区块链防重放攻击白皮书

一、前言介绍

本次区块链防重放攻击课题来源于 DACA 协会举办的清华大学 iCenter“区块链技术公开课(一期)“课题作业，其目的是解决即将发生的比特币分叉后可能发生的重放攻击问题,为比特币社区提供一个可行的解决方案，代表比特币中国社区向全世界发声。

1、什么是重放攻击

重放攻击(Replay Attacks)又称重播攻击、回放攻击，是指攻击者发送一个目的主机已接收过的包，来达到欺骗系统的目的，主要用于身份认证过程，破坏认证的正确性。重放攻击可以由发起者，也可以由拦截并重发该数据的敌方进行。攻击者利用网络监听或者其他方式盗取认证凭据，之后再把它重新发给认证服务器。重放攻击在任何网络通过程中都可能发生，是计算机世界黑客常用的攻击方式之一。

2、区块链重放攻击防止

本项目主要模拟区块链系统在硬分叉以后，如何防止原来区块链上的交易同步到新的区块链上。通过改变签名的算法，达到防止重放攻击的目的。 通过清华大学 icenter 老师们的悉心教导，通过将近 2 个月的学习，在比特币源码的基础上完成了防重放攻击 1.0 的开发。

3、防重放攻击开源代码分享&license

1) 防重放攻击开源代码分享

我们已经将防重放攻击的源码分享到 GitHub 上，地址为：https://github.com/btcgroup2/bitcoin

2) license

本项目按照 MIT license 协议. 欢迎大家多多交流 具体请参考:
https://opensource.org/licenses/MIT.

二、环境搭建

环境搭建主要分为三个主要的方面，首先是对比特币源码进行防重放攻击修改，其次是在服务器上对代码进行编译和部署，最后是采用 docker 技术实现多个比特币系统节点的模拟，目的是模拟真实情况下多个节点间的数据传输，更好的获得数据结果来进行对比，验证我们代码修改的正确性。

1、比特币源码防重放修改

1) 防重放攻击源码修改原理

我们这次解决比特币分叉防重放攻击的原理是修改比特币源代码的交易签名机制，这样比特币分叉后新链和旧链新产生的交易的是两个不同的签名机制，新链和旧链在验证交易数据的时候也会各自采用自己的机制进行验证，对方链签名的交易数据就验证失败，从而不接受对方链的交易数据，就达到了防止重放攻击的目的。

2) 防重放攻击源码修改详情

这次防重放攻击的核心是交易数据签名机制的改变，我们的修改方案是在原签名的基础上，再次取反，首先在源代码的 src/uint256.h 下的 28 行后添加如下代码：

```
void negate(){
        for(int i=0;i<WIDTH;i++){
            data[i]=~data[i];
        }
    }
```

其次是在源代码的 src/script/sign.cpp 下的 31 行后添加如下代码：

```
    hash.negate();
```

在源代码的 src/script/interpreter.cpp 下的 1265 行后添加如下代码：

```
    sighash.negate();
```

下面的两处修改是对上面修改的调用，代码详见：

https://github.com/btcgroup2/bitcoin/commit/47a52fea8efe79539248ad72157d86ca5a6cf310

3) 硬分叉源码修改原理

为了模拟真实环境中硬分叉的情况，我通过修改比特币源码中设置区块大小的值的大小，来模拟硬分叉。因为模拟环境产生大量的交易数据繁琐耗时，且意义不大，所以我们将源码中设置区块的大小由 1M 调整为 2k，这样就减少模拟交易数据的量，节省了时间。

4) 硬分叉源码修改源码

源码位置在：src/consensus/consensus.h，具体修改如下：

```
static const unsigned int MAX_BLOCK_BASE_SIZE = 1000000;
static const unsigned int MAX_BLOCK_BASE_SIZE = 2000；
```

用第二行去替代第一行，代码详见：

https://github.com/btcgroup2/bitcoin/commit/47a52fea8efe79539248ad72157d86ca5a6cf310

2、比特币源码环境搭建

1) 比特币源码编译准备命令

```
. /autogen.sh
./configure --with-incompatible-bdb
```

2) 比特币源码编译命令

```
make
```

4) 比特币源码安装命令

```
make install
```

5) 配置私有链网络环境

```
rpcuser=btc1
rpcpassword=xxl12345
addnode=60.205.162.88
addnode=47.94.165.9
whitelist=60.205.162.88
whitelist=47.94.165.9
```

6) 检测区块网络链连接是否正常

root@iZ2zegchn2k04hhikcn4yjZ:~# netstat -pant

Active Internet connections (servers and established)

Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name

cp 0 0 0.0.0.0:18444 0.0.0.0:* LISTEN 17569/bitcoind

tcp 0 0 127.0.0.1:18332 0.0.0.0:* LISTEN 17569/bitcoind

tcp 0 0 172.17.246.239:58241 60.205.162.88:18444 ESTABLISHED 17569/bitcoind

tcp6 0 0 :::18444 :::* LISTEN 17569/bitcoind

比特币源码安装参考 http://blog.csdn.net/rion_chen/article/details/51104727.

3、docker 运行环境技术使用

为了解决多节点的测试问题 ,采用当前流行的 docke 技术。著名的 Hyperledger 的 fabric 项目(由 IBM 主导的)就用了容器技术进行网络隔离。 本次开发虚拟化了 6 个节点(2 个 bitcoin 的源代码节点 , 2 个硬分叉节点和 2 个防攻击节点)

1) docker 镜像共享

docker pull xuxinlai2002/btcnew

docker pull xuxinlai2002/btchardfork

docker pull xuxinlai2002/btcorg

2) 查看容器

root@iZ2ze4wxzv9g5i5r69vu06Z:~/mydocker# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

da5c88d83f3e xuxinlai2002/btcnew "sh -c 'bitcoind -reg" 10 minutes ago Up 10 minutes 18444/tcp btcnewNode2

162954e0feac xuxinlai2002/btcnew "sh -c 'bitcoind -reg" 10 minutes ago Up 10 minutes 18444/tcp btcnewNode1

2907e68ad502 xuxinlai2002/btchardfork "sh -c 'bitcoind -reg" 10 minutes ago Up 10 minutes 18444/tcp btchardforkNode2

ab2ba80bff53 xuxinlai2002/btchardfork "sh -c 'bitcoind -reg" 10 minutes ago Up 10 minutes 18444/tcp btchardforkNode1

84955fb307e8 xuxinlai2002/btcorg "sh -c 'bitcoind -reg" 10 minutes ago Up 10 minutes 18444/tcp btcorgNode2

ed2d1223c275 xuxinlai2002/btcorg "sh -c 'bitcoind -reg" 10 minutes ago Up 10 minutes 18444/tcp btcorgNode1

3) 清空原有数据

root@iZ2ze4wxzv9g5i5r69vu06Z:~/mydocker# cleanAll

da5c88d83f3e

162954e0feac

2907e68ad502

ab2ba80bff53

84955fb307e8

ed2d1223c275

4) 配置 docker 环境

```
root@iZ2ze4wxzv9g5i5r69vu06Z:~/mydocker# setAll
9581b304488f2d2b042d5f39b038e37a8666625c46eae02e0019ba7689d9fa01
7c476428e7807267ab169f32c4e57f73bf39e6506aa71453954c2afe00c30857
b1323bd91291ca7b8cf7057b32c72029af17f04bf0036c3bc5868ba096fdb443
1d35b593c37281677d3723f51eb08e6b72b7cba2eb304223a0f822c8578e8a39
2124409a8df4a52eddd1c7acc2a38251e2cd0701d7ee4e4eb725a8902feed141
e74fa0d9398c7d7732dc1726d45445737ac7b0d6f2cb2010b775c65846a90c74
```

5) docker 环境重置

```
root@iZ2ze4wxzv9g5i5r69vu06Z:~/mydocker# resetAll
/root/mydocker
e74fa0d9398c
2124409a8df4
1d35b593c372
b1323bd91291
7c476428e780
9581b304488f
e2cbc3d893942329ccdeb514eb97d116838a297443681b5c5f3cb11730c8181d
d95f254c75686e086c6d45827438c346e5166434704e6e2f51dbd4c40462ba21
58df1c5e832aac2463f3d6a28e81865677eac6986fe9c91b8812893c3c84bf67
01ae29ac4832b87fed5d865e38c57704708f98fa170eac5b93ba4ac0a0379d65
076938e94aca1109dc690639f3775b74a6def6a6edda092b4d666ccfcbaf7e5f
272e228f0a94e474595e1f789d118624cfb1c86cb35ad70e1c215719acc4ba42
```

6) docker 执行自动化脚本目录

```
root@iZ2ze4wxzv9g5i5r69vu06Z:~/mydocker/tools# ll
drwxr-xr-x  2 root root 4096 Jul 10 21:42 ./
drwxr-xr-x 10 root root 4096 Jul 10 21:04 ../
-rwxrwxrwx  1 root root  328 Jul 10 16:43 cleanAll*
-rwxrwxrwx  1 root root   31 Jul 10 15:34 resetAll*
-rwxrwxrwx  1 root root  294 Jul 10 16:37 setAll*
-rwxrwxrwx  1 root root  705 Jul  9 23:28 startNodes*
```

三、防攻击验证测试

分布式系统测试是一项比较繁琐的工作，为了在有限的时间内完成繁琐复杂的测试， 我们开发了自动化测试脚本，并且在 docker 的宿主机环境里，测试通过。 这样可以提供测试效率，为以后的系统的持续集成打下基础

