Extensões à JCA

O que são?

- Implementações da JCA/JCE disponibilizadas por empresas como a IAIK incluem, na sua maioria, funcionalidades adicionais que são muito importantes na implementação de aplicações com criptografia.
- A titulo de exemplo vamos debruçar-nos sobre a implementação da IAIK, analisando os aspectos mais relevantes destas extensões:
  - Ferramentas de suporte à utilização do ASN.1.
  - A geração de certificados.

ASN.1

- A Abstract Syntax Notation One (ASN.1) encontra-se especificada nas normas X.208, e, embora tenha sido criada para especificar tipos de dados 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 necessário existir normas adicionais para definir como é que se codifica essa notação abstracta em sequências de bits.

Codificação do ASN.1

- As formas mais usuais usuais de codificação são:
  - 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.

Notas gerais sobre o ASN.1

- 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 semantica do ASN.1 i.e. conjuntos de espaços e linhas em branco são irrelevantes.
- 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.
- Os comentários são delimitados por **--** ou iniciados por **--** e terminados pela mudança de linha.
- Os identificadores começam por letra minúscula. As referências a tipos começam por letras maiúscula.

Alguns Tipos Universais

Type	Tag number (decimal)	Tag number (hexadecimal)
INTEGER	2	02
BIT STRING	3	03
OCTET STRING	4	04
NULL	5	05
OBJECT IDENTIFIER	6	06
SEQUENCE and SEQUENCE OF	16	10
SET and SET OF	17	11
PrintableString	19	13
T61String	20	14
IA5String	22	16
UTCTime	23	17

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
```

Implementação IAIK do ASN.1

- O package JCE da IAIK inclui uma livraria com classes que facilitam muito a manipulação de dados em ASN.1 (iaik.asn1), nomeadamente no que respeita aos tipos de dados definidos nos standards criptográficos.
- Esta livraria inclui:
  - Uma classe (**ASN1Object**) que implementa um objecto genérico ASN.1, que serve como base a todos as definições de tipos ASN.1.
  - Classes que implementam os tipos de dados nativos do ASN.1, ou definidos em standards criptográficos, derivadas da classe anterior.
  - Uma interface (**ASN1Type**) que permite a implementação de classes “serializáveis” de/para ASN.1 (e.g. Certificados X.509).
  - Classes que implementam a codificação/descodificação DER.
  - Uma classe (**ASN**) que regista os tipos ASN.1 existentes e que permite criar novas instancias de objectos ASN.1.

ASN1Object

- Esta classe abstracta implementa uma interface comum a todos os tipos ASN.1.
- Inclui métodos para:
  - Testar/definir o tipo de um objecto ASN.1.
  - Adicionar, eliminar, listar e obter elementos de tipos complexos.
  - Obter/atribuir o valor do objecto ASN.1.
  - Efectuar a interface com os objectos de codificação/descodificação DER.
- Uma classe específica de um tipo ASN.1 deve ser derivada da classe ASN1Object, fornecendo métodos adicionais para aceder ao conteúdo do tipo correspondente.

Codificação/Descodificação

- A classe **ASN1** permite codificar/descodificar objectos ASN.1 para/de o formato DER.
- O construtor desta classe aceita um array ou um stream de bytes, que transforma automaticamente numa representação em memória do objecto ASN.1 correspondente.
- O método **toASN1Object** permite gerar um objecto do tipo **ASN1Object** com base nessa representação.
- Em contrapartida, o método **toByteArray** gera um array de bytes com a codificação DER do objecto em questão.
- A classe **DerCoder** fornece métodos estáticos para converter arrays/streams de bytes em objectos ASN1Object e vice/versa.
- A classe **DerInputStream** permite descodificar streams DER de forma controlada, isto é extraindo um objecto ASN.1 de cada vez. Esta classe apresenta métodos específicos para instanciação de objectos dos tipos nativos do ASN.1.

ASN1Type

- Esta interface pode ser implementada por qualquer classe JAVA de forma a permitir a sua transformação directa de/para objectos do tipo ASN.1.
- O método **decode** desta interface deve ser implementado de forma a aceitar um objecto **ASN1Object** e a extrair desse objecto toda a informação necessária à inicialização da instancia em questão.
- O método **toASN1Object** deve implementar a operação inversa, isto é, instanciar um objecto do tipo **ASN1Object** contendo toda a informação respeitante à instancia em questão.
- Note-se que, com base nas ferramentas de codificação/descodificação descritas no slide anterior, é possível instanciar/codificar objectos que implementem a interface ASN1Type de/para o formato DER.

Geração de Certificados

- Como já foi referido, não é possivel criar certificados utilizando apenas os métodos disponibilizados na JCA.
- No entanto, alguns providers extendem a funcionalidade especificada na JCA oferecendo mecanismos para realizar essa tarefa.
- O provider IAIK é um dos que oferece essa funcionalidade. A sua opção consistui em disponibilizar uma instância de X509Certificate com métodos adicionais para criar, definir atributos e assinar certificados X.509.

Geração de Certificados (cont)

- As operações necessárias para criar um certificado X.509 com o provider IAIK são:
  - Criar um certificado vazio utilizando o construtor X509Certificate sem argumentos.
  - Definir pelo menos os atributos obrigatórios com os métodos **setPublicKey**, **setSerialNumber**, **setIssuerDN**, **setSubjectDN**, **setValidNotBefore** e **setValidNotAfter**.
  - Assinar o certificado com uma chave privada utilizando o método **sign**.
  - Note-se que um certificado gerado desta forma não está inserido numa infraestrutura de certificação: o issuer é o próprio dono da chave privada, que funciona como autoridade de certificação assinando o seu próprio certificado de chave pública.