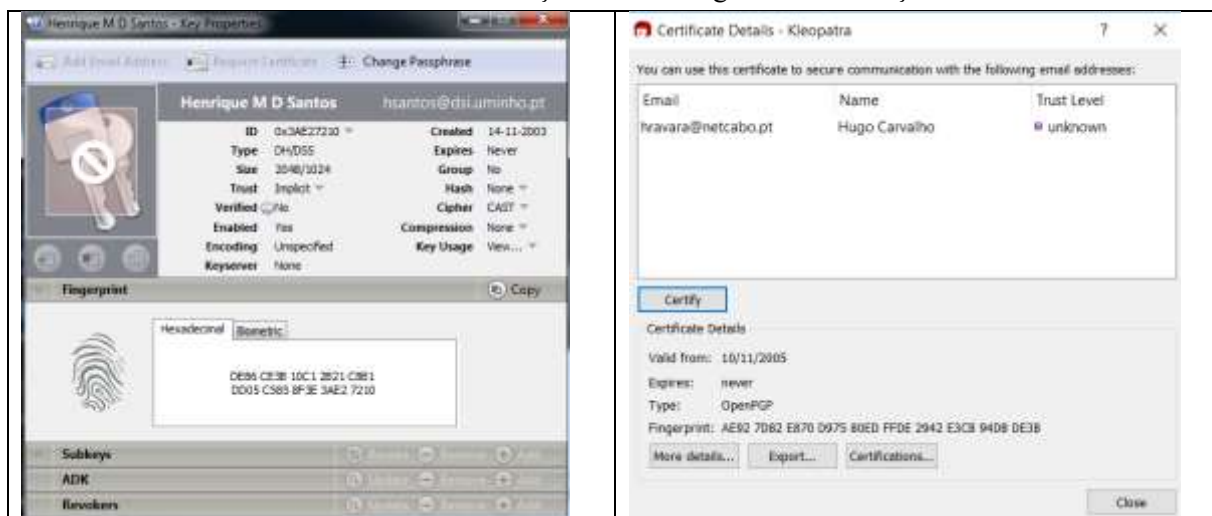


Figura 1 – Verificação da *fingerprint* nas propriedades de uma chave, usando o PGP (à esquerda) e o Kleopatra (à direita)

6. No caso do PGP, todas as operações sobre chaves são executadas, naturalmente, a partir do utilitário **PGP Desktop** – informação mais detalhada sobre este utilitário pode ser obtida através do (excelente!) *help* do PGP.
7. O OpenSSL (<https://www.openssl.org>) é um projeto de domínio público, bastante divulgado e utilizado por muitos produtos de criptografia aplicada, que implementa uma biblioteca de funções de cifra (à semelhança do GnuPG). No entanto, o OpenPGP e o OpenSSL implementam normas diferentes e não diretamente compatíveis (respetivamente, os

certificados PGP e os certificados X509). São claras as opiniões diferentes no que respeita ao desempenho de ambas as soluções!

Objetivos


1. Descrever a forma como o conceito de chave pública é tipicamente implementado.
2. Reconhecer as operações associadas à gestão das chaves públicas e privadas.
3. Desenvolver competências na utilização de ferramentas de gestão de certificados.
4. Utilizar uma *framework* de criptografia para enviar e receber mensagens de e-mail, com segurança.

Exercícios

Gestão de chaves

Opção PGP (pelo menos um dos alunos deve seguir esta alternativa)

1. Execute o PGP Desktop e siga os procedimentos de instalação indicados (apenas na primeira execução). Como foi referido, em alternativa pode usar o OpenSSL, ou ainda o gestor de certificados Kleopatra (em Linux ou em Windows, neste caso incluído no projeto GP4Win) – e que lhe permite gerir simultaneamente certificados PGP e x509, mas com algumas restrições ☺.
2. Crie uma nova chave (File -> New PGP key...). Caso esta seja a primeira vez que executa o PGP, o *wizard* de criação de uma nova chave deverá aparecer imediatamente, não sendo necessário a utilização de menus.
3. Ao caracterizar o seu par de chaves, a opção *Advanced* permite escolher o tipo de algoritmo e o tamanho de chave. **Escolha o tipo de chave RSA, com 1024/2048 bits. Não selecione nenhuma data limite de validade e mantenha para a cifra a lista de algoritmos, incluindo o AES, como preferido.**
4. Ser-lhe-á então solicitada uma **Passphrase**, a qual necessitará de introduzir sempre que utilizar a sua chave privada (assinaturas ou decifrar ficheiros). À medida que vai escrevendo, a “qualidade” da sua “frase passe” aparece medida por uma barra. Escolha uma frase de fácil memorização e que maximize o tamanho da barra.
5. De seguida poderá ser-lhe dada a opção de publicar a sua chave pública num servidor (apenas se estiver a criar um certificado PGP), ao mesmo tempo que é feita uma verificação do endereço de e-mail. Para já salte essa verificação, pois mais tarde teremos oportunidade para publicar a chave.
6. De volta à janela principal, selecione a chave que acabou de criar. Essa chave está ligada a uma auto assinatura, a qual é essencial para poder exportar a sua chave (ver Figura 2). Verifique as propriedades da assinatura e da chave, prestando particular atenção à *fingerprint* associada à chave – a *fingerprint* pode ser visualizada em dois modos: hex; e no modo texto. Experimente os dois modos. **Copie os atributos mais relevantes da assinatura e da chave, bem como da fingerprint desta última, para o seu logbook. Esta chave é a sua chave privada ou a pública? Que elementos ligam a assinatura com a sua chave privada?**

