International Journal of Computer Applications (0975 - 8887)
Volume 181 - No.4, July 2018

The Markov Chain Resulting from the States of the Bitcoin's System under the Influence of Selfish-mine Attack, and its Stationary Measure

Moustapha BA, PhD
Laboratory MODAL'X of Université Paris Nanterre, France
5 bis, avenue Michel Ricard, appartement D14
92270 Bois-Colombes, France

ABSTRACT

In this paper, we revisit the fundamental question of Bitcoins security against selfish-mine attack introduced by I. Eyal and E. G. Sirer in [5]. We study the state machine of Bitcoin's network under the influence of one pool miner adopting the selfish mine strategy while the rest of the community following the standard protocol. We prove that the process following by the states of Bitcoin's system is a irreducible, positive-recurrent, aperiodic, and discrete Markov chain. We give an invariant (stationary) distribution for this Markov chain and deduce easily the rate of convergence towards the stationary equilibrium situation.

General Terms

Computer Sciences, Security, Network

Keywords

Blockchain, Bitcoin, Security, Miners, Attack, Markov chain

1. INTRODUCTION

Bitcoin is a peer to peer electronic payment system in which transactions are performed without the need for a central clearing agency to authorize transactions. Bitcoin users conduct transactions by transmitting electronic messages which identify who is to be debited, who is to be credited, and where the change (if any) is to be deposited. Bitcoin payments use Public Key Encryption. The payers and payees are identified by the public keys of their Bitcoin wallet identities. Each Bitcoin transaction is encrypted and broadcast over the network. Suppose you receive a transaction from Bob. If you can decrypt Bobs message using her public key, then you have confirmed that the message was encrypted using Bobs private key and therefore the message indisputably came from Bob. But how can you verify that Bob has sufficient bitcoins to pay you? The Bitcoin system solves this problem by verifying transactions in a coded form in a data structure called the blockchain, which is maintained and secured by a community of participants, known as miners.
The mining is the process used to confirm and secure transactions by regrouping a finite number of transaction in a block, crypt (or hash) this one, send the result to all members of the pear-to-pear network and get a reward if the block reach consensus in the network. This is a competition between some members of the network called miners. Practically, this mining process consists in two steps: First, solving a mathematical problem called also crypto-puzzle problem. In this paper, we will say with no difference *solving a mathematical problem* or simply *solving* or *finding a block* or *solving a crypto-puzzle problem*. The second step of the mining process to secure transactions is: *spreading the result to the Bitcoin network for it to reach consensus*. The first miner to do the two steps, sees his block included in the public blockchain and this miner earns a reward in Bitcoins. As it requires computational resources, the successful miner is rewarded in bitcoins or ether for his useful work. In the current implementation of Bitcoin, this reward comes from both an ex-nihilo creation of some new bitcoins and some fees Bitcoin users can add to their transactions. In order to control the monetary base, mining is made complex than it could be. And since, in the first approximation, the probability for each miner to solve a mining problem depends on his computational power, the complexity of mining is made dependent on the total computational power of all miners. Precisely, the complexity is dynamically adjusted so that a block solving and hence a creation of bitcoins occurs every ten minutes (10 mn) in expectation. Faced with this competition where the hash power is the best weapon to win remunerations in bitcoins (BTC), pool of miners are formed to add up their computing powers. Other pools less powerful may adopt fraudulent strategies such as the selfish-mine introduced by I. Eyal and E. G. Sirer and largely detailled in [5], which consists in secretly mining a block and delaying the diffusion of their blocks in the hope to earn than their fare share in the mining protocol and by consequence throw other honest miners to join them for decreasing the variance of their revenues and make their monthly revenues more predictable. A very dangerous dynamic that could allow it to control the entire network by accumulating powers of news adherents and then growing towards a majority. By considering that the propagation delay of information between any two miners in the network (unless they are linked to the same coordinator) is not negligible and follows a normal distribution with mean proportional to the physical distance between the two miners, and a constant variance independently of others delays as in [6].
In this paper, we tackle the question relative to the states of Bitcoin's system when we have a colluding pool adopting the selfish-mine strategy and when the rest of the community follows the stan-

dard protocol. We Show the Markov chain resulting the states of Bitcoin's system and the invariant measure in function of the attacker's hash power. We also deduce easily some others properties as the convergence velocity towards the stationary and equilibrium situation.

2. MODEL AND STATEMENTS OF RESULTS

2.1 Modeling miners and pools

In this part, we formalize a model that captures the essentials of Bitcoin mining behavior and introduces notation for relevant system parameters. Then we recall the selfish mining algorithm as in [5].
The system is comprised of a set of miners $\{M_1, M_2, ..., M_n\}$. Each miner M_i has mining power α_i. Since the probability of mining a block is proportional to the computational resources used for solving the associated cryptopuzzle, we can assume that $\sum_{i=1}^{n} \alpha_i = 1$. Each miner chooses a chain head to mine, and founds a subsequent block for that head after a time interval that is exponentially distributed with mean $(\alpha_i)^{-1}$. Due the nature of the mining process, the interval between mining events exhibits high variance from the point of view of a single miner. A single home miner using a dedicated ASIC is unlikely to mine a block for years [16]. Consequently, miners typically organize themselves into mining pools. All members of a pool work together to mine each block, and share their revenues when one of them successfully add a block to the public chain. While joining pool does not change a miner's expected revenue, it decreases the variance and makes the monthly revenues predictable.
We assume that miners are rational; that is, they try to maximize their revenue, and may deviate from the protocol to do so. A group of miners can form a pool that behaves as single agent with a centralized coordinator, following an unlikely strategy well know selfish mine strategy. The mining power of a pool is the sum of mining power of its members, and its revenue is divided among its members according to their relative mining power [14]. The expected relative revenue, or simply the revenue of a pool is the expected fraction of blocks that were mined by that pool out of the total number of blocks in the longest branch.

