The RSA.jar API:

RSAKeyPair.

```
public class RSAKeyPair {
   public RSAKeyPair(PRGen rand, int numBits)
   public RSAKey getPublicKey()
   public RSAKey getPrivateKey()
   public BigInteger[] getPrimes()
}
```

For RSAKeyPair, the bulk of the interesting work is performed by the constructor. This constructor creates an RSA key pair using the algorithm discussed in class.

RSAKey.

```
public class RSAKey {
   public RSAKey(BigInteger theExponent, BigInteger theModulus)
   public BigInteger getExponent()
   public BigInteger getModulus()
   public byte[] encrypt(byte[] plaintext)
   public byte[] decrypt(byte[] ciphertext)
   public byte[] sign(byte[] message)
   public boolean verifySignature(byte[] message, byte[] signature)
   public int maxPlaintextLength()
}
```

The RSAKey class implements core RSA functions, namely encrypting/decryption as well as signing/verification. Note that the RSAKey class is used for both public and private keys.

The sign() method generates a signature (array of bytes) that can be verified by the verifySignature() method of the other RSAKey in the private/public RSAKey pair. The verifySignature() method should be used by a public RSAKey object to verify a signature generated by the corresponding private RSAKey's sign() method. The maxPlaintextLength() method returns the largest N such that any plaintext of size N bytes can be encrypted with this key.