International Journal of Software Engineering and Knowledge Engineering

A PETRI NETS MODEL FOR BLOCKCHAIN ANALYSIS

ANDREA PINNA, ROBERTO TONELLI, MATTEO ORRÚ
Department of Electrical and Electronic Engineering (DIEE),
University of Cagliari, Piazza D'Armi, 09100 Cagliari, Italy
{a.pinna,roberto.tonelli,matteo.orru}@diee.unica.it

MICHELE MARCHESI
Department of Mathematics and Computer Science,
University of Cagliari, Via Ospedale 72, 09124 Cagliari, Italy
michele@diee.unica.it
http://www.agilegroup.eu

A Blockchain is a global shared infrastructure where cryptocurrency transactions among addresses are recorded, validated and made publicly available in a peer-to-peer network. To date the best known and important cryptocurrency is the bitcoin. In this paper we focus on this cryptocurrency and in particular on the modeling of the Bitcoin Blockchain by using the Petri Nets formalism.

The proposed model allows to quickly collect information about identities owning Bitcoin addresses and to recover measures and statistics on the Bitcoin network. Using PRE and POST matrices we reconstructed an Entities network associated to Block Chain transactions gathering together Bitcoin addresses into the single entity holding permits to manage Bitcoins held by those addresses. The model allows also to identify a set of behaviours typical of Bitcoin owners, like that of using an address only once, and to reconstruct chains for this behaviour together with the rate of ring.

To our knowledge, this is the first time that the Petri Nets formalism is used to model Blockchain information and properties. Our model is highly flexible and can easily be adapted to include dierent features of the Bitcoin crypto-currency system.

Keywords: Petri Nets, Bitcoin, Blockchain, Cryptocurrency

1. Introduction

The *Bitcoin electronic cash system* was conceived in the 2008 by the scientist Satoshi Nakamoto [1] with the aim of producing digital coins whose control is distributed across the Internet, rather than owned by a central issuing authority, such as a government or a bank. It became fully operational on January 2009, when the first mining operation was completed, and since then it has constantly seen an increase in the number of users and miners.

At the beginning, the interest in the bitcoin digital currency was purely academic, and the exchanges in bitcoins were limited to a restricted elite of people more interested in the cryptography properties than in the real bitcoin value. Nowadays bitcoins are exchanged to buy and sell real goods and services as happens with traditional currencies.

The main distinctive feature introduced by the Bitcoin system is the Blockchain, that is a shared infrastructure where all bitcoins transfer are recorded. Value transfer is called transaction and is an operation between users. To send and receive bitcoins, a user needs an alphanumeric code, called *address*. Address represents the users' account and to each address a private key is associated. No personal information is usually recorded in a Blockchain and for this reason Bitcoin protocol offers pseudo-anonymity.

To date blockchain is the technology underlying Bitcoin, but is also the technology underlying other cryptocurrencies, such as Litecoin, Dogecoin, Mastercoin, Namecoin and Ethereum. By analysing this technology we can obtain many statistical properties of its associated cryptocurrency network, as well as the typical behavior of users, for example how users move bitcoins between their various accounts in order to preserve and reinforce their privacy.

In this paper we introduced a novel approach, based on a Petri Net model to analyse the Blockchain. Our purpose was to define a single useful model, a unique data structure, in which all main information about transactions and addresses are represented.

We modelled the Bitcoin system using Petri Nets, assuming that each address corresponds to a place and each Bitcoin transaction corresponds to a transition in a Petri Net (also known as Place/Transition Net or P/T Net). The proposed model, called "Addresses Petri net", allows to quickly collect information on the identities owning Bitcoin addresses and to recover measures and statistics on the Bitcoin network. We reconstructed an Entities network associated to Block Chain transactions gathering together Bitcoin addresses into the single entity holding permits to manage Bitcoins held by those addresses. In other words, the performed analysis allows us to construct first the "Addresses Petri Net", and then the "Entities Petri Net".

The paper is organized as follows. In Section 2 an overview of related works is reported, in Section 3 we illustrate the Bitcoin payment system, in Section 4 we describe our model. Specifically, Section 4.2 illustrates the Addresses Petri Net associated to Bitcoin addresses. Section 4.3 describes Entities Petri Net and the proposed algorithm in order to infer from the Addresses Petri Net, the Entites Petri Net. In Section 5 we apply the model to the Bitcoin system and present our results. Finally, Section 6 presents the results discussion and Section 7 contains our conclusions.

2. Related works

In these last years, the unique features of Blockchain have attracted more and more researchers and several are the works that examined this shared data collection. Several papers focused on heuristics and algorithms in order to analyse and cluster Bitcoin addresses identifying network of users.

Ron and Shamir [2] analysed and measured the Blockchain up to the block number 180,000, from January 03th, 2009 to May 13th, 2012, by using a model called *transaction graph*. They analysed the distribution of the number of transaction per address and introduced the concept of *entity* as a group of addresses of the same owner. They ran a variant of a Union-Find graph algorithm in order to find sets of addresses belonging to the same user. First, they constructed the transaction graph the address graph, and then constructed the contracted transaction graph the entity graph. Thanks to this entity graph, the authors determined various statistical properties of each entity, such as the distribution of the accumulated incoming bitcoins, the balance of bitcoins on May 13th 2012, and of the number of transactions per entity and per address. In addition they invested the most active entities in the system. In this paper, we analyse the same blocks by using a novel approach based on Petri Net formalism. This formalism allows us to construct first the "Addresses Petri Net"and then, taking advantage of the concept of *entity* introduced in [2], the "Entities Petri Net". As in work [2], thanks to the Entities Petri Net, we investigated behaviours typical of bitcoin owners, and associated an identity to addresses and to entities. However, we did not perform analysis about the distribution of bitcoins among entities but performed interesting analysis about a common practice used by users in order to preserve and reinforce their anonymity. In [2], the user common practice to move bitcoins between their various accounts (addresses) is considered a good practice to preserve and reinforce user's anonymity. In Bitcoin system each user can create a new address taking a new identity and can transfer all value from an old address to the new address. This last operation can be done many times as an user wants. Further, several users do this systematically using an address only one time. In this work we call the addresses used only one time, "disposable addresses", and proposed an algorithm to recognize them. However, there are many other strategies adopted in order to preserve and reinforce the anonymity of the Bitcoin users. Some of these strategies improve the privacy and anonymity including mixing protocols, eg. CoinShuffle, CoinJoin and CoinParty, [3, 4] and others are based on the TOR network. Biryukov et al. in [5] found countermeasures to block users who access in the Bitcoin network using Tor or other similar protocols.

Reid and Harrigan [15] studied how an attacker could make a map of users' coins movement tracing their addresses and joining information from others sources. They also focused in the topology of addresses network and transaction network, showing the complexity of these networks.

