

用Java的加密机制来保护你的数据 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/270/2021\\_2022\\_\\_E7\\_94\\_A8Java\\_E7\\_9A\\_84\\_c67\\_270347.htm](https://www.100test.com/kao_ti2020/270/2021_2022__E7_94_A8Java_E7_9A_84_c67_270347.htm)

Java开发工具包 (JDK)对加密和安全性有很好的支持。其中一个优势就是其内置的对Socket通信的支持。因此，很容易做到在服务器和客户之间建立安全的数据流。Java streams 是一个强大的编程工具。java.io包提供了很多标准的流类型，并能很容易的建立自己的流类型。流的一个有用的特点是和链表一样的简单处理过程。将FileReader和BufferedReader链接起来。我们在用客户机/服务器应用程序的时候也会用到类似的概念。关键字对于验证来说，关键字很重要，运行KeyGen来产生一个关键字。我们采用同步方法，所以客户机和服务器必须用相同的关键字。安全socket 我们从一个简单的类开始，它提供我们在普通socket对象之上的加密。构造器创建了变量并初始化了密码：

```
outCipher = Cipher.getInstance(algorithm).
```

```
outCipher.init(Cipher.ENCRYPT_MODE, key). inCipher =  
Cipher.getInstance(algorithm).
```

```
inCipher.init(Cipher.DECRYPT_MODE, key). 因为socket是双向的通信，所以我们采用两个密码。加密输出的数据并解密输入的数据。我们使用getInputStream()和getOutputStream(),这两种方法来加密合解密通用的输入和输出的经过包装的数据流。
```

在JCE的javax.crypto包中包含CipherInputStream和CipherOutputStream这两种流类型。他们接收输入输出的流对象和密码对象。Socket 服务器 开始写我们的socket服务器类吧。SecretSocketServer在一个端口打开ServerSocket，当接收到

连接时，使用SocketHandler产生一个线程来操作连接。Socket 句柄通过KeyGen来定位关键字，并建立一个SecretSocket 对象。 `Key key = KeyGen.getSecretKey(). this.ss = new SecretSocket(s, key).` 所有的socket 处理都是通过SecretSocket而不是Socket对象。然后我们使用下面的代码：`in = ss.getInputStream().` 记住，在SecretSocket中，`getInputStream`是和CipherInputStream以及InputStream相结合的。因为SocketHandler 是一个可执行的界面，我们为它生成一个run()方法。这个方法只是在等待socket的数据：`boolean bool = true. while (bool) { bool = listen(). }` listen方法用来监听socket。 `int aByte. while ((aByte = in.read()) >= 0) { system.out.println((char)aByte). }` Socket 客户 现在我们来看看客户端。客户端的工作和服务端很相似，只是反过来了。首先，我们创立一个套接字连接到服务器。使用KeyGen 找到关键字，创立一个安全套接字（SecretSocket）。然后我们利用它的OutputStream给服务器发送数据：`Key key = KeyGen.getSecretKey(). Socket s = new Socket("localhost", 4444). SecretSocket ss = new SecretSocket(s, key). OutputStream os = ss.getOutputStream(). Os.write("Hello World!".getBytes()). Os.flush(). Os.close(). Os.close().` 总结：通过JCE中的java流和链表，我们可以轻松的加密基于socket的网络通信。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)