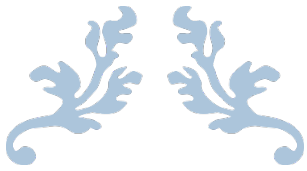


DCRM API test

2019-3-29

contents

1、Installation and Compile

1.1 Install Requirements

a) Recommended use Ubuntu 18.04 to build the *dcrm-ec2* source code.

b) Requires Git, Go and C compilers to be installed:

sudo apt-get install -y build-essential golang git

c) Requires bison, flex and gmp to be installed:

sudo apt-get install -y bison flex libgmp-dev

d) Recommended go version 1.9 or higher.

1.2 Set GOPATH environment

a) Setup a go folder: *mkdir -p ~/dcrm; echo "export GOPATH=$HOME/dcrm" >> ~/.bashrc*

b) Update your path: *echo "export PATH=$PATH:$GOPATH/bin" >> ~/.bashrc*

c) Read the environment variables into current session: *source ~/.bashrc*

1.3 Compile

a) Get source code, clone the repository to a directory of your choosing.

git clone https://github.com/FUSIONFoundation/dcrm-ec2.git

b) Building bootnode and gdcrm.

cd dcrm-ec2; make all

2、Deploy

2.1 Start-up bootnode

a) Make a bootnode.key

cd build/bin; ./bootnode -genkey bootnode.key

b) Start-up bootnode

./bootnode --nodekey bootnode.key --addr :9171 --verbosity 9

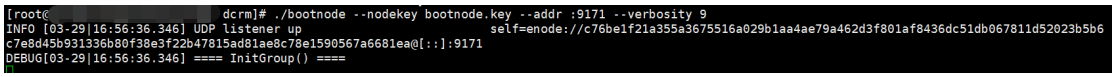
Return:

INFO [03-29|16:56:36.346] UDP listener up self=enode://c76be1f21a355a3675516a029b1a

a4ae79a462d3f801af8436dc51db067811d52023b5b6c7e8d45b931336b80f38e3f22b47815ad81ae8c78e1590567a6681ea

@[::]:9171

DEBUG[03-29|16:56:36.346] ==== InitGroup() ====



2.2 Start-up node

a) You must start three nodes to form a group. The bootnodes is from what you get in Chapter 2.1 b).

node1: *./gdcrm --datadir nodetest1 --port 1111 --syncmode full --bo*

otnodes enode://c76be1f21a355a3675516a029b1aa4ae79a462d3f801af843

6dc51db067811d52023b5b6c7e8d45b931336b80f38e3f22b47815ad81ae8c

78e1590567a6681ea@127.0.0.1:9171 --ipcpath geth1.ipc --rpccorsdomai

n "127.0.0.1" --networkid 40599 --mine console

node2: *./gdcrm --datadir nodetest2 --port 2222 --syncmode full --bo*

otnodes enode://c76be1f21a355a3675516a029b1aa4ae79a462d3f801af843

6dc51db067811d52023b5b6c7e8d45b931336b80f38e3f22b47815ad81ae8c78e1590567a6681ea@127.0.0.1:9171 --ipcpath geth2.ipc --rpccorsdomain "127.0.0.1" --networkid 40599 console

node3: *./gdcrm --datadir nodetest3 --port 3333 --syncmode full --bootnodes enode://c76be1f21a355a3675516a029b1aa4ae79a462d3f801af8436dc51db067811d52023b5b6c7e8d45b931336b80f38e3f22b47815ad81ae8c78e1590567a6681ea@127.0.0.1:9171 --ipcpath geth3.ipc --rpccorsdomain "127.0.0.1" --networkid 40599 console*

Tips: The red parts are unfixed, you can change according to actual situation.

b) Start a fourth node to connect the Group of nodes.

node4: ./gdcrm --datadir nodetest4 --port 4490 --syncmode full --bootnodes enode://c76be1f21a355a3675516a029b1aa4ae79a462d3f801af8436dc51db067811d52023b5b6c7e8d45b931336b80f38e3f22b47815ad81ae8c78e1590567a6681ea@127.0.0.1:9171 --ipcpath geth4.ipc --rpccorsdomain "127.0.0.1" --networkid 40599 console

3、LILO

Premise

Use the command of "*eth.coinbase*", checking that you have a FUSION address. If you don't have, use the command "*personal.newAccount*" to get one.

