02比特币中的密码学原理

比特里被称为加密货币(crypto-currency)，但其实加密货币是不加密的，交易内容和地址都是公开的。

比特币主要运用密码学中的哈希和签名。

密码学中的哈希函数被称为cryptographic hash function

拥有两个重要的性质：

1. collision resistance或者叫作collision free：

x != y 但是H(x) = H(y)，其中H为哈希函数

一般来说哈希碰撞是不可以避免的，因为输入空间远远大于输出空间

实际上碰撞是客观存在的，而是指的没有高效的办法可以认为创造碰撞

（往往只能通过brute force，遍历所有输入的可能性）

但是如果输入的空间非常大，brute force是不可行的，因为需要算的东西太多了

collision resistance的存在是为了保证 没有办法能够让消息m经过篡改之后而不被识别出来

e.g. 假如说有一个消息m，H(m)被称之为digest，现在将消息m篡改为m'

则H(m')和H(m)不一样，从而发现m被篡改过

我们没有高效的办法可以找到m*，使得 H(m*)=H(m)

注意：没有任何一个数学函数被证明是collision resistance，大部分都需要实践经验的检验

也有一些我们认为是collision resistance的函数，但我们找到了认为制造碰撞的办法

2. Hiding（哈希函数的运算是单向的，不可逆的）

x -> H(x) 输出值没有泄露任何输入的内容

但是可以使用brute force，遍历所有的inputs看看谁的哈希值是H(x)，谁就是x

hiding成立的前提是inputs的space够大

上述的两个性质结合在一起可以实现digital commitment，有的时候也叫做digital equivalent of a sealed envelope

现实生活中的sealed envelope:

预测股市：第一天在电视台上公布预测的结果，第二天在收盘的时候查看结果（但是公布的信息可能会影响最后的结果，拉涨或者拉跌），但是如果不提前说，怎么能证明是提前预测的？ 所以可以先写在一个密封的信封里，交给公证机构，第二天收盘的时候拿出来查看。

在电子世界中：可以第一天把预测值x对应的哈希值H(x)公布出去，因为hiding性质的存在，公众没有办法通过H(x)反向得出预测值x。第二天需要公布结果的时候可以直接公布x，因为collision resistance的性质存在，没有办法修改预测值x，否则公众经过H计算之后会发现和头一天公布的H(x)的数值不匹配。

Hiding的成立前提是输入空间足够大，常用的方法可以讲x的后面拼接一个随机数称之为nonce，写作x||nonce.

3. 除了上述的两个性质，比特币中的哈希函数还需要第三个性质，puzzle friendly（光看x，没有办法predict H(x)）

挖矿：H(block header) <= target，puzzle friendly的性质保证挖矿的过程没有捷径，必须随机尝试不同的nonce去拼接block header

这保证找到一个符合的nonce需要经过大量的工作，但是找到一个符合的nonce之后，别人验证起来很容易，只需要算一次哈希来看看是不是小于等于target

difficult to solve but easy to verify

比特币中的哈希函数叫做SHA-256, SHA -> secure hash algorithm

上述的三个性质SHA-256全部满足

比特币开账户：没有银行的类似机构，比特币的用户自己来控制开户，是去中心化的

只需要自己创建一个公私钥对（public key, private key）

这个概念来源于非对称的加密体系，asymmetric encryption algorithm

两个之间想要传输信息：

Symmetric：我们两个人可以商量好一个encryption key，加密和解密的过程中需要使用同一个密钥，需要有一种安全的方式可以把密钥传输给双方。

Asymmetric：用公钥进行encrypt，用私钥匙进行解密。当需要传输信息的时候，可以把自己的公钥传给对方，然后对方用你的公钥进行加密之后传回给你，然后你用自己本地上保存的私钥进行解密。

比特币系统中，只需要在本地产生一个公私钥对。公钥是银行账户，私钥是银行密码。别人给你转比特币的时候只需要知道你的公钥。

但是我们上面提到比特币的交易是不加密的，所以在交易的过程中我们需要用到签名。当你给别人的账户转比特币的时候，需要用你的私钥进行签名（防止别人冒充顶替），然后别人用你发布的公钥进行验证是否是你的签名。

如果两个人在本地生成的公私钥对恰好相同？怎么办？

可以大量的生成公私钥对，然后在区块链上寻找是否有公钥相同的情况，如果找到了相同的公钥，就可以用对应的私钥和解密。但是这种方法并不可行，比地球爆炸的可能性还低。目前还没有通过这种方法攻击成功的先例。

a good source of randomness 需要一个好的随机源

不仅生成公私钥对的时候需要有好的随机源，每一次签名的时候也需要有好的随机源。如果有一次签名的随机源不够好，就有可能泄露私钥，从而导致整个比特币系统的崩塌。

两个功能，哈希和签名。在比特币系统中，一般都是先对一个消息message取哈希，然后对这个哈希值进行签名。