 Nota: muitas pessoas incluem no seu cartão-de-visita este *fingerprint*, para que qualquer utilizador que detenha esse cartão possa validar a chave pública associada (ou invalidar uma falsa cópia 😊)

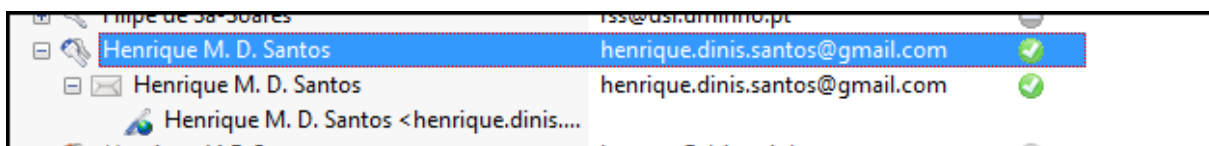


Figura 2 – Chave e correspondente auto assinatura

7. Ainda na janela que mostra as propriedades da chave, selecione o separador Subkeys². Poderá verificar que, eventualmente, a sua chave tem duas subchaves, uma para cifra e outra para assinatura. Contudo, tal como foi inicialmente referido, pode criar mais subchaves (a Figura 3 ilustra uma situação em que existem duas subchaves, de tamanhos diferentes e com funções diferentes). Crie uma ou mais subchaves. **Essas subchaves são chaves públicas ou privadas? Qual é a relação destas com a chave *master*? Qual a utilidade de ter várias subchaves associadas a uma mesma chave?**



Figura 3 – Chave com duas subchaves

8. Tem agora a possibilidade de solicitar a uma CA (*Certification Authority*) a emissão de um certificado com este par de chaves, o que lhe permitira obter um certificado X.509 - na mesma janela que mostra as propriedades, existe um controlo que lhe permite gerar esse pedido, naturalmente juntando informação adicional à chave pública. Se a aplicação que estiver a utilizar tiver essa função, experimente-a, mas não é necessário consumir o pedido.
9. O passo seguinte consiste na exportação da sua chave pública. Para isso é necessário primeiro configurar a aplicação para o servidor PGP desejado (esta operação pode ser feita de várias formas e aqui iremos ilustrar apenas uma delas). Selecione o submenu *Edit Keyserver...*, do menu *Tools*. Poderá verificar que, por defeito, o PGP já está configurado para utilizar um repositório global suportado pela organização que fornece o OpenPGP (o repositório encontra-se em keyserver.pgp.com, usando o protocolo LDAP). Deverá, no entanto, adicionar um novo servidor, que irá utilizar o protocolo **PGP Keyserver HTTP**, o nome do servidor é pgpkeys.mit.edu e a porta a utilizar será a 11371 (a partir daquele mesmo servidor poderá procurar chaves diretamente num *browser*).
10. De regresso ao utilitário PGP Desktop, seleccione a sua chave e, através do menu de contexto (botão direito do rato) seleccione *Send to > http://pgp.mit.edu* e assim enviará a sua chave para o servidor (não se preocupe, pois o PGP apenas envia a chave pública ☺).

² Pode ser necessário procurar mais informação acerca da utilização de *sub keys* (um bom ponto de partida é: <http://mareichelt.de/pub/notmine/subkeys.html>)


Aproveite ainda para, a partir do menu **Tools** ou do menu no painel esquerdo, seleccionar a função **Search for Keys**, o que lhe permite fazer uma pesquisa ao servidor – poderá aí encontrar uma chave pública de alguém que conheça e proceder à sua assinatura, como bom utilizador da *web de confiança* que acabou de integrar – ainda se recorda do que significa esse conceito e a sua importância para a comunidade de utilizadores do PGP? ☺

Utilizando o site *web* referido no passo 8, procure por chaves públicas através de um endereço de e-mail (hsantos@dsi.uminho.pt) e depois por nome (i.e., Henrique Santos). Comente o resultado obtido.

11. Uma forma alternativa de entregar a sua chave pública a alguém é enviá-la por e-mail. Para isso utilize a técnica “*drag-and-drop*” da janela do gestor de chaves do PGP, diretamente para a uma mensagem de e-mail. O PGP está perfeitamente integrado com vários clientes de e-mail e permite o envio e receção de chaves públicas como ficheiros incluídos em mensagens ou como blocos de mensagens cifradas, devidamente enquadradas em *tags* específicas, que o PGP reconhece. Ao receber a mensagem, o recetor é confrontado automaticamente com a opção de importar a chave pública para o seu *keyring*, se ainda a não tiver, naturalmente – **faça o envio por e-mail de uma chave com o colega de grupo e descreva o processo usado. Após a receção deverá fazer mais alguma operação, antes de utilizar a chave do colega?**

Nota: não se esqueça da possibilidade de assinar a mensagem, para que o receptor possa validar a sua chave pública verificando a assinatura da mensagem!

