SBRC 2019

Gramado | RS

XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos

6 a 10 de maio

Gramado | RS

ANAIS DO WBlockchain 2019

sbrc2019.sbc.org.br

SBRC 2019

Gramado | RS

XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos

6 a 10 de maio

Gramado | RS

ANAIS DO WBlockchain 2019

sbrc2019.sbc.org.br

EDITORA
Sociedade Brasileira de Computação (SBC)

ORGANIZAÇÃO
Alberto Egon Schaeffer Filho (UFRGS)
Weverton Luis da Costa Cordeiro (UFRGS)
Stênio Fernandes (UFPE)
Artur Ziviani (LNCC)
Alex Vieira (UFJF)

REALIZAÇÃO
Sociedade Brasileira de Computação (SBC)
Instituto de Informática - Universidade Federal do Rio Grande do Sul (UFRGS)
Laboratório Nacional de Redes de Computadores (LARC)

Copyright ©2019 da Sociedade Brasileira de Computação
Todos os direitos reservados

Capa: Dórian Fogliatto
Produção Editorial: Carlos Raniery P. dos Santos (UFSM)

Cópias Adicionais:
Sociedade Brasileira de Computação (SBC)
Av. Bento Gonçalves, 9500- Setor 4 - Prédio 43.412 - Sala 219
Bairro Agronomia - CEP 91.509-900 - Porto Alegre - RS
Fone: (51) 3308-6835
E-mail: sbc@sbc.org.br

II Workshop Blockchain: Teoria, Tecnologias e Aplicações (II: 2019: Gramado, RS).

Anais / II Workshop Blockchain: Teoria, Tecnologias e Aplicações - WBlockchain; organizado por Alberto Egon Schaeffer Filho, Weverton Luis da Costa Cordeiro, Stênio Fernandes, Artur Ziviani, Alex Vieira - Porto Alegre: SBC, 2019

113 p. il. 21 cm.
Vários autores
Inclui bibliografias

1. Redes de Computadores. 2. Sistemas Distribuídos. I. Schaeffer-Filho, Alberto Egon II. Cordeiro, Weverton Luis da Costa III. Fernandes, Stênio IV. Ziviani, Artur V. Vieira, Alex VI Título.

Sociedade Brasileira de Computação

Presidência
Lisandro Zambenedetti Granville (UFRGS), Presidente
Thais Vasconcelos Batista (UFRN),Vice-Presidente

Diretorias
Renata de Matos Galante (UFGRS), Diretora Administrativa
Carlos André Guimarães Ferraz (UFPE), Diretor de Finanças
Antônio Jorge Gomes Abelém (UFPA), Diretor de Eventos e Comissões Especiais
Renata Mendes de Araujo (UPM), Diretora de Educação
José Viterbo Filho (UFF), Diretor de Publicações
Claudia Lage Rebello da Motta (UFRJ), Diretora de Planejamento e Programas Especiais
Marcelo Duduchi Feitosa (CEETEPS), Diretor de Secretarias Regionais
Eliana Almeida (UFAL), Diretora de Divulgação e Marketing
Ricardo de Oliveira Anido (UNICAMP), Diretor de Relações Profissionais
Esther Colombini (UNICAMP), Diretora de Competições Científicas
Raimundo José de Araújo Macêdo (UFBA), Diretor de Cooperação com Sociedades Científicas
Cláudia Cappeli (UNIRIO), Diretora de Articulação com Empresas

Contato
Av. Bento Gonçalves, 9500
Setor 4 - Prédio 43.412 - Sala 219
Bairro Agronomia
91.509-900 – Porto Alegre RS
CNPJ: 29.532.264/0001-78
http://www.sbc.org.br

Laboratório Nacional de Redes de Computadores (LARC)

Diretor do Conselho Técnico-Científico
Paulo André da Silva Gonçalves (UFPE)

Vice-Diretora do Conselho Técnico-Científico
Rossana Maria de Castro Andrade (UFC)

Diretor Executivo
Ronaldo Alves Ferreira (UFMS)

Vice-Diretor Executivo
Danielo Gonçalves Gomes (UFC)

Membros Institucionais
SESU/MEC, INPE/MCT, UFRGS, UFMG, UFPE, UFCG (ex-UFPB Campus Campina Grande), UFRJ, USP, PUC-Rio, UNICAMP, LNCC, IME, UFSC, UTFPR, UFC, UFF, UFSCar, IFCE (CEFET-CE), UFRN, UFES, UFBA, UNIFACS, UECE, UFPR, UFPA, UFAM, UFABC, PUCPR, UFMS, UnB, PUC-RS, PUCMG, UNIRIO, UFS e UFU.

Contato
Universidade Federal de Mato Grosso do Sul (UFMS)
Faculdade de Computação
Caixa Postal 549
CEP 79.070-900 Campo Grande - MS - Brasil
http://www.larc.org.br

Organização do SBRC 2019

Coordenadores Gerais
Alberto Egon Schaeffer Filho (UFRGS)
Weverton Luis da Costa Cordeiro (UFRGS)

Coordenadores do Comitê de Programa
Antônio Jorge Gomes Abelém (UFPA)
Fabíola Gonçalves Pereira Greve (UFBA)

Coordenador de Palestras e Tutoriais
Ítalo Cunha (UFMG)

Coordenador de Painéis e Debates
Artur Ziviani (LNCC)

Coordenador de Minicursos
Miguel Elias Mitre Campista (UFRJ)

Coordenador de Workshops
Stênio Fernandes (UFPE)

Coordenador do Salão de Ferramentas
Leandro Villas (UNICAMP)

Coordenador do Concurso de Teses e Dissertações
Daniel Fernandes Macedo (UFMG)

Coordenadores do Hackaton
Raquel Lopes (UFMG)
Luis Carlos de Bona (UFPR)

Comitê de Organização Local
Avelino Zorzo (PUCRS)
Carlos Raniery Paula dos Santos (UFSM)
Cristiano Bonato Both (Unisinos)
Guilherme Rodrigues (IFSUL Charqueadas)
Jéferson Campos Nobre (UFRGS)
Juliano Wickboldt (UFRGS)
Marcelo Caggiani Luizelli (Unipampa)
Marcelo da Silva Conterato (PUCRS)
Rafael Pereira Esteves (IFRS)
Rodrigo Mansilha (Unipampa)
Tiago Ferreto (PUCRS)
Vinícius Guimarães (IFSUL Charquadas)

Comite Consultivo
Fabíola Gonçalves Pereira Greve (UFBA)

Paulo André da Silva Gonçalves (UFPE)
Fábio Luciano Verdi (UFSCAR)
Jó Ueyama (USP)
Antônio Jorge Gomes Abelém (UFPA)
Eduardo Cerqueira (UFPA)
Luiz Fernando Bittencourt (Unicamp)
Rossana Andrade (UFC)
Michele Nogueira (UFPR)
Edmundo Madeira (UNICAMP)

Mensagem dos Coordenadores Gerais

É com grande alegria e orgulho que, após 9 anos, estamos trazendo o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) novamente para a cidade de Gramado. O SBRC 2019 acontece em um ano que é especial por várias razões: primeiro, o aniversário de 50 anos da ARPANET, a primeira rede a implementar o conjunto de protocolos TCP/IP, que acabaram se estabelecendo mais tarde como os fundamentos da Internet; os 30 anos da invenção da World Wide Web pelo cientista Inglês Tim Berners-Lee; a comemoração dos 30 anos do .br (country code top-level domain), o qual tem atualmente mais de 4 milhões de domínios registrados; por fim, nesse ano comemora-se os 30 anos da Rede Nacional de Ensino e Pesquisa (RNP), que iniciou em 1989 como um projeto de pesquisa. É, portanto, um imenso privilégio termos a oportunidade de sediar o SBRC em um ano que carrega tanto significado para a área de redes de computadores e sistemas distribuídos.

A 37a edição do evento se apoia no histórico de sucesso do SBRC, que tradicionalmente inclui sessões técnicas, minicursos, painéis e debates, workshops, salão de ferramentas e palestras. Além de contar novamente com um Hackathon e com um Concurso de Teses e Dissertações, teremos muitas novidades no SBRC 2019, incluindo as reuniões de mentoria, o evento MUSAS e o workshop de estudantes latino-americanos. As reuniões de mentoria permitem oportunizar conversas 1-a-1 entre estudantes de pós-graduação e pesquisadores de excelência do Brasil e do exterior. O MUSAS (MUlheres em redeS de computadores e sistemAs diStribuídos) visa fomentar conexões entre mulheres atuando nas áreas de redes de computadores e de sistemas distribuídos, no Brasil e no exterior. Já o workshop de estudantes latino-americanos (LANCOMM) serve como um ponto de encontro para estudantes da região apresentarem e discutirem o andamento de suas pesquisas, e obterem feedback construtivo de pesquisadores estrangeiros de grande prestígio.

Neste ano, a trilha principal do SBRC recebeu 228 submissões completas e que entraram no processo de avaliação. Todos os artigos receberam pelo menos 3 revisões, e após um rigoroso processo seletivo, 80 artigos foram aceitos para publicação e organizados em 23 sessões técnicas. O Salão de Ferramentas selecionou 12 trabalhos que a serem demonstrados ao longo do SBRC 2019. Por sua vez, o Concurso de Teses e Dissertações selecionou 8 dissertações de mestrado e 8 teses de doutorado para apresentação durante o evento. Além disso, esse ano o evento conta com 4 palestrantes internacionais, 3 painéis, 1 tutorial, 5 minicursos e 11 workshops.

A organização de um evento com o porte do SBRC é um processo longo, que demanda muita energia e empenho. O evento desse ano só foi possível graças ao apoio e suporte incondicional de muitos grupos de pessoas e instituições. Agradecemos em particular o apoio da SBC, do LARC, do Comitê Consultivo do SBRC e da Comissão Especial de Redes de Computadores e Sistemas Distribuídos da SBC. O evento contou com o apoio do Comitê Gestor da Internet no Brasil (CGI.br) e do Núcleo de Informação e Coordenação do Ponto BR (NIC.br), da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), da Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) e da ACM SIGCOMM. Contamos novamente também com

o apoio institucional da Rede Nacional de Ensino e Pesquisa (RNP). Além disso, o SBRC 2019 recebeu o apoio de importantes patrocinadores, incluindo SAP, HUAWEI, DATACOM, Google, Bedu.tech, BRDigital e AdylNet. Também agradecemos à Universidade Federal do Rio Grande do Sul (UFRGS), e em especial ao Instituto de Informática (INF), pelo suporte incondicional.

