Integration of Namecoin in OpenBazaar

Chara Podimata*, Kostis Lolos†
*National Technical University of Athens - charapod@gmail.com
†National Technical University of Athens - lolos.kostis@gmail.com

***Abstract*—In this paper, we designed the integration of the open source project OpenBazaar with the cryptocurrency Namecoin. OpenBazaar is a project, a marketplace better to create a decentralized network for peer to peer commerce online—using Bitcoin—that has no fees and no restrictions. Namecoin, on the other hand, is a kind of cryptocurrency, the first fork of Bitcoin and its basic difference from it is that we can store data in it. With this integration, we give the opportunity at OpenBazaar users to declare a unique name for their store and we reinforce the security of transactions, due to the adding of Namecoin in the messages that the nodes exchange.**

Keywords – Namecoin, Bitcoin, OpenBazaar, Cryptography, Cryptocurrencies, Censorship

I. Introduction

The OpenBazaar, as a project, can by itself, solve two of the three problems of the Zooko triangle; decentralization and security. Namecoin helps as solve the third problem as well; human readability. The trust in the certification of every node does not depend (only) on the GUID of this node, but it also depends on the name that the user has declared; and this name cannot be changed, because it was created depending on the Namecoin of the node. Thus, we can visit the same store, with the same name in the future.

Below we give detailed explanations for the cryptocurrency Namecoin and its "father", Bitcoin, as long as for the OpenBazaar. In order to work on this project, we needed to use github, since OpenBazaar in an opensource project, which is currently being developed by many developers around the world. The technical features of this project are shown below.

II. Bitcoin

A. One more currency

Bitcoin is one more currency through which someone can acquire goods and services. However, it is different from the usual currencies. Bitcoin is an open-source and distributed payment system, a "banking system" that runs in the computers of its users.

The basis for its creation was set by Satoshi Nakamoto in 2008 [1], in a paper describing in theory how such a system could work. Its first implementation followed just one year later, and since then a large number of similar currencies have followed, competing for a place in our wallets.

Bitcoin is the most popular among these currencies, with its total market capitalization exceeding US$10 billion in November 2013, while at the same time the number of companies that sell their products and services in exchange for Bitcoins keeps increasing.

B. Bitcoin Transactions

The owner of a Bitcoin can transfer it to a new owner by signing a hash of the transaction through which he acquired it together with the public key of the new owner, using his own private key. This way, anyone can confirm that the transaction is valid since it is signed with the private key of the old owner, and the new owner can now spend the coin simply by signing the next transaction with his own private key.

The problem, of course, is that this method does not prevent a Bitcoin owner from spending the same coin in multiple transactions. To solve this problem, the idea of the *Block Chain* was implemented. Each transaction that gets validated is recorded in a link of a chain, where each of those links contains a hash of the previous one. This way, the chronological order of the transactions is unambiguously determined, while at the same any attempt to alter the history of transactions would result in changing the hashes of all the following links. In other words, this chain is an immutable global history of transactions.

In order to make sure that there can be no two different versions of the chain, each block must necessarily contain a proof of an expensive computation. This idea is known as a *proof-of-work*. Assuming that the majority of the users in the system are honest and have no interest in harming it, if at any point there exist two different versions of the block chain we can always accept the longest chain as the valid one. Since the length of the chain translates to a proportional amount of computation to construct it, it is reasonable to assume that the longest chain is the result of the work of the majority of the users (which possesses the majority of the computational power), and not a small minority attempting to cheat the system.

C. Conclusions

The effect of Bitcoin in the way transactions work compared to traditional currencies is significant. First of all, since the only thing a user needs to transfer money is a cryptographic key-pair, he can use a new key in each transaction. This way, Bitcoin has the potential to provide anonymity since it is not generally possible to know who the owner of a key is, with the positive and negative consequences this entails.

Additionally, the fact that transactions are performed by distributed open-source software instead of central banks allows it to work with significantly lower transaction fees. This, combined with the fact that Bitcoin's anonymity allows for tax-free transactions, makes Bitcoin an attractive payment method for a large number of companies.

Finally, as expected for a new and radically different currency, its value fluctuates significantly, which makes it a risky but potentially profitable investment.

III. Namecoin

A. Making a good idea better

The first official "child" of Bitcoin was created 3 years later, and its name is Namecoin. Namecoin does not only share the same ideas and characteristics with Bitcoin, but also shares most of the code that implements it (it is the first *fork* of Bitcoin and they differ by only 400 lines of code).

