极客学院
jikexueyuan.com

对称密码的编程使用

对称密码的编程使用 — **课程概要**

- 对称密码概述
- DES 算法的编程使用
- 3DES 算法的编程使用
- AES 算法的编程使用

对称密码概述

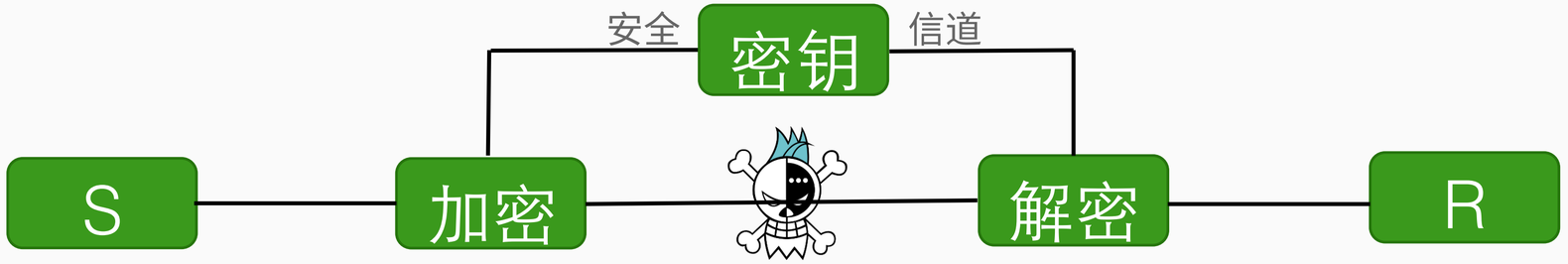
对称密码概述

- 对称密码的概念
- 对称密码的种类
- 对称密码的作用

对称密码概述 — **对称密码的概念**

1.加密密钥和解密密钥相同，对于大多数对称密码算法，加解密过程互逆

2.加解密通信模型



3.特点：算法公开、计算量小、加密速度快、加密效率高

4.弱点：双方都使用同样密钥，安全性得不到保证

5.分组密码工作模式

（1）ECB：电子密码本

（2）CBC：密文链接

（3）CFB：密文反馈

（4）OFB：输出反馈

（5）CTR：计数器

6.分组密码填充方式

（1）NoPadding

（2）PKCS5Padding

（3）ISO10126Padding

对称密码概述 — **对称密码的种类**

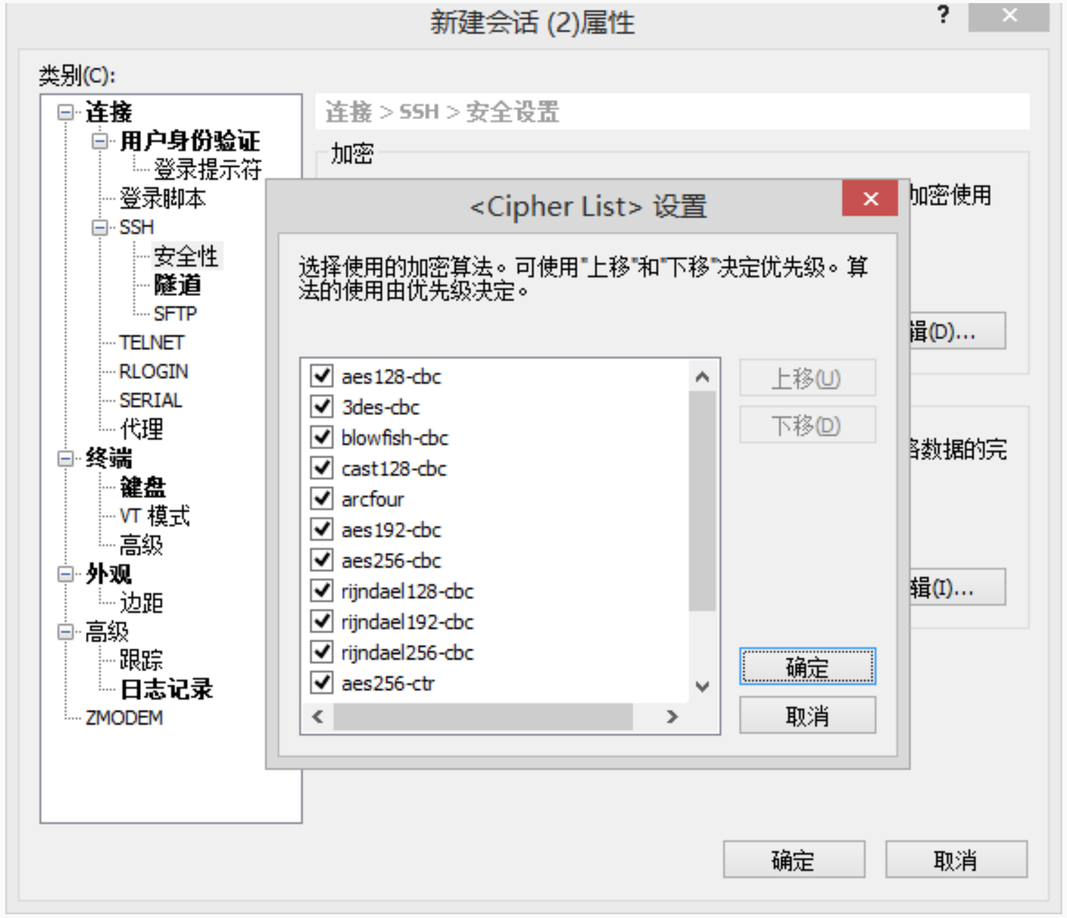
常用对称密码：

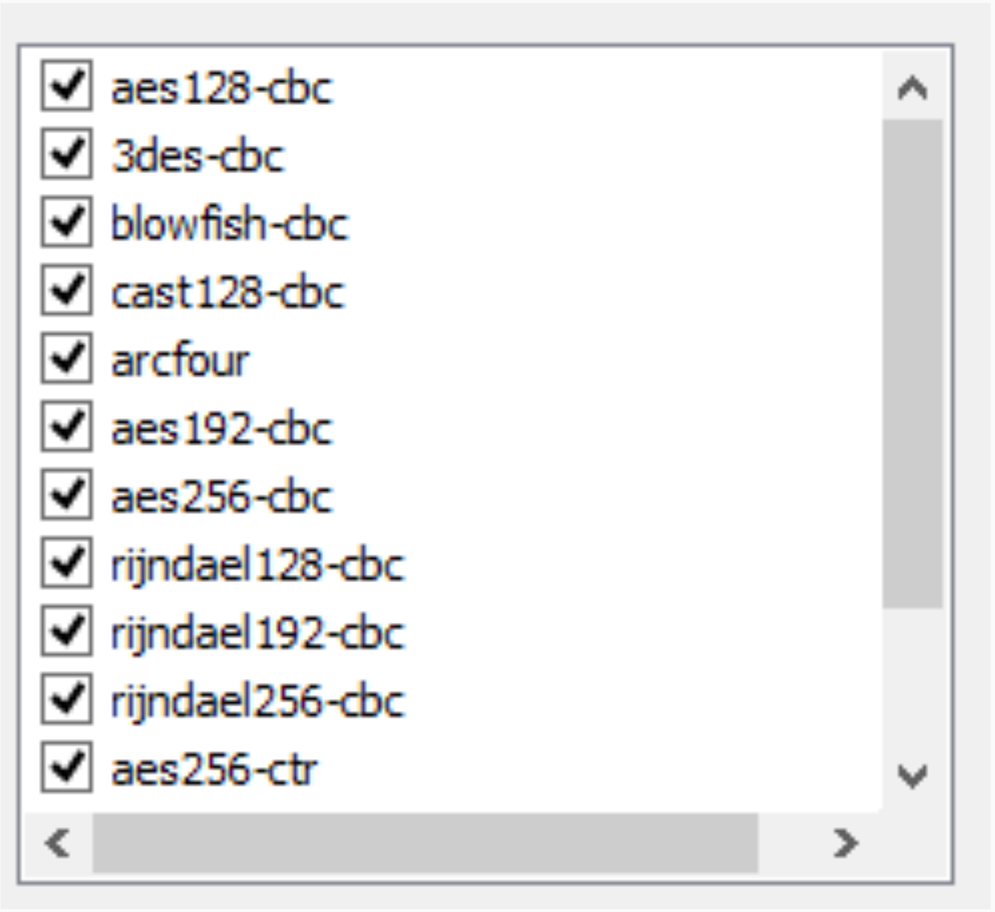
（1）DES（Data Encryption Standard）

（2）3DES（Triple DES、DESede）

（3）AES（Advanced Encryption Standard）

对称密码概述 — **对称密码的作用**





DES 算法的编程使用

DES 算法的编程使用

- DES 算法基本概念
- DES 算法编程使用

DES 算法的编程使用 — DES 算法基本概念

1.DES：数据加密标准，是对称加密算法领域中的典型算法

2.特点：密钥偏短（56位）、生命周期短

3.JDK实现

算法	密钥长度	默认密钥长度	工作模式	填充方式
DES	56	56	ECB、CBC、PCBC、CTR、CTS、CFB、CFB8-CFB128、OFB、OFB8-OFB128	NoPadding、PKCS5Padding、ISO10126Padding

DES 算法的编程使用 — **DES 算法编程使用**