Agradecemos a todas as pessoas que trabalharam diretamente na organização do evento. Obrigado pelo excelente trabalho de todos os membros do comitê de organização: Antônio Jorge Gomes Abelém/UFPA e Fabíola Gonçalves Pereira Greve/UFBA (Coordenadores do Comitê de Programa), Stênio Fernandes/UFPE (Coordenador de Workshops), Ítalo Cunha/UFMG (Coordenador de Palestras, Tutoriais e Mentoria), Artur Ziviani/LNCC (Coordenador de Painéis), Miguel Elias Mitre Campista/UFRJ (Coordenador de Minicursos), Leandro Villas/UNICAMP (Coordenador do Salão de Ferramentas), Daniel Fernandes Macedo/UFMG (Coordenador do Concurso de Teses e Dissertações), e Raquel Lopes/UFCG e Luis Carlos De Bona/UFPR (Coordenadores do Hackathon). Agradecemos também aos membros do nosso comitê local, que nos ajudaram na operacionalização de diversas tarefas relacionadas à organização: Avelino Zorzo/PUCRS, Carlos Raniery Paula dos Santos/UFSM, Cristiano Bonato Both/Unisinos, Guilherme Rodrigues/IFSUL Charqueadas, Jéferson Campos Nobre/UFRGS, Juliano Wickboldt/UFRGS, Marcelo Caggiani Luizelli/Unipampa, Marcelo da Silva Conterato/PUCRS, Rafael Esteves/IFRS, Rodrigo Mansilha/Unipampa, Tiago Ferreto/PUCRS, e Vinícius Guimarães/IFSUL Charqueadas. Agradecemos ao Luis Otávio Luz Soares, técnico administrativo do INF/UFRGS, pela imensa ajuda nos mais variados aspectos relacionados à organização do evento, assim como pelo apoio de Leandro Disconzi Vieira e Carlos Alberto da Silveira Junior, também técnicos administrativos da UFRGS. Por fim, agradecemos à nossa equipe local, formada por alunos de graduação, pós-graduação e pós-doutorando da UFRGS, pela dedicação e pronta ajuda sempre que precisamos: Arthur Selle Jacobs, Augusto Zanella Bardini, Bruno Dalmazo, Fernanda da Silva Bonetti, Guilherme Bueno de Oliveira, Guilherme Rotth Zibetti, Isadora Pedrini Possebon, Leonardo Lauryel, Libardo Andrey Quintero González, Lucas Bondan, Lucas Castanheira, Luciano Zembruzki, Mateus Saquetti, Jonatas Marques, Rafael Hengen Ribeiro, e Ricardo Parizotto.

Desejamos a todos um ótimo evento e uma semana produtiva e com muitas trocas de ideias em Gramado.

Alberto Egon Schaeffer Filho e Weverton Luis da Costa Cordeiro
Coordenadores Gerais do SBRC 2019

Mensagem do Coordenador de Workshops

Os Workshops do SBRC são uma importante oportunidade para aprofundamento no conhecimento de temas especializados ou emergentes na nossa comunidade científica. Como em anos anteriores, a coordenação manteve a *Chamada de Propostas de Workshops do SBRC*, que tem estimulado a comunidade brasileira de Redes e Sistemas Distribuídos a discutir a viabilidade de workshops com temas de pesquisa mais tradicionais, bem como estimular o debate para adoção de eventos que tratam de temas mais emergentes.

Tivemos uma diversidade salutar no Comitê de Avaliação em termos de localização geográfica e experiência, gerando, portanto, opiniões complementares sobre as propostas submetidas. Consideramos todas as propostas de altíssima qualidade, com temas, focos e escopo diversos. Nesta edição de 2019, buscamos equilibrar os benefícios trazidos aos participantes com a capacidade de alocação de espaços no evento. Desta forma, foram selecionados 10 Workshops de alta qualidade, além do tradicional Workshop da RNP (*WRNP*). Dentre as propostas aceitas, oito são reedições de workshops tradicionais do SBRC, a saber: **Gerência e Operação de Redes e Serviços** (*WGRS*), **Testes e Tolerância a Falhas** (*WTF*), **Pesquisa Experimental da Internet do Futuro** (*WPEIF*), **Segurança Cibernética em Dispositivos Conectados** (*WSCDC*), **Trabalhos de Iniciação Científica e Graduação** (*WTICG*), **Blockchain: Teoria, Tecnologias e Aplicações** (*WBlockchain*), **Clouds e Aplicações** (*WCGA*), e **Computação Urbana** (*CoUrb*). Como novidade, teremos dois novos workshops, a saber: o Workshop de **Teoria, Tecnologias e Aplicações de Slicing para Infraestruturas Softwarizadas** (*WSlice*) e o **Latin American Student Workshop on Data Communication Networks** (*LANCOMM Student Workshop*), que tem como público-alvo principal os alunos de pós-graduação e pesquisadores da América Latina, além de estreitar os relacionamentos da nossa comunidade com a Association for Computing Machinery (*ACM*).

Como coordenador dos Workshops do SBRC 2019, gostaria de agradecer a todos os envolvidos na seleção das propostas. Primeiramente aos coordenadores gerais do SBRC 2019, Weverton Cordeiro (UFRGS) e Alberto Egon Schaeffer Filho (UFRGS), pelo convite para a coordenação desta chamada de trabalhos, além de todo o apoio necessário para sua condução apropriada. Agradeço também a todos os membros do comitê de avaliação pelo imenso esforço nas revisões e discussões de alta qualidade de todas as propostas submetidas. Por fim, agradeço aos coordenadores dos workshops aceitos, pela dedicação no cumprimento dos prazos e na condução dos trabalhos internos de seleção dos artigos para seus respectivos eventos, mantendo a alta qualidade geral do SBRC. Os inscritos nos diversos workshops esperam ansiosamente pelas apresentações dos trabalhos e pelas frutíferas discussões que serão geradas.

Stênio Fernandes
Coordenador dos Workshops do SBRC 2019

Comitê de Programa

- Alex Vieira (Universidade Federal de Juiz de Fora)
- Alysson Bessani (University of Lisbon, Faculty of Sciences)
- Antônio Abelém (Universidade Federal do Pará - UFPA)
- Antonio Rocha (IC/UFF)
- Artur Ziviani (LNCC)
- Billy Pinheiro (Universidade Federal do Pará)
- Carlos Ferraz (UFPE)
- Célio Vinicius Neves de Albuquerque (UFF)
- Claudio de Farias (Universidade Federal do Rio de Janeiro)
- Cristiano Silva (Universidade Federal de São João del Rei)
- Daniel Menasche (UFRJ)
- Eduardo Alchieri (UnB)
- Fabíola Greve (Universidade Federal da Bahia)
- Guido Souza Filho (Universidade Federal da Paraíba)
- Igor Machado Coelho (Universidade do Estado do Rio de Janeiro)
- Jeferson Campos Nobre (UFRGS)
- Leobino Sampaio (Universidade Federal da Bahia)
- Luciano Porto Barreto (UERJ/CVM)
- Luis Carlos De Bona (University Federal of Parana)
- Michele Nogueira (Universidade Federal do Paraná)
- Otto Carlos Muniz Bandeira Duarte (UFRJ)
- Rostand Costa (Universidade Federal da Paraíba)

Sumário

Sessão técnica 01 1

Provendo uma Infraestrutura de Software Fatiada, Isolada e Segura de Funções Virtuais através da Tecnologia de Corrente de Blocos 2
Gabriel Antonio Rebello (UFRJ), Gustavo Camilo (UFRJ), Leonardo Silva (UFRJ), Lucas Guimarães (UFRJ), Lucas Airam Souza (UFRJ), Igor Alvarenga (UFRJ), Otto Carlos Muniz Bandeira Duarte (UFRJ)

Experimental Analysis of the Scalability of Ethereum Blockchain in a Private Network 16
Iago Ochôa (UNIVALI), Rafael Piemontez (UNIVALI), Lucas Martins (UNIVALI), Valderi Leithardt (UNIVALI), Cesar Zeferino (UNIVALI)

Um Estudo Sobre o Misterioso e Arriscado Mundo dos Misturadores de Criptomoedas 29
Rodolfo Costa (UFPB), Rostand Costa (UFPB)

Demochain – Framework destinado a criação de redes blockchain híbridas para dispositivos IoT 43
Carlos Rolim (UFRGS), Lorenzo de Freitas (URI Santo Ângelo)

Sessão técnica 02 57

Sensor Store: uma loja de dados de IoT usando Tangle 58
Caio Barbosa (UFES), Vinicius Mota (UFES), Roberta Gomes (UFES), Jose Pereira Filho (UFES)

Editora Científica Autônoma e Distribuída sobre Blockchain Privada .. 72
Elder Bruno Evaristo Correa (UFPA), Vagner Nascimento (UFPA), André Defrémont (UFPA), Billy Pinheiro (UFPA), Antônio Abelém (UFPA)

Controle de acesso a serviços usando Smart Contracts 86
Marcos Vinicius Souza (UFSCar), Helio Guardia (UFSCar)

Proof-of-Stake baseado em Tempo Discreto 100
Yoshitomi Maehara Aliaga (UNICAMP), Diego Fernandes Gonçalves Martins (UNICAMP), Marco Aurelio Amaral Henriques (UNICAMP)

II Workshop Blockchain: Teoria, Tecnologias e Aplicações
SBRC 2019
Sessão Técnica 1

Provendo uma Infraestrutura de Software Fatiada, Isolada e Segura de Funções Virtuais através da Tecnologia de Corrente de Blocos

Gabriel Antonio F. Rebello, Gustavo F. Camilo, Leonardo G. C. Silva, Lucas C. B. Guimarães, Lucas Airam C. de Souza, Igor D. Alvarenga, e Otto Carlos M. B. Duarte

[1]

Grupo de Teleinformática e Automação
Universidade Federal do Rio de Janeiro (UFRJ)

