

Mestrado em Engenharia Informática (MEI)

Mestrado Integrado em Engenharia Informática

(MiEI)

Perfil de Especialização **CSI** : Criptografia e Segurança da Informação

Engenharia de Segurança



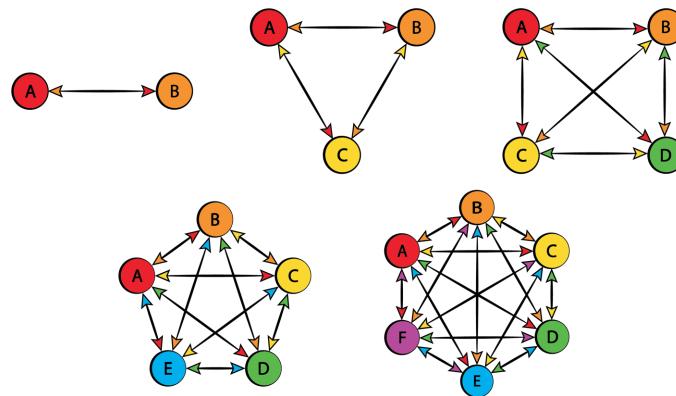
Tópicos

- **Parte VI: Acordo de chaves**
 - Protocolo Diffie-Hellman
 - Utilização
- Parte VII: Criptografia de chave pública
- Parte VIII: Infraestrutura de chave pública

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Acordo de chaves – motivação

- Utilização de criptografia (simétrica) obriga à existência de chaves partilhadas.
- Problema da distribuição de chaves:
 - Numa comunidade de n agentes, o estabelecimento de canais seguros (utilizando cifras simétricas) requer a partilha $\frac{n*(n-1)}{2}$ de chaves



- O pré-acordo de chaves é um procedimento custoso (requer a utilização de canais seguros...) e pouco flexível (e.g. considere-se a inclusão de mais um agente na comunidade...).



Acordo de chaves – motivação

Analogia com exemplos práticos sugere a possibilidade de alternativas viáveis...

- Exemplo: Admita-se que dispomos de uma cifra (simétrica) em que a operação de cifra (E) é comutativa, i.e.

$$E_{k1}(E_{k2}(X)) = E_{k2}(E_{k1}(X))$$

- Para *Alice* comunicar M com *Bob* pode:
 1. *Alice* envia a *Bob*: $E_{KA}(M)$ - em que KA é sé conhecida por *Alice*.
 2. *Bob* devolve a *Alice*: $E_{KB}(E_{KA}(M)) = E_{KA}(E_{KB}(M))$ - em que KB só é conhecida por *Bob*.
 3. *Alice* decifra mensagem recebida e reenvia a *Bob* o resultado, i.e. $E_{KB}(M)$
 4. *Bob* decifra mensagem M .

... ou seja, *Alice* e *Bob* comunicam de forma segura sem partilharem segredos... (a mensagem M circula sempre protegida com, pelo menos, uma operação de cifra)
- Obs.: mas este esquema também exibe uma vulnerabilidade importante... (c.f. *man-in-the-middle attack* que veremos adiante)



Tópicos

- **Parte VI: Acordo de chaves**
 - Protocolo Diffie-Hellman
 - Utilização
- Parte VII: Criptografia de chave pública
- Parte VIII: Infraestrutura de chave pública

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Acordo de chaves *Diffie-Hellman*

- Pode-se contornar o problema da distribuição de chaves se ambas as partes acordarem num segredo comum...
 - ...trocando mensagens sobre um canal público...
 - ...mas sem que seja possível derivar o segredo conhecendo apenas as mensagens trocadas.
- Um esquema que acomoda estes requisitos surgiu no artigo de Whitfield Diffie e Martin Hellman (*New Directions in Cryptography*, 1976, <https://ee.stanford.edu/~hellman/publications/24.pdf>).
- Segurança resulta de se acreditar que a exponenciação modular é uma função de sentido único.



Acordo de chaves *Diffie-Hellman*

Protocolo (efémero) *Diffie-Hellman*

- Parâmetros
 - Seja p um primo e g um gerador do grupo multiplicativo \mathbb{Z}_p^*
 (Nota: Dizemos que g é um gerador do grupo multiplicativo \mathbb{Z}_p^* quando qualquer um dos seus elementos pode ser escrito como g^x , para um dado inteiro x).
- Descrição
 1. *Alice* define p e g , e gera um inteiro $1 < a < p$, e envia a *Bob* p , g e $g^a \text{ mod } p$
 2. *Bob* gera um inteiro $1 < b < p$, e envia a *Alice* $g^b \text{ mod } p$
 3. *Bob* e *Alice* têm um segredo partilhado que podem começar a utilizar para cifrar a comunicação entre ambos:
 - *Alice* calcula: $(g^b \text{ mod } p)^a = (g^{ba} \text{ mod } p) = \boxed{(g^{ab} \text{ mod } p)}$
 - *Bob* calcula: $(g^a \text{ mod } p)^b = \boxed{(g^{ab} \text{ mod } p)}$

chave K

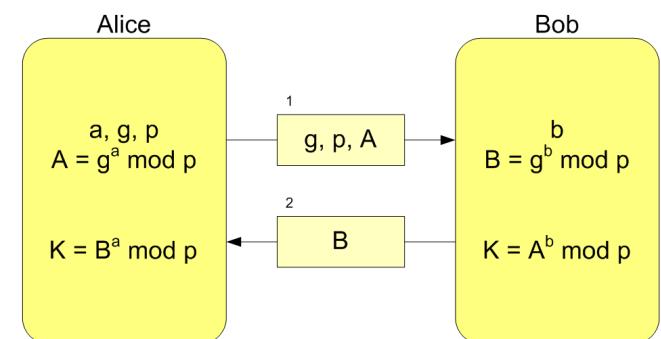


Acordo de chaves *Diffie-Hellman*

Protocolo (efémero) *Diffie-Hellman*

- Segurança
 - A segurança do protocolo exprime-se como uma assumpção de segurança própria (*Computational Diffie-Hellman problem*): sabendo g , g^a e g^b , é computacionalmente impossível determinar g^{ab} .
- Por vezes, os valores envolvidos no protocolo *Diffie-Hellman* são referidos como pares de chaves:
 - a, g^a : chave privada (a) e pública de *Alice* (g^a)
 - b, g^b : chave privada (b) e pública de *Bob* (g^b)

Alice				Bob		
Secreto	Público	Calcula	Envia	Calcula	Público	Secreto
a	p, g		$p, g \rightarrow$			b
a	p, g, A	$g^a \text{ mod } p = A$	$A \rightarrow$		p, g	b
a	p, g, A		$\leftarrow B$	$g^b \text{ mod } p = B$	p, g, A, B	b
a, s	p, g, A, B	$B^a \text{ mod } p = s$		$A^b \text{ mod } p = s$	p, g, A, B	b, s



$$K = A^b \text{ mod } p = (g^a \text{ mod } p)^b \text{ mod } p = g^{ab} \text{ mod } p = (g^b \text{ mod } p)^a \text{ mod } p = B^a \text{ mod } p$$

Imagens de https://pt.wikipedia.org/wiki/Troca_de_chaves_de_Diffie–Hellman



Acordo de chaves *Diffie-Hellman*

O protocolo *Diffie-Hellman* não garante autenticidade, o que possibilita ataques de *Man-in-the-middle* (i.e., na presença de um adversário activo, é possível este fazer-se passar por outro agente comprometendo a segurança da técnica/protocolo)

- Exemplo:
 - Suponhamos que *Alice* pretende acordar um segredo com *Bob*.
 - *Alice* gera x , calcula g^x e envia este último valor a *Bob*;
 - O *Intruso* intercepta a mensagem de *Alice*;
 - *Intruso* gera z e calcula g^z que envia para *Alice*:
 - *Alice* adopta o segredo $K = (g^z)^x = g^{xz}$ que presume acordado com *Bob*;
 - *Intruso* conhece o segredo $K = (g^x)^z = g^{xz}$ que *Alice* pensa partilhar com *Bob*.
- Este é um ataque a que estão sujeitas a generalidade das técnicas criptográficas assimétricas (que falaremos a seguir): A utilização de técnicas criptográficas assimétricas requer uma associação fidedigna entre pares de chaves e a identidades dos agentes comunicantes.

Tópicos

- **Parte VI: Acordo de chaves**
 - Protocolo Diffie-Hellman
 - Utilização
- Parte VII: Criptografia de chave pública
- Parte VIII: Infraestrutura de chave pública

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Utilização de acordo de chaves *Diffie-Hellman*

- O acordo de chaves *Diffie-Hellman* devem ser considerado quando for apropriado ao seu caso de uso.
- Não necessita (nem deve) desenvolver o código para as funções de acordo de chaves, já que existem bibliotecas/APIs que já disponibilizam o código necessário (i.e., as operações base das funções de acordo de chaves). Por exemplo:
 - Em Python, pode utilizar a cryptography (<https://cryptography.io/>);
 - Em Javascript ou Node.js pode utilizar o crypto (<https://nodejs.org/api/crypto.html>).
 - Em Java, tal como referido para as cifras simétricas, pode utilizar
 - os *default providers* da Sun (propriedade da Oracle), nomeadamente SUN, SunJCE, SunPKCS11, ...;
 - O *provider* do Bouncy Castle (<https://www.bouncycastle.org/java.html>).

Utilização de acordo de chaves *Diffie-Hellman*

- Exemplo em python, utilizando o cryptography

```
from cryptography.hazmat.primitives import hashes
from cryptography.hazmat.primitives.asymmetric import dh
from cryptography.hazmat.primitives.kdf.hkdf import HKDF
# Alice define g e tamanho de p
g = 2
key_size = 2048
# Alice inicializa parâmetros do Diffie-Hellman
alice_parameters = dh.generate_parameters(generator=g, key_size=key_size)
# Alice obtém p (para enviar a Bob)
p = alice_parameters.parameter_numbers().p
# Alice gera a e ga
a = alice_parameters.generate_private_key()
ga = a.public_key()
#
# Alice envia a Bob p, g e ga (ou seja g, p e ga)
#
# Bob inicializa parâmetro do Diffie-Hellman e gera b e gb
bob_parameters = dh.DHParameterNumbers(p, g).parameters()
b = bob_parameters.generate_private_key()
gb = b.public_key()
# Bob obtém a shared key, a partir de gb e ga
bob_sharedkey = b.exchange(ga)
#
# Bob envia a Alice gb (ou seja gb)
#
# Alice obtém a shared key a partir de ga e gb
alice_sharedkey = a.exchange(gb)
```



Utilização de acordo de chaves *Diffie-Hellman* – openssl

- O openssl (<https://www.openssl.org>) é um toolkit ("canivete suíço") para criptografia e comunicações seguras.
 - Diffie-Hellman, utilizando a linha de comando (windows, linux, macos, ...)

```
# geração do g e o p necessários ao Diffie-Hellman  
openssl genpkey -genparam -algorithm DH -out dhp.pem
```

```
# Veja o g e o p no ficheiro  
openssl pkeyparam -in dhp.pem –text
```

```
debian@vm5: /tmp$ openssl pkeyparam -in dhp.pem -text  
-----BEGIN DH PARAMETERS-----  
MIIBCAKCAQEATfguo42JDB62Mm5VTG1n5bZ475LV+i8pSgLngbdhE0vWPSM06AA  
Muop0YU4exm9NwJLsWNI9js1X/FDMLmFNy/ec9fM4riadMV+cvxfhGXMrLNQUzK37  
bwY3+EeeyG+EBBPHg+l0pkRJrWxJuW2p1Jy+3ekdPo08GBBLPZ95WFm+N/M2jXXD  
1bwjYg9ZSi1rI5raBMZbByfx5CXXN7aKrHUYray47fG5k0aVUNX+FYadabn+7Rd/  
7PA99fGH2bf1K8T2iZXsDyzDm0h5JQwNSgNEFYsE6nxg9eTQznS9Pe2far5ls6a3S  
xq1Hau4aDtS9su0Qp24PV3A2b3W07XvPUwIBAg==  
-----END DH PARAMETERS-----  
DH Parameters: (2048 bit)  
prime:  
00:ed:f8:2e:a3:8d:89:0c:1e:b6:32:6e:55:4c:6d:  
67:e5:b6:78:ef:92:d5:fa:2f:29:4a:02:e5:9e:06:  
dd:84:43:af:58:f4:8c:3b:a0:00:32:ea:29:d1:85:  
38:7b:19:bd:37:02:4b:b1:63:48:f6:3b:35:5f:f1:  
43:30:b9:85:37:2f:de:73:d7:cc:e2:b8:9a:31:5f:  
9c:c6:f7:e1:19:73:2b:94:d4:14:cc:ad:fb:6f:0c:  
b7:f8:47:9e:c8:6f:84:04:13:c7:83:e9:74:a6:44:  
49:ad:6c:49:b9:6d:a9:d4:9c:be:dd:e9:1d:3e:8d:  
3c:18:10:65:3d:9f:79:59:f9:be:37:f3:36:8d:7c:  
43:95:bc:23:62:0f:59:4a:2a:c8:e6:b6:81:31:96:  
c1:cb:27:d7:e4:25:cd:5b:b6:8a:ac:75:18:45:ac:  
b8:ed:f1:b9:93:46:95:50:d5:fe:15:86:9d:69:b9:  
fe:ed:17:7f:ec:f0:3d:f5:f1:87:d9:b7:e2:2b:c4:  
f6:89:95:ec:0f:c3:98:e8:79:25:0c:0d:4a:03:  
44:15:87:ba:9f:18:3d:79:34:33:9d:2f:4f:7b:67:  
da:af:99:6c:e9:ad:d2:c6:ad:47:6a:ee:1a:0e:d4:  
bd:b1:43:90:a7:6e:0f:57:70:36:6f:75:b4:ed:7b:  
cf:53  
-----  
generator: 2 (0x2)
```

p

g

Utilização de acordo de chaves *Diffie-Hellman* – openssl

- O openssl (<https://www.openssl.org>) é um toolkit ("canivete suíço") para criptografia e comunicações seguras.
 - Diffie-Hellman, utilizando a linha de comando (windows, linux, macos, ...)

Alice gera o a e o g^a

```
openssl genpkey -paramfile dhp.pem -out alice.pem
```

Veja o a e o g^a no ficheiro

```
openssl pkey -in alice.pem -text -noout
```

```
debian@vm5:/tmp$ openssl pkey -in alice.pem -text -noout
DH Private-Key: (2048 bit)
private-key:
44:75:f1:1a:66:04:10:70:4a:29:01:ec:2d:ce:30:
36:6b:5f:e8:0f:4f:a0:e7:47:9a:25:b0:a2:4b:0b:
c1:1d:5f:af:2e:e4:67:53:fe:9e:4a:2e:39:b1:f1:
e6:f8:3c:91:e4:77:b4:12:b8:0f:7f:7d:f7:82:77:
60:08:41:e2:28:02:ff:57:62:c0:c5:7d:d5:69:9e:
e5:ac:ed:9:1b:56:39:3c:97:3c:ze:5b:3b:fc:f1:
f8:a3:e2:fd:2f:80:c8:a5:84:8c:06:7e:64:8b:a4:
e2:2a:7e:f3:e7:38:64:46:85:1a:3c:d3:04:ad:41:
e6:f2:a2:b7:1a:55:8d:49:8e:d0:6d:99:02:4a:db:
39:08:15:fa:75:47:08:eb:b4:e1:35:b0:85:5b:20:
3d:7a:10:93:8c:61:34:99:ae:51:ad:0c:92:bd:64:
92:26:5d:6f:e1:61:0b:aa:a6:16:f6:c2:6e:00:c7:
b4:cc:a7:ba:a9:b9:38:cb:8b:19:80:8d:c4:a2:27:
9e:08:44:38:47:54:84:5c:c0:b0:8c:e0:f2:29:38:
04:29:11:6e:b0:71:5f:24:d9:18:e9:d1:19:02:4f:
89:73:c3:aa:29:78:c5:1e:03:49:3d:8e:ba:f3:52:
d3:f1:83:2a:8b:16:cb:07:57:ee:f0:16:ed:f5:0d:
c0
public-key:
5b:78:f3:a4:cf:f1:da:14:00:c8:eb:ec:66:a7:a6:
37:e5:20:54:90:eb:f9:f0:e3:e9:9d:f2:44:06:67:
7c:74:75:d5:9e:7d:bc:35:e1:64:32:0e:5f:4c:c8:
38:20:7b:10:6d:98:24:1b:3e:5b:6c:b2:74:75:da:
1d:30:49:33:70:67:4e:d0:ae:a9:c2:d1:66:3d:ff:
54:37:c7:e3:24:ff:35:73:40:7e:b7:64:73:4d:c6:
ad:f4:50:47:a5:04:41:2c:62:7e:19:9d:71:3b:38:
d5:71:e3:f6:00:03:13:ee:59:f3:ad:df:40:7b:8b:
31:71:4c:e4:63:cb:75:d4:1f:3e:97:58:67:7b:02:
5d:37:fb:e3:7f:61:71:8c:77:3a:56:a3:1b:11:52:
23:5c:c7:8d:fa:54:53:03:53:0f:32:99:52:82:c7:
07:9b:31:6d:85:22:96:22:40:cd:cd:8d:3a:d3:31:
5c:6d:9a:e8:64:8e:b3:64:bd:57:bf:02:b5:5a:81:
1b:1e:08:4e:ac:7f:14:f9:6d:a8:09:f9:5d:e9:83:
e8:49:37:67:07:fa:fb:ce:86:60:b3:2e:d6:cc:19:
96:12:20:58:08:32:be:f1:10:0e:fb:fb:d4:eb:38:
5c:d1:ac:1f:4c:8c:f5:3e:59:5e:6e:78:55:09:f5:
0e
```

a

g^a

Utilização de acordo de chaves *Diffie-Hellman* – openssl

- O openssl (<https://www.openssl.org>) é um toolkit ("canivete suíço") para criptografia e comunicações seguras.
 - Diffie-Hellman, utilizando a linha de comando (windows, linux, macos, ...)