1、测试自动化脚本

1) docker 自动化脚本目录

```
root@iZ2ze4wxzv9g5i5r69vu06Z:~/mydocker/tests# ll
total 120
-rwxr-xr-x  1 root root  659 Jul 10 22:58 confirmtx*
-rwxr-xr-x  1 root root  336 Jul 10 21:40 generate*
```

```
-rwxr-xr-x    1 root root    556 Jul 10 20:24 getblockchaininfo*
-rwxrwxrwx    1 root root    775 Jul 10 21:43 getinfo*
-rwxr-xr-x    1 root root    898 Jul 10 22:28 transaction*
-rwxr-xr-x    1 root root     514 Jul 11 14:02 getblockhash*
-rwxr-xr-x    1 root root     503 Jul 11 14:04 getnodes*
```

2) 自动化测试脚本示例

我们挑选一个脚本进行展示，脚本 transaction*具体内容如下：

```
imgs=("btcorg" "btchardfork" "btcnew")
ctns=("1" "2")
declare ctnNum=0
#test instruction
logInfo="transaction : send teansaction from $1 to $2 amount $3 "
echo $logInfo | tee tests/log/$4.log
address=`docker exec -it $2 bitcoin-cli -regtest getnewaddress`
echo "address: $address" |tee -a tests/log/$4.log
txid=`docker exec -it $1 bitcoin-cli -regtest sendtoaddress $address $3`
echo "txid: $txid"|tee -a tests/log/$4.log
for img in "${imgs[@]}"
do
   for ctn in "${ctns[@]}"
   do
            logInfo="nodes: ${img}Node$ctn, transaction info"
            echo $logInfo | tee -a tests/log/$4.log
            docker exec -it ${img}Node$ctn bitcoin-cli -regtest gettransaction $txid > temp.con
            cat -v temp.con |tr -d "^M" |tee -a tests/log/$4.log
   done
done
#docker exec -it $1 bitcoin-cli -regtest generate $2 > temp.con
#cat -v temp.con |tr -d "^M" |tee -a tests/generate.log
```

3) 测试结果日志一览

```
root@iZ2ze4wxzv9g5i5r69vu06Z:~/mydocker/tests/log0712zn# ll
-rw-r--r-- 1 root root 7155 Jul 12 01:08 test101.log
-rw-r--r-- 1 root root 4342 Jul 12 01:09 test102.log
-rw-r--r-- 1 root root 7150 Jul 12 01:09 test111.log
-rw-r--r-- 1 root root 4343 Jul 12 01:12 test112.log
-rw-r--r-- 1 root root 7221 Jul 12 01:18 test1141.log
-rw-r--r-- 1 root root 3192 Jul 12 00:45 test11.log
-rw-r--r-- 1 root root 2326 Jul 12 01:12 test121.log
-rw-r--r-- 1 root root 3498 Jul 12 01:13 test122.log
-rw-r--r-- 1 root root 4383 Jul 12 01:13 test123.log
-rw-r--r-- 1 root root 7221 Jul 12 00:46 test12.log
-rw-r--r-- 1 root root 1668 Jul 12 01:16 test131.log
-rw-r--r-- 1 root root 2077 Jul 12 01:16 test132.log
-rw-r--r-- 1 root root 4424 Jul 12 01:17 test133.log
```

-rw-r--r-- 1 root root 3205 Jul 12 00:46 test13.log
-rw-r--r-- 1 root root 4426 Jul 12 01:18 test142.log
-rw-r--r-- 1 root root 2455 Jul 12 01:19 test151.log
-rw-r--r-- 1 root root 2077 Jul 12 01:19 test152.log
-rw-r--r-- 1 root root 4468 Jul 12 01:20 test153.log
-rw-r--r-- 1 root root 2580 Jul 12 01:20 test161.log
-rw-r--r-- 1 root root 2069 Jul 12 01:20 test162.log
-rw-r--r-- 1 root root 4507 Jul 12 01:21 test163.log
-rw-r--r-- 1 root root 7150 Jul 12 01:21 test171.log
-rw-r--r-- 1 root root 4507 Jul 12 01:21 test172.log
-rw-r--r-- 1 root root 7150 Jul 12 01:21 test181.log
-rw-r--r-- 1 root root 4508 Jul 12 01:21 test182.log
-rw-r--r-- 1 root root 1732 Jul 12 01:22 test191.log
-rw-r--r-- 1 root root 3789 Jul 12 01:24 test192.log
-rw-r--r-- 1 root root 4590 Jul 12 01:24 test193.log
-rw-r--r-- 1 root root 1660 Jul 12 01:25 test201.log
-rw-r--r-- 1 root root 2077 Jul 12 01:25 test202.log
-rw-r--r-- 1 root root 4630 Jul 12 01:25 test203.log
-rw-r--r-- 1 root root 2451 Jul 12 00:46 test21.log
-rw-r--r-- 1 root root 4943 Jul 12 00:46 test22.log
-rw-r--r-- 1 root root 3253 Jul 12 00:47 test23.log
-rw-r--r-- 1 root root 1729 Jul 12 00:49 test31.log
-rw-r--r-- 1 root root 2769 Jul 12 00:50 test32.log
-rw-r--r-- 1 root root 3294 Jul 12 00:51 test33.log
-rw-r--r-- 1 root root 7150 Jul 12 00:51 test41.log
-rw-r--r-- 1 root root 3296 Jul 12 00:52 test42.log
-rw-r--r-- 1 root root 7226 Jul 12 00:52 test51.log
-rw-r--r-- 1 root root 3297 Jul 12 00:53 test52.log
-rw-r--r-- 1 root root 2458 Jul 12 00:53 test61.log
-rw-r--r-- 1 root root 3493 Jul 12 00:53 test62.log
-rw-r--r-- 1 root root 3337 Jul 12 00:54 test63.log
-rw-r--r-- 1 root root 2451 Jul 12 00:55 test71.log
-rw-r--r-- 1 root root 3498 Jul 12 00:55 test72.log
-rw-r--r-- 1 root root 3378 Jul 12 00:56 test73.log
-rw-r--r-- 1 root root 5837 Jul 12 01:00 test810.log
-rw-r--r-- 1 root root 3739 Jul 12 01:00 test811.log
-rw-r--r-- 1 root root 2585 Jul 12 00:57 test81.log
-rw-r--r-- 1 root root 2587 Jul 12 00:57 test82.log
-rw-r--r-- 1 root root 2453 Jul 12 00:57 test83.log
-rw-r--r-- 1 root root 2453 Jul 12 00:57 test84.log
-rw-r--r-- 1 root root 2453 Jul 12 00:58 test85.log
-rw-r--r-- 1 root root 2455 Jul 12 00:58 test86.log
-rw-r--r-- 1 root root 2447 Jul 12 00:58 test87.log
-rw-r--r-- 1 root root 2453 Jul 12 00:59 test88.log

-rw-r--r-- 1 root root 2451 Jul 12 00:59 test89.log
-rw-r--r-- 1 root root 4370 Jul 12 01:05 test910.log
-rw-r--r-- 1 root root 2592 Jul 12 01:05 test911.log
-rw-r--r-- 1 root root 1737 Jul 12 01:05 test912.log
-rw-r--r-- 1 root root 1737 Jul 12 01:05 test913.log
-rw-r--r-- 1 root root 1736 Jul 12 01:05 test914.log
-rw-r--r-- 1 root root 5536 Jul 12 01:05 test915.log
-rw-r--r-- 1 root root 7598 Jul 12 01:06 test916.log
-rw-r--r-- 1 root root 4340 Jul 12 01:07 test917.log
-rw-r--r-- 1 root root 2591 Jul 12 01:04 test91.log
-rw-r--r-- 1 root root 1666 Jul 12 01:04 test92.log
-rw-r--r-- 1 root root 1665 Jul 12 01:04 test93.log
-rw-r--r-- 1 root root 2449 Jul 12 01:04 test94.log
-rw-r--r-- 1 root root 2589 Jul 12 01:04 test95.log
-rw-r--r-- 1 root root 1734 Jul 12 01:04 test96.log
-rw-r--r-- 1 root root 1735 Jul 12 01:04 test97.log
-rw-r--r-- 1 root root 2589 Jul 12 01:05 test98.log
-rw-r--r-- 1 root root 2583 Jul 12 01:05 test99.log

2、测试流程及结果说明

1) 测试环境说明

本次测试所使用环境为协会提供的服务器,服务器系统信息为 Linux,Ubuntu,64 位，为了使测试更加具有说服力，我们每个版本程序两个节点，采用 docker 模拟 6 个比特币节点，各节点详细数据如下表所示：

节点名称	节点程序版本	docker container id
btcorgNode1	比特币现有程序	ed2d1223c275
btcorgNode2	比特币现有程序	84955fb307e8
btcnewNode1	比特币 2K 防重放程序	162954e0feac
btcnewNode2	比特币 2K 防重放程序	da5c88d83f3e
btchardforkNode1	比特币 2K 非防重放程序	ab2ba80bff53
btchardforkNode2	比特币 2K 非防重放程序	2907e68ad502

注：各节点 docker 信息详见第二章第三节第二小节。

2) 获取各节点主要原始数据

在测试之前，首先要获取各节点的原始数据，包括原始块数，余额，地址等等，这样做是为了对比之后的测试数据，已验证测试的正确性。具体信息在日志 test11.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"version": 149900, "protocolversion": 70015, "walletversion": 139900, "balance": 0.00000000, "blocks": 0, "timeoffset": 0, "connections": 5,	"n3VHpJYepNisre33mDz6my2eHup877ya1"	通过 getinfo 指令获取链的主要信息，此处没有一一列出，详见日志文件。
btcorgNode2	"version": 149900, "protocolversion": 70015, "walletversion": 139900, "balance": 0.00000000, "blocks": 0, "timeoffset": 0, "connections": 9,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p"	
btcnewNode1	"version": 149900, "protocolversion": 70015, "walletversion": 139900, "balance": 0.00000000, "blocks": 0, "timeoffset": 0, "connections": 9,	"muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2"	
btcnewNode2	"version": 149900, "protocolversion": 70015, "walletversion": 139900, "balance": 0.00000000, "blocks": 0, "timeoffset": 0, "connections": 9,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt"	
btchardforkNode1	"version": 149900, "protocolversion": 70015, "walletversion": 139900, "balance": 0.00000000, "blocks": 0, "timeoffset": 0, "connections": 9,	"n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"version": 149900, "protocolversion": 70015, "walletversion": 139900, "balance": 0.00000000, "blocks": 0, "timeoffset": 0, "connections": 9,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

