Litecoin Data Mining

UCL 2018 Cryptanalysis PK Project

Qin Tang (ucabqta@ucl.ac.uk)

This project is modified from the BitcoinDatabaseGenerator project (https://github.com/ladimolnar/BitcoinDatabaseGenerator). This project can help you to generate a database of Litecoin Blockchain on MS SQL 2014 Full Paid version.

To parse all the blocks of Litecoin, this project replaces the package BitcoinBlockchain.1.2.5.0 with BitcoinBlockchain project (https://github.com/ladimolnar/BitcoinBlockchain). With this operation, this project is able to rewrite the functions of parsing blocks. I refer this https://github.com/ladimolnar/BitcoinBlockchain/pull/2 to modify the parsing blocks supporting the SegWit transactions in my project.

To adapt this project to the Litecoin, I change the first block id from D9B4BEF9 to DBB6C0FB.

To do the data mining, I add the InputScript as a new column in the TransactionInput table. Also, I modify the program to obtain the InputScript of each block and add corresponding operation to operate on the database. Thus, the new column of InputScript in transactionInput table can be established successfully.

By analyzing the InputScript, a table of random can be created. Then the reused random can be extracted from the table of random. These two new tables can be created by the SQL script (https://github.com/JasonPap/Reused-Bitcoin-Numbers/blob/master/SQL/RandomsToTables.sql) with small modification.

Contents of this file

PART 1. Prerequisites
PART 2. Run this program
PART 3. Analyze the generated database

PART 1. Prerequisites

1. Download the Litecoin core wallet.
   https://download.litecoin.org/litecoin-0.15.1/win/litecoin-0.15.1-win64-setup.exe
   Backup the .dat files,
   "X:\active_live_blockchain\satoshi120"
2. Install the 2014 Microsoft SQL Server Full Paid Enterprise edition.
3. Install the 2017 Visual Studio Community Edition.

PART 2. Run this program

1. In windows “cmd”, Run:

```
d:\Litecoin\Sources\BitcoinDatabaseGenerator\bin\Debug>BitcoinDatabaseGenerator.
exe /BlockchainPath F:\Litecoin\active_live_blockchain\satoshi\blocks /SqlServer
Name LENOVO-PC /SqlDbName Litecoin2
```

D:\Litecoin\Sources\BitcoinDatabaseGenerator\bin\Debug, this route has the program **BitcoinDatabaseGenerator.exe** and other required .dll files.

/BlockchainPath is followed by the location of all .dat file of Litecoin blocks

/SqlServerName is followed by the name of your Sql Server

/SqlDbName is followed by the name of your database; here I create a database named Litecoin2

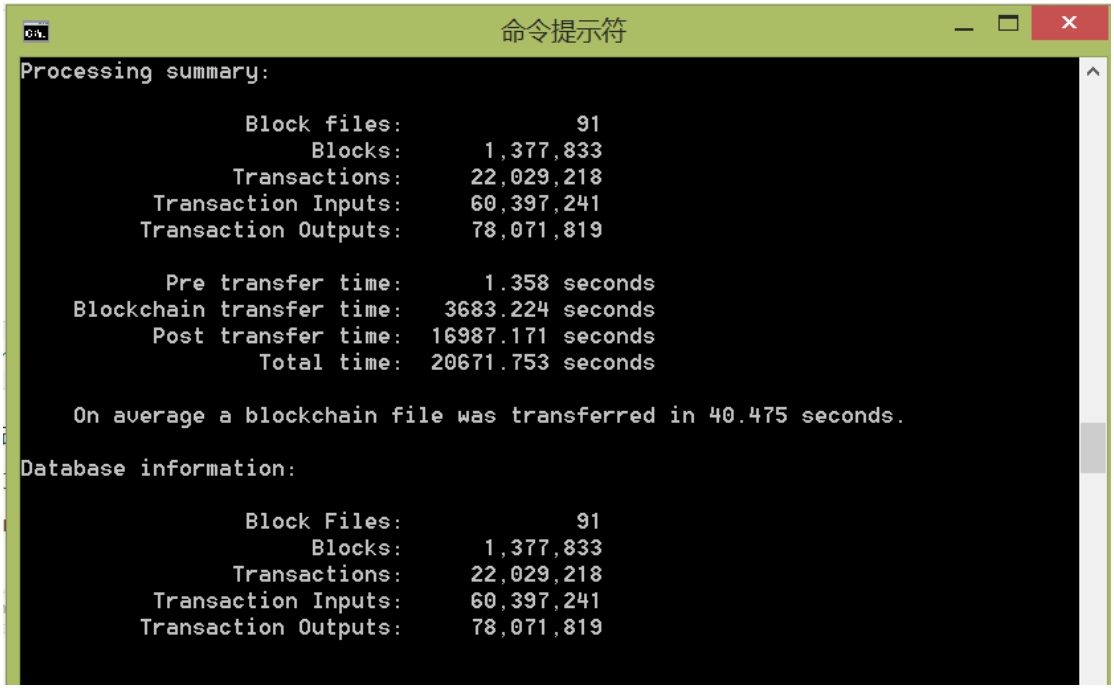
2. If your .dat files are transferred into database successfully.