2.2 Description of Selfish-mine attack

In this section, we describe briefly the strategy called *Selfish-mine strategy* as introduced in [5]. In the next subsection we detail circumstances that can occur when the propagation delays in the network is token in account.
Indeed, we can consider principally two groups of miners in the Bitcoin community : the pool miners called dishonest miners and the others who follow the Bitcoin protocol called also the honest miners.
The dishonest minority pool follows the so-called selfish-mine strategy, and the rest, constituting a majority (in terms of hash power), follows the honest mining strategy i.e the mining protocol described by the Bitcoin protocol. We say majority in term of hash power.
A miner's strategy is called a selfish-mine strategy if he finds a block and hiding it for one moment before publishing it to the network, created a fork and develop a private chain. The consequence is that this strategy allows a pool of sufficient size to obtain a revenue larger than its ratio of mining power. In others words, it is a mining strategy that enables pools of miners that adopt it to earn revenues in excess of their mining power. Higher revenues can lead new miners to join a selfish miner pool, a dangerous dynamic that enables the selfish mining pool to grow towards a majority.
The key insight behind the selfish mining strategy is to force the honest miners into performing wasted computations on the stale public branch. Specifically, selfish mining forces the honest miners to spend their cycles on blocks that are destined to not be part of the blockchain. Selfish miners achieve this goal by selectively revealing their mined blocks to invalidate the honest miners' work. Approximately speaking, the selfish mining pool keeps its mined blocks private, secretly bifurcating the blockchain and creating a private branch. Meanwhile, the honest miners continue mining on the shorter, public branch. Because the selfish miners command a relatively small portion of the total mining power, their private branch will not remain ahead of the public branch indefinitely. Consequently, selfish mining judiciously reveals blocks from the private branch to the public, such that the honest miners will switch to the recently revealed blocks, abandoning the shorter public branch. This renders their previous effort spent on the shorter public branch wasted, and enables the selfish pool to collect higher revenues by incorporating a higher fraction of its blocks into the blockchain.
With this description, we can fully specify the selfish mining strategy, shown in Algorith 1 in [5] . The strategy is driven by mining events by the selfish pool or by the others. Its decisions depend only on the relative lengths of the selfish pool's private branch versus the public branch. It is best to illustrate the operation of the selfish mining strategy by going through sample scenarios involving different public and private chain lengths.
When the public branch is longer than the private branch, the selfish mining pool is behind the public branch. Because of the power differential between the selfish miners and the others, the chances of the selfish miners mining on their own private branch and overtaking the main branch are small. Consequently, the selfish miner pool simply adopts the main branch whenever its private branch falls behind. As others find new blocks and publish them, the pool updates and mines at the current public head. When the selfish miner pool founds a block, it is in an advantageous position with a single block lead on the public branch on which the honest miners operate. Instead of naively publishing this private block and notifying the rest of the miners of the newly discovered block, selfish miners keep this block private to the pool. There are two outcomes possible at this point: either the honest miners discover a new block on the public branch, nullifying the pool's lead, or else the pool mines a second block and extends its lead on the honest miners.
In the first scenario where the honest nodes succeed in founding a block on the public branch, nullifying the selfish pool's lead, the pool immediately publishes its private branch (of length 1). This yields a toss-up where either branch may win. The selfish miners unanimously adopt and extend the previously private branch, while the honest miners will choose to mine on either branch, depending on the propagation of the notifications. If the selfish pool manages to mine a subsequent block ahead of the honest miners that did not adopt the pool's recently revealed block, it publishes immediately to enjoy the revenue of both the first and the second blocks of its branch. If the honest miners mine a block after the pool's revealed block, the pool enjoys the revenue of its block, while the others get the revenue from their block. Finally, if the honest miners mine a block after their own block, they enjoy the revenue of their two blocks while the pool gets nothing.
In the second scenario, where the selfish pool succeeds in founding a second block, it develops a comfortable lead of two blocks that provide it with some cushion against discoveries by the honest miners. Once the pool reaches this point, it continues to mine at the head of its private branch. It publishes one block from its private branch for every block the others find. Since the selfish pool is a

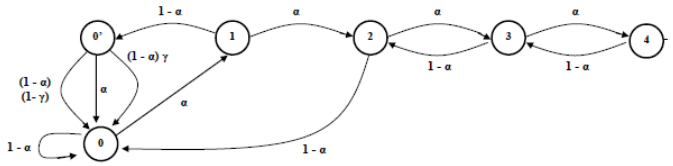

Fig. 1. the progress of the system as a state machine

minority, its lead will, with high probability, eventually reduce to a single block. At this point, the pool publishes its private branch. Since the private branch is longer than the public branch by one block, it is adopted by all miners as the main branch, and the pool enjoys the revenue of all its blocks. This brings the system back to a initial state, where there is only a single branch until the pool bifurcates it again.

