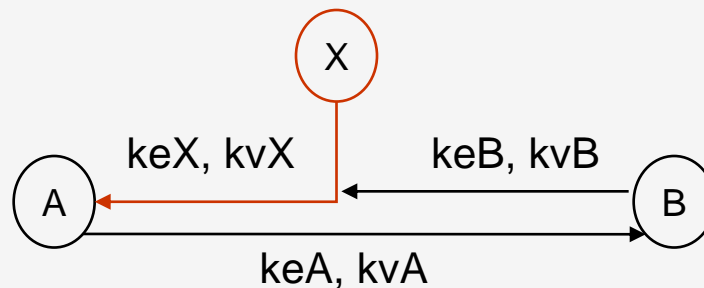

Certificados X.509 e perfil PKIX

Notas para a UC de “Segurança Informática”
Inverno de 11/12

Pedro Félix (pedrofelix@cc.isel.ipl.pt)
[Instituto Superior de Engenharia de Lisboa](#)

Autenticação de chaves públicas

- Autenticidade de chaves públicas
 - “A chave *Key* pertence a *Name*?”
- As chaves públicas tem de ter garantia de autenticidade
 - Certificados – associação (*identidade, chave pública*) certificada



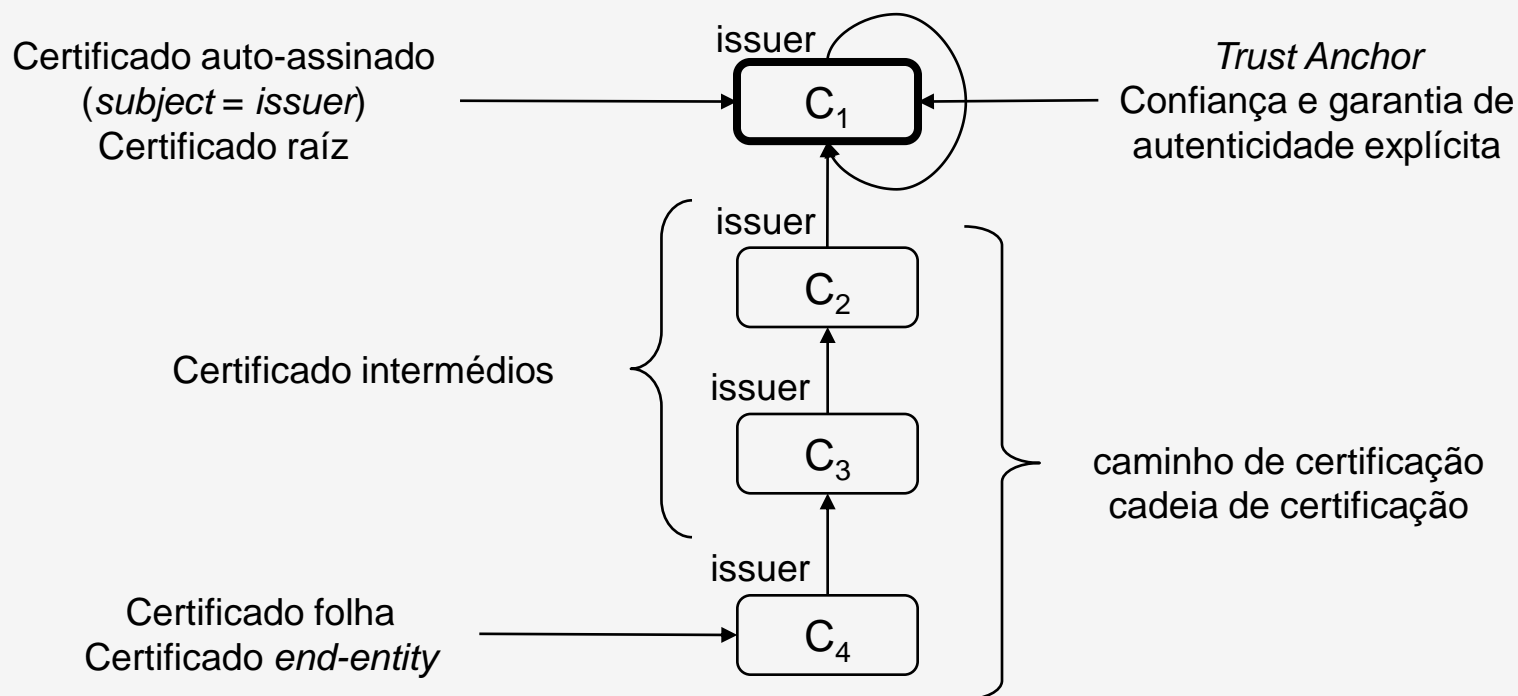
- A usa as chaves públicas de X em vez das de B
 - X decifra as mensagens enviadas para B
 - A verifica as mensagens assinadas por X como sendo de B

