

网络安全实验报告

题 目 基于口令的认证过程实现

专 业 信息安全

学 号 1190201012

学 生 张中界

指 导 教 师 王彦

# 一、实验目的

1. 客户端输入用户名，口令，随机产生认证码，使用散列函数计算用户名与口令的散列值1，使用散列值1与认证码计算散列值2，将用户名，散列值2，认证码明文传送到服务器端。
2. 服务器端以数据库（如access）保存用户名和散列值1的对应关系。收到客户端信息后，以同样的方法计算散列值2’。 如散列值2’=散列值2，则认证成功，成功后用散列值1加密认证码发送给客户端。客户端解密后写到指定文件。
3. 实验目的，掌握随机函数的使用，掌握散列函数，加解密函数的使用。包的格式与发包的顺序，可以等同于协议的三要素。掌握程序与数据库的链接。

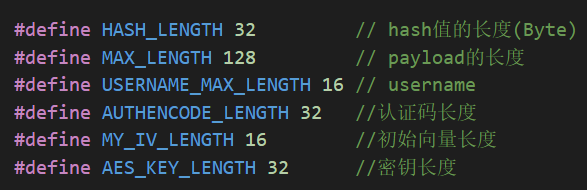
# 二、实验过程

实验环境：Ubuntu16.04 x64 编程语言：C++

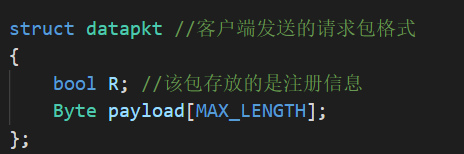
本实验使用的密码学库为Cryptopp，实现C/S架构的通信

## 1. 数据包格式

首先是一些参数长度的定义：

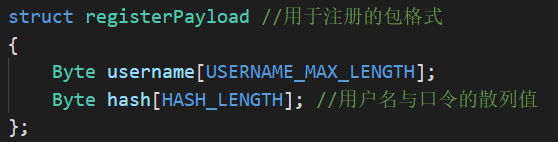


1. 客户端的请求包格式：



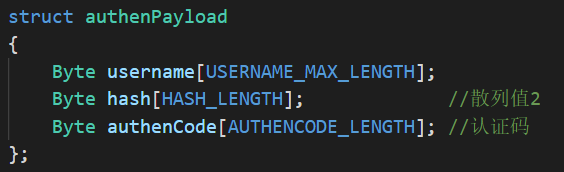
第一个字段R标识该包的功能：认证还是注册。

1. 注册包payload字段的格式



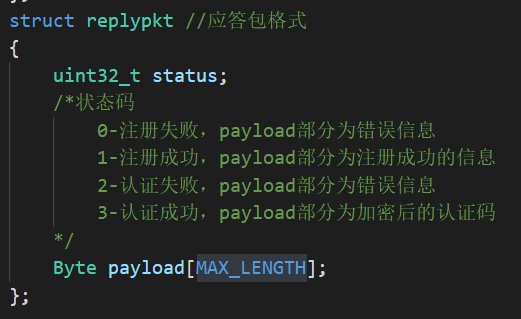
username为用户名，hash为用户名与口令的散列值，即散列值1。

1. 认证包payload字段的格式



username用户名，hash为散列值2，authenCode为认证码。

1. 应答包格式



status标识注册或认证的结果，payload字段为相关的提示信息，若status为3（认证成功），则payload字段为加密后的认证码。

## 2. 交互过程

（1） 注册模式

①由用户提供用户名和口令，客户端使用SHA256算法计算用户名和口令的散列值，将待发送包的R字段置为1，将构造好的包发送给服务端；

②服务端接收到注册包后，提取出用户名和散列值，查询数据库，若没有与用户名重复的元组则插入一条新元组，并向客户端回复一个应答包，提示注册成功，若过程中出现错误则在应答包中提示错误信息；

③客户端接收应答包。

（2） 认证模式

①用户输入用户名和口令，客户端使用SHA256算法计算用户名和口令的散列值，记为散列值1；

②客户端生成随机的认证码，与散列值1计算散列值2；

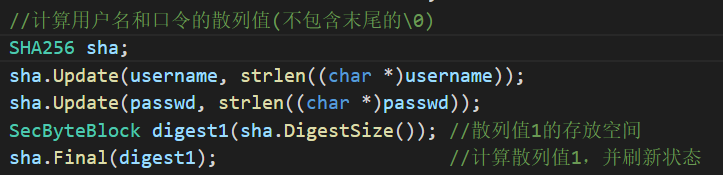
③客户端构造请求包并发送给服务端；

④服务端接收到请求包后，取出用户名，在数据库中查询得到散列值1’，由散列值1’和认证码计算散列值2’，比对散列值2与散列值2’，若相等则认证成功，使用散列值1作为密钥，散列值1的前一半作为初始向量，调用AES-CBC算法加密认证码，构造应答包并发送；

