Stalker - Uma Nova Estratégia para o Atacante Egoísta em Blockchains

Emanuel Ferreira Jesus, Vanessa R. L. Chicarino, Célio Albuquerque, Antônio A. de A. Rocha

{efjesus,vanessarlc}@id.uff.br, {celio,arocha}@ic.uff.br

[1]Instituto de Computação (IC), Universidade Federal Fluminense (UFF), Rio de Janeiro, Brasil

Resumo. *Blockchain é um banco de dados distribuído que mantém uma lista de transações em uma estrutura de dados chamada bloco, que ligados ao seu anterior, formam uma cadeia de blocos. Foi proposto com a finalidade de facilitar transações financeiras, seguras, públicas e auditáveis. No entanto, possui aplicação em diversas áreas e está sujeita a ataques.Um dos ataques mais conhecidos é a mineração egoísta que, sob certas condições, pode resultar em uma parcela desproporcional de recompensas a atacantes que se desviam do comportamento honesto. O objetivo deste artigo é apresentar e analisar o Stalker, uma estratégia de mineração egoísta com o objetivo de impedir a publicação de um bloco ou transação específica. Resultados mostram que a eficácia do ataque é proporcional à relação entre o poder computacional do atacante e do alvo.*

Abstract. *Blockchain is a distributed database that maintains a list of transactions in a data structure called block, which connected to the previous one, builds a chain of blocks. It was designed with the purpose of providing secure, public and auditable financial transactions. Nevertheless, it can be deployed in many areas and it is prone to attacks. One of the most well-known attack is the Selfish Mining, which under certain conditions, can result in a disproportionate share of rewards to attackers straying from the honest behavior. This paper introduces and analyses the Stalker, a strategy used by a Selfish Miner to prevent a specific block or transaction from being published. Results show that the efficacy of this strategy is directly proportional to the computational power relation between the attacker and the target. Results show that the Stalker can block up to 40% of target blocks and has a collateral damage of dropping blocks from other nodes.*

1. Introdução e motivação

O "protocolo da confiança" [Tapscott and Tapscott, 2016], ou Blockchain, é uma tecnologia que visa a descentralização como medida de segurança. É uma base de registros e dados, distribuídos e compartilhados, que possui a função de criar um índice global para todas as transações que ocorrem em uma determinada rede. Funciona como um livro-razão, de forma pública, compartilhada e universal, que cria consenso e confiança na comunicação direta entre duas partes, ou seja, sem o intermédio de terceiros. Está constantemente crescendo à medida que novos blocos são adicionados. A cadeia de blocos também pode ser usada para controle de trânsito de mercadorias em uma cadeia de suprimentos, contratos inteligentes, gerenciamento de identidade digital e em uma série de outras aplicações [Pilkington, 2016].

Neste contexto, a tecnologia Blockchain também pode ser usada para Autenticar, Autorizar e Auditar os dados gerados pelos mais diversos serviços. Em virtude de sua natureza descentralizada, elimina a necessidade de confiança em terceiros e não possui um ponto único de falha. Este trabalho tem o objetivo de apresentar uma estratégia do ataque do minerador egoísta, nomeada de *Stalker Mining*, que busca impedir a publicação de um bloco ou transação específica.

O restante deste trabalho está estruturado em 4 seções. A Seção 2 apresenta conceitos básicos de Blockchain. Na Seção 3 é apresentada uma revisão dos ataques discutidos na literatura. A Seção 4 introduz o *Stalker Mining* e apresenta uma análise da eficácia deste ataque. Finalmente, a última seção apresentará as considerações finais e questões em aberto.

2. A Blockchain

A Blockchain foi proposta por Satoshi Nakamoto [Nakamoto, 2008], sua ideia inicial foi permitir que operações financeira sejam realizadas sem a necessidade de intermediários. Esta tecnologia está apoiada em três pilares: Livro Razão Aberto (*Open Ledger*), descentralização, e consenso. Um livro razão aberto é uma estrutura de dados registra todas as transações ocorridas na rede, e todos os nós desta rede podem escrevê-lo. A descentralização permite que haja várias cópias do livro na rede, eliminando o ponto único de falha. Em virtude da característica pública e da descentralização, é necessário um mecanismo de consenso para sincronização das diversas cópias do livro e validação das transações escrituradas. Assim, o terceiro e último pilar é consenso, onde alguns participantes especiais, denominados mineradores ou validadores[1], concorrem entre si para para realizar três tarefas: validar as transações, ligá-la com a anterior e provar que foi o primeiro minerador a calcular uma chave especial que liga as transações. Para encontrar esta chave, é necessário resolver um problema que, normalmente, requer grande poder computacional e uma quantidade de tempo significativa. O primeiro minerador que conseguir realizar estas três tarefas informará aos demais nós da rede que a transação é válida e a chave encontrada. Assim, caso a solução esteja correta, todos os usuários atualizarão o seu próprio livro com a nova transação e a rede alcança um novo consenso.

A principal diferença entre as diversas implementações da Blockchain é o mecanismo de consenso. Alcançar o consenso em um sistema distribuído é um desafio. Os algoritmos de consenso devem ser resilientes a falhas de nós, particionamento da rede, atrasos de mensagens, mensagens que chegam fora de ordem e corrompidas. Eles também têm que lidar com nós egoístas e deliberadamente maliciosos. Vários algoritmos têm sido propostos para resolver estes problemas, cada um realizando o conjunto de suposições em termos de sincronia, transmissões de mensagens, falhas, nós maliciosos, desempenho e segurança das mensagens trocadas. Uma rede Blockchain alcançar o consenso significa garantir que todos os nós na rede concordem com um estado global da cadeia de blocos. A maioria das plataformas Blockchain existentes utilizam o mecanismo de consenso, na sua forma original e computacionalmente cara, chamada de Prova de Trabalho (PoW, do termo em inglês *Proof-of-Work*) [Nakamoto, 2008].