Certificados: introdução

- Constituição dum certificado
 - Quem certifica – emissor
 - O que certifica
 - Outros atributos – validade, condições de aplicabilidade
 - Assinatura do emissor
- Certificados X.509
 - Quem certifica (emissor): Autoridade de Certificação (AC)
 - O que certifica: associação entre uma *chave pública* e um *nome* (identidade)
 - Outros atributos – validade, usos da chaves, extensões
 - Assinatura do emissor – assinatura digital realizada com a chave de assinatura (privada) do emissor

Caminho de certificação

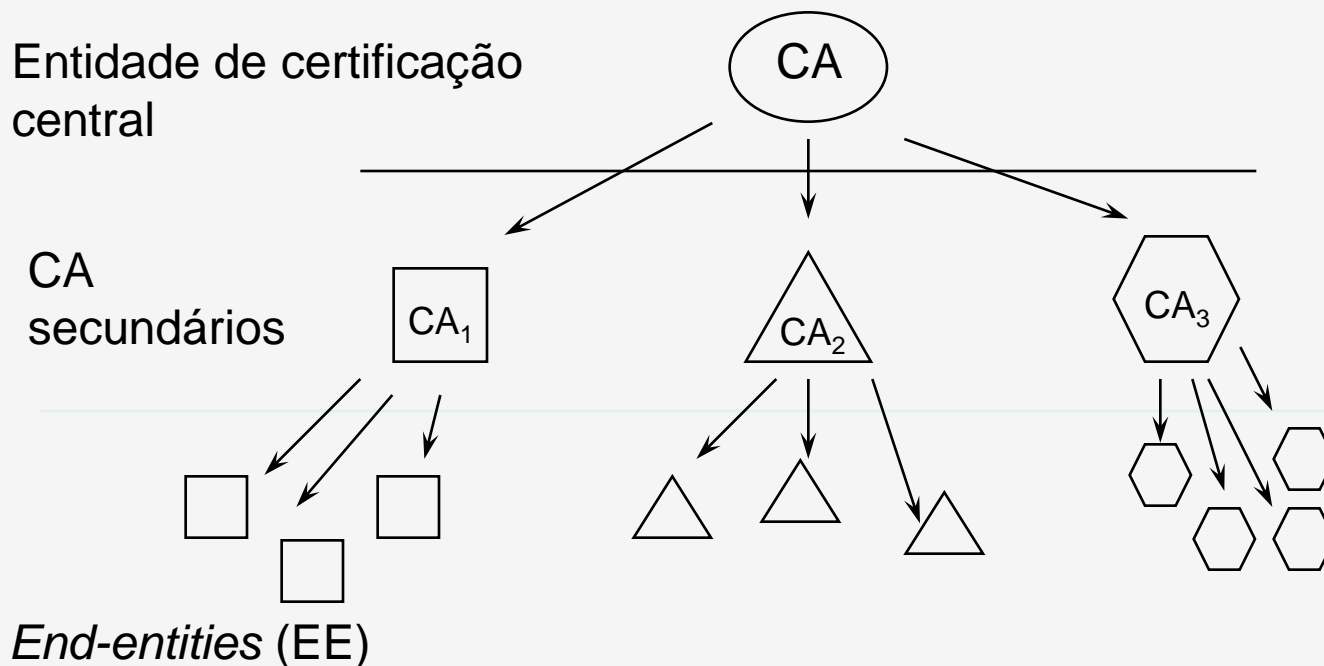
- Recursão
 - Obter chave pública \Rightarrow validar certificado \Rightarrow obter chave pública (do *issuer*)
- Condição de paragem
 - *Trust anchor* - Certificado auto-assinado (*issuer* = *subject*)