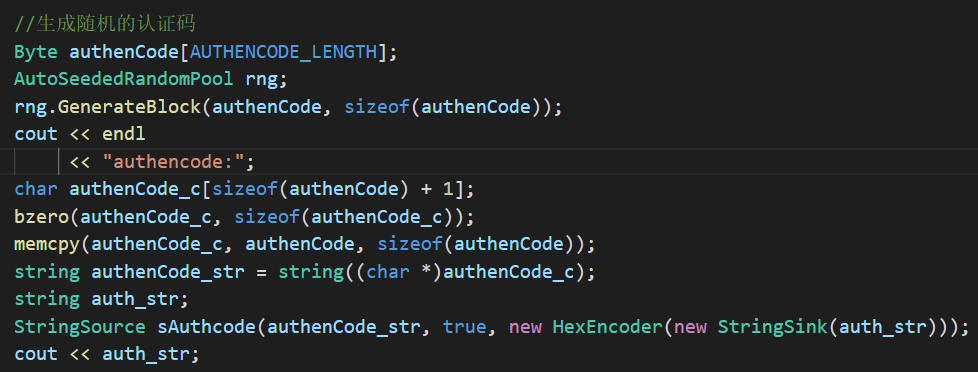
1. 客户端接收应答包，若认证成功则使用相同的方法解密得到认证码。

## 3. 关键代码

1. 客户端
   1. 计算散列值1

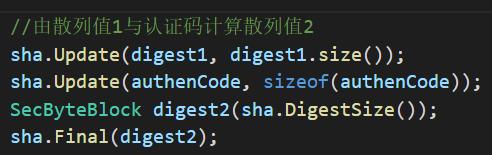


* 1. 生成认证码

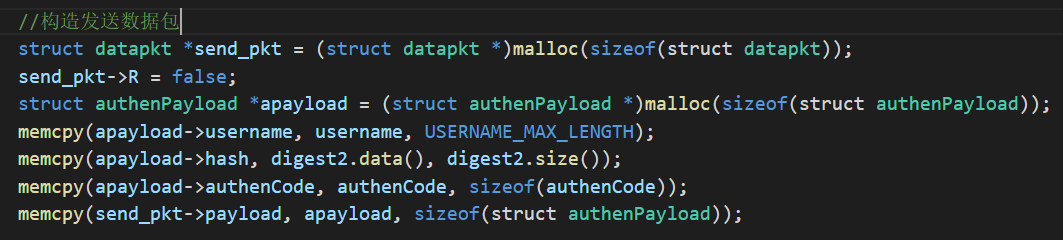


使用Cryptopp提供的随机数池生成随机数

* 1. 计算散列值2



* 1. 构造发送数据包



* 1. AES-CBC解密

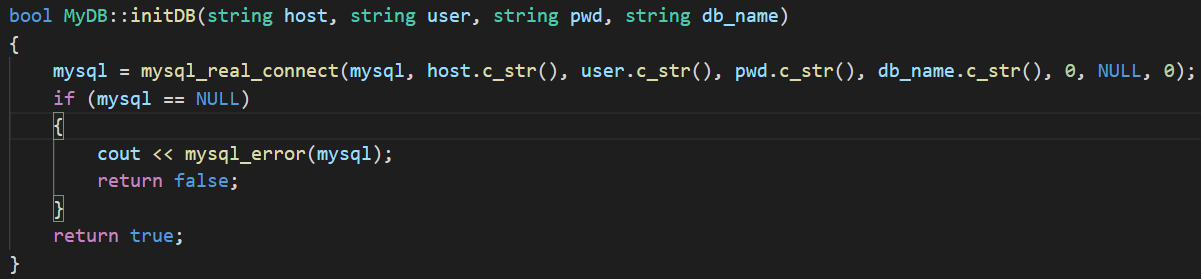


SetKeyWithIV构造解密器，得到认证码后在标准输出打印。

1. 服务端
   1. 连接数据库

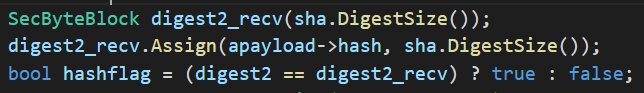


这里封装了数据库操作的一些函数，initDB的实现如下：



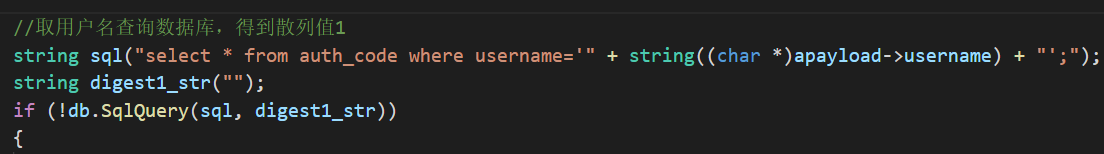
值得一提的是，数据库中存储的散列值为十六进制编码后的结果，所以在插入数据和取出数据时要进行相应的编解码。

* 1. 比较散列值2与散列值2’

