A Simple Analysis of Efficiency and Security of the Blockchain Technology

Uma análise simples de eficiência e segurança da Tecnologia Blockchain

Carlo Kleber da Silva Rodrigues [1]

[1] Universidade de Brasília (UnB),Centro Universitário de Brasília (UniCEUB), Brasília, Brasil

carlokleber@gmail.com

Abstract. *This paper aims at performing a simple analysis of the efficiency and security of the Blockchain technology. To this end, a theoretical study of the data structures and algorithms used for registering information is initially carried out, seeking to determine the complexities of time and space for the execution of search operations. Then, through analytical modeling and simulation, different scenarios are examined to assess the security level in terms of information inviolability. In particular, the final results show that search operations have logarithmic cost in function of the number of transactions within each data block and that system security mainly depends on information´s registration time. Within this context, the main contribution of this paper is thus to provide the literature with evidences of how efficient and secure the Blockchain technology is. Lastly, overall conclusions and future avenues come at the end of the article.*

Resumo. *Este artigo tem como objetivo realizar uma análise simples da eficiência e da segurança da tecnologia Blockchain. Para este fim, um estudo teórico das estruturas de dados e algoritmos utilizados para armazenar informações é inicialmente realizado, buscando determinar as complexidades de tempo e espaço para a execução das operações de busca. Em seguida, através de modelagem analítica e simulação, diferentes cenários são examinados para avaliar o nível de segurança em termos de inviolabilidade da informação. Em particular, os resultados finais mostram que as operações de pesquisa têm custo logarítmico em função do número de transações dentro de cada bloco de dados e que a segurança do sistema depende principalmente do tempo de registro da informação. Neste contexto, a principal contribuição deste artigo é, portanto, prover a literatura com evidências de quão eficiente e segura é a tecnologia Blockchain. Por fim, conclusões gerais e trabalhos futuros encerram este artigo.*

I. INTRODUÇÃO

Blockchain é uma tecnologia para armazenamento descentralizado de informações. Essas informações se referem a transações geradas por clientes de aplicações de arquitetura *peer-to-peer* (*P2P*) [1][2].

As transações são armazenadas em blocos de dados interligados entre si. Por definição, cada bloco está ligado a apenas um bloco anterior a ele, resultando em uma cadeia sequencial de blocos. Antes de ser adicionado à cadeia, cada bloco é validado por um processo matemático computacional denominado *mineração*. Esse processo pode ser executado por um único *peer*, denominado *minerador*, ou por grupos de *peers*, denominados *mining pools* [3][4].

Blockchain teve sua origem no ano de 2008, juntamente com a proposta da criptomoeda *Bitcoin* [1]. Esse conceito tecnológico se tornou a fundamentação de praticamente todas as criptomoedas que surgiram desde então. Apesar de sua concepção original para pagamentos eletrônicos, a tecnologia *Blockchain* é de aplicabilidade bem mais abrangente [5][6].

Mais especificamente, as transações podem ter significados semânticos diversos, não se restringindo absolutamente a valores monetários. Por exemplo, as transações podem se relacionar a registros de votos, montantes alocados em áreas de investimento, direitos de propriedades, acessos a dispositivos de plataformas como *Internet of Things* (*IoT*), *Smart Grids* e *Smart Cities*, dentre outros [2][5][7][8].

Este contexto de inovação tecnológica promovido pela *Blockchain* é a motivação deste artigo, o qual possui dois objetivos precípuos: primeiro, analisar a eficiência das estruturas de dados e algoritmos para consulta de transações no sistema; segundo, analisar a segurança quando da tentativa de alteração de transações registradas no sistema. Para tanto, inicialmente é feito um estudo teórico das estruturas de dados e dos algoritmos empregados para manipulação dos dados [9]. Na sequência, usando modelagem analítica e simulação [10], são examinados distintos cenários de atividade para mensurar o nível de segurança quanto à inviolabilidade das informações.

O restante deste artigo está organizado como segue. A Seção II discorre sobre trabalhos relacionados. A Seção III explica a tecnologia *Blockchain*, considerando as estruturas de dados e os algoritmos definidos originalmente em [1]. A Seção IV conduz uma análise de segurança, com foco na garantia da inviolabilidade das informações. Finalmente, conclusões gerais e trabalhos futuros constituem a Seção V.

II. TRABALHOS RELACIONADOS

Esta seção destaca alguns dos mais relevantes e recentes trabalhos da literatura que se relacionam direta ou indiretamente com o objetivo deste artigo. Opta-se por uma sequência de citação que busca enaltecer o aspecto conceitual inovador introduzido pela proposta, em vez de seus detalhes de implementação. Assim, crê-se que o leitor pode mais facilmente ter uma visão contemplativa da elaboração e da diferenciação entre as propostas, bem como ter uma visualização mais objetiva do atual estado da arte.

Em [11] e [12] são analisados ataques cibernéticos, vislumbrando comprovar a inadequação das infraestruturas de redes legadas, identificar e caracterizar os atores originadores de ameaças, suas capacidades, motivações e propósitos. São ainda apresentadas as melhores práticas para enfrentar a realidade corrente e há o delineamento de cenários de aplicabilidade da tecnologia *Blockchain*.

Em [13] é proposta uma estrutura de dados baseada em árvores em substituição à cadeia de blocos da tecnologia *Blockchain*. São realizadas análises competitivas, considerando a segurança e a efetividade que podem ser alcançadas. Utilizando

modelagem analítica, consegue-se verificar que, nos piores cenários de segurança estáticos, os protocolos mais modernos, como o protocolo *GHOST* [14], têm um desempenho inferior ou, na melhor das hipóteses, apenas similar àquele da *Blockchain*.