Androulaki *et al* [7] analysed how users try to reinforce theirs anonymity in the Bitcoin system. In particular, they studied the technique of changing address and how this makes more complex the network.

Meiklejom et al. [8] proposed an heuristic to recognize the changing addresses method, and to keep track of potential criminal users, thanks to information extracted from the Blockchain and from other source, such as forum. They tried to give a name to each address.

Kondor et al. [10] focuses on retrieving the Blockchain transaction network, studying its property along time.

Recently, Lishke and Fabian [9] proposed an explorative analysis of the Blockchain and of Bitcoin users. They studied the economy and main features of the Bitcoin cash system, but did not focus neither on the concept of ”entity”, nor on disposal addresses, as we do in this work. Their analysis revealed the major bitcoin businesses and markets, giving insights on the degree distribution (probability density function and complementary cumulative distribution function) of bitcoin transactions for several aggregations of time, businesses categories and country. These distributions revealed the existence of a scale-free network, and hence that Bitcoin network follows a power law distribution although not over the entire period. Also in our work, we found that the distributions of several investigated quantities follow a power-law very closely (see section 5 for details).

Given the surge of interest in bitcoin, blockchain analysis is not the only topic faced. Cocco et al. in [11] presented an agent-based articial cryptocurrency market in which heterogeneous agents buy or sell cryptocurrencies, in particular Bitcoins. The model proposed is able to reproduce some of the real statistical properties of the price returns observed in the Bitcoin real market. In [12] the same authors proposed an agent-based articial cryptocurrency market in order to model the economy of the mining process, starting from GPUs generation, reproducing some stylized facts found in real-time price series and some core aspects of the mining business.

Other works focus on security and privacy issues, cryptographic problems, social aspects of the Bitcoin users behavior and on and economic aspects and the implication of the cryptocurrency phenomenon, see for instance works [13, 14, 15, 16, 17, 18].

3. The Bitcoin Cash System: an overview.

Blockchain is a shared and global database where all information about movement of bitcoins are stored. It has the function of a public ledger composed of an ordered sequence of blocks. Each block contains digital data about a variable number of validated transactions. The blocks size can not be greater than 1MB. The transactions are the implementation of bitcoin value transfer. Each transaction is composed by an input section and an output section. The sections are filled by a list of addresses and by the values associated to each address.

Summarizing, information associated to each transaction in the Blockchain are characterized by:

- a list of inputs, each containing a previous transaction and an associated value;
- a non empty list of outputs (possibly coinciding with some inputs);
- the associated amounts to each output;

An address is an alphanumeric string of 32 elements which can begin only with ”1” or ”4”. For istance, a valid bitcoin address is:

$$1JQfVfzfxtfUb9kexSt7mHhcHxX6fyBJ5A.$$

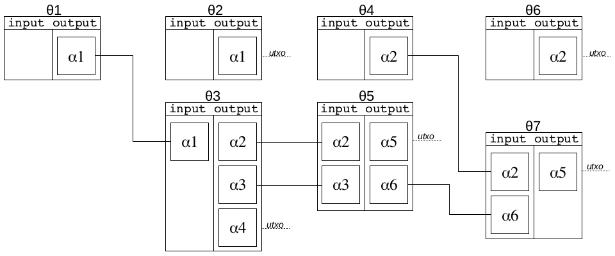


Fig. 1. Simplified transaction schematic.

Users can own one or more addresses and their creation is costless. Users' anonymity is preserved by the fact that in the Blockchain only addresses are provided, and neither usernames nor other identity information are required to create an address.

Bitcoin client, the software that allows users to interact with the Bitcoin network, manages the addresses in *digital wallets*. The wallets store the pair of public and private keys, which are used respectively to receive and to send payments.

In Fig. 1 a simplified schematic of transactions (called θ_i) and addresses (called α_j) is shown. Specifically, seven transaction and six addresses are involved in the chains. The balance of bitcoins owned by an user is associated to his own address. Balance of an address is equal to the total bitcoin value of the unspent transactions (or UTXO) that the address has received and is not spent yet. For example, the addresses $\alpha_1, \alpha_2, \alpha_4$ and α_5 have one or more UTXO, so their balance is not null.

Each transfer of bitcoins among users imply a change of the balances associated to the respective addresses, similar to what happens with a traditional bank accounts. The transaction requests wait in the peer-to-peer network until they are not validated in order both to prevent scams and to avoid double spending. A transaction between addresses can be accepted only if it satisfies the following constraints:

- the inputs have to correspond to the outputs of previous unspent transactions (UTXO) with same address and values;
- the output total value has to be less or equal to the total value of the inputs. The eventual difference being the transaction *fee*.

The validation procedure is called *mining*, and is carried out by *miners*. It consists in gathering a set of transaction requests and in computing an hash key of a new block. In addition to transaction data, each new block contains several information such as the hash code of the previous block in the Blockchain, its height (its associated progressive number), and the IP address of the miner.

Mining the blocks is a competitive task in the peer-to-peer network. All miners can try to calculate the hash code to validate a new block. The difficulty of this computational problem is automatically adjusted by the network, from time to time, in order to maintain constant, on a statistical base, the release rate of the new blocks. This rate is equal about to one block each ten minutes. The first miner validating the transactions, namely creating a valid block, receives a reward in bitcoins (presently 12.5). A special kind of nodes in the peer-to-peer network, called *full nodes*, check the new blocks, specifically their validity, verifying that they respect the Bitcoin's core consensus rules[a].

It is possible that two miners compute two different valid solutions for the same block at the same time. In this case the peer-to-peer network accepts the block belonging to the longest chain, that is belonging to the chain with the most combined difficulty[b]. The other block will be abandoned over time and the miner who mined it does not receive any prize. In Fig. 1, we can identify the mining transactions. They are the transactions $\theta_1, \theta_2, \theta_4$ and θ_6, that is the transactions having their input section empty. Nowadays miners are gathered in pool to join computational power and to make proportional and constant the incoming of pool members.

4. The model: the Blockchain Petri Net

The proposed model is based on the Petri Net formalism. Thanks to this formalism we can obtain a lightweight but useful representation of the Blockchain called Addresses Petri Net. We chose Petri Nets to take advantage of the common features between Blockchain structure and Petri Nets.

Petri Net is an oriented graph, made of two types of nodes, place and transitions, and each node can be connected only with a node of the other type. Bitcoin Blockchain can be represented as an oriented graph, made of two types of nodes, addresses and transactions, modeled by using Petri Net formalism as place and transitions, respectively.

4.1. *Petri Nets: A brief introduction*

A Petri Net [19] is a formalism to describe systems based on a bipartite graph with two kind of nodes called *places* and *transitions*. For this reason, Petri nets are also called *Place Transition nets* (*P/T nets*). Connections between nodes are made by oriented arcs. Each node can be only connected to nodes of the other type and there are two type of arcs. An arc ingoing in a transition is called *Pre-arc* and an arc outgoing from a transition is called *Post-arc*.

