

# Criptografia Aplicada

## Módulo II – Certificação e Public Key Infrastructure

M. B. Barbosa

mbb@di.uminho.pt

Departamento de Informática  
Universidade do Minho

2006/2007

# Introdução

- A Abstract Syntax Notation One (ASN.1) encontra-se especificada nas normas X.208.
- Embora tenha sido criada para especificar tipos e valores de dados nas normas OSI, é hoje utilizada sistematicamente para especificar objectos nos mais variados standards, incluindo os criptográficos.
- A ASN.1 é uma notação que permite a definição de diversos tipos de dados, desde os tipos básicos, como inteiros aos tipos estruturados, como sequências.
- A ASN.1 por si só não pode ser utilizada directamente numa implementação, sendo necessárias normas adicionais para definir como é que se codifica essa notação abstracta em sequências de bits.

# Codificação ASN.1

- As formas mais usuais de codificação são o BER e o DER.
- Basic Encoding Rules (BER) – mecanismo de codificação definido no standard X.209 e que, por permitir obter várias codificações para o mesmo valor, não é conveniente quando é necessária uma codificação sem ambiguidades.
- Distinguished Encoding Rules (DER) – subconjunto do BER definido no standard X.509 e que, introduzindo restrições adicionais à codificação, garante uma codificação única para cada valor ASN.1.
- O DER é geralmente utilizado quando se pretende garantir a compatibilidade da codificação ASN.1 com implementações heterogéneas.

## Considerações gerais

- Uma definição em ASN.1 pode ser de várias categorias, no que respeita à abrangência do seu contexto:
  - Universal,
  - Application (e.g. X.500),
  - Private (e.g. dentro de uma empresa) e
  - Context-specific (dentro de uma estrutura).
- O layout não é relevante para a semântica do ASN.1 i.e. conjuntos de espaços e linhas em branco são irrelevantes. Os comentários são delimitados por `--`.
- Todos os tipos e valores podem ter um nome definido com o operador de atribuição `::=`. Estes nomes podem ser utilizados na definição de novos tipos e valores. Nomes de tipos com letras maiúsculas. Nomes de valores com letras minúsculas

# Tipos simples

- O ASN.1 define os seguintes tipos simples, que podem ser utilizados na construção de tipos mais complexos:

Simple Type	Tag	Typical Use
BOOLEAN	1	Two-state variable values
INTEGER	2	Integer variable values
BIT STRING	3	Binary data of arbitrary length
OCTET STRING	4	Binary data where $L = k * 8$
NULL	5	Absence of a sequence element
OBJECT IDENTIFIER	6	Name information objects
REAL	9	Real variable values
ENUMERATED	10	Variables with at least three states
CHARACTER STRING	*	Strings of characters

# Tipos estruturados

- O ASN.1 define as seguintes formas de construir tipos complexos:

Structured Type	Tag	Typical Use
SEQUENCE	16	Ordered collection of different type
SEQUENCE OF	16	Ordered collection of the same type
SET	17	Unordered collection of different types
SET OF	17	Unordered collection of the same type
CHOICE	*	Collection from which to choose one type
SELECTION	*	Select a component type from a CHOICE

- Além disso, estão também definidos alguns tipos estruturados úteis: `GeneralizedTime`, `UTCTime`, `ObjectDescriptor`, **etc.**

# ASN.1: Exemplo #1

- Nas normas PKCS #1 define-se o tipo de uma chave privada RSA como o seguinte objecto:

```
RSAPrivateKey ::= SEQUENCE {  
    version Version,  
    modulus INTEGER, -- n  
    publicExponent INTEGER, -- e  
    privateExponent INTEGER, -- d  
    prime1 INTEGER, -- p  
    prime2 INTEGER, -- q  
    exponent1 INTEGER, -- d mod (p-1)  
    exponent2 INTEGER, -- d mod (q-1)  
    coefficient INTEGER, -- (inverse of q) mod p }  
Version ::= INTEGER
```

## ASN.1: Exemplo #2

- Nas mesmas normas também está definido o hash que deve ser utilizado quando se usa o RSA como mecanismo de assinaturas. Na notação ASN.1 temos:

```
DigestInfo ::= SEQUENCE {  
    digestAlgorithm DigestAlgorithmIdentifier,  
    digest Digest }  
digestAlgorithmIdentifier ::= AlgorithmIdentifier  
AlgorithmIdentifier ::= SEQUENCE {  
    algorithm OBJECT IDENTIFIER,  
    parameters ANY DEFINED BY algorithm OPTIONAL }  
Digest ::= OCTET STRING
```



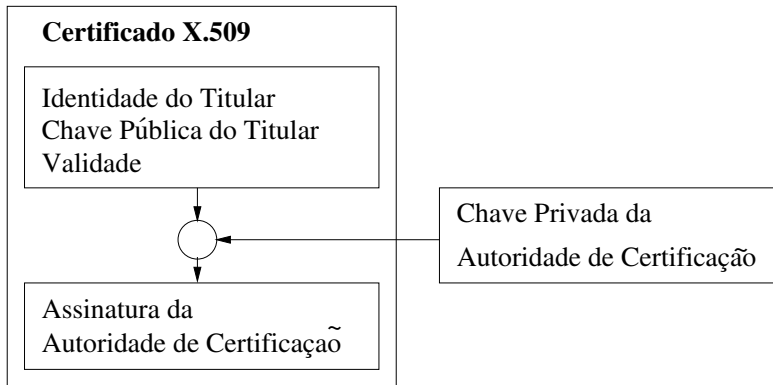
# Introdução

- Esforços recentes para melhorar a segurança na Internet levaram ao aparecimento de um grupo de protocolos (S/MIME, IPsec, etc.) que utilizam a criptografia de chave pública para garantir:
  - Confidencialidade
  - Integridade
  - Autenticação
  - Não repúdio.
- Esta utilização baseia-se no conceito de Certificado (de chave pública ou de atributos).
- A Public Key Infrastructure (PKI) tem como objectivo a gestão segura e eficiente de chaves e certificados para permitir essa mesma utilização.