   You will get following information (Here my blockchain is updated to rev00090.dat):

```
BitcoinDatabaseGenerator 1.9 [DEBUG]
Active threads: 4

Database "Litecoin2" was created.
Database schema was setup.
Database indexes were disabled.

    File: blk00000.dat. Transferring data completed in  49.861 seconds.
    File: blk00001.dat. Transferring data completed in  53.039 seconds.
    File: blk00002.dat. Transferring data completed in  50.919 seconds.
    File: blk00003.dat. Transferring data completed in  52.371 seconds.
    File: blk00004.dat. Transferring data completed in  40.561 seconds.
    File: blk00005.dat. Transferring data completed in  40.551 seconds.
    File: blk00006.dat. Transferring data completed in  41.322 seconds.
    File: blk00007.dat. Transferring data completed in  40.304 seconds.
    File: blk00008.dat. Transferring data completed in  40.773 seconds.
    File: blk00009.dat. Transferring data completed in  40.127 seconds.
    File: blk00010.dat. Transferring data completed in  39.157 seconds.
    File: blk00011.dat. Transferring data completed in  38.324 seconds.
    File: blk00012.dat. Transferring data completed in  42.942 seconds.
    File: blk00013.dat. Transferring data completed in  44.829 seconds.
    File: blk00014.dat. Transferring data completed in  44.462 seconds.
    File: blk00015.dat. Transferring data completed in  43.693 seconds.
    File: blk00016.dat. Transferring data completed in  37.833 seconds.
    File: blk00017.dat. Transferring data completed in  39.304 seconds.
    File: blk00018.dat. Transferring data completed in  41.186 seconds.
    File: blk00019.dat. Transferring data completed in  40.626 seconds.
    File: blk00020.dat. Transferring data completed in  39.360 seconds.
    File: blk00021.dat. Transferring data completed in  38.112 seconds.
    File: blk00022.dat. Transferring data completed in  40.097 seconds.
    File: blk00023.dat. Transferring data completed in  39.608 seconds.
    File: blk00024.dat. Transferring data completed in  42.151 seconds.
    File: blk00025.dat. Transferring data completed in  41.996 seconds.
    File: blk00026.dat. Transferring data completed in  41.326 seconds.
    File: blk00027.dat. Transferring data completed in  40.644 seconds.
    File: blk00028.dat. Transferring data completed in  41.475 seconds.
    File: blk00029.dat. Transferring data completed in  40.621 seconds.
    File: blk00030.dat. Transferring data completed in  41.838 seconds.
    File: blk00031.dat. Transferring data completed in  42.911 seconds.
    File: blk00032.dat. Transferring data completed in  41.903 seconds.
    File: blk00033.dat. Transferring data completed in  42.042 seconds.
    File: blk00034.dat. Transferring data completed in  40.343 seconds.
    File: blk00035.dat. Transferring data completed in  40.678 seconds.
    File: blk00036.dat. Transferring data completed in  39.633 seconds.
    File: blk00037.dat. Transferring data completed in  39.656 seconds.
    File: blk00038.dat. Transferring data completed in  39.650 seconds.
    File: blk00039.dat. Transferring data completed in  38.537 seconds.
    File: blk00040.dat. Transferring data completed in  38.974 seconds.
    File: blk00041.dat. Transferring data completed in  38.726 seconds.
    File: blk00042.dat. Transferring data completed in  39.130 seconds.
    File: blk00043.dat. Transferring data completed in  40.074 seconds.
```

```
    File: blk00044.dat. Transferring data completed in  39.793 seconds.