The main difference between the two is the fact that Namecoin gives its users the ability to register names, and store information bound to them in the Block Chain. This allows Namecoin to work as a distributed name space, similar to DNS, which offers its users the additional safety, independence, flexibility and anonymity inherited from Bitcoin.

B. Differences from Bitcoin

In order for Namecoin to offer its services, it uses an independent block chain, as well as a series of new commands that can be used to create and renew the names it hosts. The names themselves are represented internally as special coins that cannot be spent for ordinary transactions, but can be transfered from one user to another as regular coins.

Additionally, in order to increase the incentive of the *miners* (this is how the users that construct the links of the block chain are called), Namecoin allows them to search for hashes that are aimed at multiple block chains, a feature known as *merged mining*.

At the same time of course, Namecoin is itself an independent currency, and can be used to purchase goods and services just like Bitcoin.

C. The .bit domain

A necessary ingredient in the functionality of the Internet is the existence of a *domain name system*, which is allows the translation of easily understandable and memorizable website names into IP addresses. This is the responsibility of ICANN (Internet Corporation for Assigned Names and Numbers), which manages the mapping between names and addresses, and orchestrates several thousand DNS servers around the world.

One of the most characteristic consequences of the creation of Namecoin was the existence of an alternative way to achieve the same goal. Since, using the Namecoin, each user can store in the Block Chain names and information, he can use it as a distributed and non-centrally controlled Domain Name System. This domain today exists and is called the .bit domain, basing its functionality entirely on Namecoin.

The advantages of this domain are plenty and significant. Information about the addresses of webpages under the .bit domain are spread and stored in the computers of all users, while the coordination and updating of the information is done not by a central authority, but instead of Namecoin's distributed network. This way, not only is it harder for someone to forge the information about the address of a webpage, since he cannot accomplish this simply by gaining access to the records of a DNS server, but it is also much harder for any of this information to be censored. Additionally, the fact that information is stored locally in the users' computers, greatly reduces the search time for an address. Finally, being an open source project, it requires a much lower fee in order to purchase a name.

IV. OpenBazaar

A. What is OpenBazaar

OpenBazaar is an opensource project for the creation of a decentralized network for peer-to-peer commerce - using Bitcoin - which does not involve any fees or restrictions. To put it simple, OpenBazaar is the child of eBay and BitTorrent.

At the time being, internet commerce is synonym to centralized services. eBay, Amazon and other big companies have restrictive policies and charge fees for the placement and the selling of items. They, also, accept only forms of payment which charge both the sellers and the buyers, such as credit cards of debit ones. They require from the users their personal information, something which may lead to information stolen or sold to others for profit or advertising. Buyers and sellers are not always free to exchange items and services, since companies and governments control all sorts of commerce.

OpenBazaar is a different approach to online commerce. It gives the power back to the users. Since nobody intervenes in the transactions of the users, there are no fees, nobody can censor a transaction and one can reveal only the information that he/she chooses to.

B. How does OpenBazaar work

Let's say you want to sell your old laptop. Using the client of OpenBazaar (a program which everyone can download), you can create a description for this item, with every detail he/she wants, just like he/she would do in any other online commerce service. The difference is that the price offered by the seller is not in euros or in dollars, but in Bitcoin. When this description is published, it is sent to the distributed p2p network which is constructed from other users of the OpenBazaar. Whoever searches for the keywords that you have used (ie laptop, electronics etc) will find the description of your laptop. Then, they can accept your price or ask for another one.

If you both agree to a price, the client creates a contract between you both with your digital signatures, and sends it to a third party called a notary. In case there is a conflict, an arbiter can come and help the transaction. These notaries and arbiters are also folks on the OpenBazaar network—could be your neighbor or someone across the world—who the buyer and seller trust in case something goes wrong. The third party witnesses the contract and creates a multisignature Bitcoin account (multisig) that requires two of three people to agree before the Bitcoin can be released.

C. And what if something goes wrong?

What if you're buying a certain book from a seller, you pay the multisig, and they ship you the wrong one, or it was in poorer condition than advertised, or they don't even send a product at all?

This is where the third party comes in. Remember that a multisig requires two of three people to agree in order to move the Bitcoin. They control the third key to the multisig, so the funds will not move until either the buyer and seller work out an arrangement themselves, or the third party agrees with either the buyer or seller on how to deal with the transaction and funds in multisig.