Petri Nets are also well described by an algebraic formalism. The formalism provides sets to define the nodes and matrices to describe the arcs. A Petri Net N is a quadruple defined as described below.

[a] https://en.bitcoin.it/wiki/Full-node
[b] https://en.bitcoin.it/wiki/Block

Definition 1.

$$N = (P, T, Pre, Post) \tag{1}$$

where

- $P = \{p_1, p_2, ..., p_m\}$ is the set of m places,
- $T = \{t_1, t_2, ...t_n\}$ is the set of n transitions,
- $Pre : P \times T \to \mathbb{N}$ is the *Pre-incidence* function
- $Post : P \times T \to \mathbb{N}$ is the *Post-incidence* function.

Pre and *Post* incidence functions are usually defined by mean of matrices with dimension equal to $m \times n$. Each element of these matrices contains the number of arcs which connect places with transitions. The *Pre* matrix contains the numbers of ingoing (to transitions) arcs for each couple place-transition. Vice versa, each element of *Post* matrix is the number of ingoing arcs for each couple place-transition.

Petri nets are also a powerful formalism to describe discrete event systems. To model the state of a system, a marking M (that is a vector which defines the distribution of tokens in places) is needed. Transitions have the aim to modify the marking of the system. These absorb tokens from places connected with Pre-arcs and produce token intended for the places connected with Post-arcs, an operation called *firing* of a transition. Petri net and the associated initial marking compose the Network system defined as $\langle N, \mathbf{M}_0 \rangle$, where $\mathbf{M}_0$ is the initial marking. In this work we do not describe a specific state of the Blockchain so we do not need to define a marking.

4.2. *Addresses Petri Net*

In the Blockchain, transactions move bitcoins from a set of address to another. A transaction is an operation (i.e. a value transfer) between two set of addresses. One set is the input set, the other one is the output set. Addresses in the input set can be connected by a transaction to addresses in the output set. Summarizing, two typology of elements in the Blockchain are present and these elements can be directly connected only with an element of the other typology. This peculiarity, joined with the simplicity and the handiness of the matrix formalism, allows us to convert and parse the Blockchain as a Petri net.

We denote $\mathcal{A} = \{\alpha_1, \alpha_2, ..., \alpha_m\}$ the finite set of m addresses α registered either as inputs or outputs in the Blockchain, and with $\Theta = \{\theta_1, \theta_2, ..., \theta_n\}$ the set of n transactions θ validated by the Blockchain.

Let $N_\alpha = (P_\alpha, T, \mathbf{PreA}, \mathbf{PostA})$ be the network of addresses, where:

- $P_\alpha = \{p\alpha_1, p\alpha_2, ..., p\alpha_m\}$ is the set of m places with each place $p\alpha$ associated to one and only one address $\alpha \in \mathcal{A}$;
- $T = \{t_1, t_2, ...t_n\}$ is the set of n transitions where each transition t is associated to one and only one transaction $\theta \in \Theta$;
- **PreA**: is the *pre-incidence* matrix;

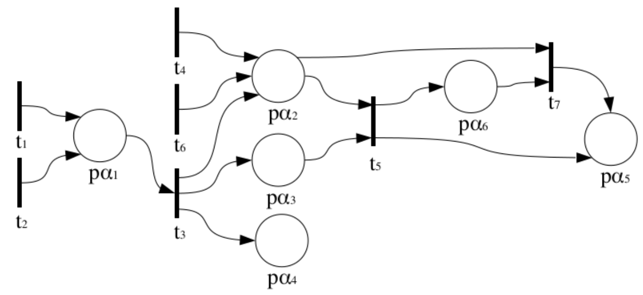


Fig. 2. Addresses Petri Net equivalent to the simplified transaction chains in Fig. 1

- **PostA**: is the *post-incidence* matrix.

The sets P_α and T have been recovered by browsing all the addresses and transactions validated in the Blockchain, which are publicly available, and inserting a new place every time a new address is found, and a new transaction every time a new transaction is encountered.

In order to build the matrices **PreA** and **PostA** let us consider one transaction θ in the Blockchain and the associated transition t. In the Blockchain, a transaction θ consists in a set of input and output addresses with the associated amounts in bitcoin. We denote by $In(\theta) \subseteq \mathcal{A}$ the set of input addresses, and by $Out(\theta) \subseteq \mathcal{A}$ the outputs set. For each address $\alpha \in In(\theta)$ we consider its associated place $p\alpha$ and we add a *pre-arc* leaving from $p\alpha$ and arriving to the transition t associated to θ. At the same time, for each address $\alpha \in Out(\theta)$ we add a *post-arc* leaving from transition t associated to θ and arriving to the place $p\alpha$ associated to α. For each couple $(p\alpha, t)$ to which a *pre-arc* has been added we set $\mathbf{PreA}(p\alpha, t) = 1$, while for each couple $(p\alpha, t)$ to which a *post-arc* has been added we set $\mathbf{PostA}(p\alpha, t) = 1$.

This model does not not keep all information available in the Blockchain (e.g. transactions amounts) and so it cannot represent completely its behavior and properties. However in contrast with the methodologies used in many other works in which different model were applied in order to analyse the Blockchain overloading the analysis, our approach allows to include into one single model and one single data structure different features and properties of the Blockchain.

Consider for instance the simplified transaction chains in Fig. 1. There are seven transaction and six places. The equivalent Address Petri Net is composed by six places and seven transitions. The graphic representation is shown in Fig. 2. This Net is defined by a set of places $P_\alpha = \{p\alpha_1, p\alpha_2, ..., p\alpha_6\}$, a set of transactions $T = \{t_1, t_2, ...t_7\}$ and by the *pre-* and *post-incidence* matrices **PreA** and **PostA**, shown in Fig. 3 and 4.