12. Está pronto para se relacionar com outros utilizadores do PGP através de mensagens “seguras e autenticas” de e-mail e proteger, por cifra e assinatura digital, informação crítica que tenha no seu computador.

 Antes de terminar, talvez seja boa ideia fazer um backup das suas chaves públicas e privadas 😊. No primeiro caso até para facilitar a comunicação segura (e válida) das chaves públicas, no segundo caso... apenas por precaução.

O PGP Desktop permite executar várias outras operações que não foram necessárias no contexto deste exercício. No entanto, após assimilado o princípio de funcionamento das PKI e em particular do PGP, não será difícil explorar completamente o potencial desta ferramenta. Por razões óbvias, um dos comandos que não foi exercitado e que tem um papel muito relevante na coerência do *web de confiança*, é o **Keys -> Revoke...** Outro conceito que pode vir a ser muito útil é o de “grupos de utilizadores”.

Opção X509 (pelo menos um dos alunos deve seguir esta alternativa)

1. Instale o OpenSSL (por defeito, a maioria das imagens do bem conhecido Ubuntu, ou relacionadas, já trazem este pacote de software instalado). Poderá utilizar qualquer outro *software* alternativo que tenha a capacidade de gerar um par de chaves, pública e privada, (como o Kleopatra, já referido) mas no resto deste exercício assume-se que está a utilizar o OpenSSL. Caso use uma alternativa, deverá adaptar os comandos/ações para o seu ambiente.
2. Crie um novo par de chaves (cada aluno deve criar uma pare de chaves). Se usar o OpenSSL e quiser obter um par de chaves RSA, deverá executar o comando (ou alguma variante)
`openssl genrsa -out privkey.pem 2048`
o qual irá criar uma chave privada e a associada chave pública, de 2048 bits, ambas guardadas no mesmo ficheiro, do tipo PEM³. A chave assim obtida é adequada para poder cifrar e assinar

³ De uma forma resumida, os ficheiros PEM contém informação binária codificada em ASCII, o que facilita o seu manuseamento (copy/paste) com ferramentas simples de processamento de texto.

e não necessita de palavra-passe para ser utilizada, o que no contexto da geração de certificados, que podem ser manuseados por servidores, é uma boa opção – mais sobre este assunto pode ser consultado na documentação do OpenSSL⁴.

Verifique o bom estado da sua chave privada com o comando

`openssl rsa -in privkey.pem -check`

e registe a resposta do comando, que inclui o “texto” com a sua nova chave privada.

3. Com vista à integração numa PKI, de seguida deverá preparar um ficheiro com um pedido de certificado. Este pedido incluirá a sua chave pública, associada à chave privada anteriormente gerada, alguma informação pessoal e organizacional (a maior parte com carácter opcional) e alguns atributos, entre eles o *Common Name* (CN) e o endereço de e-mail (aspetos de identificação particularmente importantes, como é óbvio), que serão também incluídos no seu certificado. Será este ficheiro que irá enviar para a **Autoridade Certificadora (CA)**, a qual, (supostamente) depois de validar a sua identidade, devolverá o certificado assinado por ela. No OpenSSL pode gerar o pedido de certificado usando o comando:

`openssl req -new -key privkey.pem -out cert.csr`

(ajustando devidamente os nomes dos ficheiros, conforme a sua escolha), que lhe irá gerar um pedido com o formato **PKCS#10**, um *standard* que a grande maioria das CA aceita.

Verifique o estado do seu ficheiro de pedido de certificado com o comando

`openssl req -text -noout -verify -in cert.csr`

e registe a resposta do comando, que inclui os atributos do seu pedido e do futuro certificado.

4. O OpenSSL permite-lhe ainda gerar um certificado auto assinado usando a sua chave privada. Isso é necessário num cenário idêntico ao criado pelo PGP, a *Web of Trust*, no qual não é necessário (nem desejável) ter uma entidade de topo a assinar o seu certificado, mas é também a alternativa para o certificado de topo de uma CA! Adicionalmente, um certificado auto assinado é também frequentemente necessário para importar a sua chave privada para aplicações/ambientes específicos. Para gerar o certificado auto assinado pode usar o comando `openssl x509 -req -in cert.csr -signkey privkey.pem -out privcert.crt`

O certificado assim obtido é válido por um ano (a opção `days` pode ser usada para criar certificados com durabilidade diferente).

Verifique o estado do seu ficheiro com o certificado auto assinado, com o comando

`openssl x509 -text -in privcert.crt`

e registe a resposta do comando; procure identificar os elementos que considera mais relevantes.

Os passos 2 a 4 devem ser executados por cada um dos elementos do grupo, para que cada um fique com um par de chaves e um pedido de certificado, em seu poder.

5. O passo seguinte consiste em pedir o certificado público, devidamente assinado por uma CA. Neste caso vamos usar uma **CA fictícia** preparada especificamente para o exercício. O OpenSSL inclui todas as ferramentas necessárias para criar uma CA, sendo fortemente encorajado a tentar executar tal tarefa.