Em [15] e [16] é estudada a vulnerabilidade da tecnologia *Blockchain* sob a condição de ataques de *mineradores desonestos*, os quais recebem *recompensas* pela *mineração* maiores do que aquelas dos *mineradores honestos*. A *recompensa* é o pagamento recebido pelo *minerador* ou pelo *mining pool* por ter *minerado* blocos. Por meio de modelagem analítica e simulação, são alcançadas evidências de que, sem haver regulações da *mineração* provida por *mining pools*, o sistema pode deixar de ser verificável de maneira descentralizada, vulnerabilizando consequentemente o sistema.

Em [17], [18] e [19] são analisados ataques específicos de *double-spending* [1] em sistemas baseados em *Blockchain*. Esses ataques objetivam permitir que um mesmo recurso seja usado em diferentes transações. A partir de modelos analíticos e simulação, são obtidos resultados que permitem inferir sobre a probabilidade de ocorrência desse tipo de ataque e conjecturar sobre contramedidas de defesa.

Por fim, em [14] e [20] são discutidos aspectos da escalabilidade da *Blockchain*, considerando a quantidade de informações, o tempo de verificação de transações, a otimização das estruturas de dados e dos algoritmos, o tamanho dos blocos da cadeia, a diversidade de aplicações, dentre outros. Os resultados alcançados permitem conjecturar sobre uma nova geração de aplicações em distintas áreas de conhecimento, alicerçadas pela tecnologia *Blockchain* [5][7].

Ante o exposto, em que pese a diversidade e a importância dos resultados já alcançados na literatura, a contribuição e a diferenciação deste artigo se revelam pelo relativo ineditismo da realização de uma análise bastante objetiva e concisa que contempla a análise de eficiência das estruturas de dados e dos algoritmos usados, sob o viés teórico da complexidade de tempo e espaço [9] e, ainda, a avaliação da segurança quanto à inviolabilidade das informações já registradas no sistema, sob o viés combinado da modelagem analítica e da simulação [10].

III. EFICIÊNCIA DAS ESTRUTURAS DE DADOS E ALGORITMOS

A. *Estrutura do Bloco*

Cada bloco da cadeia é constituído por um *cabeçalho* seguido por uma lista de transações, conforme descrição na Tabela 1. O cabeçalho tem 80 bytes e o tamanho médio de uma transação é de pelo menos 250 bytes. O número médio de transações em um bloco é 500. Portanto, um bloco inteiro, com todas as transações, é aproximadamente 1.000 vezes maior que o tamanho do cabeçalho isolado [21].

Conforme Tabela 2, o campo *cabeçalho* possui três grupos de metadados. O primeiro grupo se refere ao número de *versão* do protocolo e ao *hash do bloco anterior*. No segundo grupo, há os campos *dificuldade*, *instante de criação* e *nonce*. Estas informações se relacionam ao processo de *mineração*. Por último, o terceiro grupo é a *raiz da árvore de Merkle*, que possui um resumo de todas as transações armazenadas no bloco.

TABELA I - ESTRUTURA DO BLOCO

Tamanho	Campo	Descrição
4 bytes	*Tamanho do bloco*	Tamanho do bloco, medido em bytes.
80 bytes	*Cabeçalho*	Informações relacionadas ao processo de *mineração* do bloco, permitindo sua identificação.
1 a 9 bytes	*Contador de transações*	Número de transações existentes no bloco.
Variável	*Transações*	Transações armazenadas no bloco.

TABELA II - ESTRUTURA DO CABEÇALHO

Tamanho	Campo	Descrição
4 bytes	*Versão*	Número de versão do protocolo
32 bytes	*Hash do bloco anterior*	*Hash* do bloco anterior na cadeia.
32 bytes	*Raiz da árvore de Merkle*	*Hash* da raiz da árvore de *Merkle*. É um *resumo* das transações existentes no bloco.
4 bytes	*Instante de criação*	Instante de criação do bloco.
4 bytes	*Dificuldade*	Grau de dificuldade para *mineração*. Este valor é ajustado consensualmente para garantir um tempo mínimo de *mineração* por bloco.
Variável	*Nonce*	Número inteiro encontrado como solução no processo de *mineração*. Após encontrado, este valor é denominado *golden nonce*.

Cada bloco é identificado pelo *hash* de seu próprio cabeçalho. O algoritmo de *hash* criptográfico utilizado é o *SHA256* [22]. Como já mencionado, cada bloco faz referência a apenas um bloco anterior, também chamado de *bloco pai*. Essa referência se implementa por meio do campo *hash do bloco anterior*. Ou seja, cada bloco contém o *hash* de seu *pai* dentro de seu próprio cabeçalho. A sequência de *hashes* que liga cada bloco ao seu *pai* cria uma cadeia de blocos que faz o caminho de volta até o primeiro bloco já criado no sistema, denominado *bloco gênese.* A mudança da identidade de um bloco *B* geraria um efeito cascata da mudança da identidade de todos os blocos subsequentes a ele próprio (i.e., bloco *B*).

B. Processo de Mineração

Toda transação realizada entre os clientes é enviada pela rede em *broadcast*. Cada *minerador* trabalha agrupando as transações que chegam até ele em blocos de dados. Para cada bloco de dados, o *minerador* passa então a tentar encontrar a *prova de trabalho* (do inglês, *proof-of-work*) [4].