These two matrices allow us to perform several analysis about the network. We

$$\mathbf{PreA}= \begin{array}{cc} \left[\begin{array}{ccccccc} 0&0&1&0&0&0&0\\ 0&0&0&0&1&0&1\\ 0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1 \end{array}\right] & \begin{array}{c} p\alpha_1\\ p\alpha_2\\ p\alpha_3\\ p\alpha_4\\ p\alpha_5\\ p\alpha_6 \end{array}\\ \begin{array}{ccccccc} t_1&t_2&t_3&t_4&t_5&t_6&t_7 \end{array} & \end{array}$$

Fig. 3. Pre-incidence matrix of the Petri net for the example in Fig. 1.

$$\mathbf{PostA}= \begin{array}{cc} \left[\begin{array}{ccccccc} 1&1&0&0&0&0&0\\ 0&0&1&1&0&1&0\\ 0&0&1&0&0&0&0\\ 0&0&1&0&0&0&0\\ 0&0&0&0&1&0&1\\ 0&0&0&0&1&0&0 \end{array}\right] & \begin{array}{c} p\alpha_1\\ p\alpha_2\\ p\alpha_3\\ p\alpha_4\\ p\alpha_5\\ p\alpha_6 \end{array}\\ \begin{array}{ccccccc} t_1&t_2&t_3&t_4&t_5&t_6&t_7 \end{array} & \end{array}$$

Fig. 4. Post-incidence matrix of the Petri Net for the example in Fig. 1.

can compute the difference between *post* and *pre-incidence* matrices and take into account one its row. The number of not null elements in it is equal to the number UTXO which is present in the address related to the place of that row. This number must be greater or equal to zero and if is equal to zero, then the balance of associated address is null.

In addition we can compute the number of times that an address appears in input of a transaction. In fact, all the not-zero elements of the row i of the matrix **PreA** provides the number of times the address α corresponding to the place $p\alpha = i$ has been an input of a transaction. Furthermore, considering that two transitions having the same set of pre and post places refer to the same transaction, we can ascribed them to a same single transition.

In the next section we explain how recognizing the entities, in term of place sets, relying on the analysis of the **PreA** matrix.

4.3. *Entities Petri Net and Proposed Algorithm*

It is quite common for Bitcoin users to hold more than a single address, in order to manage bitcoin exchanges and anonymity more easily. As in [2] we define an *entity* as the person, the organization, the group of people, the firm that holding the control of the bitcoins of a set of addresses. All addresses appearing into an input section of a single transaction must be owned by the same entity. This is because in order to activate the bitcoin transfers from those addresses, the same entity must hold all the private keys of all corresponding wallets is necessary.

In order to build the Entities Petri Net N_ϵ we associated each entity to a collection of addresses, associating places $p\epsilon \in P_\epsilon$ in N_ϵ to a set of places $p\alpha$ of N_α. We denote by $E = \{\epsilon_1, \epsilon_2, ..., \epsilon_k\}$ the set of *entities* where each *entity* $\epsilon \in E$ is a finite set of addresses such that $\epsilon \subseteq \mathcal{A}$.

The matrix **PreA** has m rows, one for each place, and n columns, one for each transition. Given a transition t we consider the array $\mathbf{PreA}(\cdot, t)$ which is the column of **PreA** with index t. Its non zero elements correspond to places $p\alpha$ with $\mathbf{PreA}(p\alpha, t) = 1$, namely places with outgoing arcs *pre-arc* towards transition t. These places $p\alpha$ correspond to input addresses $\alpha \in In(\theta)$, for the transaction θ corresponding to transition t. As a consequence all these places belong to one single entity $\epsilon \in E$.

It is also possible that a given address appears in two or more input sections, together other addresses. In this case, the entity must be composed by all the addresses in these input sections.

To build the Entities Petri Net, E, we applied the following algorithm.

Let $T^* = T$ the set of unexplored transitions and $E = \emptyset$ the set of entities.

```
• while $T^* \neq \emptyset$
    (1) take a $t : t \in T^*$ and remove this form $T^*$
    (2) let $e = \emptyset$
    (3) for all $i : \mathbf{PreA}(p_i, t) \neq 0$ do $e = e \cup \{p_i\}$
    (4) let $e^* = e$ the set of unexplored place
    (5) while $e^* \neq \emptyset$
        (a) take a place $p \in e^*$
        (b) let $T' = \emptyset$
        (c) for all $j : \mathbf{PreA}(p, t_j) \neq 0$ do $T' = T' \cup \{t'\}$
        (d) for all $t' \in T'$
            i. let $e_{new} = \emptyset$
            ii. for all $h : \mathbf{PreA}(p_h, t') \neq 0$ do $e_{new} = e_{new} \cup \{p_h\}$
            iii. $e = e \cup e_{new}$ and $e^* = e^* \cup e_{new}$
            iv. $e^* = e^* \setminus p_h$
            v. $T^* = T^* \setminus t'$
        end
    (6) $E = E \cup e$
• end
```

Each $e \in E$ is a set of places of the Addresses Petri Net or, equivalently, is the representation of a set of addresses that compose an entity. We can define N_ϵ, the Entity Petri Net, as $N_\epsilon = (P_\epsilon, T, \mathbf{PreE}, \mathbf{PostE})$, where P_ϵ is the set of places that are one to one associated with elements of the entities set E.

The definition includes the set T of transitions. This is the same which we have in the Addresses Petri Net. We must calculate the matrices **PreE** and **PostE**.

Entity in E	Places
e_1	$\{p\alpha_1\}$
e_2	$\{p\alpha_2, p\alpha_3, p\alpha_6\}$
e_3	$\{p\alpha_4\}$
e_4	$\{p\alpha_5\}$

Table 1. Entity in the Entities Petri Net of the simplified transaction chains in Fig. 1.

$$\mathbf{PreE}= \begin{array}{cc} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \begin{array}{c} p\epsilon_1 \\ p\epsilon_2 \\ p\epsilon_3 \\ p\epsilon_4 \end{array} \\ \begin{array}{ccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 \end{array} & \end{array}$$

Fig. 5. Pre-incidence matrix of the Entities Petri Net for the simplified transaction chains in Fig. 1.

$$\mathbf{PostE}= \begin{array}{cc} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} & \begin{array}{c} p\epsilon_1 \\ p\epsilon_2 \\ p\epsilon_3 \\ p\epsilon_4 \end{array} \\ \begin{array}{ccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 \end{array} & \end{array}$$

Fig. 6. Post-incidence matrix of the Entities Petri Net for the simplified transaction chains in Fig. 1.

In order to calculate these matrices, we took each entity e from E and, starting from **PreA** and **PostA**, we substituted each row corresponding to each places $p\alpha$ contained in e with a new row equal to the sum.

For istance, looking at the Address Petri Net in Fig. 2 and at the **PerA** matrix, we recognize that places $p\alpha_2, p\alpha_3$ and $p\alpha_6$ can be joined to an entity, and that hence their related addresses $\alpha_2, \alpha_3, \alpha_6$ are owned by the same person. In total four entities are recognized as described in Table 1.

To each entity a place $p\epsilon \in P_\epsilon$ is then associated. In the following tables, **PreE** and **PostE** of the example resulting Entities Petri Net are showed in Fig. 5 and 6.

In the next section we discuss the results of our analysis performed after having modeled a wide portion of the Blockchain.

