# Optional dependencies

## [Bouncy Castle](https://www.bouncycastle.org/)

Required mainly for writing keys to PEM files or for special keys/ciphers/etc. that are not part of the standard [Java Cryptography Extension](https://en.wikipedia.org/wiki/Java_Cryptography_Extension). See [Java Cryptography Architecture (JCA) Reference Guide](https://docs.oracle.com/javase/8/docs/technotes/guides/security/crypto/CryptoSpec.html) for key classes and explanations as to how *Bouncy Castle* is plugged in (other security providers).

**Caveat**: If *Bouncy Castle* modules are registered, then the code will use its implementation of the ciphers, keys, signatures, etc. rather than the default JCE provided in the JVM.

**Note:**

* The security provider can also be registered for keys/ciphers/etc. that are already supported by the standard JCE as a **replacement** for them.
* The *BouncyCastle* code can also be used to load keys from PEM files instead or in parallel with the built-in code that already parses the standard PEM formats for the default JCE supported key types.
* One can use the BouncyCastleKeyPairResourceParser to load standard PEM files instead of the core one - either directly or via SecurityUtils#setKeyPairResourceParser for **global** usage - even without registering or enabling the provider.
* The required *Maven* module(s) are defined as optional so must be added as an **explicit** dependency in order to be included in the classpath:

<dependency>  
 <groupId>org.bouncycastle</groupId>  
 <artifactId>bcpg-jdk15on</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.bouncycastle</groupId>  
 <artifactId>bcpkix-jdk15on</artifactId>  
 </dependency>

## NIO2 default socket factory replacements

Optional dependency to enable choosing between NIO asynchronous sockets (the default - for improved performance), and "legacy" sockets. **Note:** the required Maven module(s) are defined as optional so must be added as an **explicit** dependency in order to be included in the classpath.

### [MINA core](https://mina.apache.org/mina-project/)

<dependency>  
 <groupId>org.apache.mina</groupId>  
 <artifactId>mina-core</artifactId>  
 <!-- see SSHD POM for latest tested known version of MINA core -->  
 <version>2.0.17</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.apache.sshd</groupId>  
 <artifactId>sshd-mina</artifactId>  
 <version>...same as sshd-core...</version>  
 </dependency>

### [Netty](https://netty.io/)

Another a NIO client server framework option that can be used as a replacement for the default NIO asynchronous sockets core implementation. This is also an **optional** dependency and must be add explicitly via the sshd-netty artifact.

<dependency>  
 <groupId>io.netty</groupId>  
 <artifactId>netty-transport</artifactId>  
 <version>...Netty version...</version>  
 </dependency>  
 <dependency>  
 <groupId>io.netty</groupId>  
 <artifactId>netty-handler</artifactId>  
 <version>...Netty version...</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.apache.sshd</groupId>  
 <artifactId>sshd-netty</artifactId>  
 <version>...same as sshd-core...</version>  
 </dependency>

## [ed25519-java](https://github.com/str4d/ed25519-java)

Required for supporting [ssh-ed25519](https://tools.ietf.org/html/draft-bjh21-ssh-ed25519-02) keys and [ed25519-sha-512](https://tools.ietf.org/html/draft-josefsson-eddsa-ed25519-02) signatures. **Note:** the required Maven module(s) are defined as optional so must be added as an **explicit** dependency in order to be included in the classpath:

<!-- For ed25519 support -->  
 <dependency>  
 <groupId>net.i2p.crypto</groupId>  
 <artifactId>eddsa</artifactId>  
 </dependency>

The code contains support for reading *ed25519* [OpenSSH formatted private keys](https://issues.apache.org/jira/browse/SSHD-703).