A *mineração* de um bloco é, pois, o processo para encontrar a *prova de trabalho* do bloco. Matematicamente, encontrar a *prova de trabalho* significa determinar, por meio de tentativas sucessivas, o valor de *nonce* para satisfazer a desigualdade $SHA256(SHA256(data + nonce)) < Dificuldade$ [1], onde: *data* é o *hash* do conteúdo do cabeçalho do bloco de transações a validar; *nonce* e *Dificuldade* estão definidos na Tabela 2 [21].

C. Armazenamento das Transações

Conforme Tabela 1, um bloco da cadeia armazena todas as transações nele existentes no campo *transações*. Além disso, conforme Tabela 2, todo bloco também guarda um resumo de todas as suas transações no campo *raiz da árvore de Merkle*. A computação desse resumo é explicada a seguir.

A *árvore de Merkle* [23], também conhecida como árvore binária de *hash*, é construída a partir dos seus vértices folhas. Cada folha guarda o *hash* de uma das transações existentes no bloco. São então computados recursivamente os *hashes* de pares de vértices da árvore, a partir dos vértices folhas, até que se obtenha um único *hash*, o qual corresponde ao resumo guardado na raiz da árvore. O algoritmo criptográfico utilizado é o *SHA256* [22], aplicado duas vezes.

A Figura 1 ilustra um exemplo de uma árvore de *Merkle* para um bloco de quatro transações: *A*, *B*, *C*, *D*. No caso de o número de transações ser ímpar, a última transação deve ser duplicada, obtendo uma *árvore binária cheia* [9].

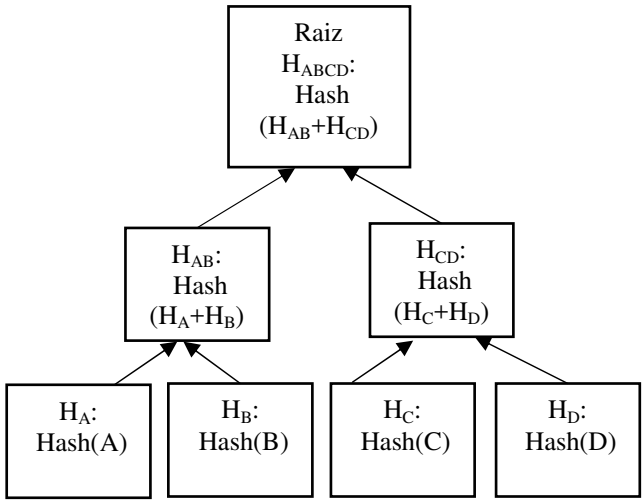

Figura 1. *Árvore de Merkle* para 4 transações: A, B, C e D.

D. Análise de Complexidade

Após sua *mineração*, o bloco deve então ser adicionado à cadeia de blocos do sistema. Para tanto, o bloco *minerado* é propagado pela rede em *broadcast* e se aguarda para que os *nós completos*, *peers* que mantêm uma cópia local de toda a cadeia de blocos, iniciada a partir do bloco *gênese*, realizem a sua adição. A cópia de cadeia de blocos local é atualizada à medida que novos blocos são *minerados*. A convergência das cópias locais dos *nós completos* comprova as transações realizadas no sistema.

Para adição de um novo bloco, B_{novo}, um *nó completo*, *P*, inicialmente examina o cabeçalho desse bloco para saber o valor do campo *hash do bloco anterior*. Seja $h(B_{anterior})$ este valor. Admita que *P* possui *m* blocos na sua cadeia local, sendo B_m o último bloco adicionado e $h(B_m)$ o *hash* do seu cabeçalho. *P* adiciona B_{novo} à sua cadeia somente se $h(B_m) = h(B_{anterior})$ e se a *prova de trabalho* tiver sido realizada. Para verificar a realização da *prova de trabalho*, basta confirmar que a desigualdade $SHA256(SHA256(data + nonce)) < D$ é satisfeita [1], onde *data* e *nonce* são obtidos a partir do cabeçalho de B_{novo}. Embora a computação do valor de *nonce* seja intrincada, a sua verificação é de complexidade de tempo constante. Assim, a complexidade de tempo da operação de adição de um bloco é $O(1)$.

Para consultar uma transação, ou seja, verificar a sua existência, são usados os nós de Verificação Simplificada de Transação (VST), ou simplesmente nós VST. Um nó VST utiliza o denominado *caminho de Merkle* ou *caminho de autenticação* [21]. Suponha que o *peer Q* seja um nó VST que deseja verificar se uma transação *s* ocorreu, tendo como uma das partes envolvidas o cliente de endereço lógico *E*.

O *nó Q* deve então criar um filtro em suas conexões com os demais *peers* da rede para restringir as transações a ele informadas, ou seja, apenas transações que se relacionam ao endereço de interesse *E* devem ser propagadas até ele. Quando o *nó Q* vê uma transação com o endereço lógico *E*, ele então informa sobre aquele bloco enviando uma mensagem especial, denominada *mensagem de bloco de Merkle* (MBM). Esta mensagem contém o *cabeçalho do bloco* e o *caminho de Merkle* que liga a transação *s* à *raiz da árvore de Merkle* do bloco.

Seja a sequência de vértices (v_1, v_2, ..., v_l) o *caminho de Merkle* informado ao nó *Q*. Cada vértice está em um nível imediatamente inferior ao anterior na *árvore de Merkle*. O vértice v_1 é uma folha e o vértice v_l está no nível imediatamente superior ao nível da raiz da árvore, que está no nível 1. A verificação da existência da transação *s* consiste em computar recursivamente o *hash* dos vértices irmãos correspondentes na árvore e, ao final, no segundo nível, verificar se o *hash* computado se iguala ao valor da raiz, obtido do exame do *cabeçalho do bloco*, que é também informado na MBM. Se a identidade for observada, então a transação de fato existe.

