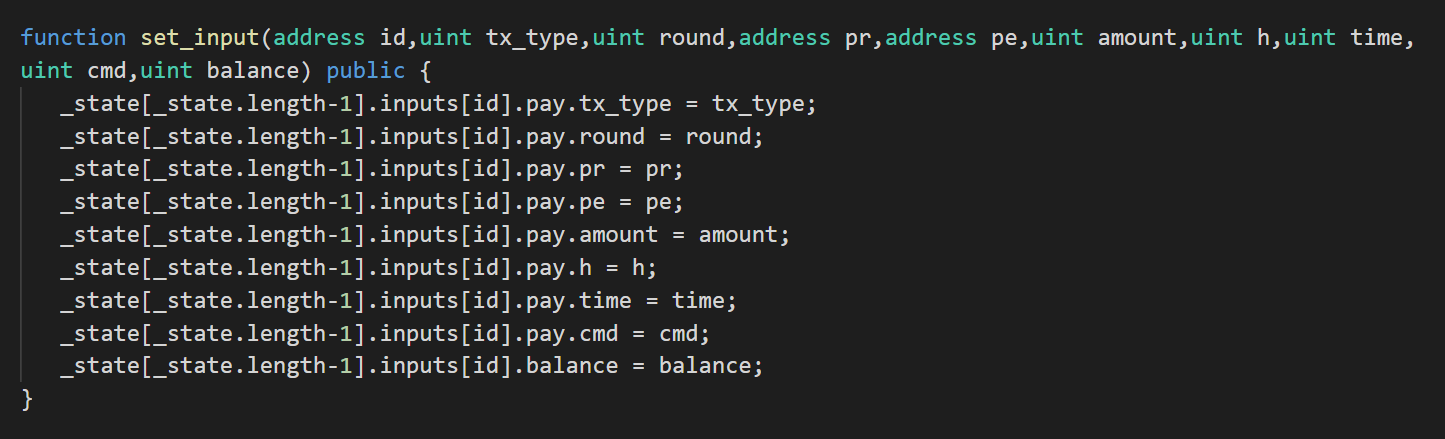
步骤：

1. 从txt文件中读取私钥，根据私钥计算出公钥和地址，上传至合约中。



【注意点：传入合约中的地址格式为python中的str类型，内容为0x开头的十六进制串。其中a~f有大小写区分，用于校验，否则无法传入合约，可用Web3.toChecksumAddress(str)进行转换。】

1. 从txt文件中读取初始状态信息，上传至合约中。





【由于solidity不支持传入类型为结构体，因此只能把结构体中的各个数据成员作为参数传入。但是由于state中有变长的数组，所以无法一次传入，需要分多次传入合约。】

1. 利用步骤1中的密钥，即可对数据进行签名。solidity中有ecrecover函数，可根据签名前的hash值和签名的值恢复出公钥，根据公钥是否相同判断签名是否正确。

存在的问题：

1. solidity的函数输入参数不支持自定义的结构体，部分需要上传state的函数无法定义。

方法1：输入结构体的功能在试验功能中可用（pragma experimental ABIEncoderV2;），但听说有bug，且python和solidity的结构不同，需要进一步探究如何传入。

方法2：分多次传入参数。这样需要提交多次，如果有不同的用户同时提交，会导致数据同步上的问题，且影响性能。

1. 对于Conpay集合，solidity并没有像python那样的集合（Set）类型，只能使用数组替代。但solidity中的数组仅内置了push和pop两个方法，针对的都是数组的最后一个元素。若想在中间插入和删除，需要自己维护数组结构。
2. 签名的验证。hash函数接收的是字符串类型，需要把结构体转化为字符串，如何转换？（solidity对字符串的支持也很有限，甚至不支持字符串的拼接等操作）