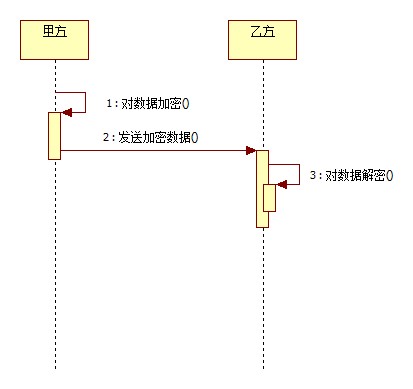
# 几种Java加密算法

1.Base64加密算法

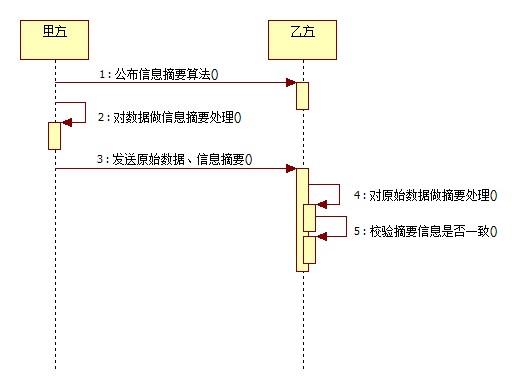
常用于邮件、http加密、截取http信息，比如在两方通过邮件交流时，需要使用Base64加密技术将信息内容在传送过程中进行加密，等接受者接收时进行解密再查看；

具体实现图：

2.MD5——信息摘要算法

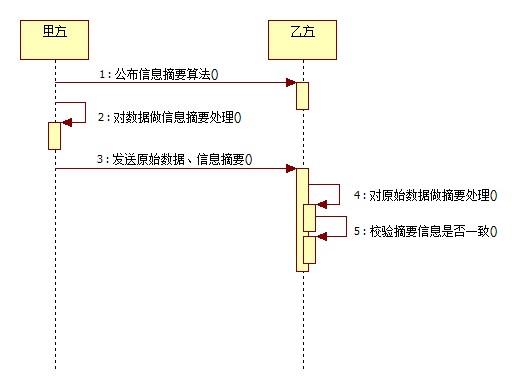
常用于文件校验

MD5——>将自己数组采用Base64技术进行加密转化成相应的字符串；在Java中有Java.security.MessageDigest类用于为应用程序提供信息摘要算法；

具体实现原理：

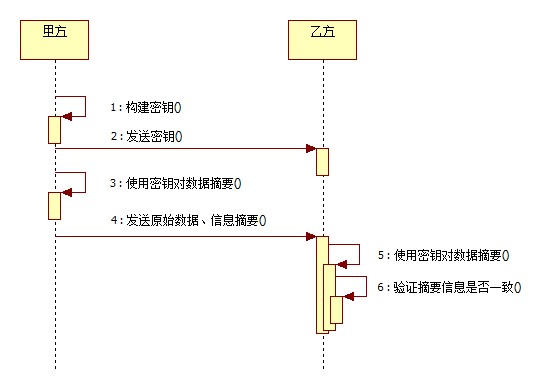
3.SHA——安全散列算法

一般广泛用于电子商务等信息安全领域，比如在淘宝在进行一些商务交易时需要进行对其中的信息进行加密管理，对卖家的相关信息要进行保密，这其中就可以使用SHA算法将信息内容转化成散列码，最后在接受者接收时进行解密；

实现原理图：

4.HAMC——散列消息鉴别码

用于密钥的Hash算法的认证协议

实现原理图：

5.对称加密算法——其实质也是通过MD5以及SHA对字符串加密生成密钥

一般用来对敏感数据等信息进行加密，

DES（数据加密算法，一般入口参数有：key、Data、Mode）——是一种数据加密标准，适合加密大量的数据的场合；

3DES：基于DES，对一块数据用三个不同的密钥进行三次加密，强度更高；

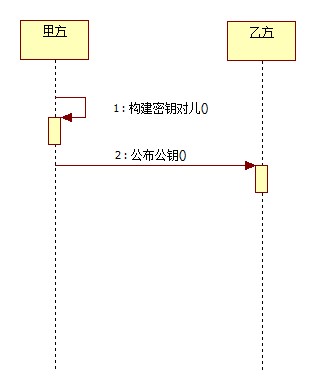
AES：高级加密标准，速度快，加密安全性高；

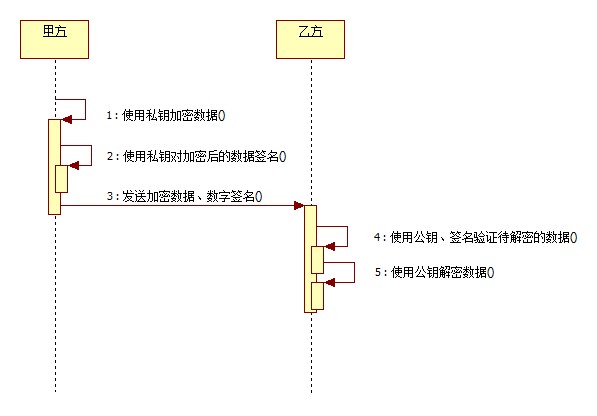
6.非对称的加密算法

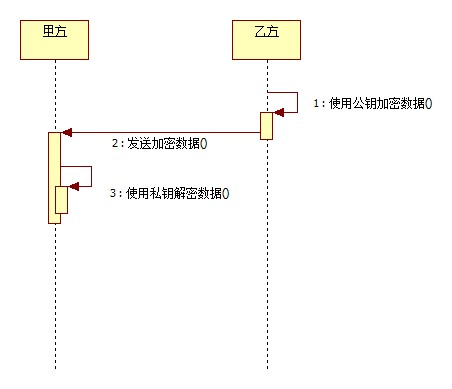
RSA——常可用于数据加密也能用于数字签名

这种算法与对称算法中的DES相比，最大的优点就是数据具有公钥和私钥，不像DES中只有一个密钥，如果这把密钥丢失，数据也就不安全啦，而RSA却相反，还同时支持数字签名，能够对传输过来的数据进行校验，确保数据在传输过程中不被修改。

具体实现序列图：

第一步：构建密钥对，公布公钥，保留私钥；

第二步：发送者私钥加密数据和数据签名，接收方用公钥验证；

第三步：接受者使用公钥加密数据，会发送给发送者，通过私钥解密。

DSA——用于数字签名，不但有公钥、私钥，还有数字签名，私钥加密生成数字签，而公钥验证数据和签名，保证数据在传输过程中不被修改，因为只是对发送者发送的数据进行加密，可以看做是一种单向的加密。