Spring Boot JAR 安全加密运行工具，同时支持的原生JAR。

基于对JAR包内资源的加密以及拓展ClassLoader来构建的一套程序加密启动，动态解密运行的方案，避免源码泄露或反编译。

## 环境依赖

JDK 1.7 +

## 功能特性

* 无需侵入代码，只需要把编译好的JAR包通过工具加密即可。
* 完全内存解密，杜绝源码以及字节码泄露或反编译。
* 支持所有JDK内置加解密算法。
* 可选择需要加解密的字节码或其他资源文件，避免计算资源浪费。

## 使用步骤

<project>

<!-- 设置 jitpack.io 仓库 -->

<repositories>

<repository>

<id>jitpack.io</id>

<url>https://www.jitpack.io</url>

</repository>

</repositories>

<!-- 添加 XJar 依赖 -->

<dependencies>

<dependency>

<groupId>com.github.core-lib</groupId>

<artifactId>xjar</artifactId>

<version>LATEST\_VERSION</version>

</dependency>

</dependencies>

</project>

注：其中LATEST\_VERSION版本号查看<https://www.jitpack.io>

// Spring-Boot Jar包加密

public static void main(String[] args) {

String password = "io.xjar";

File plaintext = new File("/path/to/read/plaintext.jar");

File encrypted = new File("/path/to/save/encrypted.jar");

XBoot.encrypt(plaintext, encrypted, password);

}

// Spring-Boot Jar包解密

public static void main(String[] args) {

String password = "io.xjar";

File encrypted = new File("/path/to/read/encrypted.jar");

File decrypted = new File("/path/to/save/decrypted.jar");

XBoot.decrypt(encrypted, decrypted, password);

}

// Jar包加密

public static void main(String[] args) {

String password = "io.xjar";

File plaintext = new File("/path/to/read/plaintext.jar");

File encrypted = new File("/path/to/save/encrypted.jar");

XJar.encrypt(plaintext, encrypted, password);

}

// Jar包解密

public static void main(String[] args) {

String password = "io.xjar";

File encrypted = new File("/path/to/read/encrypted.jar");

File decrypted = new File("/path/to/save/decrypted.jar");

XJar.decrypt(encrypted, decrypted, password);

}

// 命令行运行JAR 然后在提示输入密码的时候输入密码后按回车即可正常启动

java -jar /path/to/encrypted.jar

//也可以通过传参的方式直接启动

java -jar /path/to/encrypted.jar --xjar.password=PASSWORD

## 参数说明

* --xjar.algorithm 加解密算法名称，缺省为AES，支持JDK所有内置算法，如AES / DES ...
* --xjar.keysize 密钥长度，缺省为128，根据不同的算法选取不同的密钥长度。
* --xjar.ivsize 向量长度，缺省为128，根据不同的算法选取不同的向量长度。
* --xjar.password 密码

## 进阶用法

// 只加密自身项目及相关模块的源码不加密第三方依赖，可以通过XEntryFilter来定制需要加密的JAR包内资源

public static void main(String[] args) {

String password = "io.xjar";

File plaintext = new File("/path/to/read/plaintext.jar");

File encrypted = new File("/path/to/save/encrypted.jar");

XBoot.encrypt(plaintext, encrypted, password, (entry) -> {

String name = entry.getName();

return name.startsWith("BOOT-INF/classes/") || name.startsWith("BOOT-INF/lib/io-xjar-");

});

}

## 插件集成

[XJar-Maven-Plugin](https://github.com/core-lib/xjar-maven-plugin) GitHub: <https://github.com/core-lib/xjar-maven-plugin>

<project>

<!-- 设置 jitpack.io 插件仓库 -->

<pluginRepositories>

<pluginRepository>

<id>jitpack.io</id>

<url>https://jitpack.io</url>

</pluginRepository>

</pluginRepositories>

<!-- 添加 XJar Maven 插件 -->

<build>

<plugins>

<plugin>

<groupId>com.github.core-lib</groupId>

<artifactId>xjar-maven-plugin</artifactId>

<version>LATEST\_VERSION</version>

<executions>

<execution>

<goals>

<goal>spring-boot</goal>

</goals>

<phase>package</phase>

<configuration>

<password>io.xjar</password>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

</project>

#### 也可以通过Maven命令执行

mvn xjar:spring-boot -Dxjar.password=io.xjar

mvn xjar:jar -Dxjar.password=io.xjar -Dxjar.targetDir=/directory/to/save/target.xjar