Segurança e Desempenho de Protocolos de Consenso Baseados em Prova para Corrente de Blocos

Gabriel Antonio F. Rebello, Gustavo F. Camilo, Lucas C. B. Guimarães, Lucas Airam C. de Souza, Otto Carlos M. B. Duarte

Universidade Federal do Rio de Janeiro - GTA/COPPE/UFRJ

Resumo. *A corrente de blocos é uma tecnologia disruptiva que revolucionará a Internet e nosso modo de viver, trabalhar e negociar. Apesar da inovação, os atuais sistemas públicos baseados em corrente de blocos, como Bitcoin e Ethereum, apresentam importantes limitações de segurança, desempenho e consumo excessivo de energia devido a seus protocolos de consenso baseados em prova de trabalho. Este artigo apresenta e compara as principais alternativas à prova de trabalho, com foco na segurança e desempenho de cada protocolo de consenso baseado em prova. Os protocolos baseados em prova utilizam o modelo de consenso probabilístico e são mais adequados para ambientes públicos com muitos participantes, como em um cenário de Internet das Coisas. O artigo destaca a tendência de centralização provocada pela remuneração da tentativa de prova, e o principais ataques dos consensos de prova de trabalho e prova de posse, assim como suas contramedidas.*

1. Introdução

O problema do consenso em sistemas distribuídos com redes assíncronas é um problema conhecido que pesquisadores estudam há mais de 40 anos. No entanto, em 2008, Satoshi Nakamoto[1] revolucionou a área de consenso distribuído ao propor um novo modelo de consenso baseado em prova de trabalho (*Proof of Work* - PoW) [Nakamoto 2008]. Neste novo modelo de consenso, um participante que propõe um bloco, denominado minerador[2], deve apresentar uma prova de que pode liderar o consenso gastando recursos computacionais para resolver independentemente um desafio matemático computacionalmente custoso. O vencedor de o desafio é bem remunerado para incentivar uma competição ampla. Existe a possibilidade do desafio ser vencido por mais de um minerador ao mesmo tempo, criando uma bifurcação e, consequentemente, estados inconsistentes na corrente de blocos. Diferentemente dos protocolos de consenso estudados até então, a prova de trabalho introduz o conceito de consenso probabilístico, pois a solução probabilística dada é de manter o maior ramo da bifurcação que corresponde ao maior número de desafios vencidos e o maior gasto com a solução. A prova de trabalho dispensa a troca de mensagens e a identificação dos participantes para obter consenso, o que provê descentralização, anonimidade e escalabilidade em um nível jamais visto em sistemas distribuídos. Na proposta de Nakamoto, qualquer pessoa ou organização pode se tornar um minerador de forma anônima, e milhares de participantes podem participar do consenso simultaneamente utilizando a Internet como sistema de comunicação. Devido às suas características, pesquisadores criam sistemas que

Este trabalho foi realizado com recursos do CNPq, CAPES, FAPERJ e FAPESP (18/23292-0, 2015/24514-9, 2015/24485-9 2014/50937-1).

[1]Satoshi Nakamoto é um pseudônimo utilizado pelo criador ou criadores da moeda virtual Bitcoin. Sua identidade real é desconhecida.

[2]O nome minerador vem da dificuldade e do enorme trabalho necessário para vencer o desafio.

utilizam a corrente de blocos para prover segurança em diversas aplicações distribuídas [Palma et al. 2019, Pinno et al. 2017, Rebello et al. 2019a].

A prova de trabalho, no entanto, ainda possui desempenho baixo em relação ao desempenho de aplicações centralizadas e possui alto gasto energético. Em resposta às limitações de desempenho da prova de trabalho, diversas alternativas apresentam novos protocolos baseados em prova para substituir o protocolo do Bitcoin. O não-determinismo dos protocolos baseados em prova, sejam a prova de trabalho ou protocolos alternativos, é a principal fonte de vulnerabilidades. A natureza probabilística do consenso permite que um agente malicioso explore as bifurcações do sistema para realizar ataques de gasto duplo contra comerciantes e/ou corretoras. O atacante também pode aproveitar dos sistemas utilizarem redes par-a-par públicas, que operam sobre a Internet, para realizar ataques de rede ou aos participantes do consenso.

Este artigo apresenta e classifica os principais protocolos de consenso baseado em prova, expondo os ataques e as vulnerabilidades de segurança de cada protocolo. Os protocolos baseados em prova utilizam, assim como o Bitcoin, o modelo de consenso probabilístico que funciona em sistemas de comunicação assíncronos como a Internet e atende aplicações públicas nas quais qualquer usuário pode participar do consenso. O artigo foca sobretudo a prova de trabalho e na prova de posse (*Proof of Stake* - PoS), os protocolos alternativos baseados em prova mais populares em criptomoedas e plataformas de correntes de blocos públicas. O artigo também compara as principais criptomoedas em valor de mercado e plataformas que utilizam protocolos probabilísticos, como: Bitcoin, Ethereum, Cardano, EOSIO e Hyperledger Sawtooth.

O restante do artigo está organizado da seguinte forma. A Seção 2 introduz conceitos de sistemas distribuídos e a classificação de consensos determinísticos e probabilísticos. A Seção 3 detalha o consenso da prova de trabalho e analisa possíveis ataques a este mecanismo. A Seção 4 descreve a prova de posse, principal consenso alternativo à prova de trabalho, expondo os principais desafios para a garantia de segurança. A Seção 5 apresenta e analisa protocolos alternativos de consenso baseados em prova. A Seção 6 apresenta trabalhos relacionados a este artigo. A Seção 7 conclui o artigo, comparando as vulnerabilidades de segurança apresentadas pelos mecanismos de consenso apresentados.

