



**Universidade do Minho**  
Escola de Engenharia

**METI**

## **TP3 - Certificados e PKI's**

### **Cibersegurança**

**Grupo 4**

**Alunos:**

Fernando João Santos Mendes (PG55807)

Bruno Miguel Fernandes Araújo (PG55806)

**Docente:**

Henrique Manuel Dinis Santos

5 de abril de 2025

# Conteúdo

<b>Lista de Figuras</b>	<b>iii</b>
<b>Lista de Tabelas</b>	<b>iii</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 PGP</b>	<b>1</b>
2.1 Dados da chave . . . . .	1
2.2 Identificação da Chave . . . . .	2
2.3 Par de chaves geradas . . . . .	2
2.4 Exportação da chave pública . . . . .	3
<b>3 Envio e receção de mensagens seguras (PGP)</b>	<b>4</b>
<b>4 Revogação do certificado PGP</b>	<b>7</b>
<b>5 X509</b>	<b>8</b>
5.1 Gerar par de chaves . . . . .	8
5.2 Pedido do certificado . . . . .	9
5.3 Certificado auto-assinado . . . . .	9
5.4 Desenvolvimento da PKI . . . . .	10
5.5 Ficheiro no Formato PKCS12 . . . . .	11
<b>6 Envio e receção de mensagens seguras (X509)</b>	<b>13</b>
<b>7 Revogação do certificado X509</b>	<b>14</b>
<b>8 Encriptação de ficheiros</b>	<b>15</b>
8.1 Ficheiro .txt e Pastas . . . . .	15
8.2 Discos/Usb-Drive . . . . .	16
<b>9 Outras Questões/observações</b>	<b>16</b>
9.1 Como as chaves PGP podem ser usadas além de cifrar e autenticar? . . . . .	16
9.2 Certificação cruzada – é possível um computador que use PGP enviar uma mensagem cifrada para um que use X.509? . . . . .	16
<b>10 Divisão de tarefas</b>	<b>17</b>
<b>11 Referências</b>	<b>17</b>

## Lista de Figuras

1	Criação da chave . . . . .	1
2	Par de chaves . . . . .	2
3	Alteração do servidor . . . . .	3
4	Chave pública no servidor . . . . .	3
5	Pesquisa por email . . . . .	4
6	Pesquisa segundo o nome . . . . .	4
7	Exportação das chaves . . . . .	5
8	Importação de chaves . . . . .	5
9	Gestor de chaves PGP . . . . .	6
10	Configuração do email . . . . .	6
11	Confirmação da validade do uso e validade da assinatura . . . . .	7
12	Revogação do certificado PGP . . . . .	7
13	Erro de envio dada a revogação do certificado . . . . .	8
14	Estado da chave privada. . . . .	8
15	Estado do pedido do certificado. . . . .	9
16	Estado do certificado auto-assinado (Part1) . . . . .	9
17	Estado do certificado auto-assinado (Part2) . . . . .	10
18	Receção de requests para o servidor OCSP . . . . .	10
19	Hash de ocspsinging.crt e de privkey.pem . . . . .	11
20	Criação do ficheiro no formato pkcs12 . . . . .	11
21	Verificar o ficheiro pkcs12 (Part1) . . . . .	11
22	Verificar o ficheiro pkcs12 (Part1) . . . . .	12
23	Associação pkcs12 a conta pessoal . . . . .	13
24	Envio de um mail encriptado com X509 . . . . .	13
25	Revogação de um certificado X509 . . . . .	14
26	Menu kleopatra de encriptação . . . . .	15
27	Resultado após a encriptação da pasta e do ficheiro .txt . . . . .	15

## Lista de Tabelas

1	Atributos Principais da Chave PGP . . . . .	1
---	---	---

# 1 Introdução

A criptografia desempenha um papel fundamental na segurança da informação, garantindo a confidencialidade, integridade e autenticidade dos dados. Neste contexto, o trabalho prático realizado teve como objetivo explorar os conceitos e as ferramentas associadas à gestão de chaves criptográficas e certificados digitais. A atividade permitiu a criação e gestão de pares de chaves públicas e privadas, tanto no modelo PGP quanto no padrão X.509 e a respetiva troca de mensagens entre utilizadores.

## 2 PGP

Num cenário inicial foram criados os perfis necessários com os emails respetivos para serem criadas as chaves mediante as condições propostas no enunciado.

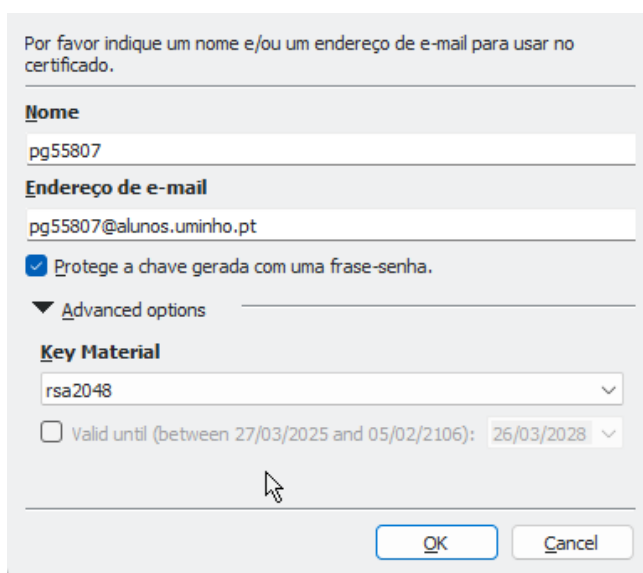


Figura 1: Criação da chave

### 2.1 Dados da chave

Atributo	Valor
User ID	pg55807 <pg55807@alunos.uminho.pt>
Fingerprint	6282 39F2 4058 FB7A FF1B 3ECE B567 6898 BBE7 AAE4
Validade	26/03/2025 -- ilimitado
Nível de Confiança	total

Tabela 1: Atributos Principais da Chave PGP

## 2.2 Identificação da Chave

A chave apresentada é a **privada**, conforme indicado explicitamente pelo campo "Private Key: on this computer". A assinatura está ligada à chave privada através de:

- **Fingerprint** (6282 39F2 4058 FB7A FF1B 3ECE B567 6898 BBE7 AAE4) - identificador único do par de chaves
- **User ID** (pg55807@alunos.uminho.pt) - vinculado à chave durante sua criação
- **Confiança "total"** - valida a autenticidade da chave
- **Algoritmo criptográfico** - garante que apenas esta chave privada pode gerar assinaturas válidas

Estes elementos criam uma relação criptográfica verificável entre a assinatura e a chave privada.