3. PROBALISTIC ANALYSIS FOR STATES OF BITCOIN'S SYSTEM AND RESULTS

We can now analyze the behavior of Bitcoin system where the selfish pool has mining power of α and the others of $1-\alpha$ for $\alpha \in]0; \frac{1}{2}[$.

DEFINITION 1. *We define a state of Bitcoin's system as the lead of the selfish pool; that is, the difference between the number of unpublished blocks in the pool's private branch and the length of the public branch.*

Zero lead is separated to states 0 and $0'$. State 0 is the state where there are no branches; that is, there is only a single, global, public longest chain. State $0'$ is the state where there are two branches of length one: the main branch, and the branch that was private to the selfish miners, and published to match the main branch.

The figure 1 illustrates the progress of the system as a state machine. The transitions in the figure correspond to mining events, either by the selfish pool or by the others. Recall that these events occur at exponential intervals with an average frequency of α and $1-\alpha$, respectively.

If the pool has a private branch of length 1 and the others mine one block, the pool publishes its branch immediately, which results in two public branches of length 1. Miners in the selfish pool all mine on the pool's branch, because a subsequent block discovery on this branch will yield a reward for the pool. The honest miners, following the standard Bitcoin protocol implementation, mine on the branch they heard of first. We denote by γ the ratio (hash power) of honest miners that choose to mine on the pool's block, and the other $1-\gamma$ of the non-pool miners mine on the other branch. For state $s = 0; 1; 2; ...;$ with frequency α, the pool mines a block and the lead increases by one to $s+1$. In the state $s = 3; 4; 5; ...;$ with frequency $1-\alpha$, the honest miners mine a block and the lead decreases by one to $s-1$.

If the others mine a block when the lead is 2, the pool publishes its private branch, and the system drops to a lead of 0. If the others mine a block when the lead is one, we arrive at the aforementioned state $0'$ because the pool publishes immediately its private block. From $0'$, there are three possible transitions, all leading to state 0 with total frequency 1: (1) the pool mines a block on its previously private branch (frequency α), (2) the others mine a block on the previously private branch (frequency $(1-\alpha)\gamma$), and (3) the others mine a block on the public branch (frequency $(1-\alpha)(1-\gamma)$).

If the pool has a lead that is greater than or equal to three (a rare occurrence), it does nothing until it is notified of the discovery of a block by the community. It then publishes its first block. However, since the pool and the community will still keep working on the blocks at the ends of their respective branches, the probability to return to state 0 is null.

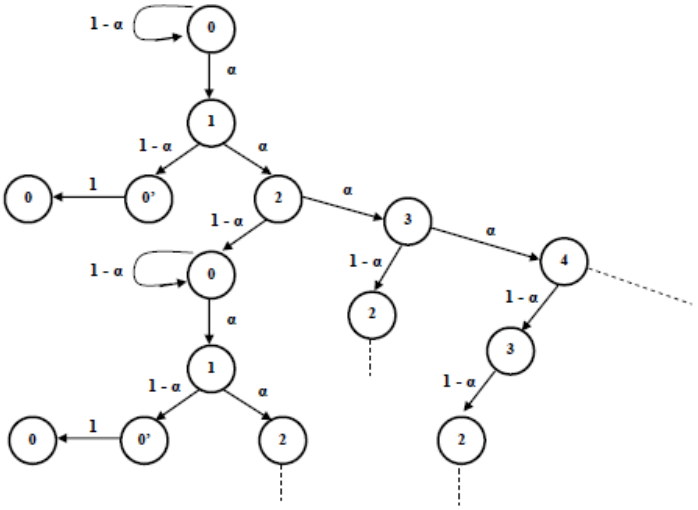

Fig. 2. The Markov chain describing the states of Bitcoin's system under selfish-mine attack of a pool miner with hash power α

The simple analyse we have done above does not take in account the bloc propagation delay in the network.

The propagation delay of information between two miners in Bitcoin network is the time needed for an announced information from a miner M_i to be received by miner M_j. This is avalaible for both honest miner and pool miners. To compare the behavior of the Bitcoin network when all miners are observing the standard protocol with its behavior and there is a pool adopting the selfish-mine strategy, the authors of [6] analyze the phenomena with more realistic assumptions. They consider that the communication between to miners M_i and M_j, either pool or honest, are not null because It can happen that different miners have different versions of the blockchain, something which occurs because of propagation delays, see Decker and Wattenhofer [3]. They assume then the propagation delay between M_i and M_j is distributed as a random variable following the Gaussian distribution with mean proportional to distance between M_i and M_j denoted by $k.d_{ij}$ and a constant variance σ^2 independently of other transmission delays. This assumption does not contradict Decker and Wattenhofer who showed that an exponential distribution provides a reasonable fit to the propagation delay distribution. Authors of [6] show that when there is no variability in the propagation delay, it follows from the Poisson network model that the value of γ (the proportion of honest miners that choose to mine on the private block released by the pool selfish miner) is zero. This is a last result of [6]. The longer the propagation delay (and thus the lead of the honest block before the secret pool is able to react), the smaller gamma becomes Our analyze start from here by considering the Figure 1 with parameter $\gamma = 0$ (the last result in [6]). By rewriting Figure 1 with $\gamma = 0$, we obtain Figure 2.

