**文件传输加密实验报告**

# 实验工具和语言

VS2010 + c#(windows form) + dll（c++ 的动态连接库）

# 实验步骤

1. 首先了解des,ras,md5算法基本原理，并把对应代码编译成动态链接库，供c# 可视化使用。
2. 利用c#的socket编程，完成server和client端的代码编写
3. 把以上两步进行结合并进行调试。

# 实验详细步骤

1. 首先利用rsa.cpp 文件产生密钥对，并把私钥文件放在客户端，把公钥文件放在服务器端。（因为是从客户端发送信息到服务器端）。
2. 信息传递需要用到des文本加密算法。des加密需要用到key，在这一步我把key通过rsa加密传输给服务器端，这样服务器端接受到通过des加密的文件能够解密。
3. 为传输信息计算MD5值，把MD5值通过rsa进行加密传输给服务器端
4. 把信息通过des加密传输给服务器端
5. 服务器通过des解密信息，并计算MD5值，服务器通过rsa解密接收的md5，比对这两个MD5是否一样，来验证信息的完整性。

详情如下图所示：两个方框分别表示客户端和服务端的数据：

rsa的公钥文件：public.txt

rsa的私钥文件：private.txt

原始信息

第二步信息传递如下所示：

des加使用的key

通过rsa加密的key

rsa加密

rsa的私钥文件：private.txt

通过rsa加密后的key

rsa的私钥文件：private.txt

原始des加密使用的key

原始信息

第三步信息传递如下所示：

计算信息MD5值

加密后的MD5值

rsa加密

rsa的私钥文件：private.txt

通过rsa加密后的key

加密后的MD5值

rsa的私钥文件：private.txt

原始des加密使用的key

信息的MD5值

原始信息

第四步信息传递如下所示：

原始信息

通过des加密后的信息

des加密

rsa的私钥文件：private.txt

通过rsa加密后的key

加密后的MD5值

通过des加密后的信息

rsa的私钥文件：private.txt

原始des加密使用的key

信息的MD5值

原始信息

第五步服务器端的验证：

rsa的私钥文件：private.txt

通过rsa加密后的key

加密后的MD5值

通过des加密后的信息

解密后的MD5值

解密后的信息

解密后信息的MD5值

rsa的私钥文件：private.txt

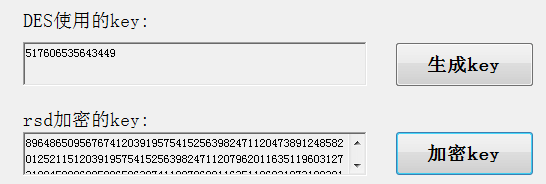
原始des加密使用的key

信息的MD5值

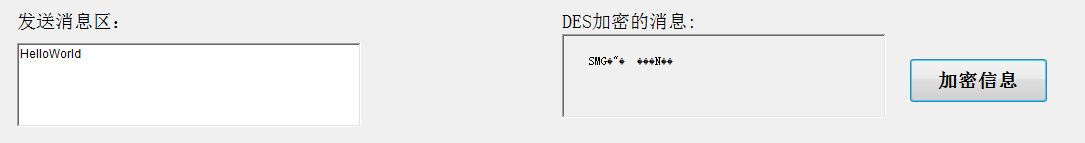
原始信息

# 实验结果

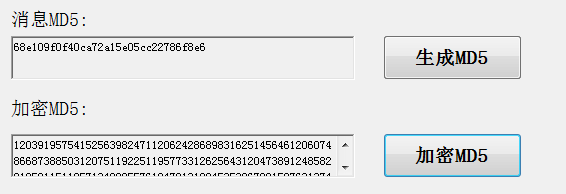
1. 能够正常生成DES算法需要的key，对key使用rsa进行加密



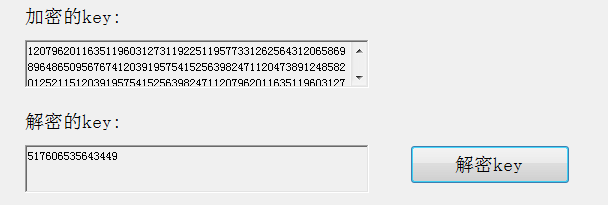
1. 能够使用DES算法对信息进行加密



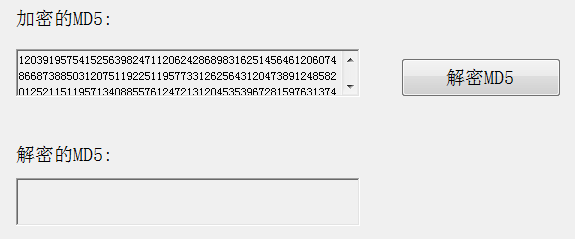
1. 能够对消息进行MD5值的计算，并且对MD5值进行加密处理



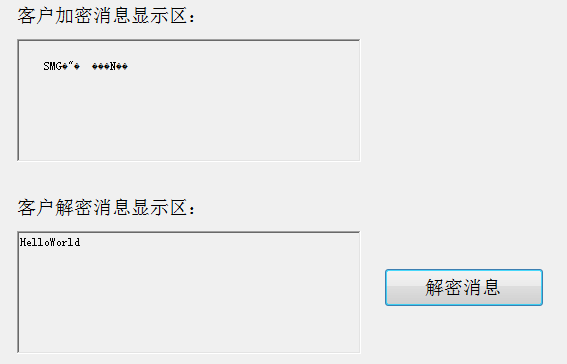
1. 服务器端能够对加密的key进行解密



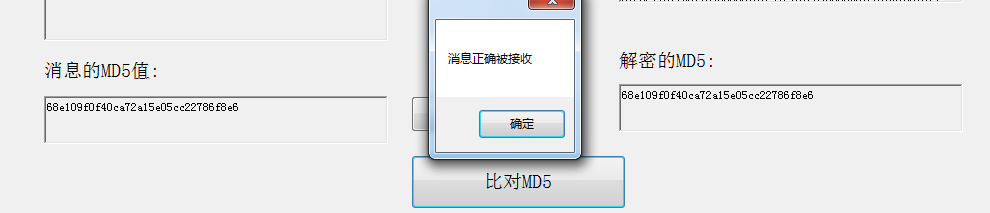
1. 服务器端能够对加密的MD5进行解密



1. 服务器端能够对加密的信息进行解密



1. 最后对两个MD5值进行比较



# 总结

1. 在做该实验收获很多，了解了加密的基本算法和原理。
2. 在编译动态链接库时也遇到一些问题，提升了解决问题的能力
3. 在用c#调用c++ dll时，存在编码不一致问题，通过解决这些问题，对编码有了更深刻的理解。