1.生成密钥

```
//KeyGenerator，密钥生成器
KeyGenerator keyGen = KeyGenerator.getInstance("DES");
//初始化密钥生成器
keyGen.init(56);
//生成密钥
 SecretKey secretKey = keyGen.generateKey();
```

DES 算法的编程使用 — DES 算法编程使用

2.加/解密

```
//恢复密钥
SecretKey secretKey = new SecretKeySpec(key, "DES");
//Cipher 完成加密或解密工作
Cipher cipher = Cipher.getInstance("DES");
// 根据密钥，对Cipher 初始化，ENCRYPT_MODE、 DECRYPT_MODE
cipher.init(Cipher.ENCRYPT_MODE, secretKey);
// 加密或解密
byte[] cipherByte = cipher.doFinal(data);
```

3DES 算法的编程使用

3DES 算法的编程使用

- 3DES 算法基本概念
- 3DES 算法编程使用

3DES 算法的编程使用 — 3DES 算法基本概念

1. 3DES：将密钥长度增至112位或168位，通过增加迭代次数提高安全性
2. 缺点：处理速度较慢、密钥计算时间较长、加密效率不高
3. JDK实现

算法	密钥长度	默认密钥长度	工作模式	填充方式
3DES	112、168	168	ECB、CBC、PCBC、CTR、CTS、CFB、CFB8-CFB128、OFB、OFB8-OFB128	NoPadding、PKCS5Padding、ISO10126Padding

3DES 算法的编程使用 — **3DES 算法编程使用**

1.生成密钥

```
KeyGenerator keyGen = KeyGenerator.getInstance("DESede");
keyGen.init(168);  //可指定密钥长度为112或168，默认为168
SecretKey secretKey = keyGen.generateKey();
```

