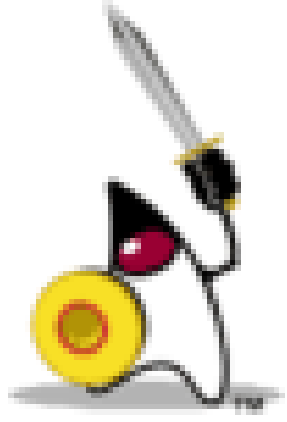
A short introduction to the JCE

(the Java Cryptography Extension)



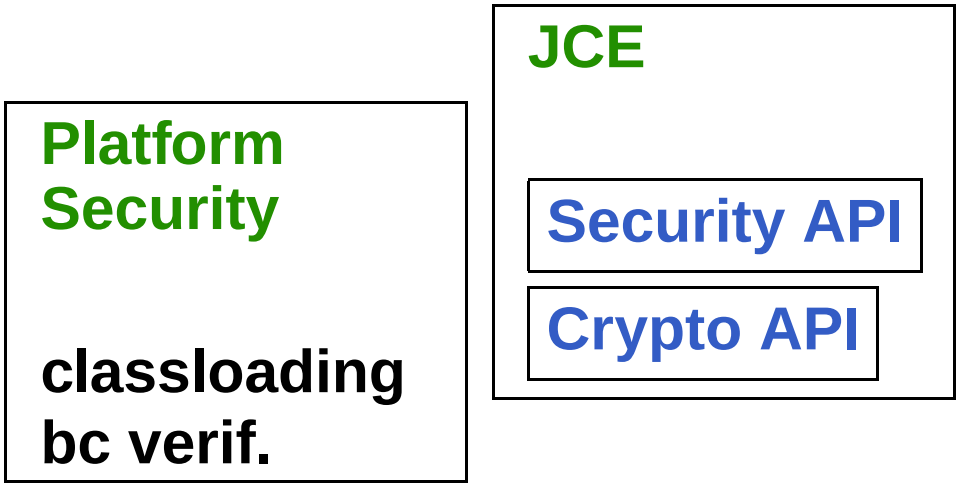
Martijn Oostdijk

University of Nijmegen

Overview of this talk

- Java & Security
- `SecureRandom`
- `MessageDigest`
- `Signature` and `Mac`
- `SecretKey`, `PublicKey` and `PrivateKey`
- `Cipher`
- `Certificate`
- What about Java Card?

Java & Security



- **Distinction Security/Crypto API due to US export regulations (now relaxed).**
- **The following countries may not receive ANY US-developed encryption items, including JCE: Afghanistan, Cuba, Iran, Iraq, Libya, North Korea, Serbia/Montenegro (Yugoslavia), Sudan, Syria.**

SecureRandom

```
...

SecureRandom random =
   SecureRandom.getInstance("SHA1PRNG");
random.setSeed(0x31415926535897 93L);

...

byte[] output = new byte[8];
random.nextBytes(output);
```

MessageDigest

```
byte[] input1 = ...;
byte[] input2 = ...;

MessageDigest digest =
   MessageDigest.getInstance("SHA-1");

digest.update(input1);
digest.update(input2);

...

byte[] output = digest.digest();
//@ assert output != null && output.length == 20;
```

`Signature` and `Mac`

```
byte[] input = ...;
PrivateKey privkey = ...;

Signature signature =
   Signature.getInstance("SHA1withRSA");
signature.initSign(privkey);

signature.update(input);

...

byte[] output = signature.sign();
//@ assert output != null && output.length == 20;
```

Mac™ *I'm lovin' it...* similar, uses `SecretKey`.

`SecretKey`, `PublicKey` and `PrivateKey`

- Use `KeyFactory` or `KeyGenerator` to construct `SecretKeys`
- Use `KeyPairGenerator` to construct `PublicKeys` and `PrivateKeys`

Cipher

```
byte[] input = ...;
PublicKey pubkey = ...;

Cipher cipher =
   Cipher.getInstance("RSA");
cipher.init(Cipher.ENCRYPT_MODE,pubkey);

cipher.update(input);

...

byte[] output = cipher.doFinal();
```

Certificate

- **Contains some basic info about the *owner* (name, organization, country, ...)**
- **Contains the *owner*'s public key**
- **Signed using the *issuer*'s private key**

```
Owner: CN=Martijn Oostdijk, OU=Unknown, O=Radboud University Nijmegen, L=N
Issuer: EMAILADDRESS=martijno@cs.kun.nl, CN=Martijn Oostdijk, OU=Security
Serial number: b
Valid from: Thu Nov 27 08:53:09 CET 2003 until: Fri Nov 26 08:53:09 CET 20
Certificate fingerprints:
         MD5:  CB:B7:16:40:87:6F:6B:F3:20:0E:DB:2A:9A:9A:29:66
         SHA1: A4:D6:2A:44:74:19:32:44:96:3B:30:B7:4B:F2:43:44:6B:E7:2A:68
```

What about Java Card?

- Very similar in style to JCE
- Limited collection of algorithms
- Algorithms not specified via `String` but via `short`
- Do not ‘hard code’ key material in source code
- Card can do RSA key generation

That’s all!

Crypto questions/problems?

- **Look at the examples on the web site**
- **Mail `martijno@cs.kun.nl`**