{gabriel,franco,leonardo,chagas,airam,alvarenga,otto}@gta.ufrj.br

Resumo. *As tecnologias de fatiamento da rede (Network Slicing), virtualização de funções de rede (Network Function Virtualization - NFV) e redes definidas por software (Software-Defined Networking - SDN) proveem serviços fim-a-fim ágeis e sob demanda. A identificação de uma função virtual defeituosa torna-se obrigatória, pois serviços alocam recursos em um ambiente distribuído e sem confiança entre os pares composto por múltiplos inquilinos e provedores de serviço concorrentes. Este artigo propõe e desenvolve uma arquitetura baseada em correntes de blocos para prover auditabilidade às operações de orquestração de fatias de rede. Um protótipo de um caso de uso foi desenvolvido e implementado,utilizando a plataforma Hyperledger Fabric na qual cada fatia de rede opera sobre um canal isolado. Os resultados mostram que é possível prover segurança à criação de fatias de rede, mas que a obtenção de consenso e o número de transações requeridas pelas fatias de rede são um grande desafio.*

1. Introdução

As redes móveis de próxima geração fornecem um modelo de conectividade com múltiplos serviços de rede adaptados para atender a demanda de cada segmento de cliente. As redes definidas por *software* (*Software-Defined Networking* - SDN) e a virtualização de funções de rede (*Network Function Virtualization* - NFV) são as principais tecnologias que utilizam a virtualização para fornecer a capacidade de programação da rede. Assim, as tecnologias NFV e SDN criam uma cadeia de funções de rede (*Service Function Chain* - SFC) fim-a-fim [Halpern and Pignataro 2017] para fornecer serviços sob demanda e adaptados a cada aplicação. Apesar de a associação de NFV e SDN fornecer a agilidade e o baixo custo desejados pelas telecomunicações, surgem novos desafios de segurança [Medhat et al. 2017]. Além disso, o impacto de possíveis ataques aumenta porque os ataques aos hospedeiros de funções de rede podem comprometer simultaneamente milhares de usuários [Bhamare et al. 2016]. Portanto, é de grande importância reduzir os possíveis vetores de ataque a funções virtuais de rede (*Virtual Network Functions* - VNF) e fornecer um gerenciamento de configuração seguro e confiável. Garantir o isolamento entre fatias de rede é essencial para evitar ataques comuns em infraestruturas compartilhadas. Além disso, os inquilinos de cada fatia compartilham a mesma infraestrutura de

[1] Este artigo é a versão em português de um artigo em inglês submetido para um congresso internacional.

nuvem, e as cadeias podem envolver funções virtualizadas instanciadas em domínios de provedores concorrentes. O ambiente multi-inquilino e multi-domínio aumenta a possibilidade de ataques dentro da nuvem, ao mesmo tempo que dificulta a responsabilização dos provedores de serviços quando ocorre uma falha. É necessário garantir que a cadeia de serviços, formada por uma cadeia de funções virtuais de rede, seja construída de maneira confiável em um ambiente sem confiança entre os pares. Em um ambiente multi-inquilino formado por provedores concorrentes, é vantajoso para um provedor criar uma ataque para prejudicar um concorrente. Logo, a capacidade de auditoria é obrigatória para identificar uma configuração de VNF defeituosa ou comprometida, e a tecnologia de corrente de blocos fornece as características necessárias de não repúdio e imutabilidade do histórico de configuração de uma função virtual.

Este artigo propõe utilizar a tecnologia de corrente de blocos para registrar, como transações assinadas, todos os comandos que criam, modificam, configuram, migram ou destroem as funções de rede de cada fatia da rede. Portanto, todos os problemas de funcionamento da rede podem ser verificados e um erro pode ser atribuído corretamente a um provedor de serviço em um ambiente de concorrência, multi-inquilino, e sem confiança.

Em trabalhos anteriores, os autores deste artigo avaliaram o desempenho do uso de correntes de blocos na virtualização de funções de rede para proteger comandos de gerenciamento, atualizações e migração de funções virtuais de rede com garantias de anonimidade [Alvarenga et al. 2018, Rebello et al. 2019].

Este artigo foca no uso de correntes de blocos para fatiamento da rede (*network slicing*). Fatias de rede suportam requisitos de serviço para atender redes veiculares tolerantes a atraso, internet das coisas (*Internet of Things* - IoT), indústria 4.0 e serviços críticos como saúde eletrônica (*e-Health*), cidades inteligentes (*smart cities*) e redes elétricas inteligentes (*smart grids*). O cenário extraordinariamente diverso requer diferentes correntes de blocos com características específicas adaptadas ao serviço requerido. Por isso, este artigo propõe atender aos diferentes requisitos de cada fatia de rede através de diferentes categorias de correntes de blocos. A estrutura de dados, o protocolo de consenso e o protocolo de comunicação das corrente de blocos são adaptados a cada funcionalidade de fatia de rede específica. O trabalho apresenta uma arquitetura baseada em correntes de blocos para criar fatias de rede fim-a-fim seguras e adaptadas para cada caso de uso. Um protótipo de caso de uso que segue a arquitetura proposta com diferentes tipos de correntes de blocos é implementado usando a plataforma de código aberto Hyperledger Fabric [Androulaki et al. 2018]. O protótipo implementa dois contratos inteligentes (Hyperledger *chaincode*) com formatos de transação específicos para proteger o gerenciamento de fatias de rede e as operações de configuração de VNFs. Cada fatia de rede é executada em um canal Hyperledger isolado. Os resultados mostram que é possível proteger a construção de fatias de rede, mas que estruturas de dados otimizadas são necessárias para aumentar a taxa de transações necessárias para atender às fatias.

O restante deste artigo está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. A Seção 3 apresenta o fatiamento seguro de redes através de corrente de blocos. A Seção 4 detalha a arquitetura proposta. A Seção 5 descreve o protótipo desenvolvido e sua avaliação. Por fim, a Seção 6 conclui o artigo.

2. Trabalhos relacionados

Diversos trabalhos exploram o estado da arte de corrente de blocos aplicada a problemas de redes de comunicação e redes de quinta geração (5G) [Yahiatene and Rachedi 2018, Ortega et al. 2018, Thuemmler et al. 2018, Capossele et al. 2018, Rawat and Alshaikhi 2018, Rosa and Rothenberg 2018, Boudguiga et al. 2017]. Os trabalhos se concentram no uso de corrente de blocos como um repositório de dados incrementais replicados, no qual todas as transações realizadas são assinadas e registradas com criptografia de chaves assimétricas. Yahiatene *et al.* e Ortega *et al.* propõem o uso de corrente de blocos como um mecanismo para fornecer segurança em redes veiculares [Yahiatene and Rachedi 2018, Ortega et al. 2018]. Eles argumentam que a corrente de blocos pode fornecer autenticidade e confiança com baixa latência para permitir redes seguras. Thuemler *et al.* e Capossele *et al.* discutem os requisitos necessários ao 5G para prover saúde eletrônica (*e-Health*) com base em experiências reais [Thuemmler et al. 2018, Capossele et al. 2018]. Os resultados do artigo mostram que a corrente de blocos fornece a privacidade e proteção de dados necessários para proteger registros médicos em um ambiente sem confiança. Rawat *et al.* propõem uma solução baseada em corrente de blocos para redes sem fio virtuais que permite a confiança em provedores de nuvem [Rawat and Alshaikhi 2018]. Rosa e Rothemberg fornecem diretrizes para incorporar os aplicativos distribuídos baseados em corrente de blocos em cenários com múltiplos domínios administrativos [Rosa and Rothenberg 2018]. Boudguiga *et al.* apresentam uma solução para atualizar dispositivos IoT através de informações armazenadas no corrente de blocos [Boudguiga et al. 2017]. A proposta do presente artigo fornece uma taxonomia de fatias baseadas em corrente de blocos que engloba e generaliza todas as aplicações de corrente de blocos como casos de uso em fatias de rede.

Outros trabalhos investigam o problema de vulnerabilidades de segurança em ambientes NFV multi-inquilino e multi-domínio [Pattaranantakul et al. 2018, Paladi et al. 2018]. Eles mostram que a confiança nos provedores de nuvem é incerta e que o comprometimento de uma única VNF no núcleo da rede põe em risco todo o serviço fim-a-fim. Zaowat e Hasan propuseram o SECAP, uma estrutura baseada em corrente de blocos para armazenar com segurança uma árvore de proveniência de aplicativos em nuvem [Zawoad and Hasan 2016]. A estrutura protege os registros das mudanças de estado do aplicativo. Bozic *et al.* propõem uma arquitetura para gerenciar estados de execução de máquinas virtuais usando um sistema baseado em corrente de blocos [Bozic et al. 2017]. O sistema usa uma estrutura de corrente de blocos para registrar as instruções do hipervisor de virtualização do sistema na forma de transações.

Com relação à segurança do fatiamento de rede, Bordel *et al.* propõem uma solução baseada em geradores de números pseudo-aleatórios para fornecer segurança dentro de fatias de rede para dispositivos IoT e estações-base em sistemas 5G [Bordel et al. 2018]. Khettab *et al.* propõem utilizar tecnologias NFV e SDN para proteger fatias de rede de múltiplos domínios instanciando funções de rede de segurança, como *firewalls* e sistemas de detecção de intrusão [Khettab et al. 2018]. Os trabalhos, no entanto, não abordam possíveis comportamentos maliciosos de administradores de rede.

Outros trabalhos propõem o uso de corrente de blocos para prover confiança nos administradores de rede e intermediários responsáveis pelo fatiamento da rede. Valtanen

et al. analisam o uso de corrente de blocos em provedores de fatia de rede para coletar, configurar e alocar recursos em automação industrial [Valtanen et al. 2018]. O artigo aponta as vantagens de usar corrente de blocos em casos de uso do 5G. Backman *et al.* propõem o uso de corrente de blocos para gerenciar recursos de rede 5G virtualizados em um cenário de múltiplos administradores [Backman et al. 2017].

Este artigo propõe uma arquitetura para construção de correntes de blocos que protegem e permitem a auditabilidade da criação de fatias de redes e também a atualização das funções de rede das fatias isoladas. A proposta atende ao ambiente multi-inquilino e multi-domínio sem confiança entre os pares.

3. Fatiamento Seguro de Redes através de Corrente de Blocos

A corrente de blocos é uma estrutura de dados replicada que garante a confiança e o funcionamento adequado de um sistema distribuído sem a necessidade de uma autoridade central comum a todos os participantes. Uma representação típica de uma corrente de blocos é mostrada na Figura 1. Um valor resultante de uma função resumo (*hash*) criptográfica identifica cada bloco. Cada bloco contém as transações realizadas em um determinado intervalo de tempo e o identificador do bloco anterior. Nesta estrutura replicada, cada nó participante do consenso possui uma cópia local da corrente de blocos que contém todas as transações desde o início. Como a cópia atual é a mesma em qualquer nó da rede, a propriedade de não repúdio entre membros é garantida porque todas as transações são assinadas e cada membro usa sua chave pública como identificação.