How can you trust the third party to begin with? For that matter, how can you trust anyone on a network that guards users' privacy, thus allowing for pseudo-anonymity? OpenBazaar has a reputation and rating system that allows all parties to give feedback on other users. If someone attempts to scam another user, their reputation will suffer, and the same is true with third parties. When you go to purchase a product and select a third party for the transaction, you can see their reputation and ratings to find out if the rest of the community trusts them—or doesn't. Ensuring that these ratings are legitimate and the reputation system isn't being gamed is a difficult technical challenge, but we will talk about that later.

These steps may sound complicated, but the details are handled by the client itself. The goal of the creators of OpenBazaar is for buyers and sellers to have an even better experience using OpenBazaar than the old centralized platforms.

We will present now some information concerning a technique used in OpenBazaar, Reputation Pledges, which are an implementation of proof-of-burn. We will describe proof-of-burn, why it is necessary and how can one use Reputation Pledges.

D. Proof-of-burn

Proof-of-burn is a term used to describe the intentional and provable destruction of bitcoin for a particular purpose. When engaging in a proof-of-burn, funds are intentionally sent to an address that is unspendable, meaning those coins are gone forever.

Why would someone destroy bitcoin on purpose? In the past, this has primarily been used to bootstrap one cryptocurrency from another. Distribution of a new cryptocurrency can be determined by people burning their coins in exchange for the new currency, thus showing they are invested.

E. Reputation Pledges

OpenBazaar implements proof-of-burn in a different way. On the OpenBazaar network, users can choose to be pseudonymous, meaning you don't know their true identity. As such, it can sometimes be difficult to determine if they are trustworthy or should be avoided. A reputation system is important to help inform the network of which participants have acted honestly in the past, and which haven't. There are several facets to the OpenBazaar reputation system, which is still being built, and you can read about the overall system here.

One part of this system is Reputation Pledges. This means that a user has chosen to prove their commitment to their OpenBazaar identity by burning a certain amount of bitcoin. The act of burning coins shows the network that the user is committed to their identity because they've now expended resources on it, and if they incur a negative reputation then those resources will have gone to waste.

To help understand the importance of this, consider a similar example in the real world. Travelling salesmen were often treated with skepticism by the inhabitants of the towns they visited. Apart from the annoyance of their house calls, why would people be reluctant to purchase items from travelling salesmen? Two reasons. One, they cannot rely on reputation to determine if the salesman is selling quality merchandise or not; the other customers of this salesmen aren't located nearby. Two, the salesman has nothing to lose if their products turn out to be poor quality – there is no reputation damage if they leave, and since there is no brick-and-mortar store they've invested in, they can simply peddle their wares elsewhere.

You can think of it similarly to OpenBazaar users. If there's no cost to creating a new identity, or if there is no brick-and-mortar store to keep you in one place, you can simply abandon an identity once it has received negative feedback. Obviously online, you don't have a physical store, but a Reputation Pledge is a similar concept: you've invested resources that create an incentive to keep a good reputation and impose a significant cost for abandoning that reputation.

V. OpenBazaar - Namecoin Integration

A. Allow users to configure their user-friendly URL in the UI

The first step was to give the users the opportunity to choose the Namecoin that he/she wants to represent the name of his/her store. This name should automatically be stored and fetched from the database, together with all the other credentials of the user.

Moreover, since the project was already functioning, in order to make any change in the database, we should write a script in order to update the database to the current version and vice versa. This way, whoever used the previous version of OpenBazaar could just execute the script and update his/her database, without having to recreate it from scratch.

```
<div class="form-group">
    <label for="inputNamecoin_id"
        class="col-sm-3 control-label">Namecoin id
    </label>
    <div class="col-sm-9">
        <input data-ng-model="settings.namecoin_id"
               data-ng-maxlength="39"
               data-ng-pattern="/^[a-z0-9\-]+$/"
               class="form-control"
               id="inputNamecoin_id"
               name="inputNamecoin_id"
               placeholder="Namecoin id"
        >
    </div>
</div>
```

```
def upgrade(db_path):
    with dbapi2.connect(db_path) as con:
        cur = con.cursor()

        # Use PRAGMA key to encrypt / decrypt database.
        cur.execute("PRAGMA key = 'passphrase';")

        try:
```

```
            cur.execute("ALTER TABLE settings "
                        "ADD COLUMN namecoin_id TEXT")
            print 'Upgraded'
            con.commit()
        except dbapi2.Error as e:
            print 'Exception: %s' % e
```

B. Relay Namecoin id to other nodes

OpenBazaar, when functioning, exchanges certain messages between its nodes, in order to exchange information between these nodes and transfer requests to them. Thus, a node which claims to be the holder of a certain Namecoin id, must publish this id in the messages that it sends. Before this, of course, we must verify that the id that was received from javascript is indeed a valid namecoin id of OpenBazaar and then, store the Python object, in order to be directly available from the code which is in charge of sending the messages and saving the database.