5. Results

Blockchain can be explored mainly in two modes. The first consists in downloading all binary data from the peer-to-peer network, and in identifying transactions,

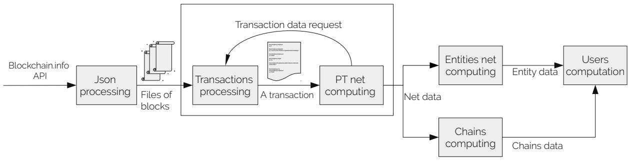


Fig. 7. Diagram of the data processing path for the study of Blockchain

addresses and other information by using protocol instructions. The second mode consists in exploring specific websites where the decoded Blockchain is showed and application interfaces, and other utilities, are provided. We followed the second way and downloaded blocks as formatted Json files from the website *blockchain.info*.

We parsed the first 180,000 blocks in the Blockchain. These blocks corresponds to a period of about three and half years, from January 2009 to March 2012.

The data processing, performed in this work, can be subdivided in steps and is shown in Fig. 7. All implementations are made with R language and RStudio IDE. Of course, our model has some limitations. At first the computational problems that can arise due to the adopted matrix approach. We built our representation of Pre and Post matrices with Long lists of vector, well supported in R language. The analysed portion of Blockchain was processed without specific hardware resources, and consequently processing a higher number of blocks would not have been possible.

In the next two sections we describe our results. The first section contains the results of measuring of the Blockchain just converted in Petri Net structure. Further it describes results of data elaboration that include the chains of transactions and collapsing of the transactions. The second section shows results of the more complex elaboration that is the Entities Petri Net.

5.1. *Results of the Addresses Petri net*

We found 3,730,480 different addresses and 3,142,019 transactions, which in our model correspond to the number of rows and columns of matrices **PreA** or **PostA**. We associated the addresses to the corresponding places in the set P_α in the Petri Net $N\alpha$. From the analysis of the matrices **PreA** and **PostA**, simply counting the non zero elements, we found 4.575.888 *pre-arc* and 7.352.494 $post-arc$ in total.

The number of non zero elements $L(i)$ on the corresponding $i-th$ row of $\mathbf{PreA}(p\alpha_i, \cdot)$ represents the number of transitions that take place from the place $p\alpha_i$ through a $pre-arc$. The number of non zero elements $L(i)$ in the row $\mathbf{PostA}(p\alpha_i, \cdot)$ represents the number of transitions connected to the place $p\alpha_i$ through a $post-arc$. Thus our model easily takes into account the total number of bitcoin transactions in input or output of each address. Figures 8 and 9 report the Complementary Cumulative Distribution Functions (CCDF) for the number of non-zero elements in

Table 2. Summary of first 10 most used addresses

Address	L pre	L post	tags
1VayNert3x1KzbpzMGt2qdqrAThiRovi8	270,204	275,398	deepbit.net
1dice8EMZmqKvrGE4Qc9bUFf9PX3xaYDp	14,606	14,605	SatoshiDICE 48%
1dice97ECuByXAvqXpaYzSaQuPVvrtmz6	13,137	13,124	SatoshiDICE 50%
159FTr7Gjs2Qbj4Q5q29cvmchhqymQA7of	8,016	8,425	- spammer ? -
1CDysWzQ5Z4hMLhsj4AKAEFwrgXRC8DqRN	6,382	9,501	Instawallet
1E29AKE7Lh1xW4ujHotoT4JVDaDdRPJnWu	7,761	8,079	- unknow -
15VjRaDX9zpbA8LVnbrCAFzrVzN7ixHNsC	6,999	7,888	faucet donation
15ArtCgi3wmpQAAfYx4riaFmo4prJA4VsK	6,578	6,622	faucet donation
1dice9wcMu5hLF4g81u8nioL5mmSHTApw	6,318	6,306	SatoshiDICE 73%
1Bw1hpkUrTKRmrwJBGdZTenoFeX63zrq33	5,498	5,498	- unknow -

the matrices **PreA** and **PostA** respectively, and show an uneven distribution of in and out transactions among addresses.

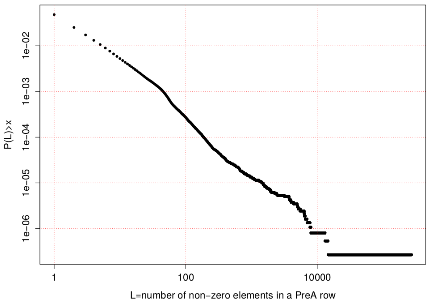


Fig. 8. CCDF of the length L for *PreA*.

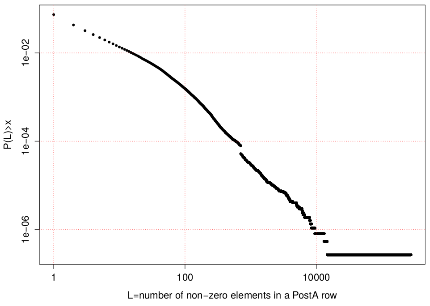


Fig. 9. CCDF of the length L for *PostA*.

In Tab. 5.1 we describe the ten most used addresses, found summing up the number of non zero elements in **PreA** rows to that of non zero elements in **PostA** rows.

Our analysis identifies also 609,295 addresses with only zeros in **PreA** rows, and at least one non null element in **PostA** rows, namely 609,295 addresses never used to spend (till the 180,000 block), but only to accumulate. Part of them are still unused up to today. Table 5.1 reports the first ten ranked by the number of *post-arcs*, namely the number of incoming transactions, and shows their current balances, as checked from blockchain.info: four of these (row 1,7,8 and 9 in the tab) have not been used more and are *dormant*. Their balance can be quite high, since they've been used like sort of bitcoin deposits.

In order to analyse users practices for keep anonymity, we focused on recognizing chains of transaction where are involved *disposable addresses*. As already mentioned,

Table 3. Summary of first 10 most imbalanced addresses

Address	L-post	current balance BTC
15S1TFTosxrgZxkqJR2n1AFJ22ZJE2rTCk	3,853	120.85215349
1PtnGiNvhAKbuUQ6nZ7nF3CDKCKGfeMsCX	1,199	0
129FTwWoi5H5ujasMZ6M6VjJzBJfsXVQGw	1,138	0.78425567
1FN9kKsZA9XttrAwuDDgsXjs6CXUR2fzmt	1,111	0
1DYvtKtZ2Ay9vTjzjb9BiRauMgXdjRDaD	973	14.5601
1STRonGxnFTeJiA7pgyneKknR29AwBM77	949	1.79274504
1Q3nqtUzBp6jw7opi674Pyfgu4MUmVRdrk	861	16.31551365
1Hh3eNNqR8MajEtDfvUF3hoxgf8CuUXVwY	819	257.32881319
14sx4sFdUE9YDpJ9XbD6xAUEKPKvc8QHq2	811	59.56546509
17igtzSD39ZAapsut2DQTTKFyqSp7CToMq	809	0

a disposable address is an address used only two times. One time to receive bitcoins and one time to give away all these bitcoins. Transactions which involve disposable addresses have only a disposable address in the input section and one disposable address in the output section, together other addresses. Usually in the output section only two addresses in total are present. This practice is commonly used and can be performed automatically. Users who adopt it, usually give rise a long chain of transactions without waiting for confirmation.

