Fundamentos de Segurança Informática (FSI)

2021/2022 - LEIC

Manuel Barbosa
mbb@fc.up.pt

Aula 18
Public-Key Infrastructure (PKI)

Porquê PKI

- A criptografia de chave pública pressupõe chaves públicas autênticas
- Caso contrário: ataques Man-in-the-Middle
- No mundo real isto pode ser resolvido de forma ad-hoc
  - confirmar manualmente que a chave pública pertence à entidade correta
  - utilizar sistemas de autenticação de chaves públicas como PGP/GPG

Porquê PKI

- Quando é necessária cobertura legal/regulamentar ⇒ PKI:
  - Normas técnicas: que algoritmos/formatos utilizar
  - Regulamentação: como devem ser utilizadas as normas técnicas
  - Mais regulamentação: responsabilidades/direitos dos participantes
  - Leis: garantias formais e penalizações em caso de violação das regras

Certificados de Chave Pública

Certificados de Chave Pública

- Objetivo:
  - Alice envia a Bob uma chave pública pk através de canal inseguro
  - Bob tem de obter garantia de que Alice detém a chave secreta correspondente
- Solução trivial:
  - Bob tem canal autenticado com Trusted-Third-Party (TTP)
  - Alice demonstrou anteriormente a TTP que é dona de pk (como?)
  - Bob pergunta a TTP (on-line) se pk pertence a Alice

Certificados de Chave Pública

- Problemas na prática:

Tecnologia: Certificados

1. Como se constrói o canal entre o Bob e a TTP?
2. E se a TTP estiver off-line?

PKI: Procedimentos/ Regulamentos

3. Como é que garantimos que o Bob e a Alice confiam na mesma TTP?
4. O que é que “Trust”/confiança na TTP significa?

Certificados de Chave Pública

- Os certificados de chave pública usam assinaturas digitais para resolver os pontos 1 e 2 no slide anterior:
  - TTP := Autoridade de Certificação/Certification Authority (CA)
  - Alice prova a CA que possui pk
    - Assinando um pedido de certificado que contém pk usando a chave secreta (PKCS#11)
    - Simplesmente porque CA fornece pk e chave secreta correspondente a Alice

Certificados de Chave Pública

- Os certificados de chave pública usam assinaturas digitais para resolver os pontos 1 e 2 no slide anterior:
  - CA fixa/verifica todos os dados relevantes para certificado:
    - Identidade de Alice + chave pública de Alice
    - Informação específica da CA: identidade e novo número de série
    - Validade (datas de início/fim)
  - CA assina documento eletrónico com esta informação

Certificados de Chave Pública

- Tecnicamente um certificado é:
  - documento codificado com regras ASN.1/DER
- O que é ASN.1?
  - Abstract Syntax Notation 1: independente de plataforma/linguagem
  - Legacy: linguagem de especificação herdada das normas para protocolos
  - Normas usam ASN.1 para especificar estruturas de dados (pacotes)
  - DER (Distinguished Encoding Rules) especificam codificação em bytes

```
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 MUST be v2 or v3
    subjectUniqueID [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                         -- If present, version MUST be v2 or v3
    extensions      [3]  EXPLICIT Extensions OPTIONAL
                         -- If present, version MUST be v3
}
```

Certificados de Chave Pública

- Como é que os certificados resolvem os aspetos 1 e 2:
  - Assinatura digital garante a Bob que informação veio de CA/TTP
  - CA pode estar off-line: Bob pode obter o certificado via Alice!
- Podemos assim transferir todos os certificados por canais inseguros?
- Outras perguntas importantes/relacionadas que iremos responder à frente:
  - Como é que Bob “conhece” a CA e verifica a sua assinatura?
  - Em que aspeto é a CA confiável para Bob e Alice?

Verificação de Certificados

- Suponhamos que Alice envia a Bob certificado com:
  - Identidade de Alice + Chave pública de Alice
  - Período de validade (datas de início/fim)
  - Meta-informação adicional
  - Tudo assinado por uma autoridade de certificação CA

Verificação de Certificados

- O Bob deve fazer o seguinte:

1. Verificar correção da identidade da Alice (e.g., DNS de um servidor)
2. Verificar que hora actual está no período de validade
3. Verificar meta-informação (específico para cada aplicação)

PKI: Procedimentos/ Regulamentos

4. Verificar que CA é de confiança
5. Obter chave pública da CA para verificar assinatura no certificado

- Já vimos: os primeiros 3 pontos e a verificação da assinatura são puramente tecnológicos.
- A PKI resolve os pontos sublinhados em 4 e 5.

Sanity check: entenderam como usar?