Essa operação de consulta tem complexidade assintótica de tempo $O(\lfloor log_2(2n-1) \rfloor)$, onde *n* é o número total de transações existentes no bloco, o que torna a *árvore de Merkle* uma estrutura de dados extremamente eficiente. Este resultado resulta da condição de que a *árvore de Merkle* construída é uma árvore binária cheia e, portanto, tem altura logarítmica [9]. Além disso, a complexidade de espaço também é bem atrativa. Por exemplo, conforme definições da Tabela 1, note que para *n* igual 65.535 transações, com um bloco consumindo cerca de 16 MB, o *caminho de Merkle* consiste de apenas 16 *hashes* e tem comprimento de 512 bytes, pois cada *hash* resulta em 32 bytes.

IV. AVALIAÇÃO DE SEGURANÇA

Esta seção analisa o nível de segurança da tecnologia *Blockchain* considerando a seguinte pergunta: qual o tempo de registro de uma transação no sistema que deve ser considerado para garantir um nível de segurança aceitável, o qual se traduz na inviolabilidade da informação já registrada?

A. *Cenário de Investigação*

Quando dois *mineradores* enviam diferentes versões daquele que seria o próximo bloco da cadeia, os nós *completos* vão receber uma ou outra versão primeiramente. Cada nó *completo* considera a primeira versão recebida, mas vai também criar uma *ramificação* (do inglês, *branch* ou *fork* [7]) em sua cadeia local para adicionar a outra versão do bloco recebida posteriormente. Passam então a existir duas *ramificações*: uma para a primeira versão do bloco e a outra para a segunda versão.

A decisão de escolher entre as duas *ramificações* ocorre quando as próximas *provas de trabalho* são encontradas, fazendo com que uma das *ramificações* se torne mais longa que a outra. Neste instante, a *ramificação* mais curta é desprezada. Para ser considerada válida, uma transação deve pertencer a uma *ramificação* que não se tornará mais curta que outra. Esse entendimento se aplica independentemente do número de *ramificações* concorrentes. Sem perda de generalidade, doravante se consideram duas *ramificações* concorrentes.

A disputa entre duas *ramificações* pode ser vista como uma *corrida* para adição de blocos: um *minerador desonesto* deseja substituir um bloco *B*, que já tem *z* blocos subsequentes a ele na cadeia, e um *minerador honesto*, que adicionou o bloco *B* à cadeia no passado e está *minerando* o bloco seguinte *z*+1. Para ter sucesso na *fraude* (i.e., substituição do bloco *B*), resta ao *minerador desonesto* ser mais rápido que o *minerador honesto* para compensar a desvantagem inicial de *z* blocos.

B. *Modelagem Analítica*

A corrida descrita na subseção anterior pode ser modelada pelo *Passeio Aleatório Binomial* [1][24]. O evento de sucesso é a *ramificação honesta* ser estendida em um bloco, aumentando sua vantagem em +1, e o evento de fracasso é a *ramificação fraudulenta* ser estendida em um 1 bloco, reduzindo sua desvantagem em −1. Daí, a probabilidade de uma *ramificação fraudulenta* compensar uma certa desvantagem de z blocos é análoga a um problema de *Ruína do Jogador*, podendo ser calculada pela Equação 1 [1][24].

$$q_z = \begin{cases} 1, & se\ p \leq q \\ \left(\frac{q}{p}\right)^z, & se\ p > q \end{cases} \qquad (1)$$

Onde: p é a probabilidade de o *minerador honesto* resolver o próximo bloco; q é a probabilidade de o *minerador desonesto* resolver o próximo bloco; e q_z é a probabilidade de o *minerador desonesto* compensar a desvantagem inicial de z blocos.

O progresso de um *minerador desonesto*, ou seja, quantos blocos ele consegue *minerar* durante o período de *mineração* de z blocos *honestos*, pode ser modelado como uma distribuição de *Poisson* de valor médio α calculado pela Equação 2 [1][24].

$$\alpha = z\frac{q}{p} \qquad (2)$$

Por meio das Equações 1 e 2, pode-se então determinar a probabilidade de o *minerador desonesto* compensar a diferença de z blocos *honestos*, considerando que ele conseguiu *minerar* k blocos enquanto esperava os z blocos *honestos* serem adicionados à cadeia.

Para tanto, realiza-se o somatório da probabilidade de o *minerador desonesto* ter conseguido *minerar* k blocos multiplicada pela probabilidade de conseguir *minerar* ($z - k$) blocos faltantes, para $k = 0, 1, 2, ..., \infty$. Essa modelagem se baseia no *Teorema da Probabilidade Total* e é expressa pela Equação 3, que pode ainda ser reescrita na forma da Equação 4, evitando-se o somatório até o infinito [1][24].

$$\sum_{k=0}^{\infty} \frac{\alpha^k e^{-\alpha}}{k!} * \begin{cases} q/p^{z-k} & se\ k \leq z \\ 1 & se\ k > z \end{cases} \qquad (3)$$

$$1 - \sum_{k=0}^{z} \frac{\alpha^k e^{-\alpha}}{k!} \left[1 - \left(\frac{q}{p}\right)^{z-k}\right] \quad (4)$$

C. Resultados da Modelagem Analítica

Os resultados numéricos apresentados nesta subseção são calculados a partir da Equação 4. A Fig. 2 apresenta cinco cenários de análise: S_1, S_2, S_3, S_4 e S_5. Em cada cenário, um valor fixo de z é considerado e se varia o valor de p para computar o valor da probabilidade de *fraude* correspondente.