Alice gera o a e o g^a

```
openssl genpkey -paramfile dhp.pem -out alice.pem
```

Veja o a e o g^a no ficheiro

```
openssl pkey -in alice.pem -text -noout
```

Alice extrai o g^a

```
openssl pkey -in alice.pem -pubout -out alice_ga.pem
```

Veja o g^a , p e g

```
openssl pkey -pubin -in alice_ga.pem -text
```

```
debian@vm5:/tmp$ openssl pkey -pubin -in alice_ga.pem -text
DH Public-Key: (2048 bit)
public-key:
5b:78:f3:a4:cf:f1:da:14:00:c8:eb:ec:66:a7:a6:
37:e5:20:54:90:eb:f9:f0:e3:e9:9d:f2:44:06:d7:
7c:74:75:d5:9e:7d:bc:35:e1:64:32:0e:5f:4c:c8:
38:20:7b:10:6d:98:24:1b:3e:5b:6c:b2:74:75:da:
1d:30:49:33:70:67:4e:d0:ae:a9:c2:d1:66:3d:ff:
54:37:c7:e3:24:ff:35:73:40:7e:b7:64:73:4d:c6:
ad:f4:50:47:a5:04:41:2c:62:7e:19:9d:71:3b:38:
d5:71:e3:f6:00:03:13:ee:59:f3:ad:df:40:7b:8b:
31:71:4c:e4:63:cb:75:d4:1f:3e:97:58:67:7b:02:
5d:37:fb:e3:7f:61:71:8c:77:3a:56:a3:1b:11:52:
23:5c:cf:8d:fa:54:53:03:53:0f:32:99:52:82:cf:
07:9b:31:6d:85:22:96:22:40:cd:cd:8d:3a:d3:31:
5c:6d:9a:e8:64:8e:b3:64:bd:57:bf:02:b5:5a:81:
1b:1e:08:4e:ac:7f:14:f9:6d:a8:09:f9:5d:e9:83:
e8:49:37:67:07:fa:fb:ce:86:60:b3:2e:d6:cc:19:
96:12:20:58:a8:32:be:f1:10:0e:fb:fb:d4:eb:38:
5c:d1:ac:1f:4c:8c:f5:3e:59:5e:6e:78:55:09:f5:
0e
prime:
00:ed:f8:2e:a3:8d:89:0c:1e:b6:32:6e:55:4c:6d:
67:e5:b6:78:ef:92:d5:fa:2f:29:4a:02:e5:9e:06:
dd:84:43:af:58:f4:8c:3b:a0:00:32:ea:29:d1:85:
38:7b:19:b3:37:02:4b:b1:63:48:f6:3b:5f:f1:
43:30:b9:85:37:2f:de:73:d7:cc:e2:b8:9a:31:5f:
9c:c6:f7:e1:19:73:2b:94:d4:14:cc:ad:fb:6f:0c:
b7:f8:47:9e:c8:6f:84:04:13:c7:83:e9:74:a6:44:
49:ad:6c:49:b9:6d:a9:44:9c:be:dd:e9:1d:3e:8d:
3c:18:10:65:3d:9f:79:59:f9:be:37:f3:36:8d:7c:
43:95:bc:23:62:0f:59:a4:2a:c8:e6:b6:81:31:96:
c1:cb:27:d7:e4:25:cd:5b:b6:8a:ac:75:18:45:ac:
b8:ed:f1:b9:93:46:95:50:d5:fe:15:86:9d:69:b9:
fe:ed:17:7f:ec:f0:3d:f5:f1:87:d9:b2:e2:b2:c4:
f6:89:95:ec:0f:2c:c3:98:e8:79:25:0c:0d:4a:03:
44:15:87:ba:9f:18:3d:79:34:33:9d:2f:4f:7b:67:
da:af:99:6c:e9:ad:d2:c6:ad:47:6a:ee:1a:0e:d4:
bd:b1:43:90:a7:6e:0f:57:70:36:6f:75:b4:ed:7b:
cf:53
generator: 2 (0x2)
generator: 2 (0x2)
```

g^a

p

g



Utilização de acordo de chaves *Diffie-Hellman* – openssl

- O openssl (<https://www.openssl.org>) é um toolkit ("canivete suíço") para criptografia e comunicações seguras.
 - Diffie-Hellman, utilizando a linha de comando (windows, linux, macos, ...)

```
# Alice envia ga, p e g ao Bob (i..e, ficheiros dhp.pem e alice_ga.pem)
```

```
# Bob gera b e o gb
openssl genpkey -paramfile dhp.pem -out bob.pem
```

```
# Veja o b e o gb no ficheiro
openssl pkey -in bob.pem -text -noout
```

```
# Bob extrai o gb
openssl pkey -in bob.pem -pubout -out bob_gb.pem
```

```
# Veja o gb, p e g
openssl pkey -pubin -in bob_gb.pem -text
```

Utilização de acordo de chaves *Diffie-Hellman* – openssl

- O openssl (<https://www.openssl.org>) é um toolkit ("canivete suíço") para criptografia e comunicações seguras.
 - Diffie-Hellman, utilizando a linha de comando (windows, linux, macos, ...)

```
# Bob envia gb a Alice (i..e, ficheiro bob_gb.pem)
```

```
# A partir deste momento, Bob e Alice podem gerar a chave partilhada
```

```
# Alice gera a chave partilhada
```

```
openssl pkeyutl -derive -inkey alice.pem -peerkey bob_gb.pem -out secret1.bin
```

```
# Bob gera a chave partilhada
```

```
openssl pkeyutl -derive -inkey bob.pem -peerkey alice_ga.pem -out secret2.bin
```

Utilização de acordo de chaves *Diffie-Hellman* – openssl

- O openssl (<https://www.openssl.org>) é um toolkit ("canivete suíço") para criptografia e comunicações seguras.
 - Diffie-Hellman, utilizando a linha de comando (windows, linux, macos, ...)

```
# Comparando as chaves partilhadas geradas pelo Bob e pela Alice  
# Comparaçao byte a byte
```

```
debian@vm5:/tmp$ cmp secret1.bin secret2.bin  
debian@vm5:/tmp$
```

Utilização de acordo de chaves *Diffie-Hellman* – openssl

- O openssl (<https://www.openssl.org>) é um toolkit ("canivete suíço") para criptografia e comunicações seguras.
 - Diffie-Hellman, utilizando a linha de comando (windows, linux, macos, ...)

```
# Comparando as chaves partilhadas geradas pelo Bob e pela Alice  
# Fazendo um hexdump dos dois ficheiros
```

```
debian@vm5:/tmp$ xxd secret1.bin  
00000000: 0aea 04f8 cc86 2ebf bd1a a870 d57b 78d3 .....p.{x.  
00000010: 1a43 8731 afd5 d643 acc6 d1f6 476c db8e .C.1...C....G1..  
00000020: db57 2075 9459 df88 49cb 9394 7c4f 398e .W u.Y..I...109.  
00000030: 38ec 2e11 383e 84e9 1eb6 76e9 4ba0 b741 8...8>....v.K..A  
00000040: bf86 f00e 8369 cdea 0534 2c4b f33a da7b .....i...4,K.:.{  
00000050: 5d1a 65c7 02d2 3efb 7342 f2e2 91e6 cf64 ].e...>.sB.....d  
00000060: 287a 3d08 6371 3d5d efce 9f85 a84a bd59 (z=.cq=]....J.Y  
00000070: 6446 1181 4488 6089 480c 656d 20d7 28b0 dF..D.`.H.em .C.  
00000080: 92fb 878a 03ea f228 9a73 6a8c ea70 b075 .....(sj..p.u  
00000090: 9d78 6dbb aeb9 be8f c580 ba21 070a 845b .xm.....!....[  
000000a0: 6e7e 3acf 18ad 7ddb a9e8 ed3c ff1e a6f3 n~:....}....<....  
000000b0: 04ef 6447 0b92 3641 5d7b ffd3 740b 5485 ..dG..6A]{..t.T.  
000000c0: 17b9 ad30 42e9 d8aa d113 a825 f5a4 7e78 ...0B.....%..~x  
000000d0: 0456 1910 68d8 d1a1 4e3b 58de e7cb 69c6 .V..h...N;X...i.  
000000e0: 3b41 38bf f423 6f6a 7212 f2fe 2900 a3a0 ;A8..#ojr...)...  
000000f0: b218 e441 423e cc10 398f a0d8 1579 0b94 ...AB>..9....y...
```

```
debian@vm5:/tmp$ xxd secret2.bin  
00000000: 0aea 04f8 cc86 2ebf bd1a a870 d57b 78d3 .....p.{x.  
00000010: 1a43 8731 afd5 d643 acc6 d1f6 476c db8e .C.1...C....G1..  
00000020: db57 2075 9459 df88 49cb 9394 7c4f 398e .W u.Y..I...109.  
00000030: 38ec 2e11 383e 84e9 1eb6 76e9 4ba0 b741 8...8>....v.K..A  
00000040: bf86 f00e 8369 cdea 0534 2c4b f33a da7b .....i...4,K.:.{  
00000050: 5d1a 65c7 02d2 3efb 7342 f2e2 91e6 cf64 ].e...>.sB.....d  
00000060: 287a 3d08 6371 3d5d efce 9f85 a84a bd59 (z=.cq=]....J.Y  
00000070: 6446 1181 4488 6089 480c 656d 20d7 28b0 dF..D.`.H.em .C.  
00000080: 92fb 878a 03ea f228 9a73 6a8c ea70 b075 .....(sj..p.u  
00000090: 9d78 6dbb aeb9 be8f c580 ba21 070a 845b .xm.....!....[  
000000a0: 6e7e 3acf 18ad 7ddb a9e8 ed3c ff1e a6f3 n~:....}....<....  
000000b0: 04ef 6447 0b92 3641 5d7b ffd3 740b 5485 ..dG..6A]{..t.T.  
000000c0: 17b9 ad30 42e9 d8aa d113 a825 f5a4 7e78 ...0B.....%..~x  
000000d0: 0456 1910 68d8 d1a1 4e3b 58de e7cb 69c6 .V..h...N;X...i.  
000000e0: 3b41 38bf f423 6f6a 7212 f2fe 2900 a3a0 ;A8..#ojr...)...  
000000f0: b218 e441 423e cc10 398f a0d8 1579 0b94 ...AB>..9....y...
```

Tópicos

- Parte VI: Acordo de chaves
- **Parte VII: Criptografia de chave pública**
 - Cifra assimétrica
 - Assinatura Digital
 - Algoritmo RSA
 - Algoritmo EL-Gamal
 - Criptografia de curvas elípticas
 - Utilização
- Parte VIII: Infraestrutura de chave pública

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Criptografia de chave pública – motivação

- Conceito introduzido por *Diffie & Hellman* em 1976.
- Ideia base:
 - Duas chaves distintas são utilizadas na operação de cifra Kc e de decifragem Kd .
$$E(Kd, E(Kc, M)) = M$$
 - O conhecimento de uma chave não permite retirar informação sobre a outra.
 - Cifra com uma das chaves deve ser uma função de sentido único – não deve ser computacionalmente viável inverter essa função.
 - Mas informação adicional (outra chave) permite calcular operação inversa...
- ...leva ao conceito de *Trapdoor function* ...
- ...em que uma das chaves pode ser “tornada pública” ...

Criptografia de chave pública – cifra assimétrica

- A utilização de chaves distintas para as operações de cifra e decifragem permite contornar o problema da pé-distribuição de chaves.
- O ponto de partida é que só a chave para decifrar necessita ser mantida secreta.
- Assim:
 - Cada agente dispõe de um par de chaves (Kc , Kd)

Cifra:

- Chave pública: Kc ; Chave privada: Kd
- Para *Alice* (A) enviar mensagem M a *Bob* (B): envia $E(Kc^B, M)$ - note que Kc^B é publicamente conhecida...
- *Bob* decifra a mensagem utilizando a sua chave privada: $E(Kd^B, E(Kc^B, M)) = M$

A dispõe de garantias que só *Bob* pode extrair o conhecimento de M porque só ele dispõe do conhecimento da chave privada.

Criptografia de chave pública – Utilização (na prática)

- Para o mesmo nível de segurança, as cifras assimétricas são várias ordens de grandeza menos eficientes do que as cifras simétricas (e.g. 1000x).
- ... por isso, são normalmente utilizadas em conjunção com estas (e não alternativamente).
- Utilização típica:
Envelope digital – utilizado para garantir confidencialidade na transmissão de uma mensagem
 - *Alice* gera uma chave de sessão K (para uma cifra simétrica)
 - *Alice* envia a *Bob* par com $E(Kc^B, K)$ e $E_K(M)$ – Note que $E_K()$ é uma cifra simétrica
 - *Bob* decifra $E(Kc^B, K)$ com a sua chave privada $E(Kd^B, E(Kc^B, K)) = K$, e utiliza K para decifrar M .

Criptografia de chave pública – *Man-in-the-middle*

Tal como no caso do acordo de chaves, também a cifra assimétrica é vulnerável perante um adversário activo (ataque *Man-in-the-middle*).

- Na sua essência, este ataque traduz-se por fazer uso da chave pública “errada”.
- Exemplo:
 - Suponhamos que *Alice* deseja cifrar uma mensagem para *Bob*.
 - Ao pedido de *Alice* relativo à chave pública de *Bob*, o *Intruso* (*I*) responde com a sua própria chave pública Kc^I .
 - *Alice* envia $E(Kc^I, M)$...
 - *Intruso* intercepta essa mensagem, decifra-a, e torna-a a cifrar utilizando a verdadeira chave pública de *Bob* Kc^B .
 - *Bob* decifra mensagem...

Alice e *Bob* supõem que M se mantém secreta mas *Intruso* decifrou a mensagem sem problemas...

- Mais uma vez observa-se que existe necessidade de confiar na associação entre os pares de chaves e as identidades: A utilização de técnicas criptográficas assimétricas requer uma associação fidedigna entre pares de chaves e as identidades dos agentes comunicantes.

Tópicos

- Parte VI: Acordo de chaves
- **Parte VII: Criptografia de chave pública**
 - Cifra assimétrica
 - **Assinatura Digital**
 - Algoritmo RSA
 - Algoritmo EL-Gamal
 - Criptografia de curvas elípticas
 - Utilização
- Parte VIII: Infraestrutura de chave pública

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Criptografia de chave pública – Assinatura Digital

O principal contributo da criptografia assimétrica foi o de permitir a definição de um *análogo digital* do conceito de assinatura de um documento.

- Em geral, podemos identificar uma assinatura digital como um “suplemento” à mensagem que nos permite verificar:
 - **Integridade**: a mensagem não é modificada após a assinatura;
 - **Autenticidade**: a identidade do *assinante* pode ser confirmada;
 - **Não repúdio**: é possível demonstrar a identidade do assinante.

Mais uma vez observa-se que existe necessidade de confiar na associação entre os pares de chaves e as identidades: A utilização de técnicas criptográficas assimétricas requer uma associação fidedigna entre pares de chaves e as identidades dos agentes comunicantes.

Assinatura Digital – Descrição

- Na utilização de uma assinatura estão envolvidas duas entidades: o **(S)ignatário** e o **(V)erificador**.
- Um esquema de assinaturas comprehende duas operações:
 - **produção da assinatura**: processo pelo qual o **Signatário** gera a assinatura $x = \text{Sig}^S(M)$ que anexa à mensagem. A mensagem assinada consiste assim num par (M, x) ;
 - **verificação da assinatura**: processo em que o **Verificador** confirma que o originante da mensagem M é S , i.e.
$$\text{Ver}^S(M, x) = \text{true}$$
- Das propriedades requeridas pela assinatura resulta que, se o **(S)ignatário** produzir uma assinatura $x = \text{Sig}^S(M)$, o **(V)erificador** com o par (M, x) :
 - pode verificar que o originante de M é S , i.e. $\text{Ver}^S(M, x) = \text{true}$
 - não pode produzir $M' \neq M$ tal que $\text{Ver}^S(M', x) = \text{true}$

Obs.1: na essência do conceito de assinatura digital está uma assimetria entre as capacidades do verificador e do signatário: o primeiro (verificador) deve estar habilitado a verificar as assinaturas produzidas pelo segundo (signatário), sem dispor da capacidade de, ele próprio, as produzir.

Obs.2: note que os MACs garantem os dois primeiros requisitos da assinatura digital (integridade e autenticação) mas falham no último (não repúdio)



Assinatura Digital – Utilização básica (conceito)

- Em relação à cifra assimétrica, as operações num esquema de assinaturas são:
 - A produção da assinatura é restrita ao Signatário;
 - A verificação pode ser pública.
- Assim é concebível trocar os papéis das chaves públicas e privadas nas cifras assimétricas para codificar um esquema de assinatura:
 - Cada agente X dispõe de um par de chaves (Kc^X, Kd^X)
 - Chave pública: Kd^X Chave privada: Kc^X
 - $Sig^X(M) = E(M, Kc^X) = S$
 - $Ver^X(M, S) = (E(S, Kd^X) == M)$

Bob (ou qualquer agente) dispõe de garantias que M foi realmente enviada por X porque só ele dispunha da chave privada com que efetuou a assinatura.

Assinatura Digital – Utilização básica (conceito)

- Em relação à cifra assimétrica, as operações num esquema de assinaturas são:
 - A produção da assinatura é restrita ao Signatário;
 - A verificação pode ser pública.
- Assim é concebível trocar os papéis das chaves públicas e privadas nas cifras assimétricas para codificar um esquema de assinatura:

Mais uma vez observa-se que existe necessidade de confiar na associação entre os pares de chaves e as identidades: A utilização de técnicas criptográficas assimétricas requer uma associação fidedigna entre pares de chaves e a identidades dos agentes comunicantes.

Bob (ou qualquer agente) dispõe de garantias que M foi realmente enviada por X porque só ele dispunha da chave privada com que efetuou a assinatura.

Assinatura Digital – Utilização (na prática)

- As considerações expostas anteriormente relativamente à eficiência das técnicas assimétricas...
- (...assim como outras relativas a aspectos de segurança...)
- ...faz com que se combine o padrão apresentado com a utilização de uma função de *hash criptográfica*..
- Assim, na prática temos:
 - *Alice* utiliza uma função de hash criptográfica para calcular $H = \text{hash}(M)$
 - *Alice* envia a *Bob* o par constituído por (M, S) , sendo $S = E(H, Kc^A) = \text{Sig}^A(H)$.
 - *Bob*
 - determina o valor de hash $H' = \text{hash}(M)$, e
 - compara-o com resultado da decifragem de S , i.e.,
$$\text{Ver}^A(H', S) = (E(S, Kd^A) == H')$$

Assinatura Digital – *Man-in-the-middle*

- Tal como as restantes técnicas assimétricas, também as assinaturas digitais são vulneráveis ao ataque *man-in-the-middle*.
- Na assinatura, esse ataque traduz-se na falha de garantias de autenticação após a verificação da assinatura (a verificação é realizada com uma chave pública “errada”)
- Mas é interessante observar que desta vez existe um certo grau de circularidade entre o que é o objectivo da técnica e a causa do problema:
 - a assinatura digital pretende estabelecer a autenticidade de uma mensagem/documento;
 - e a falha na garantia de autenticidade da associação entre as chaves públicas e a identidade leva à possibilidade do ataque *man-in-the-middle*.
 - ...ora, se considerar um documento que estabeleça essa associação...
 - ...podemos utilizar uma assinatura digital para **certificar** esse documento (assunto que abordaremos adiante)

Assinatura Digital – Certificação das chaves

Tal como já tem vindo a ser referido, nunca devemos utilizar cifras assimétricas sem uma confiança plena na associação entre pares de chaves e identidades dos agentes...