With our model, *disposable addresses* and their chains can be easily traced analysing **PreA** and **PostA** matrices. To identify the involved transactions we identified the correspondent transitions having only a pre-arc and two post-arc in the Addresses Petri Net. These are transitions that correspond to columns of **PreA** having only one non zero element and to columns of **PostA** having two non zero element but in different rows. Disposable addresses are likewise identified through the correspondents places. Give a place, the correspondent row of the Pre and Post matrices must have one and only one non zero element. The algorithm developed to build the chains of this kind of transitions is described in detail in Appendix A. Applying this algorithm we found 122,155 disposable address chains, involving 1,350,010 different addresses and transactions. Figure 10 reports the CCDF of the chains lengths, showing that these are unevenly distributed, with the longest chain counting 3,658 transactions.

We also counted how many time users repeat the same transaction in terms of the same set of addresses in input section and the same set of addresses in output section. In our model identifying these repetitions is trivial. When two or more transactions involve identical sets of addresses in input and output, the correspondents transitions are connected with the same places both in pre and in post.

So these are transitions having the same value in the correspondent columns $\mathbf{PreA}(\cdot, t_j)$ and $\mathbf{PostA}(\cdot, t_j)$ for different columns j that identify a transition. We found that about 11% of transactions are a repetition of another one. These represent repeated transfer of bitcoins from one group of addresses to another group of addresses where the two groups are always the same, revealing steady fluxes

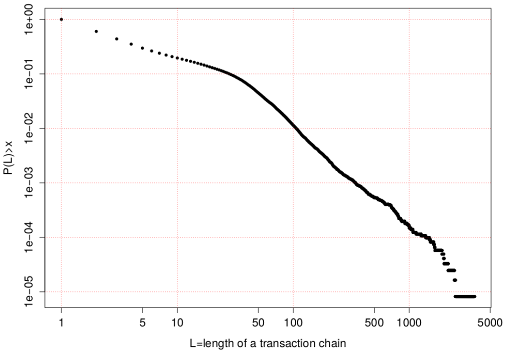


Fig. 10. CCDF of the distribution of the length of the chains.

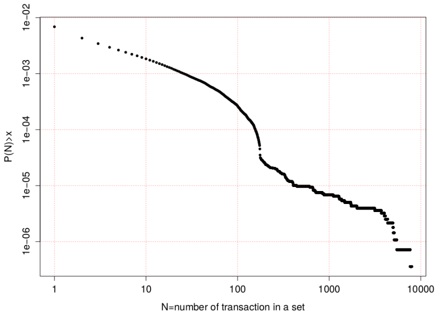


Fig. 11. CCDF of the size L of grouped transaction set for the address net

of bitcoins. Figure 11 reports the CCDF for the sizes of these groups of repeated transactions.

5.2. *Results of the Entities Petri Net*

The reducing algorithm discussed in section 4.3 was applied to the Addresses Petri Net in order to recover the corresponding Entities Petri Net making possible other analysis discussed in the following. Among the owners, we found that 2,461,010 entities hold all the 3,730,480 addresses, and the distribution of addresses among entities is highly non uniform. Figure 12 shows that also such distribution follows a power-law very closely. This means that there are many entities holding a single address but also a few entities controlling very many addresses, and thus able to control a great fraction of the bitcoins flux transactions.

There are only 246,660 entities containing two or more addresses and these con-

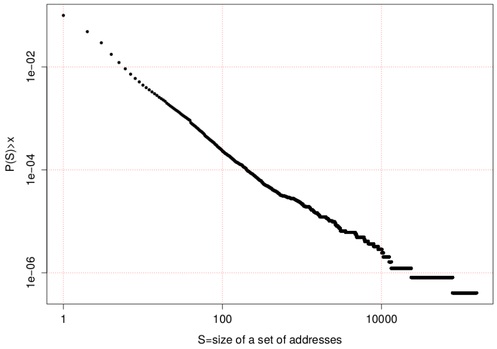


Fig. 12. CCDF of the distribution of addresses across entities.

tains 1,516,130 addresses. The number of non null elements in the rows of matrices **PreE** and **PostE** for the Entities Petri Net is reported in Figure 13 and 14 respectively. This corresponds the number of transactions where the entities are involved. They clearly show a power-law distribution for transactions among the entities.

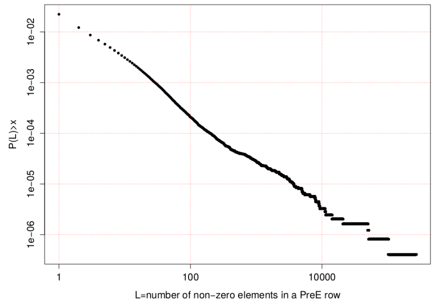


Fig. 13. CCDF of the lenght L for *PreE*.

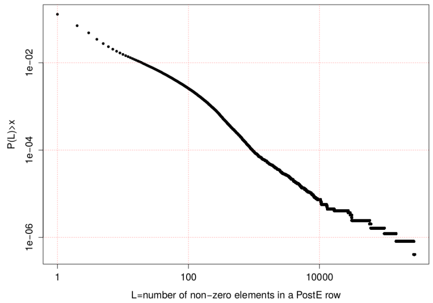


Fig. 14. CCDF of the lenght L for *PostE*.

In Tab. 5 we report the ten most used entities, found summing up the number of non zero elements in **PreE** rows to that of non zero elements in **PostE** rows. Their balances can be computed summing the balances of all the addresses belonging to the corresponding entity and are owned by a single user.

Like for the Addresses Petri Net, we compute groups of repetead trasitions for the Entities Petri Net. We found that about 22.6% of transactions are a repetition of another one occurred among the same entities in input and in output. Once more this information allows to identify steady fluxes of bitcoins at the owners level. Figure 15 reports the CCDF for the sizes of these groups of repeated transactions.

Table 4. Summary of first 10 most active entities

Entity number	L pre	L post	size	tags
95237	270,204	275,398	2	deepbit.net
2	102,186	283,973	156,725	ilovethebtc
37	51,228	147,712	78,251	jmm5699
11	49,959	97,732	10,37	- unknow -
130	20,857	58,350	23,649	Instawallet
66437	14,219	60,868	13,289	Rai, Dread88
42	9,268	31,147	10,561	Quip, iosp and other
37598	8,923	31,004	12,520	generalfault, safetyvest.com
220	11,133	27,487	9,093	zephram
1503	9,044	29,400	10,116	folk.uio.no/vegardno

Table 5.