Finally, having completed the above, we must alter the code which creates the messages sent throughout the nodes, in order to add the claimed Namecoin id.

```
# Validate that the namecoin id received is well formed
if not re.match(r'^[a-z0-9\-]{1,39}$', msg['namecoin_id']):
    msg['namecoin_id'] = ''

# Update nickname and namecoin id
self.transport.nickname = msg['nickname']
self.transport.namecoin_id = msg['namecoin_id']

if 'burnAmount' in msg:
    del msg['burnAmount']
if 'burnAddr' in msg:
    del msg['burnAddr']

# Update local settings
self.db.updateEntries(
    "settings",
    msg,
    {'market_id': self.transport.market_id}
)
```

```
self.send_raw(
    json.dumps({
        'type': 'hello',
        'pubkey': self.transport.pubkey,
        'uri': self.transport.uri,
        'senderGUID': self.transport.guid,
        'senderNick': self.transport.nickname,
        'senderNamecoin': self.transport.namecoin_id,
        'v': constants.VERSION
    }),
    cb
)
```

```
# Include sender information and version
data['guid'] = self.guid
data['senderGUID'] = self.transport.guid
data['uri'] = self.transport.uri
data['pubkey'] = self.transport.pubkey
data['senderNick'] = self.transport.nickname
data['senderNamecoin'] = self.transport.namecoin_id
data['v'] = constants.VERSION
```

C. Validate claimed Namecoin id from data received

When a node is inserted on the network, the Namecoin id that he claims to have, is connected with the rest of the information of the node. All the information is digitally signed together with the ECkey, which is connected with the GUID of this node.

More specifically, adding the namecoin id of the sender in the initial message, we obtain the encryption and digital signature for the sign-then-encrypt scheme that the OpenBazaar uses [3]. The receiver can retrieve from the Namecoin's blockchain the GUID of the node that wrote and signed the message, and certify that the person who wrote and signed the message is the same.

So, when a node receives a message from another node which claims to be the holder of a certain Namecoin id, he must search this name in the blockchain and verify that the credentials stored there, are these that the node claims.

Searching the blockchain of Namecoin is done via a python open-source project; pydnschain. Data are stored in the blockchain in JSON form, while the domain id/ is used for the names. Pydnschain transforms the JSON string received from the blockchain in a Python dictionary, which it also returns. When this is received successfully, the checking occurs and it is successful, if the registration in the Namecoin has a field called 'openbazaar' with a value equal with the GUID of the node with whom we communicate.

Finally, OpenBazaar uses an automated tool for the administration of the libraries that it need in order to be run, which is known as pypi. This tool is used during configuraion, in order to download and install all the required libraries for a project and then, verifies that they will be available during the execution, no matter what else is installed on the computer at that time. Therefore, we added the tool DNSChain in the requirements of pypi.

```
def _on_message(self, msg):

    # here goes the application callbacks
    # we get a "clean" msg which is a dict holding whatever

    pubkey = msg.get('pubkey')
    uri = msg.get('uri')
    guid = msg.get('senderGUID')
    nickname = msg.get('senderNick', '')[:120]
    msg_type = msg.get('type')
    namecoin = msg.get('senderNamecoin')

    # Checking for malformed URIs
    if not network_util.is_valid_uri(uri):
        self.log.error('Malformed URI: %s', uri)
        return

    # Validate the claimed namecoin in DNSChain
    if not trust.is_valid_namecoin(namecoin, guid):
        msg['senderNamecoin'] = ''

    self.log.info(
        'Received message type "%s" from "%s" %s %s',
        msg_type, nickname, uri, guid
    )
    self.log.datadump('Raw message: %s',
            json.dumps(msg, ensure_ascii=False))
    self.dht.add_peer(uri, pubkey, guid, nickname)
    self.trigger_callbacks(msg['type'], msg)
```

```
def is_valid_namecoin(namecoin, guid):
    if not namecoin or not guid:
        return False

    server = DNSChainServer.Server(
        constants.DNSCHAIN_SERVER_IP, ""
    )
    _log.info("Looking up namecoin id: %s", namecoin)
    try:
        data = server.lookup("id/" + namecoin)
    except (DNSChainServer.DataNotFound,
            DNSChainServer.MalformedJSON):
        _log.info(
            'Claimed remote namecoin id not found: %s',
            namecoin
        )
        return False
```

```
    return data.get('openbazaar') == guid
```

D. Display remote namecoin id name

```
<tr data-ng-show="page.senderNamecoin">
    <td class="col-xs-3">Store URL</td>
    <td style="word-wrap:break-word">
        {{page.senderNamecoin}}
    </td>
</tr>
```

Since the name of the node with which we communicate has been validated through the procedure of the last issue, this name should be visible in the details of that store. In this case, Namecoin plays the role of the user friendly name through which a user identifies a store. Thus, this information is included in the html code being produced and displayed on the browser as the "Store URL".