Let $(X_n)_{n\geq 0}$, the process describing the **states of Bitcoin's system** under the influence of one selfish-mine attack. We consider state 0 as the initial state or the starting state of our model. This assumption is realistic since we assume that all nodes follow the Bitcoin's protocole imposed by the Satoshi's algorithm, all nodes are honest and then we have only one public branch. The miners choose a number of transaction to mine in one block and after mining, they publish automatically the result over the network to reach consensus. For understanding of this phenomena, we suggest readers to learn [12]. To resume: state 0 is the state where there is one public branch of the blockchain. This protocol is following until one colluding pool join the network and decide to use the so-called selfish-mine strategy as defined above, by keeping secretly their mined blocks. In this case, we observe one dishonest pool (selfish-mine pool) with fraction hash power relative in the total network α ($0 < \alpha \leq \frac{1}{2}$) and the rest of the community following the standard Bitcoin protocol with fraction hash power $1-\alpha$. The pool finds a block at rate α moving the state of Bitcoin's system from 0 to 1, and the honest community finds a block at rate $1-\alpha$ keeping the system to state 0 . From state 1, there is a probability α that the selfish pool find a second block moving the system to state 2, and with probability $1-\alpha$ the honest miner succeed in founding a block on the public branch. Since they publish their block in the network by following the protocol, the selfish pool in response to the honest community, publishes automatically its secrete block. Then the system live from state 1 to state $0'$ where we have two branches of length 1, with probability $1-\alpha$. The situation resolves itself when the next block is discovered, and either the pool mine a second block or the honest community mine a second block, with high probability, the state will revert to 0 once communication has taken place. If the pool has a lead of exactly two (state 2)and the honest mine a block, it publishes automatically its private branch dropping the system to the lead of 0. Then with probability $1-\alpha$ the system reaches the state 0 from the state 2. If the pool has a lead that is greater than or equal to three (a rare occurrence), it does nothing until it is notified of the discovery of a block by the community. It then publishes its first block. However, the pool and the community will still keep working on the blocks at the ends of their respective branches and the lead decreases by one. Since the selfish pool is a minority, its lead will, with high probability, eventually reduce to a single block. At this point, the pool publishes its private branch and the system back to 0 again. So, we can assume that there exists a fixed integer N such that $\Xi = \{0; 1; 0'; 2; ...; N\}$. We recall that the state Bitcoin's system represent the lead of selfish-mine pool; the difference between the number of unpublished blocks in the pool's private branch and the length of the public branch.
In view of our analyses above, we can remark that the state's system changes from state i to state j with a probability which may be very low or very high, and this is possible for all i and j in Ξ. Since this change of states depends only on the last state by considering all preceding states contrasted by the system. This random dynamic represent a homogeneous Markov chain. It is easy to see that, there is no absorbing state. In other words, $\forall i \in \Xi$, $P(X_{n+1} = i|X_n = i) \neq 1$ (because $\alpha \neq 0$). The only state in which the Markov chain can stay longtime is the state 0 with probability $1-\alpha$ any time. An other remark is: from the state $0'$ (the state we are two brnch of lenght one), there is one and unique outcome which is state 0 (see Figure **??**). Finally, starting from any state, the system can reach any other state with probability strictly positive: $\forall i, j \in \Xi$, there exists one $k > 0$ such that: $P(X_{n+k} = j|X_n = i) > 0$. As soon as the pool publishes the last block they are keeping secrete, the system return to the state 0. Since the objective of pool selfish-mine is that the private chain becomes the longest chain, they, in one moment, absolutely publish their secrete blocks to avoid wasting their resources. And once communication is occurred we have: either the public chain stay the public chain or the pool's chain becomes the public chain. In both outcome, the state 0 is reached, because pool and honest community will agree about the new state of the blockchain. The state 0 is a recurrent state. The set Ξ is a countable and finite set. By consequence, the irreducible Markov chain $(X_n)_{n\geq 0}$ is recurrent because at last the state 0 is recurrent. The state 0 is also aperiodic because at the beginning, there is a positive probability $(1-\alpha \neq 1)$ that the system stay in the same state and by consequence the irreducible Markov chain is aperiodic. The matrix of transition probability: $P_{i,j} = P(X_1 = j|X_0 = i)$ is the following table:

$$P = \begin{array}{c|cccccccc} states & 0 & 1 & 0' & 2 & 3 & \dots & N \\ 0 & 1-\alpha & \alpha & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 1-\alpha & \alpha & 0 & \dots & 0 \\ 0' & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 2 & 1-\alpha & 0 & 0 & 0 & \alpha & \dots & 0 \\ 3 & 0 & 0 & 0 & 1-\alpha & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ N & 0 & 0 & 0 & 0 & 0 & \dots & \alpha \end{array} \quad (1)$$

The following theorem based on the perfect analysis above, allows us to study clearly the behavior of states of Bitcoin's system under selfish-mining attack.