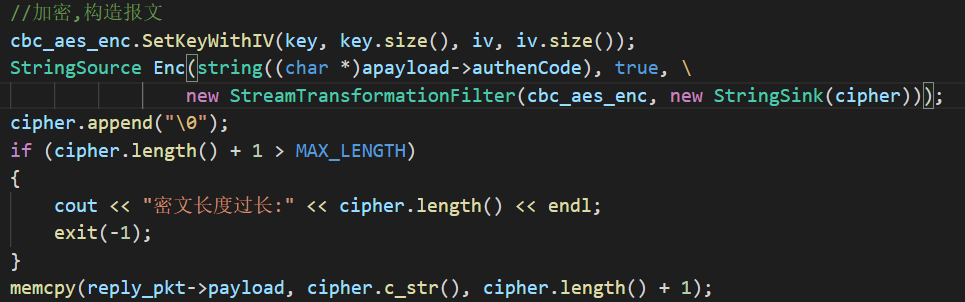


由于SecByteBlock类重载了’==’运算符，所以将要比较的两个散列值封装在两个SecByteBlock对象中，直接使用’==‘比较。

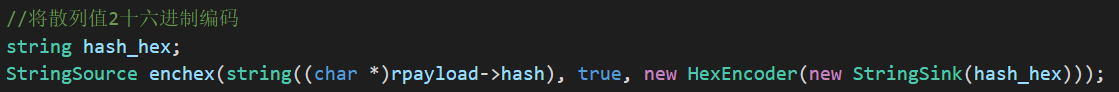
* 1. 查询数据库



* 1. AES-CBC加密

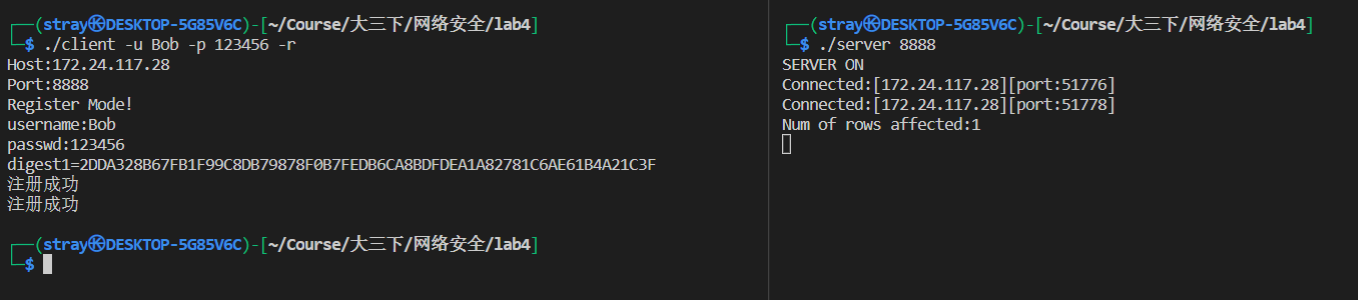


