# 电子交易的双重签名及认证

银行交易与我们的生活息息相关，为了更好的理解网络安全协议的功能和作用，本次实验选择了题目电子交易的双重签名及认证，实现了买家、商家、银行三者之间的订单信息确认和安全的信息传输。实现了三个实体的基本功能：

1. 买家要能够按照指定格式向商家发送订单信息、向银行发送转账信息（格式确定），具体信息应该采用加密或摘要求解的方式保证安全性；
2. 同时，买家还需生成订单与转账信息的双重签名，并将其由商家发送给商家和银行；商家在收到信息后，展示订单信息，并对双重签名的可信性进行检查；
3. 银行在收到信息后，展示转账信息，并对双重签名的可信性进行检查。这里，商家和银行除了自身检查双重给签名外，还需要相互通信以确保收到的双重签名一致。

本次实验的实现分为三个模块：客户模块，卖家模块，银行模块。其中，客户生成订单信息、支付信息，处理后发送给卖家；卖家得到订单信息，对整体信息摘要进行比对，保证数据未被篡改，将信息发送给银行，并等待银行返回的匹配信息；银行得到支付信息，对整体信息摘要及双重签名进行对比，保证数据未被篡改，对比成功后向卖家发送匹配信息。

在客户模块中，客户需要生成一个AES对称加密的密钥，以及一对RSA密钥对，并接收银行的公钥。其主要负责接受用户输入的付款信息PI和订单信息OI，之后分别利用sha256求取摘要得到PIMD和OIMD，连接PIMD和OIMD后对连接后的结果求取摘要得到POMD，再通过客户的私钥进行签名得到DS，最后向卖家Merchant发送E(Ks, PI | DS | OIMD) | E(KUb, Ks) | PIMD | OI | DS | KUc组成的信息，完成任务。主要函数接口为gen\_info()，它用来生成格式化信息，接受PI，OI，KRc，KUc以及KUb等参数，并返回格式化之后的信息。实现功能如下图1.1，首先生成付款信息、订单信息PI、OI，求付款信息、订单信息的摘要PIMD、OIMD，然后合并求二次摘要POMD，加密得到双重签名DS，最后将格式化的信息发给卖家Merchant。

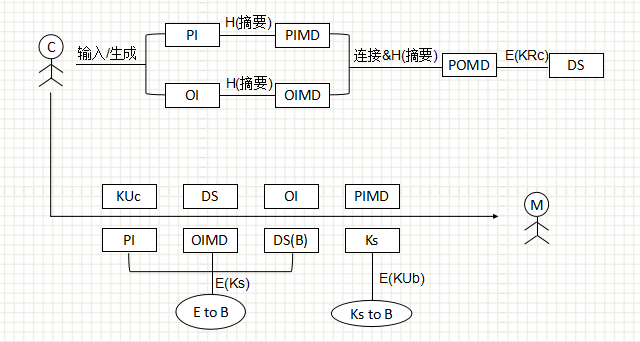


图1.1

在卖家模块中，卖家接受来自用户发送的格式化信息，并解包出需要自己解析的数据，通过得到的OI计算得到摘要OIMD，并与接收到的PIMD计算得到POMD，与通过KUc解密DS出的POMD\_recv进行比对，比对成功后将原始信息中需要发送给银行Bank的信息发送至银行，并将DS和KUc添加在之后，方便银行进行比较，此时等待接收来自银行的匹配消息。其中有两个主要函数接口，一个是check\_data()，用来对比POMD并返回<对比结果,DS>的元组，参数为const string &data，返回值类型为tuple<bool, string>；另一个是process()，它根据对比结果，输出日志信息，参数为int fd，返回值类型为void。实现过程如下图1.2，首先将OI求摘要得到OIMD，与PIMD合并求得POMD，然后将DS解密得到POMD，最后对比两个POMD是否一致，若一致，将信息发给银行Bank。