Quanto maior o valor de z, maior é o nível de segurança do sistema, pois mais difícil se torna a substituição de um bloco. No entanto, um maior valor de z também aumenta o tempo necessário de espera pela confirmação da transação no sistema. Assim, há um compromisso entre o tempo de espera tolerável e a segurança oferecida no sistema.

Sob a suposição de que todos os *peers mineradores* têm a mesma capacidade de *mineração* e que o tempo de *mineração* tem distribuição exponencial [24], a probabilidade p permite inferir sobre o percentual das populações de *mineradores honestos* e *desonestos* no sistema.

A partir dos resultados mostrados na Fig. 2, nota-se que a população de *mineradores honestos* deve ser maior que a população de *mineradores desonestos* ou, equivalentemente, p deve ser maior do que pelo menos 0,5. Caso contrário, a substituição do bloco certamente ocorrerá independentemente do valor arbitrado para z.

A Fig. 2 também possibilita estabelecer compromissos entre a segurança e o tempo tolerável para a confirmação de uma transação. Por exemplo, para garantir uma probabilidade de *fraude* menor que 0,05, é preciso uma população de *mineradores honestos* de cerca de 80% do total de *mineradores* e uma espera de seis blocos ou, alternativamente, uma população de *mineradores honestos* de cerca de 70% do total de *mineradores* e uma espera de 10 blocos.

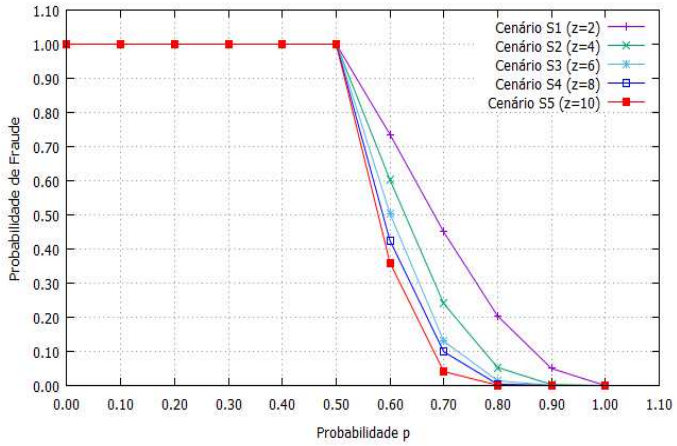

Figura 2. Análise da probabilidade de fraude em função de *p*.

A Fig. 3 traz outros cinco cenários de análise: S_6, S_7, S_8, S_9 e S_{10}. Agora se fixa um valor de p em cada cenário e se varia o valor de z para determinar o valor da probabilidade de *fraude* correspondente. Esses resultados também permitem discutir compromissos entre segurança e tempo tolerável de espera por confirmação. Por exemplo, se a população de *mineradores honestos* é 70% do total, então a probabilidade de *fraude* é menor que 0,05 quando a espera é de nove blocos (i.e., $z = 9$); ou se a população de *mineradores honestos* é de 90% do total, então a espera de apenas dois blocos (i.e., $z = 2$) já é suficiente para se ter essa mesma probabilidade de *fraude*. Por outro lado, se a população de *mineradores honestos* está abaixo de 60% do total, então mesmo uma espera de 25 blocos não é suficiente para alcançar essa probabilidade de *fraude* mencionada.

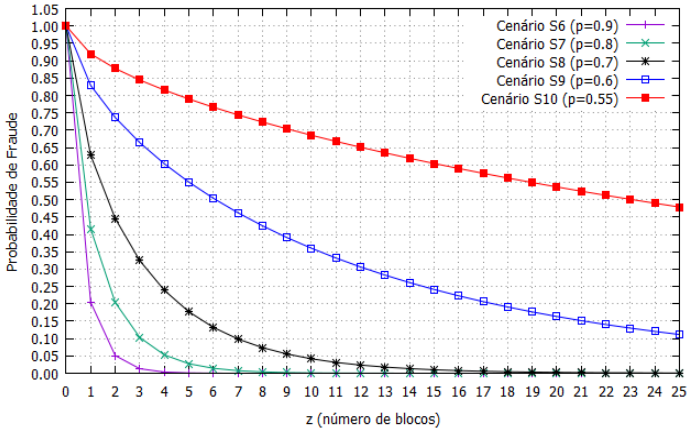

Figura 3. Análise da probabilidade de fraude em função de *z*.

D. *Simulação*

A simulação é realizada usando a ferramenta Tangram-II [25]. Esta ferramenta é um ambiente de modelagem de sistemas computacionais desenvolvido na Universidade Federal do Rio de Janeiro (UFRJ), com a participação da Universidade da Califórnia em Los Angeles (UCLA) nos EUA. Os modelos de simulação são implementados a partir da definição de objetos que interagem por troca de mensagens. Os modelos são resolvidos analiticamente ou via simulação [25].

O modelo de simulação desenvolvido para esta pesquisa é constituído de dois objetos: Obj_1 e Obj_2. O objeto Obj_1 emula os *mineradores* (*honestos* e *desonestos*) do sistema, os quais têm individualmente a mesma capacidade de *mineração*. O tempo de *mineração* de cada bloco tem distribuição exponencial com média $\mu = 10$ minutos [1][4]. O sistema é considerado no estado estacionário, i.e., embora *peers* possam entrar e sair do sistema, o número total de *peers* permanece constante. Por sua vez, o objeto Obj_2 emula a cadeia de blocos resultante das eventuais disputas entre *ramificações* concorrentes, possuindo as transações aceitas em definitivo. Os resultados de simulação têm intervalos de confiança de 95% que estão dentro do limite de 5% dos valores estimados, tendo sido consideradas 30 execuções (rodadas) com um tempo de simulação de 21.000 minutos cada.