3) 脚本使用说明

getinfo:查询所有节点的info

getinfo [日志文件名（不用写.log）]

Eg: getinfo getinfo11

generate:挖矿

generate [节点名] [块数] [日志文件名（不用写.log）]

Eg: generate btcorgNode1 101 generate12

transaction:交易

transaction [节点1] [节点2] [金额] [日志文件名（不用写.log）]

Eg: transaction btcorgNode1 btcorgNode2 10 transaction22

1 生成节点2交易地址

2 节点1向节点2发起交易

3 所有节点查询交易

confirmtx:确认交易

confirmtx [节点名] [日志文件名（不用写.log）]

Eg: confirmtx btcorgNode1 confirmtx11

1 挖矿1块打包交易

2 所有节点查询新打包区块

getblockhash:获取指定高度块hash

getblockhash [高度] [日志文件名（不用写.log）]

Eg:getblockhash 101 getblockhash101

getnodes:获取节点信息

getnodes [日志文件名（不用写.log）]

Eg:getnodes getnodes1

4) 比特币现有程序节点间挖矿及交易

(1)节点 btcorgNode1 挖矿并确认

该节点挖矿101个块（在regtest模式下，该块之后有100个块，该快才被确认，发放挖矿报酬），脚本主要代码：

```
docker exec -it $1 bitcoin-cli -regtest generate $2 > temp.con
cat -v temp.con |tr -d "^M" |tee -a tests/log/$3.log
```

执行脚本指令：

```
generate btcorgNode1 101 test12
```

日志文件test12.log部分数据:

generate 101 blocks from btcorgNode1

[
"63521a7be8bb511d03b28948176d09d13f37e72e0b77a2cf83845c5299edd218",
"7f758cde534c277368e524dca6d50cac39379320a5eeb3b3390ea93585cc62c4",
...96行...
"198e18cf357e59b3febc609248661a407d9e3bec0c72394aaf8130c817452d1f",
"66dc83c012e2694eebe512630f5a3e1611e6974e1497ab59f7e8bfbd8691929b"
]

再次获取各节点数据，具体数据在日志 test13.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"version": 149900, "protocolversion": 70015, "walletversion": 139900, "balance": 50.00000000, "blocks": 101, "timeoffset": 0, "connections": 5,	"n3VHpJYepNisre33mDz6my2eHup877ya1"	通过 getinfo 指令获取链的主要信息，此处没有一一列出，详见日志文件。
btcorgNode2	"version": 149900, "protocolversion": 70015, "walletversion": 139900, "balance": 0.00000000, "blocks": 101, "timeoffset": 0, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p"	
btcnewNode1	"version": 149900, "protocolversion": 70015, "walletversion": 139900, "balance": 0.00000000, "blocks": 101, "timeoffset": 0, "connections": 9,	"muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2"	
btcnewNode2	"version": 149900, "protocolversion": 70015, "walletversion": 139900, "balance": 0.00000000, "blocks": 101, "timeoffset": 0, "connections": 9,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt"	

btchardforkNode1	"version": 149900, "protocolversion": 70015, "walletversion": 139900, "balance": 0.00000000, "blocks": 101, "timeoffset": 0, "connections": 9,	"n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"version": 149900, "protocolversion": 70015, "walletversion": 139900, "balance": 0.00000000, "blocks": 101, "timeoffset": 0, "connections": 9,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

可见，节点 btcorgNode1 所挖之矿均为创币交易块，由于创币交易没有签名，各节点均接收通过。

(2)btcorgNode1 向 btcorgNode2 交易 10 个比特币

btcorgNode1向btcorgNode2转账10个比特币，脚本主要代码：

```
txid=`docker exec -it $1 bitcoin-cli -regtest sendtoaddress $address $3`
echo "txid: $txid"|tee -a tests/log/$4.log
```

执行脚本指令：

```
transaction btcorgNode1 btcorgNode2 10 test21
```

btcorgNode1打包交易，脚本主要代码：

```
blockid=`docker exec -it $1 bitcoin-cli -regtest generate 1`
blockid="${blockid:6:64}"
echo "block: $blockid"|tee -a tests/log/$2.log
```

执行脚本指令：

```
confirmtx btcorgNode1 test22
```

日志文件test21.log部分数据:

btcorgNode1：


```
"amount": -10.00000000,
"fee": -0.00003840,
"confirmations": 0,
"trusted": true,
"txid": "46985a452c6279f0cc6310b365ad39fe5ad449a01202bc673f7edaf484046339",
"walletconflicts": [
],
"time": 1499750964,
```

```
  "timereceived": 1499750964,
  "bip125-replaceable": "no",
  "details": [
    {
      "account": "",
      "address": "msV1hY8kbQmvgsHDWYsKPrHrYhnVBh4tQ4",
      "category": "send",
      "amount": -10.00000000,
      "vout": 0,
      "fee": -0.00003840,
      "abandoned": false
    }
  ],
```


btcorgNode2：


```
"amount": 10.00000000,
  "confirmations": 0,
  "trusted": false,
  "txid": "46985a452c6279f0cc6310b365ad39fe5ad449a01202bc673f7edaf484046339",
  "walletconflicts": [
  ],
  "time": 1499750964,
  "timereceived": 1499750964,
  "bip125-replaceable": "no",
  "details": [
    {
      "account": "",
      "address": "msV1hY8kbQmvgsHDWYsKPrHrYhnVBh4tQ4",
      "category": "receive",
      "amount": 10.00000000,
      "label": "",
      "vout": 0
    }
  ],
```


再次获取各节点数据，具体数据在日志 test23.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注

btcorgNode1	"balance": 89.99996160, "blocks": 102, "connections": 10,	"n3VHpJYepNisre33mDz6my2eHup877ya1"	通过 getinfo 指令获取链的主要信息，此处没有一一列出，详见日志文件。
btcorgNode2	"balance": 10.00000000, "blocks": 102, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p"	
btcnewNode1	"balance": 0.00000000, "blocks": 101, "connections": 10,	"muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2"	
btcnewNode2	"balance": 0.00000000, "blocks": 101, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt"	
btchardforkNode1	"balance": 0.00000000, "blocks": 102, "connections": 10,	"n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 102, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

可见比特币现有程序节点的balance数据符合要求，btcorgNode1：50+50-0.00003840=89.99996160 ,btcorgNode2:10 ;则btcorgNode1可以向btcorgNode2发起交易，同时btchardforkNode1，2两个节点也接受到了交易块，而btcnewNode1,2两个节点没有交易块，说明这两个节点已经与其他4各节点分叉。

(3)btcorgNode2 向 btcorgNode1 交易 5 个比特币

btcorgNode2 向 btcorgNode1 转账 5 个比特币：

transaction btcorgNode2 btcorgNode1 5 test31

btcorgNode2 打包交易：

confirmtx btcorgNode2 test32

再次获取各节点数据，具体数据在日志 test33.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 144.99996160, "blocks": 103, "connections": 10,	"n3VHpJYepNisre33mDz6my2eHup877ya1" "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT",	通过 getinfo 指令获取链的主要信息，此处没有一一列出，详见日志文件。
btcorgNode2	"balance": 4.99995480, "blocks": 103, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	

btcnewNode1	"balance": 0.00000000, "blocks": 101, "connections": 10,	"muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2"	
btcnewNode2	"balance": 0.00000000, "blocks": 101, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt"	
btchardforkNode1	"balance": 0.00000000, "blocks": 103, "connections": 10,	"n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 103, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

再次确认比特币现有程序节点可以交易，同时btchardforkNode1，2两个节点也接受到了交易块，而btcnewNode1,2两个节点没有交易块，说明这两个节点已经与其他4各节点分叉。

5) 比特币现有程序节点与比特币 2K 非防重放程序间挖矿及交易

(1)btcorgNode1 挖矿，确认之前的交易。

该节点挖矿 100 块，确认上一节中的交易，因为 regtest 规则原因，所以此处需要挖矿确认之前的交易，以便下面测试，脚本主要代码同上。

执行脚本指令：

generate btcorgNode1 100 test41

再次获取各节点数据，具体数据在日志 test42.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 5095.00000000, "blocks": 203, "connections": 10,	"n3VHpJYepNisre33mDz6my2eHup877ya1" "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT",	通过 getinfo 指令获取链的主要信息，此处没有一一列出，详见日志文件。
btcorgNode2	"balance":55.00000000, "blocks": 203, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	
btcnewNode1	"balance": 0.00000000, "blocks": 101, "connections": 10,	"muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2"	
btcnewNode2	"balance": 0.00000000, "blocks": 101, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt"	
btchardforkNode1	"balance": 0.00000000, "blocks": 203, "connections": 10,	"n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	

btchardforkNode2	"balance": 0.00000000, "blocks": 203, "connections": 10,	"muHHu4sqY4QaGo57sm XSgVPEkcpcDxWo"	