    File: blk00045.dat. Transferring data completed in  38.851 seconds.
    File: blk00046.dat. Transferring data completed in  38.032 seconds.
    File: blk00047.dat. Transferring data completed in  38.239 seconds.
    File: blk00048.dat. Transferring data completed in  41.026 seconds.
    File: blk00049.dat. Transferring data completed in  41.429 seconds.
    File: blk00050.dat. Transferring data completed in  38.132 seconds.
    File: blk00051.dat. Transferring data completed in  39.112 seconds.
    File: blk00052.dat. Transferring data completed in  39.987 seconds.
    File: blk00053.dat. Transferring data completed in  41.006 seconds.
    File: blk00054.dat. Transferring data completed in  41.377 seconds.
    File: blk00055.dat. Transferring data completed in  39.041 seconds.
    File: blk00056.dat. Transferring data completed in  40.180 seconds.
    File: blk00057.dat. Transferring data completed in  40.628 seconds.
    File: blk00058.dat. Transferring data completed in  40.858 seconds.
    File: blk00059.dat. Transferring data completed in  39.152 seconds.
    File: blk00060.dat. Transferring data completed in  42.109 seconds.
    File: blk00061.dat. Transferring data completed in  40.730 seconds.
    File: blk00062.dat. Transferring data completed in  40.675 seconds.
    File: blk00063.dat. Transferring data completed in  42.436 seconds.
    File: blk00064.dat. Transferring data completed in  41.712 seconds.
    File: blk00065.dat. Transferring data completed in  39.232 seconds.
    File: blk00066.dat. Transferring data completed in  39.688 seconds.
    File: blk00067.dat. Transferring data completed in  41.272 seconds.
    File: blk00068.dat. Transferring data completed in  38.779 seconds.
    File: blk00069.dat. Transferring data completed in  38.929 seconds.
    File: blk00070.dat. Transferring data completed in  39.834 seconds.
    File: blk00071.dat. Transferring data completed in  40.073 seconds.
    File: blk00072.dat. Transferring data completed in  42.075 seconds.
    File: blk00073.dat. Transferring data completed in  40.650 seconds.
    File: blk00074.dat. Transferring data completed in  39.660 seconds.
    File: blk00075.dat. Transferring data completed in  39.477 seconds.
    File: blk00076.dat. Transferring data completed in  37.580 seconds.
    File: blk00077.dat. Transferring data completed in  36.509 seconds.
    File: blk00078.dat. Transferring data completed in  38.117 seconds.
    File: blk00079.dat. Transferring data completed in  37.646 seconds.
    File: blk00080.dat. Transferring data completed in  38.122 seconds.
    File: blk00081.dat. Transferring data completed in  39.741 seconds.
    File: blk00082.dat. Transferring data completed in  39.566 seconds.
    File: blk00083.dat. Transferring data completed in  38.977 seconds.
    File: blk00084.dat. Transferring data completed in  38.834 seconds.
    File: blk00085.dat. Transferring data completed in  38.049 seconds.
    File: blk00086.dat. Transferring data completed in  38.724 seconds.
    File: blk00087.dat. Transferring data completed in  36.575 seconds.
    File: blk00088.dat. Transferring data completed in  37.693 seconds.
    File: blk00089.dat. Transferring data completed in  38.790 seconds.
    File: blk00090.dat. Transferring data completed in  32.433 seconds.

Database indexes were rebuilt successfully in 3699.472 seconds.
No stale blocks were found. The search took 61.157 seconds.
```