- Evidentemente que tal garantia pode ser conseguida por uma pré-distribuição de chaves (mas então temos um problema similar ao que já vimos nas cifras simétricas...)
- Solução alternativa consiste em utilizar os próprios mecanismos disponibilizados pelas técnicas assimétricas (em particular a assinatura digital) para estabelecer a confiança entre as associações par-de-chaves/identidades
 - Todos os agentes dispõem da chave pública de um agente fidedigno - a Entidade de Certificação (ou *Certification Authority*). Essa chave pública deve ser obtida por via de um canal seguro...
 - A Entidade de Certificação (EC) garante (assinando digitalmente) a associação entre chave pública/identidade do agente - o que designamos por certificado de chave pública ou **certificado digital**. É responsabilidade da EC garantir a correção da associação estabelecida, i.e., garantir a identificação do agente e a sua correta associação à respetiva chave pública.
 - Um qualquer agente (*relying party*) pode verificar a assinatura de um certificado (atestando assim a validade da associação pretendida).

Voltaremos a este assunto, com mais detalhe, na Parte VIII.

Tópicos

- Parte VI: Acordo de chaves
- **Parte VII: Criptografia de chave pública**
 - Cifra assimétrica
 - Assinatura Digital
 - **Algoritmo RSA**
 - Algoritmo EL-Gamal
 - Criptografia de curvas elípticas
 - Utilização
- Parte VIII: Infraestrutura de chave pública

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Criptografia de chave pública – Algoritmo RSA

- Algoritmo que realiza o conceito de criptografia de chave pública introduzido por *Diffie & Hellman*.
- Algoritmo RSA desenvolvido por *Ron Rivest, Adi Shamir & Leonard Adleman* - 1977/8.
- Baseada no problema da factorização de inteiros.

RSA – breve descrição matemática

Teoria de Números

- Função totient $\varphi(n)$ de Euler: em Z_n , o conjunto de valores $0 \leq x < n$ são designados por **resíduos**. Aos resíduos que não dispõem de factores em comum com n dizemos tratarem-se de **resíduos reduzidos**. A função totient de Euler $\varphi(n)$ é definida como o número de resíduos reduzidos de n .
 - Se p é primo, então $\varphi(p) = p - 1$
 - Se $n = p * q$ com p, q primos, então $\varphi(n) = (p - 1)(q - 1)$
- Teorema (pequeno) de Fermat: (com p primo, $0 < a < n$)
$$a^{p-1} \text{mod } p \equiv 1$$
- ...ou na versão generalizada de Euler, (com $\gcd(a, n) = 1$)
$$a^{\varphi(n)} \text{mod } n \equiv 1$$

RSA – breve descrição matemática

- Inicialização (produção do par de chaves)
 - Geram-se dois números primos grandes p, q
 - (e faz-se $n = p * q$, logo $\varphi(n) = (p - 1) * (q - 1)$)
 - Considera-se um valor e que seja primo relativo a $\varphi(n)$ (i.e. $\gcd(e, \varphi(n)) = 1$).
 - Calcula-se d como a inversa de e no grupo multiplicativo $Z_{\varphi(n)}^*$, i.e. $e * d = 1 \bmod \varphi(n)$.

Chave para cifrar: (n, e)

Chave para decifrar: (n, d)

- Utilização (como cifra)
 - Ambas as operações são a exponenciação modular.
 - **Cifra** do texto limpo x ($0 \leq x < n$) com chave (n, e) :

$$y = x^e \bmod n$$

- **Decifragem** do criptograma y ($0 \leq y < n$) com chave (n, d)

$$y^d \bmod n$$



RSA – segurança

- Derivar chave privada da chave pública:
 - É possível definir um algoritmo (probabilístico) que permite calcular a factorização de n , assumindo que dispomos de um oráculo para derivar a chave privada RSA a partir da chave pública. Ou seja, os problemas são demonstrados equivalentes...
- Extrair mensagem do criptograma:
 - Se se escolher uma mensagem arbitrária (de entre todo o espaço de mensagens admissíveis), “acredita-se” que não é possível derivar essa mensagem do criptograma respetivo.
- (Não) Indistinguibilidade de mensagens:
 - Mas é muito simples derivar a mensagem cifrada se se souber que ela pertence a um conjunto restrito de possibilidades (e.g. um único bit).

RSA – variantes aleatórias

- As maiores críticas apontadas ao RSA resultam de ele ser determinístico (i.e. uma dada mensagem cifrada repetidas vezes resulta sempre no mesmo criptograma).
- Já vimos que este facto pode comprometer completamente a segurança de uma técnica criptográfica em determinadas utilizações.
- Existem variantes aleatórias do RSA que ultrapassam esta limitação, prevendo a utilização de factores aleatórios na produção do criptograma (ou assinatura).
- É possível demonstrar (formalmente) que essas variantes cumprem requisitos de segurança mais apertados (e.g. indistinguibilidade).
- Exemplos:
 - Cifra: RSA-OAEP
 - Assinatura: RSA-PSS

RSA – Cifra RSA-OAEP

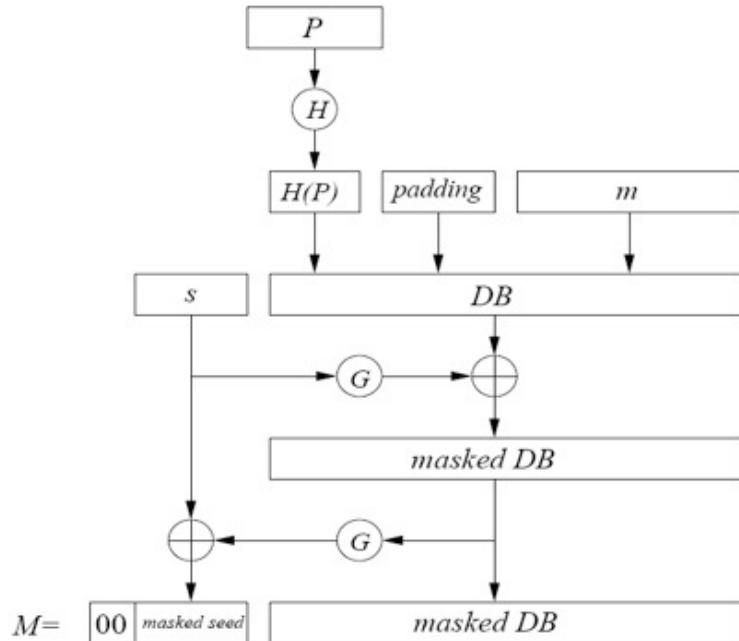


Figure 1: OAEP encoding function.

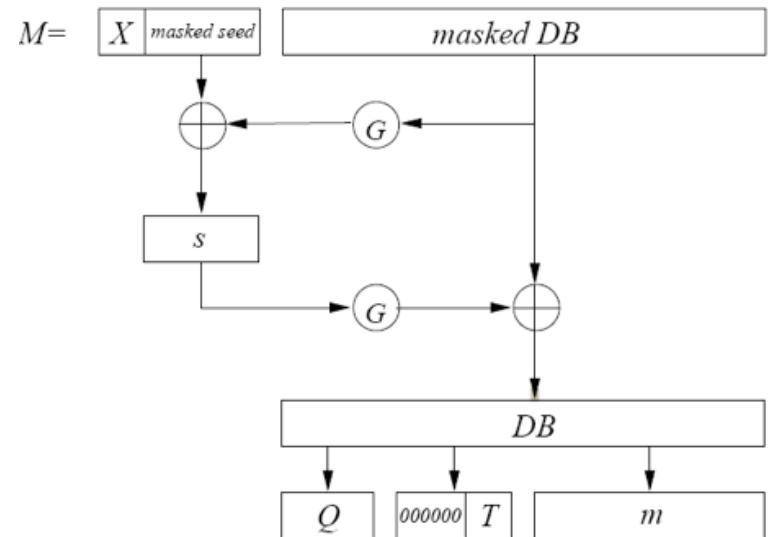
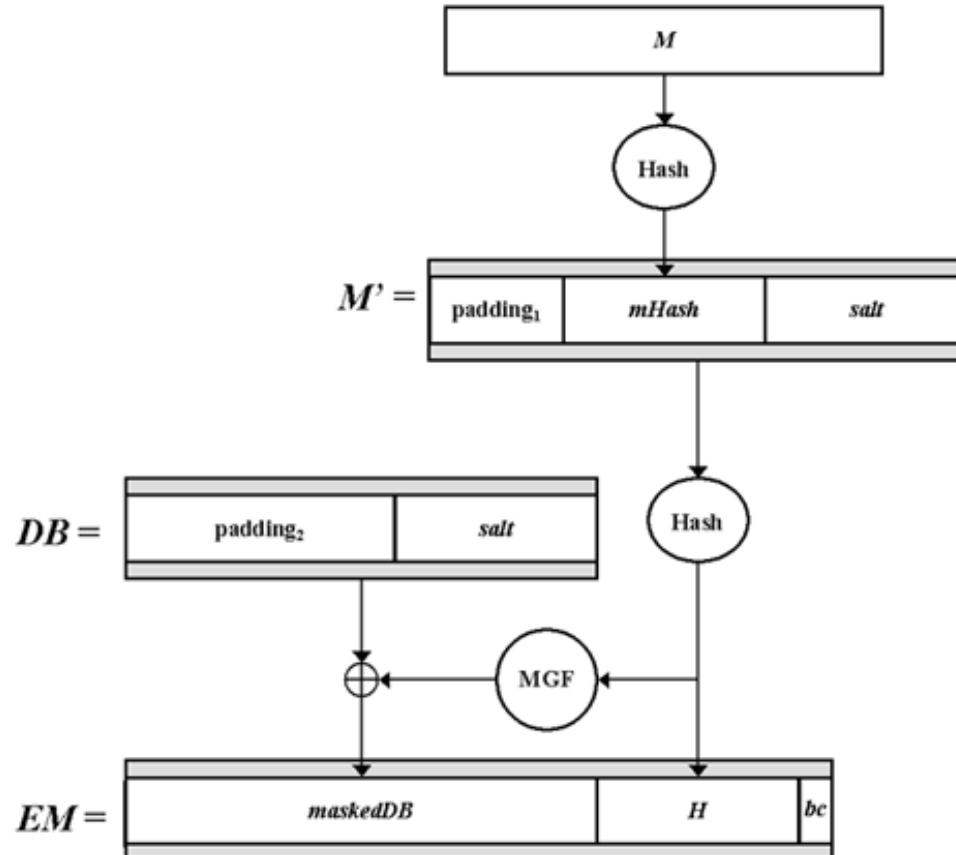


Figure 2: OAEP decoding function.

RSA – Assinatura RSA-PSS



Tópicos

- Parte VI: Acordo de chaves
- **Parte VII: Criptografia de chave pública**
 - Cifra assimétrica
 - Assinatura Digital
 - Algoritmo RSA
 - **Algoritmo EL-Gamal**
 - Criptografia de curvas elípticas
 - Utilização
- Parte VIII: Infraestrutura de chave pública

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Criptografia de chave pública – Algoritmo EL-Gamal

- Algoritmo introduzido em 1984 por *T. El Gamal*.
- Baseado no problema do logaritmo discreto.
- Variantes para funcionar como cifra ou como assinatura....

EL-Gamal – breve descrição matemática

- Inicialização (produção do par de chaves)
 - Escolher um primo p e dois inteiros, g e x , tal que g é gerador de Z_p^* e $x < p$
 - Calcular $y = g^x \text{ mod } p$
 - [chave privada, chave pública] = $[x, (y, g, p)]$
- Cifra de uma mensagem M
 - Escolher (aleatoriamente) um inteiro k , $0 < k < p - 1$
 - tal que k não foi já utilizado e $\gcd(k, p - 1) = 1$
 - Calcular $a = g^k \text{ mod } p$ e $b = M * y^k \text{ mod } p$
 - Criptograma: (a, b)
- Decifragem
 - Dada a chave pública (y, g, p) , e o criptograma (a, b)
 - $M = b/a^x \text{ mod } p$

EL-Gamal – segurança

- Derivar chave privada da chave pública:
 - Corresponde precisamente ao problema do logaritmo discreto, que se crê intratável.
- Extrair mensagem do criptograma:
 - Se se escolher uma mensagem arbitrária (de entre todo o espaço de mensagens admissíveis), “acredita-se” que não é possível derivar essa mensagem do criptograma respetivo.
- (Não) Indistinguibilidade de mensagens:
 - É possível demonstrar que, dado um criptograma c que se sabe resultante da cifra de uma de duas mensagens previamente escolhidas, não é possível saber qual a mensagem efetivamente cifrada (admitindo que o problema *Diffie-Hellman* é intratável).

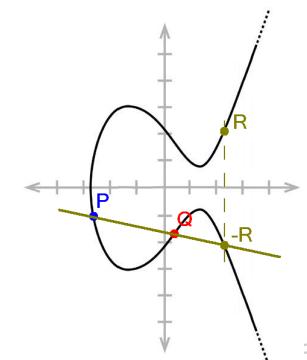
Tópicos

- Parte VI: Acordo de chaves
- **Parte VII: Criptografia de chave pública**
 - Cifra assimétrica
 - Assinatura Digital
 - Algoritmo RSA
 - Algoritmo EL-Gamal
 - **Criptografia de curvas elípticas**
 - Utilização
- Parte VIII: Infraestrutura de chave pública

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Criptografia de chave pública – Curvas elípticas (ECC)

- A criptografia de curvas elípticas (ECC) foi sugerida por Neal Koblitz e Victor S. Miller em 1985 (de forma independente).
- Baseado no *problema discreto do logaritmo da curva elíptica* (em Inglês - *Elliptic Curve Discrete Logarithm Problem* - ECDLP).
 - Assume que não é possível encontrar o logaritmo discreto de um elemento de uma curva elíptica aleatória, em relação a um ponto base conhecido.
- O NIST recomenda a utilização de criptografia de curvas elípticas, em especial:
 - *elliptic-curve Diffie–Hellman* (ECDH), para troca de chaves, e
 - *Curve Digital Signature Algorithm* (ECDSA) para assinatura: NIST P-256 - *secp256r1, prime256v1* -, NIST P-384 - *secp384r1* -, NIST P-521 - *secp521r1* -.
 - Quando utilizada para assinatura, deve ser utilizado *NIST P-256 e SHA256; NIST P-384 e SHA384; e NIST P-521 e SHA512*
- A norma ETSI TS 119 312 (*Cryptographic Suites*) recomenda, para além das curvas NIST, as curvas da família *brainpool*, nomeadamente:
 - *brainpoolP256r1, brainpoolP384r1 e brainpoolP512r1*



Curvas elípticas (ECC)

- O problema do “logaritmo discreto” pode ser expresso em qualquer corpo finito (e.g. $GF(p)$ ou $GF(p^n)$)
- - ...em particular, podemos exprimir a exponenciação no grupo cíclico determinado por uma curva elíptica sobre o corpo considerado.
- Parâmetros da curva elíptica sobre o Corpo finito F_p são definidos da seguinte forma:
 - $T = (p, F_p, a, b, G, n, h)$, em que:
 - p é um inteiro,
 - $a, b \in F_p$ especificam a curva elíptica $E(F_p)$ definida por $y^2 = x^3 + ax + b \pmod{p}$,
 - $G = (x_G, y_G)$ é um ponto base de $E(F_p)$,
 - n é um número primo que define a ordem de G (i.e., número de pontos do subgrupo gerado pelo ponto base),
 - h é um inteiro que define o cofator, $h = \#E(F_p)/n$ (i.e., número de pontos da curva dividido pelo número de pontos do subgrupo gerado pelo ponto base)
- Permite representações compactas para níveis de segurança pretendidos (e.g. 163 bit para níveis de segurança análogos aos 1024 bit em RSA) ...
- ... e implementações eficientes das operações pretendidas.

Tópicos

- Parte VI: Acordo de chaves
- **Parte VII: Criptografia de chave pública**
 - Cifra assimétrica
 - Assinatura Digital
 - Algoritmo RSA
 - Algoritmo EL-Gamal
 - Criptografia de curvas elípticas
 - **Utilização**
- Parte VIII: Infraestrutura de chave pública

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Utilização de criptografia de chave pública

- A utilização de criptografia de chave pública deve ser considerada quando for apropriado ao seu caso de uso.
- Não necessita (nem deve) desenvolver o código para as funções de criptografia assimétrica, já que existem bibliotecas/APIs que já disponibilizam o código necessário (i.e., as operações base das funções de criptografia assimétrica). Por exemplo:
 - Em Python, pode utilizar a cryptography (<https://cryptography.io/>) e o PyCryptodome (<https://www.pycryptodome.org/>);
 - Em Javascript ou Node.js pode utilizar o crypto (<https://nodejs.org/api/crypto.html>).
 - Em Java, tal como referido para as cifras simétricas, pode utilizar
 - os *default providers* da Sun (propriedade da Oracle), nomeadamente SUN, SunJCE, SunPKCS11, ...;
 - O provider do Bouncy Castle (<https://www.bouncycastle.org/java.html>).



