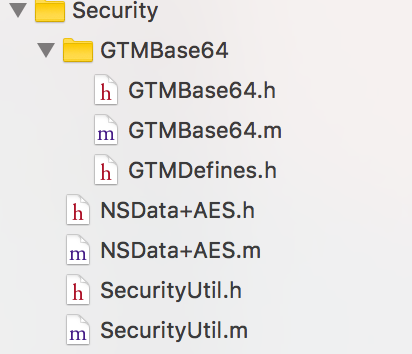
关于AES加密解密

工程添加左图文件。

一、加密

引入 头文件 #import “SecurityUtil.h"

添加如下方法

-(NSString \*)encryptAESwithStr:(NSData \*)data

{

Byte \* setPasswdWithAES\_dataToByte = (Byte \*)[data bytes];

NSString \* string=@"";

for(int i=0;i<[data length];i++){

NSString \* lastStr = [NSString stringWithFormat:@"%x",setPasswdWithAES\_dataToByte[i]&0XFF];

if (lastStr.length==1) {

lastStr=[@"0" stringByAppendingString:lastStr];

}

string=[string stringByAppendingString:lastStr];

}

string=[string uppercaseString];

return string;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

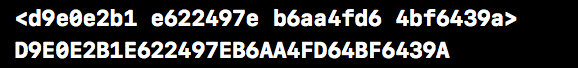
NSData \*data = [SecurityUtil encryptAESData:@"123456" app\_key:@"1234567812345678"];

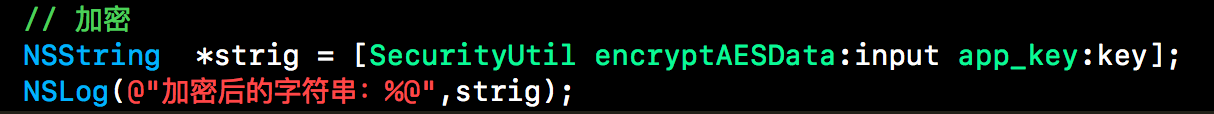
NSLog(@“%@“,data);

NSString \*string = [self encryptAESwithStr:data];

NSLog(@"%@",string);

将字符串@“123456” ，以密码@“1234567812345678”，进行加密，得到16进制NSData类型的数据。将得到的data数据转换成字符串。

打印结果如下，下边的就是加密后的字符串。

将方法整合进SecurityUtil后，只需要一句代码

二、解密

解密需要用到方法

//将带密码的data转成string

+(NSString\*)decryptAESData:(NSData\*)data app\_key:(NSString\*)key;

需要传入参数data和约定的密码key。

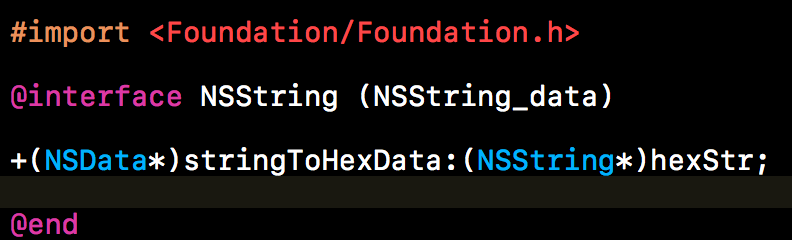
实际运用过程中，将加密及处理后的字符串 用dataUsingEncoding转换成NSData类型，发现data是10进制的，而不是16进制，造成解密失败。

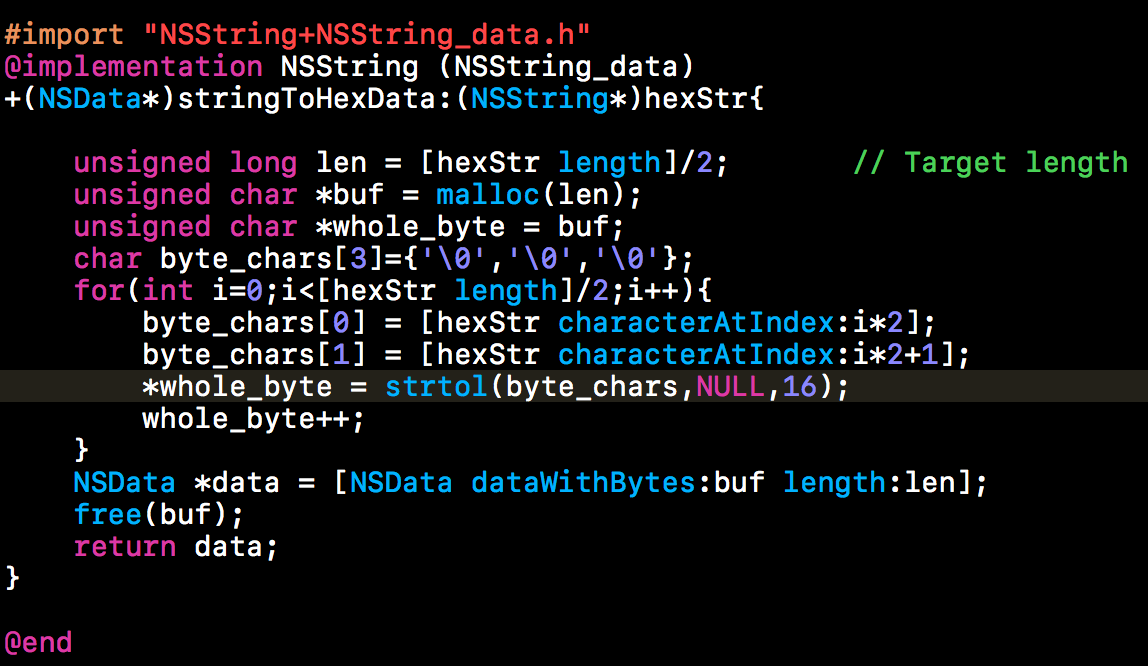
NSData \*lastData = [@"D9E0E2B1E622497EB6AA4FD64BF6439A" dataUsingEncoding:NSUTF8StringEncoding];

NSLog(@“%@“,lastData);

打印

需要将data转换16进制，添加NSString类的扩展类

.h 文件

.m 文件

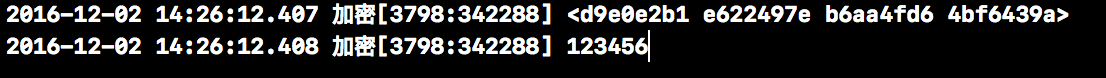
NSMutableData\*mdata=[[NSMutableData alloc]init];

[mdata appendData:[NSString stringToHexData:@"D9E0E2B1E622497EB6AA4FD64BF6439A"]];

NSLog(@"%@",mdata);

NSString \*string = [SecurityUtil decryptAESData:mdata app\_key:@"1234567812345678"];

NSLog(@“%@",string);

打印结果

完成解密。