私链上合约的相关信息和函数

1. 智能合约5.0：

ABI：

[{\"constant\":true,\"inputs\":[{\"name\":\"\_username\",\"type\":\"string\"}],\"name\":\"fetchmsgpb\",\"outputs\":[{\"name\":\"\",\"type\":\"string\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"},{\"constant\":true,\"inputs\":[{\"name\":\"\_hash\",\"type\":\"string\"}],\"name\":\"fetchmsgpv\",\"outputs\":[{\"name\":\"\",\"type\":\"string\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"},{\"constant\":true,\"inputs\":[{\"name\":\"\_hash\",\"type\":\"string\"}],\"name\":\"fetchhyperpv\",\"outputs\":[{\"name\":\"\",\"type\":\"int256\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"},{\"constant\":false,\"inputs\":[{\"name\":\"\_hash\",\"type\":\"string\"},{\"name\":\"\_filepb\",\"type\":\"string\"},{\"name\":\"\_filepv\",\"type\":\"string\"},{\"name\":\"\_hyperbolapb\",\"type\":\"string\"},{\"name\":\"\_hyperbolapv\",\"type\":\"int256\"}],\"name\":\"newsender\",\"outputs\":[{\"name\":\"\",\"type\":\"bool\"}],\"payable\":false,\"stateMutability\":\"nonpayable\",\"type\":\"function\"},{\"constant\":false,\"inputs\":[{\"name\":\"\_username\",\"type\":\"string\"},{\"name\":\"\_hyperbolapb\",\"type\":\"string\"},{\"name\":\"\_filepb\",\"type\":\"string\"}],\"name\":\"newreceiver\",\"outputs\":[{\"name\":\"\",\"type\":\"bool\"}],\"payable\":false,\"stateMutability\":\"nonpayable\",\"type\":\"function\"},{\"constant\":true,\"inputs\":[{\"name\":\"\_username\",\"type\":\"string\"}],\"name\":\"fetchhyperpb\",\"outputs\":[{\"name\":\"\",\"type\":\"string\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"},{\"constant\":true,\"inputs\":[],\"name\":\"fetchfilenum\",\"outputs\":[{\"name\":\"\",\"type\":\"uint256\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"},{\"constant\":false,\"inputs\":[{\"name\":\"\_keyworda\",\"type\":\"string\"},{\"name\":\"\_keywordb\",\"type\":\"string\"},{\"name\":\"\_cipher\",\"type\":\"string\"}],\"name\":\"newfile\",\"outputs\":[{\"name\":\"\",\"type\":\"bool\"}],\"payable\":false,\"stateMutability\":\"nonpayable\",\"type\":\"function\"},{\"constant\":true,\"inputs\":[{\"name\":\"i\",\"type\":\"uint256\"}],\"name\":\"fetchkeyword\",\"outputs\":[{\"name\":\"\",\"type\":\"string\"},{\"name\":\"\",\"type\":\"string\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"},{\"constant\":true,\"inputs\":[{\"name\":\"i\",\"type\":\"uint256\"}],\"name\":\"fetchcipher\",\"outputs\":[{\"name\":\"\",\"type\":\"string\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"}]

合约的bytecode：

""

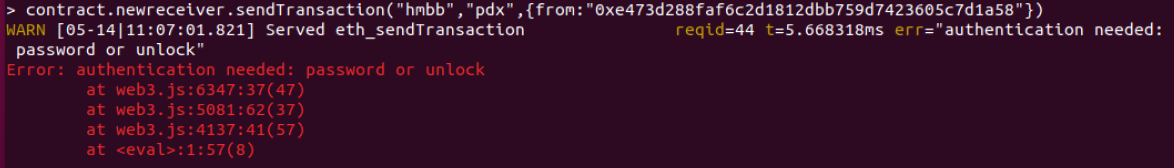
1. 部署好的合约在私链上的地址：

//自己应该有。

1. 解锁用户的命令：

personal.unlockAccount(账户地址)

这个命令在调用函数的时候时不时就要用到。



//不解锁就会报错

1. 重新启动geth时，调用原来部署好的智能合约

* myContract = eth.contract(abi) //abi在上面

如果失败了先转一下abi的格式：var abi=JSON.parse(‘文档中的abi’)，再myContract = eth.contract(abi)

* contract = myContract.at(address)

//contract.address="0xfb8fe9d6b9fde5d0383b04243bcb4f6649f2aec0"。原来保存的数据还可以接着使用。

1. 合约中的函数：

需要收取gas的函数调用要用sendTransaction(),返回的都为交易的hash。并且私类函数调用后保险起见要自己挖矿，保障它能被写入链上。

不需要收出gas的函数调用要用call(),返回的为函数的返回值。

1. newsender(string \_hash, string \_filepb, string \_filepv,string \_hyperbolapb, int \_hyperbolapv)

说明：第一个参数为用户账号和密码的hash，第二个参数为此用户文件加密的公钥，第二个参数为此用户文件加密的私钥，第三个参数为此用户双曲线公钥，第四个参数为此用户双曲线私钥。返回值为此交易的hash。

