The method of design of an own cryptocurrency based on the dash project

D.D. Tsvetkov

M.Sc., MAI (NRU)
dima121596@mail.ru

1 Introduction

This article is devoted to the method of design an own cryptocurrency based on the source codes of the Dash project with the function of anonymous payments as an instructional material for people interested in exploring the possibilities of Blockchain technology in cryptocurrencies.

The cryptocurrency Dash is an open and decentralized payment system, based on Blockchain technology and on Bitcoin cryptocurrency source codes. A distinctive feature is the use of the PrivateSend mechanism, which allows to anonymize transactions of a user.

2 List of toolkits

1. Operating system Windows 10 – a host for virtual machines (VM).
2. Oracle VM VirtualBox 5.1 – software for virtualization.
3. Operating system Debian 8.8 – for VM with compiler.
4. Operating system Windows 7 – for testing the wallet program.
3. Notepad++ v7.4 – a text editor and source code editor.
4. PuTTY 0.69 client – for management the Debian-VM.
5. FileZilla 3.25 client – access to files of the VM-compiler Debian.
6. Account on the version control repository GitHub. The GitHub Desktop client.
7. The source codes of the Dash cryptocurrency v0.12.0.60 [1].

3 Description of the design

First, necessary to install and configure the toolkit.

VM-compiler Debian parameters:

1. Disk space – 100 GB.
2. CPU – 4.
3. Base memory: 4096 MB.
4. Network – «NAT». In the section «Advanced» – «Port forwarding»: «Name» – SSH, « Protocol « – TCP, «Host IP» – 127.0.0.1, «Host Port» – 22222, «Guest IP» – 10.0.2.15, « Guest Port» – 22.

Windows 7-VM parameters (necessary two VMs for testing the wallet program):

1. Disk space – 50 GB.
2. CPU – 2.
3. Base memory – 1024 MB.
4. Network – «Internal».

At the end of the settings, the created virtual machines must be put in operation and the operating systems (OS) must be installed.

After the installation and toolkit's configuration it is necessary to start the PuTTY client and connect to the Debian-VM. The PuTTY specifies: «Host name» – 127.0.0.1, «Port» – 22222, «Connection type» – SSH.

Once connected, it is necessary to switch superuser mode with the su command and fulfill the following actions:
1. Update the software packages index and download package updates:

```
apt-get update && apt-get upgrade
```

2. Install the necessary software packages for the compiler:

```
apt-get install git ruby sudo apt-cacher-ng qemu-utils
debootstrap python-cheetah python-dev python-pip py-
thon2.7-dev lxc libssl-dev parted kpartx bridge-utils
make ubuntu-archive-keyring curl
```

3. Assign to the user partial superuser privileges (ex., debian):

```
adduser debian sudo
```

4. Setting up LXC:

```
echo «%sudo ALL=NOPASSWD: /usr/bin/lxc-start» >
/etc/sudoers.d/gitian-lxc
echo '#!/bin/sh -e' > /etc/rc.local
echo 'brctl addbr br0' >> /etc/rc.local
echo 'ifconfig br0 10.0.3.2/24 up' >> /etc/rc.local
echo 'iptables -t nat -A POSTROUTING -o eth0 -j
MASQUERADE' >> /etc/rc.local
echo 'echo 1 > /proc/sys/net/ipv4/ip_forward' >>
/etc/rc.local
echo 'exit 0' >> /etc/rc.local
echo 'export USE_LXC=1' >> /home/debian/.profile
echo 'export GITIAN_HOST_IP=10.0.3.2' >>
/home/debian/.profile
echo 'export LXC_GUEST_IP=10.0.3.5' >>
/home/debian/.profile
reboot
```

After reboot the VM, it is necessary to reconnect to it. Then run the commands:
1. Preparing the compiler:

```
wget http://archive.ubuntu.com/ubuntu/pool/universe/v/vm-
builder/vm-builder_0.12.4+bzr494.orig.tar.gz
tar -zxvf vm-builder_0.12.4+bzr494.orig.tar.gz
cd vm-builder-0.12.4+bzr494
sudo python setup.py install
cd ..
```

2. Downloading and installing the compiler of the wallet program:

```
git clone https://github.com/devrandom/gitian-builder.git
cd gitian-builder
bin/make-base-vm --lxc --arch amd64 --suite precise
cd ..
```

3. Downloading and installing the genesis blocks generator for the future cryptocurrency and the xcoin-hash module (for the algorithm X11):

```
sudo pip install scrypt construct==2.5.2
git clone https://github.com/lhartikk/xcoin-hash
cd xcoin-hash
sudo python setup.py install
cd ..
git clone https://github.com/lhartikk/GenesisH0
```

Genesis block is the initial block of a blockchain, which is always hard-coded in the source code.