- Quem envia/recebe certificado de chave pública em:
  - Cifras assimétricas:
    - Chave pública pertence ao receptor
    - Emissor tem de obter certificado do receptor a priori

Sanity check: entenderam como usar?

- Quem envia/recebe certificado de chave pública em:
  - Assinaturas digitais
    - Chave pública pertence ao emissor (signatário)
    - Pode assinar e enviar certificado juntamente com (M, σ)

Sanity check: entenderam como usar?

- Quem envia/recebe certificado de chave pública em:
  - Acordo de chaves
    - Se for mutuamente autenticado, ambos têm de enviar certificados
    - O que acontece geralmente no TLS? (veremos numa aula futura)

Sanity check: entenderam como usar?

- Exemplo: no S/MIME (email assinado) os clientes de email geralmente
  - Permitem assinar um email assim que se instala certificado pessoal
  - Permitem cifrar email para Bob depois de receber email assinado de Bob
  - Faz sentido?

Ficha técnica de certificados de chave pública

- Normalizados no X.509 e transpostos para a internet pela IETF
- Tipos de dados/constantes importantes => identificadores de objeto (OI) standard fixos pela IETF
- Versão actual é a 3, que inclui campos standard:
  - subject => identidade do titular
  - issuer => identidade da CA emissora
  - validity => período de validade
  - public key info => chave pública do titular
  - serial => número de série (único no âmbito da CA)

Ficha técnica de certificados de chave pública

- Extensões (attachments), algumas das quais podem ser assinaladas como críticas
  - todas as extensões têm um identificador de objeto (OI)
  - se estiver marcada como crítica e não for conhecida ⇒ certificado inválido!
- Extensões importantes:
  - Subject/authority key identifier: hash da chave pública
  - Basic constraints: flag que assinala certificado como pertencendo a CA
  - Key usage: CA pode restringir utilização do certificado

Public Key Infrastructure

Public Key Infrastructure

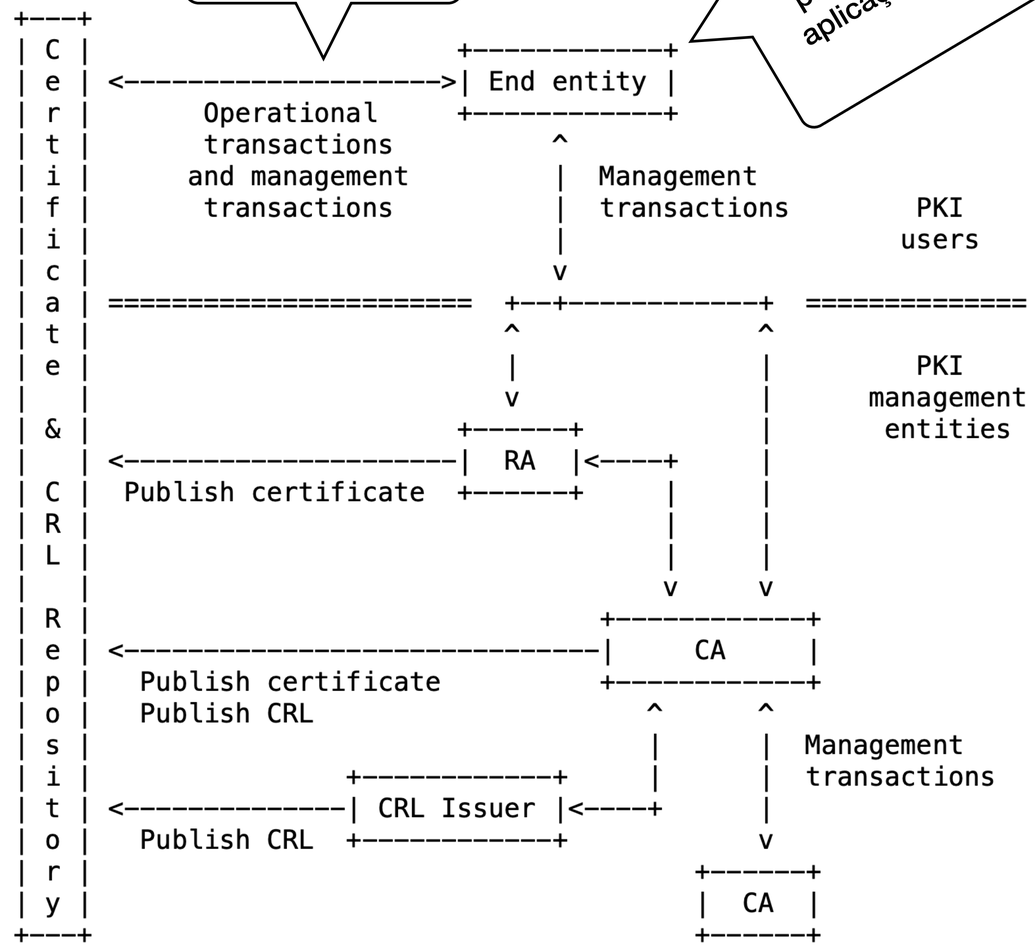
- Tudo o que é necessário para assegurar que:
  - O utilizador de um certificado (end entity) recebe garantia
  - Data por autoridade de certificação de confiança
  - De que uma chave pública pertence a outra entidade (pessoa, servidor)
  - E pode ser utilizada para um determinado fim
  - Com responsabilidades/obrigações bem definidas para todos