THEOREM 2. *The process representing the states of Bitcoin's system under one selfish-attack described above, means the difference between the number of unpublished blocks in the private branch and the number of published blocks, is a homogeneous, irreducible, aperiodic and recurrent Markov chain with states space* $\Xi = \{0; 1; 0'; 2; 3; 4; 5; 6; ...; N\}$, *and no absorbing state.*

Then, there exists an unique stationary (invariant) distribution. The following theorem gives us this invariant probability.

THEOREM 3. *The stationary distribution of the Markov chain defined above is:*
$\pi = (\pi(0), \pi(1), \pi(0'), \pi(2), ..., \pi(N))$ *where:*

$$\begin{cases} \pi(0) &= \frac{1}{\alpha}\left(\frac{\alpha-2\alpha^2}{2\alpha^3-4\alpha^2+1-\alpha(1-\alpha)(\frac{\alpha}{1-\alpha})^N}\right) \\ \pi(0') &= (1-\alpha)\left(\frac{\alpha-2\alpha^2}{2\alpha^3-4\alpha^2+1-\alpha(1-\alpha)(\frac{\alpha}{1-\alpha})^N}\right) \\ \pi(1) &= \frac{\alpha-2\alpha^2}{2\alpha^3-4\alpha^2+1-\alpha(1-\alpha)(\frac{\alpha}{1-\alpha})^N} \\ \pi(k) &= \left(\frac{\alpha}{1-\alpha}\right)^{k-1}\left(\frac{\alpha-2\alpha^2}{2\alpha^3-4\alpha^2+1-\alpha(1-\alpha)(\frac{\alpha}{1-\alpha})^N}\right) \quad \forall k \in [2, N]. \end{cases}$$

PROOF. Let us call by π the stationary (invariant) distribution that we are looking for. In other words, we are looking for a measure on Ξ satisfying the following system:

$$\begin{cases} \pi = \pi P \\ \sum_{i\in\Xi} \pi(i) = 1. \end{cases}$$

where P is the matrix of transition probability defined in (1). By rewriting the above system we obtain:

$$\begin{cases} \pi(0) = \pi(0)(1-\alpha) + \pi(0) + \pi(2)(1-\alpha) \quad (L_1) \\ \pi(1) = \pi(0)\alpha \quad (L_2) \\ \pi(0') = (1-\alpha)\pi(1) \quad (L_3) \\ \pi(2) = \pi(1)\alpha + \pi(3)(1-\alpha) \quad (L_4) \\ \pi(3) = \pi(2)\alpha + \pi(4)(1-\alpha) \quad (L_5) \\ \pi(4) = \pi(3)\alpha + \pi(5)(1-\alpha) \quad (L_6) \\ . \\ . \\ . \\ \pi(N-1) = \pi(N-2)\alpha + \pi(N)(1-\alpha) \quad (L_{n+1}) \\ \sum_{i\in\Xi} \pi(i) = 1. \end{cases}$$

Combining (L_1) and (L_3), we get: $\alpha\pi(0) = \pi(1)(1-\alpha) + \pi(2)(1-\alpha)$. Adding the ligne (L_2) we deduce that $\alpha\pi(1) = (1-\alpha)\pi(2)$. And then by replacing $\alpha\pi(1)$ in (L_4), we obtain $\alpha\pi(2) = (1-\alpha)\pi(3)$. We use this equality in (L_5) and so on. We obtain $\alpha\pi(N-1) = (1-\alpha)\pi(N)$ in (L_{n+1}). The result follows the following system:

$$\begin{cases} \pi(0) = \frac{1}{\alpha}\pi(1) \\ \pi(0') = (1-\alpha)\pi(1) \\ \pi(k+1) = (\frac{\alpha}{1-\alpha})\pi(k) \quad \forall k \in [1; N-1] \\ \sum_{k=1,2,...,N} \pi(k) + \pi(0) + \pi(0') = 1. \end{cases}$$

The solution of the system is unique. The existence of invariant measure implies that the irreducible and recurrent Markov chain is a positive recurrent Markov chain also. □

REMARK 1. *If* $\alpha \in]0; 0.38[$, *we have* $\pi(0) > \pi(1) > \pi(0') > \pi(2) > \pi(3) > ... > \pi(N)$.