Este conceito, inicialmente proposto por Adam Back [Back et al., 2002], consiste em realizar uma colisão parcial de hash. É estabelecido que o hash do cabeçalho do bloco

[1]Neste trabalho usaremos o termo *minerador* para nos referirmos a estes participantes.

deve iniciar com uma determinada quantidade de zeros. Para que isso seja possível, o cabeçalho possui um campo especial chamado de *Nonce*, que é alterado constantemente até que o hash do cabeçalho atenda aos requisitos. O objetivo é provar que foi gasto tempo e energia para encontrar a solução da colisão. A principal desvantagem deste mecanismo é que ele é computacionalmente muito custoso. Existem vários outros mecanismos que oferecem certas vantagens desejadas em relação ao modelo original, como detalhado em [Chicarino et al., 2017]: Prova de Posse (PoS, do termo em inglês *Proof of Stake*); Algoritmo de Tolerância a Falhas Bizantinas (PBFT, do inglês *Practical Byzantine Fault Tolerance*) e a Prova do Tempo Decorrido (PoeT, do inglês *Proof of Elapsed Time*).

A Blockchain pode ser classificada com base no acesso aos dados e participação no mecanismo de consenso como: Blockchain Pública (permissionless) e Blockchain Privada (permissioned) [Chicarino et al., 2017]. Muitos projetos foram iniciados para tornar a Blockchain mais popular e viável para diferentes modelos de negócios e aplicações, aproveitando as categorias existentes. A Tabela 1 resume as principais características de algumas aplicações que usam Blockchain. Bitcoin, Ethereum [Wood, 2014] são exemplos de Blockchain públicas e Hyperledger [Cachin, 2016] e Ripple [Pilkington, 2016] são exemplos de Blockchain privadas. É possível verificar uma diferença crítica entre essas duas categorias que é o modelo de mineração. Blockchains públicas usam a Prova de Trabalho, onde o poder computacional é oferecido para criar confiança. Como todos os nós, nas Blockchains privadas, são conhecidos não é necessário usar um mecanismo custoso para alcançar o consenso, o que acarreta em um menor tempo de processamento de bloco, sendo considerado, praticamente, em tempo real.

Tabela 1. Comparação entre sistemas Blockchain

Blockchain	**Bitcoin**	**Ethereum**	**Hyperledger**	**Ripple**
Natureza	Pública	Pública	Privada	Privada
Validação	PoW SHA-256	PoW - ethash	PBFT	BFT customizado
Propósito	criptomoeda	contrato inteligente	Chaincode	criptomoeda
Linguagem	Scripts em pilha	Código interno Turing completo	Go, Java	C++
T. proc. do bloco	$\sim$ 600 s	$\sim$ 15 s	$\sim$ tempo real	$\sim$ tempo real

A evolução da Blockchain é dividida em três fases [Swan, 2015]: Blockchain 1.0, 2.0 e 3.0. A Blockchain 1.0 é o uso para criptomoedas, transferências e sistemas de pagamento digital, amplamente difundido pelo Bitcoin e derivados. A Blockchain 2.0 é o uso com contratos inteligentes, toda a lista das questões econômicas, mercado e aplicações financeiras que a utilizam de maneira mais extensa, como: ações, títulos, empréstimos, hipotecas. A Blockchain 3.0 refere-se o seu uso em aplicações além da moeda, finanças e mercados, particularmente nas áreas de governo, saúde, ciência, Internet das Coisas. Diversas soluções estão sendo propostas para o uso da Blockchain 3.0, dentre as quais é possível citar: Suportar Ambientes Inteligentes [Dorri et al., 2017]; Estabelecer mecanismos de pagamento para IoT [Wörner and von Bomhard, 2014]; Realizar Computação Segura entre Múltiplos Participantes (MPC) [Zyskind et al., 2015]; e Criar serviços de Infra-Estrutura Pública de Chaves (PKI) [Nguyen et al., 2015, Ali et al., 2016].

3. Ataques a Blockchain

Nas Blockchain públicas qualquer nó pode criar um bloco, e teoricamente alterar blocos de uma cadeia. As transações são consideradas válidas quando estão em um bloco. Mas, como existe a possibilidade de um bloco ser alterado, uma transação somente é considerada confirmada quando outros blocos forem adicionados a cadeia. Isso se deve ao fato de que para alterar uma transação é necessário possuir grande poder computacional para alterar o bloco onde a transação se encontra e os posteriores, assim existe uma quantidade de blocos que torna computacionalmente impossível alterar a cadeia. Para criar blocos os nós recebem alguns incentivos econômicos. Parte desses incentivos é dada pelas taxas pagas pelos usuários em cada transação. Além de remunerar os nós que gastam recursos para criar os blocos, as taxas tem um outro papel, evitar ataques de negação de serviço.

Os nós que realizam o trabalho de gerar os blocos são chamados de mineradores. Como existem diversos mineradores gerando blocos de forma descentralizada, pode ocorrer de dois mineradores gerarem blocos quase ao mesmo tempo. Os novos blocos enviados por eles podem resultar em visões diferentes, chamada de bifurcações ou *forks*. Se uma parte dos nós adotar um bloco e outra parte adotar o outro, essas duas cadeias irão coexistir até que uma fique maior que a outra. Para resolver esta situação, os nós que se comportam de maneira honesta sempre irão adotar a maior cadeia e o fork estará resolvido. Na Figura 1 os blocos cinza bifurcaram da cadeia principal, como alcançaram uma altura maior passaram a ser a cadeia principal. Os blocos brancos 127,128 e 129 são descartados.