**Nota:** O email (pg55807@alunos.uminho.pt) e utilizador utilizados são exclusivos desta fase do trabalho prático. Na fase de envio de mensagens assinadas será utilizado o email configurado no ThunderBird.

## 2.3 Par de chaves geradas

Respostas sobre Subchaves PGP

### 1. As subchaves são públicas ou privadas?

As subchaves listadas são privadas, conforme indicado pelo campo "Armazenamento: neste computador". Cada subchave privada tem uma correspondente chave pública associada.

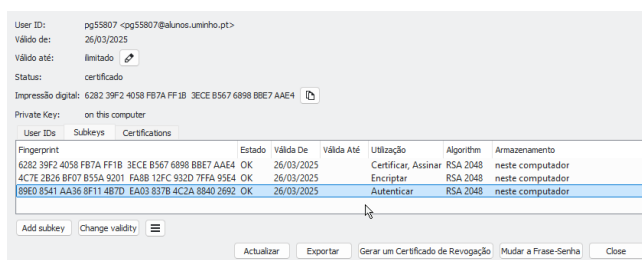


Figura 2: Par de chaves

### 2. Qual a utilidade de ter várias subchaves?

<b>Segurança</b>	Isolar funções (ex.: uma subchave só para assinar, outra só para encriptar)
<b>Flexibilidade</b>	Rotação de chaves sem alterar a identidade principal (User ID)
<b>Contingência</b>	Revogar individualmente subchaves comprometidas sem afetar outras
<b>Especialização</b>	Usar algoritmos diferentes para diferentes propósitos (ex.: RSA 2048 para assinatura, ECC para encriptação)

### 3. Qual a relação entre a chave mestra e as restantes?

A chave mestra (primária) é a identidade central no sistema PGP, vinculada ao usuário respetivo e usada para certificação e assinatura. Ela gera e gerencia subchaves, como a de encriptação que aparece na figura 4 e são usadas para funções específicas. Enquanto a chave mestra valida a autenticidade do conjunto, as subchaves executam outras operações como a de criptografar os dados ou assinar mensagens. A separação/diferenciação entre elas permite maior segurança, já que a chave mestra pode ser protegida de forma mais rigorosa.

## 2.4 Exportação da chave pública

Para realizar a devida exportação da chave pública foi necessário configurar a aplicação para o servidor e adicionar o seguinte servidor. Podemos verificar estas alterações na figura que se segue.

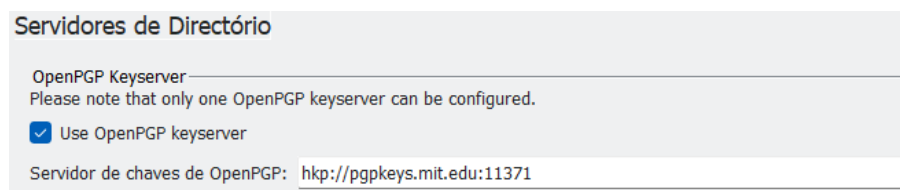


Figura 3: Alteração do servidor

De seguida foi enviada a chave pública para o respetivo servidor e averiguamos se a mesma tinha sido publicada, como podemos corroborar na figura 7.



Figura 4: Chave pública no servidor

No seguimento desta análise procuramos encontrar chaves públicas de outras entidades. Como tal, procuramos encontrar a chave pública através do nome "Henrique Santos" e do email "hssantos@dsi.uminho.pt" cujos resultados se seguem na figura 8 e 9.

# Search results for '0xcc16712c18a842ea'

Type	bits/keyID	cr. time	exp time	key expir
pub	2048R/18A842FA	2018-11-01		
uid	Henrique M D Santos <henrique.dinis.santos@gmail.com>			
sig	sig 18A842FA	2018-11-01		[selfsig]
	Notation data: preferred-email-encoding@pgp.com pgpme			
uid	HSantos <henrique.dinis.santos@dsi.uminho.pt>			
sig	sig 18A842FA	2018-11-03		[selfsig]
	Notation data: preferred-email-encoding@pgp.com pgpme			
uid	Henrique Santos <henrique.dinis.santos@gmail.com>			
sig	sig 18A842FA	2018-11-03		[selfsig]
	Notation data: preferred-email-encoding@pgp.com pgpme			
sub	2048R/C104C42F	2018-11-01		
sig	sig bind 18A842FA	2018-11-01		[]
sub	2048R/B6B739E3	2018-11-01		
sig	sig bind 18A842FA	2018-11-01		[]
sub	2048R/0B55FEAA	2018-11-03		
sig	sig bind 18A842FA	2018-11-03		[]

Figura 5: Pesquisa por email

## Search results for 'santos henrique'

Type	Website ID	Date	User ID	Url
pub	2948975438801	2023-11-06	henrique_santos@uol.com.br	henrique_santos@henrique.santoshenrique.com
pub	30729763661780	2022-02-24	Paula.Henrique.Santos_santos@uol.com	Paula.Henrique.Santos_santos@uol.com
pub	30729763661780	2021-05-19	Henrique.Santos_07f6a1e9d9a20a@Piscini.Club@uol.br	Henrique.Santos_07f6a1e9d9a20a@Piscini.Club@uol.br
pub	30729763661781	2021-05-17	alexandre.henrique_santos_alexandre_santos@henrique.br	alexandre.henrique_santos_alexandre_santos@henrique.br
pub	30729763661782	2021-05-17	alexandre.henrique_santos_alexandre_santos@henrique.br	alexandre.henrique_santos_alexandre_santos@henrique.br
pub	2948975438802	2020-04-14	lucas.h.santos@uol.com.br	lucas.h.santos@uol.com.br
pub	2948975438803	2020-04-14	lucas.h.santos@uol.com.br	lucas.h.santos@uol.com.br
pub	2948975438804	2020-04-14	ORSE_HENRIQUE_SANTOS_ORSE@uol.com.br	ORSE_HENRIQUE_SANTOS_ORSE@uol.com.br
pub	2948975438805	2018-11-18	Henrique.B.Santos_henrique@uol.com	Henrique.B.Santos_henrique@uol.com
pub	2948975438806	2018-11-18	Henrique.B.Santos_henrique@uol.com	Henrique.B.Santos_henrique@uol.com
pub	2948975438807	2018-10-20	Henrique.Santos_henrique.santos@uol.com.br	Henrique.Santos_henrique.santos@uol.com.br
pub	30729763661783	2018-08-29	Henrique.Santos_SantosContact_henrique.santos@henrique.com	Henrique.Santos_SantosContact_henrique.santos@henrique.com
pub	10140477588518	2018-06-10	Isai.Henrique.Silva.Santos_isai.henrique.silva.santos@uol.com	Isai.Henrique.Silva.Santos_isai.henrique.silva.santos@uol.com
pub	2948975438808	2018-04-16	caetano.henrique.Santos_caetano.henrique@uol.com	caetano.henrique.Santos_caetano.henrique@uol.com
pub	689682323295	2022-02-17	luciohenrrique.F95 for Netwrok Henrique do Santos	luciohenrrique.F95 for Netwrok Henrique do Santos

