Uma Análise Competitiva entre as Tecnologias *Blockchain* e *Tangle* para o Projeto de Aplicações IoT

Rafael Zerbini Alves da Mata[1], Carlo Kleber da Silva Rodrigues[2]

[1]Faculdade de Tecnologia – Departamento de Engenharia Elétrica – Universidade de Brasília (UnB) – Brasília – DF – Brasil.

[2]Centro de Matemática, Computação e Cognição – Universidade Federal do ABC (UFABC) – Santo André – SP – Brasil.

{rafael.zerbinia,carlokleber}@gmail.com

Abstract. *This paper aims at performing a competitive analysis between the Blockchain and Tangle technologies, mainly focusing on the design of IoT applications. To this end, we firstly carry out a theoretical comparative study of the data structures and algorithms used to manipulate information under each technology. Then, through simulations, we analyze different scenarios to assess the system security level in terms of data inviolability. Final results allow us to conjecture that Tangle is a more promising choice, besides constituting worthy insights for real designs of IoT applications. General conclusions and future work conclude this article.*

Resumo. *Este artigo tem por objetivo realizar uma análise competitiva entre as tecnologias Blockchain e Tangle, visando principalmente ao projeto de aplicações IoT. Para tanto, inicialmente é feito um estudo teórico comparativo das estruturas de dados e algoritmos usados para manipular informações sob cada tecnologia. Em seguida, por meio de simulação, diferentes cenários são analisados para aferir o nível de segurança do sistema em termos da inviolabilidade das informações, considerando o emprego de cada tecnologia. Os resultados finais permitem conjecturar que Tangle é uma alternativa mais promissora, além de se constituírem em subsídios para possíveis projetos de aplicações IoT. Conclusões gerais e trabalhos futuros encerram este artigo.*

1. Introdução

A Internet das Coisas (do inglês, *Internet of Things* − IoT) possui diversas aplicações, incluindo redes inteligentes, cidades inteligentes [Gaur et al. 2015], contratos inteligentes e gestão de saúde [Hassanalieragh et al. 2015]. Neste cenário, a rede de comunicação subjacente deve oferecer processamento e disseminação de dados de maneira transparente, intensa e pervasiva, o que traz preocupações relacionadas a aspectos de segurança, privacidade e escalabilidade [Dorri; Kanhere; e Jurdak 2017].

Considere, por exemplo, uma operação de crédito entre dispositivos IoT regulada por um contrato inteligente em uma rede de comunicação *peer-to-peer* (P2P). Os nós dos dispositivos IoT armazenam cópias consistentes do contrato inteligente. Um dos dispositivos pode solicitar a atualização do estado do contrato para atribuir crédito a um outro dispositivo para prestação de um serviço. Os nós constituintes da rede são

informados dessa atualização quando recebem individualmente o novo estado do contrato. Eventualmente, o dispositivo IoT destinatário do crédito verifica esse estado em seu nó e, em seguida, libera o serviço para o dispositivo solicitante.

Esse processo permite a troca de crédito na ausência de confiança mútua, uma vez que a transação é certificada por um terceiro, ou seja, a rede de nós, e é dificilmente reversível. O objetivo desse processo é, sobretudo, garantir que cada dispositivo observe o mesmo estado do contrato inteligente. A questão crucial de projeto é então fazer com que cada dispositivo permaneça sincronizado com a versão mais recente do contrato, além de garantir segurança, privacidade e escalabilidade.

Independentemente da natureza da aplicação IoT, a solução da questão ilustrada acima reside na escolha da tecnologia para manipular as informações das transações entre os dispositivos comunicantes. Nesse sentido, *Blockchain* [Nakamoto 2008] e *Tangle* [Popov 2017] são duas importantes tecnologias merecedoras de atenção na literatura [Dorri; Kanhere; e Jurdak 2017; Danzi et. al. 2017].

Blockchain se baseia na implementação de uma *lista encadeada* de blocos de transações (do inglês, *blockchain*). Esses blocos são adicionados à lista conforme são validados por meio de um processo matemático. Por sua vez, *Tangle* se baseia na implementação de um *Grafo Acíclico Direcionado* (do inglês, *Directed Acyclic Graph* − DAG), onde cada nó é uma transação. Para que a transação seja adicionada ao DAG, há um processo matemático, como na *Blockchain*, e ainda ocorre a validação de duas outras transações pré-existentes no DAG.

Este contexto é a motivação para este artigo, cujo objetivo é realizar uma análise competitiva entre as tecnologias *Blockchain* e *Tangle*, visando ao desenvolvimento de possíveis projetos de aplicações IoT. Para este fim, é feito inicialmente um estudo teórico das estruturas de dados e algoritmos usados para manipular informações sob cada tecnologia, buscando avaliar as complexidades de tempo com respeito a operações de *consulta* e *adição* de transações. Em seguida, por meio de simulação, diferentes cenários de operação são analisados para mensurar o nível de segurança sistêmica, em termos da inviolabilidade das informações registradas.

Ante o objetivo anunciado, este artigo tem, como principal contribuição, a disponibilização de um único texto que provê um entendimento comparativo das tecnologias *Blockchain* e *Tangle* e, ainda, oferece uma análise competitiva entre as mesmas. Essa contribuição se constitui valiosa por prover subsídios relevantes para o desenvolvimento de possíveis projetos de aplicações IoT.

A organização do restante deste artigo é descrita a seguir. A Seção 2 apresenta as tecnologias *Blockchain* e *Tangle*. Os trabalhos relacionados são comentados na Seção 3. Na Seção 4, é realizado o estudo teórico das estruturas de dados e algoritmos utilizados nas tecnologias supracitadas. A Seção 5 analisa a segurança do sistema, com foco na garantia da inviolabilidade das informações. Por fim, conclusões gerais e trabalhos futuros aparecem na Seção 6.