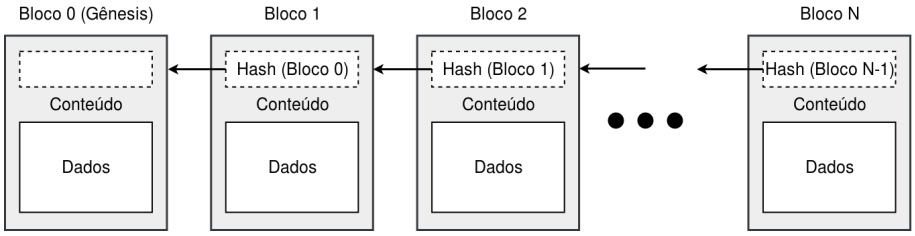


Figura 1. Estrutura de uma corrente de blocos, na qual cada bloco é associado ao bloco anterios e a função resumo criptográfica garante a integridade de cada bloco.

A utilização de corrente de blocos é necessária em ambientes distribuídos em que os participantes não conseguem chegar a um acordo sobre uma autoridade centralizada que governe todos os procedimentos sensíveis de rede. Em centros de dados virtualizados nos quais vários serviços de nuvem são orquestrados, a presença de uma VNF malintencionada em um serviço pode afetar toda a cadeia pela qual os pacotes são roteados sem o conhecimento do administrador da nuvem. Além disso, se um invasor tiver acesso ao orquestrador, o registro de operações pode ser manipulado para ocultar uma ameaça. O uso de corrente de blocos, apesar de envolver uma quantidade maior de processamento como um todo, permite gerenciar e atualizar VNFs de forma distribuída e segura, onde as transações podem ser verificadas localmente por cada nó com a garantia de não repúdio e integridade.

3.1. Modelo de Atacante

Este artigo considera um modelo de atacante de Dolev *et al.*, no qual um atacante pode ler, enviar e descartar uma transação endereçada à corrente de blocos, ou qualquer

pacote da rede [Dolev and Yao 1983]. O atacante pode se conectar passivamente à rede e capturar trocas de mensagens ou injetar, reproduzir, filtrar e trocar informações ativamente. Os ataques podem ter como alvo inquilinos, VNFs, a corrente de blocos em si e a rede.

Ataques à corrente de blocos objetivam impedir que uma transação ou bloco legítimo sejam adicionados à corrente de blocos. Para que um ataque à corrente de blocos seja bem-sucedido, o atacante deve controlar uma parcela significativa da rede para afetar o protocolo de consenso. Um protocolo de consenso tolerante a falhas mitiga esse tipo de ataque. Os ataques que exigem corrupção ou adulteração de transações são impossíveis quando todas as transações incluem seu *hash* assinado correspondente.

Ataques a inquilinos ou VNFs consistem em tentar obter informações de configuração ou personificar o alvo. Os ataques de personificação não são possíveis pois todas as transações enviadas para a corrente de blocos são assinadas por seus emissores. A encriptação de informações confidenciais mitiga ataques que buscam obter informações de configuração, nos quais o atacante precisa obter a chave privada da vítima. Este trabalho não aborda o caso em que um inquilino ou VNF é comprometido através de invasão de terminal ou seqüestro de chave. A arquitetura proposta, no entanto, elimina a necessidade de qualquer serviço de escuta ativo em um VNF, e utiliza terminais em modo somente leitura para mitigar vetores de ataque.Além disso, a arquitetura proposta permite a auditoria de todas as transações passadas. Portanto, se um invasor tentar modificar a corrente de blocos usando pares de chaves roubados, a tentativa será registrada. Após a descoberta de um incidente, o inquilino ou provedor pode facilmente substituir os pares de chaves roubados, restabelecendo a segurança e evitando mais danos.

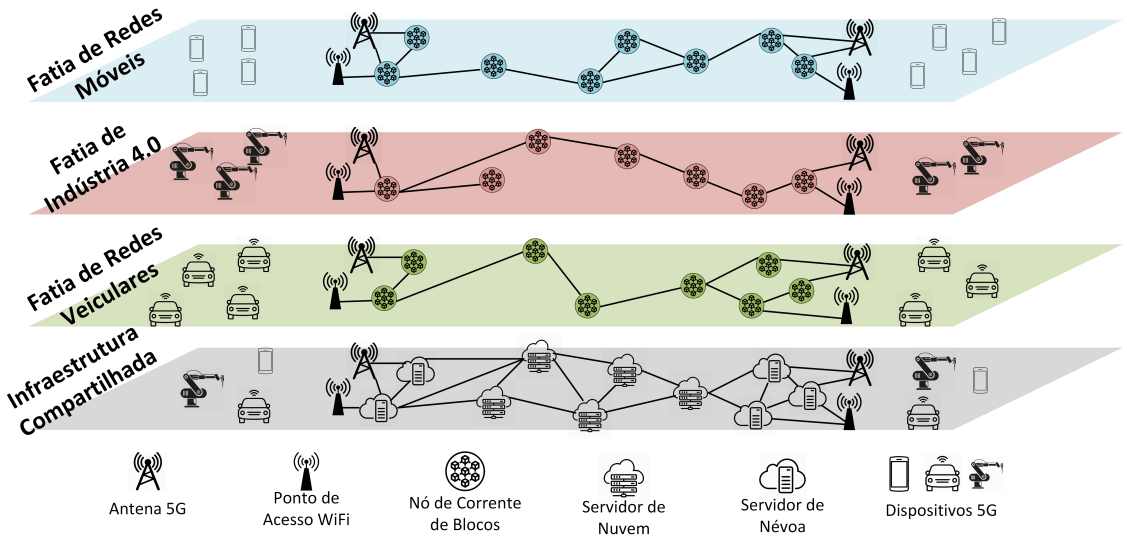


Figura 2. Fatias de rede isoladas através de corrente de blocos em uma infraestrutura física compartilhada. Cada fatia de rede é adaptada às necessidades de um caso de uso.

Ataques à rede representam a tentativa de isolar um único inquilino, um grupo de inquilinos ou um grupo de VNFs da rede, impedindo assim que a rede execute transações ou leia conteúdo da corrente de blocos. Esta categoria de ataque contempla ataques

clássicos de rede, que podem ser mitigados através do estabelecimento de caminhos redundantes entre a corrente de blocos e VNFs ou inquilinos. Este trabalho assume uma rede pública redundante, como a Internet, que interconecta todos os participantes. A suposição dificulta o isolamento de uma única entidade se o invasor não estiver em sua rede adjacente. A mitigação completa de ataques de rede está fora do escopo deste trabalho. A arquitetura proposta foca nos ataques à corrente de blocos e transações antecipadas. No entanto, ao eliminar os serviços de escuta das VNFs, a arquitetura elimina os ataques de negação de serviço da camada de aplicação, que são uma ameaça comum em ambientes de nuvem compartilhados.

4. A Arquitetura Proposta

Diferentes casos de uso exigem funcionalidades específicas e incorrem em diferentes características de corrente de blocos para cada fatia de rede. Em vez de tentar abordar todos os casos de uso, este trabalho propõe classes de corrente de blocos que atendem a muitos casos de uso. Assim, uma simples taxonomia de fatias baseadas em corrente de blocos composta de quatro categorias de fatias pode abordar uma infinidade de possíveis casos:

- **Fatias de administrador único.** Essa categoria trata casos de uso nos quais a fatia de rede inteira é administrada por uma única entidade. Neste caso, as correntes de blocos atuam apenas como um banco de dados distribuído, no qual as decisões da rede são controladas por uma autoridade central, como em redes privadas.
- **Fatias multi-domínio tolerantes a falhas desastrosas**. Essa categoria trata casos de uso nos quais nós individuais na rede podem falhar, mas a rede está livre de comportamentos mal-intencionados. Essa categoria fornece alta eficiência para ambientes descentralizados de vários domínios que garantem a segurança da rede por meio de políticas baseadas em contrato. Este cenário é ideal para redes de até dezenas de administradores. Exemplos incluem comunicação horizontal e concordância entre os controladores SDN para implementar um serviço. Este tipo de fatia é baseia-se em correntes de blocos federadas e utiliza os protocolos RAFT e PAXOS.
- **Fatias multi-domínio tolerantes a falhas bizantinas.** Essa categoria usa protocolos de consenso tolerantes a falhas bizantinas (*Byzantine Fault Tolerance* - BFT) para proteger a rede contra comportamentos maliciosos. Os protocolos BFT fornecem eficiência razoavelmente alta para ambientes com até algumas centenas de nós identificados. Casos de uso incluem ambientes NFV em vários domínios e com vários inquilinos que implementam serviços fim-a-fim. Esta categoria baseia-se em correntes de blocos federadas robustas a ataques de conluio [Alvarenga et al. 2018, Sousa et al. 2017].
- **Fatias públicas totalmente descentralizadas.** Esse tipo de fatia de rede fornece escalabilidade de milhares de nós, sacrificando a eficiência e a taxa de transferência. Essas fatias dependem de protocolos baseados em provas que determinam um estado global por meio de consenso probabilístico0. As fatias de rede baseadas em provas fornecem alta escalabilidade, pois não precisam conhecer todos os nós da rede para obter consenso. Portanto, esse tipo de fatia é mais adequado para redes públicas com muitos dispositivos, como fatias de IoT.