- A utilização da criptografia de chave pública nas telecomunicações é regulamentada pela recomendação X.509 da International Telecommunications Union (ITU).
- A aplicação dessa recomendação à Internet está definida num conjunto de Requests For Comments (RFCs) publicados pela IETF.
- A Internet Engineering Task Force é uma comunidade internacional de produtores, operadores, vendedores e investigadores das tecnologias de redes, interessados no funcionamento e evolução da Internet.
- Dentro da IETF, o grupo que gere os RFCs relacionados com o X.509 chama-se “*PKIX Working Group*”. Este grupo de trabalho mantém um conjunto de documentos que se denomina “*Internet X.509 Public Key Infrastructure*”.

# Certificados de Chave Pública

- Os utilizadores numa PKI devem poder confiar que, cada vez que utilizam uma chave pública, o agente com quem querem comunicar possui a chave privada associada.
- Esta confiança é construída com base em Certificados de Chave Pública.
- Um Certificado de Chave Pública é uma estrutura de dados que associa uma chave pública a um determinado agente (a uma representação da sua identidade).
- A associação chave/agente é estabelecida por uma entidade terceira, uma Autoridade de Certificação, que assina digitalmente cada certificado.
- A utilidade de um certificado depende unicamente da **confiança** depositada na Autoridade de Certificação.



- O utilizador do certificado confia que a Autoridade de Certificação verificou que a chave pública contida no certificado pertence de facto ao titular do certificado.
- A assinatura da Autoridade de Certificação assegura a autenticidade e integridade do certificado.
- Um Certificado de Chave Pública é válido durante um período de tempo bem definido. Esse período vem especificado no conteúdo assinado.
- Como a assinatura e a validade temporal de um certificado podem ser verificados independentemente por um utilizador, os certificados podem ser distribuídos por canais inseguros. Será isto verdade para todos os certificados?

- Os Certificados de Chave Pública são utilizados maioritariamente na validação de informação assinada digitalmente. Este processo consiste geralmente nos seguintes passos:
  - 1 O destinatário verifica que a identidade indicada pelo emissor está de acordo com a identidade indicada no certificado.
  - 2 O destinatário verifica que o certificado é **válido**:
    - que a assinatura do certificado é válida;
    - que foi efectuada por uma autoridade de certificação de confiança;
    - que o certificado está dentro do seu período de validade.
  - 3 O destinatário verifica que a informação que recebe está de acordo com as permissões/privilégios do emissor.
  - 4 O destinatário utiliza a chave pública contida no certificado para verificar a assinatura da informação recebida.

- Os Certificados de Chave Pública são utilizados maioritariamente na validação de informação assinada digitalmente. Este processo consiste geralmente nos seguintes passos:
  - 1 O destinatário verifica que a identidade indicada pelo emissor está de acordo com a identidade indicada no certificado.
  - 2 O destinatário verifica que o certificado é **válido**:
    - que a assinatura do certificado é válida;
    - que foi efectuada por uma autoridade de certificação de confiança;
    - que o certificado está dentro do seu período de validade.
  - 3 O destinatário verifica que a informação que recebe está de acordo com as permissões/privilégios do emissor.
  - 4 O destinatário utiliza a chave pública contida no certificado para verificar a assinatura da informação recebida.

- Se todos os passos anteriores forem executados sem problemas, o destinatário aceita que a informação foi assinada pelo emissor, e que essa informação permanece inalterada.
- Como é que se utiliza um certificado para proteger informação ao nível da confidencialidade?
- O certificado passa a ser utilizado pelo emissor, e contém informação relativa ao destinatário:
  - 1 o emissor valida o certificado e a identidade do destinatário;
  - 2 o emissor utiliza a chave pública contida no certificado para cifrar a informação;
  - 3 o emissor envia a informação cifrada ao destinatário que a decifra com a sua chave privada.



- Se todos os passos anteriores forem executados sem problemas, o destinatário aceita que a informação foi assinada pelo emissor, e que essa informação permanece inalterada.
- Como é que se utiliza um certificado para proteger informação ao nível da confidencialidade?
- O certificado passa a ser utilizado pelo emissor, e contém informação relativa ao destinatário:
  - 1 o emissor valida o certificado e a identidade do destinatário;
  - 2 o emissor utiliza a chave pública contida no certificado para cifrar a informação;
  - 3 o emissor envia a informação cifrada ao destinatário que a decifra com a sua chave privada.

# Certificados X.509 (V1)

- Surgiram em 1988 com a primeira versão do X.509. Um certificado deste tipo contém os seguintes campos:
  - **Version** version
  - **CertificateSerialNumber** serialNumber
  - **AlgorithmIdentifier** signature
  - **Name** issuer
  - **Validity** validity
  - **Name** subject
  - **SubjectPublicKeyInfo** subjectPublicKeyInfo
- A assinatura do certificado é efectuada pela Autoridade de Certificação (CA) sobre uma codificação DER da representação desta estrutura de dados em ASN.1.

## Certificados X.509 (V2)

- Surgiram na revisão de 1993 do X.509. Não chegaram a ser muito utilizados porque pouco tempo depois surgiu a versão actual (V3).
- Foram introduzidos dois novos campos:
  - **UniqueIdentifier** issuerUniqueID
  - **UniqueIdentifier** subjectUniqueID
- Estes campos tentam rectificar o problema de ser muito difícil garantir que os campos do tipo **Name** tenham valores únicos.
- A IETF recomenda que as CAs não utilizem estes campos e garantam, na medida do possível, a unicidade dos nomes. Por outro lado recomenda que estes campos, caso existam, não devem ser ignorados.