Arquitetura PKI

Transporte TLS, transporte S/MIME, LDAP, etc. Instalação de SO

Utilizadores comuns, servidores, produtores de aplicações, etc.

BD de certificados no SO, no browser, cliente de email, etc.



Transações operacionais/gestão

- Como é que os certificados de chave pública circulam/são distribuídos e armazenados?
- Vários protocolos especificam como isso pode ser feito:
  - podem ser armazenados em repositórios (e.g., LDAP), ou simplesmente nas aplicações/SOs
  - podem ser transferidos entre aplicações em protocolos específicos (HTTP, FTP, MIME)
  - têm de ser codificados em formatos que garantam interoperabilidade
- Já falamos de várias normas que desempenham este papel
  - No TLS o RFC especifica como é que se trocam certificados
  - No S/MIME os certificados são incluídos em attachments PKCS#7
  - Os sistemas operativos/browsers gerem certificados em componentes seguros dedicados

Funcionamento PKI

- Num slide anterior fizemos perguntas importantes:
  - Como é que um utilizador contacta/toma conhecimento de uma CA?
  - Como é que o Bob verifica a assinatura produzida por uma CA num certificado?
- Resposta:
  - Todas as chaves públicas são codificadas em certificados X.509
  - Alguns certificados especiais contêm chaves públicas de CAs
  - O Bob obtém a chave pública da CA num outro certificado
  - O Bob usa a chave pública no certificado da CA para verificar assinatura no certificado da Alice
  - Verificação OK $\Rightarrow$ Bob pode usar a chave pública da Alice