(2)btchardforkNode1 挖矿，确认之前的交易

该节点挖矿 101 块，确认上一节中的交易，因为 regtest 规则原因，所以此处需要挖矿确认之前的交易，以便下面测试，再次获取各节点数据，具体数据在日志 test52.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 8745.00000000, "blocks": 304, "connections": 10,	"n3VHpJYepNisre33mDz6 my2eHup877ya1" "mqrUYzjZy8biCsKiiGpSSJ yqV2LhcWFgoT",	通过 getinfo 指令获取链的主要信息，此处没有一一列出，详见日志文件。
btcorgNode2	"balance": 10.00000000, "blocks": 304, "connections": 10,	"n46NRrxPNkVFgXHn7G myzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg 4YzrpmurAbaVw",	
btcnewNode1	"balance": 0.00000000, "blocks": 101, "connections": 10,	"muzvG6LZE1byxbe9Dcre 9zjTQTr1YUGnE2"	
btcnewNode2	"balance": 0.00000000, "blocks": 101, "connections": 10,	"mmAx7BrNydCVdjf6K4y 29ozggJBfq1cRnt"	
btchardforkNode1	"balance": 25.00000000, "blocks": 304, "connections": 10,	"n2HiY7xtbzB3YTbYLpbH9 oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 304, "connections": 10,	"muHHu4sqY4QaGo57sm XSgVPEkcpcDxWo"	

注意此处由btchardforkNode1节点挖矿101，而得到报酬25，可知报酬已经减半了。

(3)btcorgNode1 向 btchardforkNode1 转账 10 个比特币，并挖矿确认

btcorgNode1 向 btchardforkNode1 转账 10 个比特币：

transaction btcorgNode1 btchardforkNode1 10 test61

btcorgNode1 打包交易：

confirmtx btcorgNode1 test62

再次获取各节点数据，具体数据在日志 test63.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 8734.99996160, "blocks": 305, "connections": 10,	"n3VHpJYepNisre33mDz6 my2eHup877ya1" "mqrUYzjZy8biCsKiiGpSSJ	通过 getinfo 指令获取链的主要信息，

		yqV2LhcWFgoT",	此处没有一一列出，详见日志文件。
btcorgNode2	"balance": 55.00000000, "blocks": 305, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	
btcnewNode1	"balance": 0.00000000, "blocks": 101, "connections": 10,	"muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2"	
btcnewNode2	"balance": 0.00000000, "blocks": 101, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt"	
btchardforkNode1	"balance": 60.00000000, "blocks": 305, "connections": 10,	"n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 305, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

由数据可知，由于此时区块<2k，btchardfork 接受区块，重放攻击成功。

(4)btchardforkNode1 向 btcorgNode1 转账 5 个比特币，并挖矿确认

btchardforkNode1 向 btcorgNode1 转账 5 个比特币：

transaction btchardforkNode1 btcorgNode1 5 test71

btchardforkNode1 打包交易，此时区块<2k：

confirmtx btchardforkNode1 test72

再次获取各节点数据，具体数据在日志 test73.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 8739.99996160, "blocks": 306, "connections": 10,	"n3VHpJYepNisre33mDz6my2eHup877ya1" "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT",	通过 getinfo 指令获取链的主要信息，此处没有一一列出，详见日志文件。
btcorgNode2	"balance": 55.00000000, "blocks": 306, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	
btcnewNode1	"balance": 0.00000000, "blocks": 101, "connections": 10,	"muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2"	
btcnewNode2	"balance": 0.00000000, "blocks": 101, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt"	

btchardforkNode1	"balance": 79.99996160, "blocks": 306, "connections": 10,	"n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 306, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

可见，上一章节的结论再次验证

(5)btcorgNode1 向 btchardforkNode1 发起 5 笔转账，并挖矿确认

技术原理同上，此处不再过多叙述。

再次获取各节点数据，具体数据在日志 test811.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 8694.99960240, "blocks": 307, "connections": 10,	"n3VHpJYepNisre33mDz6my2eHup877ya1" "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT", "mkELTQbVpvGhBC4sUW3aD9cCLmmDiC1PmE"	通过getinfo指令获取链的主要信息，此处没有一一列出，详见日志文件。
btcorgNode2	"balance": 55.00000000, "blocks": 307, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	
btcnewNode1	"balance": 0.00000000, "blocks": 101, "connections": 10,	"muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2"	
btcnewNode2	"balance": 0.00000000, "blocks": 101, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt"	
btchardforkNode1	"balance":149.99996160, "blocks": 307, "connections": 10,	"mgzYH88LXvn7LZFBNgHj8qooubcK3Eag6y", "mj35BkwytXvbFw7P2cN98HQZeLHaAHWKs", "mjHG3tBfpGJqPQuVVVErJsYpJrr9yd2AS", "mmW5TjvNnrcsj8FtYJuruymyBqmF1hkYA", "mpHpTt4sWy3BtxGDNcSJ96eagh7Y2wsxZ", "mqGUePwFV5KPsD32y46AZ52YWje2TxVE", "mrDpPirVKZNHkibSqu3aGKmXusWXgvPKZ", "mt2UJ28Jd97sn6tH11c5BxgFs6AwYHtK", "mtUfejmwqSSE3ZNmTx3kHbHRenuRn9eQV", "myv3EF2z37s6nKakzaHo5aBsx96JiJi7vY", "n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 307, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

btcorgNode1 向 btchardforkNode1 发起 5 笔转账,并且由 btcorgNode1 打包这 5 笔交易,按照我们的预期,此时打包的这个块的大小是超过 2k 的,经过挖矿确认, btcorgNode 节点会接收这个块,

btchardforkNode 节点由于做了 2k 限制不会接收这个块,而 btcnewNode 节点由于交易验证机制不同也不会接收这个块,但是结果是 btchardforkNode 接收了这个块,出现了和预期不同的结果,根据推测可能的原因有一下方面:一是块的大小没有超过 2k;而是 docker 配置出问题;三是源程序还有我们没有发现的点在影响这块的处理结果。不过这里不会对重放攻击的验证测试造成影响，我们后面会验证并解决这个问题。

(6)btchardforkNode1 向 btcorgNode1 发起 5 笔转账，并挖矿确认

技术原理同上，此处不再过多叙述。

再次获取各节点数据，具体数据在日志 test917.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 8814.99960240, "blocks": 308, "connections": 10,	"mgEyS1XX7uRa5SE2uRtZyc7oxBYV4Aq24B", "mh4FomgAg5YNbyy5ycaWXYpUiftBe4kHb", "miVEd6HhGXWLGmJ4WqBxhXqGwgscFKhEb", "miq8KSjWFVqUexwp5QVkjzjChAQEnQsJXK", "mjurkwfmxR7doj8ebGyNZvhn6a1pKqV6a", "mkELTQbVpvGhBC4sUW3aD9cCLmmDiC1PmE" "mmXN67LBa1JgTyWhsd6HYLQEvenwzSghi2", "moBFHuqeAD82pE4Vdu1wVSzLPQySa4fn", "moSNdcvivPDyb2YBnGBNPjQjC2SPSVUX", "mpE5UAHLXfkC9aVSWiZVTeyee1UkZX26by", "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT", "mrvGfhYzvcRLHp2APEFuBFXYDktmdTJh", "msCb1xS8ekY6stPVXLTaob3w9ox6r1Z7d", "mv1b7BBSYgVLXRDePsuTK8vLx7wtxN8bm", "mw4CVeD3JbmuLDzuerYmiuVt4cJrLJ6ZEu", "mx3LVC2twPvrsQoGF3RC4Tmw6a8kk4xYW", "mxVBfUGikPmxkgnrey1Aftmh7qkUvHUus", "n3VHpJYepNisre33mDz6my2eHup877ya1"	通过 getinfo 指令获取链的主要信息，此处没有一一列出，详见日志文件。
btcorgNode2	"balance": 55.00000000, "blocks": 308, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	
btcnewNode1	"balance": 0.00000000, "blocks": 101, "connections": 10,	"muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2"	
btcnewNode2	"balance": 0.00000000, "blocks": 101, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt"	
btchardforkNode1	"balance": 54.99906720, "blocks": 308, "connections": 10,	"mgzYH88LXvn7LZFBNgHj8qooubcK3Eag6y", "mj35BkwytXvbFw7P2cN98HQZeLHaAHWKs", "mjHG3tBfpGJqPQuVVVErJsYpJrr9yd2AS", "mmW5TjvNnrcsj8FtYJuruymyBqmF1hkYA",	

		"mpHpTt4sWy3BtxGDNcSJ96eagh7Y2wsxZ", "mqGUePwFV5KPsD32y46AZ52YWje2TxVE", "mrDpPirVKZNHkibSqu3aGKmXusWXgvPKZ", "mt2UJ28Jd97sn6tH11c5BxgFs6AwYHtK", "mtUfejmwqSSE3ZNmTx3kHbHRenuRn9eQV", "myv3EF2z37s6nKakzaHo5aBsx96JiJi7vY", "n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 308, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

btchardforkNode1 向 btcorgNode1 发起 5 笔转账，这次由 btchardforkNode1 打包，此时我们的预期是一个包不能把所有的交易数据都打包进去，但是不清楚的是会打几个包，从结果数据来看，就打一个包，我们看看包里面的数据发现有的交易数据没有打入包内，由此我们验证，当交易数据超出块所能容纳的范围时，程序会选择其中的交易数据打包，此外我们还提出了两个问题，影响交易数据大小的具体因素，以及程序打包交易的顺序是怎样的，这两个问题我们会以后探究验证。