Existem alguns tutoriais muito úteis a explicar detalhadamente como usar o OpenSSL para implantar uma PKI simples, como aquele que está disponível [neste link](#). Focada na simplicidade, essa proposta de solução não contempla um detalhe relevante, que é o suporte da operação de verificação de revogação de certificado por meio do OCSP⁵. No entanto, também

⁴ <https://www.openssl.org/docs/HOWTO/keys.txt>

⁵ O OCSP (*Online Certificate Status Protocol*) é uma alternativa ao uso do tradicional mecanismo designado por CRL (*Certification Revocation List*). Com o este último, um cliente descarrega uma

é possível encontrar descrições simples de como implementar esse serviço (ainda usando o OpenSSL), como a que está disponível [neste link](#). Como segunda alternativa, podemos escolher uma plataforma (mais) pronta para utilizar, como o [XCA](#) ou o [OpenCA](#), ou mesmo um esquema interessante para executar uma [CA dentro de um container](#). Ainda outra alternativa consiste em usar uma solução mais robusta e orientada para a empresa, como o [EJBCA](#), ou mesmo o emergente projeto designado por [smallstep](#). Por fim, existem também soluções comerciais, algumas delas com versões de teste.

Apesar da aparente simplicidade, as PKIs reais são muito mais complexas, exigindo hardware dedicado para geração de chaves (**HSM - Hardware Security Modules**), sistemas distribuídos tolerantes a falhas para garantir operação e escalabilidade contínuas (tanto quanto possível), entre outras propriedades.

Caso não esteja interessado em promover competências técnicas no desenvolvimento e gestão de PKIs, pode aceder a uma CA muito simples através do [link https://hdsca.mafica.xyz/](https://hdsca.mafica.xyz/) (consiste numa implementação da primeira alternativa descrita anteriormente). Através do interface pode: 1) submeter o seu pedido de certificado (**Signing Service**), recebendo, na sequência, o seu certificado assinado; 2) descarregar o **certificado público da CA**, que irá necessitar mais tarde para verificar os certificados assinados pela CA; e 3) revogar certificados (**Revoking Service**). Tenha em atenção que os ficheiros que lhe são devolvidos não têm extensão, devendo atribuir-lhes a extensão `.crt` (embora não seja obrigatório, é aconselhado para uma mais fácil identificação).

Tenha em atenção que **NÃO PODE SUBMETER DUAS VEZES O MESMO PEDIDO** (o ID de um certificado tem sempre que ser único e com origem num ID de pedido único, também). Como é óbvio, a operação acima descrita deve ser repetida para cada um dos elementos do grupo, para os respetivos certificados X.509. **Se (desejavelmente) decidir criar sua própria PKI, não esqueça de incluir no seu logbook uma descrição do trabalho realizado.**

6. De regresso ao OpenSSL e porque para importar a sua chave privada em diferentes aplicações irá muito provavelmente precisar de um ficheiro no formato PKCS#12, deverá ainda usar o comando:

```
openssl pkcs12 -export -in pubcert.crt -inkey privkey.pem -
certfile CAcert.crt -name "my-name" -out priv-pkcs12.p12
```

onde:

- `pubcert.crt` é o seu certificado publico, assinado pela CA – poderá ter que converter o formato da codificação do certificado, de binário para o formato de texto (pode verificar abrindo o ficheiro com um vulgar editor de texto); se tal for necessário, use o comando `openssl x509 -inform der -in cert.cer -out cert.pem`

- `privkey.pem` contém a sua chave privada

- `CAcert.crt` é o certificado da CA (também pode precisar de o converter)


Ao executar o comando para obter o ficheiro no formato PKCS#12 ser-lhe-á solicitado que escolha uma palavra-chave, a qual servirá para o autenticar quando importar a chave privada e, por opção sua, sempre que for necessário usar a sua chave privada (não é necessário realçar a importância desta palavra-chave!). Como é lógico, cada elemento do grupo tem que repetir este processo para poder importar a sua chave privada.

Verifique o estado do seu ficheiro com o certificado assinado e a chave privada, usando o comando

```
openssl pkcs12 -info -in priv-pkcs12.p12
```

lista de certificados revogados e executa uma verificação por si próprio, enquanto no primeiro caso o cliente envia o ID do certificado para um servidor OCSP, que retorna seu estado.

e registre a resposta do comando; procure identificar os elementos que considera mais relevantes. Compare o resultado obtido, com o que obteve no passo 4 e assinale as diferenças.

 Antes de terminar e à semelhança do que foi sugerido para o PGP, talvez seja boa ideia fazer uma cópia de segurança das suas chaves públicas e privadas 😊.

Enviar e receber mensagens seguras

Neste exercício iremos utilizar, como referência, o cliente de *e-mail* Thunderbird (com o *add-on* Enigmail já instalado), procurando realizar experiências quer com os certificados PGP, quer com os certificados X509. No entanto, graças à modularidade por *plugins* ou *add-ons*, a descrição aplica-se a vários outros clientes (obviamente a menos das figuras aqui mostradas 😊), tais como o Windows Live Mail, o Eudora, ou o eM Client – em alguns casos poderá sentir alguns problemas com a validação do certificado privado (X.509).