2. Fundamentos

2.1 Tecnologia *Blockchain*

Como mencionado, as transações dos clientes são agrupadas em blocos interligados formando uma *lista encadeada*, i.e., a *blockchain*. Cada bloco está ligado a apenas um bloco anterior a ele. *Mineração* é o processo matemático que precisa ser executado para que o bloco seja adicionado à *blockchain*. Esse processo é executado por um *peer*, denominado de *minerador*, ou por um grupo de *peers*, denominado de *mining pool* [Garay; Kiayias; e Leonardos 2015; Silva e Rodrigues 2016].

Os *mineradores* ou *mining pools* são também responsáveis por coletar as transações transmitidas em *broadcast* na rede e agrupá-las em blocos. Cada vez que um bloco é *minerado*, o correspondente *minerador* ou *mining pool* recebe uma recompensa, que é um incentivo financeiro para que a *mineração* seja continuamente realizada.

Na prática, a *mineração* consiste em tentativas sucessivas para determinar um valor de *nonce* que resolve o *hash* criptográfico do bloco de transações, atendendo a um certo critério. A dificuldade desse critério é ajustada com base na frequência com que os blocos são adicionados à *blockchain*. A determinação do *nonce* é a chamada *prova de trabalho* (do inglês, *proof of work*) de que o bloco foi verificado e pode ser adicionado. O algoritmo de *hash* criptográfico utilizado é o SHA256 [Gilbert e Handschuh 2004].

Além das transações, cada bloco possui um cabeçalho com *metadados* [Antonopoulos 2017]. A identificação de cada bloco é feita pelo *hash* criptográfico de seu cabeçalho. Como informado, cada bloco faz referência a apenas um bloco anterior, chamado de bloco *pai*. Essa referência é feita com o *hash* do cabeçalho do bloco *pai*. Ou seja, cada bloco contém o *hash* do cabeçalho 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 do 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. Esse efeito cascata constitui a base da segurança do sistema com relação à inviolabilidade das informações registradas, conforme explorado mais adiante na Seção 5.

A *Blockchain* teve sua origem em 2008, junto com a proposta da criptomoeda *Bitcoin* [Nakamoto 2008]. Apesar da concepção original para pagamentos eletrônicos, a *Blockchain* foi rapidamente identificada como de aplicabilidade mais abrangente [Dorri; Kanhere e Jurdak 2017; Huckle et al. 2016]. Por exemplo, na implementação de aplicações de controle de votos, de recursos em gestão orçamentária, de direitos de propriedades e, ainda, de comunicação em sistemas distribuídos, incluindo especialmente IoT [Dorri; Kanhere; e Jurdak 2017; Li e Liao 2017].

2.2 Tecnologia *Tangle*

A tecnologia *Tangle* surgiu para implementação da criptomoeda *iota* [IOTA 2017], então projetada em 2014 para a indústria de IoT. Sua aplicabilidade, no entanto, não se restringe a pagamentos eletrônicos, mas envolve um escopo bem mais abrangente, tal como ocorre com a tecnologia *Blockchain*.

Como já mencionado, a tecnologia *Tangle* utiliza a estrutura DAG para armazenamento das transações. A função de *hash* criptográfico originalmente utilizada nessa tecnologia foi a Curl, tendo sido substituída mais tarde pela Keccak-384 [Sonstebo 2017], também conhecida como SHA-3 (*Secure Hash Algorithm* 3) [Chandrana e Manuelb 2016]. Os defensores dessa tecnologia professam que ela é a evolução da *Blockchain*, levando-se em conta os três aspectos comentados a seguir.

O primeiro aspecto diz respeito a não existência de recompensas na *Tangle*. Na *Blockchain*, a recompensa pode inviabilizar a execução de transações típicas de IoT. Isso porque essas transações tendem a ser de pequenos valores. Daí, a existência de uma recompensa de valor maior que o montante que está sendo transferido na transação não seria lógico. Simplesmente eliminar a recompensa na *Blockchain* tampouco seria aceitável, pois ela é o incentivo para a contínua *mineração* de blocos.

O segundo aspecto relaciona-se à homogeneidade funcional dos dispositivos IoT na tecnologia *Tangle*. Isto é, todos os participantes desempenham a mesma função. Isso não é observado na tecnologia *Blockchain*, pois há dois tipos distintos de participantes: aqueles que emitem transações e aqueles que aprovam transações.

Essa discriminação entre dispositivos pode levar a conflitos na *mineração* de blocos em virtude das recompensas previstas. Nesse caso, uma disputa entre *mineradores* ou *mining pools* pode transformar um paradigma de validação concebido originalmente descentralizado em um paradigma centralizado, constituído por *mineradores* ou *mining pools* de maior capacidade de processamento.

O terceiro aspecto refere-se à maior simplicidade da *Tangle* para *adição* de novas transações ao sistema. Essa *adição* envolve a resolução de uma *prova de trabalho* e a validação de duas transações pré-existentes. Ocorre que a *prova de trabalho* da *Tangle*, por considerar apenas uma transação, demanda um esforço computacional bem menor que àquele da *mineração* na *Blockchain*, que envolve múltiplas transações. Por sua vez, a validação de uma transação pré-existente é simples e pode ser realizada em tempo constante. A *adição* de transações é detalhadamente analisada na Subseção 4.2.