6) 比特币 2K 防重放程序节点与比特币 2K 非防重放程序间挖矿及交易

(1)btchardforkNode1 挖矿 100 块

技术原理同上，此处不再过多叙述。

再次获取各节点数据，具体数据在日志 test102.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 8840.00000000, "blocks": 408, "connections": 10,	"mgEyS1XX7uRa5SE2uRtZyc7oxBYV4Aq24B", "mh4FomgAg5YNbyy5ycaWXYpUiftBe4kHb", "miVEd6HhGXWLGmJ4WqBxhXqGwgscFKhEb", "miq8KSjWFVqUexwp5QVkjzjChAQEnQsJXK", "mjurkwfmxR7doj8ebGyNZvhn6a1pKqV6a", "mkELTQbVpvGhBC4sUW3aD9cCLmmDiC1PmE" "mmXN67LBa1JgTyWhsd6HYLQEvenwzSghi2", "moBFHuqeAD82pE4Vdu1wVSzLPQySa4fn", "moSNdcvivPDyb2YBnGBNPjQjC2SPSVUX", "mpE5UAHLXfkC9aVSWiZVTeyee1UkZX26by", "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT", "mrvGfhYzvcRLHp2APEFuBFXYDktmdTJh", "msCb1xS8ekY6stPVXLTaob3w9ox6r1Z7d", "mv1b7BBSYgVLXRDePsuTK8vLx7wtxN8bm", "mw4CVeD3JbmuLDzuerYmiuVt4cJrLJ6ZEu", "mx3LVC2twPvrsQoGF3RC4Tmw6a8kk4xYW", "mxVBfUGikPmxkgnrey1Aftmh7qkUvHUus", "n3VHpJYepNisre33mDz6my2eHup877ya1"	通过getinfo指令获取链的主要信息，此处没有一一
btcorgNode2	"balance": 55.00000000, "blocks": 408,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	

	"connections": 10,		列出，详见日志文件。
btcnewNode1	"balance": 0.00000000, "blocks": 101, "connections": 10,	"muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2"	
btcnewNode2	"balance": 0.00000000, "blocks": 101, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt"	
btchardforkNode1	"balance": 2417.50000000, "blocks": 408, "connections": 10,	"mgzYH88LXvn7LZFBNgHj8qooubcK3Eag6y", "mj35BkwytXvbFw7P2cN98HQZeLHaAHWKs", "mjHG3tBfpGJqPQuVVVErJsYpJrr9yd2AS", "mmW5TjvNnrcsj8FtYJuruymyBqmF1hkYA", "mpHpTt4sWy3BtxGDNcSJ96eagh7Y2wsxZ", "mqGUePwFV5KPsD32y46AZ52YWje2TxVE", "mrDpPirVKZNHkibSqu3aGKmXusWXgvPKZ", "mt2UJ28Jd97sn6tH11c5BxgFs6AwYHtK", "mtUfejmwqSSE3ZNmTx3kHbHRenuRn9eQV", "myv3EF2z37s6nKakzaHo5aBsx96JiJi7vY", "n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 408, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

(2)btcnewNode1 挖矿 101 块

技术原理同上，此处不再过多叙述。

再次获取各节点数据，具体数据在日志 test112.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 8840.00000000, "blocks": 408, "connections": 10,	"mgEyS1XX7uRa5SE2uRtZyc7oxBYV4Aq24B", "mh4FomgAg5YNbyy5ycaWXYpUiftBe4kHb", "miVEd6HhGXWLGmJ4WqBxhXqGwgscFKhEb", "miq8KSjWFVqUexwp5QVkjzjChAQEnQsJXK", "mjurkwfmxR7doj8ebGyNZvhn6a1pKqV6a", "mkELTQbVpvGhBC4sUW3aD9cCLmmDiC1PmE" "mmXN67LBa1JgTyWhsd6HYLQEvenwzSghi2", "moBFHuqeAD82pE4Vdu1wVSzLPQySa4fn", "moSNdcvivPDyb2YBnGBNPjQjC2SPSVUX", "mpE5UAHLXfkC9aVSWiZVTeyee1UkZX26by", "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT", "mrvGfhYzvcRLHp2APEFuBFXYDktmdTJh", "msCb1xS8ekY6stPVXLTaob3w9ox6r1Z7d", "mv1b7BBSYgVLXRDePsuTK8vLx7wtxN8bm", "mw4CVeD3JbmuLDzuerYmiuVt4cJrLJ6ZEu", "mx3LVC2twPvrsQoGF3RC4Tmw6a8kk4xYW", "mxVBfUGikPmxkgnrey1Aftmh7qkUvHUus",	通过getinfo指令获取链的主要信息，此处没

		"n3VHpJYepNisre33mDz6my2eHup877ya1"	有一一列出，详见日志文件。
btcorgNode2	"balance": 55.00000000, "blocks": 408, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	
btcnewNode1	"balance": 50.00000000, "blocks": 202, "connections": 10,	"muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2"	
btcnewNode2	"balance": 0.00000000, "blocks": 202, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt"	
btchardforkNode1	"balance": 2417.50000000, "blocks": 408, "connections": 10,	"mgzYH88LXvn7LZFBNgHj8qooubcK3Eag6y", "mj35BkwytXvbFw7P2cN98HQZeLHaAHWKs", "mjHG3tBfpGJqPQuVVVErJsYpJrr9yd2AS", "mmW5TjvNnrcsj8FtYJuruymyBqmF1hkYA", "mpHpTt4sWy3BtxGDNcSJ96eagh7Y2wsxZ", "mqGUePwFV5KPsD32y46AZ52YWje2TxVE", "mrDpPirVKZNHkibSqu3aGKmXusWXgvPKZ", "mt2UJ28Jd97sn6tH11c5BxgFs6AwYHtK", "mtUfejmwqSSE3ZNmTx3kHbHRenuRn9eQV", "myv3EF2z37s6nKakzaHo5aBsx96JiJi7vY", "n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 408, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

比特币2K防重放程序挖矿，因为已经分叉，所以这两个节点和其他四个节点的块传输阻塞。

(3)btchardforkNode1 向 btcnewNode1 转账 10 个比特币，并挖矿确认

技术原理同上，此处不再过多叙述。

再次获取各节点数据，具体数据在日志 test123.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 8840.00000000, "blocks": 409, "connections": 10,	"mgEyS1XX7uRa5SE2uRtZyc7oxBYV4Aq24B", "mh4FomgAg5YNbyy5ycaWXYpUiftBe4kHb", "miVEd6HhGXWLGmJ4WqBxhXqGwgscFKhEb", "miq8KSjWFVqUexwp5QVkjzjChAQEnQsJXK", "mjurkwfmxR7doj8ebGyNZvhn6a1pKqV6a", "mkELTQbVpvGhBC4sUW3aD9cCLmmDiC1PmE" "mmXN67LBa1JgTyWhsd6HYLQEvenwzSghi2", "moBFHuqeAD82pE4Vdu1wVSzLPQySa4fn", "moSNdcvivPDyb2YBnGBNPjQjC2SPSVUX", "mpE5UAHLXfkC9aVSWiZVTeyee1UkZX26by", "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT", "mrvGfhYzvcRLHp2APEFuBFXYDktmdTJh",	通过getinfo指令获取链的主要

		"msCb1xS8ekY6stPVXLTaob3w9ox6r1Z7d", "mv1b7BBSYgVLXRDePsuTK8vLx7wtxN8bm", "mw4CVeD3JbmuLDzuerYmiuVt4cJrLJ6ZEu", "mx3LVC2twPvrsQoGF3RC4Tmw6a8kk4xYW", "mxVBfUGikPmxkgnrey1Aftmh7qkUvHUus", "n3VHpJYepNisre33mDz6my2eHup877ya1"	信息，此处没有一一列出，详见日志文件。
btcorgNode2	"balance": 55.00000000, "blocks": 409, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	
btcnewNode1	"balance": 50.00000000, "blocks": 202, "connections": 10,	"mkq1TB1VQPiAXnJEybTFdWzpm6hDVbUu4", "muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2"	
btcnewNode2	"balance": 0.00000000, "blocks": 202, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt"	
btchardforkNode1	"balance": 2419.99989560, "blocks": 409, "connections": 10,	"mgzYH88LXvn7LZFBNgHj8qooubcK3Eag6y", "mj35BkwytXvbFw7P2cN98HQZeLHaAHWKs", "mjHG3tBfpGJqPQuVVVErJsYpJrr9yd2AS", "mmW5TjvNnrcsj8FtYJuruymyBqmF1hkYA", "mpHpTt4sWy3BtxGDNcSJ96eagh7Y2wsxZ", "mqGUePwFV5KPsD32y46AZ52YWje2TxVE", "mrDpPirVKZNHkibSqu3aGKmXusWXgvPKZ", "mt2UJ28Jd97sn6tH11c5BxgFs6AwYHtK", "mtUfejmwqSSE3ZNmTx3kHbHRenuRn9eQV", "myv3EF2z37s6nKakzaHo5aBsx96JiJi7vY", "n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 409, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

由上面数据可知，btcnewNode不接受旧区块，节点区块数没有增加，balance也没有增加，防重放攻击成功。

(4)btcnewNode1 向 btchardforkNode1 转账 5 个比特币，并挖矿确认

技术原理同上，此处不再过多叙述。

再次获取各节点数据，具体数据在日志 test133.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 8840.00000000, "blocks": 409, "connections": 10,	"mgEyS1XX7uRa5SE2uRtZyc7oxBYV4Aq24B", "mh4FomgAg5YNbyy5ycaWXYpUiftBe4kHb", "miVEd6HhGXWLGmJ4WqBxhXqGwgscFKhEb", "miq8KSjWFVqUexwp5QVkjzjChAQEnQsJXK", "mjurkwfmxR7doj8ebGyNZvhn6a1pKqV6a", "mkELTQbVpvGhBC4sUW3aD9cCLmmDiC1PmE"	通过getinfo指令