PROOF. For all $\alpha \in]0, \frac{1}{2}[$, we have: $\pi(0) > \pi(1) > \pi(2) > ... > \pi(N)$. Note that $\pi(1) > \pi(0')$ obviously. it suffices to remark that under hypothesis of Remark below, $\pi(0') > \pi(2)$. □

This remark shows that the fraction of time spent in state 0 during the first n passages, when n goes to infinity, is stronger than the fraction of time spent in state 1, and so on. In others words, in the stationary situation, the network spends time in on public branch than in the others states. This is correct and is in line with the description of Selfish mining attack. Indeed, when the pool publishes one secrete block in response to the competing block released by the honest miners, the system returns quickly to state 0 because Bitcoin protocole requires that all miners mine on the longest chain. In state 2 also, the pool publishes automatically theirs two secrete blocks and since the honest miners following the protocol mine with the hash they hear of first, either the private branch becomes the public branch or the public branch stays the public branch. Anyway the system drops to the lead of 0. This confirms the theoretical assumption that states stronger than 3 has very low probability to occur (see [5] and [6]).
That's why, later in this paper; the study is limited in only the space $\bar{\Sigma} = \{0, 1, 0', 2, 3\}$. This reduction of state values will allow us to get best estimation on the recurrence times. By using the well-known notations in Probability theory, we denote by P_i the law of the Markov chain $(X_n)_{n\geq 0}$ starting from the state i and the associated expectation is denoted by E_i.
The first interesting quantities we define, is the random variables

$$T^i = \inf\{n \geq 1 : X_n = i\},$$

the first time that the dynamic system reaches the state i, and

$$N_j^i = \sum_{n\geq 1}^{T^i} 1_{X_n=j},$$

the number of visits of the dynamic to the state j between two passages of i.
For a better study of this dynamic and for reader's convenience, we consider the dynamic starting from the only global public branch i.e the Markov chain $(X_n)_{n\geq 0}$ defined above and starting from the state 0 ($X_0 = 0$). We denote by P_0 the law of $(X_n)_{n\geq 0}$ and by E_0 the expectation associated to P_0. The following proposition gives the average time of recurrence of the normal situation (state 0), the situation when there is only one public and globally branch, and also the **average number of visits to the dangereous situation between two passages to the initial situation**. The so-called dangereous situation is those in which the system has two public branches of length one. More clearly we give in the next proposition, in function of the attacker's power, the average number that the system will visit the state $0'$ between two passages to the normal situation. For a colluding pool miner with a hash power α; these quantities are given in the following proposition :

COROLLARY 1.

$$E_0(T^0) = \frac{1}{\pi(0)} = \frac{1}{1-2\alpha}\left[2\alpha^3 - 4\alpha^2 + 1 - \alpha(1-\alpha)(\frac{\alpha}{1-\alpha})^N\right]$$

where N is the constant defined in Part 3.

$$E_0\left(\sum_{n=1}^{T^0} 1_{\{X_n=0'\}}\right) = \frac{\pi(0')}{\pi(0)} = \alpha(1-\alpha),$$

PROOF. The proof of the first formula is a direct consequence of Proposition 2 and Proposition 3.
For the second quantity, we define the Markovian kernel irreducible, recurrent :

$$\tilde{P}(j,i) = \frac{\pi(i)}{\pi(j)}P(i,j).$$

Let $\{Y\}$, the Markov chain associated to the Kernel $\tilde{P}$. It is easy to see that $\pi(0)P[(X_0, X_1, ..., X_k) = (0, x_1, ..., x_k)] = \pi(x_k)P[(Y_0, Y_1, ..., Y_k) = (x_k, x_{k-1}, ..., 0)]$. More general, for all bounded function $F(0, x_1, x_2, ..., x_k)$ $\pi(0)E_0[1_{X_0=0}F(0, x_1, x_2, ..., x_k)1_{X_k=x_k}] = \pi(x_k)\tilde{E}_{x_k}[1_{Y_0=x_k}F(Y_k, ..., Y_0)1_{Y_k=0}]$. We denote by $\dot{T}^0$, the first time that the Markov chain Y reaches the state 0 and by $\tilde{E}_{0'}$ the expectation associated to the Kernel $\tilde{P}$ starting from $0'$. By remarking that

$$1_{X_0=0}1_{T^0\geq n}1_{X_n=0'} = 1_{X_0=0}1_{X_1\neq 0}...1_{X_{n-1}\neq 0}1_{X_n=0'}$$

and by using the last equality, we have :
$E_0[1_{X_n=0'}1_{T^0\geq n}] = \frac{\pi(0')}{\pi(0)}\tilde{E}_{0'}[1_{\dot{T}^0\geq n}1_{Y_k=0}]$
$= \frac{\pi(0')}{\pi(0)}\tilde{E}_{0'}[1_{\dot{T}^0=n}]$. It suffices now to sum n from 1 to ∞ to get the result. □

We remark here that the recurrence time in the normal situation is decreasing with respect to the hash power of the selfish-mine pool, says α. In others words, more the pool miners is power, more the state where there is only a public global branch is reached quickly. This result makes sense. Indeed, the more mining power (resources) a pool miner applies, the better are its chances to solve the puzzle first. When the pool miners find a bloc, it keeps it secrete

waiting the information from the honest miners before publishing the result. Since the honest miners follow the standard protocol, it arrives on moment that they find a block and then immediately published it in all the network. At this time ever the pool miners has two blocks and then already published, and the pool branch remains the longest branch and then becomes the public branch; or we are in the situation of two public branches of length one. We are to the situation $0'$ and then the next bloc will determine the public branch and by consequent brings the system back to the normal situation, the state 0. The question is how many power does it dispose to reach the situation $0'$ before the normal situation. It suffices to compare the two first quantities in Proposition 1.
In the following proposition, we give the convergence toward the stationary equilibrium situation.
We recall that the state of Bitcoin's system under the influence of one colluding pool adopting the so-called selfish-mine strategy ([5]), called $(X_n)_{n\geq 0}$ such that $X_0 = 0$ is the state of Bitcoin's system starting from the situation where there are one unique public branch. It is the situation where all validated transactions will be incorpored in one bloc by miners and will be destined to the the public blockchain. This process $(X_n)_{n\geq 0}$ describing the states of Bitcoin's system is a homogeneous, irreducible, recurrent-positive and aperiodic Markov chain with stationary distribution explicitly known and given in Proposition 3. We are in conditions to use the Ergodic Theorem and also the Central Limit Theorem, as known in Markov Chain theory (see [4]), to estimate the speed of convergence towards the stationary situation.

