了解Java密码扩展的基础 PDF转换可能丢失图片或格式，建议阅读原文
https://www.100test.com/kao_ti2020/145/2021_2022__E4_BA_86_E8_A7_A3Java_c104_145452.htm Java密码扩展（The Java Cryptography Extension），是JDK1.4的一个重要部分，基本上，他是由一些包构成的，这些包形成了一个框架并实现了一些加密，密钥生成算法和协议，消息认证码等算法，这篇文章将想你介绍JCE的安装和使用。 值得注意的是，尽管JCE是JDK1.4的核心包的一部分，我们将首先用JDK1.2及高一点的版本向你演示一下如果安装配置JCE（静态安装）。稍后，将向你介绍如何在不安装的情况下使用JCE（动态安装）。最后，将演示怎么生成密钥和密码，及如果进行基本的加密.解密。提供者是什么？ 提供者是特定加密算法的实现者，有的提供者（提供的加密技术）是免费的，有的不免费，IBM, Bouncy Castle, 和 RSA都是一些（加密）提供者.在本文的后面，我们将考察一下来自Bouncy Castle的RSA算法。Sun 也向大家说明了如果实现自己的提供者（需要符合jDK的一些约定）。静态安装 在安装和使用JCE之前，你需要从 Sun Web site（这里是以暗中sun的提供者为例）.获得他的安装包，JCE有sun他自己的安全提供者-sunJCE，为了吧sunJCE静态的安装到默认的提供者列表中，你需要修改安全属性文件：#8226. /jre/lib/security/java.security (UNIX)如果你把JDK安装在C:\jdk1.3,你需要编辑以下文件：C:\jdk1.3\jre\lib\security\java.security 为了安装SunJCE，你需要在以上文件中加入：security.provider.n=com.sun.crypto.provider.SunJCE把n用你加

入的提供者的优先级代替（注意：序号要保持递增，不能跳过，但可以调整前后顺序）。Listing A 用于查看你安装过的提供者的信息，结果在Listing B中列出，显示提供者的能力，比如说可用的加密算法。Listing A: ProviderInformation.javaimport java.security.Provider.import java.security.Security.import java.util.Set.import java.util.Iterator.public class ProviderInformation { public static void main(String[] args) { Provider[] providers = Security.getProviders(). for (int i = 0. i Provider provider = providers[i]. System.out.println("Provider name: " provider.getName()). System.out.println("Provider information: " provider.getInfo()). System.out.println("Provider version: " provider.getVersion()). Set entries = provider.entrySet(). Iterator iterator = entries.iterator(). while (iterator.hasNext()) { System.out.println("Property entry: " iterator.next()). } } }} 