Este trabalho propõe uma arquitetura na qual cada categoria fatia de rede baseada em corrente de blocos aborda um ou mais casos de uso do 5G, criando redes isoladas com segurança e confiança. A figura 2 descreve um cenário que usa corrente de blocos para três fatias de rede: uma fatia de rede móvel, uma fatia de indústria 4.0 e uma fatia de redes veiculares. Para garantir a justiça no protocolo de consenso, cada centro de dados pode hospedar no máximo um nó de corrente de blocos por fatia de rede. Os nós de corrente de blocos em uma fatia são invisíveis para qualquer entidade fora da fatia. O trabalho propõe fornecer a capacidade de auditoria de criação e gerenciamento de fatias, registrando todas as operações de orquestração da VNF em uma corrente de blocos de gerenciamento. A corrente de blocos de gerenciamento registra as operações de orquestração que criam ou modificam uma fatia de rede. Toda operação é assinada pelo cliente que solicitou a modificação. Os participantes da corrente de blocos devem validar cada operação por consenso e fornecer uma prova assinada irrefutável de que a transação foi aceita antes que as operações sejam realizadas. A solicitação assinada combinada com o registro permanente fornecido pela corrente de blocos garante que um comportamento malicioso seja rastreável. Assim, a proposta de corrente de blocos gerencial garante a procedência, a prestação de contas e a rastreabilidade das falhas em um ambiente NFV multi-inquilino e multi-domínio. Além disso, o trabalho propõe o uso de contratos inteligentes para fornecer automação e transparência em ambientes sem confiança distribuídos, em vez de confiar em um determinado nó para receber e processar transações. As propriedades de automação e transparência dos contratos inteligentes são ideais para criar fatias de rede fim-a-fim seguras que envolvem VNFs em vários domínios concorrentes, pois garantem que todos os nós da rede obedeçam ao mesmo conjunto de regras e que o código executado seja visível para qualquer nó participante.

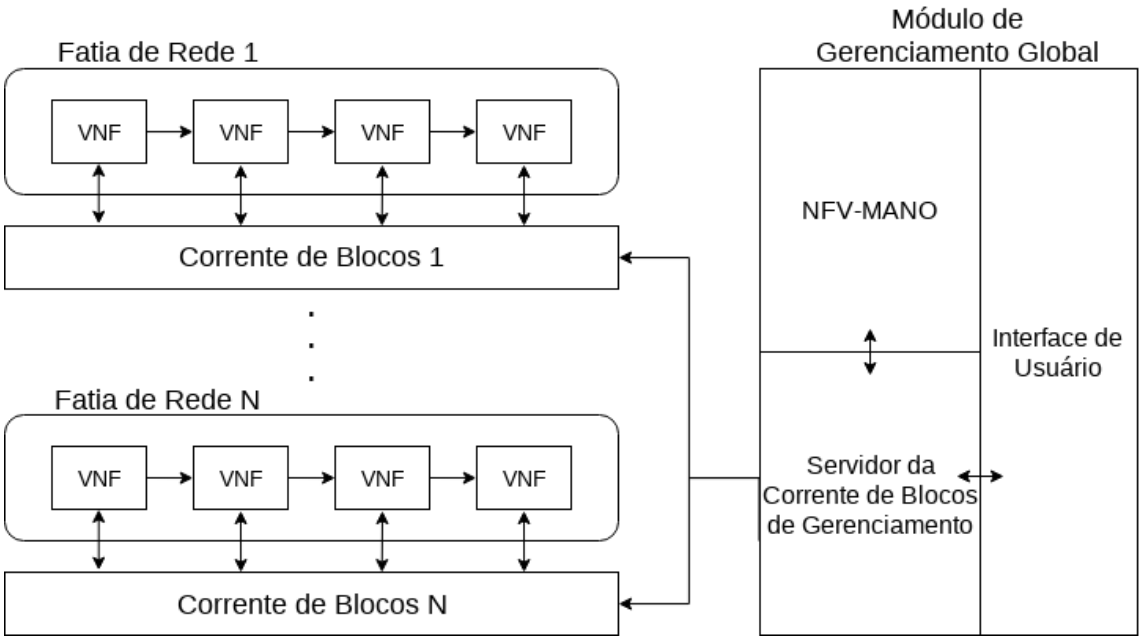


Figura 3. A arquitetura proposta baseada em corrente de blocos para fatiamento de rede. O usuário interage com o módulo de gerenciamento global para criar fatias de rede seguras. Cada VNF em uma fatia de rede é conectada a uma corrente de blocos responsável por registrar solicitações de configuração e informações relevantes, conforme especificado pelo usuário.

A arquitetura proposta, representada na figura 3, compreende quatro componentes principais: uma interface de usuário, o módulo de gerenciamento e orquestração NFV (NFV-MANO), um módulo de servidor de criação de corrente de blocos e um módulo de gerenciamento de corrente de blocos. Os módulos compõem o módulo de gerenciamento global, que é responsável por conectar o cliente aos serviços oferecidos. Nesta arquitetura, o cliente interage com o módulo de gerenciamento global por meio de uma interface de usuário para criar/modificar uma fatia de rede segura ou para solicitar informações de fatia, como configurações de VNFs e cadeias de funções. O cliente especifica as características da fatia, como VNFs desejadas e restrições de posicionamento, e a categoria de corrente de blocos correspondente. O módulo de interface do usuário converte as especificações em operações de criação de fatias/corrente de blocos e as envia como transações assinadas para aprovação na corrente de blocos de gerenciamento. Depois que as transações são aprovadas, o módulo NFV-MANO e o módulo de criação de corrente de blocos verificam a corrente de blocos de gerenciamento para obter operações pendentes. Os módulos executam as operações para criar novas fatias seguras e emitem transações de resposta assinadas para a corrente de blocos de gerenciamento. O cliente pode então interagir com o módulo de interface do usuário para verificar se a fatia segura solicitada foi criada com sucesso.

5. O Protótipo baseado no Hyperledger Fabric

Este trabalho desenvolve um protótipo da arquitetura proposta que usa a plataforma Hyperledger Fabric [Androulaki et al. 2018]. O Hyperledger Fabric é uma plataforma de código aberto para implementar correntes de blocos entre organizações em ambientes sem confiança. Cada organização mantém uma réplica da corrente de blocos e pode acrescentar blocos através de um protocolo de consenso. A arquitetura modular do Hyperledger Fabric permite aos administradores de rede projetar sistemas baseados em sub-redes isoladas que suportam correntes de blocos específicas. As correntes de blocos no Hyperledger Fabric suportam protocolos de consenso plugáveis que proveem a personalização proposta para atender a casos de uso e modelos de confiança específicos. Os desenvolvedores das correntes de blocos podem configurar permissões de leitura e escrita para criar redes privadas, federadas ou públicas. O Hyperledger Fabric fornece um serviço de identidade e associação de membros que gerencia os identificadores dos usuários e provedores, e autentica todos os participantes na rede para criar redes permissionadas. Nós e canais são os conceitos chave mais importantes de uma rede baseada em corrente de blocos permissionada do Hyperledger Fabric.

Os nós representam as entidades que participam do processamento de uma transação ou mantêm uma cópia da corrente de blocos. O Hyperledger Fabric provê três tipos de nós: clientes, pares (*peers*) e ordenadores. Um cliente representa um usuário que envia transações aos pares para validação e assinatura, e transmite propostas de transação assinadas para o serviço de ordenação. Os pares são um elemento fundamental da rede porque executam propostas de transação, validam transações e mantém os registros na corrente de blocos. Os pares também instanciam contratos inteligentes e armazenam o estado global, uma representação sucinta do estado mais recente da corrente de blocos. Os nós ordenadores formam coletivamente o serviço de ordenação, que é responsável por estabelecer a ordem total e o empacotamento de todas as transações em um bloco usando um protocolo de consenso. Os ordenadores não participam da execução da transação nem

validam transações. O desacoplamento das funcionalidades de ordenação e validação aumenta a eficiência da rede, pois permite o processamento paralelo de cada fase. A Figura 4 descreve um exemplo de uma corrente de blocos permissionada com quatro organizações no Hyperledger Fabric. Cada organização recebe transações de clientes e as retransmite para os ordenadores após a validação pelos pares. Cada organização possui um único ordenador, garantindo a justiça no protocolo de consenso.

Caminhos de mensagens diferentes, chamados canais, isolam as correntes de blocos. Um canal Hyperledger Fabric é uma sub-rede de comunicação privada e isolada entre um subconjunto de nós da rede específicos para fornecer privacidade e confidencialidade às transações. Todos os dados transmitidos em um canal, incluindo transações, contratos inteligentes, configurações de associação e informações de canal, são invisíveis e inacessíveis a qualquer entidade externa a um canal. A funcionalidade do canal é ideal para a proposta de oferecer correntes de blocos personalizadas para diferentes serviços de rede, pois permite que os administradores dos canais estabeleçam diferentes formatos de bloco e transação, além do protocolo de consenso, para cada canal. Portanto, podemos usar canais para oferecer fatias de rede protegidas por correntes de blocos configuradas de forma específica. Formatos de transação são definidos em contratos inteligentes, chamados de *chaincode* no Hyperledger Fabric, escritos em Go, Node.js ou linguagem Java.

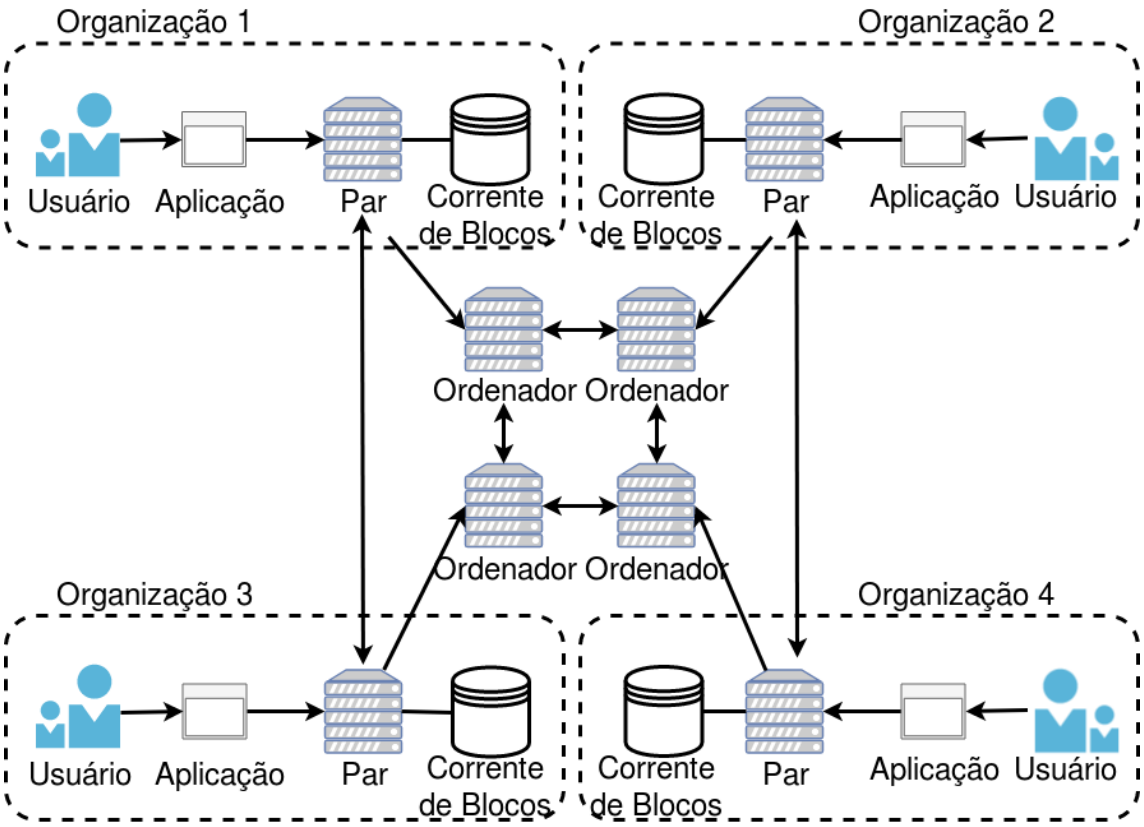


Figura 4. Uma corrente de blocos permissionada do Hyperledger Fabric. Usuários de cada organização usam aplicações para emitir transações as retransmitem para os ordenadores para o ordenamento global e a adição em um bloco. Depois de um bloco ser proposto e aceito pelos ordenadores através do consenso, os pares atualizam a corrente de blocos e o estado global.