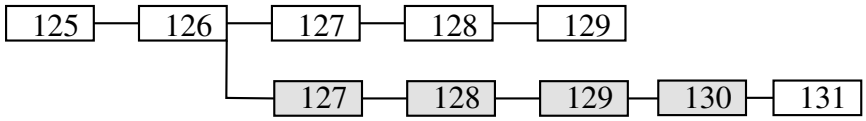

Figura 1. Bifurcação

Conforme descrito anteriormente, podem surgir bifurcações de curta duração. Essa bifurcações coexistem por um pequeno espaço de tempo. Quando o próximo bloco é minerado os nós escolhem a cadeia mais longa e descartam a outra. A maioria das bifurcações ocorre naturalmente, sem má intensão, fazendo com que as poucas transações do lado descartado sejam atrasadas. Esta abordagem funciona bem, sob o pressuposto crucial de que nenhum atacante deve ser capaz de reunir tanto poder computacional que seja capaz de falsificar e publicar uma "cadeia alternativa" que tenha maior dificuldade total. Nesse caso, as regras de consenso fariam com que cadeia alternativa fosse adotada ao invés da cadeia honesta. O intervalo para a inclusão de novos blocos é crucial para a quantidade de *forks* observados na rede. Quanto menor for esse intervalo, maior será a quantidade de blocos gerados e consequentemente maior será a probabilidade de ocorrência de *forks*. Decker [Decker and Wattenhofer, 2013] verificou a existência de 169 *forks* durante sua observação de 10.000 blocos na rede Bitcoin, ou seja 1,69% dos blocos foram descartados pela ocorrência de *forks*. Gervais [Gervais et al., 2016] analisou o impacto da diminuição deste tempo, variando a taxa de inclusão de novos blocos entre 0,5 segundos a 25 minutos, ele obteve seus resultados via simulações no NS-3 [Carneiro, 2010], conforme apresentado na Tabela 2.

Tabela 2. Impacto do intervalo entre blocos na taxa de *forks*

Intervalo entre blocos	Taxa de Forks(%)	Mediana do tempo de propagação
20s	3,2	1,21
30s	2,54	1,43
1m	2,15	2,08
2,5m	1,82	4,18
10m	1,51	14,7
25m	1,72	35,73

Uma das preocupações mais comuns para os sistemas de moedas digitais é a possibilidade do gasto duplo, quando um usuário malicioso gasta um mesmo valor em duas transações diferentes da cadeia. Note que para ocorrer a tentativa de um gasto duplo é necessário uma bifurcação, pois se o gasto ocorrer na mesma cadeia, quando o novo bloco for criado, ele não passará nas verificações iniciais de consistência e será descartado. Com o fork, o usuário malicioso faz um gasto e envia para rede, gasta a mesma quantia novamente em outro lugar e começa a minerar sobre esse gasto. Desta forma, há a possibilidade de ele conseguir minerar um bloco e realizar o fork. A partir deste momento, a rede estará dividida e haverá uma corrida que será vencida pela maior cadeia. Uma das transações será descartada e o gasto duplo será rejeitado. Como uma das cadeias irá ser aceita pela rede, e a outra descartada, eventualmente o gasto duplo será detectado. É usualmente aceito na rede Bitcoin que uma transação é considerada confirmada quando existem seis novos blocos com altura maior que a sua, pois será necessário muito esforço para alterá-la.

Um ataque contra o mecanismo de consenso é o "ataque de 51%". Nesse cenário, um grupo de mineradores, controlando uma maioria (51%) [Kroll et al., 2013] do poder de hash total da rede, conspira para atacar o mecanismo de consenso. Com a habilidade de minerar a maioria dos blocos, os mineradores atacantes podem gerar bifurcações deliberadas na Blockchain, gerar transações de gasto duplo ou executar ataques de negação de serviço (DoS) contra endereços ou transações específicas. Um ataque de bifurcação/gasto duplo é um ataque onde o atacante faz com que blocos já confirmados sejam invalidados ao fazer uma bifurcação em um nível abaixo deles, com uma posterior re-convergência em uma cadeia alternativa. Com poder suficiente, um atacante pode invalidar seis ou mais blocos em uma sequência, invalidando transações que antes eram consideradas imutáveis (com seis confirmações). Note que o gasto duplo só pode ser feito nas transações do próprio atacante, para as quais o atacante pode produzir uma assinatura válida. Fazer um gasto duplo da própria transação é rentável quando, ao invalidar uma transação, o atacante puder receber um pagamento irreversível ou um produto sem ter que pagar por isso.

Um atacante pode desviar do comportamento padrão e manter secretamente uma cadeia onde somente ele produza blocos, e adote uma heurística própria para divulgação destes blocos. Neste momento, todos, inclusive o atacante estariam minerando sobre o bloco n. Ao realizar esta ação, caso ele consiga produzir o próximo bloco, o atacante conseguirá uma vantagem em relação aos demais mineradores, mesmo sem possuir maior poder computacional, pois ele poderá iniciar o processo de mineração do bloco $n + 1$ antes de todos os outros mineradores. Se o atacante receber um bloco da rede ele pode

decidir adotar este bloco e jogar fora seu trabalho, ou ignorar o bloco recebido e continuar minerando na cadeia privada. Como ele começou a minerar antes, há uma probabilidade de minerar bloco $n + 1$ antes dos outros, gerando um fork com altura maior que a cadeia principal. Como os demais nós se comportam honestamente, eles também irão adotar esta cadeia e o atacante atingirá seu objetivo. Estes ataques são chamados na literatura de *Selfish Mining* [Eyal and Sirer, 2014, Nayak et al., 2016]. Em [Eyal and Sirer, 2014], Eyal faz uma análise matemática e propõe um modelo de transição de estados para capturar o melhor momento do atacante liberar seus blocos privados. Ele analisa as probabilidades de ocorrência de cada estado e chega a conclusão que para o atacante conseguir êxito, ou seja, publicar mais blocos que a cadeia honesta, seu poder de mineração deve satisfazer a seguinte inequação:

$$\frac{1-\gamma}{3-2\gamma} < \alpha < \frac{1}{2}$$

onde α é o poder de mineração do atacante e γ é a razão dos nós honestos que escolhem minerar a cadeia desonesta. Isso é uma vantagem ao atacante, pois ele necessitará de menor poder para conseguir suplantar os nós honestos. Assim, a partir desta inequação é obtido o menor valor de α para um determinado γ. O pior caso para o atacante é quando nenhum nó honesto adota a sua cadeia, neste caso é necessário que o atacante possua um terço do poder computacional da rede.

O *Eclipse* [Heilman et al., 2015] é um ataque a nível de rede. Ocorre quando um atacante monopoliza todas as conexões de um determinado nó, isolando a vítima e filtrando todas as mensagens enviadas e recebidas. Como resultado, a vítima tem uma visão diferente da cadeia, pode ter seus blocos impedidos de serem incluídos na cadeia e pode ser forçada a trabalhar na cadeia do atacante. Na rede Bitcoin, os nós mantém no máximo 125 conexões com seus vizinhos, sendo 8 conexões de saída, e 117 de entrada. As conexões de saída são aquelas iniciadas pelo próprio nó, e as de entrada são as que outros nós solicitam. Para armazenar o endereço destas conexões os nós usam duas tabelas, uma tabela de conexões bem sucedidas, onde são armazenadas as informações de todas as conexões estabelecidas, de entrada e de saída, e uma tabela de endereços fornecidos, por solicitação ou não. A primeira é chamada de *Tried Table* e a segunda de *New Table*. O ataque consiste em encher a *Tried Table* com endereços controlados pelo atacante e encher a *New Table* com endereços inválidos. Desta forma, sempre que a vítima buscar uma nova conexão ela se conectará a um endereço controlado pelo atacante. Em seus experimentos o autor usou uma *botnet* com 400 máquinas. Conseguiu popular totalmente a *New Table* e 60% da *Tried Table*, alcançando sucesso em controlar todas as conexões da vítima em 80% dos ataques.

Nayak [Nayak et al., 2016] amplia a pesquisa de Eyal e verifica que o ataque proposto anteriormente não é ótimo. Ele propõe novas estratégias para aumentar o ganho do atacante, levando em conta não apenas o tamanho das cadeias, mas também seu poder computacional. Por exemplo, mesmo que o atacante esteja perdendo a corrida, caso ele possua uma parcela significativa de poder computacional, é melhor que ele continue a minerar em sua cadeia privada, pois ele terá uma grande chance de alcançar e ultrapassar a cadeia honesta. Outra contribuição de seu trabalho é mostrar que se o *Selfish Mining* for combinado com o *Eclipse* [Heilman et al., 2015], quando o atacante controla todas as conexões com um determinado nó, o atacante aumentará seus ganhos e surpreendentemente, com determinados parâmetros, o nó que foi alvo do *Eclipse* também conseguirá

publicar mais blocos, em relação aos nós honestos.

O autor modela a mineração como um processo de decisão de Markov, que usa como informação as transições de estado, quando os mineradores acham um novo bloco. Quando um nó honesto produz um novo bloco ele o publica imediatamente, enquanto o atacante mantém seus blocos ocultos. Depois, ele captura quantos blocos cada cadeia estão a frente uma da outra, e se os nós honestos estão minerando sobre a cadeia honesta ou desonesta. Suas estratégias são chamadas de *Stubborn Mining*.

4. A Estratégia *Stalker*

O atacante *Stalker* é uma variante do *Selfish Miner*. Enquanto o *Selfish* tem por objetivo auferir maiores ganhos em relação aos nós honestos o *Stalker* tem o objetivo de impedir que um determinado nó publique um bloco, ou que uma determinada transação seja incluída na cadeia.

O atacante mantém duas cadeias e usa suas alturas relativas para tomar as decisões de ataque. O atacante se comporta conforme algoritmo 1, onde la corresponde a altura da cadeia do atacante e lh a altura da cadeia honesta. A cada novo bloco, recebido ou minerado, o atacante verifica qual cadeia é maior, e usa uma matriz de decisão para verificar que ação tomar. As ações são descritas a seguir:

Algoritmo 1: Stalker Attack

```
1  if Próximo Bloco == vítima then
2      if la > lh then
3          Publicar a cadeia do atacante;
4          return;
5      end
6  end
7  switch Ler_Matriz do
8      case Esperar do
9          Continuar minerando na cadeia do atacante;
10     end
11     case Adotar do
12         Re-iniciar o ataque;
13     end
14 end
```

- **Publicar:** Corresponde a revelar sua cadeia, ocorre quando o atacante estava no estado esperar e recebe um bloco do alvo (Algoritmo 1 linha 3).
- **Esperar:** O atacante minera constantemente em sua cadeia, sem a revelar. Ocorre quando a cadeia do atacante é maior que a honesta, mas o atacante não recebeu um bloco do alvo (Algoritmo 1 linha 8).
- **Adotar:** O Adversário adota a cadeia honesta. Isto corresponde a reiniciar o ataque, pois o atacante infere que a cadeia dos nós honestos possuem uma probabilidade maior de vencer a corrida (Algoritmo 1 linha 11).