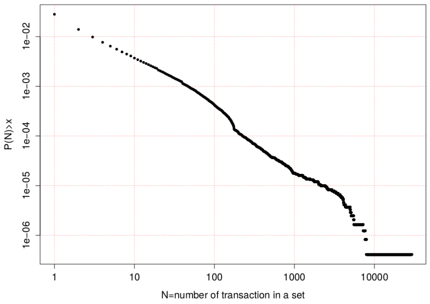


Fig. 15. CCDF of the size L of grouped transaction set for the Entity Petri Net net

6. Discussion

The Petri Net can be used to infer some information about Bitcoin users. We use the Petri Net model to gather together group of addresses (as entities or groups of disposable addresses) trying to associate an identity to each group. We can estimate how many users was been actually involved in the first three years and half of Bitcoin activity.

First of all, computing entities we found that 1,516,130 addresses are controlled by 246,660 owners at most.

Second, with our model we were able to trace transactions chains where are involved disposable addresses. Each chain holds addresses belonging to one owner, but one owner may control more than one chain. So, according to our results, the 1,350,010 addresses involved are owned at most by 122,155 owners. Third, analysing matrices we found that 609,295 addresses are used only as output of bitcoin transactions and are not involved in entities or chains. These three facts, enable us to estimate a threshold for the number of different Owners or users in the Bitcoin system.

We compute that there were 368,815 *engaged* (or expert) owners that adopted disposable addresses practice or used two or more addresses in their operations. We compute that the addresses such owners own are involved in the $72,6\%$ of transactions. We suppose that the 609,295 addresses that appear only in output are used by some *engaged* owners for the purpose of bitcoin depositing. Finally, the 255,045 remaining addresses are owned by occasional users.

Further we tried to associate an identity to addresses showed in Tab. 5.1 and to entities showed in Tab. 5. To find an identity or a *tag* for these addresses we looked for information from several sources like forums or exchange website. We found an address that appears almost 270,204 times as input. Through a direct inspection we found that it belongs to a pool called deepbit (https://deepbit.net/). The entity most used as output is composed by 156,725 addresses and covers 4,2% of the total number of addresses. We found that this owner used *ilovethebtc*, *mikeo*, *FredericBastiat*, *edgeworth*, and others tags. Furthermore this user owns the *bitcoinmoxy.com* website and it is related to the *P2PFoundation*. For example, one of addresses he owns is: "1JkYYmnqAL8USvu1EsdF76jax2fNBbsJi2".

From all the reported CCDFs it is clear that all the distributions are characterized by a strongly uneven amount of transactions across the addresses, either for pre and post transactions. This means that there are many addresses where the bitcoins are hardly exchanged, and few addresses where the rate of bitcoin exchange is particularly high. This analysis can be helpful for identifying addresses which are used by pool of miners. In fact, when miners join together in a pool to share computational facilities for mining operations, they need to define a common address where the mining rewards is accounted to. Then they need to redistribute the amount of gained bitcoins among all the pool users. As a consequence the address will be affected by a number of transitions in the corresponding Petri Net as large as the pool's size.

Finally, since we analysed a limited window of 180,000 blocks, the amount of transitions found in the matrices are also a signature of the average rate of Bitcoin transferred between different entities and such rate can be used to infer information on the organizations which can manage massive Bitcoin transfers.

7. Conclusions

In this paper we introduced a novel approach, based on a Petri Net model to parse the Blockchain. Our purpose was to define a single useful model in which all main information about transactions and addresses are represented. By using this model, we were able to pick out significant and original results.Collecting the first 180 thousand blocks, we were able to associate a place for each address and a transition for each bitcoin transaction. Our Petri net includes *pre* and *post-incidence* matrices where all links between addresses and transactions are modeled.

We are aware about the limitation of computational problem of a matrix approach. The portion of Blockchain which we chosen was processed without specific hardware resources. Anyway, the current size of the Blockchain (over 430,000 blocks and the total number of transaction is over 150 million) could not allow us to handle easily all the blocks information.

However, this formalism has proven powerful methodology for performing many kinds of measurements and analysis. Analysing the number of pre and post arcs, we had proof of the presence of power-law like distributions. We made use of both incidence matrices for determining all transactions chains, identifying a typical *disposable addresses* usage by Bitcoin users. By measuring the chains' lengths, we found again power-law like distributions. We were also able to determine that some transactions involve the same group of address in input and in output. We gathered these transaction in sets and the size of such sets follow again a power-law like distribution. By reading information of *pre-incidence* matrix, we were able to identify the entities and we built the Entities Petri Net, repeating on such Petri net all the measures done for the Addresses Petri Net.

On the basis of all the obtained results, we believe that our model can be used for studying a large set of other issues related to other systems based on Blockchain technology, such as Ethereum. Today, Ethereum attracts increasing attention and will be one of our future research topic.

Appendix A. APPENDIX: Chains of disposable addresses

We modeled the Blockchain as a Petri net, a bipartite oriented graph N, defined as $N = (P_\alpha, T, Pre, Post)$, where P_α is the set of the *places (addresses)*, T is the set of the *transitions (transactions)*, Pre is the *Pre-incidence* matrix and $Post$ is the *Post-incidence* matrix. The element ij in the Pre matrix defines how many times the address i is in the input section of the transaction j, instead, in the Post matrix, it defines how many times the address i is in the output section of the transaction j. After having built of the Petri net N, we focused our attention on the chains of disposable addresses, and hence on the transactions having only one address, α_a, in the input section and only two addresses, α_b and α_c, in the output section. In more detail, the address α_a in the input section is used by a user u_1 to send bitcoins to one of the addresses in the output section, α_b, belonging to a user u_2. The other address, α_c, in the output section is created by the user u_1 to collect the change. We created the set of potentially disposable addresses A_d, starting from the set A of the addresses α and from the set Θ of the transactions θ in the Blockchain.

Let Θ_d be the set of transaction θ_d such that:

$$\Theta_d \subseteq \Theta = \{\theta_d : |IN(\theta_d)| = 1, |OUT(\theta_d)| = 2, IN(\theta_d) \in A_d,$$

$$\exists\ \alpha \in OUT(\theta_d) : \alpha \in A_d, \forall \theta_d \in \Theta_d\}.$$

In order to build a chain, for each θ_d we need to know the previous transactions $\theta_{dp} = PREV(\theta_d)$. Using Pre and $Post$ matrices, it is very easy to look for these previous transactions. We call $\Theta_{ds} \subseteq \Theta_d$ the set of transaction θ_{ds} that could be considered the starting point of a chain because it does not have a previous transaction inside Θ_d . We denote with α_{ds} the address in input to a transaction θ_{ds}. Finally, we call $NEXT(\theta_d)$ the transaction $\theta_{d'}$ which has, in the input section, the disposable address that is contained in the output section of the transaction θ_d. To find the chains c of disposable addresses, we defined and implemented the following algorithm:

```
(1) Let $C = \emptyset$ be a set of empty chains, $c$,
(2) for each $\theta_{ds} \in \Theta_{ds}$:
    (a) take a empty chain, $c$,
    (b) insert $\theta_{ds}$ in $c$
(3) for each $c \in C$
    (a) take the last element inserted in $c$, $\theta_d$,
    (b) while $\exists\ \theta_{d'} = NEXT(\theta_d)$
        i. insert $\theta_{d'}$ in c
           end
```

The algorithm returns a set C of chains c. Each chain c contains the transactions ordered by execution order.