Para terminar esta subseção, cabe mencionar que as validações são representadas por arestas direcionadas no DAG, que partem da nova transação em direção às transações pré-existentes. Se não existe uma aresta direcionada entre uma transação *A* e uma transação *B*, mas há um caminho direcionado entre elas, diz-se então que *A* indiretamente aprova *B*. Quanto maior é o número de validações que uma transação possui, mais confiável é a validade dessa transação. Por definição, há ainda a transação denominada *gênese*, que representa a primeira transação do sistema e é validada direta ou indiretamente por todas as outras transações do DAG.

3. Trabalhos Relacionados

A tecnologia *Blockchain* possui um considerável número de trabalhos na literatura. O mesmo não pode ser afirmado para tecnologia *Tangle*, a qual é mais alvo de *blogs* e de outros veículos informais. Estudos comparativos entre essas duas tecnologias tampouco são fáceis de encontrar. Neste contexto, esta seção discorre sobre alguns dos mais recentes e relevantes trabalhos da literatura que contribuem para o objetivo deste artigo, buscando ofertar uma visão do estado da arte desta área de pesquisa.

Opara e Soluade (2015) e Singh (2014) analisam ataques cibernéticos a plataformas baseadas na tecnologia *Blockchain*. Busca-se comprovar a inadequação das infraestruturas de redes legadas, identificar e caracterizar os atores das ameaças, suas capacidades, motivações e propósitos. São também apresentadas as melhores práticas para enfrentar essa realidade e informados cenários de aplicabilidade dessa tecnologia.

Eyal e Sirer (2013) e Courtois e Bahack (2014) estudam a vulnerabilidade da *Blockchain* sob a condição de ataques de *mineradores desonestos*, os quais conseguem receber recompensas maiores que aquelas de *mineradores honestos*. Usando modelagem analítica e simulação, são alcançadas evidências de que, sem regulações da *mineração* por *mining pools*, o sistema pode deixar de ser verificável de maneira descentralizada.

Karame, Androulaki e Capkun (2012), Rosenfeld (2014) e Gervais et al. (2016) estudam ataques de *double-spending* em sistemas baseados na *Blockchain*. Esses ataques objetivam permitir que um mesmo recurso seja usado em diferentes transações. A partir de modelagem analítica e simulação, são obtidos resultados que permitem inferir a probabilidade de ocorrência desses ataques e propor contramedidas de defesa.

Sompolinsky e Zohar (2015) e Karame (2016) discutem sobre a 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 aspectos. Os resultados permitem conjecturar sobre uma nova geração de aplicações alicerçadas pela *Blockchain*.

Dorri, Kanhere e Jurdak (2017) propõem uma versão mais otimizada da *Blockchain* para o desenvolvimento de aplicações IoT. Essa versão tenciona diminuir o custo de processamento criptográfico, otimizar a banda e diminuir as latências de transmissão. Emprega-se um paradigma de camadas funcionais, permitindo que bases de dados locais centralizadas se comuniquem com bases públicas distribuídas. Os blocos são *minerados* localmente e adicionados à *blockchain* de sua camada, que posteriormente se torna consistente com à *blockchain* de camadas públicas superiores.

Danzi et al. (2017) descrevem arquiteturas gerais e protocolos de sincronização para nós de clientes IoT usando tecnologia *Blockchain*. Consideram-se diferentes capacidades de transmissão da rede, níveis de segurança e características intrínsecas de ambientes *wireless*. Nesse trabalho são modelados e caracterizados analiticamente os tráfegos gerados pelos protocolos de sincronização e, usando simulações, são avaliados o consumo de energia e o custo de banda para sincronização da *blockchain*.

Rodrigues (2017b) analisa a eficiência e a segurança da tecnologia *Blockchain*. Usando modelagem analítica e simulação, são realizados experimentos em diferentes cenários de operação. Os resultados apontam para uma promissora eficiência na manipulação de dados e uma atrativa segurança quanto à inviolabilidade da informação.

Li e Liao (2017) apresentam o protocolo GHOST, o qual propõe a utilização de uma estrutura de dados baseada em árvores em substituição à *blockchain*. Utilizando principalmente modelagem analítica, os autores verificam a possibilidade da redução do tempo de confirmação das transações e a otimização do nível de segurança sistêmica. Esse trabalho pode ser visto como um dos precursores da tecnologia *Tangle*.

Além do protocolo GHOST, há ainda algumas outras propostas que podem ser também consideradas precursoras da tecnologia *Tangle* como, por exemplo, os trabalhos

de Lerner (2015), Sompolinsky e Zohar (2013) e Lewenberg, Sompolinsky e Zohar (2015). Este ultimo, em especial, propõe um modelo de criptomoeda também baseada em DAG. No entanto, o modelo difere da *Tangle* nos seguintes dois pontos principais: as unidades de informação são blocos de dados e não transações individuais e, ainda, o conceito de recompensa para validação de transações é aplicado.

Por fim, Popov (2017) analisa os fundamentos teóricos matemáticos da criptomoeda *iota*. Esses fundamentos constituem a descrição e a explicação da tecnologia *Tangle*, incluindo discussões sobre o número de validações que as transações devem possuir para sua aceitação, a estabilidade do sistema, as possibilidades de ataques e vulnerabilidades, e os algoritmos de seleção de transações pré-existentes. Em que pese a existência apenas de abordagem teórica, esse trabalho constitui uma das mais importantes referências da literatura sobre a análise da tecnologia *Tangle*.

Ante o exposto, a contribuição e a diferenciação deste artigo se revelam pelo relativo ineditismo de uma análise competitiva entre as tecnologias *Blockchain* e *Tangle* com foco em IoT, contemplando a eficiência das estruturas de dados e algoritmos, sob o viés teórico da complexidade de tempo, além da avaliação da segurança quanto à inviolabilidade das informações, sob o viés da simulação.