E. Test a basic complete request/response

```
class TestDNSChainLookup(unittest.TestCase):

    def setUp(self):
        self.dnschain_server = Server(
            "192.184.93.146",
            "NOTYETIMPLEMENTED"
        )
        self.test_cases = {
            "id/dionyziz": "dionyziz@gmail.com",
            "id/greg": ["contact@taoeffect.com",
                        "hi@okturtles.com"]
        }
        self.responses = {}
        for name in self.test_cases:
            self.responses[name] = \
                self.dnschain_server.lookup(name)

    def test_valid_JSON(self):
        for response in self.responses.itervalues():
            try:
                json.dumps(response)
            except:
                self.fail("Response was not valid JSON")

    def test_contains_email(self):
        for response in self.responses.itervalues():
            self.assertIn("email", response)

    def test_correct_email(self):
        for name in self.test_cases:
            if "email" not in self.responses[name]:
                self.skipTest(
                    'Response does not contain key "email"'
                )
            self.assertEqual(self.test_cases[name],
                             self.responses[name]["email"])
```

As seen previously, in order to access the DNSChain, the *pydnschain* tool was used, which is itself an open source project currently under development. As part of this work, in order to reassure the correctness of OpenBazaar's code, two pending issues were resolved in that project which had to do with testing the correctness of the queries it performs.

The first issue was the creation of a test that performs queries on known records within Namecoin's block chain, and asserts that the values returned were correct. For that purpose we used Python's unit testing framework called *unittest*. This framework allows for the isolation and testing of pieces of code within a project in an automated manner.

We wrote three tests, which perform queries on names of our choice that we know exist in the block chain. The first of these tests asserts that the data was indeed received from the block chain, and was in a valid JSON format. The second test asserts that the data include a field called "email", while the third test that the email received is indeed the email we expect. These test are now executed automatically along with every other test specified and their results are summarized and presented simply by executing the "nosetests" command.

F. Make Travis run the tests

```
language: python

python:
  - "2.7"

script:
  - nosetests
```

For the execution of the tests, Github supports the Travis CI tool, which automatically executes all the tests in a project each time a change is committed to its code. To enable this feature, all we had to do (after configuring Github to call Travis through its web interface), was to write a configuration file that tells Travis what exactly to execute. However, since we were already using the testing framework explained in the previous issue, the only thing needed was to tell Travis to run the "nosetests" command. This way, each time someone submits a Pull Request in pydnschain, all test are automatically run and their results are summarized along with the pull request.

VI. Future Work

OpenBazaar is currently under Beta construction and so, there are a number of functions that are not yet implemented. Concerning the integration of the Namecoin and OpenBazaar, some of the most obvious future extensions is searching stores via their claimed Namecoin, and giving the DNSChain user the opportunity to choose which server he/she wants to make the searches in the blockchain of Namecoin.

References

[1] Satoshi Nakamoto: *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008.

[2] Dionysis Zindros: *A pseudonymous trust system for a decentralized anonymous marketplace*, 2014
https://gist.github.com/dionyziz/e3b296861175e0ebea4b

[3] Don Davis: *Defective Sign & Encrypt in S/MIME, PKCS#7, MOSS, PEM, PGP, and XML*, 2001

[4] Greg Slepak: *DNSChain + okTurtles*, April 26, 2014

[5] OpenBazaar: *A decentralized marketplace*
https://openbazaar.org/
https://blog.openbazaar.org/
https://github.com/OpenBazaar/OpenBazaar

[6] pydnschain: *Python library for looking up blockchain data via DNSChain*
https://github.com/okTurtles/pydnschain

[7] Bitcoin: *An innovative payment network and a new kind of money*
https://bitcoin.org/en/
Wikipedia: http://en.wikipedia.org/wiki/Bitcoin

[8] Namecoin: *A decentralized open source information registration and transfer system based on the Bitcoin cryptocurrency*
http://namecoin.info/
Wikipedia: http://en.wikipedia.org/wiki/Namecoin