5.1. Avaliação do Protótipo

O trabalho implementa duas correntes de blocos que representam, respectivamente, a corrente de blocos de gerenciamento e um exemplo de corrente de blocos para proteger uma fatia de rede. O exemplo de corrente de blocos protege a atualização de configuração da VNF e a migração em uma fatia de rede por registrar permanentemente configurações na corrente de blocos. Um consórcio com três organizações controla as duas implementações de corrente de blocos. Cada uma das três organizações controla um nó que possui direitos administrativos sobre a rede. Todas as três organizações recebem transações de um número variável de nós clientes na rede de corrente de blocos. Um computador Intel Core i7-7700 CPU 3.60GHz com 64 GB RAM cria todos os nós como contêineres Docker. Contêineres constroem múltiplos ambientes isolados de espaços de usuários que permitem a otimização de recursos computacionais da rede de corrente de blocos.

O trabalho implementa dois contratos inteligentes[1] escritos em Go, que são executados em todos os nós da rede [Alvarenga et al. 2018, Rebello et al. 2019]. O primeiro contrato inteligente, parcialmente descrito na Lista 1, gerencia autonomamente o gerenciamento e a orquestração de VNF através de transações de instrução e resposta. Quando um cliente solicita uma fatia, o servidor da corrente de blocos de gerenciamento de emite uma transação de instrução com o comando de instrução. O contrato coloca a transação de instrução em uma fila de transações pendentes de instrução. O código notifica o módulo NFV-MANO, que executa a instrução pendente e envia a saída do comando para o servidor da corrente de blocos de gerenciamento. O módulo NFV-MANO emite uma transação de resposta que inclui um campo contendo o identificador da transação de instrução correspondente. Isso fornece a rastreabilidade de cada transação executada na corrente de blocos e, portanto, a responsabilização de entidades maliciosas.

```
struct instructionTransaction
{
    command                 string
    transactionType         string
    transactionName         string
    issuer                  string
}
initialize queue
initInstruction (instruction <command,name,issuer>)
{
    if instruction is not unique or instruction is not well-formatted:
        return error
    putState (instruction.name, instruction)
    put (transactionID , queue)
    notify orchestrator
    return success
}
```

Listing 1. Parte do pseudo-código do contrato inteligente que emite transações de instrução. O campo de comando contém a operação de orquestração a ser executada pelo módulo NFV-MANO. O contrato estabelece uma fila de instruções pendentes a serem processadas pelo orquestrador NFV.

[1]O código completo está disponível em https://github.com/gta-ufrj/hpsr-smart-contracts

O segundo contrato inteligente, descrito na Lista 2, define e atualiza uma configuração de uma VNF. Um cliente emite uma transação para a corrente de blocos conectada a cada VNF em uma fatia de rede. A transação contém um texto descritivo com a configuração associada no campo de descrição, assim como os dados de configuração no campo de configuração.

```
struct configurationTransaction
{
    configurationIdentifier string
    versionIdentifier       string
    description             string
    configuration           string
    transactionType         string
    transactionName         string
    issuer                  string
}
initConfiguration (configuration <description,configuration,name,issuer
    >){
    if configuration is not unique or configuration is not well-formed:
        return error
    putState (configuration.name, configuration)
    return success
}
```

Listing 2. Pseudo-código parcial para emitir transações de configuração. O campo configurationIdentifier contêm um identificador único para a configuração.

O protótipo usa os certificados de autoridade (*Certificate Authorities* - CA) do Hyperledger Fabric para criar e gerenciar certificados digitais em cada nó da rede de corrente de blocos. Certificados digitais garantem auditabilidade e que somente nós certificados e autorizados podem participar da rede de corrente de blocos.

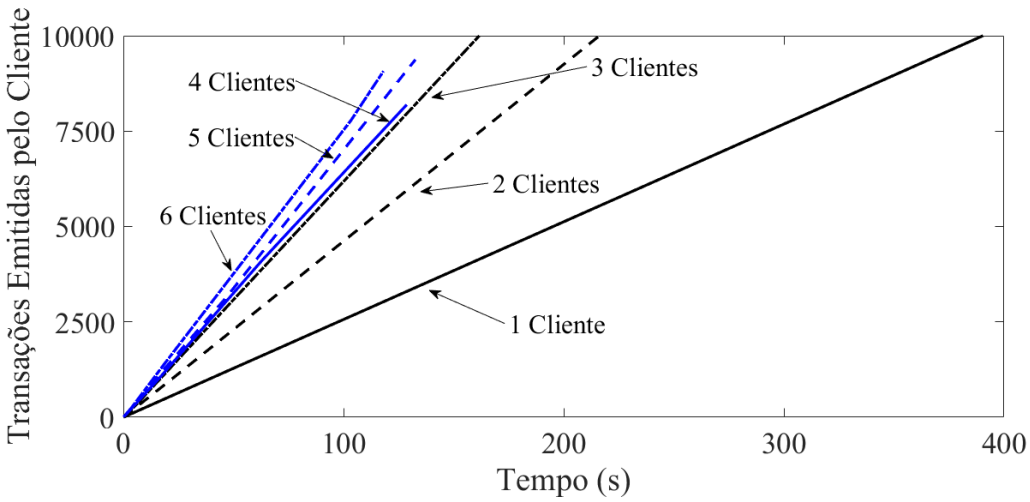


Figura 5. Tempo total decorrido para processar as transações na corrente de blocos à medida que o número de transações emitidas e o número de clientes aumentam. Os resultados mostram que a vazão aumenta com o número de clientes que emitem transações.

O protótipo implementa cada nó da rede do Hyperledger Fabric como um

contêiner em um único computador e envia transações simultaneamente. O trabalho avalia a vazão de transação enviando um número crescente de transações e medindo o tempo decorrido que os clientes precisam para propor todas as transações para a rede baseada em corrente de blocos. A Figura 5 apresenta o resultado da avaliação da taxa de transação do cliente. A taxa de transação na fatia de rede atinge um valor de pico de 71,31 transações por segundo no lado do cliente.

O trabalho ajusta as configurações de criação de bloco de acordo com avaliações de desempenho realizadas anteriormente no Hyperledger Fabric [Thakkar et al. 2018, Gorenflo et al. 2019]. O trabalho define o tamanho preferido do bloco sem cabeçalho em 99 MB, o número máximo de transações em um bloco em 10 e o tempo limite para inicializar uma rodada de consenso em dois segundos. Se alguma das condições for atendida, o nó ordenador inicia uma nova rodada de consenso e envia a proposta de novo bloco. Depois de alcançar o consenso, os nós acrescentam o novo bloco à sua cópia da corrente de blocos.

6. Conclusão

A tecnologia de fatia de redes fornece serviços fim-a-fim customizados através encadeando funções virtuais de rede entre infraestruturas de nuvens concorrentes em um ambiente distribuído multi-inquilino e multi-domínio sem confiança entre os pares. A alta programabilidade resultante proveniente da virtualização da rede expõe todo o tráfego a um maior número de ameaças. Portanto, é obrigatório precisamente definir e localizar falhas e usos indevidos para identificar agentes maliciosos que podem comprometer simultaneamente o bom comportamento e a qualidade de serviço de milhares de usuários.

O trabalho propõe uma arquitetura baseada em corrente de blocos para proteger fatias de redes customizadas. A multiplicidade de características diferentes exigidas por cada fatia impõe o uso de várias correntes de blocos personalizadas com diferentes números de participantes, estruturas de dados, tipo de transações, taxa de transferência de transação, tipo de consenso, tipo de redes, etc.

O trabalho implementa um protótipo de prova de conceito usando duas correntes de blocos que garantem proteger a criação de fatia e a atualização e migração de funções virtualizadas de rede. O trabalho usa a plataforma Hyperledger Fabric que facilita a criação de várias correntes de blocos em diferentes canais. Para trabalhos futuros, usaremos estruturas de dados otimizadas para melhorar a taxa de transação e usar um protocolo de consenso diferente.

7. Agradecimentos

Este trabalho foi financiado pela CAPES, CNPq, FAPERJ e FAPESP (2015/24514-9, 2015/24485-9 e 2014/50937-1).

Referências

Alvarenga, I. D., Rebello, G. A. F., and Duarte, O. C. M. B. (2018). Securing management, configuration, and migration of virtual network functions using blockchain. In *IEEE/IFIP NOMS*.

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, page 30. ACM.

Backman, J., Yrjölä, S., Valtanen, K., and Mämmelä, O. (2017). Blockchain network slice broker in 5G: Slice leasing in factory of the future use case. In *Internet of Things Business Models, Users, and Networks*, pages 1–8.

Bhamare, D., Jain, R., Samaka, M., and Erbad, A. (2016). A survey on service function chaining. *Journal of Network and Computer Applications*, 75:138–155.

Bordel, B., Orúe, A. B., Alcarria, R., and Sánchez-De-Rivera, D. (2018). An intra-slice security solution for emerging 5G networks based on pseudo-random number generators. *IEEE Access*, 6:16149–16164.