Na Figura 2 é ilustrado um exemplo do *Stalker*, cujo objetivo é impedir que o nó azul publique seus blocos. Os blocos em cinza correspondem aos nós honestos e os blocos

Tabela 3. Matriz de Decisão

la \ lh	1	2	3	4	5	6	7	8	9	10	11
1	“w”, “w”	-, “a”	-	-	-	-	-	-	-	-	-
2	“w”, “w”	“w”, “w”	-, “a”	-	-	-	-	-	-	-	-
3	“w”, “w”	“w”, “w”	“w”, “w”	-, “a”	-	-	-	-	-	-	-
4	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	-, “a”	-	-	-	-	-	-
5	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	-, “a”	-	-	-	-	-
6	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	-, “a”	-	-	-	-
7	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	-, “a”	-	-	-
8	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	-, “a”	-	-
9	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “w”	“w”, “a”	-, “a”	-
10	“a”, “w”	“a”, “w”	“a”, “w”	“a”, “w”	“a”, “w”	“a”, “w”	“a”, “w”	“a”, “w”	“a”, “w”	“w”, “w”	-, “a”
11	-	-	-	-	-	-	-	-	-	“a”, “a”	“w”,“w”

encarnados são os minerados pelo atacante. Em um primeiro cenário, a esquerda, o nó azul não publica nenhum bloco, e o atacante decide adotar a cadeia principal, descartando seu trabalho. No segundo cenário, a direita, o nó azul publica o bloco 4. O atacante decide publicar sua cadeia, que é mais longa que a principal, impedindo que o bloco azul faça parte da cadeia.

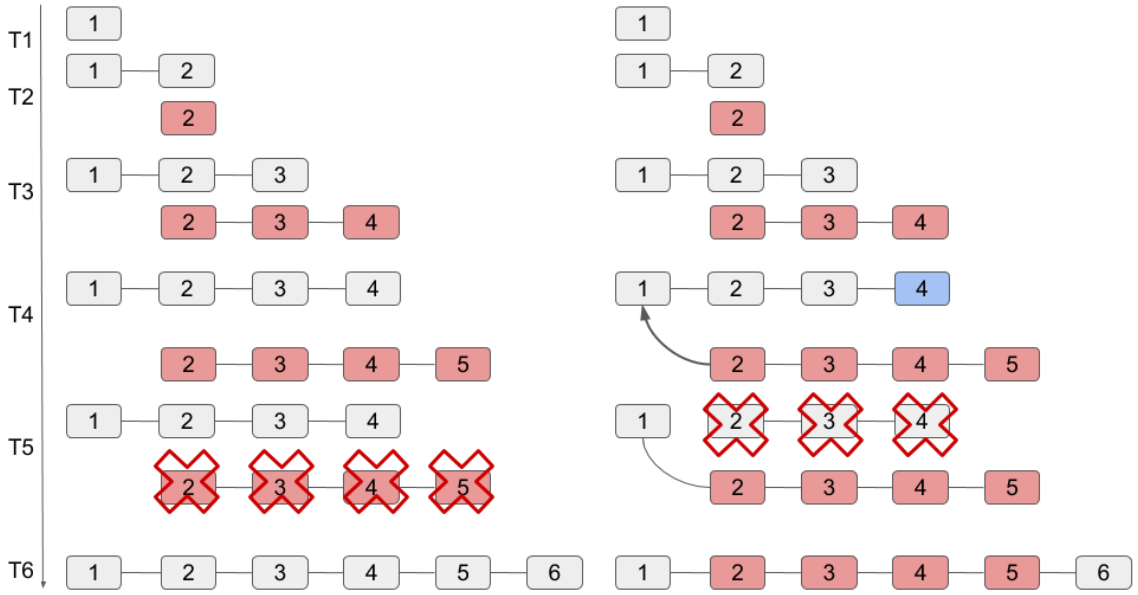

Figura 2. Exemplo do ataque

A heurística adotada é a mesma usada pelo *Selfish miner*, e pode ser observada na Tabela 3. As linhas da tabela correspondem à altura relativa da cadeia do atacante, e as colunas à cadeia dos nós honestos, “a” corresponde a ação adotar, e o “w” a ação esperar. O primeiro valor de cada célula é usado quando o próximo bloco foi minerado por um nó honesto, e o segundo valor é usado quando o próximo bloco for minerado pelo atacante. Existem duas cadeias concorrentes. Uma pública vista por todos os nós, e uma privada que apenas o atacante tem conhecimento. A partir do momento que o atacante inicia o ataque ele atribui valor zero a altura das duas cadeias. Quando recebe ou minera o próximo bloco o atacante altera também a altura da cadeia correspondente, consulta a matriz e toma a ação decorrente. Sempre que o atacante recebe um bloco do alvo e a

cadeia privada é maior que a honesta ele publica sua cadeia.

Figura 3. Topologia

Para realizar as simulações foi utilizado o NS-3 com o módulo do ataque de mineração egoísta criado por Gervais [Gervais et al., 2016]. Foram feitas alterações de modo a alterar o comportamento do atacante para que ele realize os ataque conforme o Algoritmo 1. A topologia utilizada foi composta por 128 nós, sendo 16 mineradores, dentre os quais 1 atacante, conforme ilustrado na Figura 3. A fim de avaliar a influência da centralidade no ataque foram realizadas simulações alterando a quantidade de conexões entre os nós. O objetivo é verificar se a centralidade ajuda o atacante a ser mais eficaz. Conforme pode ser observado na Tabela 4 a quantidade de blocos publicados pelo nós honestos tem uma leve elevação. Contudo, não é possível afirmar que a centralidade exerce influência no ataque, pois a variação da quantidade de blocos publicados é de aproximadamente 2%, que está dentro do intervalo de confiança.