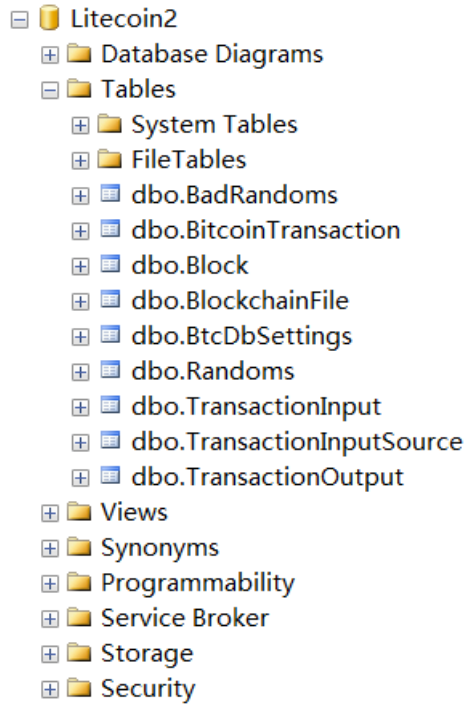
```
Setting direct links: inputs to source outputs (this may take a long time)... 2%
Setting direct links: inputs to source outputs (this may take a long time)... 11
Setting direct links: inputs to source outputs (this may take a long time)... 21
Setting direct links: inputs to source outputs (this may take a long time)... 30
Setting direct links: inputs to source outputs (this may take a long time)... 40
Setting direct links: inputs to source outputs (this may take a long time)... 49
Setting direct links: inputs to source outputs (this may take a long time)... 59
Setting direct links: inputs to source outputs (this may take a long time)... 68
Setting direct links: inputs to source outputs (this may take a long time)... 78
Setting direct links: inputs to source outputs (this may take a long time)... 87
Setting direct links: inputs to source outputs (this may take a long time)... 95
Setting direct links: inputs to source outputs completed in 13170.109 seconds.

Shrinking database files completed in 56.427 seconds.
```