Validação de certificados

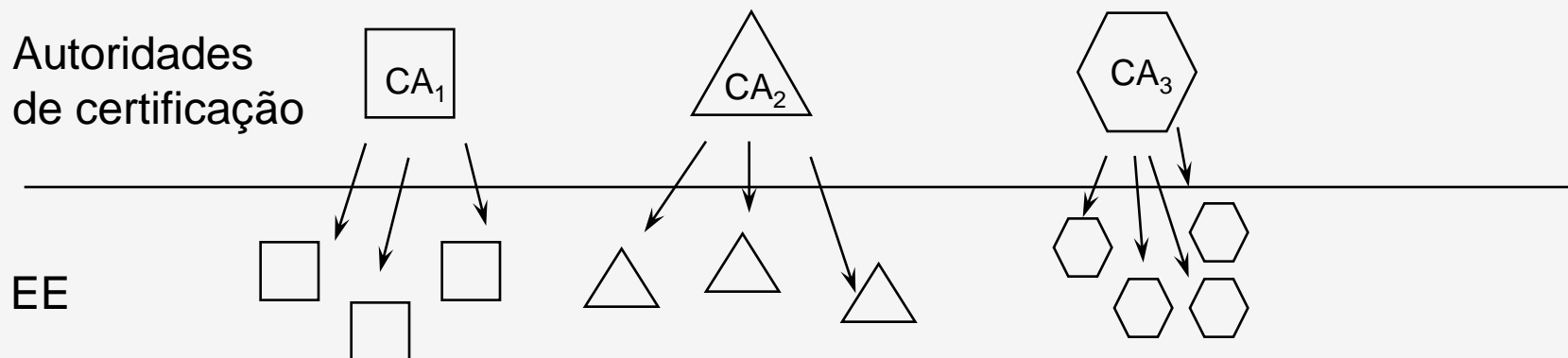
- Validação de certificados
 - verificar a associação entre uma identidade e uma chave pública
 - verificar a aplicabilidade do certificado para a utilização considerada
- Um caminho de certificação é uma sequência de n certificados onde
 - Para qualquer $i \in \{0, n-2\}$
 - **$C[i].\text{subject} = C[i+1].\text{issuer}$**
 - **$C[0]$** é um certificado emitido por um *trust anchor*
 - **$C[n-1]$** é o certificado a validar

Modelo de hierarquia estrita



- Todos os elementos devem possuir a chave pública de CA com garantia de autenticidade implícita
- Para um elemento da rede i obter a chave pública (autenticada) de um elemento A da rede j , basta obter o certificado de A_j e o certificado de CA_j (cadeia de certificados de A_j)

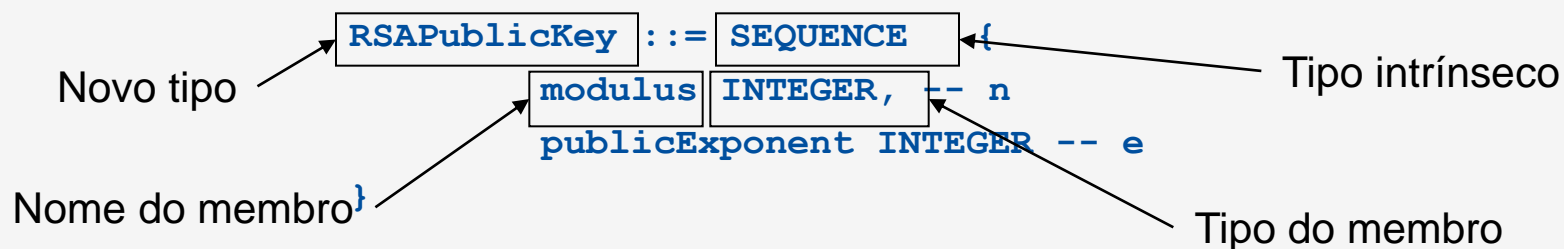
Modelo de domínios separados



- Para existir interoperabilidade, todos os utilizadores devem confiar em todas as autoridades de certificação
- Modelo usado na *Internet* – não existe CA central

Síntaxe - ASN.1

- *Abstract Syntax Notation 1*
 - Síntaxe e regras para a especificação de objectos abstractos
- Regras de codificação
 - Forma de representar os objectos abstractos como sequências de *bits*
 - DER – *Distinguished Encoding Rules*
 - BER – *Basic Encoding Rules*
- *Object Identifier* (OID)
 - Identificador único constituído por uma sequência de inteiros que representa uma hierarquia
 - Ex.: RSA "1.2.840.113549.1.1.1"
- Exemplo
 - chave pública RSA (norma PKCS #1)



Certificado X.509: constituição (1)

- Certificado

```
Certificate ::= SEQUENCE {  
  tbsCertificate      TBSCertificate,  
  signatureAlgorithm AlgorithmIdentifier,  
  signatureValue     BIT STRING  
}
```