1. O primeiro passo consiste na importação dos certificados PGP e X.509 para a sua plataforma/aplicação. No caso do Thunderbird, a própria aplicação inclui a função de importação dos dois tipos de certificados:
 - os certificados X.509 são importados através de um gestor acessível a partir da configuração da conta (em ambiente Windows, através do menu Tools → Account Settings → Security; em ambiente Linux, através do menu Edit → Account Settings → Security); e
 - os certificados PGP são importados através do menu Enigmail → Key Management.Outras aplicações utilizam os repositórios do próprio SO, ou um dedicado da implementação do OpenPGP que estiver a ser utilizada. Deve consultar a documentação respetiva, sem esquecer os eventuais procedimentos de validação. **Documente, no seu logbook, todos os passos realizados para a instalação dos certificados, em todos os computadores dos elementos do grupo, referindo o ambiente utilizado – no relatório deverá incluir uma secção por elemento do grupo.**
2. No que respeita aos certificados X.509, tem que indicar ao cliente de e-mail os certificados que vai usar para funções específicas (isto porque poderá ter vários). No menu Tools (Windows) ou Edit (Linux) escolha a opção Account Settings → Security (para a conta de email que estiver a usar). Tem agora a possibilidade de:
 - a. Gerir os certificados que dispõe no seu computador, através da opção Manage Certificates (ver Figura 4); pode ver os seus próprios certificados (chave pública e privada, tipicamente), os de outras pessoas, os de servidores, os de CAs reconhecidas e outros; tem ainda a possibilidade de importar certificados, para qualquer das classes anteriores – o que já terá feito no ponto anterior. Como nesta fase está a importar certificados X.509 assinados por uma CA, o respetivo certificado público tem que ser também importado. **Verifique e registre no seu logbook as evidências desse facto.**
 - b. Escolher o certificado que pretende usar para assinar e para decifrar (embora possam ser diferentes, na maioria dos casos é o mesmo). **Registe no logbook uma imagem que documente a configuração realizada.**