References

[1] N. Satoshi, Bitcoin: A Peer-to-Peer Electronic Cash System, from *http://www.bitcoin.org/bitcoin.pdf*, 2009.

[2] D. Ron and A. Shamir, Quantitative Analysis of the Full Bitcoin Transaction Graph, in *Lecture Notes in Computer Science*, 2013.

[3] T. Ruffing, P. Moreno-Sanchez and A. Kate, CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin. *MMCI, Saarland University*, 2014.

[4] J.H. Ziegeldorf, F. Grossmann, M. Henze, N. Inden and K. Wehrle, CoinParty: Secure Multi-Party Mixing of Bitcoins, in *Proc. 5th ACM Conference on Data and Application Security and Privacy*, 2015.

[5] A. Biryukov, D. Khovratovich and I. Pustogarov, Deanonymisation of clients in Bitcoin P2P network, in *CCS '14 Proc. of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[6] F. Reid and M. Harrigan, An Analysis of Anonymity in the Bitcoin System, in *Security and Privacy in Social Networks, Springer*, 2012.

[7] E. Androulaki, G. Karame, M. Roeschlin, T. Scherer and S. Capkun, Evaluating User Privacy in Bitcoin, in *Lecture Notes in Computer Science, Vol. 7859.*, 2013.

[8] S. Meiklejohn, M. Pomarole, G. Jordan,K. Levchenko, D. McCoy, G. M. Voelkrt and S. Savage, A Fistful of Bitcoins: Characterizing Payments among Men with No Names, in *Communications of the ACM, Vol. 59 NO. 4*, 2016.

[9] M. Lischke and B. Fabian, Analyzing the Bitcoin Network: The First Four Years, in *Future internet*, 2016.

[10] D. Kondor, M. Psfai, I. Csabai and G. Vattay, Do the Rich Get Richer? An Empirical Analysis of the Bitcoin Transaction Network, in *PLoS ONE*, 2014.

[11] L. Cocco, G. Concas and M. Marchesi, Using an Artificial Financial Market for studying a Cryptocurrency Market, in *Journal of Economic Interaction and Coordinations*, 2015.

[12] L. Cocco and M. Marchesi, Modeling and Simulation of the Economics of Mining in the Bitcoin Market, from *arXiv.org http://EconPapers.repec.org/RePEc:arx:papers:1406.6496*, 2016.

[13] B. Marie, O. Kim and S. Ariane, Virtual Currency, Tangible Return: Portfolio Diversification with Bitcoins, *ULB – Universite Libre de Bruxelles*, 2013.

[14] H. Ivan, B. Masooda, J. Gahyun and B. Jeremiah, Are Bitcoin Users Less Sociable? An Analysis of Users' Language and Social Connections on Twitter, in *proc. HCI International 2014 - Posters' Extended Abstracts - International Conference*, 2014.

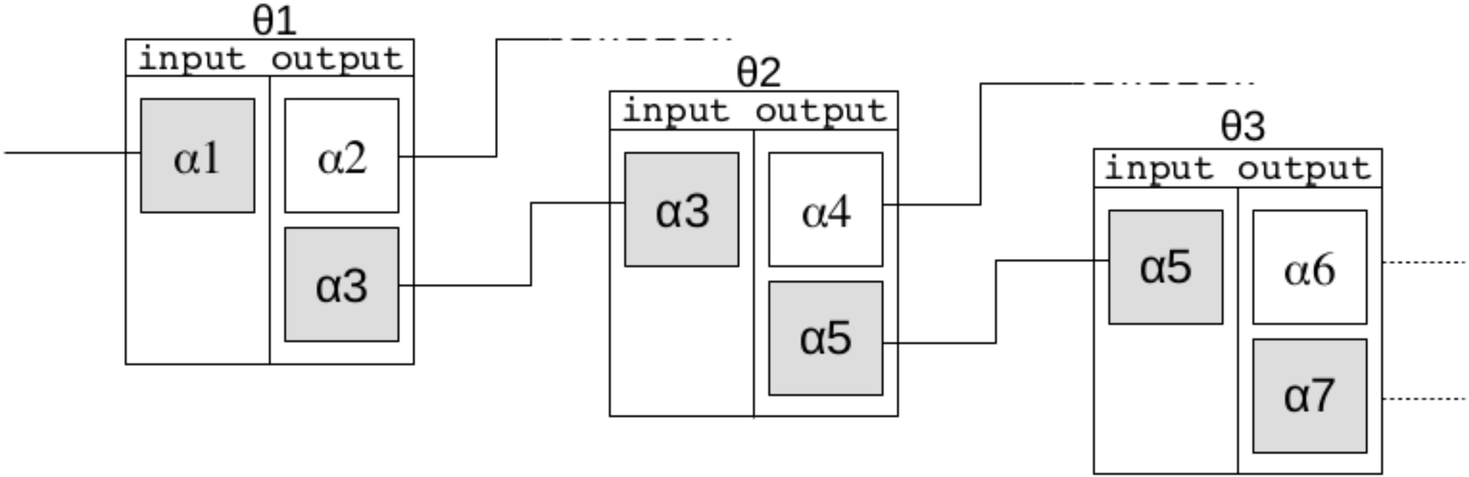
[15] F. Reid and M. Harrigan, An Analysis of Anonymity in the Bitcoin System, in *CoRR, http://dblp.uni-trier.de/db/journals/corr/corr1107.html*, 2011.

[16] A. Saxena, J. Misra and A. Dhar, Increasing anonymity in bitcoin, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*, 2014.

[17] B. Weber, Bitcoin and the Legitimacy Crisis of Money, in *Cambridge Journal of Economics*, 2013.

[18] M. Wilson and A. Yelowitz, Characteristics of Bitcoin Users: An Analysis of Google Search Data, *University Library of Munich, Germany*, 2014.

[19] T. Murata, Petri nets: Properties, analysis and applications, in *MProc. of the IEEE, 77(4):541580*, 1989.


θ1
input output
α1
α2
α3
θ2
input output
α3
α4
α5
θ3
input output
α5
α6
α7