Tabela 4. Centralidade

{Grau} {Closeness} {Betweeness}	Cadeia Principal Honestos	σ	Cadeia Principal Atacante	σ
{0,26} {0,41} {0,02}	4272	95	2154	59
{0,64} {0,54} {0,08}	4307	80	2185	103
{1,39} {0,66} {0,24}	4330	135	2167	122
{2,15} {0,80} {0,44}	4362	143	2146	90
{2,88} {1,00} {0,61}	4377	107	2138	99

Para verificar a eficácia do ataque foi utilizado como métrica a taxa de blocos descartados, blocos que foram produzidos pelos nós honestos, mas não foram incluídos na cadeia principal em virtude do *fork* produzido pelo atacante. Como comparação utilizamos os trabalhos de Decker [Decker and Wattenhofer, 2013] e Gervais [Gervais et al., 2016]. O primeiro observou 10.000 blocos na cadeia da Bitcoin, com a ocorrência natural de 169 *forks*, ou seja 1,69%. O segundo autor, em resultados de simulação, obteve uma taxa

de forks de 1,51%, demonstrando que os resultados de simulação são bem próximos aos resultados observados em uma aplicação em produção.

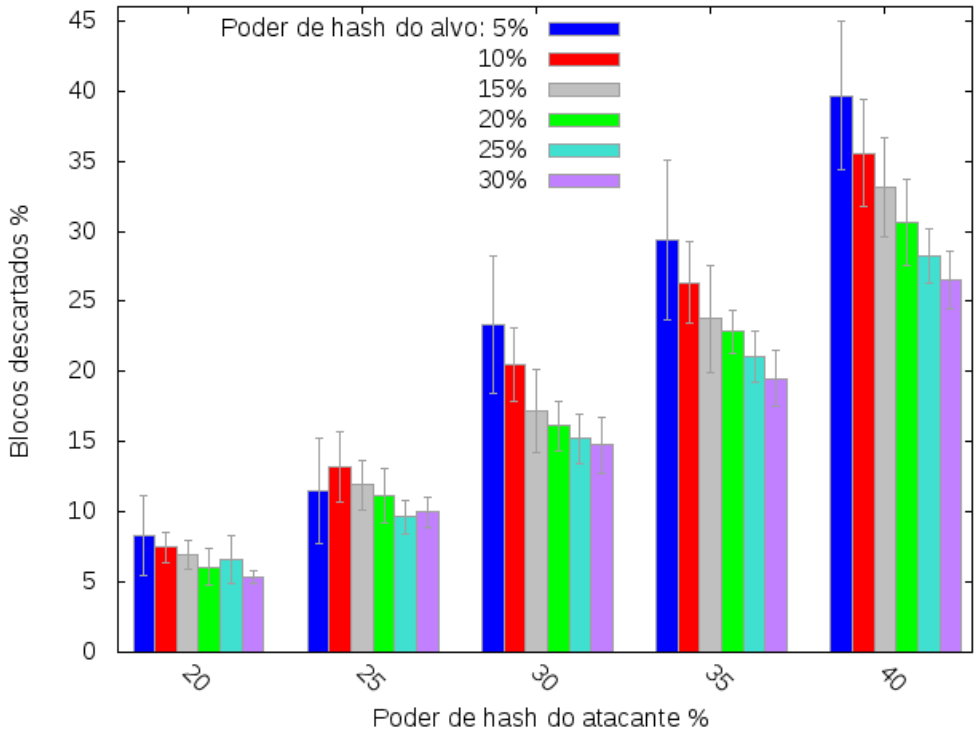

Figura 4. Blocos Descartados do alvo

Na Figura 4 verifica-se a influência do atacante sobre o alvo. O eixo "X" representa o poder computacional do atacante em relação ao restante da rede. O eixo "Y" apresenta a taxa de blocos do alvo que foram descartados. O poder computacional do atacante inicia com 20% e é incrementado de 5% até alcançar o máximo de 40%. Para cada faixa de poder do atacante é também, variado o poder computacional do alvo até 30%. A primeira conclusão obtida é que quanto maior o poder do alvo menor a influência do atacante sobre ele. Esse fato pode ser explicado pelo fato de que o alvo irá publicar mais blocos e o atacante não consegue gerar tantos forks quantos são necessários. A segunda conclusão é que, quanto maior a força do atacante maior é a taxa de blocos descartados. O melhor resultado alcançado pelo atacante ocorre quando ele possui 40% do poder de hash e o alvo possui 5%, quando 39% dos blocos do alvo são descartados.

Existe também um efeito colateral de descarte de alguns blocos dos demais nós da rede que pode ser observado na Figura 5, o poder computacional destes nós é de 50%. Este efeito pode ser explicado pelo fato de que o atacante gera os forks antes de receber um bloco do alvo, desta forma diversos blocos dos outros nós são perdidos como consequência. Estes nós chegam a perder até 16% dos seus blocos, quantidade que está muito acima da taxa natural de forks observada que é de 1,69%.

Sempre que o atacante toma a ação de "adotar" ele joga fora todo o esforço dispendido para gerar o fork. A Figura 6 apresenta a quantidade de blocos jogados fora pelo atacante. É possível ver que quando o alvo tem baixo poder computacional o atacante é pouco eficiente, chegando a descartar 500 blocos, 33% do total de blocos produzidos pela

