第八讲 冲突与双花攻击

前言

通过上一讲，我们知道了记账人通过挖矿构造区块链，也就是工作量证明，来赢得普通用户的信任；也知道了需要通过激励机制来鼓励普通用户参与记账。

可是一旦有了奖励机制，就可能会有很多人一起抢着记账，他们记的账本不一样怎么办？奖励又该给谁呢？这就是冲突问题。

冲突问题

很简单，只要所有用户坚持一条原则来选择所信任的账本即可。

这条原则就是：**信任最长的区块链。**

普通用户行为

我们来看看，在这条原则下，普通用户和矿工的具体行为和产生的效果是怎样的。

首先，对于普通用户A来说，他并没有记账，所以在他那里并没有账本的记录。

那么当A需要查账时，比如查询与自己账户有关的交易历史记录，他就需要向P2P网络的其它用户发起获取账本的请求。

此时，可能会有多个用户B、C、D响应他的请求，告知A自己所拥有区块链的长度，

A收到响应后，只会选择那个拥有最长区块链的用户的回应，从他那里获得区块链账本。

矿工行为

我们再来看看矿工的行为。

由于有了奖励机制，所以可能会有多个，甚至很多个矿工同时挖矿记账以求获得奖励。

而挖矿需要解决哈希值合法问题，需要花费巨大的工作量。例如，在比特币中，大约需要10分钟才能完成一次挖矿，创建一个区块。

如果你是一个矿工，你从P2P网络中得知当前最长区块链长度是1000，那么你会从第0个区块开始创建区块链吗？

当然不会，因为即便你以最快速度挖出了第0个区块，并在其中加入一条奖励自己50个比特币的交易记录，然后向全网公开。那么你所创建的区块链由于长度太短，不会得到其它用户的信任，也就相当于作废了。

所以，聪明的做法是，你从收到的那些还没有被记录到哪1000个区块中的交易信息中，选择若干交易，打包形成区块，计算Nonce值。当然，你肯定要把那笔奖励给自己的交易记录进去，而且为了获得更多奖励，你肯定要优先选择那些提供手续费的交易。一旦你计算出了Nonce值，你就应当立即向全网公布，你挖到了第1001个区块，这样才可能让全网认可这个长度为1001的区块链，从而认可你记录在第1001个区块中的给自己的奖励。

不仅你会这么做，所有其他矿工都会这么做。他们为了获得奖励，都会争着抢着，想以最快速度挖出第1001个区块，并以最快速度向全网公布。

如果你正在挖第1001个区块B时，突然收到别的矿工公布过来的信息，说他已经挖到了第1001个区块B'，你会怎么做？

是继续挖第1001个区块B吗？

如果你继续坚持，即便你最终挖出了第1001个区块B，别人可能已经在挖出第1002个区块了。那你的第1001个区块B，就形同作废了。那么你记录在你的第1001个区块B中的给自己的奖励也不会得到其他人的认可。

所以聪明的做法时，一旦听说别人挖出了下一个区块，就应该立即停止手头的工作，转而投入下下个区块的争夺中。

听我这么描述，你肯定已经感觉到了，

挖矿就是一群矿工之间的竞争。他们为了获得奖励，必须让自己挖出的区块成为当前区块链的的最顶端。

冲突问题的解决

搞清楚了在这条原则下普通用户和矿工的的行为，我们开始提出的两个问题就迎刃而解了。

第一，很多矿工一起记账，记出的账本不一致怎么办？

由于矿工会自发的从当前已知最长区块链开始记录下一个区块，所以P2P网络中流转的区块链的绝大部分都是一致的。为什么说是大部分一致，而不是全部一致呢？因为消息在P2P网络中的传播必然存在时延，所以可能存在多个矿工几乎同时挖出下一区块，而互相暂时不知道的情况，此时他们周围的矿工就会沿着各自的最长区块链，继续挖下一个区块。从而导致区块链在较短的时间内出现了顶端部分区块不一致的现象。这就叫区块链分叉。但是随着时间的推移，只有其中一支能够最终存活下来。