Utilização de criptografia de chave pública

- Exemplo em python, utilizando o pycryptodome

```
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP
import binascii
# Gerar par de chaves RSA de 3072 bita
keyPair = RSA.generate(3072)
# Obter as components (n, e) da chave pública
pubKey = keyPair.publickey()
print(f"Chave publica: (n={hex(pubKey.n)}, \ne={hex(pubKey.e)})")
# Obter as components (n, d) da chave privada
print(f"Chave privada: (n={hex(keyPair.n)}, \nd={hex(keyPair.d)})")

# Cifrar com RSA-OAEP
msg = b'A mensagem que vou cifrar'
encryptor = PKCS1_OAEP.new(pubKey)
encrypted = encryptor.encrypt(msg)
print("Mensagem cifrada:", binascii.hexlify(encrypted))

#Decifrar
decryptor = PKCS1_OAEP.new(keyPair)
decrypted = decryptor.decrypt(encrypted)
print("Mensagem decifrada:", decrypted)
```



Utilização de criptografia de chave pública

- Exemplo em python, utilizando o pycryptodome

```
from Crypto.PublicKey import RSA
from Crypto.Signature import pss
from Crypto.Hash import SHA256
import binascii

# Gerar par de chaves RSA de 3072 bits
keyPair = RSA.generate(3072)

# Obter as componentes (n, e) da chave pública
pubKey = keyPair.publickey()
print(f"Chave pública: (n={hex(pubKey.n)}, \ne={hex(pubKey.e)})")

# Obter as componentes (n, d) da chave privada
print(f"Chave privada: (n={hex(keyPair.n)}, \nd={hex(keyPair.d)})")

# Assinar com RSA-PSS
msg = b'A mensagem que vou assinar'
h = SHA256.new(msg)
print("Hash da mensagem:", h.hexdigest())
signature = pss.new(keyPair).sign(h)
print("Assinatura do hash:", binascii.hexlify(signature))

# Validar a assinatura
h1 = SHA256.new(msg)
verifier = pss.new(pubKey)
try:
    verifier.verify(h1, signature)
    print("The signature is authentic.")
except (ValueError, TypeError):
    print("The signature is not authentic.")
```



Utilização de criptografia de chave pública – openssl

- O openssl (<https://www.openssl.org>) é um toolkit ("canivete suíço") para criptografia e comunicações seguras.
 - Curvas elípticas, utilizando a linha de comando (windows, linux, macos, ...)

```
# ver qual a lista de curvas suportada pela sua versão de openssl
```

```
openssl ecparam -list_curves
```

```
# Gerar a chave privada com a curva NIST P-256
```

```
openssl ecparam -name prime256v1 -genkey -noout -out privatekey.pem
```

```
debian@vm3:/tmp$ more privatekey.pem
-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIHYkG0RpvgdZMTUWzKXp0fTcUzMxTYMf+q1xPHR3I836oAoGCCqGSM49
AwEHoUQDQgAEg6qZeiTl1XEHC3CfTKXcMJc2PGqhGo816pjtxWZf4f1qQDu1mfBP
NZh9JJ5giXmlb34j8/h/phrEWUqIBLFT4g==
-----END EC PRIVATE KEY-----
```

```
# Gerar a correspondente chave pública
```

```
openssl ec -in privatekey.pem -pubout -out publickey.pem
```

```
debian@vm3:/tmp$ more publickey.pem
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEg6qZeiTl1XEHC3CfTKXcMJc2PGqh
Go816pjtxWZf4f1qQDu1mfBPNZh9JJ5giXmlb34j8/h/phrEWUqIBLFT4g==
-----END PUBLIC KEY-----
```



Utilização de criptografia de chave pública – openssl

- O openssl (<https://www.openssl.org>) é um toolkit ("canivete suíço") para criptografia e comunicações seguras.
 - Curvas elípticas, utilizando a linha de comando (windows, linux, macos, ...)

```
# Obter o hash do texto que quero assinar
```

```
echo -n "Mensagem que vou assinar" | openssl dgst -sha256 -binary > hash.sha256
```

```
# ver a hash obtida
```

```
hd hash.sha256
```

```
debian@vm3:/tmp$ hd hash.sha256
00000000  ca 03 cf cb f8 92 d3 f2 11 ab cc 38 dc 08 ed 40  |.....8...@I
00000010  43 d7 56 0d f1 f4 36 d6  ec f9 70 b7 4d 4f b6 ac  |C.V...6...p.M0..I
00000020
```

```
# obter a assinatura da hash
```

```
openssl pkeyutl -sign -inkey privatekey.pem -in hash.sha256 > prime256v1.sig
```

```
# validar a assinatura
```

```
openssl pkeyutl -in hash.sha256 -inkey publickey.pem -pubin -verify -sigfile prime256v1.sig
```

```
debian@vm3:/tmp$ openssl pkeyutl -in hash.sha256 -inkey publickey.pem -pubin -verify -sigfile prime256v1.sig
Signature Verified Successfully
```

Tópicos

- Parte VI: Acordo de chaves
- Parte VII: Criptografia de chave pública
- **Parte VIII: Infraestrutura de chave pública**
 - Arquitetura e aspectos operacionais
 - Certificados X.509
 - Cadeias de certificação / confiança
 - Perfis e políticas de certificados
 - Lista de revogação de certificados e OCSP
 - Serviço de *Timestamp*
 - Legislação relevante (eIDAS, DL 12/2021, ...)
 - Utilização

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Infraestrutura de chave pública – ... relembrando

- A combinação de técnicas criptográficas simétricas e assimétricas permite ultrapassar um aspecto crítico limitativo da aplicabilidade em larga escala das técnicas criptográficas simétricas – **a pré-distribuição de chaves.**
- Mas a segurança das técnicas criptográficas assimétricas depende da correta associação entre a chave pública e a identidade do titular da mesma.
- Para garantir essas associações, faz-se uso de uma entidade externa de confiança (**Entidade de Certificação - EC**):
 - Os Certificados de Chave Pública, assinados pela EC, atestam a associação entre a chave pública e a identidade do **titular** da mesma;
 - As partes confiantes (*relying parties*) aceitam como válida essa associação (por via da confiança depositada na EC).
- As partes confiantes (*relying parties*), de cada vez que necessitarem de uma chave pública, solicitam o respetivo certificado:
 - Confirmam a validade do certificado verificando a assinatura nele contido utilizando para isso a chave pública da EC (essa sim, terá de ser distribuída de uma forma segura)

Infraestrutura de chave pública – ... relembrando

- Não são necessários canais seguros para transmitir os certificados
- Interessa reforçar o **papel crítico** que as Entidade de Certificação exercem na segurança de todo o sistema.

?e porquê?

- É por isso normal impôr-se nas ECs padrões de segurança muito elevados (utilização de HSMs; segurança física; planos de segurança rigorosos; etc.).
- Para operacionalizar todos estes aspectos, torna-se necessário fixar toda uma série de formatos, regras e procedimentos relativos aos mecanismos de certificação das chaves públicas que são adoptados por uma **Infraestrutura de Chave Pública (ICP)** – ou *Public Key Infrastructure (PKI)*

Infraestrutura de chave pública – Enquadramento

- A utilização em larga escala de certificados de chave pública pressupõe uma base sólida de fixação e sistematização de:
 - Formatos e tecnologia;
 - Convenções e assumpções;
 - Práticas e procedimentos;
 - Enquadramento normativo e legal;
 - Etc.
- De facto, uma ICP faz a ponte entre conceitos e elementos de natureza tecnológica (chaves criptográficas), com aspetos que no limite detém um cariz “social” (identidade; individualidade; personalidade).
- Por outro lado, e quando considerada em toda a sua abrangência, envolve questões que tocam aspetos tão diversos quanto:
 - Tecnológicos (especificação de formatos, protocolos, etc.);
 - Comerciais (direitos de propriedade, vantagens competitivas, quotas de mercado, etc.);
 - Políticas (supervisão e controlo, jurisdição, soberania, etc.)
 - ...



Infraestrutura de chave pública – Enquadramento

- Trata-se pois de um área que, em grande medida, tem evoluído em resposta a estímulos localizados e bem sucedidos na solução de problemas concretos, que depois de suficientemente estabilizados/ amadurecidos são “adotados” para abordar questões mais abrangentes...
 1. Definições básicas: âmbito e objetivos; formatos; codificações; etc.
 2. Processos: interações ao nível dos intervenientes “locais” (EC e partes confiantes); operações e ciclo de vida; protocolos de gestão; etc.
 3. Inter-operacionalidade: mecanismos de interação e inter-relação entre diferentes utilizações e comunidades; impacto nas relações de confiança e validade; etc.
 4. Regulamentação e acreditação: boas práticas, regras, organismos, etc.
 5. Enquadramento legal: jurisdição, direitos e responsabilidades; valor jurídico de operações eletrónicos; etc.
 6. ICPs públicas: suporte aos organismos atividades dos estados e cidadãos

Infraestrutura de chave pública – Origem histórica

- O primeiro esforço bem sucedido para standardização dos Certificados de Chave Pública aconteceu no âmbito dos protocolos associados ao serviço de diretoria X.500. Em 1988, surge o standard X.509 responsável por estabelecer os mecanismos de autenticação para o X.500.
- Por se tratar da primeira proposta abrangente e sistemática para o mecanismo de certificação, acabou por servir de base para muito do trabalho de normalização e standardização relacionado com ICP.
- É assim que hoje, os certificados X.509 estão na base de praticamente todas as abordagens à ICP (com honrosas exceções, como o esquema de certificação do PGP).
- O Grupo de Trabalho IETF PKIX WG foi constituído em 1995 para promover o desenvolvimento de standards técnicos (RFCs) que suportassem a ICP baseada em certificados X.509.



Infraestrutura de chave pública – Origem histórica

- O PKIX-WG, assim como outros organismos e consórcios (e.g. NIST, W3C, ...), produziu um vasto conjunto de documentos que constituem o suporte tecnológico para a ICP (e.g. <https://datatracker.ietf.org/wg/pkix/documents/>)
- Certos domínios de aplicação exerceram também um papel catalizador na adopção das propostas tecnológicas que iam sendo desenvolvidas (como navegação segura na web, correio eletrónico, etc.), promovendo elas próprias avanços nessas tecnologias.
- Em termos comerciais, a área atraiu muita interesse, assistindo-se a uma explosão da oferta de serviços, e contribuindo para a sua consolidação.
- Por último, assistiu-se também a um impulso significativo promovido por organismos públicos (e.g. as iniciativas da União Europeia na promoção da economia digital, etc.), que alargaram a adopção e aplicabilidade da tecnologia.



Infraestrutura de chave pública – Âmbito

- Interessa reforçar que, mesmo com todo o volume de normas e standards que preveem ICPs de âmbito verdadeiramente global, continua a fazer sentido considerarem-se também ICPs de âmbito local com atribuições específicas.
- Os requisitos e os procedimentos são nesses casos ajustados de acordo com a criticidade do sistema
- Diferentes âmbitos para uma ICP:
 - ICP *in-house*
 - ICP comercial
 - ICP em *outsourcing*
 - ICP comercial com atribuições especiais
 - ICP pública.

Infraestrutura de chave pública – ICP *in-house*

- ICP criada para dar resposta à necessidade de certificação locais à organização.
- Requisitos e procedimentos associados à ICP são normalmente muito simplificados, que resulta num ponto de falha crítico para segurança do sistema.
- Por norma, é benéfico adoptar os mesmos formatos/procedimentos/etc. estabelecidos pelos standards, por forma a permitir a (re)utilização de software standard.
- Vantagens:
 - Flexibilidade;
 - Cadeia de confiança não depende de terceiros;
 - Existe suporte nos sistemas operativos.
- Desvantagens:
 - Requer recursos humanos qualificados;
 - Tendência para “relaxar demasiado” aspectos da segurança;
 - Normalmente não é auditada por terceira entidade.
- Utilizações típicas:
 - Projetos piloto;
 - Segurança de Intranets.

Infraestrutura de chave pública – ICP comercial

- Empresas comerciais que fornecem o serviço de emitirem certificados de chave pública.
- Por regra, essas entidades estão acreditadas (e são auditadas) para o efeito, pelo que é credível que ofereçam níveis de credibilidade/segurança aceitáveis.
- Organismos “adquirem” os certificados que necessitam dessas ICPs.
- Vantagens:
 - Simplicidade e baixo custo;
 - Grande oferta (escolha);
 - Garantia de padrões de segurança/qualidade;
 - Certificados das ICPs são por norma válidos na configurações standard (por omissão) dos sistemas operativos.
- Desvantagens:
 - Dependência de terceiros.
- Utilizações típicas:
 - Sítio de comércio electrónico;
 - Email seguro.

Infraestrutura de chave pública – ICP em *outsourcing*

- Um serviço que algumas empresas (normalmente empresas que têm Ecs comerciais) disponibilizam é o de alojarem/gerirem ICPs de clientes.
- Dessa forma, recursos e *knowhow* são da empresa que disponibiliza o serviço, sendo que o controlo sobre os certificados emitidos se mantém no cliente.
- Vantagens:
 - Controle sobre a ICP;
 - Garantia de padrões de segurança/qualidade;
 - Certificados das ICPs são por norma válidos na configurações standard dos sistemas operativos e *browsers*.
- Desvantagens:
 - Custo;
 - Dependência de terceiros.
- Utilizações típicas:
 - Certificados para colaboradores/serviços de uma organização (email, TLS, etc.).

Infraestrutura de chave pública – ICP comercial com atribuições especiais

- Certas empresas de certificação estão habilitadas a emitir certificados com atribuições especiais.
- Normalmente, pressupõe um processo de acreditação específico.
- O suporte desses certificados é muitas vezes um *token* criptográfico (e.g. *smartcard*).
- Vantagens:
 - Ter acesso à atribuição especial concreta;
 - ICPs estão por norma obrigadas a requisitos específicos e/ou mais apertados.
- Desvantagens:
 - Escolha limitada (ou inexistente).
- Utilizações típicas:
 - Emissão de certificados qualificados (aptos para assinatura qualificada);
 - Emissão de alguns tipos de certificados TLS;
 - Certificados para *code-signing* em sistemas fechados.



Infraestrutura de chave pública – Normas e standards

- O esforço de normalização, standardização e regulação das ICPs é levado a cabo por organismos internacionais com competências nas diferentes áreas envolvidas. Ao nível técnico, destacam-se:
 - *International Telecommunications Union (ITU)*, responsável pela recomendação X.509 que introduz a utilização de certificados em sistemas de telecomunicações;
 - *Internet Engineering Task Force (IETF)*, uma comunidade internacional de produtores, operadores, vendedores e investigadores das tecnologias de redes, que são responsáveis por expressar a aplicação das tecnologias criptográficas num conjunto de *Requests For Comments (RFCs)*;
 - *PKIX Working Group*, um grupo da IETF que gere os RFCs relacionados com os certificados X.509. Este grupo de trabalho mantém um conjunto de documentos que se denomina “*Internet X.509 Public Key Infrastructure*”;
 - *ETSI – European Telecommunications Standard Institute*, responsável pela standardização Europeia nas áreas das tecnologias de informação e comunicação.
 - *ENISA – European Union Agency for Cybersecurity*, responsável pela elaboração de boas práticas (e “interpretação” de normas e standards) de operação das ICPs.
 - *CAB (CA/Browser) Forum*, uma comunidade internacional de Entidades de Certificação, produtores de browsers web e outro software que utiliza certificados X.509, responsável pelas recomendações para emissão de certificados SSL/TLS para servidores Web.

Infraestrutura de chave pública – Componentes

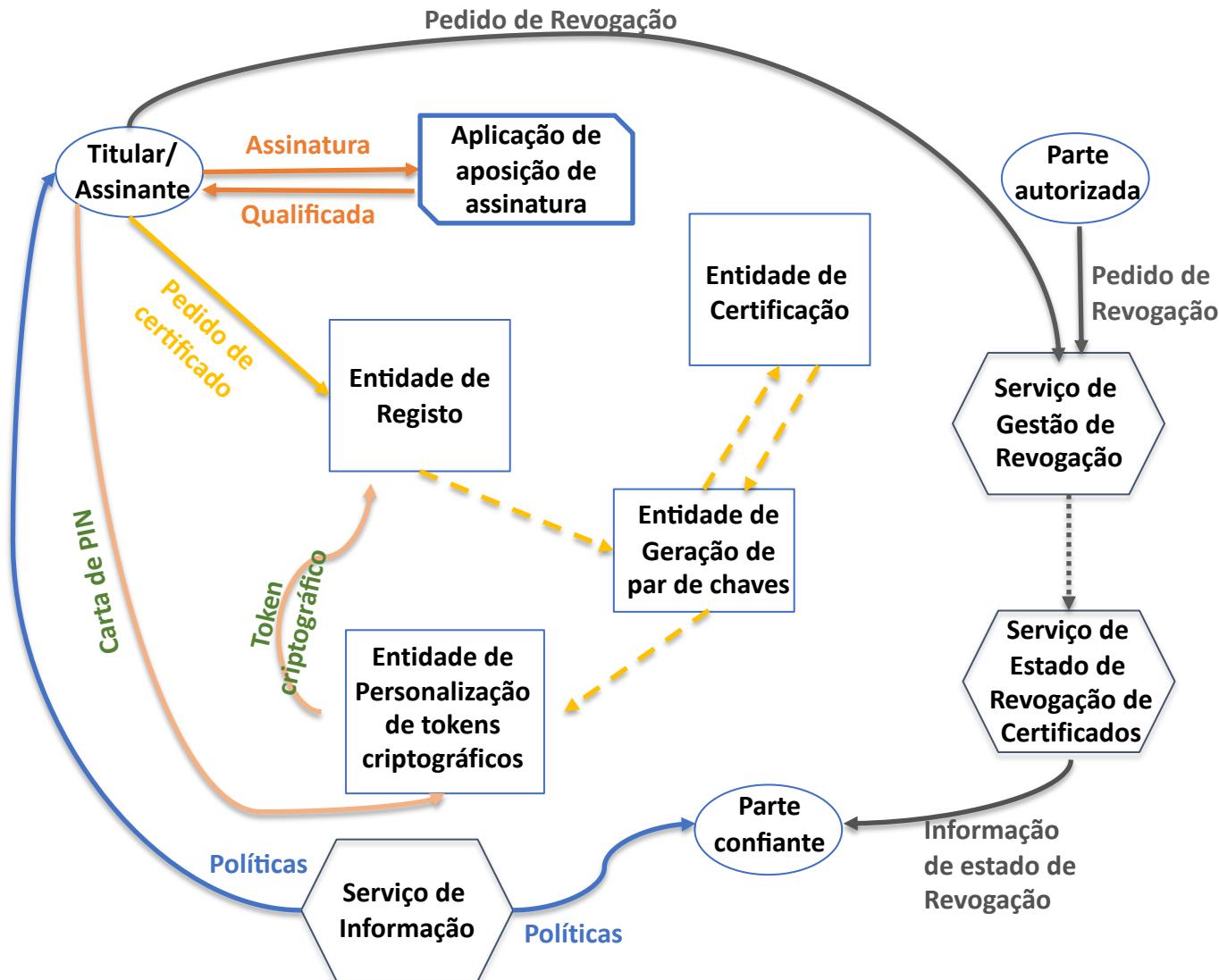
- Uma **Infraestrutura de Chave Pública (ICP)** define-se como o conjunto de *hardware*, *software*, pessoas, políticas, processos e procedimentos necessários para criar, gerir, armazenar, distribuir e revogar certificados de chave pública.
- Intervêm numa ICP diferentes entidades, sendo as mais comuns:
 - **Titulares de Certificados**: possuem as respetivas chaves privadas que utilizam para decifrar mensagens ou produzir assinaturas digitais.
 - **Parte confiante (*Relying parties*)**: Utilizam a chave pública contida num certificado para cifrar mensagens e verificar assinaturas.
 - **Entidades de Certificação (EC)**: Emitem/renovam/revogam certificados.
 - **Entidades de Registo (ER)**: Garantem a associação entre chaves públicas e identidades de titulares.
 - **Repositórios**: Armazenam e disponibilizam certificados e outra informação relevante. E.g., Lista de Certificados Revogados – LRC ou CRL (*Certification Revocation List*) –, etc.