DEFINITION 4. *Let $(X_n)_{n\geq 0}$ the state of Bitcoin's system described above and let π the invariant measure associated to this Markov chain. Let us recall $\Xi = \{0; 1; 0'; 2; ...; \mathbf{N}\}$ the countable and finite set of states defined above. For all probability measures ν in Ξ, we defined the distance in total variation between ν and π by:*

$$\|\nu - \pi\|_{VT} = \sup_{A\subseteq\Xi} |\nu(A) - \pi(A)| = \frac{1}{2}\sum_{i\in\Xi} |\nu(i) - \pi(i)|$$

where $\nu(A)$(resp. $\pi(A)$)$= \sum_{i\in A}\nu(i) := \nu[1_A]$. We define also by: $< f, g >_\pi = \sum_{i\in\Xi} f(i)g(i)\pi(i)$ and $var_\pi(f) := \pi[(f - \pi[f])^2]$.

For readers convenience, we announce the very known convergence velocity theorem of Markov chain. This result is gotten by the irreducibility and the aperiodicity of P and does not give any precision on convergence velocity toward the invariant probability measure. We recall the following notations : $P^n(0,.) = P_0(X_n = .) = P(X_n = .|_{X_0=0}$ and $P^n f(x) = \sum_{y\in\Xi} P^n(x,y)f(y)$. The following theorem can be found in all book relating the probability theory.

THEOREM 5 :WELL KNOW CONVERGENCE VELOCITY. *There exists a $\rho \in]0, 1[$ such for all f defined in Ξ,*

$$var_\pi(P^n f) \leq \rho^n var_\pi(f). \quad (2)$$

In next Proposition, we provide a geometric upper bound for the speed of convergence towards the equilibrium measure. This allows us to control the convergence error and give the needed time to get the stationary situation of the dynamic.

PROPOSITION 1. *Let $\{X_n, P_0, P, \pi\}$ the irreducible, aperiodic, and positive recurrent Markov chain starting from 0 in values in $\Xi = \{0, 1, 0', 2, ..., N\}$, with matrix of transition probabilities P defined in (1). Let π the unique invariant measure. Then, there exists $\rho \in]0, 1[$ such that:*

$$d(n) := \|P^n(0,.) - \pi\|_{VT} \leq \frac{\rho^{\frac{n}{2}}}{2}\frac{1}{min_{i\in\Xi}\pi(i)} \leq \frac{\rho^{\frac{n}{2}}}{2\pi(N)}, \quad (3)$$

where $\pi(N) = \left(\frac{\alpha}{1-\alpha}\right)^{N-1}\left(\frac{\alpha-2\alpha^2}{2\alpha^3-4\alpha^2+1-\alpha(1-\alpha)(\frac{\alpha}{1-\alpha})^N}\right)$.

PROOF. We set:

$$P^*(i,j) = \frac{\pi(j)}{\pi(i)}P(j,i). \quad (4)$$

Remark that for all $i \in \Xi$, $P^*(i,.)$ is a probability measure on Ξ and P^* is the matrix adjoin of P in the sens of (4). Since $\pi P = P$, then $\pi P^n = \pi$ and $\pi(i)P^{*n}(i,j) = \pi(j)P^{*n}(j,i)$ also.
By recalling δ_0 the mass of Dirac in 0, we set $(\delta_0 P^n)(.) := P^n(0,.)$. Let us recall Ξ defined above and let A a subset of Ξ. We have:
$(\delta_0 P^n)(A) - \pi(A) = (\delta_0 P^n)[1_A] - \pi[1_A]$
$= \delta_0(P^n 1_A) - \pi[1_A] =< \frac{\delta_0}{\pi}, P^n 1_A >_\pi - < 1_\Xi, 1_A >_\pi$.
We remark also that for all $i \in \Xi$, $1 = 1_\Xi(i) = P^n 1_\Xi(i) = \frac{1}{\pi(i)}.\pi P^{*n}(i)$. So we obtain by using (4):

$$(\delta_0 P^n)(A) - \pi(A) =< P^{*n}\frac{\delta_0}{\pi} - \frac{1}{\pi}.\pi P^{*n}, 1_A >_\pi .$$