# Certificados X.509 (V3)

- Esta é a versão utilizada hoje em dia. Foi standardizada em 1996 e é compatível com as versões anteriores.
- Esta versão veio colmatar as deficiências que as versões anteriores apresentavam para algumas aplicações e que consistiam basicamente na necessidade de mais atributos.
- Como inovação, esta versão introduziu um novo campo do tipo **Extensions**: uma colecção de elementos do tipo **Extension**.
- As extensões permitem associar atributos genéricos a um agente ou à sua chave pública, de forma flexível.
- Cada extensão é ela própria uma estrutura de dados com um identificador e um valor adequado ao tipo do atributo que representa.

## Certificados X.509 (V3): Atributos

- **version** Tem de estar de acordo com o conteúdo do certificado.
- **serialNumber** Número único atribuído pela CA.
- **signature** Estrutura que identifica o algoritmo utilizado para gerar a assinatura da CA que acompanha o certificado.
- **validity** Estrutura com as duas datas que delimitam o período de validade do certificado.
- **subjectPublicKeyInfo** Estrutura contendo a chave pública do titular do certificado e identificação do algoritmo correspondente.

- Os atributos **issuer** e **subject** identificam a CA e o titular do certificado respectivamente. Ambos são do tipo **Name**.
- O tipo **Name** provém da norma X.501 e é utilizado porque permite a compatibilidade com os sistemas de directório definidos nas normas X.500 (e.g. DAP e LDAP).
- O tipo **Name** é uma colecção de atributos, geralmente *strings* da forma “< nome > = < valor >”. Estes atributos definem um **Distinguished Name** para o agente titular. Na realidade o tipo name é uma SEQUENCE de RelativeDistinguishedNames
- O **Distinguished Name** tem uma estrutura hierárquica. Inclui por exemplo, o país, a organização e o nome próprio do agente ou entidade.

- A norma X.520 standardiza alguns dos componentes de um Distinguished Name. Os seguintes são de reconhecimento obrigatório e são muito utilizados:
  - country (C)
  - organization (O)
  - organizational-unit (OU)
  - common name (CN)
  - serial number (SN)
- Algumas aplicações importantes utilizam também o endereço de e-mail como um dos atributos centrais da construção do Distinguished Name.
- Exemplo: C=PT, O=UMINHO, OU=DI, CN=MBB

## Certificados X.509 (V3): Extensões

- As extensões são marcadas como **Critical** ou **Non Critical**.
- Uma aplicação que encontre uma extensão crítica que não reconheça tem de rejeitar o certificado.
- Não são permitidas várias instâncias da mesma extensão.
- O RFC3280 da IETF normaliza as extensões recomendadas para utilização na Internet, definindo o identificador (OBJECT IDENTIFIER) e o tipo de dados associado.
- São desaconselhados desvios desta recomendação, nomeadamente no que diz respeito a extensões críticas, apesar de não haver qualquer limitação a nível do standard.



- **Subject Key Identifier** Serve para identificar o certificado que contém uma determinada chave pública e.g. quando um agente tem várias. É, em geral, um valor de hash derivado da chave pública que, caso esta extensão não conste do certificado, pode ser calculado em run-time.
- **Authority Key Identifier** Serve para identificar a chave pública da CA que assinou o certificado, caso existam várias, o que facilita a verificação de cadeias de certificação. Isto pode ser feito
  - identificando o certificado da CA através do seu **Issuer e Serial Number**;
  - ou através da extensão **Subject Key Identifier** do certificado da CA.
- **Subject/Issuer Alternative Name** Permitem associar formas de identificação alternativas ao titular do certificado ou à CA que o emitiu (e-mail, DNS, endereço IP, URI, etc.).

- **Basic Constraints** Permite assinalar um certificado como pertencendo a uma CA e limitar o comprimento de cadeias de certificados.
- **Certificate Policies** Permite incluir informação relativa às políticas 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 CAs, permite definir as políticas de certificação aplicáveis por CAs hierarquicamente inferiores.
- **Policy Mappings** Permite a uma CA declarar que algumas das suas políticas são equivalentes às políticas de certificação de outra CA.

Por exemplo, duas companhias certificam-se mutuamente: o que acontece às políticas de segurança que foram definidas em contextos diferentes. (As hierarquias de CAs

- **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** Protecção da confidencialidade de chaves.
  - **dataEncipherment** Protecção da confidencialidade de dados.
  - **keyAgreement** Protocolos de acordo de chaves.
  - **keyCertSign** Assinatura de certificados.
  - **cRLSign** Assinatura de CRLs.
  - **encipherOnly/decipherOnly** Restringem a funcionalidade **keyAgreement**.

- **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 ao utilizador de um certificado onde pode obter informação quanto à revogação do certificado na forma de **Certificate Revocation Lists** (CRLs).

## Certificados X.509 (V3): Codificação

- Os certificados, como todas as estruturas de dados na PKI, são definidos e representados em ASN.1. A codificação é feita utilizando as Distinguished Encoding Rules (DER).
- O ficheiro que contém um certificado X.509 consiste na codificação DER da seguinte estrutura ASN.1:

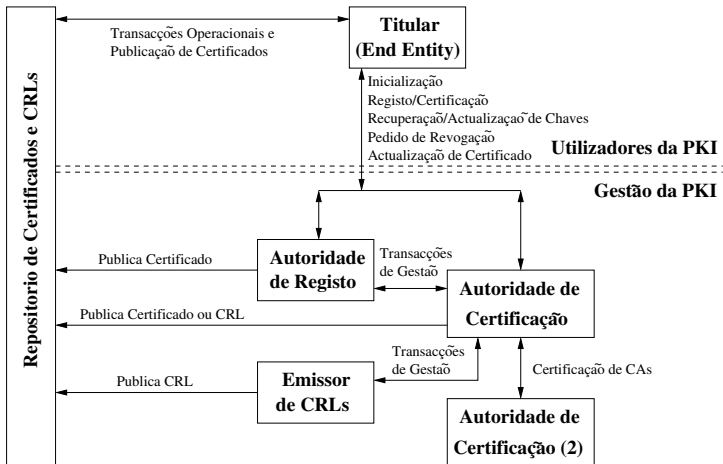
```
Certificate ::= SEQUENCE {  
    tbsCertificate      TBSCertificate,  
    signatureAlgorithm  AlgorithmIdentifier,  
    signatureValue      BIT STRING }
```

- Além da estrutura de atributos apresentada nos slides anteriores (**tbsCertificate**), aparece também a assinatura da Autoridade de Certificação (**signatureAlgorithm** e **signatureValue**). O campo **signatureAlgorithm** é uma repetição do campo **signature** da estrutura **tbsCertificate**.