De maneira geral, o modelo de simulação tem seu comportamento definido a partir da mesma visão conceitual usada na modelagem analítica. Os parâmetros p, q e z têm os mesmos significados e são usadas as mesmas distribuições de probabilidade. A principal diferença é que os experimentos de simulação permitem melhor analisar a evolução do estado do sistema ao longo do tempo, focando especialmente na diferença de tamanho entre as *ramificações* concorrentes.

E. *Resultados da Simulação*

A Fig. 4 traz resultados de seis cenários de análise: S_1, S_2, S_3, S_4, S_5 e S_6. Nos três primeiros cenários, considera-se $p = 0{,}6$ e varia-se z para determinar a probabilidade de a diferença de tamanho entre as *ramificações* concorrentes exceder o valor de X blocos. Nos três cenários seguintes, repete-se a mesma análise para $p = 0{,}8$. A partir desses resultados, existe a confirmação de que maiores valores de p e z tendem a fornecer uma maior segurança, pois esses valores tendem a aumentar a diferença de tamanho entre as *ramificações*.

Por exemplo, a maior probabilidade associada a $X = 0$ é obtida para $p = 0{,}8$ e $z = 6$ (i.e., Cenário S_6). Esse maior valor significa que a diferença de tamanho entre as *ramificações* neste cenário tem mais chance de ser maior do que zero do que nos demais cenários. Essa superioridade se mantém destacada para todos os valores de X em que a curva correspondente está definida. Assim, a *fraude* tem menor chance de ocorrer neste cenário, pois a *fraude* somente ocorreria se as *ramificações* se igualassem em tamanho, ou seja, se a desvantagem inicial de z blocos fosse compensada.

Por outro lado, aumentos arbitrários de p e z, sem uma prévia avaliação, podem ser inócuos. Por exemplo, os resultados obtidos para os Cenários S_1 e S_2, respectivamente, são praticamente idênticos, ou seja, o aumento de z, de 1 para 3, pouco contribui para segurança quando p é mantido em 0,6. Diferentemente, quando $p = 0{,}8$, o aumento de z, também de 1 para 3, aumenta a segurança do sistema, pois a diferença

entre as *ramificações* se torna maior, conforme pode ser visto a partir dos resultados dos Cenários S_4 e S_5, respectivamente.

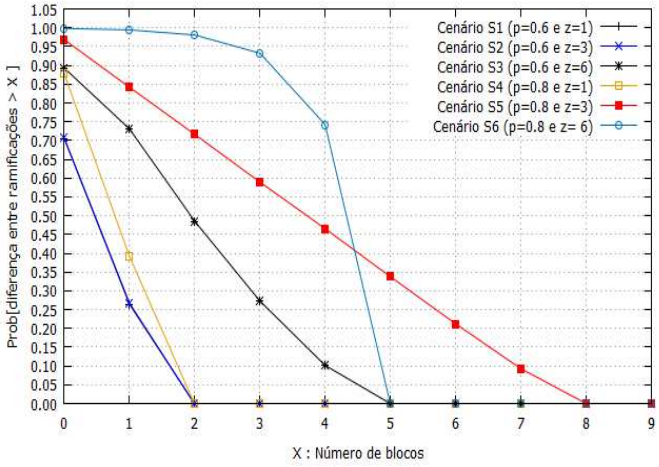

Figura 4. Análise da diferença de tamanho entre *ramificações*

A Fig. 5 traz resultados referentes aos mesmos seis cenários de análise anteriores. Desta vez se destacam os valores médios das diferenças, em número de blocos, entre as *ramificações* em cada um dos cenários. Como conclusão imediata, confirma-se novamente que, de maneira geral, maiores valores de p e z levam a uma maior segurança do sistema. Por exemplo, a diferença observada no Cenário S_6 (com p = 0,8 e z = 6) é quase cinco vezes maior que no Cenário S_1 (com p = 0,6 e z = 1). No entanto, como antes, confirma-se também que aumentos arbitrários de p e z, sem uma prévia análise, podem resultar inócuos. Por exemplo, dobrar o valor de z, de 3 para 6, não faz a diferença média ser dobrada, conforme pode ser visto no caso dos Cenários S_5 e S_6.

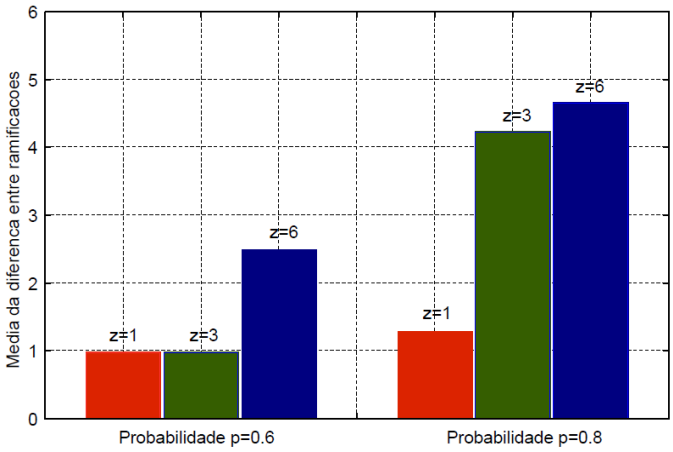

Figura 5. Médias das diferenças de tamanho entre *ramificações*.