Infraestrutura de chave pública – Componentes

- O funcionamento de uma PKI baseia-se em dois tipos de protocolos:
 - **Protocolos Operacionais:** Estes protocolos são necessários para entregar certificados e CRLs aos sistemas que os utilizam. Estas operações podem ser efetuadas de diversas formas. Para todos estes meios estão especificados protocolos operacionais que definem, inclusivamente, os formatos das mensagens.
 - **Protocolos/Cerimónias de Gestão:** Estes protocolos são necessários para dar suporte às interações entre os utilizadores e as entidades de gestão da PKI, nomeadamente:
 - Inicialização.
 - Registo e Identificação.
 - Emissão e renovação de par de chaves.
 - Pedido de revogação ou suspensão.
 - Emissão de certificados de ECs.



Infraestrutura de chave pública – Arquitetura



Infraestrutura de chave pública – Operações

- **Inicialização:** Processo de instalação e arranque de operação de uma ICP, estabelecendo as diversas entidades, assim como as políticas, processos, procedimentos e recursos que lhes permita funcionar eficazmente. A ICP pode iniciar o serviço após auditoria de conformidade e aprovação por entidade supervisora (caso exista).
- **Registo:** Um utilizador dá-se a conhecer a uma ER (pelo método “cara-a-cara” ou outro) para que a sua identificação possa ser verificada.
- **Geração de Par de Chaves:** Em algumas implementações, as ECs encarregam-se de gerar os pares de chaves dos utilizadores, que enviam de forma segura junto com o certificado (em *token* criptográfico).

Infraestrutura de chave pública – Operações

- **Certificação:** A EC recebe a chave pública do utilizador e a sua identificação e emite o respetivo certificado, de acordo com legislação, normas, processos e procedimentos internos.
- **Publicação de Certificados e CRLs:** Esta tarefa pode ser feita diretamente pela EC, ou indiretamente por entidades como ERs. Além de colocar os certificados e CRLs em repositórios é muitas vezes necessário fazer estes documentos chegar aos utilizadores finais por outros meios.
- **Revogação:** Quando um certificado é emitido o seu período útil de vida está pré-definido. No entanto, pode haver a necessidade de invalidar o certificado antes do fim desse período por diversos motivos (e.g. atributos deixam de ser aplicáveis, a chave privada é comprometida, etc.).

Infraestrutura de chave pública – Operações

- **Recuperação de um Par de Chaves:** Em algumas implementações as ECs armazenam de forma segura (usualmente em HSM – *Hardware Security Module* - ou *token* criptográfico) o par de chaves do titular como back-up e proteção (e.g. no caso do par de chaves de uma EC, ou no caso de uma empresa, se o par de chaves for utilizado para cifra). Nestes casos o par de chaves pode ser restaurado em caso de extravio ou danificação do seu suporte.
- **Renovação de certificados e/ou atualização de Par de Chaves:** Uma vez esgotada a validade de um certificado, existe necessidade de construir um novo certificado. Neste processo pode ou não ser mantido a chave pública do utilizador (normalmente, e de acordo com boas práticas, a chave pública/par de chaves não é mantida).

Tópicos

- Parte VI: Acordo de chaves
- Parte VII: Criptografia de chave pública
- **Parte VIII: Infraestrutura de chave pública**
 - Arquitetura e aspectos operacionais
 - **Certificados X.509**
 - Cadeias de certificação / confiança
 - Perfis e políticas de certificados
 - Lista de revogação de certificados e OCSP
 - Serviço de *Timestamp*
 - Legislação relevante (eIDAS, DL 12/2021, ...)
 - Utilização

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Certificados X.509 – Estrutura

- Conteúdo básico de um certificado de chave pública:
 - Dados Informativos:
 - **Identificação do titular do certificado** (i.e. quem detém a chave privada associada à chave pública contida no certificado);
 - **Chave pública do titular;**
 - **Identificação da EC;**
 - Outra informação relevante para a operacionalização do certificado (**número de série; datas de validade; URL de CRL; URL de OCSP**; etc.)
 - **Assinatura dos dados informativos realizada pela EC** (estabelece a ligação entre chave pública e identificação do titular).
- Os certificados X.509 são uma instância de certificados de chave pública que foram introduzidos para autenticar os nós do serviço de diretoria X.500.
- Os dados são representados como estruturas de dados “atributo/ valor” (dicionários).
- As codificações desses dados está standardizada, garantindo a sua interoperabilidade.
 - **DER** - formato binário definido pelo standard (notação ASN.1)
 - **PEM** - representação da informação contido no formato DER em caracteres imprimíveis
- A assinatura do certificado é efetuada pela Entidade de Certificação (EC) sobre a codificação DER dos dados nele contidos.

Certificados X.509v3

- A versão de certificados utilizada atualmente (v3, standardizada em 1996) veio colmatar as deficiências que as versões anteriores apresentavam em alguns domínios de aplicações, e que se traduziam essencialmente na necessidade de mais atributos.
- Esta versão introduziu um novo campo do tipo ***Extensions***, equipando assim os certificados com a flexibilidade necessária às novas utilizações.
- As extensões permitem associar atributos genéricos a uma entidade ou à sua chave pública.
- Cada extensão é, ela própria, uma estrutura de dados com um identificador e um valor adequado ao tipo do atributo que representa.

Certificados X.509v3 – Atributos básicos

- **version** – Versão do standard X509 (v3).
- **serialNumber** – Número único atribuído pela EC ao certificado.
- **subject** – Identificação do titular da chave pública contida no certificado.
- **subjectPublicKeyInfo** – Estrutura contendo a chave pública do titular do certificado e identificação do algoritmo correspondente.
- **issuer** – Identificação da EC que emite o certificado.
- **signature** – Estrutura que identifica o algoritmo utilizado para gerar a assinatura da EC que acompanha o certificado.
- **validity** – Estrutura com as duas datas que delimitam o período de validade do certificado.

Certificados X.509v3 – Identificadores

- Os atributos **issuer** e **subject** que identificam a EC e o titular do certificado respectivamente são do tipo **Name**.
 - O tipo **Name** provém da norma X.501 e é utilizado porque permite a compatibilidade com os sistemas de diretório definidos nas normas X.500 (e.g. DAP e LDAP).
 - O tipo **Name** é uma coleção de atributos, geralmente **strings** da forma “**<nome> = <valor>**”. Estes atributos definem um **Distinguished Name (DN)** para o titular do certificado e para a EC emissora do certificado.
 - O **DN** tem uma estrutura hierárquica. A norma X.520 standardiza alguns dos componentes de um DN. Os seguintes são de reconhecimento obrigatório e muito utilizados:
 - *Country (C)*
 - *Organization (O)*
 - *Organizational Unit (OU)*
 - *Common Name (CN)*
 - *Serial Number (SN)*
 - Exemplo de DN de titular do certificado: C=PT, O=“Universidade do Minho”, OU=“Departamento de Informática”, CN=“João Ratão”.

Certificados X.509v3 – Extensões

- Permitem personalizar os dados contidos no certificado (que são comprovados pela EC por via da respetiva assinatura)
- As extensões são marcadas como **Critical** ou **Non Critical**. Uma aplicação que encontre uma extensão crítica que não reconheça deve rejeitar o certificado.
- O IETF RFC 5280 (*Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*) normaliza as extensões recomendadas para utilização na Internet, definindo como estas devem ser codificados no certificado.
- São desaconselhados desvios desta recomendação, nomeadamente no que diz respeito a extensões críticas, apesar de não haver qualquer limitação a nível do standard.

Certificados X.509v3 – Extensões

- **Basic Constraints** permite assinalar um certificado como pertencendo a uma (sub-)EC, e limitar o comprimento de cadeias de certificados.
- **Certificate Policies** permite incluir informação relativa às políticas e práticas de certificação aplicáveis ao certificado:
 - Para certificados de utilizador, permite especificar em que condições o certificado foi emitido e quais as restrições associadas à sua utilização.
 - Para certificados de ECs, permite definir as políticas de certificação aplicáveis por ECs hierarquicamente inferiores.
 - Usualmente, este é um documento público da EC, e tem o conteúdo identificado no IETF RFC 3647 (*Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*)

Certificados X.509v3 – Extensões

- **Key Usage** - permite restringir as utilizações do par de chaves associado ao certificado e.g. quando uma chave apenas pode ser utilizada para verificar assinaturas digitais. Contempla as seguintes utilizações:
 - ***digitalSignature*** - assinaturas digitais para autenticação e integridade de dados, excepto certificados e CRLs.
 - ***nonRepudiation*** - assinaturas digitais para não repúdio.
 - ***keyEncipherment*** - proteção da confidencialidade de chaves.
 - ***dataEncipherment*** - proteção da confidencialidade de dados.
 - ***keyAgreement*** - protocolos de acordo de chaves.
 - ***keyCertSign*** - assinatura de certificados.
 - ***cRLSign*** - assinatura de CRLs.
 - ***encipherOnly/decipherOnly*** - restringem a funcionalidade ***keyAgreement***.

Certificados X.509v3 – Extensões

- **Extended Key Usage** - Permite especificar ou restringir as utilizações previstas para o par de chaves associado ao certificado, em adição ou em alternativa à extensão **Key Usage**. Estão definidas diversas utilizações, bem como a sua relação com as especificadas na extensão **Key Usage**:
 - *WWW server authentication*
 - *WWW client authentication*
 - *Signing of downloadable executable code*
 - *E-mail protection*
- **CRL Distribution Points** serve para indicar à parte confiante de um certificado onde pode obter informação quanto à revogação do certificado na forma de *Certificate Revocation Lists* (CRLs).
- **Authority Information Access**: serve para indicar à parte confiante onde pode obter informação quanto à revogação do certificado na forma de serviço OCSP (*Online Certificate Status Protocol*), cf IETF RFC 6960, assim como indicar onde está publicado o certificado da EC que emitiu este certificado.

Tópicos

- Parte VI: Acordo de chaves
- Parte VII: Criptografia de chave pública
- **Parte VIII: Infraestrutura de chave pública**
 - Arquitetura e aspectos operacionais
 - Certificados X.509
 - **Cadeias de certificação / confiança**
 - Perfis e políticas de certificados
 - Lista de revogação de certificados e OCSP
 - Serviço de *Timestamp*
 - Legislação relevante (eIDAS, DL 12/2021, ...)
 - Utilização

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Cadeias de certificação / confiança

- Para utilizar um serviço que requeira o conhecimento de uma chave pública, é necessário obter e validar um certificado que a contenha.
- A validação do certificado implica, por sua vez, o conhecimento da chave pública da Entidade de Certificação que o emitiu.
- Aqui, existem duas alternativas:
 1. A chave pública já é do conhecimento do utilizador (e.g. foi pré-instalada de forma segura), ou
 2. é também fornecida por via de um certificado emitido por uma outra EC
- Naturalmente que, no segundo caso, há necessidade de proceder à verificação de validade desse certificado.

Que resulta num procedimento de validação recursivo!

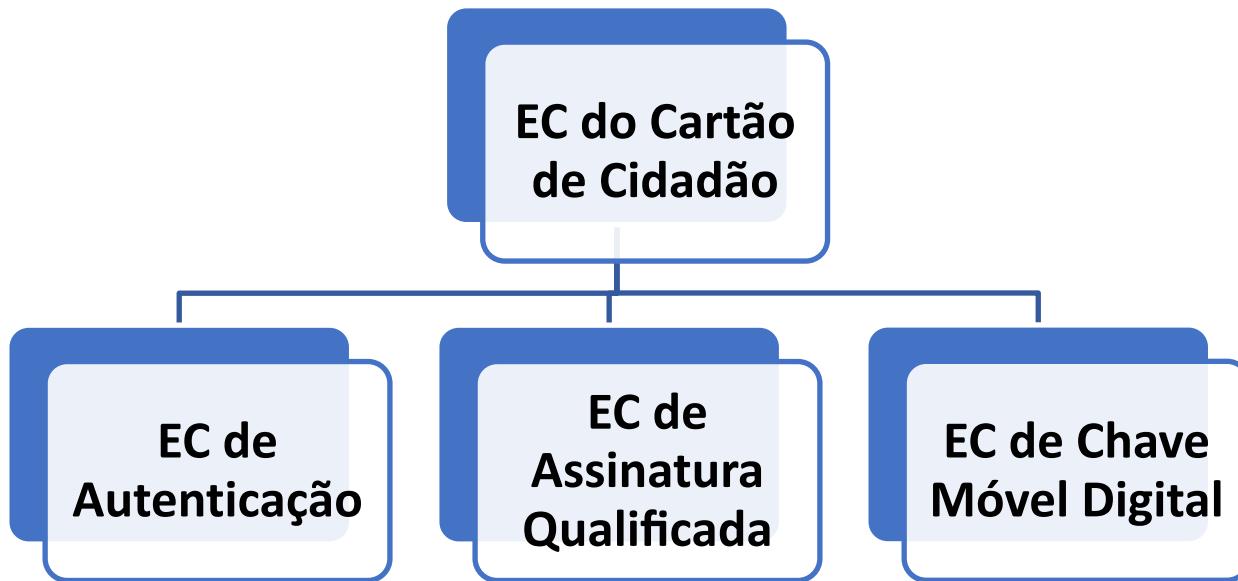
Que só termina quando se encontrar um certificado de uma EC que já se confia!

Cadeias de certificação / confiança

- A esta sequência de certificados envolvidos no processo de validação dá-se o nome de **cadeia de certificação** (também chamada de **cadeia de confiança**).
- Note que, numa cadeia de certificação bem formada, o **issuer** de um certificado deverá ser o **subject** do antecessor.
- As cadeias de certificação refletem uma **hierarquia de Entidades de Certificação**: as ECs hierarquicamente superiores emitem os certificados das ECs hierarquicamente inferiores.
- No(s) topo(s) da hierarquia reside uma EC denominada **Root** ou raiz. O certificado desta EC é emitido e assinado por ela própria – ou seja, um certificado auto-assinado, i.e. os campos **subject** e **issuer** do seu certificado são iguais.
- A confiança na chave pública de uma Root EC é estabelecida por um meio externo à ICP.
- Por exemplo: sistemas operativos comuns (e.g. MS Windows, macOS, iOS, Android) e browsers (e.g. Firefox) incluem certificados de dezenas de Root ECs!.

Cadeias de certificação / confiança

Exemplo: PKI do Cartão de Cidadão



Cadeias de certificação / confiança

Exemplo: PKI do Cartão de Cidadão

EC do Cartão de Cidadão

- Encontra-se englobada na hierarquia do Sistema de Certificação Eletrónica do Estado Português (SCEE) – Infraestrutura de Chaves Públicas do Estado;
- Regulada pela “Política de Certificados da SCEE e Requisitos mínimos de Segurança”;
- É uma ECEstado, encontra-se no nível imediatamente abaixo da ECRaizEstado;
- É a EC raiz da hierarquia da PKI do Cartão de Cidadão;
- Emite certificados para:
 - as suas sub-ECs (assinatura, autenticação e chave móvel digital) e
 - serviço de estado de revogação de certificados (CRL e OCSP).
- Tem uma validade de 14 anos.

Cadeias de certificação / confiança

Exemplo: PKI do Cartão de Cidadão

- Está englobada na hierarquia do SCEE;
- É uma subECEstado;
- Emite certificado de autenticação para o cidadão e, selo eletrónico avançado para os serviços de estado de revogação de certificados.
- Regulada:
 - pela “Política de Certificados da SCEE e Requisitos mínimos de Segurança”;
 - pelo Regulamento eIDAS – Regulamento (UE) n.º 910/2014 – (art. 6º a 12º)
 - Reconhecimento mútuo (UE) de sistemas de identificação eletrónica
 - Nível de garantia (***Level of Assurance***) elevado, após avaliação pela ***Cooperation Network***, baseado em processos de:
 - Registo/*Enrolment*;
 - Gestão, Autenticação e Interoperabilidade do sistema de identificação eletrónica;
 - Gestão e organização.

EC de
Autenticação

Cadeias de certificação / confiança

Exemplo: PKI do Cartão de Cidadão



Cooperation Network (Rede de cooperação)

- Criada pela Decisão de Execução (UE) 2015/296;
- Avalia, através de processo de avaliação pelos pares, a interoperabilidade e a segurança dos sistemas de identificação eletrónica notificados (de acordo com o procedimento e formulário indicados na Decisão de Execução (UE) 2015/1984) pelos Estados-Membros ao abrigo da artigo 7º, alínea g), do Regulamento (UE) n.º 910/2014;
- Emite parecer com indicações sobre a conformidade dos sistemas descritos com as exigências previstas no artigo 7º, no artigo 8º, nº 1 e 2, e no artigo 12º, nº 1, do Regulamento (UE) n.º 910/2014 e no ato de execução previsto no artigo 8º, nº 3, do referido regulamento (Decisão de Execução (UE) 2015/1502).