After preparing the VM-compiler, we should proceed to the design of cryptocurrency by editing the Dash source code.

Initially, we must login to GitHub hosting account and go to the official Dash repository [2]. There choose «Fork» for cloning the repository. Next, it is necessary to rename the cloned repository («Settings» – «Repository name»), then download the repository to the computer (in GitHub Desktop: «Create, clone or add a local repository», select the cloned repository and click «Clone»). After successful downloading, it is necessary to unzip the files from the archive with the Dash 0.12.0.60 source code to the folder with the downloaded repository with the full replacement of all files.

While editing the source code, it is necessary to change the repository address and the name of cryptocurrency Dash to its own (in this example, is used the name «Flowercoin»). For this action, in Notepad++: «Search» – «Find» – «Find in files», we select the directory with the saved repository from GitHub and set the parameters: «Filters» – «*.*», «Match case», «In all sub-folders», «In hidden folders».

It is necessary to replace:

1. «github.com/dashpay/dash» on «github.com/dmrtsvetkov/flowercoin».
2. Repository name: «DASHPAY», «Dashpay», «dashpay» on «DMRTSVETKOV», «Dmrtsvetkov», «dmrtsvetkov»
3. Name of the cryptocurrency: «DASH», «Dash», «dash» on «FLOWERCOIN», «Flowercoin», «flowercoin».

During the replacement, it is necessary to make sure that there are no files with encoding errors in the search results (in other case, files should be copied to another directory, renamed and moved back with a replacement).

Then in the folder with repository we must change the word «dash» on the word «flowercoin» in all names of files and folders. The folders «dashpay» must be renamed as «dmrtsvetkov». Далее в папке с репозиторием изменить во всех названиях файлов и папок слово «dash» на «flowercoin». Папки «dashpay» следует переименовать в «dmrtsvetkov».

After renaming, one can replace the Dash cryptocurrency logos and icons with own in the following folders:

FOLDER_WITH_REPOSITORY\flowercoin\src\qt\res\icons
FOLDER_WITH_REPOSITORY\flowercoin\src\qt\res\images
FOLDER_WITH_REPOSITORY\flowercoin\share\pixmaps

Next, it is necessary to generate digital signatures for genesis blocks generation. The algorithm used in the generation of signatures is ECDSA. To do this, the following commands are run in the Debian VM:

```
# generate signature
openssl ecparam -genkey -name secp256r1 –out main.pem
openssl ecparam -genkey -name secp256r1 -out mainal-
ert.pem
openssl ecparam -genkey -name secp256r1 -out test-
netalert.pem
openssl ecparam -genkey -name secp256r1 -out main-
spork.pem
openssl ecparam -genkey -name secp256r1 -out test-
netspork.pem
# output of generated signatures in text
openssl ec -in main.pem -noout –text
openssl ec -in mainalert.pem -noout -text
openssl ec -in testnetalert.pem -noout -text
openssl ec -in mainspork.pem -noout -text
openssl ec -in testnetspork.pem -noout -text
```

The results achieved should be copied from the command line to a text file. In this file we need to remove all the colons and align the keys in one line.

After the generation of all signatures, genesis blocks for the future cryptocurrency are generated. To do this, on the Debian command line (for the main block):

```
cd GenesisH0
python genesis.py -a X11 -z «I created flowercoin
01/05/2017 for my dissertation» –p
«047e30c6c8304ac9f6f100e9eb00976e751b38
e3af46989cc8e1927a944c65613c3c97c550aef46cc9f2da62ebed9b5
bb010afc1314a175d8d4d5e2c02caa8792» -t 1495303200
```

where parameter `-a` – hash algorithm; `-z` – pszTimestamp – the input content of the transaction generating the genesis block (an arbitrary phrase, usually the title of the news); `-p` – the public key of the signature «main»; `-t` – time of genesis block generation in unix (is converted into special services [3]), the launch of the wallet program and the generation of the first block must be done no later than one hour after the specified time.

In addition to the main genesis block of the main network, it is also necessary to generate the testnet and regtest genesis blocks:

```
python genesis.py -a X11 -z «I created flowercoin
01/05/2017 for my dissertation» –p
«047e30c6c8304ac9f6f100e9eb00976e751b3
8e3af46989cc8e1927a944c65613c3c97c550aef46cc9f2da62ebed9b
5bb010afc1314a175d8d4d5e2c02caa87924» -t 1495303201
python genesis.py -a X11 -z «I created flowercoin
01/05/2017 for my dissertation» –p
«047e30c6c8304ac9f6f100e9eb00976e751b38
e3af46989cc8e1927a944c65613c3c97c550aef46cc9f2da62ebed9b5
bb010afc1314a175d8d4d5e2c02caa87924» -t 1495303202 -b
0x207fffff
```

The results achieved after the commands'performing should be copied to a text file.