If return is “Error: authentication needed: password or unlock”, use the command “*personal.unlockAccount*” to unlock your account.

Use FUSION address “*0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2*” as an example.

e.g.

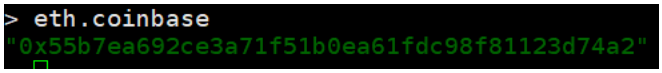
Request:

> eth.coinbase

Return fusion address:

"0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2"

See figure below:



3.1、lilo.dcrmReqAddr

Request a DCRM-address in block. You can request the DCRM-address with FUSION address and the coin type you want. Let’s take ETH DCRM-address as an example. Use the command of “*lilo.dcrmReqAddr*”.

e.g.

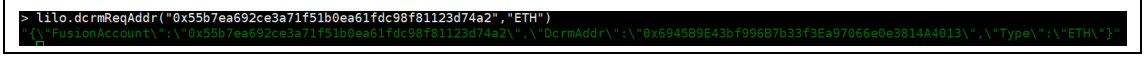
Request:

>lilo.dcrmReqAddr("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2","ETH")

Return:

"{\"FusionAccount\":\"0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2\",\"DcrmAddr\":\"0xb28fD54E02aaECc7289B9c99677640B72Fa08A10\",\"Type\":\"ETh\"}"

See figure below:



3.2、lilo.dcrmConfirmAddr

Confirm ETH DCRM-address store into block.

You need to use the return of *"DcrmAddr"* what you get in

"lilo.dcrmReqAddr" , to confirm the ETH DCRM-address. Use the command of

"*lilo.dcrmConfirmAddr*" .

e.g.

Request:

> lilo.dcrmConfirmAddr("0x6945B9E43bf996B7b33f3Ea97066e0e3814A4013","ETH")

Return TXhash:

"0x1d3cfce09cad6958a2e1435f7350fd608983a2dffe6fe6e5f5eeb85fa110bf20"

See figure below:

```
> lilo.dcrmConfirmAddr("0x6945B9E43bf996B7b33f3Ea97066e0e3814A4013","ETH")
Error: authentication needed: password or unlock
    at web3.js:3143:20
    at web3.js:6347:15
    at web3.js:5081:36
    at <anonymous>:1:1

> personal.unlockAccount(eth.coinbase,"block")
true
> lilo.dcrmConfirmAddr("0x6945B9E43bf996B7b33f3Ea97066e0e3814A4013","ETH")
"0x1d3cfce09cad6958a2e1435f7350fd608983a2dffe6fe6e5f5eeb85fa110bf20"
```

3.3、lilo.dcrmGetAddr

Get ETH DCRM-address from block.

You can get the ETH DCRM-address with the FUSION address and "ETH" . Use

the command of "*lilo.dcrmGetAddr*" .

e.g.

Request:

> lilo.dcrmGetAddr("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2","ETH")

Return TXhash:

"0x6945B9E43bf996B7b33f3Ea97066e0e3814A4013"

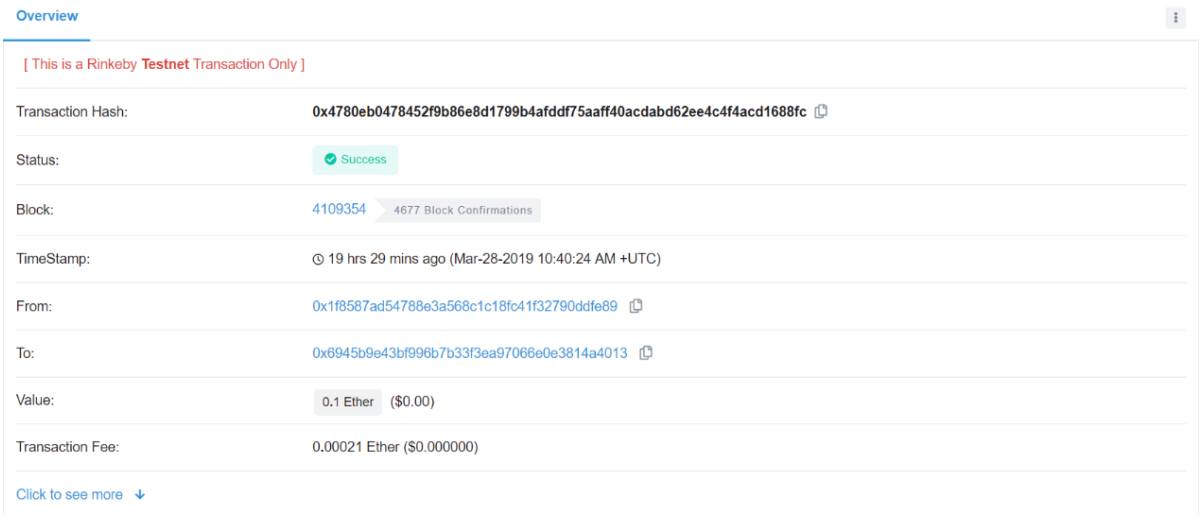
See figure below:

```
> lilo.dcrmGetAddr("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2","ETH")
"0x6945B9E43bf996B7b33f3Ea97066e0e3814A4013"
```