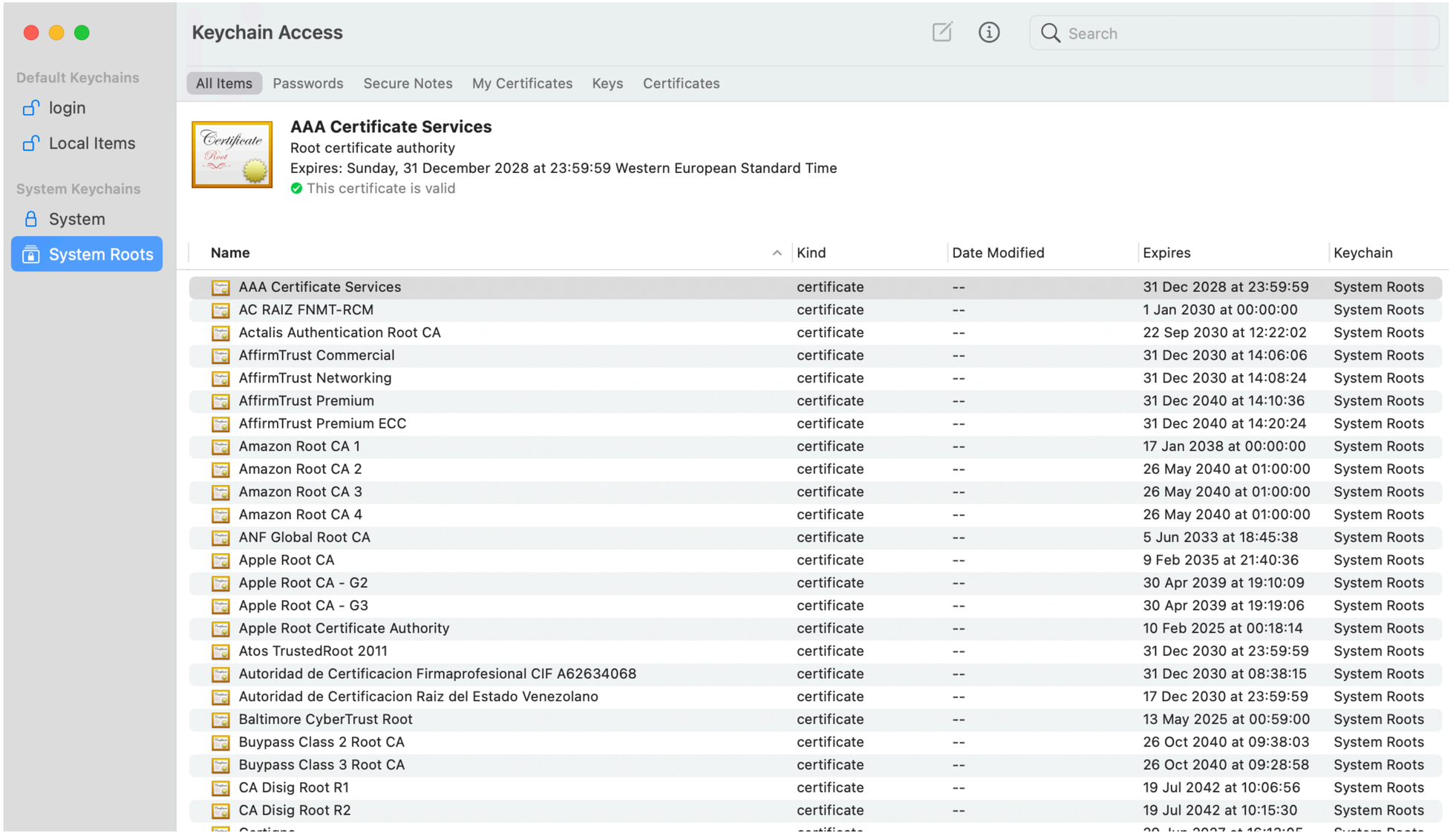
Funcionamento PKI

- Num slide anterior fizemos perguntas importantes:
  - Como é que um utilizador contacta/toma conhecimento de uma CA?
  - Como é que o Bob verifica a assinatura produzida por uma CA num certificado?
- Conclusão: a chave pública da Alice é autêntica se:
  - O Bob conhece de antemão o certificado da CA que emitiu o certificado da Alice
  - Bob confia nessa CA para ter verificado que, de facto, a Alice é titular daquela chave pública (possui a chave privada correspondente)

Gestão PKI: Inicialização

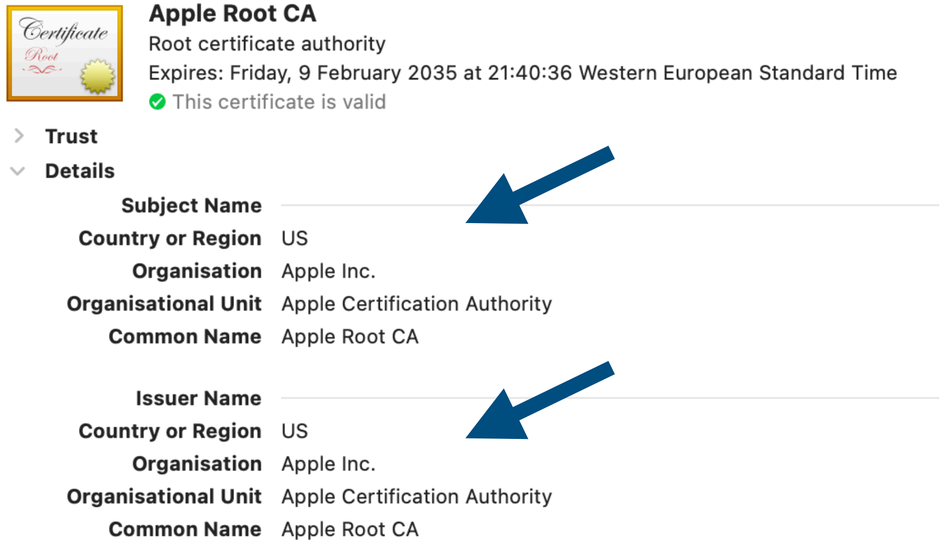
- Pergunta: como é que o Bob sabe se pode confiar na CA?
- Nos cenários mais simples:
  - O Bob obtém o certificado diretamente da CA via canal seguro
  - O Bob confia na CA implicitamente (e.g., porque a comunidade confia)
- Exemplos:
  - Os sistemas operativos trazem um grande número de certificados pré-instalados
  - O Cartão de Cidadão Português contém certificados de autoridades geridas pelo Estado
- Isto são exemplos da operação de inicialização da relação entre um utilizador e uma PKI