4. Eficiência: Complexidade de Tempo

Esta seção avalia a eficiência das tecnologias *Blockchain* e *Tangle*, considerando a seguinte questão de pesquisa: quais as complexidades de tempo para a execução de operações de *adição* de transações (ou blocos) e *consulta* de transações?

4.1. Tecnologia *Blockchain*

Cada bloco *minerado* é propagado em *broadcast* na rede. Em seguida, aguarda-se que *nós completos*, *peers* da rede que mantêm uma cópia local de toda a cadeia de blocos, realizem a sua *adição*. A cópia da cadeia de blocos local de cada *nó completo* é atualizada à medida que novos blocos são *minerados* e propagados. A convergência das cópias locais dos *nós completos* ocorre conforme as informações são disseminadas.

Para *adição* de um novo bloco, B_{novo}, um *nó completo*, *P*, inicialmente examina o cabeçalho desse bloco para conhecer o valor do campo *hash do bloco anterior*, que é a referência para o bloco *pai*. 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.

O nó *P* então adiciona B_{novo} à sua cadeia somente se $h(B_m) = h(B_{anterior})$ e se a *prova de trabalho* tiver sido realizada, ou seja, se o valor do *nonce* tiver sido determinado. Embora a computação do valor de *nonce* seja intrincada, a sua verificação é de complexidade constante, pois basta que o valor de *nonce* seja testado. Assim, a complexidade de tempo de *adição* de um bloco é de apenas $O(1)$. No entanto, é preciso ressaltar que, para que a *adição* ocorra, é necessário que o bloco já tenha sido *minerado*. Ou seja, existe um custo computacional anterior que, na prática, não pode ser ignorado.

Para *consultar* sobre a existência de uma transação em um bloco no sistema são usados nós especiais da rede, chamados de nós de Verificação Simplificada de Transação (VST). Esses nós utilizam o chamado *caminho de Merkle* ou *caminho de autenticação* [Antonopoulos 2017], conforme explicado a seguir.

Seja *Q* um nó VST que deseja consultar a existência da transação *s*, tendo como uma das partes envolvidas um cliente de endereço lógico *E*. Esse 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 blocos com transações que se relacionam ao endereço *E* devem ser informados a ele. Assim, quando um *peer* da rede vê um bloco com uma transação de endereço *E*, ele então informa sobre aquele bloco ao *nó Q*. Para tanto, ele envia uma mensagem especial, denominada *mensagem de bloco de Merkle* (MBM). Essa mensagem contém o *cabeçalho do bloco* e o *caminho de Merkle* que liga a transação *s* à *raiz da árvore de Merkle* do bloco.

A *árvore de Merkle* [Berman; Karpinski; e Nekrich 2007], 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 de mesmo nível (i.e., vértices *irmãos*), a partir dos vértices *folhas* em direção à raiz, até que se obtenha um único *hash*, o qual constitui o *resumo* de todas as transações existentes no bloco. Esse *resumo* é guardado em um campo do cabeçalho do bloco, denominado *raiz da árvore de Merkle*. No caso de o número de transações ser ímpar, a última transação é duplicada, obtendo sempre uma *árvore binária cheia* [Cormen; Leiserson; e Rivest, 2009].

Seja então a sequência de vértices (v_1, v_2, ..., v_l) o *caminho de Merkle* informado ao nó *Q*. Cada vértice da sequência está em um nível imediatamente inferior ao anterior na *árvore de Merkle*. Mais especificamente, 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. Além disso, o vértice v_1 é *irmão* do vértice *folha* que armazena o *hash* da transação *s*.

A *consulta* da transação *s* consiste então em computar recursivamente o *hash* de vértices *irmãos* correspondentes na árvore em diferentes níveis sucessivos, a partir do vértice *folha* v_1 em direção à raiz. Ao atingir-se o segundo nível, é então verificado se o *hash* computado é igual ao valor da raiz, obtido do exame do *cabeçalho do bloco*, que é também informado na MBM. Se ocorre a identidade, então a transação *s* de fato existe.

Essa operação de *consulta* tem complexidade de tempo $O(\lfloor log_2(2n-1) \rfloor)$, onde *n* é o número total de transações existentes no bloco. Esse resultado é bastante eficiente por ser logarítmico em função da quantidade de dados. Isso ocorre porque a *árvore de Merkle* construída é sempre uma *árvore binária cheia* e, portanto, goza da propriedade de altura logarítmica [Cormen; Leiserson; e Rivest 2009].

4.2. Tecnologia *Tangle*

Como mencionado, cada nó do DAG armazena uma única transação. Considerando nós VST, a *consulta* de uma transação *s* pode então ser realizada em tempo constante *O*(1), dado que se saiba qual o nó que armazena a transação *s*. Em termos de complexidade de tempo, tem-se então que *Tangle* é mais eficiente que *Blockchain* para realizar *consultas*, pois a complexidade desta última é maior, sendo igual a $O(\lfloor log_2(2n-1) \rfloor)$, onde *n* é o número total de transações existentes no bloco (vide Subseção 4.1).

Com relação à *adição* de uma transação *s* na *Tangle*, tem-se a análise baseada nos dois pontos a seguir. Primeiro, é preciso realizar um processamento matemático para resolução de um *hash* criptográfico, i.e., obtenção do valor *nonce*. Esse *hash*

criptográfico envolve metadados relacionados à identificação da transação *s*, além da transação *s* propriamente dita. Seja C_T a complexidade dessa operação.

Segundo, é preciso validar duas transações pré-existentes no sistema. O processo de validação é simples, consistindo apenas em verificar se há conflitos entre as transações e se a *prova de trabalho* foi realizada pelo respectivo nó da transação a ser validada. Essa verificação pode ser feita em $O(1)$. Resulta então que a complexidade final de *adição* é $C_T + O(1) = C_T$, ou seja, é determinada pela complexidade da realização da *prova de trabalho*.