命令提示符
Processing summary:
Block files: 91
Blocks: 1,377,833
Transactions: 22,029,218
Transaction Inputs: 60,397,241
Transaction Outputs: 78,071,819
Pre transfer time: 1.358 seconds
Blockchain transfer time: 3683.224 seconds
Post transfer time: 16987.171 seconds
Total time: 20671.753 seconds
On average a blockchain file was transferred in 40.475 seconds.
Database information:
Block Files: 91
Blocks: 1,377,833
Transactions: 22,029,218
Transaction Inputs: 60,397,241
Transaction Outputs: 78,071,819

PART 3. Analyze the generated database

1. To create the table of random and table of bad random, I use the SQL script (https://github.com/JasonPap/Reused-Bitcoin-Numbers/blob/master/SQL/RandomsToTables.sql). I made small modification on this script to be suitable for my generated database.

2. Then the new database is shown as below:



There are two columns in the dbo.Randoms:

TansactionInputId and Random

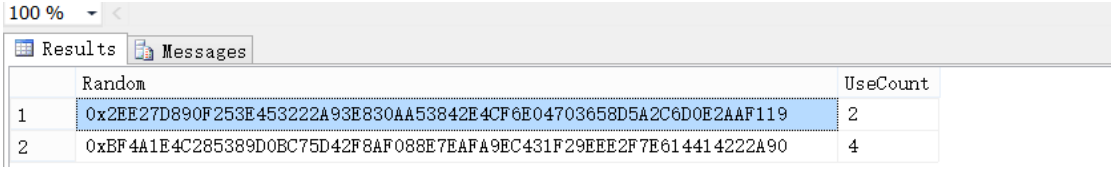
100 %

Results | Messages

	TransactionInp...	Random
1	17268017	0x00800017B228D6DA086300D66FB521DC155474F1953C8...
2	26965330	0x0080001A842CD9B03EA6FF48A7B87C6DFC26450708F07...
3	37483948	0x0080001AC2B8D9EDE365D2F721723A6A417DFE1CA2B43...
4	14897997	0x0080002FBB50B84FAE58AAB9FDD84263EFDEAB4B8A378...
5	26518488	0x00800031B6EF219F1C483FE211AF026D438788ADDF9D0...
6	9497840	0x00800032A81364BCEBAD9CE9DB48B9C4151B553FA2A35...
7	38855518	0x0080003320520195AB4FAA3FFE3D39DF7EA7B1FC1A301...
8	17951446	0x00800036B87CC3DB9D89D3381ED2B5C0872431AD01FA0...
9	38809814	0x00800044A782E7F3802759E87DF097D804959DA64ECDD...
10	57055890	0x00800059D9D21A5139D8FFE69530D842DC458F448BDEF...
11	25438793	0x0080005E8F5A7D5EBD1277D10053AAF00F1684FF8AE47...
12	20013793	0x0080005F02F1E5111E4C959859E104E4A0F4D939E03C2...
13	32962834	0x00800060894FC246011429762E45B2477536BD4D4F7DE...
14	33349877	0x008000652A5A2F20E4F6BC888F286910ACFBE5B1B31FA...
15	35201942	0x0080007F60D50AF06F8BA2DE1E3360D4BFE13B949CDFB...

There are two columns in the dbo.BadRandoms:

Random and UseCount



	Random	UseCount
1	0x2EE27D890F253E453222A93E830AA53842E4CF6E04703658D5A2C6D0E2AAF119	2
2	0xBF4A1E4C285389D0BC75D42F8AF088E7EAFA9EC431F29EEE2F7E614414222A90	4

3. Then it is necessary to relate the BadRandoms with other corresponding information.

I write SQL script:

SELECT [TransactionInputId] ,[Random] FROM [Litecoin2].[dbo].[Randoms]

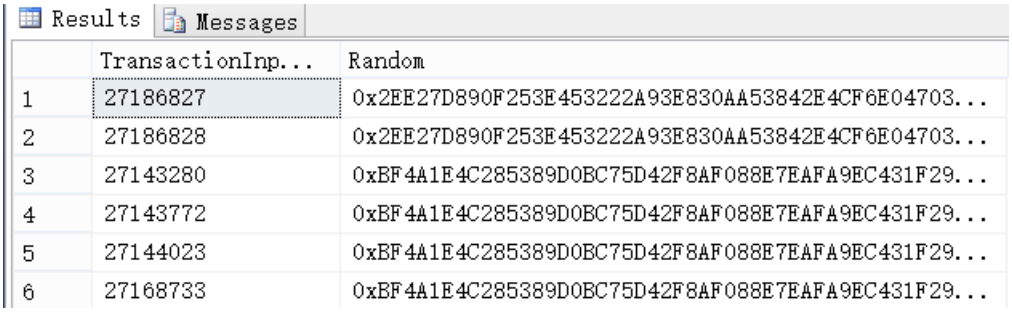
WHERE

Random = 0x2EE27D890F253E453222A93E830AA53842E4CF6E04703658D5A2C6D0E2AAF119

OR

Random = 0xBF4A1E4C285389D0BC75D42F8AF088E7EAFA9EC431F29EEE2F7E614414222A90

Then I can get:



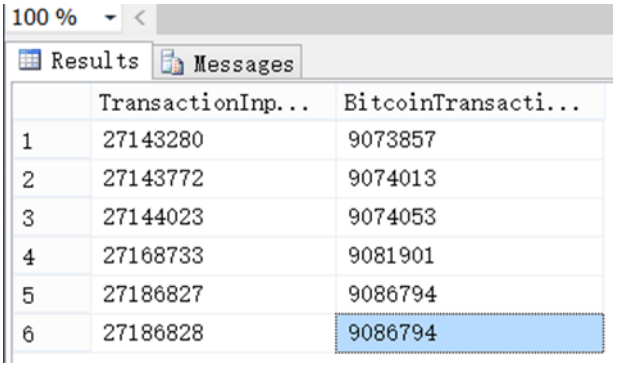
	TransactionInp...	Random
1	27186827	0x2EE27D890F253E453222A93E830AA53842E4CF6E04703...
2	27186828	0x2EE27D890F253E453222A93E830AA53842E4CF6E04703...
3	27143280	0xBF4A1E4C285389D0BC75D42F8AF088E7EAFA9EC431F29...
4	27143772	0xBF4A1E4C285389D0BC75D42F8AF088E7EAFA9EC431F29...
5	27144023	0xBF4A1E4C285389D0BC75D42F8AF088E7EAFA9EC431F29...
6	27168733	0xBF4A1E4C285389D0BC75D42F8AF088E7EAFA9EC431F29...