# Introdução

- Uma Public Key Infrastructure define-se como o conjunto de hardware, software, pessoas, políticas e procedimentos necessários para criar, gerir, armazenar, distribuir e revogar Certificados de Chave Pública.
- Uma PKI é composta por cinco tipos de componentes:
  - **Titulares de Certificados** Possuem as chaves privadas e as utilizam para decifrar mensagens e assinar documentos.
  - **Clientes** Utilizam a chave pública contida num certificado para cifrar mensagens e verificar assinaturas.
  - **Autoridades de Certificação** Emitem e revogam certificados.
  - **Autoridades de Registo** Garantem a associação entre chaves públicas e identidades de titulares (são opcionais).
  - **Repositórios** Armazenam e disponibilizam certificados e CRLs.

# Arquitectura



- 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 efectuadas de diversas formas, incluindo o LDAP, HTTP e FTP. Para todos estes meios estão especificados protocolos operacionais que definem, inclusivamente, os formatos das mensagens.
  - **Protocolos de Gestão** Estes protocolos são necessários para dar suporte às interacções entre os utilizadores e as entidades de gestão da PKI, nomeadamente:
    - Inicialização.
    - Registo e Certificação.
    - Recuperação e Actualização de pares de chaves.
    - Pedido de revogação.
    - Certificação de CAs.



# PKI: Operações

- **Inicialização** Processo inicial que permite ao utilizador comunicar com a PKI: toma conhecimento das CAs em que confia e adquire as chaves públicas e certificados correspondentes, gera o seu par de chaves, etc.
- **Registo** Um utilizador dá-se a conhecer a uma CA (directamente, ou através de uma RA) para que a CA lhe possa emitir um certificado; para isso fornece informação de identificação que deve ser verificada pela CA (RA).
- **Geração de Par de Chaves** Nalgumas implementações, as CAs encarregam-se de gerar o par de chaves.
- **Certificação** A CA recebe a chave pública do utilizador e a sua identificação e emite o respectivo certificado, segundo regras internas.

- **Publicação de Certificados e CRLs** Esta tarefa pode ser feita directamente pela CA, ou indirectamente por entidades como RAs. Além de colocar os certificados e CRLs em repositórios é muitas vezes necessário fazer estes documentos chegar aos utilizadores finais por outros meios (on-line ou não).
- **Revogação** Quando um certificado é emitido o seu período util 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. um despedimento, o comprometimento da chave privada, etc.). A revogação de certificados faz-se através de CRLs. As CRLs vão ser analisadas em detalhe mais tarde.

- **Recuperação de um Par de Chaves** Nalgumas implementações as CAs armazenam o par de chaves da entidade como *back-up* e protecção e.g. no caso de uma empresa e os seus empregados. Nestes casos o par de chaves pode ser restaurado em caso de extravio ou danificação do seu suporte.
- **Actualização de Par de Chaves** Todos os pares de chaves precisam de ser alterados, periodicamente por razões de segurança, ou simplesmente porque a segurança da chave privada foi corrompida.
- **Certificação de CAs** Os certificados das CAs chamam-se **cross certificates**. São utilizados para a validação de cadeias de certificados, mas também podem ser utilizados para outros fins e.g. comunicação segura entre uma entidade e a CA.

# Cadeias de Certificação e 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 Autoridade de Certificação que o emitiu e, conseqüentemente, a obtenção e validação do certificado que a contém.
- A validação do certificado da CA poderá implicar o conhecimento da Chave Pública de outra CA que o tenha emitido, e assim sucessivamente.
- Chama-se a esta sequência uma **Cadeia de Certificação**.
- Confiança numa chave pública implica validar o certificado dessa chave, e zero ou mais certificados de CAs.

- A validação de um certificado segue o seguinte algoritmo:
  - Para todo o  $x = 1, \dots, n - 1$ , o `subject` do certificado  $x$  é o emissor do certificado  $x + 1$ ;
  - O certificado 1 é emitido pela **raiz da relação de confiança**;
  - O certificado  $n$  é o certificado a ser validado; e ...
  - Para todo o  $x = 1, \dots, n$ , verifica-se que o certificado é válido na altura da sua utilização.
- Perguntas:
  - Onde termina a validação de uma cadeia de certificados?
  - O que é a raiz da relação de confiança?
  - Se cada chave pública implica um certificado, e vice-versa, o que aparece primeiro?

- As cadeias de certificação reflectem uma **hierarquia** de Autoridades de Certificação.
- As CAs hierarquicamente superiores emitem os certificados das CAs hierarquicamente inferiores.
- No topo da hierarquia reside uma CA denominada **Root** ou raiz. O certificado desta CA é emitido e assinado por ela própria: os campos `subject` e `issuer` do seu certificado são iguais.
- A confiança na chave pública de uma Root CA **não depende** de outra CA. é estabelecida por um meio externo à PKI.
- Por exemplo, a utilização de uma instalação comum do MS Windows implica a “confiança” em dezenas de Root CAs!

- Um utilizador conhece um número limitado de chaves públicas pertencentes a CAs (em geral Root CAs) e que funcionam como raízes das relações de confiança.
- Isso significa que o utilizador aceitará um certificado emitido por uma dessas CAs e que depositará um determinado nível de confiança no seu conteúdo.
- A validação de uma cadeia de certificados terminará quando for encontrado um certificado com essa característica.
- Conclusão: o grau de confiança depositada num certificado validado baseia-se apenas na confiança depositada na CA que funcionou como raiz da relação de confiança.