Para fins de comparação, tem-se que a *adição* de um bloco é feita em $O(1)$ na tecnologia *Blockchain*. Na prática há, contudo, a necessidade da *mineração* prévia do bloco a ser adicionado (vide Subseção 4.1). Seja o custo dessa *mineração* igual a C_B. O custo final de adição na *Blockchain* é, portanto, $C_B + O(1) = C_B$.

Tendo em vista que a *mineração* de um bloco na *Blockchain* envolve um *hash* criptográfico com metadados relacionados à identificação de todo o bloco, além de todas as transações nele existentes, tem-se que $C_B > C_T$. Ou seja, a tecnologia *Tangle* resulta mais eficiente que a tecnologia *Blockchain* quanto à complexidade de tempo da *adição*. Essa conclusão é comparativa e independe dos valores absolutos de C_B e C_T.

Como observação final, deve-se mencionar que cada *adição* na *Blockchain* considera *n* transações de uma só vez, e que cada adição na *Tangle* se refere a apenas uma única transação. Mesmo assim, é possível conjecturar que $C_B/n > C_T$ em virtude da natureza do cálculo de *hash* envolvido [Cormen; Leiserson; e Rivest 2009].

5. Segurança: Inviolabilidade da Informação

Esta seção avalia a segurança das tecnologias *Blockchain* e *Tangle*, considerando a seguinte questão de pesquisa: qual o tempo de registro de uma transação no sistema para garantir um nível de segurança aceitável, significando inviolabilidade da informação?

5.1. Descrição dos Cenários de Investigação

Para entender a questão de pesquisa na *Blockchain*, considere o cenário descrito a seguir [Nakamoto 2008]. Quando dois *mineradores* distintos enviam diferentes versões daquele que seria o próximo bloco da *blockchain*, os nós *completos* vão receber uma ou outra versão primeiramente. Cada nó *completo* considera a primeira versão recebida, mas vai também criar uma ramificação em sua *blockchain* local para adicionar a outra versão do bloco recebida posteriormente. Passam a existir duas ramificações: uma para a primeira versão e a outra para a segunda versão.

A seleção 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 então pertencer à ramificação mais longa.

A competição entre duas ramificações pode ser modelada como uma *corrida* para *adição* de blocos à *blockchain*, como explicado a seguir [Nakamoto 2008]. Um *minerador desonesto* deseja realizar uma fraude, substituindo um bloco *B* legítimo adicionado por um *minerador honesto*. Para isso, o *minerador desonesto* espera até que *z* blocos subsequentes a *B* sejam adicionados à *blockchain*. Neste instante, o *minerador*

honesto está *minerando* o bloco *z*+1. Explica-se que essa espera de *z* blocos é o tempo que o cliente da aplicação tolera esperar para ter sua transação confirmada no sistema.

O *minerador desonesto* então inicia a *mineração* explícita de blocos e tenta ser mais rápido que o *minerador honesto* para compensar a desvantagem de *z* blocos, fazendo sua ramificação se tornar mais longa que a ramificação contendo o bloco legítimo *B*. Um detalhe importante é que o *minerador desonesto minera* blocos secretamente em taxa compatível com sua capacidade para que, ao iniciar a competição explícita com o *minerador honesto*, já possua blocos *minerados* para inserir no sistema.

Iniciada a competição, tem-se o seguinte modelo parametrizado [Nakamoto 2008]: com probabilidade *p*, o próximo bloco *minerado* da *blockchain* é do *minerador honesto*, e com probabilidade $q = (1 - p)$, o próximo bloco *minerado* é do *minerador desonesto*. A chegada de blocos *minerados* à *blockchain* segue um processo de Poisson com parâmetro μ. Esse modelo permite estimar a probabilidade de fraude para um dado valor de *z*. Quanto maior é *z*, menor é a probabilidade de fraude e maior é o tempo de espera (i.e., tempo de registro da transação). Ou seja, há um compromisso entre a segurança e o tempo de espera. Esse modelo é resolvido por simulação na Subseção 5.2.

O cenário de investigação da *Tangle* é análogo ao da *Blockchain* [Popov 2017]. A competição, porém, é pelo número de validações da transação. O nó fraudador deseja substituir uma transação legítima *T* do DAG. Ele então espera até que a transação *T* atinja *v* validações, que reflete o tempo que o cliente da aplicação tolera esperar para ter a transação confirmada. Em seguida, ele insere uma transação fraudulenta *F* no DAG para tentar substituir a transação *T*. Para isso, ele passa a criar novas transações que só validam a transação *F*. O objetivo é que, eventualmente, a transação *F* passe a ter mais validações que a transação *T*. Como na *Blockchain*, o fraudador na *Tangle* trabalha secretamente em taxa compatível com sua capacidade para que, ao iniciar a competição explícita, ele já possua transações prontas para inserir no DAG.

O modelo parametrizado a seguir é usado para análise da *Tangle* [Popov 2017]: a chegada de transações ao DAG segue um processo de Poisson com parâmetro β; a fração de transações que valida a transação legítima *T* é *k*% do total de transações, e a fração de transações que valida a transação fraudulenta *F* é *r*% do total de transações, sendo $k + r = 100$. Esse modelo permite calcular a probabilidade de fraude para um dado valor de *v*. Quanto maior é *v*, menor é a probabilidade de fraude e maior é o tempo de espera. Ou seja, como na *Blockchain*, há um compromisso entre a segurança e o tempo de espera. Esse modelo é resolvido por simulação na Subseção 5.2.