2. O Consenso Probabilístico

O protocolo de consenso[3] é o algoritmo distribuído que garante que o sistema evolui, adicionando novos blocos corretamente. A Figura 1 exibe a estrutura da corrente de blocos. Para obter consenso, no entanto, o protocolo deve lidar com possíveis falhas de rede ou de participantes. Um protocolo de consenso pode tolerar falhas de parada (*crash faults*) ou falhas bizantinas (*byzantine faults*). Um participante que está em falha de parada não responde e não executa novas operações durante uma rodada do consenso. No modelo de falha bizantina, o participante em falha pode ser um agente malicioso que exibe um comportamento arbitrário que desvia do protocolo e executa qualquer ação. O agente malicioso pode emitir respostas corretas, incorretas ou contraditórias, além de não emitir respostas. O modelo de falhas bizantinas é o que melhor capta o comportamento de participantes em correntes de blocos públicas, como as do Bitcoin e Ethereum, nos quais

[3]O consenso é o processo pelo qual um grupo de participantes independentes atinge a mesma decisão coletiva de aceitar ou recusar um novo bloco a ser incorporado na corrente de blocos.

os usuários do sistema podem participar do consenso de forma anônima, sem necessidade de permissão, e agir de forma maliciosa.

O objetivo principal dos protocolos de consenso é prover as propriedades de consistência (*safety*) e vivacidade (*liveness*) ao sistema. O protocolo garante a vivacidade quando há a certeza que as rodadas do consenso sempre finalizam e, consequentemente, o sistema sempre adiciona novos blocos. A consistência garante que os blocos adicionados são idênticos em todos os participantes honestos que não estão em estado de falha e que o bloco foi proposto por um participante honesto ao início da rodada de consenso. Garantir que o sistema funciona corretamente e com tolerância a falhas envolve construir um protocolo de consenso que proveja ambas as propriedades.

Um dos principais desafios de consenso em sistemas distribuídos é o resultado de impossibilidade de garantia do consenso, conhecido como resultado FLP em homenagem aos autores Fischer, Lynch e Patterson que formularam o teorema. O resultado prova que o consenso não possui solução determinística mesmo na presença de uma única falha de parada em um sistema que opera sobre uma rede assíncrona como a Internet [Fischer et al. 1985]. Durante décadas, as diversas propostas de consenso contornaram o resultado FLP assumindo sistemas de comunicação síncronos, parcialmente síncronos ou eventualmente síncronos, que proveem diferentes níveis de garantias de entrega das mensagens enviadas durante o consenso. Assim, protocolos de consenso focavam na propriedade de consistência e confiavam que o sistema de comunicação proveria a propriedade de vivacidade ao garantir a entrega das mensagens. Estes protocolos dependentes da sincronia da rede, no entanto, não atendem o comportamento de redes de melhor esforço como a Internet, onde não há garantia de entrega de mensagens.

Desde Nakamoto, existem duas alternativas para contornar o resultado FLP: garantir consistência, como faziam os protocolos anteriores, ou garantir vivacidade. Assim, surgem duas famílias de protocolos para corrente de blocos. Os protocolos inspirados no consenso determinístico clássico, como PBFT [Castro and Liskov 2002], BFT-SMaRt [Bessani et al. 2014], Tendermint [Buchman 2016] e Ripple [Schwartz et al. 2014], privilegiam a consistência em detrimento da vivacidade, criando protocolos que não possuem bifurcações, mas que podem travar caso o sistema de comunicação se comporte de forma assíncrona. Os protocolos baseados em consenso probabilístico, como prova de trabalho e prova de posse, privilegiam a vivacidade em detrimento da consistência, garantindo seu funcionamento na Internet, mas sendo suscetíveis a bifurcações (*forks*) na corrente de blocos. Como consequência das duas abordagens, o problema do consenso pode ser tratado, respectivamente, de maneira determinística ou probabilística.

Os protocolos de consenso mais comuns em sistemas públicos são os baseados em algoritmos de prova, nos quais um proponente de um bloco deve apresentar uma prova de que pode liderar o consenso [Nakamoto 2008, Olson et al. 2018]. Os algoritmos baseados em prova proveem consenso probabilístico e seguem modelos similares à prova de trabalho de Nakamoto. Consensos probabilísticos possuem como principal vantagem a escalabilidade, uma vez que não é necessário conhecer todos os participantes da rede para se obter consenso. Portanto, este tipo de consenso é mais adequado a correntes de blocos públicas com muitos participantes. Em consensos probabilísticos, dois ou mais participantes podem propor blocos corretos simultaneamente, causando uma bifurcação na corrente de blocos. Selecionar o ramo da bifurcação que possui a cadeia mais longa é

a regra de desempate no Bitcoin e a mais conhecida em correntes de blocos.

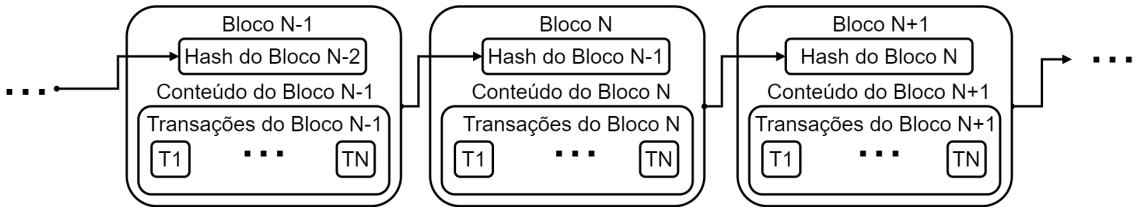

Figura 1. Estrutura simplificada de uma corrente de blocos.

3. A Prova de Trabalho (*Proof of Work* - PoW)

A prova de trabalho (*Proof of Work* - PoW) é um mecanismo de consenso proposto por Satoshi Nakamoto, utilizado nas duas principais criptomoedas em valor de mercado: Bitcoin [Nakamoto 2008] e Ethereum. O objetivo da prova de trabalho é determinar o participante responsável por propor a adição de um bloco à corrente. Para isso, o mecanismo de consenso implementa um desafio criptográfico computacionalmente custoso que deve ser resolvido por um participante do consenso antes de divulgar um bloco na rede. A principal vantagem da prova de trabalho é sua alta escalabilidade, pois qualquer um pode participar e minerar blocos, sendo um consenso adequado para redes públicas.

O desafio criptográfico da prova de trabalho envolve encontrar um número (*nonce*), de modo que uma função resumo (*hash*) aplicada ao bloco resulte em um número menor que um número pré-determinado, o que corresponde a um número de zeros nos bits iniciais. Após resolver o desafio, o participante divulga o bloco com o número encontrado. Outros membros da rede podem facilmente verificar se o desafio foi resolvido de forma correta calculando o resumo do bloco e verificando o número de zeros obtido. A quantidade mínima de zeros nos bits iniciais define a dificuldade do desafio e é ajustada de maneira periódica para estabelecer uma taxa constante de criação de blocos. O PoW recompensa o minerador de um bloco para incentivar os participantes a gastarem poder computacional e resolverem o desafio.