Figura 6: Pesquisa segundo o nome

Após a pesquisa concluímos que a pesquisa por email retornou uma única chave PGP associada a Henrique Santos, com detalhes precisos (ID: 2048R/18A842EA, criada em 2018, contendo subchaves e múltiplos identificadores). Já a busca por nome trouxe diversos resultados diferentes e irrelevantes devido a variações do nome e emails não relacionados. Desta forma podemos concluir que a procura por email é mais eficaz para encontrar chaves específicas, enquanto a busca por nome gera muitos resultados o que se pode tornar um pouco ambíguo.

### 3 Envio e recepção de mensagens seguras (PGP)

Para testarmos a troca de mensagens foi usado o Thunderbird (com a extensão do Enigmail) através dos certificados PGP. Para ativar a opção foi necessário realizar os seguintes passos:

**Nota:** De notar que o email que seguirá para o exemplo do envio das mensagens é diferente para que estivesse em conformidade do o email usado para no Thunderbird.

## 1. Exportação das chaves criadas

Através do método de exportação de chaves disponível nas opções do software da Kleopatra como podemos averiguar na Figura 7

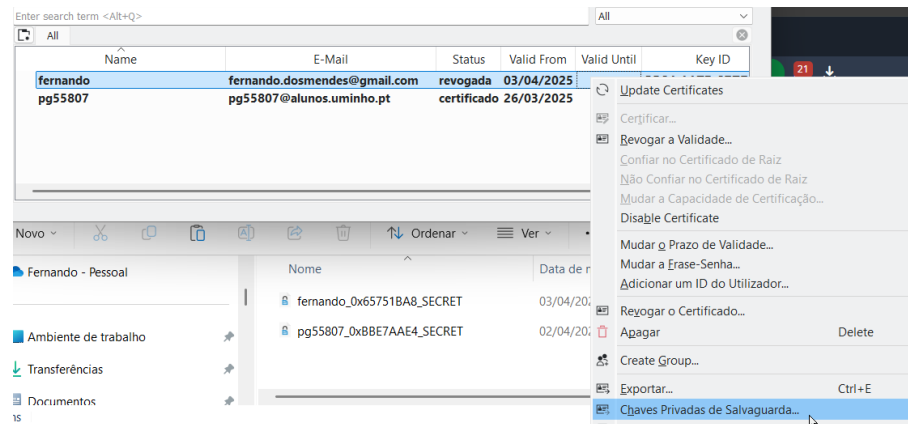


Figura 7: Exportação das chaves

## 2. Importação da chave para o Thunderbird

De seguida foi importada a chave que nos permitirá mais tarde assinar as mensagens que desejamos enviar. Podemos averiguar este processo na figura que se segue.

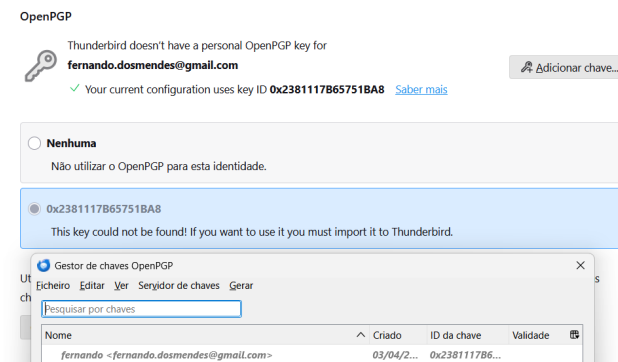


Figura 8: Importação de chaves

## 3. Importação da chave pública do destinatário

Para que fosse possível encriptar o email a enviar e que apenas o destinatário tivesse a oportunidade de a ler foi necessário adicionar a chave pública do mesmo, visto que ainda não havia conhecimento da mesma. De notar que a partilha de desta chave foi feita anteriormente. Podemos averiguar na figura 12 a importação desta mesma chave.



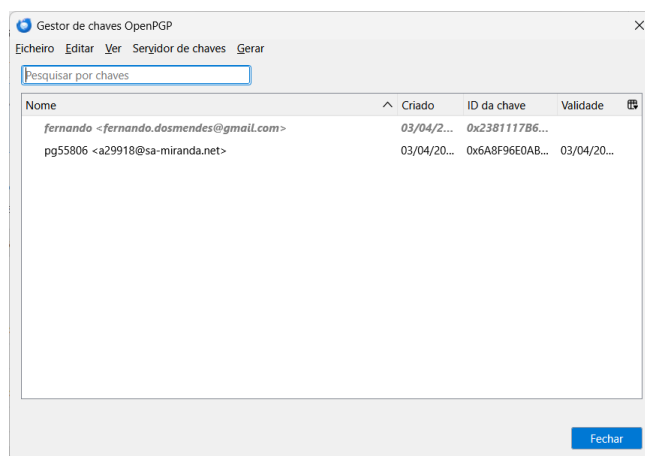


Figura 9: Gestor de chaves PGP

#### 4. Envio do email com a opção do PGP

Por último foi enviado o email e foi selecionada a opção PGP como pretendido. A configuração do email pode ser consultada na figura 13