3.4、lilo.dcrmLockin

3.4.1 ETH Transaction

Use another ether address transfer some eth to your ETH DCRM-address. Make sure you can get the "Transaction Hash" and the "Value" of the transaction. Search the transaction in https://rinkeby.etherscan.io/.



3.4.2 Lockin

You can transfer to your ETH DCRM-address with the "Transaction Hash" and the "Value" what you get in *chapter 4.1*. Use the command of "*lilo.dcrmLockin*".

Tips: The "Value" must be convert to *Wei*.

e.g.

Request:

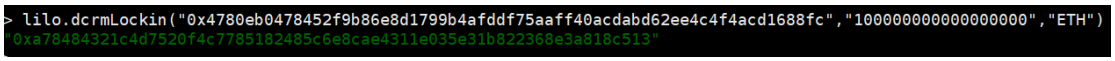
> lilo.dcrmLockin("0x4780eb0478452f9b86e8d1799b4afddf75aaff40acdabd62ee4c4f4a

cd1688fc","100000000000000000","ETH")

Return:

"0xa78484321c4d7520f4c7785182485c6e8cae4311e035e31b822368e3a818c513"

See figure below:



3.5、lilo.dcrmGetBalance

Get account balance from block.

You can query balance of the ETH DCRM-address with your FUSION address.

Use the command of "*lilo.dcrmGetBalance*".

e.g.

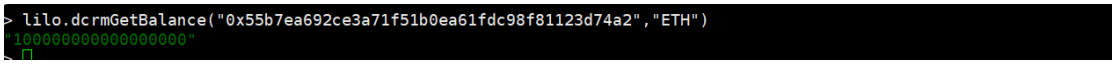
Request:

> lilo.dcrmGetBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2","ETH")

Return balance:

"100000000000000000"

See figure below:



3.6、lilo.dcrmSendTransaction

3.6.1 Fail dcrmTransaction

Send some ETH to another FUSION address. The transfer will fail, Because the fee for the transaction is more than what you have. Use the command of "*lilo.dcrmSendTransaction*".

Tips: Must use the FUSION address as target address and the fee for transfer is from the FUSION address.

e.g.

Request:

> lilo.dcrmSendTransaction("0x4c162012c26fa0a98a3ab6080a330680e8a83b3a","10000000000000000","ETH")

Return Error:

Error: value is great than gfsn balance.

at web3.js:3143:20

at web3.js:6347:15

at web3.js:5081:36

at <anonymous>:1:1

See figure below:

```
> eth.getBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2")
0
> lilo.dcrmSendTransaction("0x4c162012c26fa0a98a3ab6080a330680e8a83b3a","10000000000000000","ETH")
Error: authentication needed: password or unlock
    at web3.js:3143:20
    at web3.js:6347:15
    at web3.js:5081:36
    at <anonymous>:1:1

> personal.unlockAccount(eth.coinbase,"block")
true
> lilo.dcrmSendTransaction("0x4c162012c26fa0a98a3ab6080a330680e8a83b3a","10000000000000000","ETH")
Error: value is great than gfsn balance.
    at web3.js:3143:20
    at web3.js:6347:15
    at web3.js:5081:36
    at <anonymous>:1:1
```

3.6.2 Transfer to another FUSION address

You can transfer some ETH to another FUSION address.

```
> eth.coinbase
"0x4c162012c26fa0a98a3ab6080a330680e8a83b3a"
```

Transfer amount is deducted from ETH DCRM-address and the fee is from FUSION address.

e.g.