# Políticas de Certificação

- A confiança que é depositada numa CA depende, em última instância, da sua política de certificação, e da forma como essa política é implementada.
- Essa confiança é influenciada por diversos factores internos e externos à PKI. Factores externos, como a credibilidade da instituição ou empresa que suporta a CA e o seu país de origem são obviamente importantes.
- No entanto, o conceito de PKI prevê uma forma de "ancorar" a confiança que se deposita numa CA, naquilo que de facto importa: as leis da sociedade em que a PKI opera.
- Isto é feito através das **Certificate Policies** ou CP.



# Certificate Policies

- Uma CP é um conjunto de regras que define a aplicabilidade de certificados a uma determinada comunidade ou classe de aplicações:
  - A legislação em que se baseará a emissão e utilização dos certificados.
  - Os requisitos e as responsabilidades (nomeadamente legais e financeiras) associados a CAs e RAs.
  - Os requisitos e as responsabilidades associados a Titulares e Clientes.
  - Restrições ao conteúdo e utilização dos certificados e.g. somas máximas envolvidas numa transacção, etc.
  - Procedimentos a serem implementados relativamente a diversos aspectos do funcionamento de CAs e RAs.
- Cada CP é identificada por um `Object Identifier` que pode ser incluído na extensão **Certificate Policies**.

# Certification Practice Statements

- Cada CA publica uma ou mais **Certification Practice Statements** (CPS), nas quais publicita as suas normas de operação internas. Uma CPS explica a forma como uma CA implementa um determinado conjunto de **CPs**.
- A acreditação de uma CA de acordo com uma determinada CPS implica uma auditoria efectuada 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 protecção de dados. Uma CA que pretenda emitir certificados que em conformidade com estas políticas tem de ser credenciada pelo estado Canadiano.

# Políticas de Certificação na Prática

- Uma parte significativa do RFC3280 é dedicada às políticas de certificação e ao efeito de uma política de certificação imposta num determinado ponto da hierarquia.
- 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 CA, 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 informação contida nos certificados inclui uma componente processável num algoritmo de resolução deste problema, algoritmo esse que está também definido no RFC3280.
- 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.
- Compete ao utilizador determinar se a política associada a um determinado certificado é aceitável ou não.
- É também possível incluir num certificado uma CPS, de forma directa ou referenciada, bem como informação dirigida ao utilizador final sobre as condições de emissão do certificado.

# Introdução

- 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 CA 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, em geral, pelas próprias CAs. É possível que a CA delegue esta função numa outra autoridade denominada **CRL Issuer**. Uma CRL emitida nestas condições denomina-se **CRL Indirecta**.
- Cada CRL tem um contexto específico (o conjunto de certificados passíveis de aparecerem no seu conteúdo), que deve estar bem definido.

# Estrutura e Atributos

- Uma CRL contém os seguintes campos:
  - **Version** version
  - **AlgorithmIdentifier** signature
  - **Name** issuer
  - **Time** thisUpdate
  - **Time** nextUpdate
  - Lista de certificados revogados.
- A assinatura da CRL é efectuada pela Autoridade de Certificação (CA) sobre uma codificação DER da representação desta estrutura de dados em ASN.1.
- O campo **version** é opcional e deve ter o valor 2 uma vez que as CRLs foram introduzidas na versão 2 do X.509. O campo **signature** tem o mesmo significado que nos certificados.

- O campo **thisUpdate** indica a data de publicação da CRL.
- O campo **nextUpdate** é muito importante uma vez que permite ao utilizador conhecer uma data a partir da qual é garantido ter sido publicada uma nova CRL.
- A lista de certificados revogados é representada por uma sequência de registos que indicam o titular e número de série de cada certificado, bem como a data em que foi revogado.
- A definição de CRL contempla também a inclusão de extensões, tanto dos atributos globais da CRL, como dos atributos associados a cada certificado revogado.

## Extensões Globais

- **Authority Key Identifier** Semelhante à utilizada nos certificados, permite identificar o certificado que contém a chave pública da CA que permite validar a CRL.
- **Issuer Alternative Name** Semelhante à utilizada nos certificados, permite indicar identificações alternativas para a CA.
- **CRL Number** Permite incluir um número sequencial que serve como número de ordem da CRL dentro do seu contexto específico.
- O CRL Number serve também para as Delta CRLs referenciarem as Base CRLs correspondentes.



- **Delta CRL Indicator** Marca a CRL como sendo uma Delta CRL, e referencia a Base CRL correspondente.
- **Issuing Distribution Point** Identifica o ponto de distribuição da CRL, bem como o seu contexto e.g. o tipo de certificados que revoga, as possíveis razões de revogação, etc.
- **Delta CRL Distribution Point** Permite, nas Base CRLs, indicar onde pode ser obtida a Delta CRL correspondente mais recente.
- Os pontos de distribuição são indicados como URIs, e apontam sempre para a CRL mais recente, num dado contexto.

## Extensões às Entradas

- **Reason Code** Permite indicar uma razão para o aparecimento do certificado na CRL.
- Em geral, a razão será a revogação de um certificado por um determinado motivo e.g. chave privada comprometida, segurança da CA comprometida, alteração de filiação ou de privilégios, existência de uma versão mais recente do certificado, fim da operação da CA.
- No entanto, é possível um certificado não ser imediatamente revogado, mas ser suspenso. O valor `certificateHold` está definido para este efeito.
- Um certificado suspenso numa CRL pode ser reactivado numa CRL posterior atribuindo o valor `removeFromCRL` a esta extensão.

- **Hold Instruction Code** Quando um certificado é suspenso, é possível indicar a acção que deve ser tomada quando a utilização do certificado for necessária: rejeitar o certificado ou contactar a CA.
- **Invalidity Date** Permite indicar a data em que se suspeita que o certificado deixou de ser válido.
- **Certificate Issuer** Nas CRLs Indirectas utiliza-se esta extensão para identificar a CA que emitiu o certificado revogado.