Gestão PKI: Inicialização



Cadeias de Certificação

- Vimos um caso simples: Bob confia na CA usada pela Alice de forma implícita
- Em geral isto não é assim tão simples:
  - O Bob é inicializado com certificados de raiz (root CAs)
  - O Bob confia implicitamente nestas CAs
  - Os certificados de raiz são auto-assinados (self-signed):
    - A CA gera um par de chaves $(\mathsf{sk}, \mathsf{pk})$
    - A CA cria o seu próprio certificado com subject = issuer = CA name
    - O certificado inclui pk e a CA assina-o com sk



Cadeias de Certificação

- Importante: qualquer pessoa pode gerar um certificado auto-assinado:
  - a dizer que é uma CA específica!
- Validar um certificado de raiz/auto-assinado implica:
  - acreditar que a chave secreta associada pertence, de facto, à CA
  - acreditar que essa CA é de confiança => só produz certificados bons
    - vimos nos slides anteriores o que é suposto fazer uma CA

Cadeias de Certificação

- Geralmente as Root CAs não emitem diretamente certificados para utilizadores finais
  - Existe uma hierarquia de CAs
    - Se a CA com nome **A** assina o certificado da CA com nome **B**
    - Então confiança em **B** $\leq$ confiança em **A**

Cadeias de Certificação

- Podemos ter muitos níveis nesta hierarquia/árvore:
  - Para autenticar chave da Alice, Bob obtém certificado da Alice
  - Para validar certificado da Alice, Bob obtém certificado da CA da Alice
  - Bob verifica que certificado da Alice é válido com base no certificado da CA
- Bob ainda precisa de decidir se confia na chave da CA da Alice:
  - pode precisar de uma sequência de certificados de CAs
- Confiança = validação continua até encontrar Root CA em quem Bob confia

Gestão PKI: Registo

- Autoridades de Registo/Registration Authorities (RA):
  - Front-end: contacto direto com os utilizadores/titulares
  - Responsáveis por verificar os dados colocados nos certificados
  - Responsáveis por verificar posse de chave privada correspondente à chave pública presente no certificado

Gestão PKI: Certificação

- Autoridades de Certificação/Certification Authorities (CA):
  - Back-end: infra-estrutura que produz chaves e certificados
  - Tipicamente high-security: air gaps, segurança física, etc.

Gestão PKI: Exemplo

- Cartão de Cidadão
  - Autoridade de Registo: Registo Civil, Loja do Cidadão, etc.
  - Autoridade de Certificação: em instalações próprias na INCM
  - A CA gera chaves, assina certificados, produz cartões
  - A RA entrega cartões/chaves aos titulares depois de verificação de identidade

Gestão PKI: Revogação

- Os certificados são sempre inválidos quando fora do período de validade assinado
- Mas como se pode invalidar um certificado válido?
  - E.g., perda de chave secreta, CA corrompida, meta-dados deixam de ser válidos
- É preciso revogar os certificados enquanto eles ainda são válidos

Gestão PKI: Revogação

- Isto é feito pelas CAs usando Certificate Revokation Lists (CRL):
  - CA publica periodicamente black-list de certificados revogados
  - Os consumidores de certificados devem publicar CRL mais recente
  - CRLs podem ser publicadas excepcionalmente, mas apenas *best-effort*
- Como é que sabemos onde está a CRL mais recente?
- Os próprios certificados geralmente contém extensão com URLs de CRL
- Tradicionalmente pouco suporte por parte das aplicações

Gestão PKI: Revogação

- No mundo real há três soluções para este problema
  - Trusted Service Provider Lists (TSL):
    - white-list actualizada de certificados
    - usada em comunidades pequenas/fechadas (e.g., banking) e high-security
  - On-line Certificate Status Protocol (OCSP)):
    - servidor seguro (geralmente gerido pela própria CA) verifica estado de revogação
    - usado típicamente em contextos organizacionais (e.g., eGov)
  - Certificate pinning:
    - web servers/browsers/aplicações gerem white-lists próprias
    - permitem identificar certificados mais importantes para entidades críticas (e.g., Google)

Políticas de Certificação

- A PKI pode ser utilizada para dar valor legal a assinaturas digitais => assinaturas qualificadas
- Uma política de certificação (Certificate Policy) é um conjunto de regras de operação:
  - Direitos e responsabilidades para titulares e utilizadores
  - Direitos e responsabilidades para as CAs
- Estes direitos e responsabilidades podem ser estipulados legalmente
  - (https://dre.pt/home/-/dre/156848060/details/maximized)

Políticas de Certificação

- Uma política de certificação recebe um object identifier (OID):
  - Os certificados podem incluir este OID
- Por lei, apenas pode usar este OID uma autoridade que:
  - Foi sujeita a auditoria e credenciada para o efeito