Bouncy Castle (Java + C#)

http://www.bouncycastle.org

Current version

1.41

License

What it is

The **Bouncy Castle Crypto APIs** for Java consist of the following:

- A lightweight cryptography API for Java and C#.
- A provider for the Java Cryptography Extension and the Java Cryptography Architecture.
- A clean room implementation of the JCE 1.2.1.
- A library for reading and writing encoded ASN.1 objects.
- A light weight client-side TLS API.
- Generators for Version 1 and Version 3 X.509 certificates, Version 2 CRLs, and PKCS12 files.
- Generators for Version 2 X.509 attribute certificates.
- Generators/Processors for S/MIME and CMS (PKCS7/RFC 3852).
- Generators/Processors for OCSP (RFC 2560).
- Generators/Processors for TSP (RFC 3161).
- Generators/Processors for OpenPGP (RFC 2440).
- A signed jar version suitable for JDK 1.4-1.6 and the Sun JCE.

Example

```
Security.addProvider(new BouncyCastleProvider());
KeyPairGenerator    kpg = KeyPairGenerator.getInstance("RSA", "BC");

kpg.initialize(1024);

KeyPair                 kp = kpg.generateKeyPair();

FileOutputStream    out1 = new FileOutputStream("secret.bin");
FileOutputStream    out2 = new FileOutputStream("pub.bin");

String id = "Test";
String passPhrase = "Ahoj";

PGPSecretKey    secretKey = new PGPSecretKey(
       PGPSignature.DEFAULT_CERTIFICATION, PGPPublicKey.RSA_GENERAL,
       kp.getPublic(), kp.getPrivate(), new Date(), id,
       PGPEncryptedData.CAST5, passPhrase.toCharArray(), null, null,
       new SecureRandom(), "BC");
secretKey.encode(out1);
out1.close();

PGPPublicKey    key = secretKey.getPublicKey();
key.encode(out2);
out2.close();
```

Supported cryptographic algorithms

Name	KeySizes (in bits)	Block Size	Notes
AES	0 .. 256 **(192)**	128 bit	
AESWrap	0 .. 256 **(192)**	128 bit	A FIPS AES key wrapper
Blowfish	0 .. 448 **(448)**	64 bit	
Camellia	128, 192, 256	128 bit	
CamelliaWrap	128, 192, 256	128 bit	
CAST5	0 .. 128**(128)**	64 bit	
CAST6	0 .. 256**(256)**	128 bit	
DES	64	64 bit	
DESede	128, 192	64 bit	
GOST28147	256	64 bit	
IDEA	128 **(128)**	64 bit	
Noekeon	128**(128)**	128 bit	
RC2	0 .. 1024 **(128)**	64 bit	
RC5	0 .. 128 **(128)**	64 bit	Uses a 32 bit word
RC5-64	0 .. 256 **(256)**	128 bit	Uses a 64 bit word
RC6	0 .. 256 **(128)**	128 bit	
Rijndael	0 .. 256 **(192)**	128 bit	
SEED	128**(128)**	128 bit	
SEEDWrap	128**(128)**	128 bit	
Serpent	128, 192, 256 **(256)**	128 bit	
Skipjack	0 .. 128 **(128)**	64 bit	

TEA	128 **(128)**	64 bit
Twofish	128, 192, 256 **(256)**	128 bit
XTEA	128 **(128)**	64 bit

Symmetric (Stream)

Note: default key sizes are in bold.

Name	KeySizes (in bits)	Notes
RC4	40 .. 2048 bits **(128)**	
HC128	(128)	
HC256	(256)	
Salsa20	128/256**(128)**	
VMPC	128/6144**(128)**	

Block Asymmetric

Encoding:

- OAEP - Optimal Asymmetric Encryption Padding
- PCKS1 - PKCS v1.5 Padding
- ISO9796-1 - ISO9796-1 edition 1 Padding

Note: except as indicated in PKCS 1v2 we recommend you use OAEP, as mandated in X9.44.