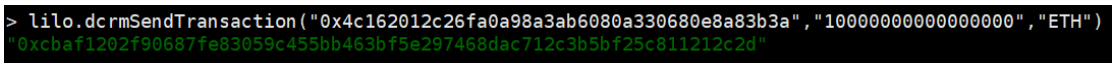
Request:

> lilo.dcrmSendTransaction("0x4c162012c26fa0a98a3ab6080a330680e8a83b3a","10000000000000000","ETH")

Return TXhash:

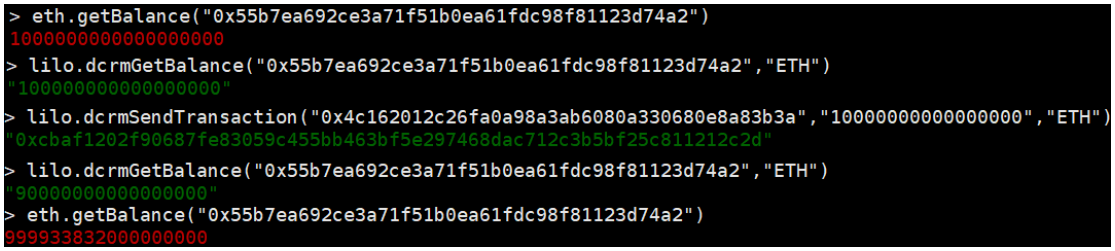
"0xcbaf1202f90687fe83059c455bb463bf5e297468dac712c3b5bf25c811212c2d"

See figure below:

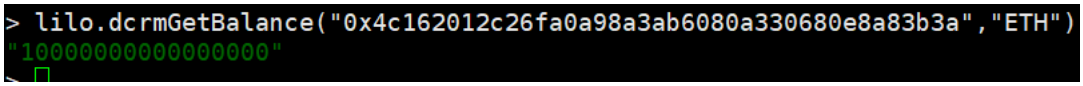


3.6.3 Checking transfer

After transfer, the ETH DCRM-address is deducted by how much you transfer, the FUSION address is deducted by the fee.



Another FUSION account:



3.7、lilo.dcrmLockout

3.7.1 Lockout

Lockout eth to another ether address. If you want to lockout, the fee is deducted from both ETH DCRM-address and FUSION address. For the ETH DCRM-address, the fee is fixed, 0.01 eth each time. For the FUSION address, the fee is unfixed costs.

e.g.

Request:

> lilo.dcrmLockout("0xf471D2b707A53fa27FfeA16739c374EC55B92476","10000000000000000","ETH")

Return TXhash:

"0x72aca19b4de07dfe72219030441195122b5a77271d66914dddcc76f49e748ca2"

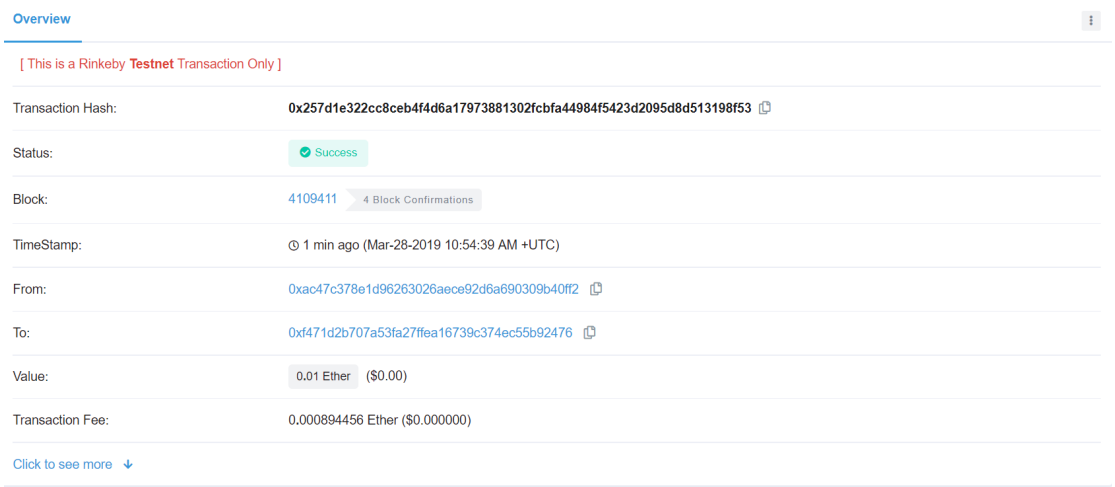
See figure below:

```
> lilo.dcrmGetBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2","ETH")
"90000000000000000"
>
> eth.getBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2")
999933832000000000
>
>
> lilo.dcrmLockout("0xf471D2b707A53fa27FfeA16739c374EC55B92476","10000000000000000","ETH")
Error: authentication needed: password or unlock
    at web3.js:3143:20
    at web3.js:6347:15
    at web3.js:5081:36
    at <anonymous>:1:1

> personal.unlockAccount("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2","block")
true
> lilo.dcrmLockout("0xf471D2b707A53fa27FfeA16739c374EC55B92476","10000000000000000","ETH")
"0x72aca19b4de07dfe72219030441195122b5a77271d66914dddcc76f49e748ca2"
>
> eth.getBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2")
999867936000000000
> lilo.dcrmGetBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2","ETH")
"70000000000000000"
```