Next, the source code is edited. Initially, the file chainparams.cpp is edited, which is responsible for the main network parameters. This file is located in: FOLDER_WITH_REPOSITORY\flowercoin\src\.

Initially, the checkpoints are changed – static Checkpoints (check points, preventing DoS-attacks, attempts of cluttering with unsuitable blocks, and attacks with isolation of network nodes).

First checkpoint (main network) example:

1. In the `MapCheckpoints` the values 1500, 4991, 9918, … are the numbers of the last generated blocks in the network at the moment of checkpoint. The value in parentheses after 0x (if any is in existance) is the hash value of the block. Here it is necessary to delete all the lines except the first one, change the number 1500 to 0 (since in the new network only one block is generated so far – zero, which is also the genesis block). The hash value after 0x is replaced by the one generated for the genesis block «main», in this example – 00000f1a26168 dce5b83f48002ce8a29933398535c92709163a5ac4be045592.
2. In the CCheckpointData the first value is the generation time of the last block-checkpoint in the unix format, equal to the generation time of the genesis block – 1495303200 (for the regtest network it is necessary to set the parameter value as 0). The second value is the total number of transactions performed since the generation of the genesis block till the time of the block-checkpoint generation – 0, since transactions have not yet been done. The third value is the estimated number of transactions per day after the checkpoint, one can leave it as it is.

   Similarly, the checkpoints for the testnet and regtest networks are edited.

Next, in the same file, the network parameter code is edited – public CChainParams. Main network example:

1. The Public key for translation the alerts over the network (`vAlertPubKey`) – The «pub» key of the previously generated signature «mainalert», in this example: 04dfe1ae214b0332437db6517f7f013c62e6d9c154659f9712bc5a6fe2 6454c74611a492287cee9ba35236bb11bee0b227ef96375ce5e36a88c1d6d4ea087a 7e8a (for testnet network – the key of the signature «testnetalert»; for regtest is not needed).
2. The port used for connection on the network (`nDefaultPort`) needs to be replaced to any officially unused port; in this example the port 1993 is used (11993 for testnet and 19931 for regtest).
3. Conversion of the complexity of mining (`nTargetTimespan`) can be edited once per 10 hours, the block generation time (`nTargetSpacing`) is 1 block per minute.
4. The content of the input of the transaction of the generating genesis block (`const char* pszTimestamp`) is «I created flowercoin 01/05/2017 for my dissertation». The public key main (`scriptPubKey`) – is the «pub» key of the previously generated signature «main» (047e30c6c8304ac9f6f100e9eb00 976e751b38e3af46989cc8e1927a944c65613c3c97c550aef46cc9f2da62ebed9b5bb 010afc1314a175d8d4d5e2c02caa87924). Generation time of the genesis block (`genesis.nTime`) – 1495303200. A simplified coded threshold below which the header of the block (`genesis.nBits`) should be: 0x1e0ffff0 for main and testnet, and 0x207fffff for regtest networks. The first value of the «nonce» field to be used for the hash search of the block (`genesis.nNonce`) should be changed to the value obtained after generation of the genesis block. In this example, the value is 1428875 (for main network).
5. The hash value of the genesis block (`hashGenesisBlock`) «main» is 00000f1a26168dce5b83f48002ce8a29933398535c92709163a5ac4be045592. The hash value of the block tree (`genesis.hashMerkleRoot`) is b7ca21f27081b001872e697fef9ee2ac25666e88414144c6ab0849cae9837454. The hash values are obtained from the previously generated genesis block. Values must be replaced after 0x (if any is existed).
6. DNS-servers `vSeeds.push_back` are removed using the replace to «`vSeeds.clear();`».
7. The public key «spork» (`strSporkKey`) is required to prevent branching of the network in the event that part of the nodes is not updated to the current version of the wallet. The public key «spork» and the public key «spork» for masternodes (`strMasternodePaymentsPubKey`) are the same, their value was generated previously in the signature of mainspork (ключ «pub»): 04c03a20082e774c6c93da0251e00ed00e73b28d2f3bbb692b7ab7ec64a8a7b9374c 16afa15b6d7355fc482b5aa6b8fc7ec4d0fb71fcc21bdd9b07bf06a7c9d611 (for testnet – in the signature of testnetspork; for regtest is not needed). The time for the start of payments to masternodes (`nStartMasternodePayments`) is set to the same as the creation time of the genesis block – 1495303200.

   Similarly, the network's parameter testnet and regtest are edited.

Now it is necessary to replace the ports in the other source codes. In order to do this, by analogy with renaming, we use Notepad ++. The ports 9999, 19999, 19994 are need to be replaced to their own (in this example: 1993, 11993, 19931). It is also need to replace the ports for RPC: 9998 and 19998 with unoccupied ones. For example: 3991 and 39911. During the search, it should also be set the parameter «Match whole word only» and we must attentively look through the search results.