第二，很矿工一起竞争，奖励该给谁呢？

由于所有用户都只信任最长区块链，所有那些胜出的矿工，也就是能让自己挖出的区块称为最长区块链一部分的矿工记录下来的奖励才会被所有用户认可。这就相当于，奖励自然而然的给了每一场挖矿竞赛中胜出的矿工。

双花攻击

这套机制看起来已经很完美了可是在实际支付场景中还是可能遭受恶意攻击。攻击者的目的当然是为了，所以他们通常都是利用漏洞，将一份钱花两遍，也就是双花攻击。常见的双花攻击有以下三种。

Race Attack

按照区块链的记账机制，矿工需要完成工作量证明才能产生下一区块，而这需要花费一段时间。比如，比特币区块链大约每十分钟才会产生一个新的区块。所以一笔交易从创建，传播，到最终被记录到区块链上是会有时延的。而在实际支付场景中，买家创建支付交易后，未必会等到这笔交易记录到区块链后才交付商品，否则太影响消费体验。

攻击者正是利用这一漏洞，开始Race Attack。假设一杯咖啡需要10块钱，那么攻击者可以在支付时，同时创建两笔交易T和T'。在交易T中，从自己仅有11块钱的账户A转账到咖啡店老板账户B。而在交易T'中，攻击者从自己仅有11块钱的账户A转账10快去拿到自己的另一个账户A'，同时附加1块钱交易费。说白了，就是一女双嫁。请你想一想，如果这两笔交易同时向全网传播，那一笔会先被记录到区块链上呢?

没错，当然是T'，因为T'中有手续费，所以矿工会有限记录有手续费的交易。

当T'被记录下来后，账户A中就一分不剩了，当有矿工要记录交易T时，会发现该交易是不合法的，所以就不会记录这笔交易。

因此结果是，咖啡店老板交付了咖啡，他的账户却分文未入；攻击者虽然付出了1块钱的手续费，但获得了价值10美元的咖啡。

这个攻击原理，就像是让那两笔交易自己在P2P网络中赛跑，看谁能先进入区块链，所以这个攻击被命名为Race Attack。

Finney Attack

Finney Attack则更加直接一点，不用让两笔交易赛跑了。攻击者先创建交易T'转账给自己，然后通过挖矿将这笔交易记录在下一区块中，但是暂时不公布这个区块。此时，迅速创建交易T，出示给商家，获得咖啡后，迅速公布他挖出的区块。如此一来，交易T当然就变成了非法的交易了，自然也就不会被记录在区块链上了。攻击者同样没有支付任何费用，就获得了咖啡。

以上两种攻击利用的都是卖家还未确认交易T被记录到区块链上就交付商品的漏洞。

Majority Attack

Majority Attack，又称51%攻击。在51%攻击中，假设攻击者拥有了全网51%的算力，也就是说在每一轮挖矿的竞争中，他都有一半以上的概率获得下一个区块。这样一来，即便卖家确认交易T记录到了区块链上之后再交付商品，攻击者也可以动用51%算力，从交易T所在区块的前一个区块强行分叉，分叉出更长的链，从而使T所在的区块失效。

正因为如此，所以比特币建议大额交易需要在交易得到6个区块以上的确认之后才生效，因为一旦有6个区块以上确认就很那再分叉出更长的链了。

小结

巧妙的密码学哈希函数运用和简单的最长区块链信任原则，让一群用户建立起了对统一的账本的信任，公钥密码机制由能让任何用户在这个统一账本上拥有自己的资产账户。而且一切都无需中心化可信节点的介入，完全依靠所有用户的自发行为。用奇、妙来形容毫不为过。

还记得用户在怎么像别人证明自己对资产账户的所有权吗？没错是用自己的私钥生成交易信息的数字签名。也就是说，私钥是你打开资产账户的钥匙，一旦泄露给别人，别人就同样具备了支配这个账户的权力。你该怎样保护私钥的安全呢？下一单元，对称密码，为您讲述。