As principais desvantagens da prova de trabalho são o alto consumo energético e a baixa vazão de transações. O alto custo computacional envolvido no cálculo da prova de trabalho no Bitcoin consome anualmente mais de quatro vezes a energia gerada pelas usinas nucleares de Angra [Eletrobras 2017]. Além disso, a adição de novos blocos à corrente no Bitcoin é lenta, com a criptomoeda apresentando vazão média de 7 transações por segundo. Esse valor é consideravelmente inferior à vazão média de 2000 transações por segundo registrada por empresas de cartão de crédito, impedindo a utilização de criptomoedas baseadas em PoW para compras do dia a dia.

3.1. Análise de Segurança da Prova de Trabalho

Mesmo sendo utilizado por criptomoedas com alto valor de mercado, o protocolo apresenta vulnerabilidades descritas em múltiplos artigos. As vulnerabilidades podem ser categorizadas como ataques de gasto duplo, ataques ao consenso ou ataques à rede.

Ataques de gasto duplo (*double-spending attacks*) objetivam utilizar a mesma moeda em múltiplas transações. Ao contrário da moeda física, a moeda digital pode ser facilmente replicada e com isto existe o risco do uso da mesma moeda mais de uma vez. O Bitcoin propõe a estrutura de corrente de blocos que armazena publicamente

todo o histórico de transações de forma distribuída e ordenada para prevenir o gasto duplo [Nakamoto 2008]. No entanto, ataques de gasto duplo ainda são possíveis na rede Bitcoin [Karame et al. 2012]. Para efetuar esse ataque, um atacante A envia uma transação T_A^V para um vendedor V e uma transação T_A^A para uma conta controlada pelo atacante, de modo que a diferença de tempo entre as duas transações $\Delta t \approx 0$. Neste cenário, uma parte da rede confirma a transação T_A^V e o vendedor V envia o produto adquirido pelo atacante. Enquanto isso, o atacante divulga a transação T_A^A com a ajuda de múltiplas contas para outra parte da rede, que confirma T_A^A. Se um minerador adicionar a transação T_A^A em um bloco antes da transação T_A^V, o vendedor perde o produto enquanto o atacante mantém seu dinheiro, finalizando o ataque de gasto duplo.

Uma outra forma de efetuar o gasto duplo é através do ataque Finney, descrito por Hal Finney em um fórum do Bitcoin em 2011 [Finney 2011]. Neste ataque, o atacante A é um minerador que emite uma transação T_A^A em um tempo $t_{T_A^A}$ para uma conta controlada por ele, e minera um bloco B_A contendo essa transação. O atacante então guarda o bloco minerado para si e envia uma transação T_A^V para um vendedor V em um tempo $t_{T_A^V}$. Como o bloco B_A não foi divulgado e a transação T_A^A não foi validada, V aceita a transação T_A^V e envia o produto ao atacante. Após receber o produto, A divulga o bloco B_A contendo a transação T_A^A. Assim, como $t_{T_A^V} > t_{T_A^A}$, os participantes da rede descartam a transação T_A^V e V perde o produto sem ser remunerado.

O ataque de 51% consiste em um atacante ou grupo de atacantes possuir mais de 50% do poder computacional da rede, porque, neste caso, os atacantes podem efetuar um gasto duplo. Embora um ataque de 51% nunca tenha sido executado com sucesso no Bitcoin , as quatro maiores *pools* da rede Bitcoin já somam mais de 50% do poder computacional[4]. Um conluio entre apenas quatro entidades independentes seria capaz de subverter o sistema por completo. Assim, contrariando a proposta inicial de descentralização do Bitcoin, quatro agentes centralizam o poder da rede. Além disso, esse tipo de ataque ocorreu em protocolos baseados em prova alternativos ao Bitcoin[5,6].

A mineração egoísta [Eyal and Sirer 2018] é um ataque que explora o algoritmo de convergência ou resolução de bifurcações. Um atacante com poder de mineração menor do que 51% da rede pode adotar a estratégia de mineração egoísta para obter vantagens na remuneração ou para realizar gasto duplo. Para isso, o nó malicioso minera e mantém em sigilo novos blocos, criando uma corrente de blocos particular. Eventualmente o atacante compartilha seus blocos para criar bifurcações, dividindo o poder computacional dos mineradores. Ao criar uma bifurcação maior do que a dos mineradores honestos, o participante malicioso faz a rede convergir para o seu estado. Dessa forma, o atacante consegue realizar ataques de gasto duplo com sucesso se possuir 25% do poder computacional total da rede. Mineradores que possuam versões antigas ou bifurcações abandonadas da corrente de blocos desperdiçam recursos computacionais na tentativa de encontrar um novo bloco. Além disso, os nós esquecem todas as transações existentes na bifurcação abandonada caso estas não existam nos blocos do atacante, permitindo o gasto duplo.

O ataque de descarte de blocos (*block discarding*) [Bahack 2013] é uma extensão do ataque de mineração egoísta. Nesse ataque, o atacante controla um conjunto de nós

[4]Disponível em: https://btc.com/stats/pool. Acessado em 6 de agosto de 2020.

[5]A moeda Bitcoin Gold, na época a 26^a maior moeda, sofreu um ataque de 51% em maio de 2018. Os atacantes efetuaram gasto duplo por diversos dias e roubaram mais de US$18 milhões em Bitcoin Gold.

[6]As correntes de blocos Krypton e Shift sofreram ataques de 51% em agosto de 2016.

da rede responsáveis por descartar novos blocos descobertos conforme eles são recebidos. Esses nós somente divulgarm os blocos obtidos pelo atacante, tornando a mineração egoísta mais efetiva ao atrasar a propagação de blocos descobertos por outros nós da rede.