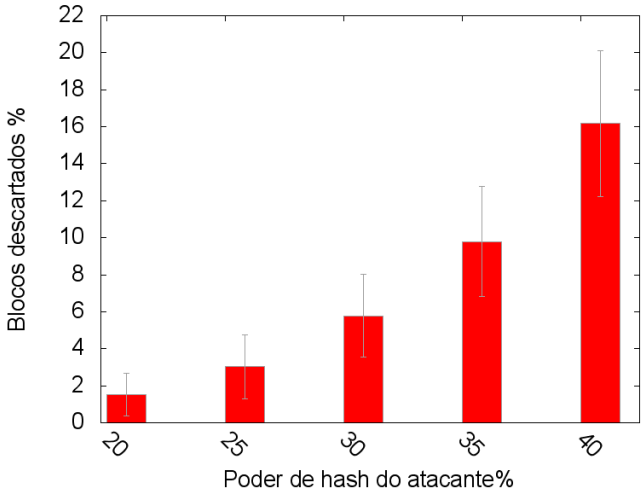

Figura 5. Blocos Descartados dos outros nós

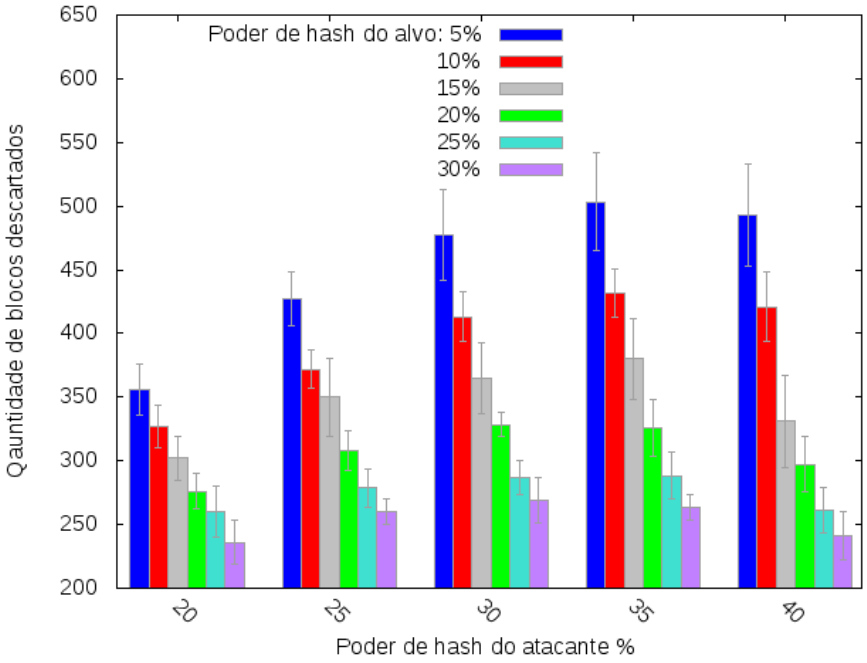

Figura 6. Blocos Descartados do atacante

rede. Analisando o cenário onde o atacante possui 20% do poder de hash e o alvo 5% vemos que foi gerados, em média, 370 blocos pelo atacante, e 340 foram descartados, o que nos dá uma taxa de 92% de blocos desperdiçados. A Figura 7 mostra as médias dos maiores forks observados pelos nós da rede, que aliada a taxa de blocos descartados pode ser uma boa métrica para identificar a presença de um nó malicioso.

Em sistemas que utilizam a Prova de Trabalho o atacante deverá possuir enorme poder computacional e certamente arcará com grandes gastos financeiros. Mas as recompensas futuras podem ser maiores. Imagine que duas empresas decidam adquirir um grande ativo de uma terceira, e haja um *Smart-Contract* com uma cláusula de prioridade

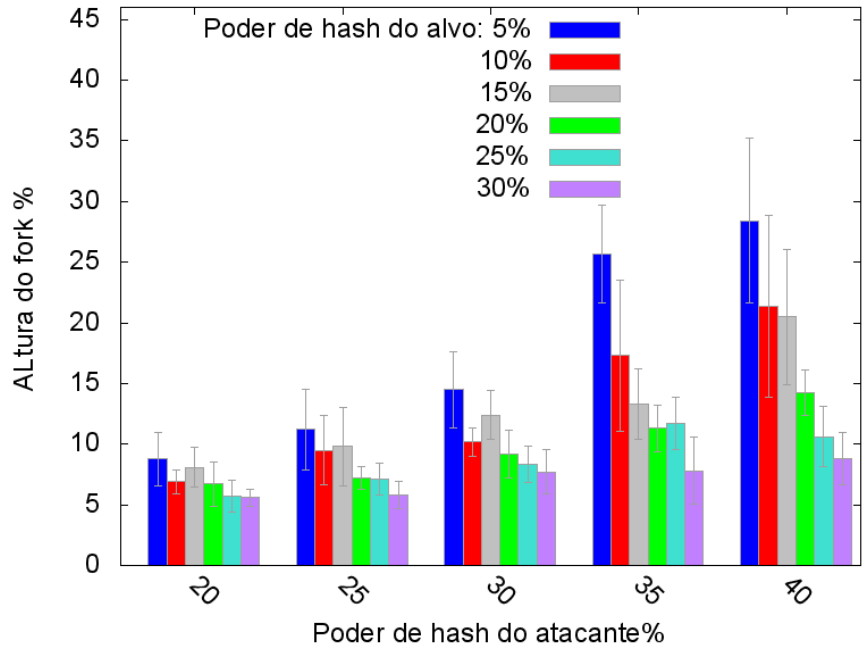

Figura 7. Altura dos forks

para uma das empresas que diga que a segunda somente poderá adquirir o ativo caso a primeira não realize o pagamento em criptomoeda até uma data específica. A segunda empresa pode então realizar o ataque de modo a impedir que a transação realizada pela primeira empresa seja confirmada pela rede. Ali [Ali et al., 2016] anunciou o primeiro ataque, conhecido, de Minerador Egoísta a uma rede em produção, provando que o ataque é factível, apesar das motivações ou de ser muito dispendioso.