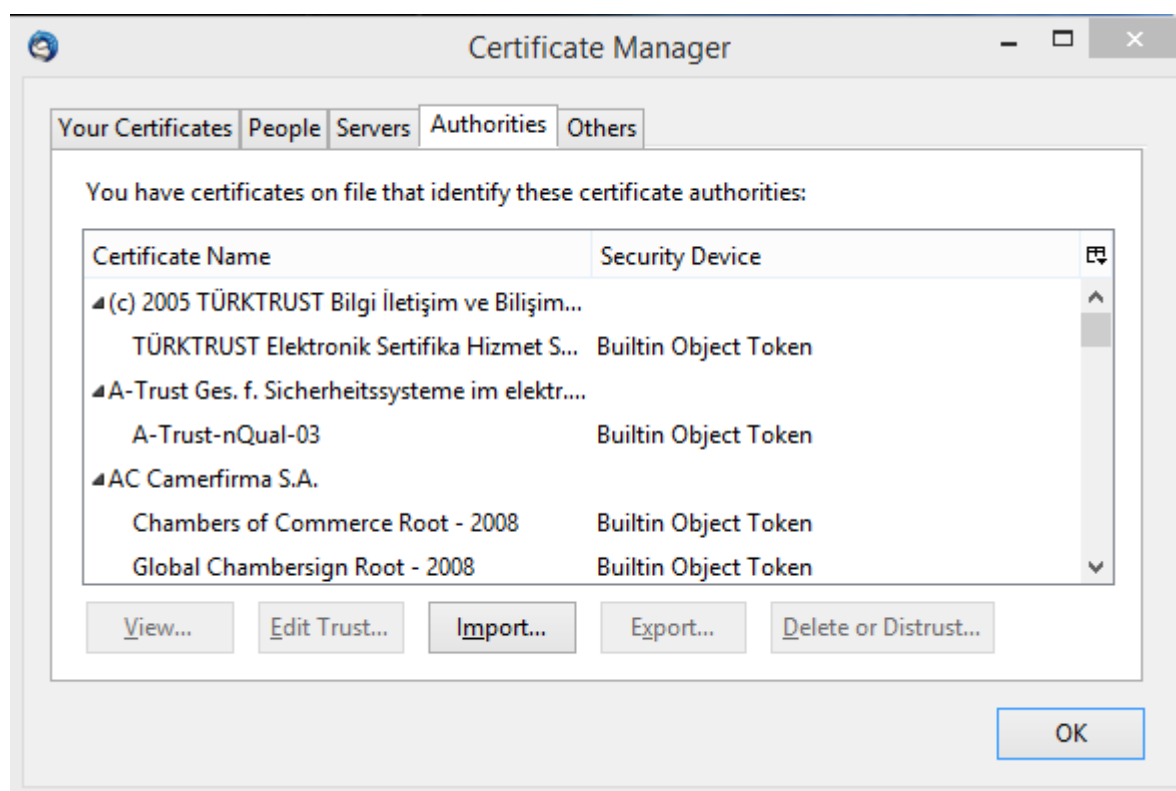


Figura 4 - Gestor de certificados do Thunderbird

3. No que respeita aos certificados PGP, a operação equivalente é realizada através do menu Enigmail, selecionando a opção Preferences e ativando Display Expert Settings and Menus. A janela que lhe aparece irá dar-lhe acesso as várias funções, que poderá explorar posteriormente, mas aqui iremos apenas referir o separador Key Selection. Nesse separador deverá selecionar as três primeiras opções (ver Figura 5), o que lhe permitirá que a aplicação selecione a chave adequada, usando como identificador principal o endereço de e-mail, apenas exigindo uma intervenção manual caso não seja possível inferir qual o certificado a usar. É ainda importante realçar a possibilidade de criar regras específicas para determinados endereços de e-mail (botão Edit Rules), o que permite um interessante grau de flexibilidade na gestão da forma como o Enigmail responde a mensagens cifradas/assinadas, dependendo do emissor e do destinatário.

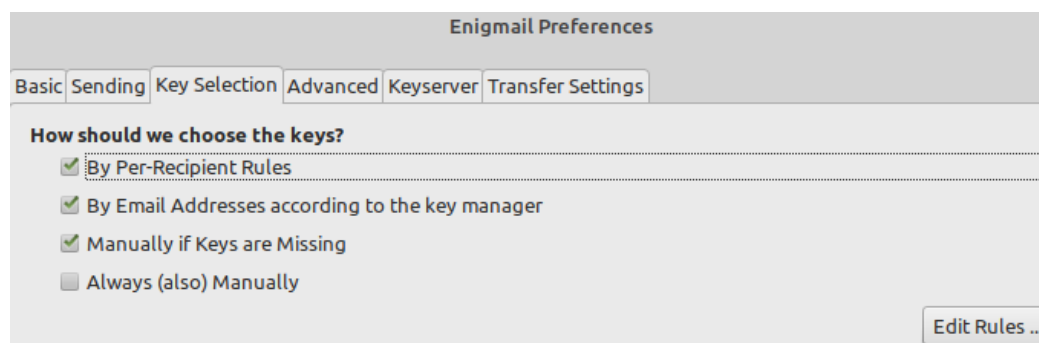



Figura 5 - Janela de configuração da escolha de chaves PGP, com o Enigmail

Tenha em atenção que (quer use certificados PGP, quer use certificados X509):

- a. o cliente de e-mail usa habitualmente o endereço de e-mail para escolher os certificados, se o seu certificado de assinatura tem um endereço de e-mail diferente do que usa para enviar e-mail, poderá não conseguir assinar mensagens!
- b. se alguém lhe enviar um certificado público por e-mail, ele será automaticamente guardado; no caso dos certificados X509, isso só acontece se a CA for reconhecida – no seu caso isso não acontecerá, porque a CA que usa é fictícia e não está devidamente registada; mas pode efetivamente “forçar” o seu sistema a reconhecer e aceitar a sua CA, bastando para isso carregar o respetivo certificado público na categoria de Autoridades de Raíz.
- c. A utilização do webmail não permite, habitualmente, executar este tipo de operações, assumindo-se que tal é feito no computador pessoal, ao nível do ficheiro, usando algum software para o efeito. Como exemplos refira-se a *suite* de segurança iSafeguard™, a extensão FlowCrypt do Google Chrome e o GPG (já anteriormente referido, mas que está orientado à utilização de certificados PGP). Para fazer assinaturas digitais o Adobe Reader serve perfeitamente, assim como o HelloSign (uma aplicação web que integra muito bem com o ambiente Google).


Os diversos clientes de e-mail de todos os elementos do grupo devem ser devidamente configurados. Após isso, devem exercer a troca de mensagens, com a assinatura e cifra. Deverão documentar todas as experiências no logbook, com os respetivos resultados, sendo de esperar que cada um dos elementos envie e receba pelo menos uma mensagem, para e de todos os outros elementos.

4. O exercício seguinte consiste em revogar um dos certificados e verificar o que acontece.

 Esta operação não é reversível, pelo que deve ter algum cuidado com o que faz.

A forma de revogar um certificado e o resultado expectável é diferente nos dois modelos – PGP, com o modelo *Web of Trust*, com um repositório central partilhado e sem gestão centralizada; e X509, com uma hierarquia bem estruturada, assente numa CA de topo.

(i) No primeiro caso (**PGP**), a revogação consiste na emissão de um certificado de revogação, assinado pela chave privada – em caso de perda da chave privada, não há forma de fazer a revogação ☹, o que representa uma ameaça para a consistência do processo.

 Por isso mesmo, uma boa prática consiste na produção de um certificado de revogação assim que se cria o par de chaves, guardando esse certificado de revogação cuidadosamente (eventualmente no mesmo *backup* onde se guarda a chave privada).

O certificado de revogação deve ser enviado para o servidor, para que quem o descarregue futuramente perceba o seu estado. Os servidores PGP, na sua maioria, trocam informação periodicamente, pelo que esse certificado revogado acabará por se disseminar, mas não existe nenhum mecanismo automático para que os clientes se atualizem (o Kleopatra oferece uma função `Tools → Refresh OpenPGP Certificates` que permite atualizar todos os certificados PGP existentes no `pubkeyring`, mas, naturalmente, obtendo a informação do servidor com o qual está configurado – a Figura 6 ilustra o registo de um certificado revogado).