O ataque de suborno (*bribery*) ocorre quando um atacante sem poder computacional suficiente para atacar a rede suborna mineradores com maior capacidade de processamento para formar um conluio durante um determinado período [Bonneau et al. 2016]. Entretanto, a rede perde confiança caso o nó malicioso consiga utilizar essa estratégia para realizar outros ataques como o gasto duplo, desvalorizando assim a moeda. Desse modo, nós mineradores que são investidores da moeda, pois possuem ativos graças ao incentivo obtido pela descoberta de novos blocos, perdem dinheiro investido ou tem seu lucro reduzido. Portanto, para subornar mineradores, o atacante deve gastar uma quantia que supere as perdas, tornando a estratégia cara e inviável em redes com alto poder computacional.

Os ataques de rede representam uma grande ameaça para a prova de trabalho, pois a comunicação em ambientes de correntes de blocos é distribuída e o protocolo permite inconsistências temporárias. Caso o atacante tenha sucesso, as vítimas de ataques de rede podem permanecer em estados incorretos durante longos períodos por falta de informações sobre o estado global da rede.

A prova de trabalho mitiga o uso de ataques Sybil [Douceur 2002], comum em redes P2P como as utilizadas em correntes de blocos, para manipular o consenso. Como a adição de blocos à corrente depende da solução de um desafio criptográfico computacionalmente custoso, criar novas identidades não aumenta a probabilidade de um atacante solucionar o problema, dado que ele terá de dividir o processamento entre as identidades criadas. Entretanto, devido a comunicação distribuída, um atacante pode criar diversas identidades para controlar informações entregues e enviadas por determinados nós. Assim, o ataque de Sybil pode ser aplicado à fases intermediárias de ataques mais complexos, como a mineração egoísta, gasto duplo e eclipse. Este último citado em seguida.

Outra forma de controlar as informações de parte da rede é realizando o ataque de eclipse [Heilman et al. 2015]. Para isso, o nó malicioso cria diversas identidades e força sua vítima a adicionar as contas controladas pelo atacante à lista de nós conhecidos. Dessa forma, caso a vítima conheça apenas os nós controlados pelo atacante, o participante malicioso passa a controlar as informações e pode criar uma visão local diferente do estado atual da corrente de blocos para o nó atacado.

Devido à descentralização da corrente de blocos, causar indisponibilidade na rede requer um enorme poder computacional, além do conhecimento de um grande número de participantes. Entretanto, como alguns pontos da rede apresentam maior centralização, a negação de serviço distribuída pode afetar nós que têm maior importância, como gerentes de grupos (*pools*) de mineração [Johnson et al. 2014].

4. O Consenso por Prova de Posse

A Prova de Posse (*Proof of Stake* - PoS) é o consenso alternativo mais conhecido à Prova de trabalho, por proverem características similares sem necessitar de alto gasto energético. As principais vantagens da prova de posse em comparação à prova de trabalho incluem a alta eficiência energética, alto desempenho e maior segurança.

A prova de posse é uma categoria de algoritmos baseados em prova para correntes de blocos públicas cuja principal característica é realizar o consenso baseando-se nas

quantidades de recursos em posse de cada participante [Rebello et al. 2019b]. Comparado à prova de trabalho, na qual a probabilidade de um participante propor um bloco é proporcional somente a seu poder computacional (*hashpower*), na prova de posse a probabilidade de propor um bloco é proporcional à quantidade de ativos que o participante aposta (*stake*) no momento do consenso. Devido à ausência do ato de "minerar", i.e. gastar poder computacional para obter recompensas, os protocolos PoS introduzem o conceito chamado de "mineração virtual" (*virtual mining*) e definem seus participantes como validadores ou partes interessadas (*stakeholder*) em vez de mineradores [Xiao et al. 2020b, Wang et al. 2018]. Na mineração virtual baseada em prova de posse, qualquer participante que possuir ativos pode se tornar um validador ao disponibilizar seus ativos como depósito. Então, ocorre uma rodada de consenso no qual o poder de cada participante é proporcional aos seus respectivos depósitos em relação ao total.

A implementação de um consenso baseado em prova de posse pode seguir duas abordagens principais: i) uma prova de posse probabilística na qual um participante com mais posses possui maior chance de propor um bloco; ou ii) uma prova de posse determinística baseada em acordo bizantino (*BFT-based PoS*), no qual um conjunto de validadores confirma todos os blocos propostos através de votação com pesos proporcionais à posse de cada validador [Buterin 2019, Wang et al. 2018, Xiao et al. 2020b]. O critério para selecionar o proponente pode ser um sorteio baseado nas posses, como na criptomoeada Ouroboros [Kiayias et al. 2017], ou uma eleição, como na criptomoeda EOSIO [Larimer 2017]. Além das duas abordagens, cada protocolo de consenso apresenta detalhes específicos, como a forma de incentivar os validadores e os mecanismos para prevenir ataques, o que gera diversas maneiras práticas de se implementar a prova de posse. Em vez de analisar protocolos em específico, este artigo foca a abordagem probabilística para prover uma análise de segurança geral da prova de posse.

A prova de posse probabilística herda características similares à prova de trabalho de Nakamoto [Nakamoto 2008], como a seleção pseudo-aleatória de um participante para adicionar um bloco, a regra da maior cadeia e a finalidade probabilística. Os desenvolvedores do Bitcoin propõem a partir de 2011 a primeira família de consensos de prova de posse probabilística, que hoje é conhecida como Nakamoto-PoS ou *chain-based PoS*. Nesta implementação, assim como na prova de trabalho de Nakamoto, cada participante deve calcular um *hash* criptográfico para atingir um alvo, mas limitado a uma janela de tempo e cuja dificuldade diminui de acordo com a posse do participante. Ainda que o processo de validação seja similar ao procedimento da prova de trabalho, a dificuldade média de atingir o alvo do desafio computacional é significativamente menor que a do Bitcoin. Portanto, o PoS evita a competição baseada somente em força bruta característica da prova de trabalho e, consequentemente, reduz os gastos energéticos.

Propostas mais recentes, como o Ouroboros, selecionam pseudo-aleatoriamente validadores que podem propor blocos em um intervalo de tempo. Esses protocolos, conhecidos como prova de posse baseada em comitê (*commitee-based PoS*), utilizam técnicas de computação de múltiplas partes (*multi-party computation* - MPC) para simular um sorteio entre os participantes, dando mais chance a participantes com mais posse investida. O MPC recebe o estado atual da corrente de blocos, que inclui as posses de cada participante e seleciona uma sequência pseudo-aleatória de próximos proponentes que pode ser verificada por qualquer participante. Participantes podem ser escolhidos mais de uma vez e recebem mais tempo para propor blocos caso tenham mais posse.