Otherwise, for all constant $c :< P^{*n}\frac{\delta_0}{\pi} - \frac{1}{\pi}.\pi P^{*n}, c >_\pi = 0$. By taking $c = \pi(A) = \pi[1_A]$, we get:
$(\delta_0 P^n)(A) - \pi(A) =< P^{*n}\frac{\delta_0}{\pi} - \frac{1}{\pi}.\pi P^{*n}, 1_A - \pi[1_A] >_\pi$
$=< P^{*n}\frac{\delta_0}{\pi} - 1, 1_A - \pi[1_A] >_\pi$.
Applying Cauchy-Schwartz, and remarking that $\pi[P^{*n}\frac{\delta_0}{\pi}] = 1$, we obtain:
$|(\delta_0 P^n)(A) - \pi(A)| \leq \sqrt{var_\pi(P^{*n}\frac{\delta_0}{\pi})}\sqrt{var_\pi(1_A)}$. In view of definition (4) and the fact that $\pi(i) > 0$ for all $i \in \Xi$, P^* is also irreducible. Then, applying (2) above, we obtain:
$|(\delta_0 P^n)(A) - \pi(A)| \leq \rho^{\frac{n}{2}}\sqrt{var_\pi(\frac{\delta_0}{\pi})}\sqrt{var_\pi(1_A)}$
$\leq \frac{1}{2}\rho^{\frac{n}{2}}\sqrt{var_\pi(\frac{\delta_0}{\pi})} \leq \frac{\rho^{\frac{n}{2}}}{2}\frac{1}{\min_{i\in\Xi}\pi(i)}$
$\leq \frac{\rho^{\frac{n}{2}}}{2}\frac{1}{\pi(N)}$, where $\pi(N) = \left(\frac{\alpha}{1-\alpha}\right)^{N-1}\left(\frac{\alpha-2\alpha^2}{2\alpha^3-4\alpha^2+1-\alpha(1-\alpha)(\frac{\alpha}{1-\alpha})^N}\right)$. □

We give in following Remark, the best speed of convergence for $N = 3$. Recall that the probability to get the state 3, says, the probability that pool miners gets three blocs ahead, by having only a hash power α is very lower. That's why the study of this model for $N = 3$ is enough to get a high level control on the system.

REMARK 2. *If $N = 3$, the best constant is reached at $\alpha = \frac{1}{2}$.*

PROOF. $\pi(N) = \pi(N, \alpha)$ is an increasing function on $]0; \frac{1}{2}[$ for all fixed N. Minimizing $\frac{1}{\pi(N)}$ as a function of α, we get the result. □

4. CONCLUSION

We have shown in this model that the process resulting the states of the Bitcoin system under the influence of the attack is an irreducible Markov Chain, positive-recurrent and aperiodic. We have also given its invariant measure and the speed of convergence to the stationary equilibrium situation. The ergodic theorem and the

Central limit Theorem can be deduced easily.
We emphasize that our model is purely analytic and is based on an analysis of an algorithm of article [5]. These results nevertheless remain a big step for a more thorough study of the selfish attack. Combined with tomographic techniques to obtain the exact topology of the Bitcoin network, our analyzes will make it possible to characterize this attack under the effect of the propagation delay of the information.

5. AKNOWLEDGEMENTS

The author would like to thank some members of MODAL'X for theirs lectures, discussions and comments. He would like also to thank the many thoughtful individuals from the Bitcoin Forum, whose ideas helped form the basis of this work, as well as the community of bitcoin for their encouragement and enthusiasm. Finally, he is thankful to Marie Delahaye for his continuous support, appreciation and help for writing this manuscript. []

6. REFERENCES

[1] Andes. Bitcoin's cryptonite: the 51% attak, 2011.

[2] Gavin Andressen. Back-of-the-envelope calculations for marginal cost of transactions, 2014, 03.

[3] Christian Decker and Roger Wattenhofer. Information Propagation in the Bitcoin Network. September 2013.

[4] Pardoux Etienne. *Markov Processes and Applications: Algorithms, Networks, Genome and Finance*. Willey series of Probability and Statistic, 2008.

[5] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. pages 436–454, 2014.

[6] Johannes Göbel, Holger Paul Keeler, Anthony E. Krzesinski, and Peter G. Taylor. Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay. *Perform. Eval.*, 104:23–41, 2016.

[7] Jens Hougaard, Juan Moreno-Ternero, Mich Tvede, and Lars Peter sterdal. Sharing the proceeds from a hierarchical venture. *Games and Economic Behavior*, 104:23–41, 2016.

[8] Nicolas Houy. It will cost you nothing to 'kill' a proof-of-stake crypto-currency, 2014,01.

[9] Nicolas Houy. The bitcoin mining game. *Ledger*, vol. 1:pp. 53–68, 2016.

[10] Joshua A. Kroll, Ian C. Davey, and Edward W. Felten. The economics of bitcoin mining, or bitcoin in the presence of adversaries, 2013.

[11] Yoad Lewenberg, Yoram Bachrach, Yonatan Sompolinsky, Aviv Zohar, and Jeffrey S. Rosenschein. Bitcoin mining pools: A cooperative game theoretic analysis. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 919–927, 2015.

[12] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Bitcoin.org*, 2008.

[13] John Von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[14] Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems, 2011.

[15] Yonatan Sompolinsky and Aviv Zohar. Bitcoin's security model revisited. *CoRR*, abs/1605.09193, 2016.

[16] E. Swanson. Bitcoin's mining calculator, 2013.