Outra consideração importante a ser feita se refere a identificação do usuário ou da transação. Ao realizar as transações, os nós não fornecem seus endereços IP e nem suas identidades, apenas as chaves criptográficas que provam a posse dos recursos. Por esta razão, é possível dizer que a Blockchain fornece um certo grau de privacidade aos usuários. Descobrir quais nós estão realizando transações específicas é muito difícil, mas não impossível. É possível inferir o endereço IP e suas identidades através de várias técnicas [Spagnuolo et al., 2014, Miller et al., 2015, Herrera-Joancomartí, 2015, Moser et al., 2013, Koshy et al., 2014, Valenta and Rowan, 2015]. Desta forma, parte-se do pressuposto que o usuário já foi identificado.

5. Considerações Finais

A "cadeia de blocos" possibilita não meramente o movimento do dinheiro, mas também pode ser usada para transferir informações e servir como meio de autorização e autenticação entre os dispositivos, habilitando o uso da Blockchain como um serviço [Swanson, 2015]. Essas vantagens, juntamente com as possíveis exposições em relação a ações criminosas, levantam cada vez mais interesse e dúvidas em relação às diversas aplicações da Blockchain.

Neste trabalho foi apresentado uma estratégia do ataque do minerador egoísta, chamado de *Stalker mining*, cujo objetivo é impedir que um determinado nó publique

um bloco, ou que uma determinada transação, deste nó seja incluída na cadeia. Foram realizadas análises utilizando o simulador NS3 e os resultados alcançados demonstraram que a eficácia deste ataque é proporcional a razão de poder de hash entre o atacante e o alvo. Além disso, foi verificado um efeito colateral deste ataque, onde com o aumento do poder do atacante, observa-se a perda de blocos dos outros nós da rede. Foi observado que a centralidade não influenciou na eficácia do ataque.

Pesquisa futuras são necessárias para melhorar as heurísticas do ataque de forma a diminuir o impacto sobre outros nós da rede. Além disso, para assegurar a publicação de transações são necessárias técnicas que permitam inferir o comportamento malicioso do atacante.

Referências

Ali, M., Nelson, J. C., Shea, R., and Freedman, M. J. (2016). Blockstack: A global naming and storage system secured by blockchains. In *USENIX Annual Technical Conference*, pages 181–194.

Back, A. et al. (2002). Hashcash-a denial of service counter-measure.

Cachin, C. (2016). Architecture of the hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*.

Carneiro, G. (2010). Ns-3: Network simulator 3. In *UTM Lab Meeting April*, volume 20.

Chicarino, V. R., Jesus, E. F., Albuquerque, C. V. N., and Aragão Rocha, A. A. (2017). Uso de blockchain para privacidade e segurança em internet das coisas. *Livro de Minicursos do VII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais. Brasília: SBC,*.

Decker, C. and Wattenhofer, R. (2013). Information propagation in the bitcoin network. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pages 1–10. IEEE.

Dorri, A., Kanhere, S. S., Jurdak, R., and Gauravaram, P. (2017). Blockchain for iot security and privacy: The case study of a smart home.

Eyal, I. and Sirer, E. G. (2014). Majority is not enough: Bitcoin mining is vulnerable. In *International Conference on Financial Cryptography and Data Security*, pages 436–454. Springer.

Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., and Capkun, S. (2016). On the Security and Performance of Proof of Work Blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16*.

Heilman, E., Kendler, A., Zohar, A., and Goldberg, S. (2015). Eclipse attacks on bitcoin's peer-to-peer network. In *USENIX Security*, pages 129–144.

Herrera-Joancomartí, J. (2015). Research and challenges on bitcoin anonymity. In *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance*, pages 3–16. Springer.

Koshy, P., Koshy, D., and McDaniel, P. (2014). An analysis of anonymity in bitcoin using p2p network traffic. In *International Conference on Financial Cryptography and Data Security*, pages 469–485. Springer.

Kroll, J. A., Davey, I. C., and Felten, E. W. (2013). The economics of bitcoin mining, or bitcoin in the presence of adversaries. In *Proceedings of WEIS*, volume 2013.

Miller, A., Litton, J., Pachulski, A., Gupta, N., Levin, D., Spring, N., and Bhattacharjee, B. (2015). Discovering bitcoin's public topology and influential nodes.(2015).

Moser, M., Bohme, R., and Breuker, D. (2013). An inquiry into money laundering tools in the bitcoin ecosystem. In *eCrime Researchers Summit (eCRS), 2013*, pages 1–14. IEEE.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf.

Nayak, K., Kumar, S., Miller, A., and Shi, E. (2016). Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 305–320. IEEE.

Nguyen, K. T., Laurent, M., and Oualha, N. (2015). Survey on secure communication protocols for the internet of things. *Ad Hoc Networks*, 32:17–31.

Pilkington, M. (2016). Blockchain technology: principles and applications. research handbook on digital transformations, edited by f. xavier olleros and majlinda zhegu.

Spagnuolo, M., Maggi, F., and Zanero, S. (2014). Bitiodine: Extracting intelligence from the bitcoin network. In *International Conference on Financial Cryptography and Data Security*, pages 457–468. Springer.

Swan, M. (2015). *Blockchain: Blueprint for a new economy*. "O'Reilly Media, Inc.".

Swanson, T. (2015). Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems. *Report, available online, Apr*.

Tapscott, D. and Tapscott, A. (2016). *Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world*. Penguin.

Valenta, L. and Rowan, B. (2015). Blindcoin: Blinded, accountable mixes for bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 112–126. Springer.

Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151.

Wörner, D. and von Bomhard, T. (2014). When your sensor earns money. *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp '14 Adjunct*.

Zyskind, G., Nathan, O., and Pentland, A. (2015). Enigma: Decentralized Computation Platform with Guaranteed Privacy. *arXiv:1506.03471 [cs]*.