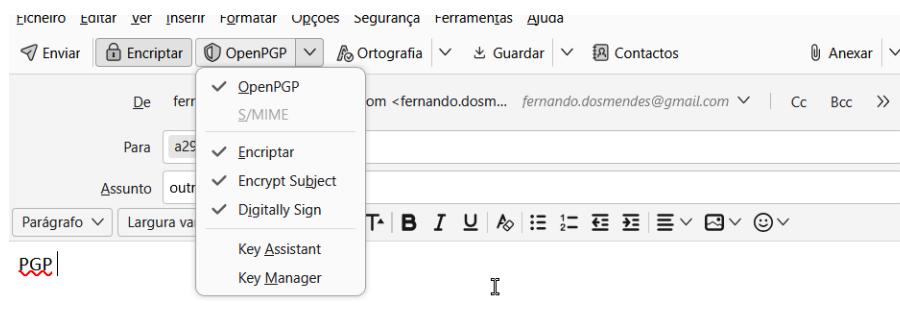


Figura 10: Configuração do email

Da mesma forma o aluno PG55806 enviou um email a responder ao aluno com o email Fernando.dosmendes@gmail.com, cujo é possível de analisar através da figura 14 e averigurar que a comunicação foi bem sucedida e assinada.

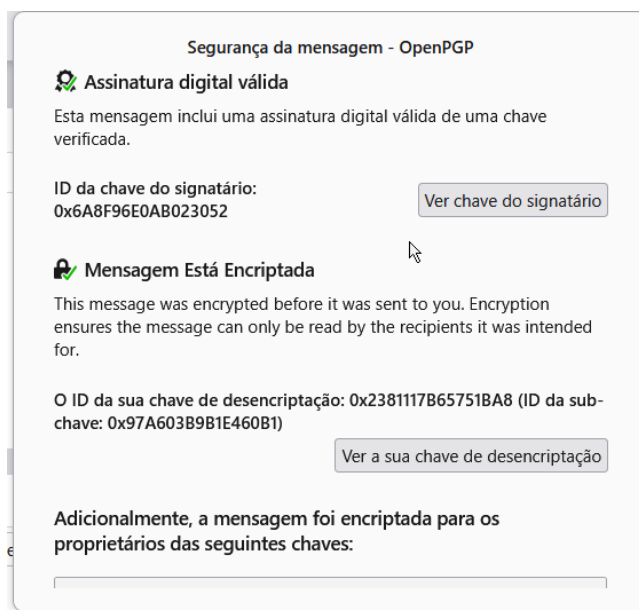


Figura 11: Confirmação da validade do uso e validade da assinatura

## 4 Revogação do certificado PGP

Para revogar o certificado foi necessário ir ao gestor de chaves (figura 12) e ao clicar sobre aquele que era pretendido revogar usamos a opção devida como podemos identificar na figura 15.

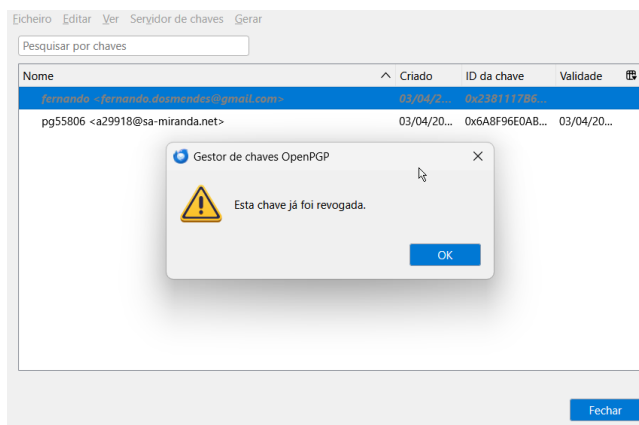


Figura 12: Revogação do certificado PGP

De seguida ,e para averiguar quais os impactos da revogação, foi ralizado uma nova tentativa de envio de um email e verificamos (figura 16) que o envio passa a ser inviável.



## 5.2 Pedido do certificado

Geramos um pedido do certificado com o comando **openssl req -new -key privkey.pem -out cert.csr** e verificamos o seu estado com o comando **openssl req -text -noout -verify -in cert.csr**, onde obtivemos o seguinte:

```
C:\Users\Bruno\Desktop\Mestre\Cyber\TP3\Cyber\pg55806>openssl req -text -noout -verify -in cert.csr
Certificate request self-signature verify OK
Certificate Request:
Data:
  Version: 1 (0x0)
  Subject: C=PT, ST=Braga, L=Braga, O=Cyber, OU=Cyber, CN=pg55806, emailAddress=pg55806@alunos.uminho.pt
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
    Modulus:
      00:a7:4a:74:de:fd:ce:54:76:a9:c6:b2:0a:d5:04:
      81:f3:d5:40:d1:e2:71:f1:72:7f:8b:d7:3d:04:fb:
      7e:70:7a:e2:5c:a4:71:2c:c7:03:5a:fa:68:d5:cb:
      52:8e:ee:97:6c:18:18:4d:23:e8:3b:cc:96:65:50:
      d1:5f:e8:a4:96:81:1a:8f:30:e8:4a:9b:e9:bd:7b:
      fe:2f:27:08:af:14:c6:e3:9a:d1:c6:31:95:5d:e9:
      34:9a:7c:6d:21:48:dc:a3:a0:32:32:cf:9b:00:ae:
      38:98:41:00:1d:4e:a6:b0:8a:22:ab:51:8e:0f:6f:
      81:b2:1d:7d:0f:26:b5:70:ef:79:c7:ec:25:b3:9a:
      9e:18:cc:2f:f4:4d:03:f1:24:70:bf:37:16:8b:1f:
      9f:26:1f:d8:59:53:5e:19:0b:0f:d4:df:fe:91:3a:
      6a:45:9c:67:65:ef:19:c3:0d:56:4e:ed:6c:35:74:
      4f:c0:83:b7:9b:d3:88:42:14:3b:8e:8c:ba:ad:38:
      c7:18:0e:a6:fc:66:c5:6f:4d:5c:b3:9a:c4:b9:85:
      f0:e2:0b:71:75:13:53:06:93:7e:c6:81:36:35:86:
      b0:88:97:6b:1b:3e:a7:0a:0a:d3:63:77:f6:91:1f:
      5b:2a:09:c3:78:ce:b2:3c:b4:23:c2:d5:65:19:0f:
      ed:d9
    Exponent: 65537 (0x10001)
  Attributes:
    unstructuredName          :Cyber
    challengePassword         :PG55806pg.
  Requested Extensions:
  Signature Algorithm: sha256WithRSAEncryption
  Signature Value:
    8c:71:eb:45:3a:58:8d:e3:c5:43:7b:51:63:e4:00:bc:70:ef:
    bb:5e:81:66:6b:64:24:17:d1:44:7c:f8:dd:9b:89:11:88:58:
    df:5a:1a:34:e7:32:4e:ff:71:ed:62:c4:14:30:8e:da:9d:5b:
    22:67:58:1b:3a:fd:0c:9d:ca:35:95:08:81:dc:b6:de:6d:28:
    be:78:8a:c5:0f:6b:f6:70:ab:1a:1f:b2:ba:82:b5:21:d3:d7:
    1ea5:19:46:de:40:96:fa:f9:1b:a0:9f:92:e7:c0:8b:51:f4:
    3f:17:50:be:25:fd:b2:df:5f:e0:fa:3c:aa:38:84:82:a0:cf:
    ae:4d:f2:91:94:28:c0:90:7c:96:34:e4:7b:08:72:e4:56:ef:
    17:93:e6:7d:68:7c:3d:8e:4c:6e:ee:d1:4b:64:66:1a:40:c9:
    73:17:ef:3e:0b:7f:e2:92:00:29:44:50:b6:d8:61:43:dc:e6:
    e8:fd:9a:ef:24:34:ff:67:0c:2d:76:82:43:fe:f3:49:2e:ce:
    a1:83:92:51:55:f2:07:76:b0:9b:7e:28:a6:a5:68:70:97:0a:
    d3:05:c0:d2:1a:0a:da:82:c6:8c:04:28:07:4c:19:72:90:6b:
    31:b1:d7:01:03:a0:4c:52:df:d5:32:a8:6f:f9:b1:70:92:
    37:70:6d:f7
```