5.2. Ambiente de Simulação e Modelos

A simulação usa o Tangram-II [De Souza e Silva; Figueiredo; e Leão 2009]. Tangram-II é um ambiente de modelagem de sistemas computacionais concebido na Universidade Federal do Rio de Janeiro, com a participação da Universidade da Califórnia em Los Angeles nos EUA. Os modelos são definidos em termos de objetos que interagem entre si por mensagens, podendo ser resolvidos analiticamente ou via simulação.

O modelo de simulação da *Blockchain* é explicado a seguir. São criados 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 processo de Poisson tem seu parâmetro μ igual a um bloco *minerado* a cada 10 minutos,

conforme definição da *Blockchain* [Rodrigues 2017a]. O objeto Obj_2 emula a *blockchain* do sistema, permitindo monitorar a diferença de blocos entre as ramificações *honestas* e *desonestas*. O sistema é considerado no estado estacionário.

O modelo de simulação da tecnologia *Tangle* é explicado a seguir. São criados três objetos: Obj_1, Obj_2 e Obj_3. Os dois primeiros objetos emulam os nós *honestos* e *desonestos*, respectivamente, os quais têm individualmente a mesma capacidade de processamento. Os nós *honestos* coletivamente contribuem com k% das transações totais do sistema, e os *desonestos* com r% das transações totais, sendo $k + r = 100$. O processo de Poisson tem seu parâmetro β igual a 500 transações a cada 10 minutos. Esse valor permite uma comparação mais equânime com a tecnologia *Blockchain*, pois o número médio de transações por bloco é 500 [Antonopoulos 2017]. O objeto Obj_3 emula o DAG, permitindo monitorar a diferença de validações entre as transações *honestas* e *desonestas*. O sistema é considerado no estado estacionário.

Cabe destacar que 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. Estes resultados e as correspondentes análises estão na subseção seguinte. Por restrição de espaço, menciona-se que apenas os resultados mais relevantes para suporte às conclusões atingidas são apresentados no que se segue.

5.3. Resultados e Análises

Os resultados obtidos para *Blockchain* estão nas Figuras 1 e 2. A Figura 1 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 análise para $p = 0{,}8$.

A partir dos resultados dessa figura, conclui-se que maiores valores de p e z tendem a fornecer 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 $z = 6$ (Cenário S_6). Isso significa que a diferença de tamanho 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 está definida. Assim, a fraude tem menor chance de ocorrer, pois apenas existiria se as ramificações se igualassem, compensando a desvantagem inicial de z blocos.

Por outro lado, aumentos arbitrários de p e z, sem 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, como visto nos resultados dos Cenários S_4 e S_5.

A Figura 2, por sua vez, apresenta resultados para os mesmos seis cenários anteriores. Porém, agora são destacados valores médios das diferenças de tamanhos entre as ramificações. Confirma-se novamente que maiores valores de p e z tendem a ocasionar uma maior segurança. Por exemplo, a diferença registrada no Cenário S_6 é quase cinco vezes maior que aquela do Cenário S_1. Porém, como antes, também se

confirma que aumentos arbitrários de *p* e *z* podem resultar inócuos. Por exemplo, nos Cenários S_5 e S_6, dobrar o valor de *z*, de 3 para 6, não faz a diferença média ser dobrada. Enfim, como diretriz de projeto, sugere-se admitir valores de *p* acima de 0,5 e valores de *z* que devem observar a tolerância do cliente da aplicação para espera de confirmação.

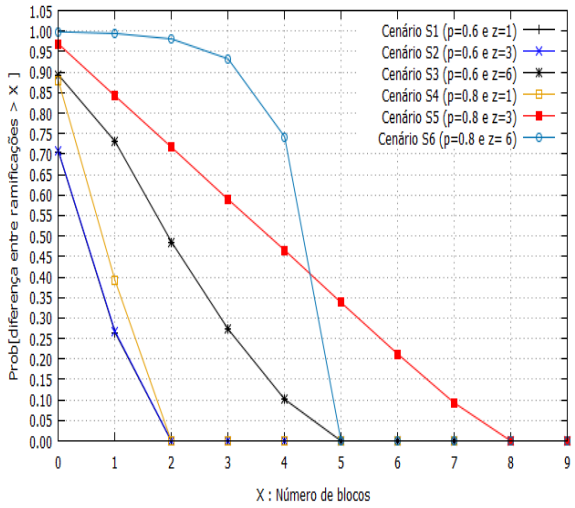

Figura 1. Diferença entre ramificações.

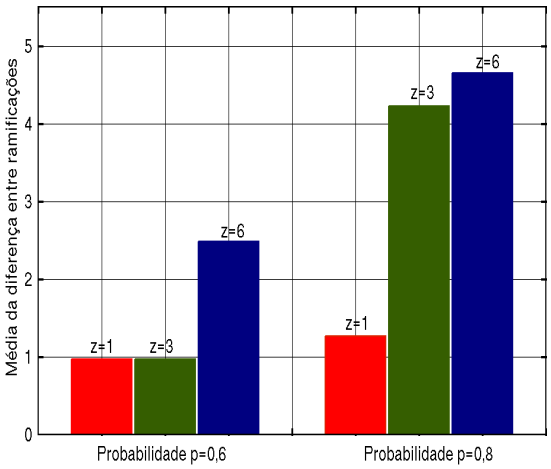

Figura 2. Média da diferença.

As Figuras 3 e 4, por sua vez, trazem os resultados para *Tangle*. São considerados seis cenários de análise: S_7, S_8, S_9, S_{10}, S_{11} e S_{12}. Considere inicialmente a Figura 3. Nos três primeiros cenários, tem-se k = 60% e v sendo variado para determinar a probabilidade de a diferença de validações entre as transações *T* e *F* exceder um valor *Y*. Nos três cenários seguintes, faz-se a mesma análise para k = 80%. Considere agora a Figura 4. Nesta figura são destacados os valores médios das diferenças de validações entre as transações *T* e *F* para os mesmos seis cenários anteriores.