4.1. Análise de Segurança da Prova de Posse Probabilística

Nas primeiras implementações da prova de posse, é suficiente possuir ativos para participar e obter vantagem no processo de consenso. No entanto, a não-exigência de depósitos permite o ataque de "nada a perder" (*nothing at stake*), no qual os participantes podem utilizar ativos para participar simultaneamente na validação de blocos conflitantes quando uma bifurcação ocorre na rede. Este é o comportamento mais vantajoso, que será seguido por todo validador racional porque não há custo computacional para validar transações em múltiplas bifurcações, ao contrário da prova de trabalho. Portanto, torna-se computacionalmente eficiente o comportamento de validar várias bifurcações simultaneamente o que corresponde a várias chances de se ganhar sem nenhum risco de perda; Assim, o comportamento que maximiza a probabilidade de ganhos é participar de todas as bifurcações possíveis. Consequentemente, todo participante racional que deseja maximizar seu lucro segue este comportamento.

O problema "nada a perder" (*nothing at stake*) pode ser modelado matematicamente como um problema de maximização de probabilidades. Seja uma corrente de blocos bifurcada com dois caminhos conflitantes[7] A e B e um participante genérico que possui uma parcela (*stake*) $s \in [0, 1]$ do total de recursos no sistema. Definem-se os possíveis eventos:

- F_A: o sistema eventualmente finaliza[8] o caminho A e o caminho B é abandonado;
- F_B: o sistema eventualmente finaliza o caminho B e o caminho A é abandonado;
- Val_X: o participante utiliza seus recursos para validar o caminho X;
- R: o participante vence a rodada e recebe as recompensas pré-acordadas.

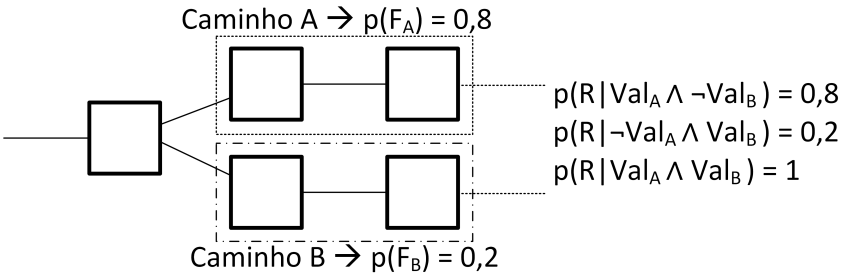

Figura 2. Uma corrente de blocos bifurcada com dois caminhos conflitantes A e B com diferentes probabilidades de serem finalizados pelo sistema. A melhor estratégia para um participante garantir uma recompensa R é validar os dois caminhos, contribuindo para o prolongamento da bifurcação.

Na prova de posse, não há gasto de recursos para validar um dos caminhos possíveis nem mecanismos de punição para evitar a validação de múltiplos caminhos. Assim, ainda que F_A e F_B sejam eventos mutuamente exclusivos, o sistema permite que o participante utilize todos os seus recursos para validar os dois os caminhos, i.e. $Val_A \wedge Val_B$, realizando uma validação dupla (*double stake*) sem punição [Buterin 2019]. Na prova de trabalho, a capacidade computacional do participante seria dividida entre as validações de cada caminho. As probabilidades de que o participante seja recompensado considerando

[7]Caminhos conflitantes são caminhos que partem do mesmo bloco de origem e possuem mesma altura, de forma que não basta simplesmente aplicar a regra da maior corrente de Nakamoto [Nakamoto 2008].

[8]Finalizar um caminho significa considerá-lo como o caminho correto entre os caminhos conflitantes.

cada possível cenário são:

$$p(R|(Val_A \wedge \neg Val_B)) = s.p(F_A), \tag{1}$$

quando o participante valida apenas o caminho A,

$$p(R|(\neg Val_A \wedge Val_B)) = s.p(F_B), \tag{2}$$

quando o participante valida apenas o caminho B, e

$$p(R|(Val_A \wedge Val_B)) = s[p(F_A) + p(F_B)], \tag{3}$$

quando o participante valida ambos os caminhos. Utilizando a propriedade de exclusão mútua entre F_A e F_B, a Equação 3 pode ser simplificada, pois $p(F_A) = 1 - p(F_B)$:

$$p(R|(Val_A \wedge Val_B)) = s[p(F_A) + 1 - p(F_A)] = s. \tag{4}$$

Como $s > s.p(A)$ e $s > s.p(B)$, o valor esperado de validar ambos os caminhos será sempre maior que escolher apenas um dos caminhos. Esta é a decisão que maximiza a probabilidade de ser recompensado em uma rodada de consenso e que, consequentemente, maximiza os ganhos do participante no longo prazo. Este resultado mostra que todo participante racional no sistema valida os dois os caminhos, e consequentemente, a finalidade de um dos caminhos pode não ocorrer mesmo sem a presença de atacantes. Além disso, realizar um ataque de gasto duplo torna-se muito mais fácil, uma vez que a atacante precisa apenas possuir mais recursos do que os participantes altruístas[9]. A Figura 2 ilustra o cenário do problema com os caminhos conflitantes. Na prova de trabalho, este problema não ocorre, pois dividir o poder computacional entre as bifurcações não aumenta a chance de minerar um bloco.

A principal contramedida ao problema "nada a perder" é a punição dos participantes que validarem dois caminhos conflitantes. O Ethereum recompensa financeiramente usuários que descobrirem votos conflitantes de um validador a qualquer momento. O sistema destrói todas as posses de um validador que confirme dois caminhos conflitantes e o impede temporariamente de participar em novas rodadas de validação de blocos.