Figura 15: Estado do pedido do certificado.

## 5.3 Certificado auto-assinado

Assim como as duas tarefas anteriores, fizemos dois comandos, um que gera o certificado auto-assinado **openssl x509 -req -in cert.csr -signkey privkey.pem -out privcert.crt** e um que irá verificar o estado deste **openssl x509 -text -in privcert.crt**, onde obtivemos o seguinte:

```
C:\Users\Bruno\Desktop\Mestre\Cyber\TP3\Cyber\pg55806>openssl x509 -text -in privcert.crt
Certificate:
Data:
  Version: 1 (0x0)
  Serial Number:
    2d:51:02:1a:cd:8d:97:a0:1b:46:e2:78:8e:03:3f:8a:72:45:b6:40
  Signature Algorithm: sha256WithRSAEncryption
  Issuer: C=PT, ST=Braga, L=Braga, O=Uminho, OU=Uminho, CN=pg55806, emailAddress=pg55806@alunos.uminho.pt
  Validity
    Not Before: Apr  2 08:47:42 2025 GMT
    Not After : May  2 08:47:42 2025 GMT
  Subject: C=PT, ST=Braga, L=Braga, O=Uminho, OU=Uminho, CN=pg55806, emailAddress=pg55806@alunos.uminho.pt
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
    Modulus:
      00:c5:21:dc:89:64:52:33:be:a6:1b:aa:9d:db:3e:
      7a:78:fe:36:24:fb:cd:62:34:93:67:71:f7:d7:ce:
      9f:36:40:29:f3:06:3f:0b:6c:df:25:94:6d:ea:ea:
      2d:dc:c1:91:35:18:0f:a6:14:1e:80:f5:57:be:13:
      01:d7:bc:74:64:67:d1:f1:8e:54:b3:6e:ee:89:70:
      67:ff:9e:92:a4:f0:52:56:54:a2:98:05:cb:bf:7f:
      6f:41:e1:9f:43:17:03:4b:67:eb:3d:4b:11:c7:a0:
      91:b3:54:a0:78:78:1b:72:ea:fl:37:f9:69:3d:85:
      d8:c4:ad:b1:40:16:df:ab:15:0b:7b:32:1b:04:5d:
      8b:23:a5:1f:4c:6d:3d:f8:75:5d:5b:c0:27:f5:ab:
      51:b0:db:e8:8f:9b:ac:75:70:58:6d:c6:28:e8:01:
      b3:ee:1b:00:27:04:09:fb:01:23:01:41:a5:be:fe:
      46:28:84:40:90:73:55:0e:93:el:ea:7a:71:39:e3:
      73:e3:e2:e1:bb:94:32:3c:12:cf:c3:fe:96:b8:9b:
      51:e7:70:fa:b3:b2:ce:e1:2f:6d:b8:7c:88:a7:33:
      a5:d0:44:cb:a7:eb:84:ad:de:a5:46:40:4c:c0:3a:
      cb:c0:08:31:c7:e8:56:f6:06:1b:9c:91:19:2a:ee:
      22:a3
    Exponent: 65537 (0x10001)
```

Figura 16: Estado do certificado auto-assinado (Part1)



## 5.5 Ficheiro no Formato PKCS12

No passo anterior, ao seguirmos o guia, criamos os ficheiros necessários para o comando indicado no enunciado, **pubcert.crt** será o **ocspSigning.crt**, a **privkey.pem** seria a previamente gerada e **CA-cert.crt** será o **rootCA.crt**.

Neste passo tivemos algumas dificuldades de identificar o ficheiro correto para ser usado, sendo que inicialmente tínhamos seguido o guia um pouco às cegas, nesta busca acabamos por encontrar um problema, as hashes usadas para as chaves do ficheiro **privkey.pem** e do certificado **ocspSigning.crt**, tivemos de recriar a chave com o comando **openssl req -new -key privkey.pem -out new.csr** e de re-assinar com **rootCA.crt** com **rootCA.key** através do comando

```
openssl x509 -req -in new.csr -CA rootCA.crt -CAkey rootCA.key -CAcreateserial -out ocs-  
Signing.crt -days 365.
```

Agora comparando as hashes já podemos verificar que têm o mesmo.

```

netsim@netsim-vm:~/Desktop/Cyber/X509/Cenas Geradas$ openssl x509 -noout -modulus -in ocspSigning.crt | openssl md5
MD5(stdin)= df51b7409617375ae92ac906d836676e
netsim@netsim-vm:~/Desktop/Cyber/X509/Cenas Geradas$ openssl rsa -noout -modulus -in privkey.pem | openssl md5
MD5(stdin)= df51b7409617375ae92ac906d836676e