Similarly, I can use the related column to find the other information related with Random.

It can be summarized as:

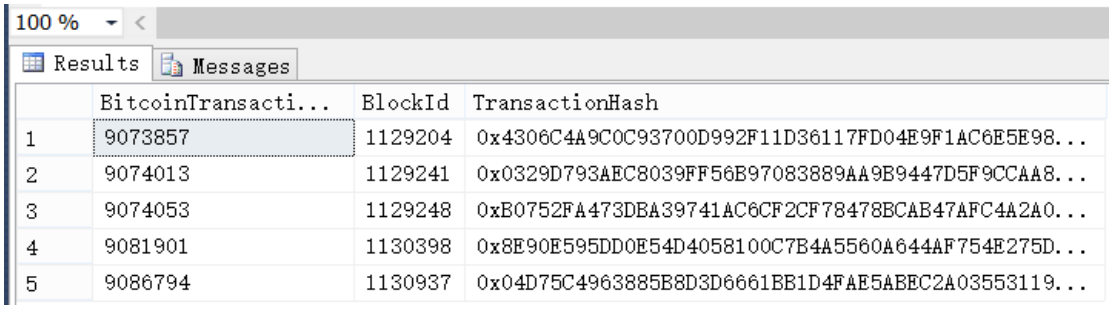
Use value of Random to get value of TransactionInputId in dbo.Randoms

⇨ Use value of TransactionInputId to get value of BitcoinTransactionId in dbo.TransactionInput



	TransactionInp...	BitcoinTransacti...
1	27143280	9073857
2	27143772	9074013
3	27144023	9074053
4	27168733	9081901
5	27186827	9086794
6	27186828	9086794

⇨ Use value of BitcoinTransactionId to get value of TransactionHash in dbo.BitcoinTransaction



100 %

Results | Messages

	BitcoinTransacti...	BlockId	TransactionHash
1	9073857	1129204	0x4306C4A9C0C93700D992F11D36117FD04E9F1AC6E5E98...
2	9074013	1129241	0x0329D793AEC8039FF56B97083889AA9B9447D5F9CCAA8...
3	9074053	1129248	0xB0752FA473DBA39741AC6CF2CF78478BCAB47AFC4A2A0...
4	9081901	1130398	0x8E90E595DD0E54D4058100C7B4A5560A644AF754E275D...
5	9086794	1130937	0x04D75C4963885B8D3D6661BB1D4FAE5ABEC2A03553119...

To conclude, the result I get is:

BadRandoms	TransactionHash
0x2EE27D890F253E453222A93E830AA53842E4CF6E04703658D5A2C6D0E2AAF119	04d75c4963885b8d3d6661bb1d4fae5abec2a035531192c67626f2245979f3b1
0x2EE27D890F253E453222A93E830AA53842E4CF6E04703658D5A2C6D0E2AAF119	04d75c4963885b8d3d6661bb1d4fae5abec2a035531192c67626f2245979f3b1
0xBF4A1E4C285389D0BC75D42F8AF088E7EAFA9EC431F29EEE2F7E614414222A90	4306C4A9C0C93700D992F11D36117FD04E9F1AC6E5E98460CE9B8C8E7E7B5B4E
0xBF4A1E4C285389D0BC75D42F8AF088E7EAFA9EC431F29EEE2F7E614414222A90	0329D793AEC8039FF56B97083889AA9B9447D5F9CCAA8ED1D74B131397E2887B
0xBF4A1E4C285389D0BC75D42F8AF088E7EAFA9EC431F29EEE2F7E614414222A90	B0752FA473DBA39741AC6CF2CF78478BCAB47AFC4A2A0FB865E2E28336229068
0xBF4A1E4C285389D0BC75D42F8AF088E7EAFA9EC431F29EEE2F7E614414222A90	8E90E595DD0E54D4058100C7B4A5560A644AF754E275D44CCB9A23E9162CD985