A partir dos resultados da Figura 3, observa-se que maiores valores de k e v redundam em maior segurança. Isso também ocorre na *Blockchain* ao aumentar-se os valores de *p* e *z*. Entretanto, na *Tangle* esse aumento de segurança é mais contundente. Por exemplo, vê-se que as curvas de probabilidade têm valor inicial mais próximo ao valor unitário que aquelas da Figura 1. Além disso, mesmo aumentos independentes de v ou de k, sem avaliação prévia, são capazes de aumentar as probabilidades. Essa independência entre parâmetros não foi verificada na *Blockchain*. Ou seja, o nível de segurança é mais facilmente otimizado na *Tangle*.

Na Figura 4, por sua vez, tem-se que, para k = 60% ou 80%, ao dobrar-se ou triplicar-se o valor de v, há um aumento de mesma magnitude nos correspondentes valores médios das diferenças de validações entre as transações *T* e *F*, implicando maior segurança. Essa correspondência é diferente daquela observada na *Blockchain*. Em síntese, a tecnologia *Tangle* é uma solução mais atrativa que *Blockchain*. Por fim, como diretriz de projeto, sugere-se admitir valores de k acima de 50% e valores de v que devem observar a tolerância do cliente da aplicação para espera de confirmação.

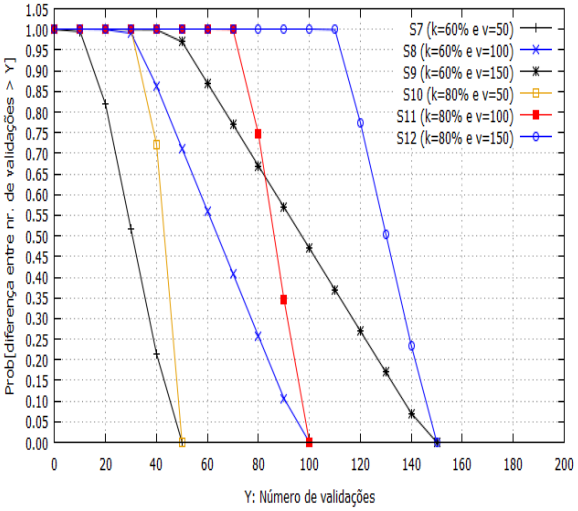

Figura 3. Diferença entre validações.

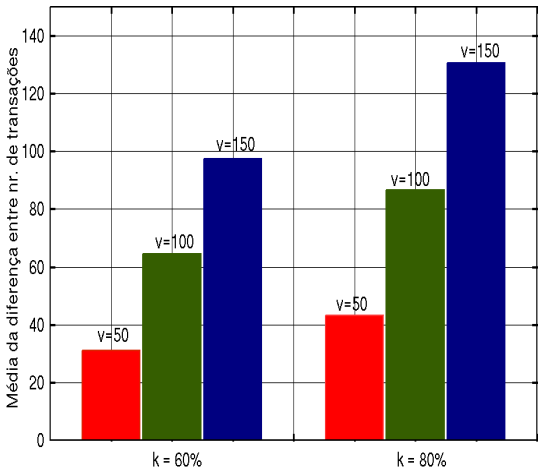

Figura 4. Média da diferença.

6. Conclusões e Trabalhos Futuros

Este artigo realizou uma análise competitiva entre as tecnologias *Blockchain* e *Tangle*. Para tanto, inicialmente foi feito um estudo teórico das estruturas de dados e dos algoritmos para manipulação das informações. Na sequência, por meio de simulação, foram examinados diferentes cenários para mensurar o nível de segurança do sistema.

O estudo teórico mostrou principalmente que o uso da *árvore de Merkle* para *Blockchain* é uma solução teórica efetiva, pois garante a otimização da complexidade de tempo para execução de *consultas* e *adições* de transações. Essa efetividade se mostra promissora na prática conforme sejam maiores os blocos de dados. O contraponto é que o processamento matemático da *prova de trabalho* demanda significativa capacidade de processamento, justamente por lidar com o bloco inteiro de informações.

Por sua vez, o uso de DAG para *Tangle* teve seu destaque pela sua simplicidade de implementação: uma transação por vértice. Isso possibilita uma execução prática otimizada de *consultas* e *adições* de transações. Além disso, a realização da *prova de trabalho* é menos onerosa que aquela da *Blockchain* por lidar com menos informações. Dado que o processamento típico em cenários de IoT envolve poucos dados, a *Tangle* termina sendo então uma tecnologia mais atrativa na prática que a *Blockchain*.

Em relação à segurança, tem-se precipuamente o que se segue. Os experimentos permitiram ver que as tecnologias *Blockchain* e *Tangle* apresentam satisfatório nível de segurança no que se refere à inviolabilidade das informações, podendo ser consideradas para projetos reais de aplicações IoT. No entanto, sob o viés da competição, a *Tangle* foi entendida aqui como uma solução mais atrativa. Essa conclusão adveio do fato de ela ser uma solução que mais objetivamente e facilmente responde à mudança de configuração de parâmetros para alcançar o nível de segurança desejado.

Como trabalhos futuros, tem-se: (i) análise de cenários mais complexos, com diferentes valores de validação de transações na *Tangle* e com disputas entre múltiplas ramificações na *Blockchain*; (ii) desenvolvimento de modelos analíticos e novos modelos de simulação com distribuições diferentes de Poisson; e (iii) análise competitiva dos algoritmos de consenso *proof of work* e *proof of stake* [Li e Liao 2017].