Por fim, a Fig. 6 traz uma visualização gráfica da variabilidade no tempo da diferença de tamanho entre as ramificações, medida em número de blocos, nos dois cenários mais extremos da análise: Cenário S1 e Cenário S6. É imediato perceber o intervalo aproximado de [0; 2] para S1, e o intervalo aproximado de [3; 5] para S6. Os intervalos e a variabilidade que se observa visualmente corroboram as conclusões anteriores sobre a maior segurança do Cenário S.

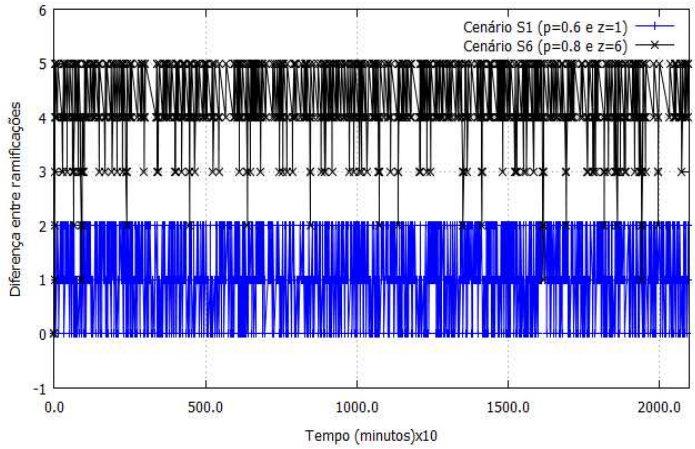

Figura 6. Variabilidade das diferenças de tamanho das *ramificações*

V. CONCLUSÕES E TRABALHOS FUTUROS

Este artigo teve o objetivo de realizar uma análise de eficiência e segurança da tecnologia *Blockchain*. Para tanto, inicialmente foi feito um estudo teórico das estruturas de dados e dos algoritmos de armazenamento e manipulação das informações, buscando determinar as complexidades de tempo e de espaço para a execução de operações de pesquisa. Na sequência, por meio de modelagem analítica e simulação, foram examinados diferentes cenários para mensurar o nível de segurança do sistema, sob o aspecto da inviolabilidade das informações registradas.

Ante os principais resultados e conclusões alcançadas, destacam-se os dois pontos que seguem. Primeiro, o estudo teórico demonstrou que a utilização da estrutura

de dados *árvore de Merkle* consiste em uma solução efetiva, pois garante a otimização de espaço com relação ao armazenamento das transações realizadas no sistema, bem como assegura que operações de pesquisa por transações no sistema possam ser realizadas em tempo logarítmico.

Segundo, os experimentos realizados por meio de modelagem analítica e simulação permitiram demonstrar que a segurança do sistema, em termos de inviolabilidade da informação, é dependente do tempo de registro da transação. Quanto mais antigo é o registro da transação, maior é o nível de segurança. Porém, um maior tempo de registro significa uma maior espera para que uma transação seja considerada aceita. É preciso então estabelecer um compromisso entre a segurança e o tempo de espera. Ainda, foi possível verificar que o tempo de registro da transação, para um certo nível de segurança, é influenciado pela probabilidade de *mineração honesta* de blocos. Quanto maior é essa probabilidade, menor é esse tempo.

Por fim, como trabalhos futuros, sugerem-se os seguintes caminhos. Primeiro, realizar novas análises de tempo da operação de pesquisa na *Blockchain*, considerando adicionalmente as complexidades de caso médio, melhor caso e amortizada [9]. Segundo, realizar uma análise competitiva entre o algoritmo *SHA256* e outros algoritmos criptográficos [22] para verificar a possibilidade de robustecer a segurança da *Blockchain,* bem como avaliar a possibilidade de otimização do tempo total de *mineração* [26][27].

Terceiro, realizar uma análise competitiva entre as mais recentes implementações de segurança baseadas em *Blockchain* ou de concepção semelhante a ela [5][28][29]. Por fim, realizar uma análise competitiva específica entre o método de consenso *proof-of-work* (PoW) [4], considerado nesta pesquisa, e aquele denominado de *proof-of-stake* (PoS) [2][7]. De forma simples, a diferença entre esses dois métodos é que, para estabelecer um consenso e evitar o problema do *double-spending* [1], o método PoS considera que o nó deve provar que possui acesso a uma certa quantidade de recursos (p. ex., criptomoedas) antes de ter sua transação aceita na rede, em vez de considerar a resolução de um problema matemático criptográfico como ocorre sob o método PoW [17][18]. A quantidade de recursos é determinada pela rede em um processo de ajustes semelhante àquele do método PoW. Ressalta-se que, além da eventual identificação absoluta de qual método é potencialmente mais seguro entre os dois, essa análise pode ainda trazer indícios para fins de categorização de aplicações distribuídas (em função de seus requisitos) em mais ou menos aderentes a um ou outro dos métodos [2][7][29].

REFERÊNCIAS

[1]S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system”. Available at:<https://bitcoin.org/bitcoin.pdf>. Accessed on: Oct. 8th, 2017.

[2] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, “A Survey on the security of blockchain systems,” Future Generation Computer Systems, 2017, in press.

[3]J. Garay, A. Kiayias, and N. Leonardos, “The bitcoin backbone protocol: analysis and applications,” Lecture Notes in Computer Science (LNCS), Springer, Berlin, Heidelberg, vol. 9057, pp. 281-310, 2015.