作用：通过调用此函数来存储用户的公钥私钥信息到链上，此函数需要收取gas。

调用：

contract.newsender.sendTransaction("tmf123456","1","2","3","0x45",{from:"0xe473d288faf6c2d1812dbb759d7423605c7d1a58"})

测试：



1. fetchhyperpv(string \_hash)

说明：函数参数为用户密码与用户名的hash。返回值为int类型为对应此用户的双曲线私钥。

作用：调用此函数来获取用户的双曲线私钥。此函数不收取gas。

调用：contract.fetchhyperpv.call("tmf123456")

测试：



1. fetchmsgpv(string \_hash)

说明：函数参数为用户密码与用户名的hash。返回值为string类型为对应此用户的文件加密私钥。

作用：调用此函数来获取用户的双曲线私钥。此函数不收取gas。

调用：contract.fetchmsgpv.call("tmf123456")



1. newreceiver(string \_username, string \_hyperbolapb, string \_filepb)

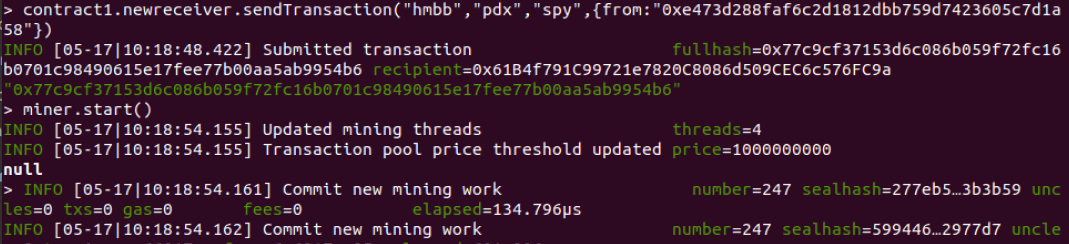
说明：函数第一个参数为用户的用户名，第二个参数为用户的双曲线公钥，第三个参数为用户的文件加密公钥。返回值为交易的hush。

作用：调用此函数来把用户的公钥信息存储到链上，此函数收取gas。

调用：

contract.newreceiver.sendTransaction("hmbb","pdx",”spy”,{from:"0xe473d288faf6c2d1812dbb759d7423605c7d1a58"})

测试：



1. fetchhyperpb(string \_username)

说明：函数参数为用户的用户名，返回值为string类型为用户对应的双曲线公钥。

作用：调用此函数来获取用户的公钥。此函数不收取gas。

调用：contract.fetchhyperpb.call("hmbb")

测试：



1. fetchmsgpb(string \_username)

说明：函数参数为用户的用户名，返回值为string类型为用户对应的文件加密公钥。

作用：调用此函数来获取用户的公钥。此函数不收取gas。

调用：contract.fetchmsgpb.call("hmbb")

测试：



1. newfile(string \_keyworda, string \_keywordb, string \_cipher)

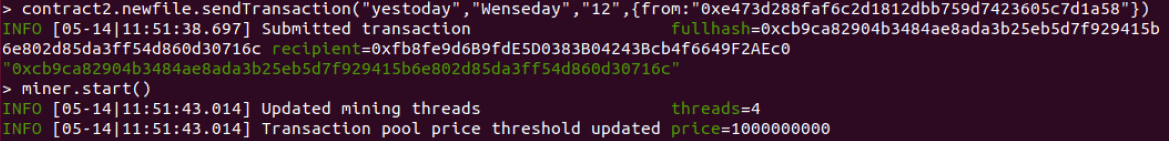
说明：函数第一个参数为加密关键词A，第二个参数为加密关键词B，第三个参数为此加密关键词对应的密文。返回值为交易的hush。

作用：调用此函数来在链上存储用户的加密关键词以及对应的密文。此函数收取gas。

调用：

contract.newfile.sendTransaction("yestoday","Wenseday","12",{from:"0xe473d288faf6c2d1812dbb759d7423605c7d1a58"})

测试：



1. fetchfilenum()

说明：函数不用传参，返回值为uint类型为当前存储的加密文件的个数。

作用：调用此函数来查看当前在链上存的加密文件的个数。此函数不收取gas。

调用：contract.fetchfilenum.call()

测试：



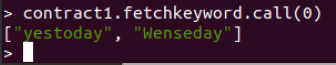
1. fetchkeyword(uint i)

说明：函数参数为正整数i，i为小于filenum的值。返回值为两个string类型为对应i的两个加密关键字A和B。

作用：检索时调用此函数来获得加密关键词。此函数不收取gas。

调用：contract.fetchkeyword.call(0)

测试：返回类型的格式如下所示。



1. fetchcipher(uint i)

说明：函数参数为i。返回值为string是索引对应的密文。

作用：调用此函数来获得对应的密文。此函数不收取gas。

调用：contract.fetchcipher.call(0)

测试：