## Publicação periódica e segurança

- A segurança de uma PKI 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, um utilizador nunca saberia qual a CRL mais recente. Isto possibilitaria ataques do tipo:
  - Um intruso controla o meio de comunicação que liga o utilizador ao ponto de publicação de uma CRL.
  - O utilizador pretende utilizar um certificado cuja chave privada foi comprometida, e que é conhecida pelo intruso.
  - O utilizador tenta obter a CRL mais recente, que revogaria o certificado. O intruso fornece uma versão antiga da CRL.
  - O utilizador aceita o certificado porque não tem como saber que a CRL que utilizou estava desactualizada.

## Publicação periódica e segurança

- A segurança de uma PKI 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, um utilizador nunca saberia qual a CRL mais recente. Isto possibilitaria ataques do tipo:
  - Um intruso controla o meio de comunicação que liga o utilizador ao ponto de publicação de uma CRL.
  - O utilizador pretende utilizar um certificado cuja chave privada foi comprometida, e que é conhecida pelo intruso.
  - O utilizador tenta obter a CRL mais recente, que revogaria o certificado. O intruso fornece uma versão antiga da CRL.
  - O utilizador aceita o certificado porque não tem como saber que a CRL que utilizou estava desactualizada.

- 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 ao utilizador 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.

- O utilizador 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.
- 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 CA hierarquicamente inferior.
- No entanto, a granularidade garantida nunca é inferior ao período de publicação da CRL: não é possível garantir que os utilizadores obtenham a CRL extraordinária antes da data de publicação da próxima CRL periódica.

# Codificação

- As CRLs, como todas as estruturas de dados na PKI, são definidos e representados em ASN.1. A codificação é feita utilizando as Distinguished Encoding Rules (DER).
- O ficheiro que contém uma CRL consiste na codificação DER da seguinte estrutura ASN.1:

```
CertificateList ::= SEQUENCE {  
    tbsCertList          TBSCertList,  
    signatureAlgorithm  AlgorithmIdentifier,  
    signatureValue      BIT STRING }
```

- Além da estrutura de atributos apresentada nos slides anteriores (**tbsCertificate**), aparece também a assinatura da Autoridade de Certificação (**signatureAlgorithm** e **signatureValue**).



# Principais Críticas

- O X.509 baseia-se num sistema de directório. A chave primária desse directório é o Distinguished Name (DN): um DN tem de ser único. Mas como é que se garante esta unicidade?
- Se cada DN apenas pode ser associado a uma entidade, nada impede uma entidade de ter vários DNs. Além disso, nada obriga a que o DN contenha elementos de identificação baseados na identidade real da entidade. Assim, um certificado associa um DN a uma chave pública, mas não se pode dizer que identifique a entidade.
- As garantias que se esperam de uma CA relativamente à verificação da informação constante de um certificado não são abrangidas pela norma!

- Em resumo, o X.509 não foca o esforço necessário para validar a informação constante num certificado, nem define um significado global para essa informação, um significado absoluto, exterior às CAs.
- O X.509 remete para as CPSs das CAs todos os aspectos relacionados com confiança e com semântica da informação constante num certificado.
- Os certificados não são legíveis directamente pelo utilizador: quem garante que uma aplicação mostra ao utilizador toda a informação relevante acerca de um certificado?
- O X.509 implica a utilização de um directório ou repositório para distribuição de certificados e CRLs.

- As CRLs são muito pouco utilizadas, e reconhecidas por poucos como a solução para o problema das revogações de certificados.
- O conceito de “raiz da relação de confiança” e Root CA transfere o problema da confiança de um nível local para um nível global, potenciando os efeitos da ignorância ou da fraude.
- O processo de inicialização e manutenção de uma lista de certificados, nomeadamente de Root CAs em que uma entidade confia, está mal definido.
- Qualquer assinatura torna-se efectivamente inválida se deixar de existir um certificado válido que permita verifica-la!
- Qual a duração ideal do período de validade de um certificado?

# Introdução

- Os riscos associados à utilização indevida de um certificado revogado podem não ser aceitáveis.
- Em alternativa ou adição à consulta de uma CRL, pode ser necessária informação actual sobre o estado de revogação de um certificado.
- O OCSP permite a uma aplicação determinar o estado de um certificado com maior frescura temporal, ou mesmo para obter informação adicional sobre o certificado.
- O Cliente OCSP emite um pedido a um Responder OCSP (Servidor) e suspende a aceitação do certificado até que este forneça uma resposta.
- A extensão `AuthorityInfoAccess` permite indicar num certificado que a CA suporta este serviço.

# Pedidos OCSP

- Um pedido OCSP contém a seguinte informação, podendo ser assinados:
  - Versão do protocolo
  - Identificação do serviço requisitado
  - Identificador do certificado alvo
  - Extensões opcionais
- O processamento de um pedido pelo Responder passa pelas seguintes fases:
  - Verificação do formato da mensagem.
  - Verificação de que o servidor está configurado para fornecer o serviço requisitado.
  - Verificação de que o pedido contém toda a informação necessária.
  - Construção da resposta.

- Num pedido OCSP, os certificados para os quais é solicitada a informação de revogação são indicados numa lista de estruturas ASN.1 `CertID` :

```
CertID ::= SEQUENCE {
    hashAlgorithm      AlgorithmIdentifier,
    issuerNameHash     OCTET STRING,
                    -- Hash of Issuer's DN
    issuerKeyHash      OCTET STRING,
                    -- Hash of Issuers public key
    serialNumber       CertificateSerialNumber }
```

- A CA emissora de cada certificado é indicada através de valores de hash do seu *Distinguished Name* e da sua chave pública.

# Respostas OCSP

- As respostas OCSP consistem
  - num **identificador do tipo da resposta** e
  - numa **sequência de bytes com seu conteúdo**.
- As respostas OCSP são assinadas digitalmente por uma das seguintes entidades:
  - A CA que emitiu o certificado em questão.
  - Um Trusted Responder, i.e. um Responder em cuja chave pública o cliente confie.
  - Um CA Designated Responder i.e. um Responder autorizado pela CA que emitiu o certificado em questão (detentor de um certificado com uma extensão específica para este fim – `extendedKeyUsage` – emitido pela mesma CA).