- Informação assinada

```
TBSCertificate ::= SEQUENCE {  
  version [0] EXPLICIT Version DEFAULT v1,  
  serialNumber CertificateSerialNumber,  
  signature AlgorithmIdentifier,  
  issuer Name,  
  validity Validity,  
  subject Name,  
  subjectPublicKeyInfo SubjectPublicKeyInfo,  
  issuerUniqueID [1] IMPLICIT UniqueIdentifier OPTIONAL,  
    -- If present, version shall be v2 or v3  
  subjectUniqueID [2] IMPLICIT UniqueIdentifier OPTIONAL,  
    -- If present, version shall be v2 or v3  
  extensions [3] EXPLICIT Extensions OPTIONAL  
    -- If present, version shall be v3  
}
```

Certificado X.509: constituição (2)

- Validade


```
Validity ::= SEQUENCE {  
    notBefore      Time,  
    notAfter       Time  
}
```

- Chave pública

```
SubjectPublicKeyInfo ::= SEQUENCE {  
    algorithm      AlgorithmIdentifier,  
    subjectPublicKey BIT STRING  
}
```

- Extensões

```
Extensions ::= SEQUENCE SIZE (1..MAX) OF Extension
```



```
Extension ::= SEQUENCE {  
    extnID      OBJECT IDENTIFIER,  
    critical    BOOLEAN DEFAULT FALSE,  
    extnValue   OCTET STRING  
}
```

Extensões

- A versão 3 da norma X.509 acrescenta *extensões* à informação assinada (`tbsCertificate`)
- As extensões são a forma normalizada de acrescentar informação não considerada na norma base
- Constituição duma extensão:
 - Identificador da extensão
 - Valor da extensão
 - *flag critical* (se verdadeira, a extensão não pode ser ignorada)
- Perfil
 - Conjunto de extensões e respectiva semântica, usados num domínio de aplicação
 - ex.:
 - PKIX - *Public Key Infrastructure for the Internet*

- Algumas extensões:
 - *Authority Key Identifier* – identificador da chave do emissor
 - *Subject Key Identifier* – identificador da chave do *subject*
 - *Key Usage* – usos permitidos para o par de chaves
 - *Alternative Name* – nome alternativo (email, IP, URI)
 - *Policy Identifiers* – identificador de política
 - *Basic Constraints* – restrições ao uso do certificado
 - *Name Constraints* – restrições ao espaço de nomes do certificado
 - *Policy Constraints* – restrições de política
 - *Extended Key Usage* – usos permitidos para o par de chaves
 - *CRL Distribution Points* – pontos de distribuição das listas de revogação

KeyUsage

- Usos permitidos para o par de chaves

id-ce-keyUsage OBJECT IDENTIFIER ::= { id-ce 15 }

```
KeyUsage ::= BIT STRING {  
    digitalSignature      (0),  
    nonRepudiation       (1),  
    keyEncipherment      (2),  
    dataEncipherment     (3),  
    keyAgreement         (4),  
    keyCertSign          (5),  
    cRLSign              (6),  
    encipherOnly         (7),  
    decipherOnly         (8)  
}
```

Políticas

- A extensão *certificate policies* contém uma sequência de *policy information items*
- Cada *policy information* item é constituído por um OID e um qualificador opcional e indica a política associada à emissão e utilização do certificado
- Aplicações com requisitos próprios contem um lista de políticas que são utilizadas para comparar com os OID do certificado
- Os dois qualificadores mais usados são
 - ponteiro para o *Certification Practice Statement* (CPS) da entidade emissora
 - *User notice* – informação a mostrar ao utilizador quando o certificado é utilizado

Políticas

```
id-ce-certificatePolicies OBJECT IDENTIFIER ::= { id-ce 32 }

certificatePolicies ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
    policyIdentifier    CertPolicyId,
    policyQualifiers    SEQUENCE SIZE (1..MAX)
                        OF PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
    policyQualifierId  PolicyQualifierId,
    qualifier          ANY DEFINED BY policyQualifierId }

id-qt                OBJECT IDENTIFIER ::= { id-pkix 2 }
id-qt-cps             OBJECT IDENTIFIER ::= { id-qt 1 }
id-qt-unotice         OBJECT IDENTIFIER ::= { id-qt 2 }

Qualifier ::= CHOICE {
    cPSuri             CPSuri,
    userNotice         UserNotice }
```