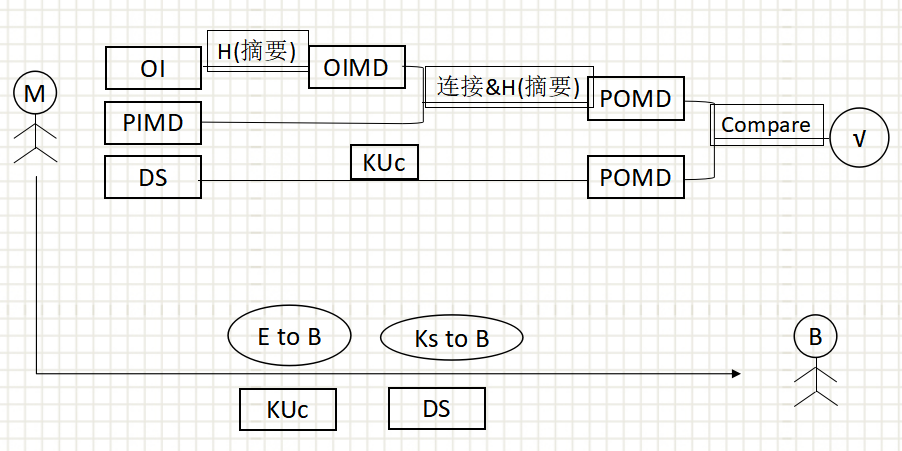


图1.2

在银行模块中，其工作过程如图1.2所示，银行接收由客户生成的AES密钥Ks加密的信息和KUb（银行公钥）加密的Ks的值以及卖家尾置的DS和KUc（客户公钥），其通过KRb（银行私钥）解密得到Ks，并通过Ks解密得到PI、OIMD以及DS(B)，通过计算PI的摘要得到PIMD，连接OIMD后计算摘要得到POMD(B)，与通过KUc验证DS得到的POMD相比对，若验证成功则继续比对DS(B)和卖家尾置的DS，比对成功则向卖家Merchant发送成功消息，否则向卖家Merchant发送失败消息。

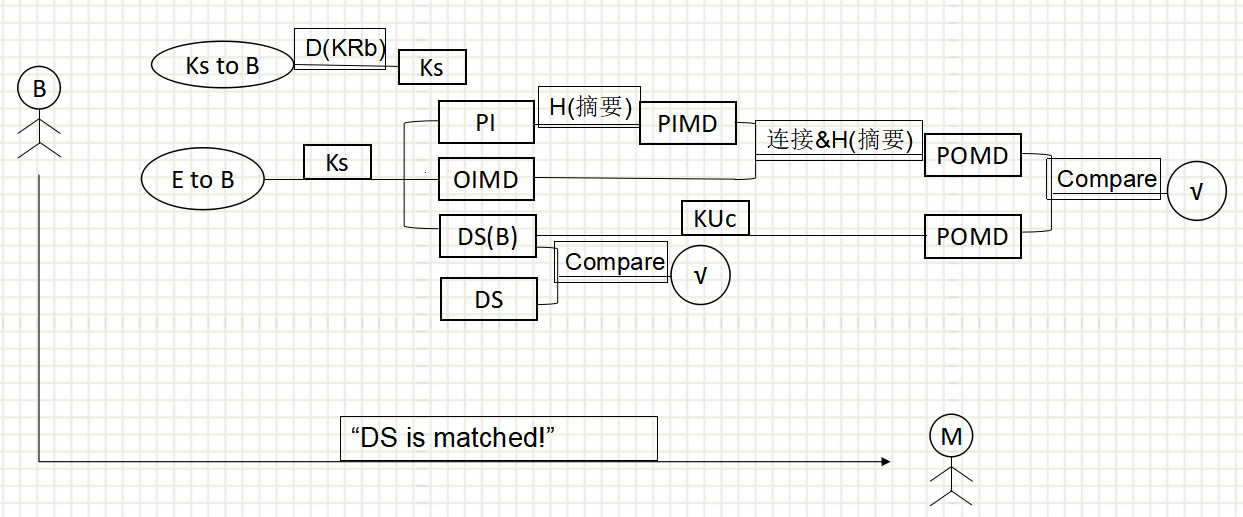


图1.3

实现过程采用的信息格式如图1.4所示

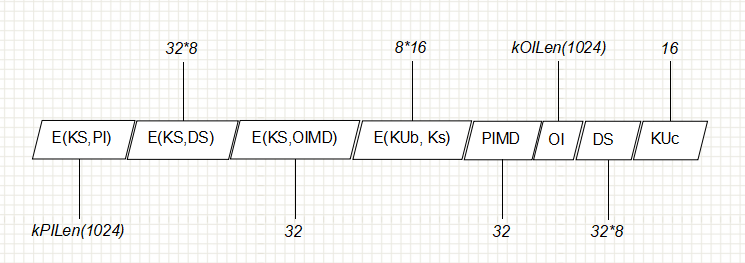


图1.4

在本次实验中主要的算法是SHA（摘要求解），RSA（非对称加密传递密钥和签名认证），AES（对称加密）。利用SHA验证数据的完整性，利用RSA传递密钥以及实现签名认证，利用AES对称加密数据，使得数据破解难度较高。