Boudguiga, A., Bouzerna, N., Granboulan, L., Olivereau, A., Quesnel, F., Roger, A., and Sirdey, R. (2017). Towards better availability and accountability for IoT updates by means of a blockchain. In *IEEE EuroS&PW*, pages 50–58.

Bozic, N., Pujolle, G., and Secci, S. (2017). Securing virtual machine orchestration with blockchains. In *CSNet'17*.

Capossele, A., Gaglione, A., Nati, M., Conti, M., Lazzeretti, R., and Missier, P. (2018). Leveraging blockchain to enable smart-health applications. In *IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI)*, pages 1–6.

Dolev, D. and Yao, A. (1983). On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208.

Gorenflo, C., Lee, S., Golab, L., and Keshav, S. (2019). Fastfabric: Scaling hyperledger fabric to 20,000 transactions per second. https://arxiv.org/pdf/1901.00910.pdf.

Halpern, J. and Pignataro, C. (2017). Service Function Chaining (SFC) architecture. RFC7665. `http://www.rfc-editor.org/rfc/rfc7665.txt`. Accessed Mar. 14, 2019.

Khettab, Y., Bagaa, M., Dutra, D. L. C., Taleb, T., and Toumi, N. (2018). Virtual security as a service for 5G verticals. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6.

Medhat, A. M., Taleb, T., Elmangoush, A., Carella, G. A., Covaci, S., and Magedanz, T. (2017). Service function chaining in next generation networks: State of the art and research challenges. *IEEE Comm. Mag.*, 55(2):216–223.

Ortega, V., Bouchmal, F., and Monserrat, J. F. (2018). Trusted 5G vehicular networks: Blockchains and content-centric networking. *IEEE Vehicular Technology Magazine*, 13(2):121–127.

Paladi, N., Michalas, A., and Hai-Van, D. (2018). Towards secure cloud orchestration for multi-cloud deployments. In *EuroSys-CrossCloud*.

Pattaranantakul, M., He, R., Song, Q., Zhang, Z., and Meddahi, A. (2018). NFV security survey: From use case driven threat analysis to state-of-the-art countermeasures. *IEEE Communications Surveys & Tutorials*.

Rawat, D. B. and Alshaikhi, A. (2018). Leveraging distributed blockchain-based scheme for wireless network virtualization with security and QoS constraints. In *International Conference on Computing, Networking and Communications (ICNC)*, pages 332–336.

Rebello, G. A. F., Alvarenga, I. D., Sanz, I. J., and Duarte, O. C. M. B. (2019). BSec-NFVO: A blockchain-based security for network function virtualization orchestration. In *IEEE International Conference on Communications (ICC)*. To be published.

Rosa, R. and Rothenberg, C. E. (2018). Blockchain-based decentralized applications for multiple administrative domain networking. *IEEE Communications Standards Magazine*, 2(3):29–37.

Sousa, J., Bessani, A., and Vukolić, M. (2017). A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. *arXiv preprint arXiv:1709.06921*.

Thakkar, P., Nathan, S., and Viswanathan, B. (2018). Performance benchmarking and optimizing hyperledger fabric blockchain platform. In *IEEE MASCOTS*, pages 264–276.

Thuemmler, C., Rolffs, C., Bollmann, A., Hindricks, G., and Buchanan, W. (2018). Requirements for 5G based telemetric cardiac monitoring. In *14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–4.

Valtanen, K., Backman, J., and Yrjölä, S. (2018). Creating value through blockchain powered resource configurations: Analysis of 5G network slice brokering case. In *IEEE WCNCW'18*, pages 185–190.

Yahiatene, Y. and Rachedi, A. (2018). Towards a blockchain and software-defined vehicular networks approaches to secure vehicular social network. In *IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 1–7.

Zawoad, S. and Hasan, R. (2016). SECAP: Towards securing application provenance in the cloud. In *2016 IEEE 9th International Conference on Cloud Computing*, pages 900–903.

Experimental Analysis of the Processing Cost of Ethereum Blockchain in a Private Network

Iago S. Ochôa[1, 2], Rafael A. Piemontez[1], Lucas A. Martins[1]
Valderi R. Q. Leithardt[1], and Cesar A. Zeferino[1]

[1]Laboratory of Embedded and Distributed Systems – LEDS
University of Vale do Itajaí – UNIVALI
Rua Uruguai, 458 – C. P. 360 – 88302-901 – Itajaí– SC – Brazil

[2]Serviço Nacional de Aprendizagem Industrial – SENAI
Av. 1 de maio, 670 – 88353-202 – Brusque – SC – Brazil

{iago.ochoa, piemontez, lucasmartins}@edu.univali.br

{valderi, zeferino}@univali.br

Abstract. *Blockchain technology has aroused the interest of researchers as it ensures security and privacy in decentralized applications. One of the platforms currently used for the development of new blockchain-based solutions is the Ethereum network. This article presents an experimental analysis of this network to identify its gas cost and performance for several contracts in a simulated private Ethereum network. The results obtained demonstrate that storage operations for large amounts of data can prevent the use of blockchain due to the high gas cost charged for this type of operation. Also, the experiments have pointed out that processing operations which do not store information have a small gas cost.*

1. Introduction

The blockchain is a technology that applies community validation to maintain the ledger content synced and replicated through all the nodes. The blockchain nodes consent on the validity of a sequence of blocks which have the record of a series of transactions, making the blockchain a record of ordered transactions [Dinh et al. 2018, Saraf and Sabadra 2018, Aste et al. 2017]. This characteristic of replication provides support to the central premises of the blockchain, which include security, trustability, transparency, and decentralization. These premises are highly attractive for the development of several applications, and the cryptocurrencies are the most known.

In addition to applications for the cryptocurrency market, several other applications from different domains have employed the blockchain technology to develop more robust solutions. Privacy control and identity management are examples of the services that have been used blockchain in the development of new solutions [Saraf and Sabadra 2018] and have found in the blockchain a way to manage user identities independently of a centralizing authority. We can also cite the supply chain management area as another major use of this technology [Miller 2018]. Because a supply chain has multiple links with several authorities, which are responsible for the flow of processes, there has been a great interest in applying blockchain to offer more transparency to the

customers. The employment of the blockchain in the development of new solutions has brought a great focus to this technology, perhaps more than the one expected at the time it was designed, and some of its features are reaching their physical limitations.

With the increasing popularity and adoption of cryptocurrencies, such as Bitcoin and Ethereum, these applications require that a fundamental concept present in the initial design of this technology be rethought, i.e., its scalability. As digital currencies become more commonplace on a day-to-day basis, the number of transactions has increased exponentially [Chauhan et al. 2018]. With the number of transactions increasing day by day, the blockchain becomes too bulky, and each node needs to store all transactions to validate if those transactions have not been "spent" before. This blockchain characteristic directly influences the ability of nodes to replicate information in a short period to not spend a very high amount of time in the data replication process instead of the transaction validation process, which also tends to be a current bottleneck of this technology.

In addition to the problem of scalability due to the volume of data, constraints related to the block size and the time interval for generating a new block influence the capacity and processing time within the network. According to [Zheng et al. 2017], in the Bitcoin blockchain, the processing capacity is only seven blocks per second, while [Blom and Farahmand 2018] shows that the Ethereum blockchain reaches a capacity close to twenty transactions per second. These numbers are remarkably lower than the major electronic payment methods currently used, such as Paypal and Visa, which deal with thousands of transactions per second [Chauhan et al. 2018]. For the continuation of this technology as a viable base for large applications, it is necessary to develop new techniques that allow the blockchain to overcome the problems being identified today, which has brought considerable attention to this subject.

As mentioned before, the need for data replication on all blockchain nodes is one of the critical features that compromise network scalability. This feature is assessed in the work of [Decker and Wattenhofer 2013], in which the authors evaluate how the delay in the data replication and process validation may be the bottlenecks causing the forks in the Bitcoin blockchain. The authors evaluate these points through unilateral simulations, forcing the limits of the Bitcoin protocol by merely varying the behavior of the network nodes. This work serves as a reference for several other studies focused on the scalability analysis of blockchain-based networks.

A framework for the performance analysis of private blockchains is presented in [Dinh et al. 2017]. This framework simulates different workloads applied to a network with a fixed number of nodes for throughput. For the scalability tests based on network growth, the authors defined a fixed frequency of requests and increased the number of nodes. As results, they presented the behavior of three private blockchains for validation purposes, namely: Ethereum, Parity, and Hyperledger. This work also served as a basis for the development of the work presented by [Pongnumkul et al. 2017], in which the authors similarly performed an analysis of two networks through diverse workloads. In this last work, three different types of operation were used to evaluate how networks behave through operations with different levels of complexity. Similarly, we have defined the proposal for our work.

In this paper, we analyze the processing cost of the Ethereum blockchain in a smaller private network. We investigate the behavior of this network using three types of operation with different complexities. We have divided the operations into Linear Operations, Data Manipulation, and Sorting Algorithms. Our work evaluates the gas consumption, as well as the impact of the input parameter value of the algorithm in that consumption. In this sense, we are mainly evaluating the consumption of the selected algorithms in the Ethereum network and the limits of the blockchain to process these algorithms. This experimental analysis is the main contribution of the work and in this context lies our analysis of network scalability. From these tests, it was possible to identify the behavior of the Ethereum network for different types of algorithm. It was also possible to obtain the average gas cost for each operation performed in the network.

The remainder of this paper is structured as follows. Section 2 describes briefly the Ethereum blockchain, while Section 3 presents the proposed analysis used to evaluate the blockchain, including the simulation and test applied. Section 4 presents and discusses the experimental results. Concluding, Section 5 gives the final remarks.

2. Ethereum Blockchain

The Ethereum platform was introduced to the world by [Buterin 2014]. Ethereum is not just a cryptocurrency, but it is also a programming language running in a blockchain. This platform allows users to create smart contracts and perform the transfer of cryptocurrency to other users by validating these contracts. According to [Mohanta et al. 2018], a smart contract is a computer program that self-checks and self-executes, and is anti-fraud. Such smart contracts are intended to facilitate negotiation between unknown persons, eliminating the need for a third-party to audit the negotiation. The existing applications for smart contracts are diverse, such as prevention of violation, financial services, and application of credit, among others. The validation of contracts and information contained in the blockchain employs mining processes.