3DES 算法的编程使用 — **3DES 算法编程使用**

2.加/解密

```
SecretKey secretKey = new SecretKeySpec(key, "DESede");
Cipher cipher = Cipher.getInstance("DESede");
cipher.init(Cipher.ENCRYPT_MODE, secretKey);
byte[] cipherByte = cipher.doFinal(data);
```

AES 算法的编程使用

AES 算法的编程使用

- AES 算法基本概念
- AES 算法编程使用

AES 算法的编程使用 — AES 算法基本概念

1.AES：高级数据加密标准，能够有效抵御已知的针对DES算法的所有攻击

2.特点：密钥建立时间短、灵敏性好、内存需求低、安全性高

3.JDK实现

算法	密钥长度	默认密钥长度	工作模式	填充方式
AES	128、192、256	128	ECB、CBC、PCBC、CTR、CTS、CFB、CFB8-CFB128、OFB、OFB8-OFB128	NoPadding、PKCS5Padding、ISO10126Padding

AES 算法的编程使用 — **AES 算法编程使用**

1.生成密钥

```
KeyGenerator keyGen = KeyGenerator.getInstance("AES");
keygen.init(128);  //默认128，获得无政策权限后可为192或256
SecretKey secretKey = keyGen.generateKey();
```

AES 算法的编程使用 — **AES 算法编程使用**

2.加/解密

```
SecretKey secretKey = new SecretKeySpec(key, "AES");
Cipher cipher = Cipher.getInstance("AES");
cipher.init(Cipher.ENCRYPT_MODE, secretKey);
byte[] cipherByte = cipher.doFinal(data);
```

极客学院

jikexueyuan.com

中国最大的IT职业在线教育平台