Cadeias de certificação / confiança

Exemplo: PKI do Cartão de Cidadão

Level of Assurance High

- Certificado de Autenticação do CC pré-notificado a 30/05/2018
- Opinião da *Cooperation Network* a 10/10/2018
- Publicado no Jornal Oficial da UE a 28/02/2019

**EC de
Autenticação**

Title of the scheme	eID means under the notified scheme	Notifying Member State	Level of assurance	Authority responsible for the scheme	Date of publication in the Official Journal of the European Union
Cartão de Cidadão (CC)	Portuguese national identity card (eID card)	Portuguese Republic	High	AMA - Administrative Modernisation Agency Rua Abrantes Ferrão n.º 10, 3º 1600 - 001 Lisbon ama@ama.pt +351 217231200	28.2.2019

Cadeias de certificação / confiança

Exemplo: PKI do Cartão de Cidadão

- Englobada na hierarquia do SCEE;
- É uma subECEstado;
- Emite certificados de assinatura qualificada para o cidadão, selo eletrónico avançado para a Entidade de Validação Cronológica e para os serviços de estado de revogação de certificados.
- Regulada:
 - Pela “Política de Certificados da SCEE e Requisitos mínimos de Segurança”;
 - Pelo Regulamento eIDAS – Regulamento (UE) n.º 910/2014
 - Corresponde ao serviço qualificado de confiança de Emissão de Certificados qualificados de assinaturas eletrónicas (artigo 28º do regulamento eIDAS) por Prestador qualificado de serviço de confiança (IRN).
 - Auditada por um organismo de avaliação da conformidade.
 - Supervisionada pela Entidade supervisora nacional (Gabinete Nacional de Segurança).

EC de
Assinatura
Qualificada

Cadeias de certificação / confiança

Exemplo: PKI do Cartão de Cidadão

- Englobada na hierarquia do SCEE;
- É uma subECEstado;
- Emite certificados de assinatura qualificada para o cidadão, no âmbito da CMD (Chave Móvel Digital).
- Regulada:
 - Pela “Política de Certificados da SCEE e Requisitos mínimos de Segurança”;
 - Pelo Regulamento eIDAS – Regulamento (UE) n.º 910/2014
 - Corresponde ao serviço qualificado de confiança de Emissão de Certificados qualificados de assinaturas eletrónicas (artigo 28º do regulamento eIDAS) por Prestador qualificado de serviço de confiança (AMA).
 - Auditada por um organismo de avaliação da conformidade.
 - Supervisionada pela Entidade supervisora nacional (Gabinete Nacional de Segurança).

**EC de Chave
Móvel Digital**

Validação de certificados

- Para cada certificado da **cadeia de certificação** bem formada, verificar (cf. secção 6 do IETF RFC 5280):
 1. Validade da assinatura
 2. Aplicabilidade do certificado (face às extensões e política)
 3. Estado de revogação (e.g. consultando CRLs ou OCSP)
- A raiz da cadeia de certificação deverá ser de uma EC que já se conheça a chave pública – designa-se por **raiz** ou **âncora** da relação de confiança.
- A convenção é que os certificados de “raiz” são auto-assinados (subject é igual ao issuer).

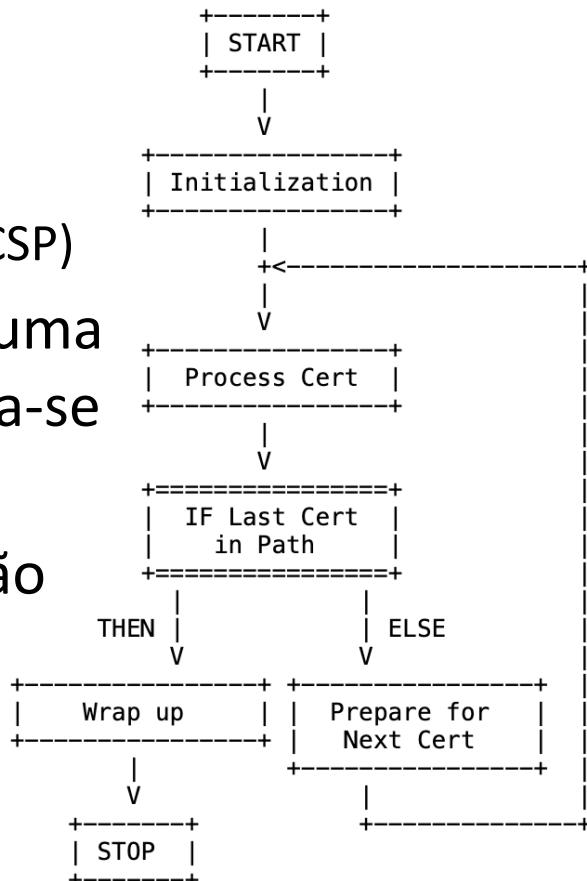


Imagen de <https://datatracker.ietf.org/doc/html/rfc5280>

Validação de certificados – Âncoras de confiança

- Uma parte confiante (ou *relying party*) conhece um número limitado de chaves públicas pertencentes a ECs (em geral Root ECs) e que funcionam como raízes das relações de confiança.
- Isso significa que a parte confiante aceitará um certificado emitido por uma dessas ECs e que depositará um determinado nível de confiança no seu conteúdo.
- A validação de uma cadeia de certificados termina então quando for encontrado um certificado com essa característica. Esses são normalmente certificados auto-assinados.

Conclusão: o grau de confiança depositada num certificado válido baseia-se, em última análise, na confiança depositada na EC que funcionou como raiz da relação de confiança.

Validação de certificados – Âncoras de confiança

- A gestão da lista com âncoras de confiança, assim como do próprio processo de validação das cadeias de certificados, é normalmente assegurada pelo próprio Sistema Operativo ou pela aplicação que os utiliza (e.g., Firefox, Adobe Reader).
- Em particular, a compilação dos certificados Root adoptados, assim como a sua atualização/manutenção, é assegurada pelo fabricante.
- Torna o processo de utilização de certificados praticamente transparente para o utilizador.
 - **O que é bom!** porque, quer os conceitos envolvidos na certificação, quer a própria manipulação dos certificados é complexa.
 - **O que é mau!** porque toda a segurança que supostamente eles suportam fica comprometida se não houver plena consciência das relações de confiança envolvidas.

Tópicos

- Parte VI: Acordo de chaves
- Parte VII: Criptografia de chave pública
- **Parte VIII: Infraestrutura de chave pública**
 - Arquitetura e aspectos operacionais
 - Certificados X.509
 - Cadeias de certificação / confiança
 - **Perfis e políticas de certificados**
 - Lista de revogação de certificados e OCSP
 - Serviço de *Timestamp*
 - Legislação relevante (eIDAS, DL 12/2021, ...)
 - Utilização

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Perfis de certificados (*Certificate Profiles*)

- O standard X509v3 oferece flexibilidade para se definirem extensões à medida das necessidades
- A semântica desses atributos é assim “aberta”, mas que deve ser fixada por regras que estabeleçam, num dado contexto (cenário de utilização, aplicação, protocolo, etc.), quais os atributos que devem estar presentes e qual o seu significado.
- Um **Perfil de Certificados** denota uma classe de certificados, e comprehende:
 - Quais os atributos/extensões de podem ou devem estar presentes e qual a criticidade desses atributos;
 - Qual o significado desses atributos e gama de valores admissível;
 - Quais os algoritmos criptográficos suportados e tamanho de chaves correspondentes;
 - Formato de nomes adoptado e restrições que se lhe devem impôr;
 - Política de Certificados associada e respetiva identificação;
 - Regras de validação para as extensões críticas consideradas.

Políticas de certificados (*Certificate Policies*)

- A confiança depositada numa EC depende desde:
 - Fatores externos, como a credibilidade da instituição ou empresa que suporta a EC e o seu país de origem; etc.
 - Informação sobre as práticas adotadas pela EC, e garantias que elas cumprem os requisitos apropriados (e.g. por via de acreditação)
- Mas a confiança que é depositada num certificado individual depende, em última instância, do critério adotado pela EC na emissão do respetivo certificado.
- Obs: note que uma EC pode emitir certificados para diferentes fins (perfis), sendo que é concebível que esses diferentes perfis ofereçam garantias distintas ...
- Numa ICP, prevê-se a forma de basear a confiança que se deposita num certificado incluindo nele explicitamente a referência para a respectiva **Política de Certificados** e respetiva documentação.

Políticas de certificados (*Certificate Policies*)

- Uma **Política de Certificados (CP)** é um conjunto de regras que define a aplicabilidade de certificados a uma determinada comunidade ou classe de aplicações com requisitos de segurança comuns:
 - A legislação em que se baseará a emissão e utilização dos certificados.
 - Os requisitos e as responsabilidades (nomeadamente legais e financeiras) associados a ECs e ERs.
 - Os requisitos e as responsabilidades associados a Titulares e Partes confiantes.
 - Restrições ao conteúdo e utilização dos certificados
 - Procedimentos a serem implementados relativamente a diversos aspetos do funcionamento de ECs e ERs.
- As Políticas de Certificados permitem:
 - às partes confiantes ajuizarem se devem, num contexto específico, confiar no certificado em questão;
 - que a EC limite a sua responsabilidade explicitando o âmbito de utilização, enquadramento legal, etc. dos certificados por si emitidos.
- As Políticas de Certificados:
 - são documentos disponibilizados pelas ECs para serem consultados pelos titulares e partes confiantes;
 - devem ser explicitamente referenciadas no certificado por recurso às extensões apropriadas;
 - têm conteúdo conforme identificado no IETF RFC 3647 (*Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*).

Políticas de certificados na validação de certificados

- O IETF RFC 5280 dedica algum detalhe às políticas de certificados e ao efeito de uma política de certificados imposta num determinado ponto da hierarquia de certificação.
- Como foi já referido, esta especificação define também as extensões que permitem incluir este tipo de informação nos certificados X.509.
- De facto, associada a cada certificado pode estar uma lista de políticas aplicáveis à sua utilização ou, no caso do certificado de uma EC, uma lista das políticas aceitáveis para os certificados hierarquicamente inferiores.
- Durante a validação de um certificado é necessário propagar as políticas impostas desde o topo da hierarquia até à sua base.
- A política em vigor na base da hierarquia de certificação resulta da reunião das políticas em vigor nos níveis superiores, com a ressalva de que uma política inserida num determinado nível não pode contradizer uma política de nível superior.

Declaração de Práticas de Certificação (*Certification Practice Statement*)

- Está ainda previsto que as ECs publiquem um documento onde explicitam as práticas seguidas na emissão e gestão dos certificados por si emitidos.
- Cada EC publica então uma ou mais **Declaração de Práticas de Certificação (CPS)**, nas quais publicita as suas normas de operação internas.
- Em particular, explica a forma como a EC implementa um determinado conjunto de **Políticas de Certificados**.
- A acreditação de uma EC de acordo com uma determinada CPS implica uma auditoria efetuada por (ou em nome de) uma *Policy Management Authority*.
 - Por exemplo, a PKI Governamental do Canadá define oito CPs correspondentes a quatro níveis de segurança na utilização de certificados em assinaturas digitais e proteção de dados. Uma CA que pretenda emitir certificados em conformidade com estas políticas tem de ser credenciada pelo estado Canadiano.
- É também possível (e recomendado) incluir a referência explícita à CPS nos certificados.

Exemplo de documentos públicos das ECs

- Exemplos de Políticas de certificados:
 - http://pki.cartadecidadao.pt/publico/politicas/PJ.CMDA_34_signed.pdf
 - http://pki.cartadecidadao.pt/publico/politicas/PJ.CC_24.1.2_0003_pt_Root.pdf
- Exemplos de Declaração de Práticas de Certificação:
 - http://pki.cartadecidadao.pt/publico/politicas/PJ.CC_24.1.1_0002_pt_AsC.pdf
 - http://pki.cartadecidadao.pt/publico/politicas/PJ.CMDA_33_signed.pdf
- Exemplos de Declaração de Divulgação de Princípios
 - http://pki.cartadecidadao.pt/publico/politicas/PJ.CC_24.1_0001_pt_Root-AsC-AuC.pdf
 - http://pki.cartadecidadao.pt/publico/politicas/PJ.CMDA_37_signed.pdf

Exemplo de perfis de certificados

- Proteção de Email (S/MIME)
- Autenticação de Sítios (TLS-server)
- Autenticação em Serviços (TLS-client)
- Assinatura Qualificada de Documentos (eIDAS)
- Classificação do Método de Validação de certificados TLS (cf. *CA/browser forum*):
 - **Domain Validation (DV)** - identidade verificada unicamente com base em evidência de controlo do domínio DNS.
 - **Organization Validation (OV)** - verifica existência/controlo de uma organização (e.g. empresa, organismo público, etc.)
 - **Extended Validation (EV)** - critérios mais rigorosos de validação fornecendo evidência de controlo legal sobre a entidade.

Tópicos

- Parte VI: Acordo de chaves
- Parte VII: Criptografia de chave pública
- **Parte VIII: Infraestrutura de chave pública**
 - Arquitetura e aspectos operacionais
 - Certificados X.509
 - Cadeias de certificação / confiança
 - Perfis e políticas de certificados
 - **Lista de revogação de certificados e OCSP**
 - Serviço de *Timestamp*
 - Legislação relevante (eIDAS, DL 12/2021, ...)
 - Utilização

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Revogação de Certificados

- Por vezes há necessidade de revogar certificados que ainda se encontram no seu período de validade
- Motivos para a revogação de certificados:
 - Chave privada comprometida;
 - Circunstância que justificava associação do titular à chave pública já não se verifica (e.g. titular é o detentor de um cargo temporário);
 - Dados contidos no certificado deixam de ser corretos (e.g. atributo já não se aplica);
 - Etc.
- Mecanismo originalmente previsto para a revogação de certificados são as **Listas de Revogação de Certificados (CRL)**

Certificate Revocation Lists (CRL)

- As ***Certificate Revocation Lists*** (CRL) são o canal previsto no X.509 para a revogação de certificados dentro do período de validade. Uma CRL diz-se:
 - **Base CRL** quando lista todos os certificados revogados por uma EC que ainda estão no seu período de validade.
 - **Delta CRL** quando apenas lista os certificados revogados desde a publicação de uma Base CRL referenciada.
- As CRLs são emitidas e assinadas, em geral, pelas próprias ECs. É possível que a EC delegue esta função numa outra autoridade denominada **CRL Issuer**.
- Cada CRL tem um contexto específico (o conjunto de certificados passíveis de aparecerem no seu conteúdo), que deve estar bem definido.

Certificate Revocation Lists (CRL)

- A segurança de uma ICP depende da eficácia com que são revogados os certificados que se tornaram inválidos. Este facto sugere que, assim que um certificado se torna inválido, uma nova CRL deva ser publicada.
- No entanto, desta forma, uma parte confiante nunca saberia qual a CRL mais recente.
- Admitindo que o atacante controla o meio de comunicação que liga a parte confiante ao ponto de publicação de uma CRL, possibilitaria ataques do tipo:
 - Vamos admitir que a parte confiante pretende utilizar um certificado cuja chave privada foi comprometida, e que é conhecida pelo intruso.
 - A parte confiante tenta obter a CRL mais recente, que revogaria o certificado.
 - Mas o intruso fornece uma versão antiga da CRL onde ainda não aparece a revogação desse certificado.
 - A parte confiante aceita o certificado porque não tem como saber que a CRL que utilizou estava desatualizada.

Certificate Revocation Lists (CRL)

- De facto, a utilidade de uma CRL depende do facto de ela ser publicada periodicamente (e.g. diariamente, semanalmente, mensalmente, etc.).
- Isto permite também que a CRL seja pública, e distribuída por canais não seguros.
- Compete à parte confiante estar ao corrente da frequência de publicação das CRLs, e definir uma política sobre o que é uma CRL “suficientemente recente”.
- O atributo *nextUpdate* permite indicar na própria CRL a altura a partir da qual é garantida a publicação de uma nova CRL.

Certificate Revocation Lists (CRL)

- A parte confiante (*relying party*) está consciente de que, a menos que obtenha a última versão da CRL, estará a correr o risco de aceitar certificados inválidos.
- Isto não quer dizer que não possam ser publicadas CRLs extraordinárias, fora da frequência normal de publicação
 - Nota: Normalmente não é efetuado, porque introduz questões legais.
- Isto pode ocorrer, por exemplo, se um certificado importante tem de ser revogado porque a chave privada correspondente foi comprometida (e.g. o certificado de uma EC hierarquicamente inferior).
 - Nota: Muito complexo em termos jurídicos.
- No entanto, a granularidade garantida nunca é inferior ao período de publicação da CRL: não é possível garantir que as partes confiantes obtenham a CRL extraordinária antes da data de publicação da próxima CRL periódica.

Online Certificate Status Protocol (OCSP)

- Os riscos associados à utilização indevida de um certificado revogado podem não ser aceitáveis.
- Em alternativa ou em adição à consulta de uma CRL, pode ser necessária **informação atual (e assinada)** sobre o estado de revogação de um certificado.
- O OCSP (definido no IETF RFC 6960) permite a uma aplicação determinar o estado de um certificado com maior frescura temporal.
- O cliente OCSP emite um pedido a um *OCSP responder* (Servidor) e suspende a aceitação do certificado até que este forneça uma resposta.
- Estão ainda previstos serviços análogos ao OCSP para *Delegated Path Validation* e *Delegated Path Discovery* (IETF RFC 3379).

Tópicos

- Parte VI: Acordo de chaves
- Parte VII: Criptografia de chave pública
- **Parte VIII: Infraestrutura de chave pública**
 - Arquitetura e aspectos operacionais
 - Certificados X.509
 - Cadeias de certificação / confiança
 - Perfis e políticas de certificados
 - Lista de revogação de certificados e OCSP
 - **Serviço de *Timestamp***
 - Legislação relevante (eIDAS, DL 12/2021, ...)
 - Utilização

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Serviço de *Timestamp*

- Associado a uma ICP, também existe normalmente um serviço de *Timestamp* (ou serviço de selo temporal).
- Porquê um serviço de selo temporal?
 - A assinatura ou a cifra de um documento não permitem ter uma prova de quando o documento existia (ou foi assinado/cifrado).
 - A prova de quando um documento existia (selo temporal), é de importância primordial nas transações bancárias, para estabelecer prova de propriedade intelectual, na resposta a concursos públicos, para evidência legal, entre outros.
- O que é um serviço de selo temporal?
 - É um serviço disponibilizado por uma terceira parte de confiança, que detém um relógio (usualmente uma *appliance*) sincronizado com fontes de tempo confiável, garantindo rastreabilidade para o tempo UTC(k) através de um dos laboratórios UTC(k) identificados pelo BIPM (*Bureau International des Poids et Mesures*) na sua Circular T (<https://www.bipm.org/en/bipm-services/timescales/time-ftp/Circular-T.html>).
 - Esse serviço gera uma prova (**selo temporal**) baseada no documento (para o qual se quer ter uma prova temporal) e na hora/data obtida do relógio com as características indicadas anteriormente.

