比特币地址有 1 打头地址 ，也有 3 打头的地址，这两者有什么区别吗？

在哪种情况下，地址上的比特币会被锁死？

到底是谁拥有比特币的控制权，是你？还是你的钱包？

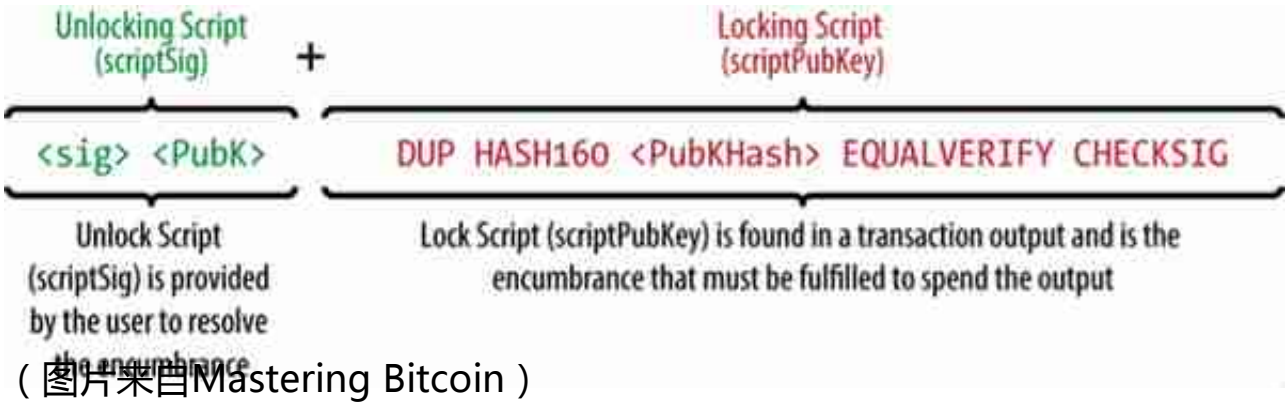
如果你在使用比特币钱包，但却无法回答上面三个问题，那么这篇文章是为你而写。

安比（SECBIT）实验室在对数字钱包源码审计时，发现一个名为 pywallet 的比特币钱包开源库包含了一个严重缺陷。如果向 pywallet 生成的 OmniLayer 收款地址转账，将导致资产永久丢失。

据安比（SECBIT）实验室区块链技术专家 zer0to0ne 解释，OmniLayer 协议允许在比特币区块链上发行自定义资产（比如 USDT）。OmniLayer 资产交易的本质是比特币交易。比特币交易的代码库有很多，pywallet 便是其中一种。它可以方便的构造符合 OmniLayer 格式的比特币交易。目前 pywallet 已经被应用在一些数字钱包软件中。

但是，开源库 pywallet 在生成 OmniLayer 钱包地址的时候，误将地址的前缀写反了，若干资产被锁死在无效的地址内！

下面是 pywallet 相关错误代码截图：



（图片来自Mastering Bitcoin）

脚本执行过程如图所示，Bob将交易签名后得到的数据（实际上还要包含数据长度信息），真正的scriptSig 应该为 ，比特币脚本执行器从 PUSH 数据开始，PUSH 操作会读取第一个字节获取将要入栈的数据长度信息，然后持续执行比特币脚本，

直到最后执行完毕检查执行结果。

首先入栈的是 ，然后将 入栈，一次DUP操作将在栈顶复制一份 ，HASH160弹出栈顶的 并计算Hash，将结果压回栈中，之后使用 EQUALVERIFY 弹出Hash对比是否和 相等，如果相等则返回True，不相等便标记交易为无效。执行到这一步，暴露了公钥，确保了签名者的身份的正确性，但是黑客或矿工可以通过暴露的公钥构造出一个新的交易替换原始交易，无法保证安全，那么便需要下一步来保证交易无法伪造。此时栈上还有 和 ，执行 CHECKSIG，将校验数字签名的正确性，确保了签名者拥有地址对应的私钥。

数字签名除了持有私钥的人，谁也无法伪造，执行至此，一笔比特币P2PKH交易已经安全地完成了。

再解释一遍：当 Bob 要花费 Alice 给他的比特币时，Bob 只有用正确的钥匙才能打开 Alice 留给他的保险箱，把钱放入 Bob 新构造的一个保险箱里。

这时候一些聪明的读者会注意到一个细节：如果 Bob 取出钥匙，在还未打开保险箱的时刻，区块链上的任何矿工都能看得见这把钥匙的形状，理论上他们是可以立即复制一把钥匙，把 Alice 留给 Bob 的保险箱打开并花掉（俗称 Front-running 攻击）。

真的可以这样做吗？显然中本聪考虑了这个问题，这把钥匙中的交易签名是 Bob 发起的交易的完整签名。假设 Bob 要将 Alice 构造的保险箱中的比特币 装入一个新的保险箱（留给Charlie），这时候 Bob 出示的钥匙包含了 Charlie 的公钥Hash，矿工虽然可以复制 Bob 的钥匙，但是这把钥匙已经隐藏了下一个新保险箱的关键信息，因此矿工无法使用这个复制钥匙来完成别的动作（无法挪用数字签名）。

P2SH——后中本聪时代的重大创新

中本聪设计了一个这么强大的脚本系统，只用来构造转账交易似乎太浪费了，我们试试用其他指令构造一些特别的锁定脚本，并使用其他方式来解锁。

例如我们可以构造一个用 Hash 原象（Pre-image）来解锁交易的脚本：

OP_HASH160

OP_EQUAL

这个脚本的含义是：当满足 Hash160(Pre-image)==这个条件时，便可成功将脚本解锁。

我们继续通过保险箱的例子来解释，并给这类保险箱起名为 3-类保险箱。现在 Alice 给 Bob 的比特币锁定在一个由上述 Hash160保护的保险箱里，我们姑且称之为哈希锁吧。

这把锁依然需要正确的形状才能开启，但是安全性却弱很多，缺少数字签名机制导致钥匙隐藏的关键信息不会随着Bob 新建的保险箱而变化。任何矿工都能在 Bob 亮出钥匙的一瞬间复制出一摸一样的钥匙，抢着去开Alice留给Bob的保险箱（Front-running），将币转给另一个人 Eve，于是原本属于 Bob 的比特币会被洗劫一空。

虽然这个脚本非常不安全，但是它却有两个非常神奇的功能：

1.交易构造的输出足够短，意味着比特币节点维护的 UTXO 缓存占用空间将会大大减小

2.Pre-image 总是在交易被花费时作为 input 来引用，不会在交易的 output 侧出现，UTXO依然保持精简，同时可以把手续费负担转嫁给接收方。

既然所述的输出脚本好处很多，那我们是否有办法让这种交易方式变得安全呢？这就需要讲讲什么是 P2SH了。

比特币核心开发者 Gavin Adresen 提出了一种叫做 Pay to Script Hash (P2SH) 的技术。

P2SH 的交易输出依然是判断 Hash160(Script)==