When placed together with RSA this gives a specification for an algorithm as;

- RSA/NONE/NoPadding
- RSA/NONE/PKCS1Padding
- RSA/NONE/OAEPWithSHA1AndMGF1Padding
- RSA/NONE/OAEPWithMD5AndMGF1Padding
- RSA/NONE/OAEPWithSHA224AndMGF1Padding
- RSA/NONE/OAEPWithSHA256AndMGF1Padding
- RSA/NONE/OAEPWithSHA384AndMGF1Padding
- RSA/NONE/OAEPWithSHA512AndMGF1Padding
- RSA/NONE/ISO9796-1Padding

Name	KeySizes (in bits)	Notes
RSA	any multiple of 8 bits large enough for the encryption**(2048)**	
ElGamal	any multiple of 8 bits large enough for the encryption**(1024)**	

Key Agreement

Diffie-Hellman key agreement is supported using the "DH", "ECDH", and "ECDHC" (ECDH with cofactors) key agreement instances.

Note: with basic "DH" only the basic algorithm fits in with the JCE API, if you're using long-term public keys you may want to look at the light-weight API.

ECIES

An implementation of ECIES (stream mode) as described in IEEE P 1363a.

Note: At the moment this is still a draft, don't use it for anything that may be subject to long term storage, the key values produced may well change as the draft is finalised.

Digest

Name	Output (in bits)	Notes
GOST3411	256	
MD2	128	
MD4	128	
MD5	128	
RipeMD128	128	basic RipeMD
RipeMD160	160	enhanced version of RipeMD
RipeMD256	160	expanded version of RipeMD128
RipeMD320	160	expanded version of RipeMD160
SHA1	160	
SHA-224	256	FIPS 180-2
SHA-256	256	FIPS 180-2
SHA-384	384	FIPS 180-2
SHA-512	512	FIPS 180-2
Tiger	192	
Whirlpool	512	

MAC

Name	Output (in bits)	Notes
Any MAC based on a block cipher, CBC (the default) and CFB modes.	half the cipher's block size (usually 32 bits)	
VMPC-MAC	128	
HMac-MD2	128	
HMac-MD4	128	
HMac-MD5	128	
HMac-RipeMD128	128	
HMac-RipeMD160	160	
HMac-SHA1	160	
HMac-SHA224	224	
HMac-SHA256	256	
HMac-SHA384	384	
HMac-SHA512	512	
HMac-Tiger	192	

Examples:

- DESMac
- DESMac/CFB8
- DESedeMac
- DESedeMac/CFB8
- DESedeMac64
- SKIPJACKMac
- SKIPJACKMac/CFB8

- IDEAMac
- IDEAMac/CFB8
- RC2Mac
- RC2Mac/CFB8
- RC5Mac
- RC5Mac/CFB8
- ISO9797ALG3Mac

Signature Algorithms

Schemes:

- GOST3411withGOST3410 (GOST3411withGOST3410-94)
- GOST3411withECGOST3410 (GOST3411withGOST3410-2001)
- MD2withRSA
- MD5withRSA
- SHA1withRSA
- RIPEMD128withRSA
- RIPEMD160withRSA
- RIPEMD160withECDSA
- RIPEMD256withRSA
- SHA1withDSA
- SHA1withECDSA
- SHA224withECDSA
- SHA256withECDSA
- SHA384withECDSA
- SHA512withECDSA
- SHA224withRSA
- SHA256withRSA
- SHA384withRSA
- SHA512withRSA
- SHA1withRSAandMGF1
- SHA256withRSAandMGF1
- SHA384withRSAandMGF1
- SHA512withRSAandMGF1

PBE

Schemes:

- PKCS5S1, any Digest, any symmetric Cipher, ASCII
- PKCS5S2, SHA1/HMac, any symmetric Cipher, ASCII
- PKCS12, any Digest, any symmetric Cipher, Unicode

Defined in Bouncy Castle JCE Provider

Name	Key Generation Scheme	Key Length (in bits)
PBEWithMD5AndDES	PKCS5 Scheme 1	64
PBEWithMD5AndRC2	PKCS5 Scheme 1	128
PBEWithSHA1AndDES	PKCS5 Scheme 1	64
PBEWithSHA1AndRC2	PKCS5 Scheme 1	128
PBEWithSHAAnd2-KeyTripleDES-CBC	PKCS12	128
PBEWithSHAAnd3-KeyTripleDES-CBC	PKCS12	192
PBEWithSHAAnd128BitRC2-CBC	PKCS12	128
PBEWithSHAAnd40BitRC2-CBC	PKCS12	40

PBEWithSHAAnd128BitRC4	PKCS12	128
PBEWithSHAAnd40BitRC4	PKCS12	40
PBEWithSHAAndTwofish-CBC	PKCS12	256
PBEWithSHAAndIDEA-CBC	PKCS12	128