		"mmXN67LBa1JgTyWhsd6HYLQEvenwzSghi2", "moBFHuqeAD82pE4Vdu1wVSzLPQySa4fn", "moSNdcvivPDyb2YBnGBNPjQjC2SPSVUX", "mpE5UAHLXfkC9aVSWiZVTeyee1UkZX26by", "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT", "mrvGfhYzvcRLHp2APEFuBFXYDktmdTJh", "msCb1xS8ekY6stPVXLTaob3w9ox6r1Z7d", "mv1b7BBSYgVLXRDePsuTK8vLx7wtxN8bm", "mw4CVeD3JbmuLDzuerYmiuVt4cJrLJ6ZEu", "mx3LVC2twPvrsQoGF3RC4Tmw6a8kk4xYW", "mxVBfUGikPmxkgnrey1Aftmh7qkUvHUus", "n3VHpJYepNisre33mDz6my2eHup877ya1"	获取链的主要信息，此处没有一一列出，详见日志文件。
btcorgNode2	"balance": 55.00000000, "blocks": 409, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	
btcnewNode1	"balance": 94.99996160, "blocks": 203, "connections": 10,	"mkq1TB1VQPiAXnJEybTFdWzpm6hDVbUu4", "muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2"	
btcnewNode2	"balance": 0.00000000, "blocks": 203, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt"	
btchardforkNode1	"balance": 2419.99989560, "blocks": 409, "connections": 10,	"mgzYH88LXvn7LZFBNgHj8qooubcK3Eag6y", "mj35BkwytXvbFw7P2cN98HQZeLHaAHWKs", "mjHG3tBfpGJqPQuVVVErJsYpJrr9yd2AS", "mmW5TjvNnrcsj8FtYJuruymyBqmF1hkYA", "mpHpTt4sWy3BtxGDNcSJ96eagh7Y2wsxZ", "mqGUePwFV5KPsD32y46AZ52YWje2TxVE", "mrDpPirVKZNHkibSqu3aGKmXusWXgvPKZ", "mt2UJ28Jd97sn6tH11c5BxgFs6AwYHtK", "mtUfejmwqSSE3ZNmTx3kHbHRenuRn9eQV", "myv3EF2z37s6nKakzaHo5aBsx96JiJi7vY", "n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 409, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

再次验证上章节结论，btcnewNode 链长度增加，btchardforkNode 和 btcorgNode 不增加。

7) 比特币 2K 防重放程序节点间挖矿及交易

(1)btcnewNode1 挖矿 101 块

技术原理同上，此处不再过多叙述。

再次获取各节点数据，具体数据在日志 test142.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注

btcorgNode1	"balance": 8840.00000000, "blocks": 409, "connections": 10,	"mgEyS1XX7uRa5SE2uRtZyc7oxBYV4Aq24B", "mh4FomgAg5YNbyy5ycaWXYpUiftBe4kHb", "miVEd6HhGXWLGmJ4WqBxhXqGwgscFKhEb", "miq8KSjWFVqUexwp5QVkjzjChAQEnQsJXK", "mjurkwfmxR7doj8ebGyNZvhn6a1pKqV6a", "mkELTQbVpvGhBC4sUW3aD9cCLmmDiC1PmE" "mmXN67LBa1JgTyWhsd6HYLQEvenwzSghi2", "moBFHuqeAD82pE4Vdu1wVSzLPQySa4fn", "moSNdcvivPDyb2YBnGBNPjQjC2SPSVUX", "mpE5UAHLXfkC9aVSWiZVTeyee1UkZX26by", "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT", "mrvGfhYzvcRLHp2APEFuBFXYDktmdTJh", "msCb1xS8ekY6stPVXLTaob3w9ox6r1Z7d", "mv1b7BBSYgVLXRDePsuTK8vLx7wtxN8bm", "mw4CVeD3JbmuLDzuerYmiuVt4cJrLJ6ZEu", "mx3LVC2twPvrsQoGF3RC4Tmw6a8kk4xYW", "mxVBfUGikPmxkgnrey1Aftmh7qkUvHUus", "n3VHpJYepNisre33mDz6my2eHup877ya1"	通过getinfo指令获取链的主要信息，此处没有一一列出，详见日志文件。
btcorgNode2	"balance": 55.00000000, "blocks": 409, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	
btcnewNode1	"balance": 3770.00000000, "blocks": 304, "connections": 10,	"mkq1TB1VQPiAXnJEybTFdWzpm6hDVbUu4", "muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2"	
btcnewNode2	"balance": 0.00000000, "blocks": 304, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt"	
btchardforkNode1	"balance": 2419.99989560, "blocks": 409, "connections": 10,	"mgzYH88LXvn7LZFBNgHj8qooubcK3Eag6y", "mj35BkwytXvbFw7P2cN98HQZeLHaAHWKs", "mjHG3tBfpGJqPQuVVVErJsYpJrr9yd2AS", "mmW5TjvNnrcsj8FtYJuruymyBqmF1hkYA", "mpHpTt4sWy3BtxGDNcSJ96eagh7Y2wsxZ", "mqGUePwFV5KPsD32y46AZ52YWje2TxVE", "mrDpPirVKZNHkibSqu3aGKmXusWXgvPKZ", "mt2UJ28Jd97sn6tH11c5BxgFs6AwYHtK", "mtUfejmwqSSE3ZNmTx3kHbHRenuRn9eQV", "myv3EF2z37s6nKakzaHo5aBsx96JiJi7vY", "n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 409, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

由数据可见，分叉后，btcnewNode 链长度增加，btchardforkNode 和 btcorgNode 不增加

(2)btcnewNode1 向 btcnewNode2 转账 10 个比特币，并挖矿确认

技术原理同上，此处不再过多叙述。

再次获取各节点数据，具体数据在日志 test153.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 8840.00000000, "blocks": 409, "connections": 10,	"mgEyS1XX7uRa5SE2uRtZyc7oxBYV4Aq24B", "mh4FomgAg5YNbyy5ycaWXYpUiftBe4kHb", "miVEd6HhGXWLGmJ4WqBxhXqGwgscFKhEb", "miq8KSjWFVqUexwp5QVkjzjChAQEnQsJXK", "mjurkwfmxR7doj8ebGyNZvhn6a1pKqV6a", "mkELTQbVpvGhBC4sUW3aD9cCLmmDiC1PmE" "mmXN67LBa1JgTyWhsd6HYLQEvenwzSghi2", "moBFHuqeAD82pE4Vdu1wVSzLPQySa4fn", "moSNdcvivPDyb2YBnGBNPjQjC2SPSVUX", "mpE5UAHLXfkC9aVSWiZVTeyee1UkZX26by", "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT", "mrvGfhYzvcRLHp2APEFuBFXYDktmdTJh", "msCb1xS8ekY6stPVXLTaob3w9ox6r1Z7d", "mv1b7BBSYgVLXRDePsuTK8vLx7wtxN8bm", "mw4CVeD3JbmuLDzuerYmiuVt4cJrLJ6ZEu", "mx3LVC2twPvrsQoGF3RC4Tmw6a8kk4xYW", "mxVBfUGikPmxkgnrey1Aftmh7qkUvHUus", "n3VHpJYepNisre33mDz6my2eHup877ya1"	通过getinfo指令获取链的主要信息，此处没有一一列出，详见日志文件。
btcorgNode2	"balance": 55.00000000, "blocks": 409, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	
btcnewNode1	"balance": 3784.99996160, "blocks": 305, "connections": 10,	"mkq1TB1VQPiAXnJEybTFdWzpm6hDVbUu4", "muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2"	
btcnewNode2	"balance": 10.00000000, "blocks": 305, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt"	
btchardforkNode1	"balance": 2419.99989560, "blocks": 409, "connections": 10,	"mgzYH88LXvn7LZFBNgHj8qooubcK3Eag6y", "mj35BkwytXvbFw7P2cN98HQZeLHaAHWKs", "mjHG3tBfpGJqPQuVVVErJsYpJrr9yd2AS", "mmW5TjvNnrcsj8FtYJuruymyBqmF1hkYA", "mpHpTt4sWy3BtxGDNcSJ96eagh7Y2wsxZ", "mqGUePwFV5KPsD32y46AZ52YWje2TxVE", "mrDpPirVKZNHkibSqu3aGKmXusWXgvPKZ", "mt2UJ28Jd97sn6tH11c5BxgFs6AwYHtK", "mtUfejmwqSSE3ZNmTx3kHbHRenuRn9eQV", "myv3EF2z37s6nKakzaHo5aBsx96JiJi7vY",	

		"n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 409, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

由数据可见，分叉后，btcnewNode 链长度增加，btchardforkNode 和btcorgNode不增加，防重放成功。

(3)btcnewNode2 向 btcnewNode1 转账 5 个比特币，并挖矿确认

技术原理同上，此处不再过多叙述。

再次获取各节点数据，具体数据在日志 test163.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 8840.00000000, "blocks": 409, "connections": 10,	"mgEyS1XX7uRa5SE2uRtZyc7oxBYV4Aq24B", "mh4FomgAg5YNbyy5ycaWXYpUiftBe4kHb", "miVEd6HhGXWLGmJ4WqBxhXqGwgscFKhEb", "miq8KSjWFVqUexwp5QVkjzjChAQEnQsJXK", "mjurkwfmxR7doj8ebGyNZvhn6a1pKqV6a", "mkELTQbVpvGhBC4sUW3aD9cCLmmDiC1PmE" "mmXN67LBa1JgTyWhsd6HYLQEvenwzSghi2", "moBFHuqeAD82pE4Vdu1wVSzLPQySa4fn", "moSNdcvivPDyb2YBnGBNPjQjC2SPSVUX", "mpE5UAHLXfkC9aVSWiZVTeyee1UkZX26by", "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT", "mrvGfhYzvcRLHp2APEFuBFXYDktmdTJh", "msCb1xS8ekY6stPVXLTaob3w9ox6r1Z7d", "mv1b7BBSYgVLXRDePsuTK8vLx7wtxN8bm", "mw4CVeD3JbmuLDzuerYmiuVt4cJrLJ6ZEu", "mx3LVC2twPvrsQoGF3RC4Tmw6a8kk4xYW", "mxVBfUGikPmxkgnrey1Aftmh7qkUvHUus", "n3VHpJYepNisre33mDz6my2eHup877ya1"	通过getinfo指令获取链的主要信息，此处没有一一列出，详见日志文件。
btcorgNode2	"balance": 55.00000000, "blocks": 409, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	
btcnewNode1	"balance": 3814.99996160, "blocks": 306, "connections": 10,	"mkq1TB1VQPiAXnJEybTFdWzpm6hDVbUu4", "muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2", "n22F9DeejoS3baZwKoFW63SeB39W4xPSx"	
btcnewNode2	"balance": 4.99995480, "blocks": 306, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt", "n1Y7hQbotCPz4xpfYX6apoKEX9FuiWvv3q"	
btchardforkNode1	"balance": 2419.99989560, "blocks": 409,	"mgzYH88LXvn7LZFBNgHj8qooubcK3Eag6y", "mj35BkwytXvbFw7P2cN98HQZeLHaAHWKs", "mjHG3tBfpGJqPQuVVVErJsYpJrr9yd2AS",	