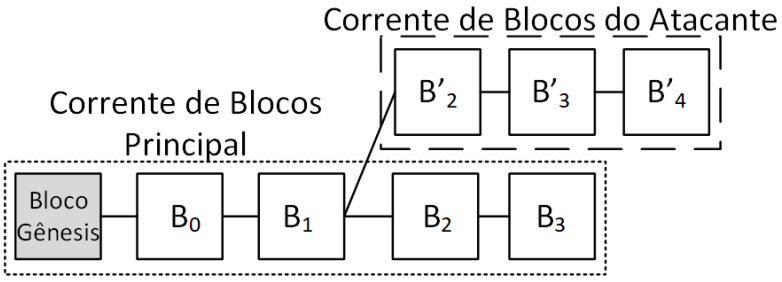

Figura 3. Efetuação de um ataque de longo alcance. O atacante cria uma bifurcação em um bloco aceito pela rede e tenta reescrever a corrente principal.

Outra vulnerabilidade da prova de posse é o ataque de longo alcance (*long-range attack*), que objetiva reescrever blocos antigos e já aceitos pelos participantes da

[9]Participantes altruístas são participantes que preservam o bom funcionamento do sistema, validando apenas um dos caminhos possíveis

rede [Deirmentzoglou et al. 2019]. Para efetuar esse ataque em uma corrente de blocos $B = (b_0, b_1, b_2, ..., b_h)$, o atacante A deve gerar uma bifurcação em uma altura f anterior ao comprimento h atual da corrente. Com isso, A gera uma corrente de blocos $B' = (b'_0, b'_1, b'_2, ..., b'_f, b'_{f+1}, ..., b'_{f_h})$ em que $B = B'$ para blocos $b'_i, i < f$. Na bifurcação gerada, A copia diversas transações da corrente principal para maximizar a recompensa por gerar os blocos. O objetivo do atacante é minerar blocos sem revelar aos outros participantes para alcançar e substituir a corrente de blocos principal. O atacante A precisa controlar parte significativa dos ativos da rede na altura f da bifurcação. Ataques de longo alcance aproveitam o custo baixo ou nulo de criação de blocos na prova de posse para recriar sequências de blocos mais longas do que a corrente de blocos principal, subvertendo facilmente a regra da maior cadeia. Esse ataque não é efetivo em correntes de blocos que utilizam a prova de trabalho porque o custo computacional de reescrever a corrente de blocos desde o início é altíssimo. A Figura 3 ilustra o ataque de longo alcance. Uma das maneiras de mitigar ataques de longo alcance envolve a implementação de ponto de controles (*checkpoints*) que restringem a bifurcação da corrente em uma altura anterior ao ponto definido. Essa contramedida limita o alcance do ataque ao impedir que atacantes gerem bifurcações em pontos muito distantes da corrente de blocos principal.

5. Alternativas Baseadas em Prova (*Proof-of-X* – PoX)

Os algoritmos baseados em prova alternativos a prova de trabalho procuram mitigar as limitações de desempenho e o excesso de gasto energético presentes na prova de trabalho, além do problema do "nada a perder" e do ataque de longo alcance da prova de posse [Kiayias et al. 2017]. A Tabela 1 apresenta uma comparação de desempenho e escalabilidade entre os principais protocolos alternativos e a prova de trabalho. A seguir está uma explicação resumida dos protocolos mais conhecidos.

Delegated Proof of Stake (DPoS)[10]**.** Os participantes utilizam seus ativos para eleger delegados em um quórum que define o bloco a ser adicionado. A quantidade de votos de um minerador é proporcional aos seus ativos [Kiayias et al. 2017, Larimer 2017]. A centralização nos delegados traz como vantagem o aumento da eficiência. No entanto, a centralização do modelo DPoS apresenta vulnerabilidades claras, como: (i) um conluio entre poucos usuários com grandes posses é suficiente para eleger delegados maliciosos; (ii) a eleição de apenas poucos delegados maliciosos já permite ataques de gasto duplo; e (iii) após eleitos, os delegados possuem o mesmo poder independente da quantidade de votos recebidos. O fato dos delegados não necessitarem da mesma quantidade de votos recebidos facilita o conluio, pois os atacantes precisam apostar apenas nos delegados menos votados, que corresponde a um pequeno conjunto de posse.

Proof of Authority (PoA)[11]**.** Similar ao DPoS, porém o conjunto de delegados (autoridades) é pré-determinado em acordo e suas identidades são públicas e verificáveis por qualquer participante da rede [Angelis et al. 2018]. A principal vantagem é a fácil fiscalização das autoridades e a principal desvantagem é a centralização em autoridades sem possibilidade de eleição. Ekparinya *et al.* desenvolveram o Ataque de Clonagem, em que um delegado malicioso clona sua chave privada e passa a agir em duas instâncias de uma corrente de blocos [Ekparinya et al. 2019]. Em uma rede com n ímpar delegados ele comunica a apenas $\frac{(n-1)}{2}$ delegados uma transação, de forma que ambos os grupos, cientes

[10] A plataforma EOSIO utiliza o DPoS como protocolo de consenso.

[11] As criptomoedas VeChain Thor e POA utilizam a prova de autoridade como protocolo de consenso.

Tabela 1. Comparação dos protocolos de consenso em corrente de blocos.

Plataforma/Protocolo	Tipo de Consenso	Vazão máxima	Nº de validadores
Bitcoin	Proof of Work (PoW)	≈ 7 tx/s	Milhares
Ethereum/Ethash	Proof of Work (PoW)	≈ 15 tx/s	Milhares
Cardano/Ouroboros	Proof of Stake (PoS)	≈ 250 tx/s	Centenas
EOSIO	Delegated Proof of Stake (DPoS)	≈ 4000 tx/s	Dezenas
VeChain Thor, POA	Proof of Authority (PoA)	≈ 165 tx/s	Milhares
Hyperledger Sawtooth	Proof of Elapsed Time (PoET)	≈ 1150 tx/s	Centenas

ou não da transação, acreditam ser a maioria de $\frac{(n-1)}{2} + 1$. Para realizar o gasto duplo ele explora a topologia da rede conectando os delegados, de forma que a ramificação com a transação fique atrasada tempo o suficiente para que a outra passe a ser a maior.