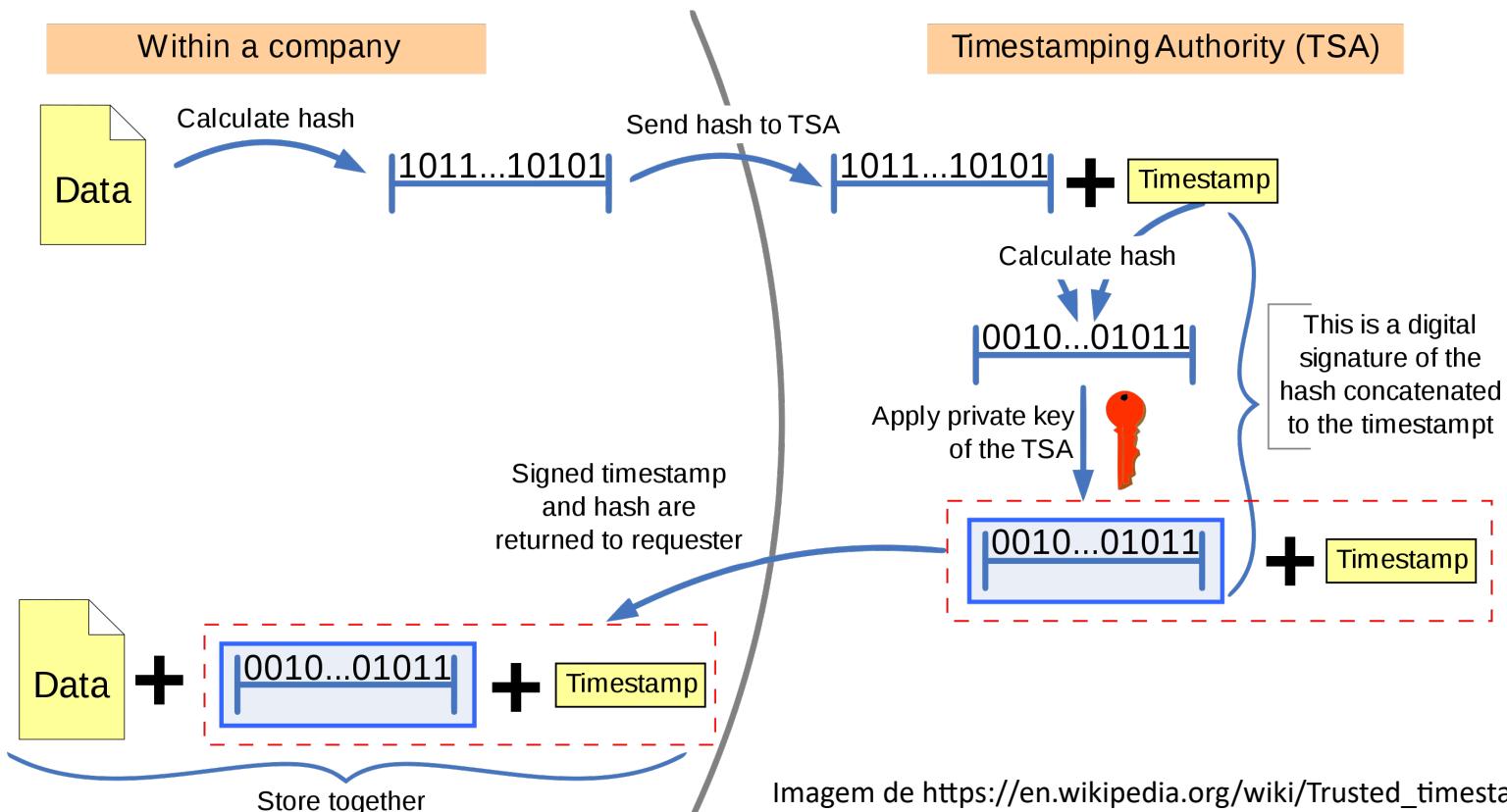
Serviço de *Timestamp*

- O que é um serviço confiável de selo temporal (digital)?
 - É um serviço de selo temporal que assina o hash do documento (para o qual se quer ter uma prova temporal) em conjunto com a hora obtida do relógio com as características indicadas anteriormente, de acordo com o formato IETF RFC 3161 (*Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)*) e, ETSI EN 319 422 (*Electronic Signatures and Infrastructures (ESI); Time-stamping protocol and time-stamp token profiles*) no caso de selo temporal qualificado (eIDAS).
- Exemplo de selo temporal
 - Selo temporal qualificado do Cartão de Cidadão, emitido pela Entidade de Validação Cronológica do Cartão de Cidadão.
 - Serviço disponível em <http://ts.cartaoecidadao.pt/rsa/server>;
 - Declaração de práticas de validação cronológica em http://pki.cartaoecidadao.pt/publico/politicas/PJ.CC_24.1.1_0005_pt.pdf.

Serviço de *Timestamp*

- Criação de selo temporal

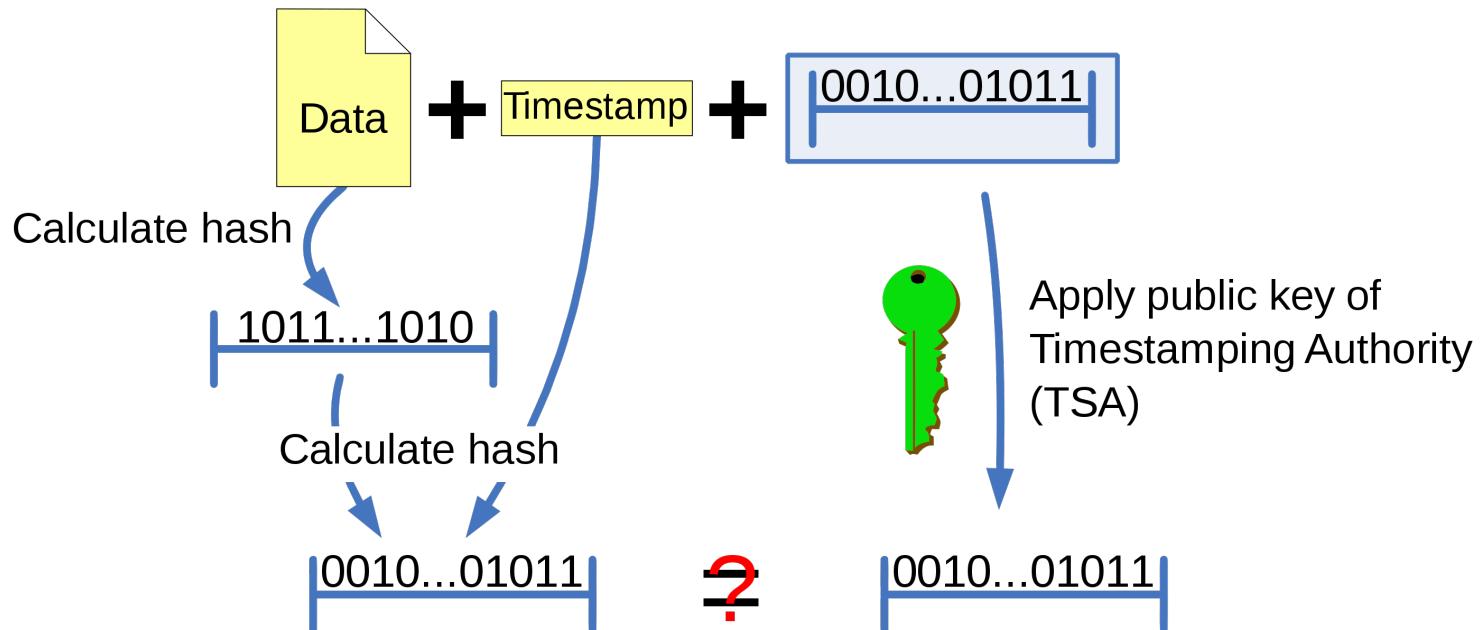
Trusted timestamping



Serviço de *Timestamp*

- Validação de selo temporal

Checking the trusted timestamp



If the calculated hashcode equals the result of the decrypted signature, neither the document or the timestamp was changed and the timestamp was issued by the TTP. If not, either of the previous statements is not true.

Imagen de https://en.wikipedia.org/wiki/Trusted_timestamping

Tópicos

- Parte VI: Acordo de chaves
- Parte VII: Criptografia de chave pública
- **Parte VIII: Infraestrutura de chave pública**
 - Arquitetura e aspectos operacionais
 - Certificados X.509
 - Cadeias de certificação / confiança
 - Perfis e políticas de certificados
 - Lista de revogação de certificados e OCSP
 - Serviço de *Timestamp*
 - **Legislação relevante (eIDAS, DL 12/2021, ...)**
 - Utilização

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Infraestrutura de chave pública – Legislação

- A legislação depende da área geográfica em que se vive. Em Portugal, a legislação diretamente aplicável é a seguinte:
 - REGULAMENTO (UE) N.º 910/2014 DO PARLAMENTO EUROPEU E DO CONSELHO de 23 de julho de 2014 relativo à identificação eletrónica e aos serviços de confiança para as transações eletrónicas no mercado interno e que revoga a Diretiva 1999/93/CE (usualmente designado por **Regulamento eIDAS**);
 - Decreto-Lei n.º 12/2021 de 09/02/2021, que assegura a execução na ordem jurídica interna do Regulamento (UE) 910/2014, relativo à identificação eletrónica e aos serviços de confiança para as transações eletrónicas no mercado interno

Infraestrutura de chave pública – Legislação

- A legislação depende da área geográfica em que se vive. Em Portugal, a legislação diretamente aplicável é a seguinte:
 - REGULAMENTO (UE) N.º 910/2014 DO PARLAMENTO EUROPEU E DO CONSELHO de 23 de julho de 2014 relativo à identificação eletrónica e aos serviços de confiança para as transações eletrónicas no mercado interno e que revoga a Diretiva 1999/93/CE (usualmente designado por **Regulamento eIDAS**);
 - Decreto-Lei n.º 12/2021 de 09/02/2021, que assegura a execução na ordem jurídica interna do Regulamento (UE) 910/2014, relativo à identificação eletrónica e aos serviços de confiança para as transações eletrónicas no mercado interno

Infraestrutura de chave pública – Legislação

- A legislação depende da área geográfica em que se vive. Em Portugal, a legislação diretamente aplicável é a seguinte:
 - REGULAMENTO (UE) N.º 910/2014 DO PARLAMENTO EUROPEU E DO CONSELHO de 23 de julho de 2014 relativo à identificação eletrónica e aos
- Mecanismos de identificação eletrónica notificados à UE de acordo com a DECISÃO DE EXECUÇÃO (UE) 2015/1984 DA COMISSÃO de 3 de novembro de 2015, com nível de garantia (LoA – Level of Assurance) Reducido / Substancial / Elevado (conforme REGULAMENTO DE EXECUÇÃO (UE) 2015/1502 DA COMISSÃO de 8 de setembro de 2015) - <https://ec.europa.eu/digital-building-blocks/wikis/display/EIDCOMMUNITY/Overview+of+pre-notified+and+notified+eID+schemes+under+eIDAS>
- Portugal tem dois mecanismos de identificação eletrónica notificados, ambos com LoA Elevado:
 - Certificado de Autenticação do Cartão de Cidadão;
 - Chave Móvel Digital (componente de autenticação).

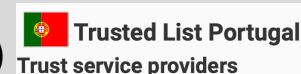
Cartão de Cidadão (CC)	Portuguese national identity card (eID card)	Portuguese Republic	High	AMA - Administrative Modernisation Agency Rua Abranches Ferrão n.º 10, 3º 1600 - 001 Lisbon ama@ama.pt +351 217231200	28.2.2019
------------------------	--	---------------------	------	--	-----------

Chave Móvel Digital (CMD)	Digital Mobile Key (mobile eID)	Portuguese Republic	High	AMA – Administrative Modernisation Agency Rua de Santa Marta 55, 3º 1150 – 294 Lisbon ama@ama.pt +351 217231200	8.4.2020
---------------------------	---------------------------------	---------------------	------	---	----------

Infraestrutura de chave pública – Legislação

- A legislação depende da área geográfica em que se vive. Em Portugal, a legislação diretamente aplicável é a seguinte:
 - REGULAMENTO (UE) N.º 910/2014 DO PARLAMENTO EUROPEU E DO CONSELHO de 23 de julho de 2014 relativo à identificação eletrónica e aos serviços de confiança para as transações eletrónicas no mercado interno e

- Lista dos serviços de confiança (<https://esignature.ec.europa.eu/efda/tl-browser/#/screen/home>):
 - Certificados qualificados de assinaturas eletrónicas (QCert for ESig)
 - Certificados qualificados de selos eletrónicos (QCert for ESeal)
 - Certificados qualificados de autenticação de sítios web (QWAC)
 - Serviço qualificado de validação de assinaturas eletrónicas qualificadas (QVal for QESig)
 - Serviço qualificado de preservação de assinaturas eletrónicas qualificadas (QPres for QESig)
 - Serviço qualificado de validação de selos eletrónicos qualificados (QVal for QESeal)
 - Serviço qualificado de preservação de selos eletrónicos qualificados (QPres for QESeal)
 - Selo temporal qualificado (QTimestamp)
 - Serviço qualificado de envio registado eletrónico (QeRDS)



Currently active trust service providers

ACIN iCloud Solutions, Lda AMA - AGÊNCIA PARA A MODERNIZAÇÃO ADMINISTRATIVA I. P.

CEGER - Centro de Gestão da Rede Informática do Governo

Instituto dos Registos e do Notariado I.P.

DigitalSign - Certificadora Digital

MULTICERT - Serviços de Certificação Electrónica S.A.

Infraestrutura de chave pública – Serviços Qualificados eIDAS

- Prestadores qualificados de serviços de confiança são auditados anualmente, por organismo de avaliação da conformidade (Regulamento eIDAS).
- Prestadores qualificados de serviços de confiança supervisionados por Entidade supervisora (Regulamento eIDAS) – em Portugal essa entidade é o Gabinete Nacional de Segurança (GNS).

Normas aplicáveis a serviços qualificados eIDAS:

- Normas desenvolvidas pelo ETSI, por delegação da UE.
- CEN/ISO
- RFC
- ...

Infraestrutura de chave pública – Serviços Qualificados eIDAS

Normas aplicáveis aos prestadores qualificados de serviços de confiança, para a emissão de certificados qualificados:

ETSI EN 319 401	Electronic Signatures and Infrastructures (ESI); General Policy Requirements for Trust Service Providers
ETSI TS 119 312	Electronic Signatures and Infrastructures (ESI); Cryptographic Suites
CEN EN 419 221-5	Protection Profiles for TSP Cryptographic Modules - Part 5: Cryptographic Module for Trust Services
CEN/TS 419 221-6	Conditions for use of EN 419221-5 as a qualified electronic signature or seal creation device
CEN EN 301 549	Accessibility requirements for ICT products and services
ETSI TR 103 684	Electronic Signatures and Infrastructures (ESI); Global Acceptance of EU Trust Services
ETSI TR 119 460	Electronic Signatures and Infrastructures (ESI); Survey of technologies and regulatory requirements for identity proofing for trust service subjects
ETSI TS 119 615	Electronic Signatures and Infrastructures (ESI); Trusted lists; Procedures for using and interpreting European Union Member States national trusted lists

IETF RFC 3647	Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework
ETSI EN 319 411-1	Electronic Signatures and Infrastructures (ESI); Policy and security requirements for Trust Service Providers issuing certificates; Part 1: General requirements
ETSI EN 319 411-2	Electronic Signatures and Infrastructures (ESI); Policy and security requirements for Trust Service Providers issuing certificates; Part 2: Requirements for trust service providers issuing EU qualified certificates
ETSI EN 319 412-1	Electronic Signatures and Infrastructures (ESI); Certificate Profiles; Part 1: Overview and common data structures
ETSI EN 319 412-2	Electronic Signatures and Infrastructures (ESI); Certificate Profiles; Part 2: Certificate profile for certificates issued to natural persons
ETSI EN 319 412-3	Electronic Signatures and Infrastructures (ESI); Certificate Profiles; Part 3: Certificate profile for certificates issued to legal persons
ETSI EN 319 412-4	Electronic Signatures and Infrastructures (ESI); Certificate Profiles; Part 4: Certificate profile for web site certificates
ETSI EN 319 412-5	Electronic Signatures and Infrastructures (ESI); Certificate Profiles; Part 5: QCStatements
ETSI TS 119 461	Electronic Signatures and Infrastructures (ESI); Policy and security requirements for trust service components providing identity proofing of trust service subjects
CEN EN 419 211-1	Protection profiles for secure signature creation device - Part 1: Overview
CEN EN 419 211-2	Protection profiles for secure signature creation device - Part 2: Device with key generation
CEN EN 419 211-3	Protection profiles for secure signature creation device - Part 3: Device with key import
CEN EN 419 211-4	Protection profiles for secure signature creation device - Part 4: Extension for device with key generation and trusted channel to certificate generation application
CEN EN 419 211-5	Protection profiles for secure signature creation device - Part 5: Extension for device with key generation and trusted channel to signature creation application
CEN EN 419 211-6	Protection profiles for secure signature creation device - Part 6: Extension for device with key import and trusted channel to signature creation application



Tópicos

- Parte VI: Acordo de chaves
- Parte VII: Criptografia de chave pública
- **Parte VIII: Infraestrutura de chave pública**
 - Arquitetura e aspectos operacionais
 - Certificados X.509
 - Cadeias de certificação / confiança
 - Perfis e políticas de certificados
 - Lista de revogação de certificados e OCSP
 - Serviço de *Timestamp*
 - Legislação relevante (eIDAS, DL 12/2021, ...)
 - **Utilização**

Nota: Apontamentos baseados nos slides de “Tecnologia Criptográfica” do Professor José Bacelar Almeida (com permissão do mesmo)

Infraestrutura de chave pública - Utilização

- A utilização de infraestrutura de chave pública deve ser considerada quando for apropriado ao seu caso de uso.
- Não necessita (nem deve) desenvolver o código para as operações ou funções necessárias, já que existem bibliotecas/APIs que já disponibilizam o código necessário (i.e., as operações base para operar/utilizar a infraestrutura de chave pública). Por exemplo:
 - Em Python, pode utilizar a cryptography (<https://cryptography.io/>) e/ou pyOpenSSL (<https://pypi.org/project/pyOpenSSL/>);
 - Em Javascript ou Node.js pode utilizar o crypto (<https://nodejs.org/api/crypto.html>) e tls (<https://nodejs.org/api/tls.html>).
 - Em Java, tal como referido para as cifras simétricas, pode utilizar
 - os *default providers* da Sun (propriedade da Oracle), nomeadamente SUN, SunJCE, SunPKCS11, ...;
 - O provider do Bouncy Castle (<https://www.bouncycastle.org/java.html>).