```

Figura 19: Hash de ocspSinging.crt e de privkey.pem

Agora sim, é possível criar o ficheiro pkcs12 através do comando indicado no enunciado, verificamos o estado do seu ficheiro.

```

netsim@netsim-vm:~/Desktop/Cyber/x509/Cenas_Geradas$ openssl pkcs12 -export -in ocspSigning.crt -inkey private/ocspSigningKey.pem -certfile rootCA.crt -name "my-ocsp-responder" -out priv-pkcs12.p12
Enter Export Password:
Verifying - Enter Export Password:

```

Figura 20: Criação do ficheiro no formato pkcs12

[illegible]

Figura 21: Verificar o ficheiro pkcs12 (Part1)

```

1sD/yL7rXU9+350u3Rq/LjkwqQD2bfS24z2o8J10WQIDAQABo1MwUTAdBgNVHQ4E
FgQ0a2jstwmJk46iwiBFeQSA4NzeuAwHwYDVR0jBBgwFoAUa2jstwmJk46iwiB
FeQSA4NzeuAwDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0BAQsFAA0BgQBAJv0h
Y0dZBERq5Jiz9obX3D3B1M3d3dH0fZU/0J1A8Gud+6K//VVqfXacDzBLTDA5op56
Fq3CW0ZRBjC6ihyrlcdSAgLSKWEJwraMavckQJ9hFqqweFF9jMv7g6ZpoJHqEei
InexM06/so2D2h42xTxzV+FqloeTDads4MPspA==
-----END CERTIFICATE-----
PKCS7 Data
Shrouded Keybag: PBES2, PBKDF2, AES-256-CBC, Iteration 2048, PRF hmacWithSHA256
Bag Attributes
    localKeyID: 99 E1 B1 91 AC 4E B7 9F C8 0C 1F 24 C8 69 21 57 7A 71 02 EF
    friendlyName: my-ocsp-responder
Key Attributes: <No Attributes>
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFLTBxGkqhkiG9w0BBQ0wSjApBgkqhkiG9w0BBQwwHAQIUP045KBF75kCAgGA
MAwGCCqGSIb3DQIJBQAwHQYJYIZIAWUDBAEqBBBKgdhFubaLy0yHKSiBJS2PBIIE
0EAJWj598QU2JwSuRlyZ1K5ludGahLE7IhikHN+pLrLphFKfUJDVLVWEFmp4gqvF
wLv/yGIG+p7oc77JloY24VY0K62Dgz0Tb7DYVXmNdrPqL1XKc2ZQX1n3kCcv8zyM
Pb7Ewkofv9E+hTQbaxFrrHz04lffo4qs+xPw0nU4K9k37gham4eGe5XyMdLUUhDe
mKcL+HiWHiIqd0WoztdUMttjehNwTLy2eckfYyyC1NeWaxUb644qc67ihLEHlDA
x20qa1YiLYV2Mfhq6lrihGmk5gL6dcIQ9XTQ/GeFZud9V9b1wVtphDhqvD0h2U3
ZCIwKEBwY6E0f9ZqI3xL0A/1TDd8cv19S8w8n8DHCzcCwGR+/4SsE3Nw09dPay7
RRDPzfoQlnqXMLELAVLE1s02baUW/Z09XI8i+5FhJCsXG1mT7UT9T2XTjXNxytJ
0lbGpempNt1myRCQRdS90/Fw+kNhjqnS2V7U0jVgV8yyxMtdclD7jRAArv1501kF
s1b+qErCAXG5iVRlSXpQL0k20h7w0peC+/Bv88HK4dQGScG4rd50o2DQ0tVaxD5lH
MT2bYmPc5keIEgMmquEqBIWH/LPthWtSFcndbxyj8uzI04QrEVPhJZVhIsUy99
8BYRIUdq3exbvNKRk49rZZogU6HZ+7X0brpSvAdrpTXxa5bc+HPR01jyLgDYwv
KnBIvWbWNO0Qz20wC62Cp3dlZoPMG+8WhliSfcdyzvj/zwx0GZuPkjB/qUvZnBie
ngsterSkMfYgG3C90jsSkYuLpY8X8jIxx0N0hrJzmKDig3xfKc7fPA0mo1YYXIOF
yFHy7WTdAw2kdGeQ+om02Uks5i8cQ0KI7mNPt3DtMsFN//v1XL3kVi09G5+bnK6X
Rhq3vn36TNR8xUwiERb8SGfk24xtAIyc3uTKQl+4l2dJl2844mXpZ2QhWEfPGRd+
dSvXzCnpa01cv2Kg20GVVCdoCgWfyTvj+5DctkzZB0PkLXgEqs0xTwwxS0azZsUW
m2fdsuaUVYRizhee/00BCAQJrfHasknIDmTv5mfWrl3nVGEEvtZRSJb0E+DB5h7N
lTmZ9Mo8Dma0g2kXAHT50Dxx7eMre/vo+CI60VBRiThzTTsive7P5Raw09JTCsKB
6+pGo1hobBMJgqw5iC4zwI0STvtYRR57PYZkSnvTc8WB4Upg0EPo6N+N8Dq47cAN
6hk+GK004XDP8MqZ4dRqfLXVxX0APusHEUMtNgwW6RIYGLSugK7zQo1ljyk4YQMP
mjUfM0yJhr2fw+3K0pG0rB3CDzsRJYqMxtqPx70bJuP0gezYGNLSxw5S0L2zNU6
T9C18Y51n7CmYQlxCvHvZrc4Y3iP5Sr3PhHdbtk9r9FI0dLWJoML09Dd0Kmr8v09
a4QDZ+UmE6uSPWhtGGyANiKbZjUr42c+BRx+PEPK+h+/7Uqpb8Azr6krna0fUc+Q
d8tg8fopeUQ6pWPMY00QFejvYWyK2W6b1CVvB+ls0u6ZtLuZzcl/q3ygsqTbPwm9
2Z0hyad4rIByiDNPLogofy9Vt5XyQ1pfiH1a7taVu9yh8500DdcnMLJYJgmDUSai
G3JJNcRGdYCPg7ftjz0E8q3AjVLgcSTR09Kuk/VVrsrt
-----END ENCRYPTED PRIVATE KEY-----

```

Figura 22: Verificar o ficheiro pkcs12 (Part1)

No ficheiro pkcs12, estão presentes tanto as características do certificado como da chave privada, enquanto que no passo 4 ( 5.3 ) são exibidas apenas as informações públicas, como a chave pública. Podemos também observar que tanto o pkcs12 como o ficheiro do passo 4 partilham o mesmo emissor.