- O conteúdo da resposta consiste nos seguintes itens:
  - A versão da sintaxe da resposta.
  - O nome do Responder.
  - **Respostas para cada um dos certificados contidos no pedido.**
  - Uma assinatura digital da resposta.
- Para cada certificado é enviada a seguinte informação:
  - O identificador do certificado alvo.
  - O estado do certificado: **good**, **revoked** ou **unknown**.
  - Período de validade da resposta.
- Uma resposta **good** significa apenas que não foi encontrado nenhum registo de revogação. Não indica que o certificado sequer exista!



- A indicação da validade da resposta pode ser feita de três formas diferentes:
  - **thisUpdate** Hora a que o Responder sabe que a resposta é correcta.
  - **nextUpdate** Hora a que o Responder sabe que vai ser capaz de fornecer uma resposta actualizada. Se não for indicada indica que qualquer nova resposta conterá informação actualizada.
  - **producedAt** Hora a que o Responder assinou a resposta.
- A validação de uma resposta OCSP por parte de um cliente implica verificar que:
  - há correspondência entre os certificados do pedido e da resposta.
  - a assinatura da resposta é válida e provém de um agente autorizado.
  - a validade da resposta é suficientemente recente.

- No serviço OCSP básico a resposta usa a seguinte estrutura ASN.1:

```
ResponseData ::= SEQUENCE {  
  version          [0] EXPLICIT Version DEFAULT v1,  
  responderID      ResponderID,  
  producedAt       GeneralizedTime,  
  responses         SEQUENCE OF SingleResponse,  
  responseExtensions [1] EXPLICIT Extensions OPTIONAL }  
  
SingleResponse ::= SEQUENCE {  
  certID           CertID,  
  certStatus       CertStatus,  
  thisUpdate       GeneralizedTime,  
  nextUpdate       [0] EXPLICIT GeneralizedTime OPTIONAL,  
  ... }  
}
```

## Serviços Extendidos

- O OCSP é standardizado no RFC2560. Neste documento é apenas definido o serviço básico descrito anteriormente.
- A IETF publicou entretanto um Internet Draft definindo o OCSP V2, uma extensão à funcionalidade básica que define três serviços:
  - Online Revocation Status (ORS)
  - Delegated Path Validation (DPV)
  - Delegated Path Discovery (DPD)
- O serviço é indicado numa extensão incluída nos pedidos OCSP no campo `requestExtensions`.
- O ORS corresponde à funcionalidade básica e é o caso default i.e. quando nenhum serviço é indicado trata-se de uma invocação do ORS.

# Delegated Path Validation

- Permite a uma aplicação transferir o processamento da validação de cadeias de certificados complexas para um servidor central.
- Isto permite simplificar as aplicações cliente, reduzindo ao mínimo a funcionalidade de validação de certificados a implementar.
- O protocolo permite que a aplicação cliente controle os aspectos essenciais do comportamento do servidor, por exemplo:
  - Restringir as cadeias de certificação aceitáveis.
  - Se o servidor se deve basear em CRLs e/ou OCSP para fazer a validação.
  - Quais as políticas de certificação que são relevantes para a aplicação.

# Delegated Path Discovery

- Permite a uma aplicação que processe certificados obter do servidor a informação disponível sobre um certificado e a sua revogação: cadeias de certificados, CRLs, respostas de outros servidores OCSP, etc.
- A aplicação pode validar certificados obtendo informação de todos os pontos de acesso à PKI disponíveis no servidor OCSP e.g. X.500, LDAP, HTTP, FTP, etc.
- Tal como no serviço anterior, a aplicação pode controlar diversos aspectos da operação do servidor:
  - Restringir as cadeias de certificação aceitáveis.
  - Quais as políticas de certificação relevantes.
  - Mecanismo interactivo de selecção de uma cadeia de validação aceitável para o cliente em termos de políticas de certificação.

# Introdução

- A empresa RSA Data Security, formada pelos inventores das técnicas RSA de criptografia de chave pública, tem um papel importante na criptografia moderna.
- A sua divisão RSA Laboratories mantém os Public Key Cryptography Standards (PKCS), muito importantes na implementação e utilização de PKIs.
- Os PKCS visam preencher o vazio que existe nas normas internacionais relativamente a formatos para transferência de dados que permitam a compatibilidade / interoperabilidade entre aplicações que utilizem criptografia de chave pública.
- Existem doze standards deste tipo: PKCS#1, #3, #5, #6, #7, #8, #9, #10, #11, #12, #13 e #15.

- Os objectivos da RSA na publicação destes standards são, segundo eles próprios, os seguintes:
  - Manter a compatibilidade com os standards existentes, nomeadamente com PEM (Privacy-Enhanced Mail Protocol).
  - Ir além dos standards existentes, para permitir uma melhor e mais completa integração entre aplicações, normalizando a troca segura de qualquer tipo de dados.
  - Produzir um standard que possa ser incluído numa futura versão dos standards OSI (Open Systems Interconnection).
- Os PKCS descrevem a sintaxe de mensagens de uma forma abstracta, utilizando o ASN.1, e não restringem a sua codificação.
- A aproximação utilizada é a de especificar algoritmos criptográficos e a sintaxe de mensagens de forma ortogonal.

# O que falta standardizar?

- Assinaturas digitais:
  - Como codificar e transferir este tipo de informação.
  - Como utilizar os diferentes algoritmos criptográficos (funções de hash e assinaturas digitais) para gerar e utilizar este tipo de informação.
  - Como armazenar, proteger e transferir chaves privadas.
  - Como codificar chaves privadas de algoritmos específicos.
  - Definir algoritmos de Password Based Encryption para proteger chaves privadas com passwords.
- Envelopes digitais:
  - Como codificar e transferir este tipo de informação.
  - Como utilizar os diferentes algoritmos criptográficos (cifras simétricas e cifras assimétricas) para gerar e utilizar este tipo de informação.
  - Como no caso anterior, como lidar com chaves privadas.