Infraestrutura de chave pública - Utilização

- Exemplo em python, utilizando o cryptography

```
from datetime import datetime, timedelta
from cryptography import x509
from cryptography.x509.oid import NameOID
from cryptography.hazmat.primitives import hashes
from cryptography.hazmat.backends import default_backend
from cryptography.hazmat.primitives import serialization
from cryptography.hazmat.primitives.asymmetric import rsa
# Gerar par de chaves RSA de 3072 bits
key = rsa.generate_private_key(public_exponent=65537, key_size=3072, backend=default_backend())
# Distinguished Name
name = x509.Name([x509.NameAttribute(NameOID.COMMON_NAME, u"Entidade de Certificação UMinho"),
                  x509.NameAttribute(NameOID.COUNTRY_NAME, u"PT"),
                  x509.NameAttribute(NameOID.ORGANIZATIONAL_UNIT_NAME, u"Departamento de Informática"),
                  x509.NameAttribute(NameOID.ORGANIZATION_NAME, u"Universidade do Minho")])
# path_len=1 significa que pode assinar um nível de certificados.
basic_constraints = x509.BasicConstraints(ca=True, path_length=1)
now = datetime.utcnow() # hora actual
# emissão de certificado auto-assinado
cert = (x509.CertificateBuilder()
         .subject_name(name)
         .issuer_name(name)
         .public_key(key.public_key())
         .serial_number(12345)
         .not_valid_before(now)
         .not_valid_after(now + timedelta(days=10*365))
         .add_extension(basic_constraints, False)
         .sign(key, hashes.SHA256(), default_backend()))
print(cert.public_bytes(encoding=serialization.Encoding.PEM)) # imprime certificado em formato PEM
print(key.private_bytes(encoding=serialization.Encoding.PEM, format=serialization.PrivateFormat.TraditionalOpenSSL,
                       encryption_algorithm=serialization.NoEncryption())) # imprime chave privada em formato PEM, não cifrada
```



Infraestrutura de chave pública – Utilização com openssl

- O openssl (<https://www.openssl.org>) é um toolkit ("canivete suíço") para criptografia e comunicações seguras.
 - Emissão de certificado EC root auto-assinado

```
# Geração de par de chaves RSA com 4096 bits
```

```
openssl genrsa -out rootCA.key 4096
```

```
# Criação de certificado auto-assinado com validade de 365 dias
```

```
openssl req -x509 -new -nodes -key rootCA.key -sha256 -days 365 -out rootCA.crt -subj "/C=PT/O=Universidade do Minho/OU=Departamento de Informática/CN=Entidade de Certificação UMinho"
```

```
# Ver o conteúdo do certificado
```

```
openssl x509 -in rootCA.crt -noout -text
```

```
debian@vm5:/tmp$ openssl x509 -in rootCA.crt -noout -text
Certificate:
Data:
    Version: 3 (0x2)
    Serial Number:
        19:64:a2:d9:d1:a6:fa:be:7b:0f:fa:a6:60:09:ab:a0:06:72:c3:b5
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C = PT, O = Universidade do Minho, OU = Departamento de Inform\83\C2\A1tica, CN = Entidade de Certifica\83\C2\A7\83\C2\A3o UMinho
    Validity
        Not Before: Mar 28 00:23:43 2022 GMT
        Not After : Mar 28 00:23:43 2023 GMT
    Subject: C = PT, O = Universidade do Minho, OU = Departamento de Inform\83\C2\A1tica, CN = Entidade de Certifica\83\C2\A7\83\C2\A3o UMinho
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
        RSA Public-Key: (4096 bit)
            Modulus:
                00:c9:1e:a2:ab:c3:65:04:a8:ab:47:af:11:03:a3:
                d0:1b:67:56:19:fd:90:90:9c:de:70:a3:a2:7a:bc:
```

Infraestrutura de chave pública – Utilização com openssl

- Emissão de certificado TSL para servidor Web pela EC root auto-assinada

```
# Geração de par de chaves RSA com 3072 bits
```

```
openssl genrsa -out univminho.pt.key 3072
```

```
# Criação de certificado request (CSR)
```

```
openssl req -new -sha256 -key univminho.pt.key -subj "/C=PT/O=Universidade do  
Minho/CN=www.univminho.pt" -out univminho.pt.csr
```

```
# Ver o conteúdo do certificate request
```

```
openssl req -in univminho.pt.csr -noout -text
```

```
# Assinar o certificate request pela EC root auto-assinada, com uma validade de 180 dias
```

```
openssl x509 -req -in univminho.pt.csr -CA rootCA.crt -CAkey rootCA.key -CAcreateserial -  
out univminho.pt.crt -days 180 -sha256
```

```
# Ver o conteúdo do certificado
```

```
openssl x509 -in univminho.pt.crt -noout -text
```

```
debian@vm5:/tmp$ openssl x509 -in univminho.pt.crt -noout -text  
Certificate:  
Data:  
    Version: 1 (0x0)  
    Serial Number:  
        22:cf:65:d9:5e:10:14:51:54:14:c5:b7:8f:d9:7e:a9:36:76:26  
Signature Algorithm: sha256WithRSAEncryption  
Issuer: C = PT, O = Universidade do Minho, OU = Departamento de Inform\c3\83\C2\A1tica, CN = Entidade de Certifica\c3\83\C2\A7\c3\83\C2\A3o UMinho  
Validity  
    Not Before: Mar 28 00:39:09 2022 GMT  
    Not After : Sep 24 00:39:09 2022 GMT  
Subject: C = PT, O = Universidade do Minho, CN = www.univminho.pt  
Subject Public Key Info:  
    Public Key Algorithm: rsaEncryption  
        RSA Public-Key: (3072 bit)  
        Modulus:  
            00:c3:9b:6b:3d:81:e2:cf:23:80:22:a2:25:4d:d6:
```

Infraestrutura de chave pública – Utilização com openssl

- Emissão de certificado TSL para servidor Web pela EC root auto-assinada

Também pode ver o conteúdo do certificado com
openssl asn1parse -in univminho.pt.crt

```
debian@vm5:/tmp$ openssl asn1parse -in univminho.pt.crt
0:d=0 hl=4 l=1238 cons: SEQUENCE
4:d=1 hl=4 l= 702 cons: SEQUENCE
8:d=2 hl=2 l= 20 prim: INTEGER          :22CF65D95E1014515414C56FB78FD97EA9367626
30:d=2 hl=2 l= 13 cons: SEQUENCE
32:d=3 hl=2 l=  9 prim: OBJECT         :sha256WithRSAEncryption
43:d=3 hl=2 l=  0 prim: NULL
45:d=2 hl=3 l= 134 cons: SEQUENCE
48:d=3 hl=2 l= 11 cons: SET
50:d=4 hl=2 l=  9 cons: SEQUENCE
52:d=5 hl=2 l=  3 prim: OBJECT        :countryName
57:d=5 hl=2 l=  2 prim: PRINTABLESTRING :PT
61:d=3 hl=2 l= 30 cons: SET
63:d=4 hl=2 l= 28 cons: SEQUENCE
65:d=5 hl=2 l=  3 prim: OBJECT        :organizationName
70:d=5 hl=2 l= 21 prim: UTF8STRING    :Universidade do Minho
93:d=3 hl=2 l= 39 cons: SET
95:d=4 hl=2 l= 37 cons: SEQUENCE
97:d=5 hl=2 l=  3 prim: OBJECT        :organizationalUnitName
102:d=5 hl=2 l= 30 prim: UTF8STRING    :Departamento de InformÁtica
134:d=3 hl=2 l= 46 cons: SET
136:d=4 hl=2 l= 44 cons: SEQUENCE
138:d=5 hl=2 l=  3 prim: OBJECT        :commonName
143:d=5 hl=2 l= 37 prim: UTF8STRING    :Entidade de CertificaÃ§Ã£o UMinho
182:d=2 hl=2 l= 30 cons: SEQUENCE
184:d=3 hl=2 l= 13 prim: UTCTIME      :220328003909Z
199:d=3 hl=2 l= 13 prim: UTCTIME      :220924003909Z
214:d=2 hl=2 l= 72 cons: SEQUENCE
```

```
216:d=3 hl=2 l= 11 cons: SET
218:d=4 hl=2 l=  9 cons: SEQUENCE
220:d=5 hl=2 l=  3 prim: OBJECT        :countryName
225:d=5 hl=2 l=  2 prim: PRINTABLESTRING :PT
229:d=3 hl=2 l= 30 cons: SET
231:d=4 hl=2 l= 28 cons: SEQUENCE
233:d=5 hl=2 l=  3 prim: OBJECT        :organizationName
238:d=5 hl=2 l= 21 prim: UTF8STRING    :Universidade do Minho
261:d=3 hl=2 l= 25 cons: SET
263:d=4 hl=2 l= 23 cons: SEQUENCE
265:d=5 hl=2 l=  3 prim: OBJECT        :commonName
270:d=5 hl=2 l= 16 prim: UTF8STRING    :www.univminho.pt
288:d=2 hl=4 l= 418 cons: SEQUENCE
292:d=3 hl=2 l= 13 cons: SEQUENCE
294:d=4 hl=2 l=  9 prim: OBJECT        :rsaEncryption
305:d=4 hl=2 l=  0 prim: NULL
307:d=3 hl=4 l= 399 prim: BIT STRING
710:d=1 hl=2 l= 13 cons: SEQUENCE
712:d=2 hl=2 l=  9 prim: OBJECT        :sha256WithRSAEncryption
723:d=2 hl=2 l=  0 prim: NULL
725:d=1 hl=4 l= 513 prim: BIT STRING
```



Infraestrutura de chave pública – Utilização com openssl

- Inspecionar lista de revogação de certificados (CRL)

Primeiro há que ir buscar a CRL indicada no seu certificado

```
X509v3 CRL Distribution Points:  
  
Full Name:  
URI:http://pki.cartaoecidadao.pt/publico/lrc/cc_sub-ec_cidadao_cmd_crl0003_p0001.crl
```

```
wget http://pki.cartaoecidadao.pt/publico/lrc/cc_sub-ec_cidadao_cmd_crl0003_p0001.crl
```

Ver o conteúdo da CRL

```
openssl crl -inform DER -text -noout -in cc_sub-ec_cidadao_cmd_crl0003_p0001.crl
```

```
Certificate Revocation List (CRL):  
Version 2 (0x1)  
Signature Algorithm: sha256WithRSAEncryption  
Issuer: C = PT, O = AMA - AG\C3\8ANCIA PARA A MODERNIZA\C3\87\C3\830 ADMINISTRATIVA I. P., OU = Cart\C3\A3o de Cidad\C3\A3o, OU = subECEstado, CN = EC de Chave M  
\C3\B3vel Digital de Assinatura Digital Qualificada do Cart\C3\A3o de Cidad\C3\A3o 00003  
Last Update: Mar 25 19:01:37 2022 GMT  
Next Update: Apr 1 19:01:37 2022 GMT  
CRL extensions:  
X509v3 Authority Key Identifier:  
keyid:6A:C6:E5:B3:7A:BC:06:74:EF:E7:90:A6:96:D8:70:C3:B7:9A:83:A4  
  
X509v3 CRL Number:  
816  
X509v3 Issuing Distribution Point: critical  
Full Name:  
URI:http://pki.cartaoecidadao.pt/publico/lrc/cc_sub-ec_cidadao_cmd_crl0003_p0001.crl  
  
X509v3 Freshest CRL:  
  
Full Name:  
URI:http://pki.cartaoecidadao.pt/publico/lrc/cc_sub-ec_cidadao_cmd_crl0003_delta_p0001.crl  
  
Revoked Certificates:  
Serial Number: 743C3042C1DA962A  
Revocation Date: Dec 8 12:37:47 2020 GMT  
Serial Number: 2DF208F502F96B90  
Revocation Date: Feb 19 09:37:29 2021 GMT
```



Infraestrutura de chave pública – Utilização com openssl

- Inspecionar lista de revogação de certificados (CRL)

Para ver se o seu certificado faz parte da CRL

```
openssl crl -inform DER -text -noout -in cc_sub-ec_cidadao_cmd_crl0003_p0001.crl | grep -i -A 1 SERIAL_NUMBER_DO_CERTIFICADO
```

```
debian@vm5:/tmp$ openssl crl -inform DER -text -noout -in cc_sub-ec_cidadao_cmd_crl0003_p0001.crl | grep -i -A 1 584d79eca592b100
Serial Number: 584D79ECA592B100
Revocation Date: Apr 25 14:51:26 2020 GMT
```

Ver quando foi o último update da CRL

```
openssl crl -inform DER -lastupdate -noout -in cc_sub-ec_cidadao_cmd_crl0003_p0001.crl
```

```
debian@vm5:/tmp$ openssl crl -inform DER -lastupdate -noout -in cc_sub-ec_cidadao_cmd_crl0003_p0001.crl
lastUpdate=Mar 25 19:01:37 2022 GMT
```

Ver quando é o próximo update da CRL

```
openssl crl -inform DER -nextupdate -noout -in cc_sub-ec_cidadao_cmd_crl0003_p0001.crl
```

```
debian@vm5:/tmp$ openssl crl -inform DER -nextupdate -noout -in cc_sub-ec_cidadao_cmd_crl0003_p0001.crl
nextUpdate=Apr 1 19:01:37 2022 GMT
```

Infraestrutura de chave pública – Utilização com openssl

– Obter informação de revogação do serviço OCSP

Primeiro há que obter o URL do serviço OCSP indicado no seu certificado

```
openssl x509 -noout -ocsp_uri -in <certificado>
```

```
?1 jepm@minione ECCMD (WP1+WP5)Gitlab % openssl x509 -noout -ocsp_uri -in Signuser.CMD00003.pem  
http://ocsp.cmd.cartaodecidadao.pt/publico/ocsp
```

Tem também que obter o certificado da EC que emitiu o seu certificado. No caso do Cartão de Cidadão ou da Chave Móvel Digital, encontra-os em <http://pki.cartaoecidadao.pt>.

O comando genérico para obter informação de revogação é o seguinte:

```
openssl ocsp -issuer <cert EC> -cert <cert enduser> -url <URL OCSP> -text
```

```
✓ jepm@minione ECCMD (WP1+WP5)Gitlab % openssl ocsp -issuer CMD003.pem -cert Signuser.CMD00003.pem -url http://ocsp.cmd.cartaoecidadao.pt/publico/ocsp -text  
OCSP Request Data:  
Version: 1 (0x0)  
Requestor List:  
Certificate ID:  
    Hash Algorithm: sha1  
    Issuer Name Hash: 192F56C017F5AA65B3DBC084EA6E98B07D92F6F8  
    Issuer Key Hash: 6AC6E5B37ABC0674EFE790A696D870C3B79A83A4  
    Serial Number: 584D79ECA592B100  
Request Extensions:  
    OCSP Nonce:  
        0410F188A5FB011094FF2BB4C9542FCE7056  
OCSP Response Data:  
OCSP Response Status: successful (0x0)  
Response Type: Basic OCSP Response  
Version: 1 (0x0)  
Responder Id: C = PT, O = Cartão de Cidadão, OU = Serviços do Cartão de Cidadão, OU = Validação on-line, CN = Serviço de Chave Móvel de Assinatura Digital Qualificada  
Produced At: Mar 28 01:30:19 2022 GMT  
Responses:  
Certificate ID:  
    Hash Algorithm: sha1  
    Issuer Name Hash: 192F56C017F5AA65B3DBC084EA6E98B07D92F6F8  
    Issuer Key Hash: 6AC6E5B37ABC0674EFE790A696D870C3B79A83A4  
    Serial Number: 584D79ECA592B100  
Cert Status: revoked  
Revocation Time: Apr 25 14:51:27 2020 GMT  
Revocation Reason: unspecified (0x0)  
This Update: Mar 28 01:30:19 2022 GMT
```

Infraestrutura de chave pública – Utilização com openssl

- Obter informação de revogação do serviço OCSP

Ou de um modo mais simples:

```
openssl ocsp -issuer <cert EC> -cert <cert enduser> -url <URL OCSP> -noverify
```

```
jepm@minione ECCMD (WP1+WP5)Gitlab % openssl ocsp -issuer CMD003.pem -cert Signuser.CMD00003.pem -url http://ocsp.cmd.cartaodecidadao.pt/publico/ocsp -noverify
Signuser.CMD00003.pem: revoked
  This Update: Mar 28 01:19:14 2022 GMT
  Reason: unspecified
  Revocation Time: Apr 25 14:51:27 2020 GMT
```



Infraestrutura de chave pública – Utilização com openssl

– Timestamp

1. Criar timestamp request

```
openssl ts -query -data <ficheiro a timestampar> -cert -sha256 -no_nonce -out request.tsq
```

2. Obter o timestamp

```
cat request.tsq | curl -s -S -H 'Content-Type: application/timestamp-query' --data-binary @-  
http://ts.cartaodecidadao.pt/tsa/server -o response.tsr
```

#3. Ver a resposta

```
openssl ts -reply -in response.tsr -text
```

```
?1 jepm@minione aux (develop)CC-newDocs % openssl ts -reply -in response.tsr -text  
Status info:  
Status: Granted.  
Status description: TS Service Status  
Failure info: unspecified  
  
TST info:  
Version: 1  
Policy OID: 0.4.0.2023.1.1  
Hash Algorithm: sha256  
Message data:  
    0000 - c7 ef f6 67 d2 c4 6d 9e-ed c6 c0 e1 0a 2a f2 d8 ...g..m.....*..  
    0010 - a0 e5 44 07 09 2b a1 7a-45 d7 70 d2 15 55 1d 1f ..D..+.zE.p..U..  
Serial number: 0x236B0EC04D47BA4207D0359099187B0A00128206  
Time stamp: Mar 28 01:41:16.386 2022 GMT  
Accuracy: 0x0 seconds, unspecified millis, 0x8B microseconds  
Ordering: no  
Nonce: unspecified  
TSA: unspecified  
Extensions:
```

Infraestrutura de chave pública – Utilização com openssl

– Timestamp

#4. Verificar a resposta

Nota: neste caso, a cadeia de certificação vai desde a ECRAizEstado até à TSA, por essa ordem

```
openssl ts -verify -in response.tsr -queryfile request.tsq -CAfile <cadeia de certificação>
```

```
?1 jepm@minione aux (develop)CC-newDocs % openssl ts -verify -in response.tsr -queryfile request.tsq -CAfile TSACartaoCidadao000011.bundle.pem
Using configuration from /opt/homebrew/etc/openssl@3/openssl.cnf
Verification: OK
```