	"connections": 10,	"mmW5TjvNnrcsj8FtYJuruymyBqmF1hkYA", "mpHpTt4sWy3BtxGDNcSJ96eagh7Y2wsxZ", "mqGUePwFV5KPsD32y46AZ52YWje2TxVE", "mrDpPirVKZNHkibSqu3aGKmXusWXgvPKZ", "mt2UJ28Jd97sn6tH11c5BxgFs6AwYHtK", "mtUfejmwqSSE3ZNmTx3kHbHRenuRn9eQV", "myv3EF2z37s6nKakzaHo5aBsx96JiJi7vY", "n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 409, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

由数据可见，分叉后，btcnewNode 链长度增加，btchardforkNode 和btcorgNode不增加，防重放成功。

8) 比特币 2K 防重放程序节点和比特币现有程序间挖矿及交易

(1)btcorgNode1 挖矿 100 块

技术原理同上，此处不再过多叙述。

再次获取各节点数据，具体数据在日志 test172.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 8840.00000000, "blocks": 509, "connections": 10,	"mgEyS1XX7uRa5SE2uRtZyc7oxBYV4Aq24B", "mh4FomgAg5YNbyy5ycaWXYpUiftBe4kHb", "miVEd6HhGXWLGmJ4WqBxhXqGwgscFKhEb", "miq8KSjWFVqUexwp5QVkjzjChAQEnQsJXK", "mjurkwfmxR7doj8ebGyNZvhn6a1pKqV6a", "mkELTQbVpvGhBC4sUW3aD9cCLmmDiC1PmE" "mmXN67LBa1JgTyWhsd6HYLQEvenwzSghi2", "moBFHuqeAD82pE4Vdu1wVSzLPQySa4fn", "moSNdcvivPDyb2YBnGBNPjQjC2SPSVUX", "mpE5UAHLXfkC9aVSWiZVTeyee1UkZX26by", "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT", "mrvGfhYzvcRLHp2APEFuBFXYDktmdTJh", "msCb1xS8ekY6stPVXLTaob3w9ox6r1Z7d", "mv1b7BBSYgVLXRDePsuTK8vLx7wtxN8bm", "mw4CVeD3JbmuLDzuerYmiuVt4cJrLJ6ZEu", "mx3LVC2twPvrsQoGF3RC4Tmw6a8kk4xYW", "mxVBfUGikPmxkgnrey1Aftmh7qkUvHUus", "n3VHpJYepNisre33mDz6my2eHup877ya1"	通过getinfo指令获取链的主要信息，此处没有一一列出，详
btcorgNode2	"balance": 55.00000000, "blocks": 509, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	
btcnewNode1	"balance": 3814.99996160,	"mkq1TB1VQPiAXnJEybTFdWzpm6hDVbUu4", "muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2",	

	"blocks": 306, "connections": 10,	"n22F9DeejoS3baZwKoFW63SeB39W4xPSx"	见日志文件。
btcnewNode2	"balance": 4.99995480, "blocks": 306, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt", "n1Y7hQbotCPz4xpfYX6apoKEX9FuiWvv3q"	
btchardforkNode1	"balance": 3670.00000000, "blocks": 509, "connections": 10,	"mgzYH88LXvn7LZFBNgHj8qooubcK3Eag6y", "mj35BkwytXvbFw7P2cN98HQZeLHaAHWKs", "mjHG3tBfpGJqPQuVVVErJsYpJrr9yd2AS", "mmW5TjvNnrcsj8FtYJuruymyBqmF1hkYA", "mpHpTt4sWy3BtxGDNcSJ96eagh7Y2wsxZ", "mqGUePwFV5KPsD32y46AZ52YWje2TxVE", "mrDpPirVKZNHkibSqu3aGKmXusWXgvPKZ", "mt2UJ28Jd97sn6tH11c5BxgFs6AwYHtK", "mtUfejmwqSSE3ZNmTx3kHbHRenuRn9eQV", "myv3EF2z37s6nKakzaHo5aBsx96JiJi7vY", "n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 509, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

由数据可知，btcnewNode 链长度不增加，btchardforkNode 和 btcorgNode 增加

(2)btcnewNode1 挖矿 100 块

技术原理同上，此处不再过多叙述。

再次获取各节点数据，具体数据在日志 test182.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 8840.00000000, "blocks": 509, "connections": 10,	"mgEyS1XX7uRa5SE2uRtZyc7oxBYV4Aq24B", "mh4FomgAg5YNbyy5ycaWXYpUiftBe4kHb", "miVEd6HhGXWLGmJ4WqBxhXqGwgscFKhEb", "miq8KSjWFVqUexwp5QVkjzjChAQEnQsJXK", "mjurkwfmxR7doj8ebGyNZvhn6a1pKqV6a", "mkELTQbVpvGhBC4sUW3aD9cCLmmDiC1PmE" "mmXN67LBa1JgTyWhsd6HYLQEvenwzSghi2", "moBFHuqeAD82pE4Vdu1wVSzLPQySa4fn", "moSNdcvivPDyb2YBnGBNPjQjC2SPSVUX", "mpE5UAHLXfkC9aVSWiZVTeyee1UkZX26by", "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT", "mrvGfhYzvcRLHp2APEFuBFXYDktmdTJh", "msCb1xS8ekY6stPVXLTaob3w9ox6r1Z7d", "mv1b7BBSYgVLXRDePsuTK8vLx7wtxN8bm", "mw4CVeD3JbmuLDzuerYmiuVt4cJrLJ6ZEu", "mx3LVC2twPvrsQoGF3RC4Tmw6a8kk4xYW", "mxVBfUGikPmxkgnrey1Aftmh7qkUvHUus", "n3VHpJYepNisre33mDz6my2eHup877ya1"	通过getinfo指令获取链的主要信息，此处没有

btcorgNode2	"balance": 55.00000000, "blocks": 509, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	一一列出，详见日志文件。
btcnewNode1	"balance": 6215.00000000, "blocks": 406, "connections": 10,	"mkq1TB1VQPiAXnJEybTFdWzpm6hDVbUu4", "muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2", "n22F9DeejoS3baZwKoFW63SeB39W4xPSx"	
btcnewNode2	"balance": 17.50000000, "blocks": 406, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt", "n1Y7hQbotCPz4xpfYX6apoKEX9FuiWvv3q"	
btchardforkNode1	"balance": 3670.00000000, "blocks": 509, "connections": 10,	"mgzYH88LXvn7LZFBNgHj8qooubcK3Eag6y", "mj35BkwytXvbFw7P2cN98HQZeLHaAHWKs", "mjHG3tBfpGJqPQuVVVErJsYpJrr9yd2AS", "mmW5TjvNnrcsj8FtYJuruymyBqmF1hkYA", "mpHpTt4sWy3BtxGDNcSJ96eagh7Y2wsxZ", "mqGUePwFV5KPsD32y46AZ52YWje2TxVE", "mrDpPirVKZNHkibSqu3aGKmXusWXgvPKZ", "mt2UJ28Jd97sn6tH11c5BxgFs6AwYHtK", "mtUfejmwqSSE3ZNmTx3kHbHRenuRn9eQV", "myv3EF2z37s6nKakzaHo5aBsx96JiJi7vY", "n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 509, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

由数据可知，btcnewNode 链长度增加，btchardforkNode 和 btcorgNode 不增加

(3)btcorgNode1 向 btcnewNode1 转账 10 个比特币，并挖矿确认

技术原理同上，此处不再过多叙述。

再次获取各节点数据，具体数据在日志 test193.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 8832.49985040, "blocks": 510, "connections": 10,	"mgEyS1XX7uRa5SE2uRtZyc7oxBYV4Aq24B", "mh4FomgAg5YNbyy5ycaWXYpUiftBe4kHb", "miVEd6HhGXWLGmJ4WqBxhXqGwgscFKhEb", "miq8KSjWFVqUexwp5QVkjzjChAQEnQsJXK", "mjurkwfmxR7doj8ebGyNZvhn6a1pKqV6a", "mkELTQbVpvGhBC4sUW3aD9cCLmmDiC1PmE" "mmXN67LBa1JgTyWhsd6HYLQEvenwzSghi2", "moBFHuqeAD82pE4Vdu1wVSzLPQySa4fn", "moSNdcvivPDyb2YBnGBNPjQjC2SPSVUX", "mpE5UAHLXfkC9aVSWiZVTeyee1UkZX26by", "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT", "mrvGfhYzvcRLHp2APEFuBFXYDktmdTJh", "msCb1xS8ekY6stPVXLTaob3w9ox6r1Z7d",	通过getinfo指令获取链的主要信