When all the necessary editions in the source code have been performed, we need to compile the wallet program. First of all, we must download all the changes to own repository on GitHub. To do this, in the GitHub Desktop in the «Changes» tab: the «Summary» and «Description» fields are filled in, then one must click «Commit to master» – «Sync».

To compile a wallet program in the Debian-VM console, the following commands must be run:

```
git clone https://github.com/dmrtsvetkov/flowercoin.git
cd gitian-builder
make -C ../flowercoin/depends download
SOURCES_PATH=`pwd`/cache/common
./bin/gbuild --commit flowercoin=master
../flowercoin/contrib/gitian-descriptors/gitian-win.yml
```

When the compilation is completed we should connect to the Debian-VM with the help of FileZilla client and download the archive with the compiled wallet program. «Host» for connect – sftp://127.0.0.1; «Port» – 22222. The compiled files are in the folder: /gitian-builder/build/out.

4 Testing of wallet program

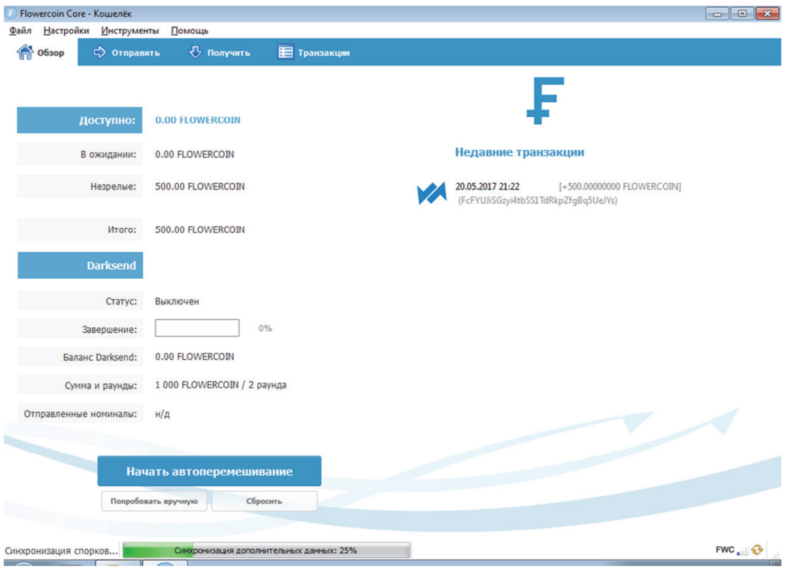
Testing of the compiled wallet program is performed on pre-prepared virtual machines running Windows 7.

After running all the virtual machines, we must set the network parameters at each of them: «IP address» – 10.0.1.10 (11) – depending on the VM number; «Subnet mask » – 255.255.0.0; «Default gateway» – 10.0.1.10; «Preferred DNS server» – 10.0.1.10, and enable network discovery.

Then, the wallet program on all VMs can be run. To do this, we run flowercoin-qt.exe in the folder with the installed program at each VM and select the location for storing the wallet settings. After starting the program, it is necessary to edit the file flowercoin.conf, located in the created directory with the settings, by writing to it:

```
# start of mining
gen=1
# node synchronization
addnode=10.0.1.10
```

After saving the file, we must restart the wallet program. During a minute after restart, the mining will begin and a reward will appear for the first generation blocks (fig. 1).



Доступно:	0.00 FLOWERCOIN
В ожидании:	0.00 FLOWERCOIN
Незрелые:	500.00 FLOWERCOIN
Итого:	500.00 FLOWERCOIN

Darksend

Статус:	Выключен
Завершение:	0%
Баланс Darksend:	0.00 FLOWERCOIN
Сумма и раунды:	1 000 FLOWERCOIN / 2 раунда
Отправленные номиналы:	н/д

Недавние транзакции

20.05.2017 21:22 [+500.00000000 FLOWERCOIN]
(FcFYUJiSGzyi4tbSS1TdRkpZfgBq5UeJYs)

Синхронизация спорков... Синхронизация дополнительных данных: 25%

Fig. 1. Start of the mining and the first transaction with reward

References

1. Release v0.12.0.60. Available at: https://github.com/dashpay/dash/releases/tag/v0.12.0.60 (Accessed 12.03.2017).
2. GitHub – dashpay/dash. Available at https://github.com/dashpay/dash/. (Accessed 12.03.2017).
3. Epoch Unix Time Stamp Converter. Available at: http://www.unixtimestamp.com/. (Accessed 12.03.2017).
4. Dash Official Website. Available at: https://www.dash.org. (Accessed 12.03.2017).
5. Dash: Official Documentation. Available at: https://dashpay.atlassian.net/wiki/display/DOC/Official+Documentation. (Accessed 12.03.2017).