Proof of Elapsed Time (PoET)[12]**.** Cada participante possui um temporizador aleatório decrescente e o nó cujo temporizador terminar primeiro propõe o próximo bloco [Olson et al. 2018]. O protocolo de consenso funciona exclusivamente em *hardware* que suporta a tecnologia *Intel Software Guard eXtensions* (SGX). O Intel SGX garante, através de regiões privadas de memória, a distribuição aleatória de temporizadores e que nenhuma entidade tem acesso a mais de um participante do consenso. A principal vantagem é prover consenso seguro e eficiente sem grandes custos de processamento e a principal desvantagem é a dependência de hardware específico. A segurança do protocolo PoET depende da segurança do SGX e dos enclaves de *hardware* da Intel, que já foram explorados por atacantes no passado. Chen *et al.* demonstram que, se a tecnologia puder ser comprometida, a segurança do protocolo é inversamente proporcional ao número de participantes, o que prejudica sua escalabilidade [Chen et al. 2017]. Os autores provam que é necessário comprometer apenas $\Theta(\frac{log(log(n))}{log(n)})$ dos participantes para subverter o consenso, o que corresponde a cerca de 30% para 1000 participantes.

6. Trabalhos Relacionados

As criptomoedas desempenham uma mudança de paradigma na sociedade atual, com Bitcoin e Ethereum liderando o mercado e sendo as precursoras de diversas outras critopmoedas. Por isso, os protocolos de consenso para corrente de blocos atraem a atenção de diversos grupos de pesquisa [Melo et al. 2018, Aliaga et al. 2018, Miers et al. 2019, Oliveira et al. 2019]. Entretanto, as vulnerabilidades associadas a cada protocolo de consenso e suas respectivas contramedidas possuem poucos estudos.

Gervais *et al*. propõem um arcabouço para análise de segurança em correntes de bloco baseadas na prova de trabalho [Gervais et al. 2016]. Xiao *et al*. modelam a segurança da prova de trabalho de acordo com a conectividade dos participantes em relação aos ataques de mineração egoísta e o conluio entre participantes [Xiao et al. 2020a]. Conti *et al.* analisam diversos componentes e suas respectivas vulnerabilidades na corrente de blocos do Bitcoin [Conti et al. 2018]. Li *et al.* analisam a segurança de consensos baseados em prova de posse [Li et al. 2017]. Li *et al.* sumarizam as principais vulnerabilidades de segurança em ambientes de correntes de blocos [Li et al. 2020]. Além disso, os autores apresentam casos reais de ataques nas duas maiores criptomoedas com maior capital de mercado: Bitcoin e Ethereum. Entretanto, os trabalhos se baseiam apenas em um consenso, sem estender a análise e as propostas para outros protocolos probabilísticos.

[12]O protocolo PoET é o principal protocolo de consenso da plataforma Hyperledger Sawtooth.

Xiao *et al.* [Xiao et al. 2020b] e Joshi *et al.* [Joshi et al. 2018] reúnem diferentes protocolos de consenso determinísticos e probabilísticos para corrente de blocos. Os artigos analisam a segurança de diferentes correntes de blocos probabilísticas e determinísticas. Zhang *et al*. dividem a arquitetura de corrente de blocos em seis camadas e analisam a segurança de cada uma [Zhang and Zhou 2020]. Entretanto, a camada de consenso não é amplamente analisada.

Este artigo, diferente dos trabalhos anteriores, sumariza os principais aspectos dos protocolos de consenso baseados em prova mais utilizados, focando nas principais vulnerabilidades e nos ataques de cada protocolo, com suas respectivas contramedidas.

7. Conclusão

Os protocolos baseados em prova, ao contrário dos protocolos determinísticos, apresentam possibilidades de bifurcações, pois qualquer participante pode propor um bloco e a probabilidade de propor blocos ao mesmo tempo não é baixa. Participantes maliciosos exploram essas inconsistências temporárias para realizar diversos ataques, que não são possíveis em protocolos determinísticos. A prova de trabalho é o primeiro protocolo de consenso probabilístico aplicado com sucesso em uma rede pública. No entanto, o seu custo energético é proibitivo. A mineração remunerada leva a centralização de poderosos mineradores que podem comprar *hardware* de alto desempenho.

A prova de posse é a principal alternativa à prova de trabalho em termos de custo energético, mas apresenta novas vulnerabilidades como o "nada a perder" e a de "longo alcance". Na prova de posse, também é necessária a remuneração para se fazer as "apostas" e a centralização deve ser um problema. Como não há gasto de tempo para se resolver um desafio, a produção de blocos e a consequente vazão de transações é elevada. Entretanto, o número de bifurcações elevado aumenta o risco de ataques. A prova de posse delegada combina a escalabilidade de consensos baseados em prova ao determinismo de protocolos baseados em voto. Porém, o modelo delegado é mais centralizado do que os anteriores, sendo mais sensível ao conluio entre participantes maliciosos. Apesar das diferentes vulnerabilidades do consenso por prova de trabalho, é fato que, na prática, a segurança do Bitcoin é excepcional, pois não houve nenhum ataque bem-sucedido ao protocolo em mais de 11 anos de existência. Qualquer outro consenso que o venha substituir deve provar que possui esta robustez a ataques.

Em trabalhos futuros, os autores pretendem estudar os protocolos híbridos, uma vez que é provável que a melhor proposta de consenso seja uma proposta híbrida que combina o consenso determinístico com o consenso probabilístico.

Referências

Aliaga, Y. E. M., Leal, V. C., de Lucena, A. U., and Henriques, M. A. A. (2018). Avaliação de mecanismos de consenso para blockchains em busca de nova estratégia mais eficiente e segura. In *SBSeg*, pages 33–40. SBC.

Angelis, S. D., Aniello, L., Baldoni, R., Lombardi, F., Margheri, A., and Sassone, V. (2018). PBFT vs proof-of-authority: applying the CAP theorem to permissioned blockchain. In *Italian Conference on Cyber Security (06/02/18)*.

Bahack, L. (2013). Theoretical Bitcoin attacks with less than half of the computational power (draft). *arXiv preprint arXiv:1312.7013*.

Bessani, A. N., Sousa, J., and Alchieri, E. A. P. (2014). State machine replication for the masses with bft-smart. In *IEEE/IFIP DSN*, pages 355–362.

Bonneau, J., Felten, E. W., Goldfeder, S., Kroll, J. A., and Narayanan, A. (2016). Why buy when you can rent? In *ICFCDS*, pages 19–26. Springer.

