Java and Cryptography

Java and Cryptography

The cryptographic functionality in Java used to be split between two different libraries:

- Java Cryptography Architecture (JCA)(*tightly integrated with the core Java API*)
- Java Cryptography Extensions (JCE)(*many of the advanced cryptographic operations that were previously under US export control*)

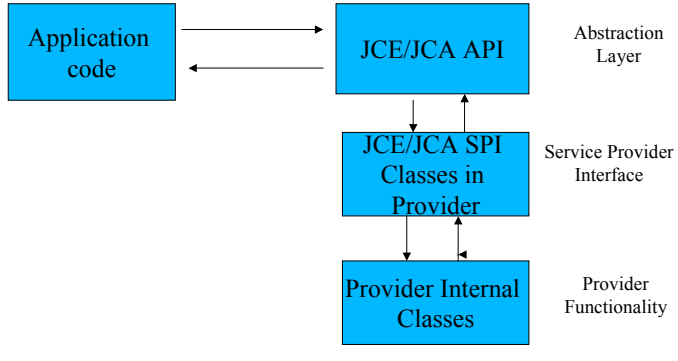
Now they are both shipped with Java SE and this division is not so obvious or important

Basic architecture

Provider-based architecture:

- JCA and JCE provide a set of classes and interfaces + factories enabling the creation of objects that conform to this classes;
- The *objects* that give functionality are provided by underlying implementation and are not directly visible to the developer;
- The collections of classes that provide implementation objects are called *providers*
- JCA and JCE have some simple mechanisms to add providers and to choose specific provider

Basic architecture



Design principles of JSA/JSE

- Algorithm independence
- Algorithm extensibility
- Implementation independence
- Implementation interoperability

Engines

Classes in JCA/JCE corresponding to categories of cryptographic operations are called *engines*

JCA engines

- **MessageDigest** (produces a hash value for a message)
- **Signature** (produces a digital signature of a document)
- **KeyPairGenerator** (produces a pair of keys)
- **KeyFactory** (breaks down a key into its discrete parts)
- **SecureRandom** (produce random numbers)
- **AlgorithmParameters** (manages the encoding/decoding of the parameters)
- **AlgorithmParameterGenerator** (generates a complete set of parameters required for a given algorithm)

JCA engines (cont.)

JCA engines (cont.)

- **Certificate factory** (creates public key certificates)
- **CertPathBuilder** (establishes relationship chains between certificates)
- **CertStore** (manages and stores certificates)

JCE engines

JCE engines

- **Cipher** (performs encryption/decryption)
- **KeyGenerator** (produces secret keys used by ciphers)
- **SecretKeyFactory** (operates on SecretKey instances)
- **KeyAgreement** (embodies a key agreement protocol)
- **Mac** (message authentication code functionality)

Location

- JCA classes are located in *java.security* package
- JCE classes are located in *javax.crypto* package

Providers

- SUN provider comes with the JCA
- SunJCE provider comes with the JCE

Third party providers:

- The Legion of the Bouncy Castle (http://www.bouncycastle.org)
- Cryptix (http://www.cryptix.org)
- …

Small Example: DES encryption

```
. . .
KeyGenerator kg = KeyGenerator.getInstance("DES");
SecretKey key = kg.generateKey();
SecretKeySpec keySpec = new
SecretKeySpec(key.getEncoded(), "DES");
Cipher cipher = Cipher.getInstance("DES");
cipher.init(Cipher.ENCRYPT_MODE, keySpec);
String plainText = "This is a secret";
byte[] cipherText = cipher.doFinal(plainText.getBytes());
. . .
```