[4]G. A. Silva and C. K. S. Rodrigues, “Mineração individual de bitcoins e litecoins no mundo,” in Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2016), Niterói, Rio de Janeiro, Brasil, 2016.

[5]A. Dorri, S. Kanhere, and R. Jurdak, “Towards an optimized blockchain for IoT,” in International Conference on Internet-of-Things Design and Implementation (IoTDI´17), Pittsburgh, PA, USA, 2017.

[6]S. Huckle, R. Bhattachary, M. White, and N. Beloff, “Internet of Things, blockchain and shared economy applications,” Procedia Computer Science, vol. 98, pp. 461-466, 2016.

[7]Iuon-Chang Li and Tzu-Chun Liao, “A Survey of blockchain security issues and challenges,” International Journal of Network Security, vol.19, no.5, pp.653-659, 2017.

[8]A. Dorri, S. Kanhere, and R. Jurdak, “Blockchain in Internet of Things: challenges and solutions,” CoRR, abs/1608.05187, 2016. Available at: <https://arxiv.org/abs/1608.05187>. Accessed on: Oct. 8th, 2017.

[9]T. H. Cormen, C. E. Leiserson, and R. L. Rivest. Introduction to Algorithms. 3rd Edition. Cambridge, Massachussets: MIT Press, 2009.

[10] S. M. Ross. Simulation. 5th Edition. San Diego, California: Elsevier, 2013.

[11] E. U. Opara and O. A. Soluade, “Straddling the next cyber frontier: the empirical analysis on network security, exploits, and vulnerabilities,” International Journal of Electronics and Information Engineering, vol. 3, no. 1, pp. 10-18, 2015.

[12] J. Singh, “Cyber attacks in cloud computing: a case study,” International Journal of Electronics and Information Engineering, vol 1, no. 2, pp. 78-87, 2014.

[13] A. Kiayias and G. Panagiotakos, “On trees, chains and fast transactions in the blockchain,” Cryptology ePrint Archive, Report 2016/545, 2016. Available at: <https://eprint.iacr.org/2016/545>. Accessed on: Oct. 8th, 2017.

[14] Y. Sompolinsky and A. Zohar, “Secure high-rate transaction processing in bitcoin,” Lecture Notes in Computer Science (LNCS), Springer, Berlin, Heidelberg, vol 8975, pp. 507-527, 2015.

[15] I. Eyal and E. G. Sirer, “Majority is not enough: bitcoin mining is vulnerable,” CoRR, abs/1311.0243, 2013. Available at: <https://arxiv.org/abs/1311.0243>. Accessed on: Oct. 8th, 2017.

[16] N. T. Courtois and L. Bahack, “On subversive miner strategies and block withholding attacks in bitcoin digital currency,” CoRR, abs/1402.1718, 2014. Available at: < https://arxiv.org/abs/1402.1718>. Accessed on: Oct. 8th, 2017.

[17] G. O. Karame, E. Androulaki, and S. Capkun, “Two bitcoins at the price of one? Double-spending on fast payments in bitcoin,” in ACM Conference on Computer and Communications Security (CCS´12), Raleigh, NC, USA, 2012.

[18] M. Rosenfeld, “Analysis of hasrate-based double spending,” *CoRR*, abs/1402.2009, 2014. Available at: <https://arxiv.org/abs/1402.2009>. Accessed on: Oct. 8th, 2017.

[19] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in ACM Conference on Computer and Communications Security (CCS´16), Vienna, Austria, 2016.

[20] G. O. Karame, "On the security and scalability of bitcoin´s blockchain," in *ACM* Conference on Computer and Communications Security (CCS´16), Vienna, Austria, 2016.

[21] M. Antonopoulos, Mastering Bitcoin: Programming the Open Blockchain. 2nd Edition. Sebastopol, California: O´Reilly Media, 2017.

[22] H. Gilbert and H. Handschuh, "Security analysis of SHA-256 and sisters," Lecture Notes in Computer Science (LNCS), Springer, Berlin, Heidelberg, vol. 3006, pp. 175-193, 2004.

[23] P. Berman, M. Karpinski, and Y. Nekrich, "Optimal trade-off for Merkle tree traversal," Theoretical Computer Science, vol. 372, no. 1, pp. 26-36, 2007.

[24] S. M. Ross, Stochastic Processes. 2nd Edition. New York: John Wiley & Sons, 1996.

[25] E. de Souza e Silva, R. Figueiredo, and R. Leão, "The TANGRAM-II integrated modeling environment for computer systems and networks," ACM SIGMETRICS Performance Evaluation Review, vol. 36, no. 4, pp. 64-69, 2009.

[26] J. E. Pazmiño and C. K. S. Rodrigues, "Simply dividing a Bitcoin network node may reduce transaction verification time," The SIJ Transactions on Computer Networks & Communication Engineering, vol. 3, no. 2, pp. 17-21, 2015.

[27] J. J. Chávez and C. K. S. Rodrigues, "Automatic hopping amog pools and distributed applications in the Bitcoin Network," in XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA 2016), Bucaramanga, Colombia, 2016.

[28] Yonatan Sompolinsky and Yoad Lewenberg, "SPECTRE: Serialization of proof-of-work events – Confirming transactions via recursive elections", 2017. Available at:<https://eprint.iacr.org/2016/1159.pdf>. Accessed on: Nov. 14th, 2017.

[29] Mauro Conti, Sandeep Kumar E and Chhagan Lal, "A survey on security and privacy Issues of Bitcoin." CoRR, abs/1706.00916, 2017. Avalilable at:< http://arxiv.org/abs/1706.00916>. Accessed on: Nov. 14th, 2017.