(ii) No segundo caso (**X.509**) o processo é diferente, uma vez que existe uma entidade de topo, centralizada, responsável por esses aspetos da gestão de certificados. Neste caso existem dois mecanismos disponíveis: **CRL (Certification Revocation List)** e **OCSF (Online Certificate Status Protocol)**.

- a. CRL, como o nome indica, consiste numa lista mantida e assinada pela CA, com os IDs das chaves revogadas. A frequência com que essa lista é atualizada depende da política da CA, mas, seja como for, é da responsabilidade do cliente descarregar a lista e verificar o estado dos certificados que mantém localmente armazenados. A maioria dos programas de gestão de certificados permite configurar essa função de uma forma

automática. Os certificados emitidos por uma CA incluem, habitualmente (mas não forçosamente), um URL com a indicação do local onde a respetiva lista pode ser obtida – **CDP (CRL Distribution Point)**. Exceto pelo facto de ser centralizado, comparando com o modelo do OpenPGP, este mecanismo evidencia as mesmas limitações quanto ao tempo de resposta da atualização.

- b. OCSP, por seu lado, consiste num serviço *online*, destinado a fornecer o estado de utilização de um certificado, imediatamente. As CAs que implementam este serviço permitem uma resposta temporal mais eficiente, apenas evidenciando limitações quando o utilizador está *offline*. No entanto, é possível implementar e manter ambos os mecanismos, que se complementam nas vantagens / limitações.

Os certificados que obteve da CA na fase inicial deste exercício suportam o OCSP, incluindo um atributo necessário nos certificados produzidos (verifique o atributo **Authority Info Access** – por vezes referido apenas por **authInfo** – do(s) seu(s) certificado(s) X.509).

Revogue pelo menos dois dos certificados (um PGP e um X509) e verifique o impacto da operação no processo de troca de mensagens. Descreva a experiência no *logbook*, tendo o cuidado de indicar claramente as eventuais alterações e verificações que fez.

Henrique M. D. Santos	henrique.dinis.santos@gmail.com	certified	18/03/2010		OpenPGP	2D75 B09D CA...
Henrique M-D Santos	hsantos@dsi.uminho.pt	not certified	14/11/2003		OpenPGP	C585-8F3E-3AE...
Hans Hedbom	hans.hedbom@kau.se	not certified	04/09/2014	03/09/2019	OpenPGP	C6B5 5146 E1A8 ...
Filipe de Sa-Soares	fss@dsi.uminho.pt	not certified	18/11/2002		OpenPGP	C94C 781B 6206 ...
Charles Heselton	charles-heselton@cox.net	not certified	11/01/2004		OpenPGP	7BF8 D1F6 4829 ...

Figura 6 - Exemplo da visualização de um certificado PGP revogado

Proteger documentos locais

1. A larga maioria das aplicações que permitem gerir certificados permitem algumas operações adicionais, como por exemplo, cifrar ficheiros ou pastas. O PGP e o Kleopatra não são exceção. No caso do PGP essas operações podem ser executadas diretamente do utilitário PGP Desktop (ou PGP tools, dependendo da versão que tem instalada), utilizando o grupo de funções PGP Zip, mais precisamente o *PGP Zip Assistant* – ver Figura 7. Através de uma simples operação de *drag and drop* podem ser encriptados e/ou assinados diversos ficheiros e pastas. O Kleopatra inclui as funções equivalentes no menu *File* (ver Figura 7)
2. O mesmo conjunto de comandos está disponível a partir dos chamados menus de contexto do Windows e do Linux. Em qualquer janela, selecione um ou mais ficheiros e pressione o botão direito do rato. O menu de contexto que aparece dá acesso direto às funções de cifra e/ou assinatura, conforme mostra a Figura 8.
3. Existem ainda outras operações muito úteis no PGP Desktop, como sejam as associadas ao menu *PGP Disk*. Essas operações permitem criar um novo disco virtual cifrado, cifrar um disco completo, ou ainda eliminar definitivamente o conteúdo do espaço livre de um disco. Experimente algumas destas funcionalidades, tendo o cuidado de documentar todas essas experiências.