Buchman, E. (2016). *Tendermint: Byzantine fault tolerance in the age of blockchains*. PhD thesis, University of Guelph.

Buterin, V. (2019). Proof-of-Stake FAQ. Acessado em 20 de julho de 2020.

Castro, M. and Liskov, B. (2002). Practical Byzantine Fault-Tolerance and Proactive Recovery. *ACM Transactions on Computer Systems*, 20(4):398–461.

Chen, L., Xu, L., Shah, N., Gao, Z., Lu, Y., and Shi, W. (2017). On security analysis of proof-of-elapsed-time (poet). In *SSS*, pages 282–297. Springer.

Conti, M., Kumar, E. S., Lal, C., and Ruj, S. (2018). A survey on security and privacy issues of Bitcoin. *IEEE Communications Surveys & Tutorials*, 20(4):3416–3452.

Deirmentzoglou, E., Papakyriakopoulos, G., and Patsakis, C. (2019). A survey on long-range attacks for proof of stake protocols. *IEEE Access*, 7:28712–28725.

Douceur, J. R. (2002). The sybil attack. In *International workshop on peer-to-peer systems*, pages 251–260. Springer.

Ekparinya, P., Gramoli, V., and Jourjon, G. (2019). The attack of the clones against proof-of-authority. *arXiv preprint arXiv:1902.10244*.

Eletrobras (2017). Relatórios de sustentabilidade socioambiental. Technical report, Eletrobras S.A. Acessado em 20 de julho de 2020.

Eyal, I. and Sirer, E. G. (2018). Majority is Not Enough: Bitcoin Mining is Vulnerable. *Commun. ACM*, 61(7):95–102.

Finney, H. (2011). Best practice for fast transaction acceptance-how high is the risk? Disponível em https://bitcointalk.org/index.php?topic=3441.msg48384#msg48384. Acessado em 20 de julho de 2020.

Fischer, M. J., Lynch, N. A., and Paterson, M. S. (1985). Impossibility of distributed consensus with one faulty process. *JACM*, 32(2):374–382.

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. In *ACM SIGSAC*, pages 3–16.

Heilman, E., Kendler, A., Zohar, A., and Goldberg, S. (2015). Eclipse attacks on bitcoin's peer-to-peer network. In *USENIX Security'15*, pages 129–144.

Johnson, B., Laszka, A., Grossklags, J., Vasek, M., and Moore, T. (2014). Game-theoretic analysis of DDoS attacks against Bitcoin mining pools. In *ICFCDS*, pages 72–86.

Joshi, A. P., Han, M., and Wang, Y. (2018). A survey on security and privacy issues of blockchain technology. *Mathematical foundations of computing*, 1(2):121.

Karame, G. O., Androulaki, E., and Capkun, S. (2012). Double-spending fast payments in bitcoin. In *ACM CCS 2012*, pages 906–917.

Kiayias, A., Russell, A., David, B., and Oliynykov, R. (2017). Ouroboros: A provably secure proof-of-stake blockchain protocol. In *CRYPTO 2017*, pages 357–388.

Larimer, D. (2017). EOS.IO White Paper. Disponível em https://developers.eos.io/-welcome/latest/protocol/consensus_protocol. Acessado em 20 de julho de 2020.

Li, W., Andreina, S., Bohli, J.-M., and Karame, G. (2017). Securing proof-of-stake blockchain protocols. In *DPM/CBT*, pages 297–315. Springer.

Li, X., Jiang, P., Chen, T., Luo, X., and Wen, Q. (2020). A survey on the security of blockchain systems. *Future Generation Computer Systems*, 107:841–853.

Melo, W., Carmo, L. F., Bessani, A., Neves, N., and Santin, A. (2018). How blockchains can improve measuring instruments regulation and control. In *I2MTC*, pages 1–6. IEEE.

Miers, C., Koslovski, G., Pillon, M., Simplício, M., Carvalho, T., Rodrigues, B., and Battisti, J. (2019). *Análise de Mecanismos para Consenso Distribuído Aplicados a Blockchain*, chapter 3. SBC.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Acessado em 20 de julho de 2020.

Oliveira, M. T. et al. (2019). Towards a performance evaluation of private blockchain frameworks using a realistic workload. In *ICIN*, pages 180–187. IEEE.

Olson, K., Bowman, M., Mitchell, J., Amundson, S., Middleton, D., and Montgomery, C. (2018). Sawtooth: An Introduction. *The Linux Foundation*.

Palma, L. M., Vigil, M. A., Pereira, F. L., and Martina, J. E. (2019). Blockchain and smart contracts for higher education registry in Brazil. *IJNM*, 29(3):e2061.

Pinno, O. J. A., Gregio, A. R. A., and De Bona, L. C. (2017). ControlChain: Blockchain as a central enabler for access control authorizations in the IoT. In *IEEE GLOBECOM*, pages 1–6.

Rebello, G. A. F., Alvarenga, I. D., Sanz, I. J., and Duarte, O. C. M. (2019a). Bsec-nfvo: A blockchain-based security for network function virtualization orchestration. In *IEEE International Conference on Communications (ICC)*, pages 1–6.

Rebello, G. A. F. et al. (2019b). Correntes de Blocos: Algoritmos de Consenso e Implementação na Plataforma Hyperledger Fabric. In *CSBC 2019 - 38º JAI*, pages 1–59.

Schwartz, D., Youngs, N., and Britto, A. (2014). The Ripple Protocol Consensus Algorithm. *Ripple Labs Inc White Paper*.

Wang, W., Hoang, D. T., Xiong, Z., Niyato, D., Wang, P., Hu, P., and Wen, Y. (2018). A survey on consensus mechanisms and mining management in blockchain networks. *CoRR*, abs/1805.02707.

Xiao, Y., Zhang, N., Lou, W., and Hou, Y. T. (2020a). Modeling the impact of network connectivity on consensus security of proof-of-work blockchain. *arXiv preprint arXiv:2002.08912*.

Xiao, Y., Zhang, N., Lou, W., and Hou, Y. T. (2020b). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, 22(2):1432–1465.

Zhang, P. and Zhou, M. (2020). Security and trust in blockchains: Architecture, key technologies, and open issues. *IEEE TCSS*, 7(3):790–801.