## 6 Envio e recepção de mensagens seguras (X509)

Para o envio das mensagens encriptadas com X509, o processo foi similar ao do PGP, usamos novamente a ferramenta thunderbird e associamos o ficheiro pkcs12 á conta através do campo de definições da conta.

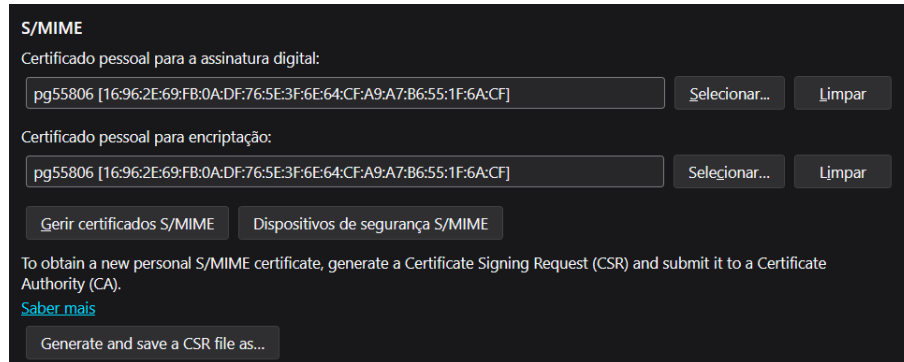


Figura 23: Associação pkcs12 a conta pessoal

Após isto, ao enviar o mail aparece a opção de encriptar usando S/MIME, que seria então X509.

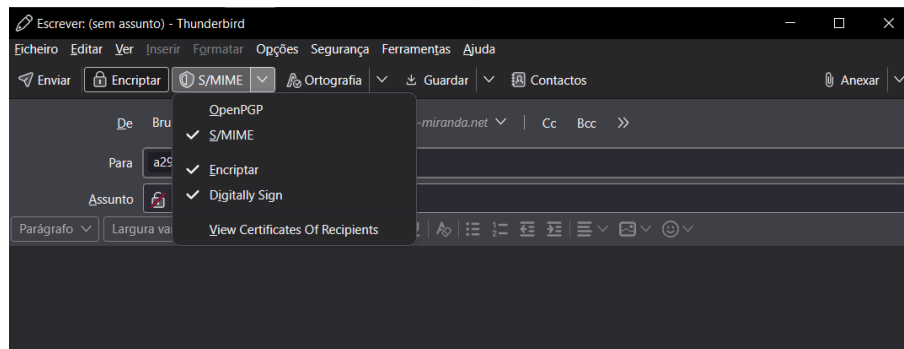


Figura 24: Envio de um mail encriptado com X509

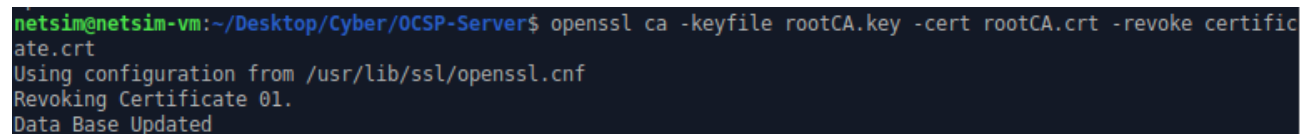
Infelizmente, ao ativar esta opção, conseguimos enviar o e-mail, mas suspeitamos que não está a ser enviado de forma encriptada. Não conseguimos identificar exatamente o problema e então não foi possível estabelecer uma troca correta de e-mails encriptados utilizando X509.



## 7 Revogação do certificado X509

A revogação do certificado X509 será feita através do comando

**openssl ca -keyfile rootCA.key -cert rootCA.crt -revoke certificate.crt**



```
netsim@netsim-vm:~/Desktop/Cyber/OCSP-Server$ openssl ca -keyfile rootCA.key -cert rootCA.crt -revoke certificate.crt
Using configuration from /usr/lib/ssl/openssl.cnf
Revoking Certificate 01.
Data Base Updated
```

Figura 25: Revogação de um certificado X509

Como já tínhamos implementado um servidor OCSP, após um restart, é possível verificar o estado do certificado através de um pedido enviado por um cliente. Este pedido é executado noutro terminal, direcionado ao servidor OCSP local (assim como já foi previamente demonstrado). Para isso foi utilizado o comando

**openssl ocsd -CAfile rootCA.crt -issuer rootCA.crt -cert certificate.crt -url http://127.0.0.1:8080 -resp\_text -noverify**

A nossa ideia era associar este servidor ao thunderbird de forma a que este depois fizesse o pedido automaticamente e verificasse que o certificado foi revogado ou não, infelizmente não o conseguimos fazer, não temos a certeza se o problema é por causa do servidor ser local ou se é do próprio thunderbird que não autorize/não confie servidores não oficiais.

Como somos apenas dois alunos, pretendíamos utilizar a conta pessoal de cada um e criar ainda uma terceira conta. Depois, revogaríamos dois certificados associados a duas dessas contas e verificaríamos as diferenças na troca de mensagens entre as três.

## 8 Encriptação de ficheiros

Mostraremos como foi feita a encriptação de um ficheiro de texto e um folder, e ainda explicaremos como iríamos encriptar um disco ou uma usb drive.

### 8.1 Ficheiro .txt e Pastas

Para a encriptação do .txt e da pasta, seguimos os passos indicados no enunciado, onde demos uso das ferramentas do kleopatra, Encriptamos um ficheiro de texto e um folder dando uso da Ferramenta do Kleopatra. Clicamos no botão direito no que queremos encriptar e escolhemos a opção "Assinar e encriptar", escolhemos o certificado que pretendemos usar e se queremos ou não atribuir uma palavra pass.