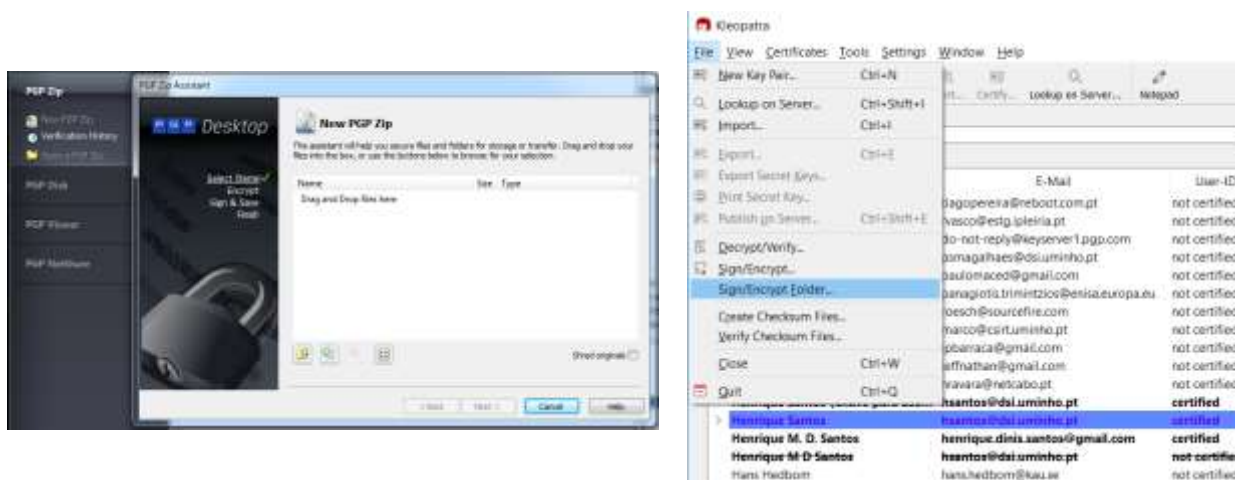


Figura 7 – Cifrar ficheiros / pastas a partir do PGP (esquerda) e Kleopatra (direita)

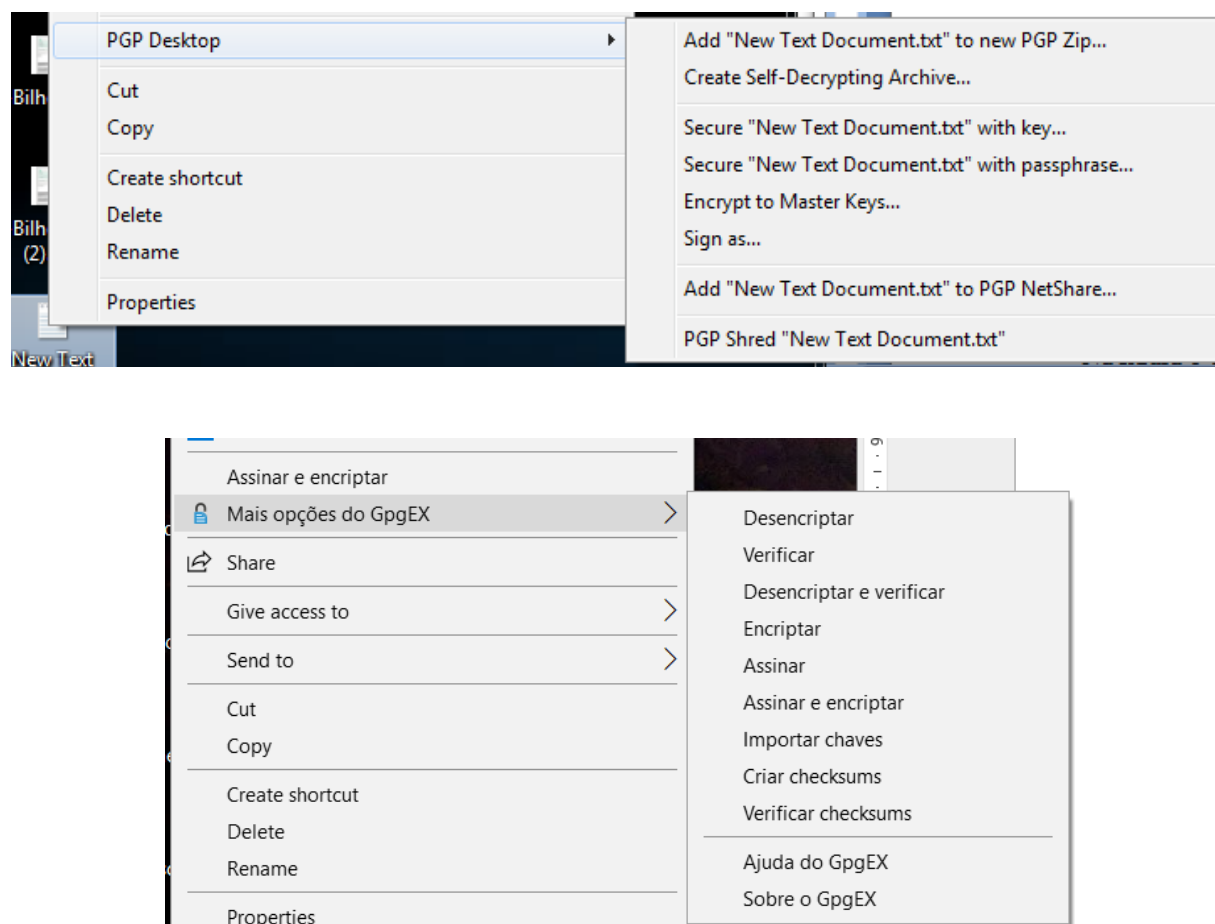


Figura 8 – Acesso às funções criptográficas a partir do menu de contexto: PGP em cima; e Kleopatra em baixo