谈谈Java加密技术（四）Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/644/2021_2022__E8_B0_88_E8_B0_88Java_c104_644438.htm 接下来我们分析DH加密算法，一种适基于密钥一致协议的加密算法。 DH Diffie-Hellman算法（D-H算法），密钥一致协议。是由公开密钥密码体制的奠基人Diffie和Hellman所提出的一种思想。简单的说就是允许两名用户在公开媒体上交换信息以生成"一致"的、可以共享的密钥。换句话说，就是由甲方产出一对密钥（公钥、私钥），乙方依照甲方公钥产生乙方密钥对（公钥、私钥）。以此为基线，作为数据传输保密基础，同时双方使用同一种对称加密算法构建本地密钥（SecretKey）对数据加密。这样，在互通了本地密钥（SecretKey）算法后，甲乙双方公开自己的公钥，使用对方的公钥和刚才产生的私钥加密数据，同时可以使用对方的公钥和自己的私钥对数据解密。不单单是甲乙双方两方，可以扩展为多方共享数据通讯，这样就完成了网络交互数据的安全通讯！该算法源于中国的同余定理中国馀数定理。 流程分析： 1.甲方构建密钥对儿，将公钥公布给乙方，将私钥保留；双方约定数据加密算法；乙方通过甲方公钥构建密钥对儿，将公钥公布给甲方，将私钥保留。 2.甲方使用私钥、乙方公钥、约定数据加密算法构建本地密钥，然后通过本地密钥加密数据，发送给乙方加密后的数据；乙方使用私钥、甲方公钥、约定数据加密算法构建本地密钥，然后通过本地密钥对数据解密。 3.乙方使用私钥、甲方公钥、约定数据加密算法构建本地密钥，然后通过本地密钥加密数据，发送给甲方加密后的数据；甲方使用私钥、乙方公钥

、约定数据加密算法构建本地密钥，然后通过本地密钥对数据解密。 通过java代码实现如下： import java.security.Key. import java.security.KeyFactory. import java.security.KeyPair. import java.security.KeyPairGenerator. import java.security.PublicKey. import java.security.spec.PKCS8EncodedKeySpec. import java.security.spec.X509EncodedKeySpec. import java.util.HashMap. import java.util.Map. import javax.crypto.Cipher. import javax.crypto.KeyAgreement. import javax.crypto.SecretKey. import javax.crypto.interfaces.DHPrivateKey. import javax.crypto.interfaces.DHPublicKey. import javax.crypto.spec.DHParameterSpec. /** *//** * DH安全编码组件 * * @version 1.0 * @since 1.0 */ public abstract class DHCoder extends Coder { public static final String ALGORITHM = "DH". /** *//** * 默认密钥字节数 * * gt. * DH * Default Keysize 1024 * Keysize must be a multiple of 64, ranging from 512 to 1024 (inclusive). * gt. */ private static final int KEY_SIZE = 1024. /** *//** * DH加密下需要一种对称加密算法对数据加密，这里我们使用DES，也可以使用其他对称加密算法。 */ public static final String SECRET_ALGORITHM = "DES". private static final String PUBLIC_KEY = "DHPublicKey". private static final String PRIVATE_KEY = "DHPrivateKey". /** *//** * 初始化甲方密钥 * * @return * @throws Exception */