3.7.2 check

Check the transaction in https://rinkeby.etherscan.io/.



appendix

A complete operation of the LILO is shown in the figure below.

```
> eth.getBalance(eth.coinbase)
0
> eth.coinbase
"0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2"
> lilo.dcrmReqAddr("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2","ETH")
"{\"FusionAccount\":\"0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2\",\"DcrmAddr\":\"0x6945B9E43bf996B7b33f3Ea97066e0e3814A4013\",\"Type\":\"ETH\"}"
> lilo.dcrmConfirmAddr("0x6945B9E43bf996B7b33f3Ea97066e0e3814A4013","ETH")
Error: authentication needed: password or unlock
    at web3.js:3143:20
    at web3.js:6347:15
    at web3.js:5081:36
    at <anonymous>:1:1

> personal.unlockAccount(eth.coinbase,"block")
true
> lilo.dcrmConfirmAddr("0x6945B9E43bf996B7b33f3Ea97066e0e3814A4013","ETH")
"0x1d3cfce09cad6958a2e1435f7350fd608983a2dffe6fe6e5f5eeb85fa110bf20"
> lilo.dcrmGetAddr("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2","ETH")
"0x6945B9E43bf996B7b33f3Ea97066e0e3814A4013"
> lilo.dcrmGetBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2","ETH")
"0"
> lilo.dcrmLockin("0x4780eb0478452f9b86e8d1799b4afddf75aaff40acdabd62ee4c4f4acd1688fc","100000000000000000","ETH")
"0xa78484321c4d7520f4c7785182485c6e8cae4311e035e31b822368e3a818c513"
> lilo.dcrmGetBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2","ETH")
"100000000000000000"
> eth.getBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2")
0
> lilo.dcrmSendTransaction("0x4c162012c26fa0a98a3ab6080a330680e8a83b3a","10000000000000000","ETH")
Error: value is great than gfsn balance.
    at web3.js:3143:20
    at web3.js:6347:15
    at web3.js:5081:36
    at <anonymous>:1:1

> eth.getBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2")
0
>
>
>
> eth.getBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2")
1000000000000000000
> lilo.dcrmGetBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2","ETH")
"100000000000000000"
> lilo.dcrmSendTransaction("0x4c162012c26fa0a98a3ab6080a330680e8a83b3a","10000000000000000","ETH")
"0xcbaf1202f90687fe83059c455bb463bf5e297468dac712c3b5bf25c811212c2d"
> lilo.dcrmGetBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2","ETH")
"90000000000000000"
> eth.getBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2")
999933832000000000
> lilo.dcrmLockout("0xf471D2b707A53fa27FfeA16739c374EC55B92476","10000000000000000","ETH")
"0x72aca19b4de07dfe72219030441195122b5a77271d66914dddcc76f49e748ca2"
> eth.getBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2")
999867936000000000
> lilo.dcrmGetBalance("0x55b7ea692ce3a71f51b0ea61fdc98f81123d74a2","ETH")
"70000000000000000"
```