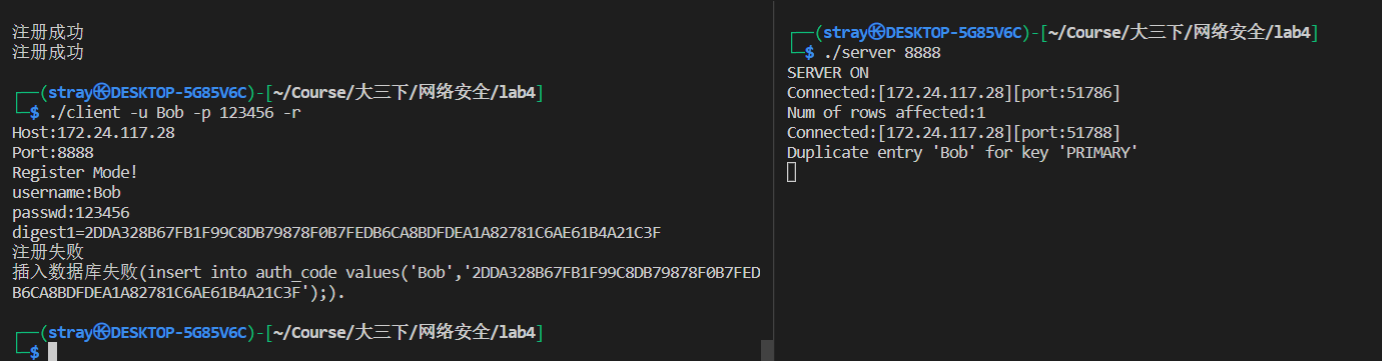
* 1. 十六进制编码



## 4. 运行结果

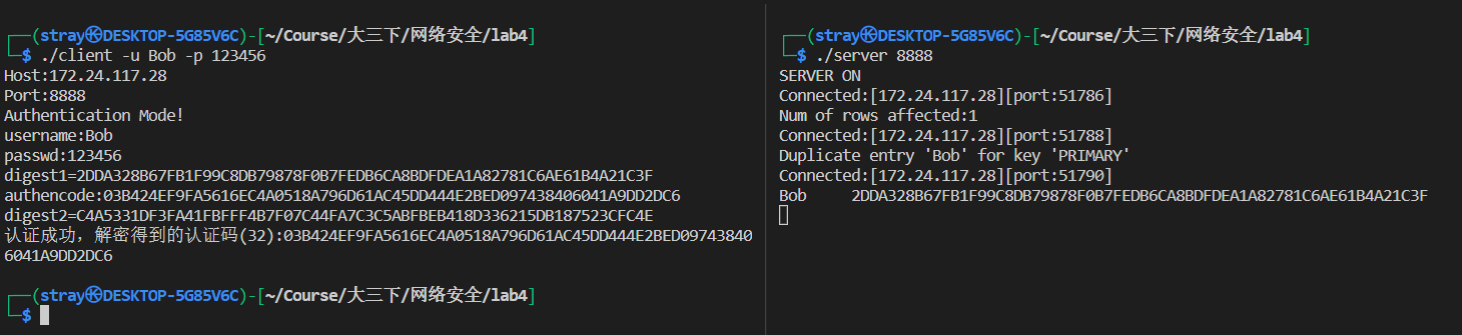
注册模式：

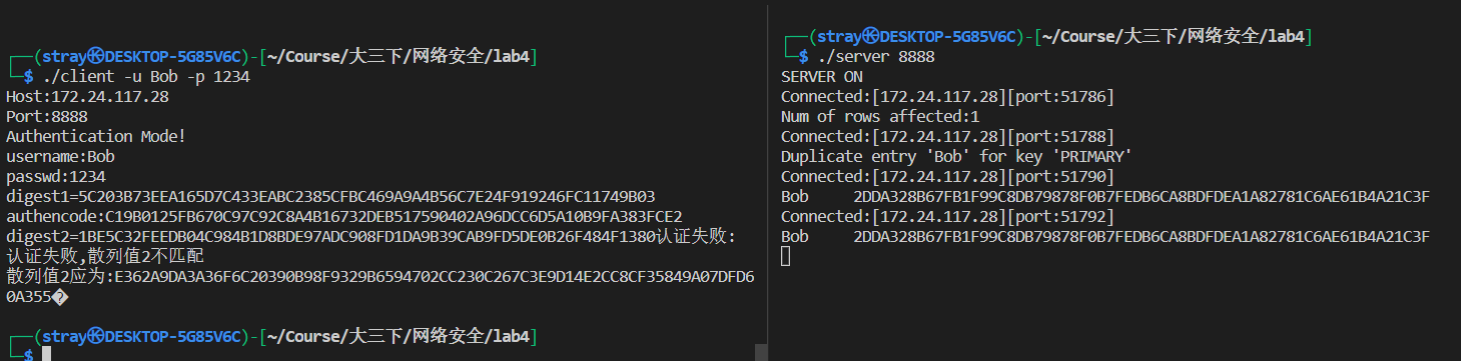




(用户名不能重复)

认证模式：





(密码错误)