- Certificação e PKI
  - Como codificar e transferir pedidos de certificação.
  - Como codificar e transferir certificados.
  - Como incluir um conjunto alargado de atributos num certificado.
  - Como codificar e transferir CRLs.
  - Como utilizar algoritmos de assinatura digital.
  - Como codificar as chaves públicas de algoritmos específicos.
- Acordo de Chaves
  - Como codificar e transferir as mensagens envolvidas, em termos abstractos.
  - Concretizar para algoritmos específicos e.g. Diffie-Hellman.

# PKCS#1: RSA Encryption Standard

- Normaliza a utilização do algoritmo RSA nas seguintes aplicações:
  - **Assinaturas Digitais** A informação a assinar é inicialmente reduzida a um valor de hash utilizando um algoritmo de message digest (e.g. MD5). O resultado é então cifrado com a chave privada RSA.
  - **Envelopes Digitais** A informação a proteger é cifrada com a chave de sessão utilizando um algoritmo simétrico (e.g. DES). Posteriormente, a chave de sessão é cifrada com a chave pública RSA.
- O formato de uma mensagem contendo uma assinatura ou envelope digital PKCS#1 está definido no PKCS#7.
- Esta norma também inclui uma sintaxe para chaves RSA que é compatível com as normas X.509 e PEM.

# PKCS#3: Diffie-Hellman Key Agreement Standard

- Normaliza a utilização do protocolo de acordo de chaves Diffie-Hellman no estabelecimento de chaves secretas (de sessão).
- Destina-se a ser incluído, ao nível das Camadas de Rede e de Transporte, numa versão futura do modelo OSI.
- Este protocolo permite a dois utilizadores acordarem uma chave secreta, sobre um canal inseguro, sem trocarem informação que permita a um intruso obter essa mesma chave.

# PKCS#5: Password-Based Encryption Standard

- Descreve um método para cifrar um array de bytes utilizando uma chave secreta calculada a partir de uma password (Password-Based Encryption ou PBE).
- Destina-se à protecção de chaves privadas em situações que exijam a sua transferência.
- Isto pode ser necessário, por exemplo, quando as chaves são geradas pela CA, e não pelo utilizador; ou quando o utilizador necessita transferir a chave para outra máquina.
- A cifragem utilizada baseia-se no DES.

# PKCS#6: Extended-Certificate Syntax Standard

- Estende a definição de certificados X.509 permitindo a associação de outros atributos à entidade titular do certificado.
- O “PKCS#9: Selected Attribute Types” lista diversos atributos que podem ser incluídos num certificado X.509.
- Um exemplo de um atributo definido nestas normas é o endereço de e-mail do titular. Este atributo é bastante utilizado.

# PKCS#7: Cryptographic Message Syntax Standard

- Define uma sintaxe para mensagens criptográficas, nomeadamente assinaturas digitais e envelopes digitais.
- Esta sintaxe admite recursividade, e.g. pode haver uma assinatura digital de um envelope digital.
- No caso das assinaturas digitais, permite a associação de atributos de natureza arbitrária aos dados assinados.
- Como caso particular desta sintaxe é também definido um meio para distribuir certificados e CRLs.
- É compatível com o PEM no sentido em que uma mensagem PKCS#7 pode ser convertida de e para mensagens PEM sem necessidade de operações criptográficas: basta alterar o formato.

# PKCS#8: Private-Key Information Syntax Standard

- Define uma sintaxe para informação relativa a chaves privadas: o valor da chave, o algoritmo correspondente, e um conjunto de atributos associados.
- Define também uma sintaxe para chaves cifradas e.g. recorrendo às técnicas PBE definidas no PKCS#5.
- A norma PKCS#9 lista alguns dos atributos que podem ser associados a uma chave privada.
- Como exemplo de um atributo que pode ser associado a uma chave privada temos a identificação de uma Root CA. Desta forma é possível inicializar o utilizador com uma raiz para as suas relações de confiança.

# PKCS#10: Certification Request Syntax Standard

- Define uma sintaxe para pedidos de certificação.
- Um pedido de certificação inclui:
  - os atributos de identificação do futuro titular do certificado;
  - outros atributos e.g. o endereço da entidade que faz a requisição para que lhe seja enviado o certificado.
  - a chave pública a incluir no certificado;
  - uma assinatura digital do pedido que simultaneamente demonstra o conhecimento da chave privada e assegura a integridade da mensagem.
- Pretende-se que um pedido deste tipo forneça à CA toda a informação necessária para gerar o certificado. Note-se no entanto que existem outros aspectos a ter em conta no processo de certificação, nomeadamente a prova de identidade que tem de ser fornecida pelo titular.



# PKCS#12: Personal Information Exchange Syntax

- Descreve uma sintaxe para a transferência de informação de identificação pessoal, incluindo chaves privadas, certificados, chaves secretas e extensões.
- É uma norma muito útil uma vez que é utilizada por diversas aplicações (e.g. IE e Mozilla) para importar e exportar este tipo de informação.
- Suporta a transferência de informação pessoal em diferentes condições de manutenção da privacidade e integridade.
- O grau de segurança mais elevado prevê a utilização de assinaturas digitais e cifras assimétricas para protecção da informação.

- Isto implica a utilização de certificados e pares de chaves associados às plataformas de origem e destino, entre as quais se transfere a informação.
- Um nível de segurança intermédio prevê a utilização de PBE para protecção dos segredos.
- Esta norma é uma extensão do PKCS#8 para transferência de informação de identificação pessoal.

# PKCS#11, PKCS#13 e PKCS#15

- As normas PKCS#11 e PKCS#15 referem-se à utilização de dispositivos portáteis em criptografia, e serão abordadas mais adiante.
- A norma PKCS#13 está ainda em desenvolvimento e será dedicada às técnicas criptográficas baseadas em curvas elípticas.