Subject Alternative Name

- Nome alternativo para o *subject*

```
id-ce-subjectAltName OBJECT IDENTIFIER ::= { id-ce 17 }
```

```
SubjectAltName ::= GeneralNames
```

```
GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName
```

```
GeneralName ::= CHOICE {  
    otherName          [0]      OtherName,  
    rfc822Name         [1]      IA5String,  
    dNSName            [2]      IA5String,  
    x400Address        [3]      ORAddress,  
    directoryName      [4]      Name,  
    ediPartyName       [5]      EDIPartyName,  
    uniformResourceIdentifier [6]  IA5String,  
    iPAddress          [7]      OCTET STRING,  
    registeredID       [8]      OBJECT IDENTIFIER }
```


Basic Constraints

- A extensão *basic constraints* indica se o *subject* é uma autoridade de certificação e qual a maior dimensão do caminho de certificação

```
id-ce-basicConstraints OBJECT IDENTIFIER ::= { id-ce 19 }
```

```
BasicConstraints ::= SEQUENCE {  
    cA                      BOOLEAN DEFAULT FALSE,  
    pathLenConstraint       INTEGER (0..MAX) OPTIONAL }
```

Name Constraints

- A extensão *name constraints*, usada apenas no certificado do CA, define o espaço de nomes ao qual todos os nomes de *subject* devem pertencer

```
id-ce-nameConstraints OBJECT IDENTIFIER ::= { id-ce 30 }
```

```
NameConstraints ::= SEQUENCE {  
    permittedSubtrees      [0]      GeneralSubtrees OPTIONAL,  
    excludedSubtrees       [1]      GeneralSubtrees OPTIONAL }
```

```
GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree
```

```
GeneralSubtree ::= SEQUENCE {  
    base                GeneralName,  
    minimum              [0]      BaseDistance DEFAULT 0,  
    maximum              [1]      BaseDistance OPTIONAL }
```

```
BaseDistance ::= INTEGER (0..MAX)
```

Extended key usage

```
id-ce-extKeyUsage OBJECT IDENTIFIER ::= {id-ce 37}
```

```
ExtKeyUsageSyntax ::= SEQUENCE SIZE (1..MAX) OF KeyPurposeId
```

```
KeyPurposeId ::= OBJECT IDENTIFIER
```

```
id-kp OBJECT IDENTIFIER ::= { id-pkix 3 }
```

```
id-kp-serverAuth OBJECT IDENTIFIER ::= {id-kp 1}
```

```
-- TLS Web server authentication
```

```
-- Key usage bits that may be consistent: digitalSignature, keyEncipherment or  
keyAgreement
```

```
id-kp-clientAuth OBJECT IDENTIFIER ::= {id-kp 2}
```

```
-- TLS Web client authentication
```

```
-- Key usage bits that may be consistent: digitalSignature and/or keyAgreement
```

```
id-kp-codeSigning OBJECT IDENTIFIER ::= {id-kp 3}
```

```
-- Signing of downloadable executable code
```

```
-- Key usage bits that may be consistent: digitalSignature
```

```
id-kp-emailProtection OBJECT IDENTIFIER ::= {id-kp 4}
```

```
-- E-mail protection
```

```
-- Key usage bits that may be consistent: digitalSignature, nonRepudiation, and/or  
-- (keyEncipherment or keyAgreement)
```

```
id-kp-timeStamping OBJECT IDENTIFIER ::= { id-kp 8 }
```

```
-- Binding the hash of an object to a time from an agreed-upon time
```

```
-- source. Key usage bits that may be consistent: digitalSignature, nonRepudiation
```

CRL Distribution Points

- A extensão *CRL Distribution Points* define a forma como as listas de revogação podem ser obtidas

```
id-ce-cRLDistributionPoints OBJECT IDENTIFIER ::= { id-ce 31 }
```

```
cRLDistributionPoints ::= {  
    CRLDistPointsSyntax }
```

```
CRLDistPointsSyntax ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint
```

```
DistributionPoint ::= SEQUENCE {  
    distributionPoint      [0]      DistributionPointName OPTIONAL,  
    reasons                [1]      ReasonFlags OPTIONAL,  
    cRLIssuer              [2]      GeneralNames OPTIONAL }
```

```
DistributionPointName ::= CHOICE {  
    fullName                [0]      GeneralNames,  
    nameRelativeToCRLIssuer [1]      RelativeDistinguishedName }
```