Referências

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

Berman, P.; Karpinski, M.; and Nekrich, Y. (2007). Optimal trade-off for Merkle tree traversal. Theoretical Computer Science, v. 372, n. 1, pp. 26-36.

Chandrana, N. R. and Manuelb, E. M. (2016). Performance Analysis of Modified SHA-3. Procedia Technology, v. 24, pp. 904-910.

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

Courtois, N. T. and Bahack, L. (2014). On subversive miner strategies and block withholding attacks in bitcoin digital currency. CoRR, abs/1402.1718. Disponível em: https://arxiv.org/abs/1402.1718. Acessado em: 8 de dezembro de 2017.

Danzi, P. et al. (2017). Analysis of the Communication Traffic for Blockchain Synchronization of IoT Devices. CoRR, abs/1711.00540. Disponível em: https://arxiv.org/abs/1711.00540v1. Acessado em: 8 de dezembro, 2017.

De Souza e Silva, E.; Figueiredo, R.; and Leão, R. (2009). The TANGRAM-II integrated modeling environment for computer systems and networks. ACM SIGMETRICS Performance Evaluation Review, v. 36, n. 4, pp. 64-69.

Dorri, A.; Kanhere, S.; and Jurdak, R. (2017). Towards an Optimized Blockchain for IoT. In: International Conference on Internet-of-Things Design and Implementation (IoTDI´17), Pittsburgh, PA, USA.

Eyal, I. and Sirer, E. G. (2014). Majority is not enough: bitcoin mining is vulnerable. In: International conference on financial cryptography and data security, pp. 436-452. Springer, Berlin, Heidelberg.

Garay, J.; Kiayias, A.; Leonardos, N. (2015). The bitcoin backbone protocol: analysis and applications. LNCS, Springer, Berlin, Heidelberg, v. 9057, pp. 281-310.

Gaur, A. et al. (2015). Smart city architecture and its applications based on IoT. Procedia Computer Science, v. 52, p. 1089-1094.

Gervais, A. et al. (2016). On the security and performance of proof of work blockchains. In: ACM Conference on Computer and Communications Security, Vienna, Austria.

Gilbert, H. and Handschuh, H. (2004). Security analysis of SHA-256 and sisters. LNCS, Springer, Berlin, Heidelberg, v. 3006, pp. 175-193.

Hassanalieragh, M. et al. (2015). Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges. In: IEEE International Conference on Services Computing, p. 285-292.

Huckle, S. et al. (2016). Internet of Things, blockchain and shared economy applications. Procedia Computer Science, v. 98, pp. 461-466.

IOTA. (2017). What is IOTA? Disponível em: https://iota.readme.io/docs. Acessado em: 8 de dezembro de 2017.

Karame, G. O. (2016). On the security and scalability of bitcoin´s blockchain. In: ACM Conference on Computer and Communications Security (CCS´16), Vienna, Austria.

Karame, G. O.; Androulaki, E.; Capkun, S. (2012). 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.

Lerner, S. D. (2015). DagCoin: a cryptocurrency without blocks. Disponível em: https://bitslog.files.wordpress.com/2015/09/dagcoin-v41.pdf. Acessado em: 6 de dezembro de 2017.

Lewenberg, Y.; Sompolinsky, Y.; and Zohar, A. (2015). Inclusive Block Chain Protocols. Disponível em: http://fc15.ifca.ai/preproceedings/paper_101.pdf. Acessado em: 8 de dezembro de 2017.

Li, I-C and Liao, T-C. (2017). A Survey of blockchain security issues and challenges. International Journal of Network Security, v. 19, n. 5, pp.653-659, 2017.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Disponível em: https://bitcoin.org/bitcoin.pdf . Acessado em: 8 de dezembro de 2017.

Opara, E. U. and Soluade, O. A. (2015). Straddling the next cyber frontier: the empirical analysis on network security, exploits, and vulnerabilities. International Journal of Electronics and Information Engineering, v. 3, n. 1, pp. 10-18.

Popov, S. (2017). The Tangle – Version 1.3. Disponível em: https://iota.org/IOTA_Whitepaper.pdf. Acessado em: 6 de dezembro, 2017.

Rodrigues, C. K. S. (2017a). Sistema Bitcoin: uma análise da segurança das transações. Revista Brasileira de Sistemas de Informação, v. 10, n. 3, pp. 5-23.

Rodrigues, C. K. S. (2017b). Uma análise simples de eficiência e segurança da tecnologia Blockchain. Revista de Sistemas e Computação, v. 7, n. 2, pp. 147-162.

Rosenfeld, M. (2014). Analysis of hasrate-based double spending. CoRR, abs/1402.2009. Disponível em: https://arxiv.org/abs/1402.2009 . Acessado em: 8 de dezembro de 2017.

Silva, G. A. e Rodrigues, C. K. S. (2016). 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.

Singh, J. (2014). Cyber attacks in cloud computing: a case study. International Journal of Electronics and Information Engineering, v. 1, n. 2, pp. 78-87.

Sompolinsky, Y. and Zohar, A. (2013). Accelerating Bitcoin's Transaction Processing − Fast Money Grows on Trees, Not Chains. Cryptology ePrint Archive, Report 2013/881. Disponível em: https://eprint.iacr.org/2013/881. Acessado em: 8 de dezembro de 2017.

Sompolinsky, Y. and Zohar, A. (2015). Secure high-rate transaction processing in bitcoin. LNCS, Springer, Berlin, Heidelberg, v. 8975, pp. 507-527.

Sonstebo, D. (2017). Curl disclosure, beyond the headline. Disponível em: https://blog.iota.org/curl-disclosure-beyond-the-headline-1814048d08ef. Acessado em: 6 de dezembro de 2017.