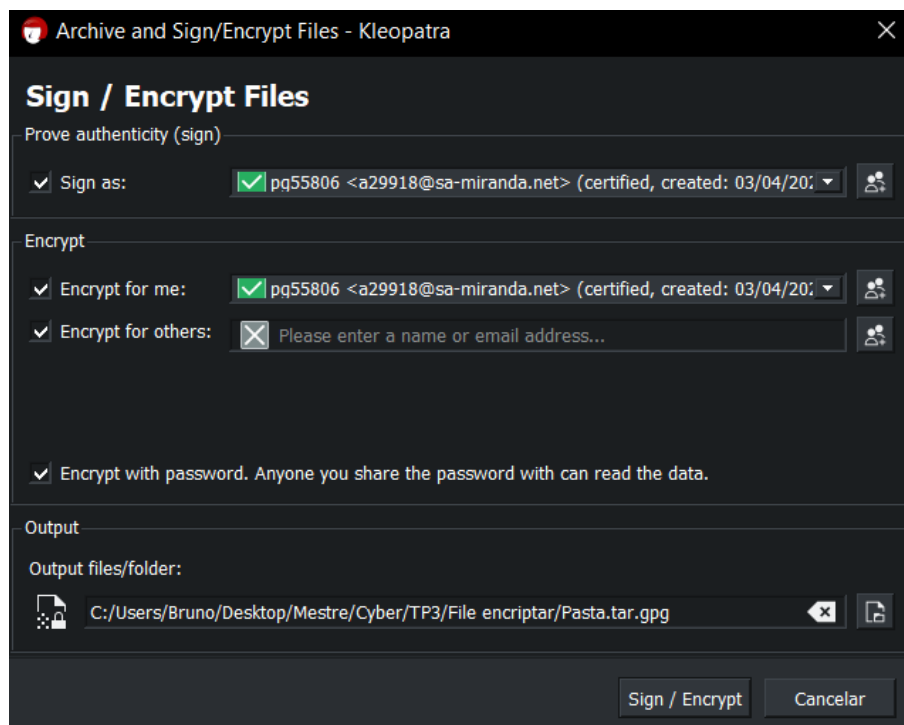


Figura 26: Menu kleopatra de encriptação

Name	Date modified	Type	Size
Pasta	02/04/2025 20:05	File folder	
Pasta.tar	05/04/2025 19:28	OpenPGP Binary Fi...	1 KB
Teste	02/04/2025 20:12	Text Document	0 KB
Teste.txt	05/04/2025 19:30	OpenPGP Binary Fi...	1 KB

Figura 27: Resultado após a encriptação da pasta e do ficheiro .txt

Para a desincriptação é apenas clicar nos ficheiros encriptados, clicar no botão direito do rato e escolher a opção "Dencriptar e verificar".

## 8.2 Discos/Usb-Drive

No que diz respeito a discos ou USB drives, identificámos duas opções: a primeira, mencionada no enunciado, consiste em utilizar a ferramenta **PGP Desktop** para criar um disco virtual cifrado ou encriptar um disco completo; a segunda, dando uso da ferramenta **VeraCrypt**, cujo processo descreveremos em maior detalhe.

Com o **VeraCrypt**, começa-se por criar um volume encriptado, seleccionando o dispositivo externo que se pretende encriptar (seja um disco ou uma pen). De seguida, escolhe-se o algoritmo de encriptação e define-se uma palavra-passe. Por fim, formata-se a unidade, eliminando todos os dados existentes. Desta forma, apenas quem tiver esta ferramenta e conhecer a palavra-passe conseguirá aceder ao disco ou pen.

## 9 Outras Questões/observações

### 9.1 Como as chaves PGP podem ser usadas além de cifrar e autenticar?

As chaves PGP, embora sejam mais conhecidas para cifrar e autenticar mensagens de e-mail, têm uma variedade de aplicações adicionais que podem ser úteis no dia a dia. Por exemplo, uma das utilizações é a criptografia de ficheiros ou discos. Outro uso é em VPNs, onde as chaves PGP podem servir para autenticar utilizadores ou gerar chaves de sessão temporárias. Embora não seja tão comum como o uso de certificados X.509, algumas implementações de VPN, como o OpenVPN, permitem configurações personalizadas que podem integrar chaves PGP para maior flexibilidade. Além disso, as chaves PGP são utilizadas na assinatura de software.

### 9.2 Certificação cruzada – é possível um computador que use PGP enviar uma mensagem cifrada para um que use X.509?

Não é possível uma certificação direta entre PGP e X.509 devido às diferenças fundamentais nos seus modelos de confiança. O PGP baseia-se na "web of trust", onde os utilizadores validam mutuamente as chaves uns dos outros, enquanto o X.509, usado em certificados SSL/TLS e S/MIME, depende de autoridades certificadoras (CAs) centralizadas. Esta diferença faz com que os dois sistemas não consigam comunicar nativamente. No entanto, existem soluções indiretas, ainda que não sejam ideais. Uma delas é a conversão de chaves entre os dois formatos usando ferramentas como o 'gpgsm', que faz parte do GnuPG. Este processo permite transformar uma chave PGP num certificado X.509 e vice-versa, mas requer conhecimentos técnicos avançados e configuração manual. Outra abordagem envolve o uso de um servidor intermediário que desencripta a mensagem recebida em PGP e a volta a encriptar em X.509 antes de a reenviar. Contudo, esta solução compromete a segurança de ponta a ponta, pois o intermediário teria acesso ao conteúdo original. A maneira mais simples de resolver este problema é ambos os lados adotarem o mesmo padrão. Se uma organização ou grupo de utilizadores decidir usar apenas PGP ou apenas X.509, a comunicação cifrada torna-se possível sem complicações.

## 10 Divisão de tarefas

Ambos os membros do grupo contribuíram de forma equitativa para este projeto, investindo aproximadamente 15 horas cada. Este projeto teve uma escala muito superior aos anteriores, onde sentimos o impacto de termos apenas dois elementos. Grande parte do tempo foi dedicado à implementação da PKI, à pesquisa e desenvolvimento de respostas para as questões levantadas e, naturalmente, à elaboração deste relatório.

## 11 Referências

- OpenPGP. (n.d.). *OpenPGP.org*. <https://www.openpgp.org/>
- Bhashineen. (n.d.). *Create your own OCSP server*. Medium.  
<https://bhashineen.medium.com/create-your-own-ocsp-server-ffb212df8e63>
- PKI Tutorial. (n.d.). *Simple PKI Tutorial*. Read the Docs.  
<https://pki-tutorial.readthedocs.io/en/latest/simple/index.html>
- OpenSSL. (n.d.). *OpenSSL Software Foundation*. <https://www.openssl.org/>
- Slides fornecidos pelo educando.