		"mv1b7BBSYgVLXRDePsuTK8vLx7wtxN8bm", "mw4CVeD3JbmuLDzuerYmiuVt4cJrLJ6ZEu", "mx3LVC2twPvrsQoGF3RC4Tmw6a8kk4xYW", "mxVBfUGikPmxkgnrey1Aftmh7qkUvHUus", "n3VHpJYepNisre33mDz6my2eHup877ya1"	息，此处没有一一列出，详见日志文件。
btcorgNode2	"balance": 55.00000000, "blocks": 510, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	
btcnewNode1	"balance": 6215.00000000, "blocks": 406, "connections": 10,	"mkq1TB1VQPiAXnJEybTFdWzpm6hDVbUu4", "muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2", "n22F9DeejoS3baZwKoFW63SeB39W4xPSx"	
btcnewNode2	"balance": 17.50000000, "blocks": 406, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt", "n1Y7hQbotCPz4xpfYX6apoKEX9FuiWvv3q"	
btchardforkNode1	"balance": 3670.00000000, "blocks": 510, "connections": 10,	"mgzYH88LXvn7LZFBNgHj8qooubcK3Eag6y", "mj35BkwytXvbFw7P2cN98HQZeLHaAHWKs", "mjHG3tBfpGJqPQuVVVErJsYpJrr9yd2AS", "mmW5TjvNnrcsj8FtYJuruymyBqmF1hkYA", "mpHpTt4sWy3BtxGDNcSJ96eagh7Y2wsxZ", "mqGUePwFV5KPsD32y46AZ52YWje2TxVE", "mrDpPirVKZNHkibSqu3aGKmXusWXgvPKZ", "mt2UJ28Jd97sn6tH11c5BxgFs6AwYHtK", "mtUfejmwqSSE3ZNmTx3kHbHRenuRn9eQV", "myv3EF2z37s6nKakzaHo5aBsx96JiJi7vY", "n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 510, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

由数据可知，在 tcorgNode1 打包交易后，btchardforkNode 和 btcorgNode 接收区块，btcnewNode 不接受，btcnewNode 链长度不增加，btchardforkNode 和 btcorgNode 增加，防重放成功

(4)btcnewNode1 向 btcorgNode1 转账 5 个比特币，并挖矿确认

技术原理同上，此处不再过多叙述。

再次获取各节点数据，具体数据在日志 test203.log 中，主要数据如下表所示：

节点名称	getinfo 指令主要信息	getaddressesbyaccount "" 指令主要信息	备注
btcorgNode1	"balance": 8832.49985040, "blocks": 510, "connections": 10,	"mgEyS1XX7uRa5SE2uRtZyc7oxBYV4Aq24B", "mh4FomgAg5YNbyy5ycaWXYpUiftBe4kHb", "miVEd6HhGXWLGmJ4WqBxhXqGwgscFKhEb", "miq8KSjWFVqUexwp5QVkjzjChAQEnQsJXK", "mjurkwfmxR7doj8ebGyNZvhn6a1pKqV6a", "mkELTQbVpvGhBC4sUW3aD9cCLmmDiC1PmE"	通过getinfo指令

		"mmXN67LBa1JgTyWhsd6HYLQEvenwzSghi2", "moBFHuqeAD82pE4Vdu1wVSzLPQySa4fn", "moSNdcvivPDyb2YBnGBNPjQjC2SPSVUX", "mpE5UAHLXfkC9aVSWiZVTeyee1UkZX26by", "mqrUYzjZy8biCsKiiGpSSJyqV2LhcWFgoT", "mrvGfhYzvcRLHp2APEFuBFXYDktmdTJh", "msCb1xS8ekY6stPVXLTaob3w9ox6r1Z7d", "mv1b7BBSYgVLXRDePsuTK8vLx7wtxN8bm", "mw4CVeD3JbmuLDzuerYmiuVt4cJrLJ6ZEu", "mx3LVC2twPvrsQoGF3RC4Tmw6a8kk4xYW", "mxVBfUGikPmxkgnrey1Aftmh7qkUvHUus", "n3VHpJYepNisre33mDz6my2eHup877ya1"	获取链的主要信息，此处没有一一列出，详见日志文件。
btcorgNode2	"balance": 55.00000000, "blocks": 510, "connections": 10,	"n46NRrxPNkVFgXHn7GmyzD9bubt2gBZf4p" "mgjj6JYP9wZtotU3ZsFFg4YzrpmurAbaVw",	
btcnewNode1	"balance": 6222.49996160, "blocks": 407, "connections": 10,	"mkq1TB1VQPiAXnJEybTFdWzpm6hDVbUu4", "muzvG6LZE1byxbe9Dcre9zjTQTr1YUGnE2", "n22F9DeejoS3baZwKoFW63SeB39W4xPSx"	
btcnewNode2	"balance": 17.50000000, "blocks": 407, "connections": 10,	"mmAx7BrNydCVdjf6K4y29ozggJBfq1cRnt", "n1Y7hQbotCPz4xpfYX6apoKEX9FuiWvv3q"	
btchardforkNode1	"balance": 3670.00000000, "blocks": 510, "connections": 10,	"mgzYH88LXvn7LZFBNgHj8qooubcK3Eag6y", "mj35BkwytXvbFw7P2cN98HQZeLHaAHWKs", "mjHG3tBfpGJqPQuVVVErJsYpJrr9yd2AS", "mmW5TjvNnrcsj8FtYJuruymyBqmF1hkYA", "mpHpTt4sWy3BtxGDNcSJ96eagh7Y2wsxZ", "mqGUePwFV5KPsD32y46AZ52YWje2TxVE", "mrDpPirVKZNHkibSqu3aGKmXusWXgvPKZ", "mt2UJ28Jd97sn6tH11c5BxgFs6AwYHtK", "mtUfejmwqSSE3ZNmTx3kHbHRenuRn9eQV", "myv3EF2z37s6nKakzaHo5aBsx96JiJi7vY", "n2HiY7xtbzB3YTbYLpbH9oU2E5b2NEWvn1"	
btchardforkNode2	"balance": 0.00000000, "blocks": 510, "connections": 10,	"muHHu4sqY4QaGo57smXSgVPEkcpcDxWo"	

再次验证防重放成功。

三、重放攻击验证解决过程中的问题

本次在解决比特币分叉后重放攻击的问题的过程中，我们遇到了一些问题，有些问题比较轻微容易解决，有些问题是经过我反复研究才得以验证的，但是由于时间和经理有限，仍有部分问题有待验证和解决，下面就将我们遇到的已经验证的问题和待解决的的问题展示给大家。

1) 验证的问题

Q 1. 测试服务器间通讯端口配置。

需要开启 18444 端口

Q 2. 如何得到一个非防护的硬分叉来验证重放攻击可能发生?

改写 MAX_BLOCK_BASE_SIZE，做一个 2k 容量节点，当原始节点发出正常交易，分叉即形成。

Q 3. 关于钱包 address。

一个钱包可以有一到多个地址，我们可以为每一笔交易生成一个新地址，或者多笔交易用同一个地址。

Q 4. 关于手工造交易发起重放。

交易是可以手工制造的，但是签名是最后一步，如果我们篡改了交易的内容但是没有发起方的私钥，是没有办法正确签名的。重放攻击应该是把一条链上正常交易，提交到另一条链上，而不能制造一个假交易。需要 vin 里不包含分叉后的新 UTXO 才能重放成功。

Q 5. 硬分叉后，节点间通讯中断。

规则改变，验证失败，被定义为问题节点，断开通讯。设置白名单后，不再断开。

Q 6. 新旧节点分叉后仍可以同步区块。

创币交易，不包含签名，可以通过验证

Q 7. 谨慎清除 regtest。

Root 下.bitcoin/regtest，注意不要删掉配置文件。

Q 8. 对等节点 gettransaction 查询不到交易数据，交易能被打包。

gettransaction 仅用于查询与自己有关的交易，即：发送交易的双方可以通过 txid 查询交易详情，其他节点不能查询。gettransaction 查询不到交易，不代表节点不能验证交易。

Q 9. 分叉一段时间后，查询指定高度的区块详情。

用 getblockhash 查询指定高度块 hash，再用 getblock 通过 hash 查询块详情。

Q 10. Getblockchaininfo 得到的 blocks 和 headers 数目不一致。

Header 在内存中没有在文件中，删除文件后 header 仍然存在。

Q 11. 2k 节点分叉，挖矿报区块尺寸错误。

源程序打包区块未做 MAX_BLOCK_BASE_SIZE 的限制，导致内存池交易大于可打包大小，打包报错。

2) 待解决验证的问题

Q 1. MAX_BOCK_SERIALIZED_SIZE、AX_BOCK_SWEIGHT、AX_BLOCK_BASE_SIZE三个常量需要确认一下都是做什么用的?

Q 2. Connections的数字到底代表什么?

Q 3. 分叉之后的短链，是怎么处理的？

Q 4. 什么影响着交易大小?

Q 5. 其他系统安装bitcoin程序（比如win，centos）?

Q 6. 为什么不用区块高度而要用时间戳区分新旧?

四、总结

我们兴趣小组本次解决比特币分叉防重放攻击这个问题,大家都付出了不少,由于大家平时都有自己的正向工作需要处理,都是抽时间来分工合作解决问题,有时候讨论解决问题到凌晨一两点。虽然这样但是由于本次开发由于时间比较短，团队成员的能力和精力有限，还是存在很多不足之处。比如，代码规范有待提高(注释最好用英文)，尚未完全模拟比特币真实环境，没有进行大规模的压力测试等等。 总之还有很多值得改进的地方，同时在验证测试的时候还发现了其他的一些问题有待验证和解决。

我们通过验证解决防重放攻击这个问题，感觉收获颇多。同时深知个体或者几个人的力量是有限的，因此我们希望通过代码开源的形成，结合 DACA 开源社区的力量，能够不断完善本项目，如有任何问题欢迎交流。邮箱地址:xuxinlai2002@gmail.com

最后再次感谢 清华大学 icener 老师们的教导和 DACA 协会支持！