The mining process consists of the execution of a consensus algorithm. This type of algorithm is used to validate the information transmitted in a distributed network. As an example, consider the double spending problem. A user named Bob, who has only ten coins, intends to send these coins to Alice and Ana at the same time. If there is no adequate consensus algorithm, Ana and Alice will each receive ten coins, generating double spending.

The Ethereum platform employs the PoW (Proof-of-Work) consensus algorithm, which was proposed in [Dwork and Naor 1992]. This algorithm consists of solving a cryptographic problem that satisfies all the data present in the blocks of the blockchain. In a consolidated network, the PoW algorithm improves security, but it is energy inefficient. According to [Kim and Kim 2018], the energy consumption generated by the Ethereum and Bitcoin blockchains already surpasses the energy consumption of Syria, which is the 72nd country with the highest consumption of electricity in the world.

Another consensus algorithm is the PoS (Proof-of-Stake), which was proposed by [Sunny King 2012] to solve the energy issues and guarantee security to the information contained in the blockchain. This algorithm defines the miner of the next block by the

amount of cryptocurrency that a node has. The user with the highest number of cryptocurrency could be defined as the miner of the next block. However, to avoid the monopoly of the network by whoever has the higher amount of cryptocurrency, the process uses a random number generator to define the mining node. This algorithm becomes advantageous over PoW because it spends less electricity on the mining process. Currently, the Ethereum platform uses an optimized version of PoW, but it is estimated that it will be completely migrated to PoS in 2019.

When the network confirms the validation of a block, the miners of the block receive a specific value in Ether for the work done. It is necessary to mention that the cryptocurrency of the Ethereum platform has pseudonyms (see Table 1) that are also used to charge fees for users using the Ethereum blockchain. Each operation performed on the Ethereum platform has a cost called *gas* [Wood 2014]. The cost of gas varies for each operation. For instance, storing a contract on the Ethereum platform has a cost of 32,000 gas, while storing a value with 256-bit size has a cost of 20,000 gas.

Table 1. Ethereum pseudonyms

Multiplier	Pseudonym
10^0	Wei
10^{12}	Szabo
10^{15}	Finne
10^{18}	Ether

The users of the network set the gas price. At the time of the development of this work, the cost of a gas corresponded to 3.6 GWei. It is worth noting that the gas price varies according to the operation of the network. The amount previously shown is the standard gas price, which means that the user paying that price for the used gas will have their transaction confirmed on the Ethereum platform within 30 minutes. If he/she needs to confirm the transaction be more quickly in the blockchain, he/she should pay 15 GWei perused gas, and the transaction will be confirmed in less than 2 minutes. Information regarding the Ethereum platform values can be obtained in [EthStats 2018].

With the popularization of the platform, the number of users in the Ethereum network has been growing exponentially. Figure 1 shows the increase of addresses in the platform since 2015. Nowadays (Dec. 2018), the platform has about 48 million active addresses.

3. Implementation

This section describes the methods used to evaluate the processing costs of the Ethereum network.

We have implemented eight contracts of three different classes: Linear Operations, Data Manipulation, and Sorting Algorithms. As development and experimental platform, we have employed Truffle Suite to implement, compile, and run the tests, and Ganache software to build the private blockchain network.

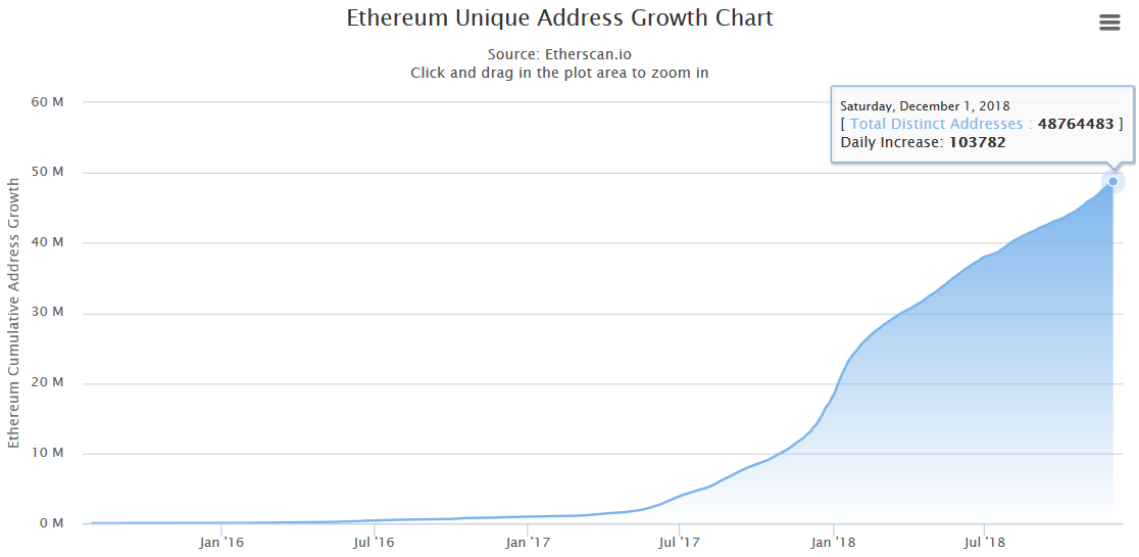


Figure 1. Address growth on the Ethereum platform from 2015 to 2018 (data obtained using [Etherscan 2018]).

We have employed the Linear Operations class for assessing and identifying whether the Ethereum network has a pattern of growth in the gas cost and the performance of data processing. This class is composed of two contracts named Single_Loop and Double_Loop. These contracts execute one and two nested loops, respectively, which do not perform any computation or access to the blockchain (see Algorithms 1 and 2).

Algorithm 1: Single_Loop

Input : n, Number of iterations
for $i \in \{0, \ldots, n-1\}$ **do**
end
return 1

Algorithm 2: Dual_Loop

Input : n, Number of iterations in each loop
for $i \in \{0, \ldots, n-1\}$ **do**
  for $j \in \{0, \ldots, n-1\}$ **do**
  end
end
return 1

The Data Manipulation class was intended to assess the gas cost and performance of operations that update the blockchain. This class comprises three contracts: (*i*) Store_nInt, which performs several stores of an integer variable into the blockchain; (*ii*) Store_nChar, which updates several times a char variable in the blockchain; and (*iii*) Store_String, which saves a string into the blockchain. Algorithms 3–5 illustrate these contracts.

Algorithm 3: Store_nInt

Input : n, Number of store operations in the blockchain
Data: x, Integer variable in the blockchain
$x \leftarrow 0$
for $i \in \{0, \ldots, n-1\}$ **do**
| $x \leftarrow x + i$
end
return 1

Algorithm 4: Store_nChar

Input : n, Number of store operations in the blockchain
Data: x, Char variable in the blockchain
for $i \in \{0, \ldots, n-1\}$ **do**
| $x \leftarrow Generate_Char()$
end
return 1

Algorithm 5: Store_String

Input : n, Size of the string to be stored in the blockchain
Data: x, String variable in the blockchain
$x \leftarrow Generate_Char_Array(n)$
return 1

The contracts of the Ordering Algorithms class were planned to evaluate the performance of the network for operations requiring high processing. This class is composed of three algorithms (Bubble_Sort, Quick_Sort, and Merge_Sort) that perform the ordering of the values of a linear array. The contracts are also responsible for generating an unsorted array before ordering it.

Table 2 summarizes the computational complexity of each contract employed in the experiments. It is worth highlighting that all the contracts of the Data Manipulation class have low complexity because they focus on identifying the gas cost related to the updating of the blockchain.

Table 2. Computational complexity of the contracts

Contract name	Contract class	Best Case	Worst Case
Single_Loop	Linear Operations	$O(n)$	$O(n)$
Double_Loop	Linear Operations	$O(n^2)$	$O(n^2)$
Store_String	Data Manipulation	$O(n)$	$O(n)$
Store_nChar	Data Manipulation	$O(n)$	$O(n)$
Store_nInt	Data Manipulation	$O(n)$	$O(n)$
Bubble_Sort	Ordering Algorithms	$O(n)$	$O(n^2)$
Quick_Sort	Ordering Algorithms	$O(n.\log n)$	$O(n^2)$
Merge_Sort	Ordering Algorithms	$O(n.\log n)$	$O(n.\log n)$

3.1. Simulations and Tests

After defining the contracts, we have applied tests to evaluate their performance. Each contract has a single input parameter (n) whose meaning is defined in Table 3.

Table 3. Meaning of the input parameter (n) in each contract

Contract	Input parameter
Single_Loop	Number iterations of the loop
Double_Loop	Number of iterations in each loop
Store_String	Size of the string to be stored into the blockchain
Store_nChar	Number of times it stores a char into the blockchain
Store_nInt	Number of times it stores an integer into the blockchain
Bubble/Quick/Merge_Sort	Size of the array to be ordered

The algorithms listed above allowed to evaluate the results regarding the processing and gas cost of the network.

Each test was performed sequentially through a script written to request 12 operations of each contract. The parameter n was set to vary from 2 to 4096, in discrete steps based on powers of 2. All the tests were performed five times to obtain an average value in the processing time.

The Ethereum network allows building three types of function for contracts. Standard functions allow to read and store data in the blockchain. View functions allow only to view the data in the blockchain. Finally, Pure functions, which do not allow reading and storing in the blockchain, but allow to use the network as a processing source. Due to this difference, two test groups were developed, according to Table 4. The Pure group was built to evaluate contracts that do not make any modification in the database. The Standard group was developed to evaluate the contracts that make modifications in the database and to identify differences in the gas cost. As the contracts of the Data Manipulation class modify the database, we did not use it in the Pure group tests.

Table 4. Evaluation type x Test group

Evaluation type	Pure group	Default group
Linear Operations	•	•
Data Manipulation		•
Sorting Algorithms	•	•

The tests were performed on a computer with an Intel® i5-5200U processor, with 4 cores, and 8GB of RAM running Manjaro Linux distribution.

4. Experimental Results

This section presents the results of the experiments carried out with the Ethereum network for the three different classes of contract. The analysis is performed based on the gas cost and runtime (i.e., the execution time) of each contract.

4.1. Linear Operations

Figure 2 presents the gas cost and the runtime for the Linear Operations contracts. The contracts of the Pure group did not consume gas, and they are not presented in the Gas Cost chart. We observe that the standard Single_Loop contract has a linear growth pattern, while the Dual_Loop presents a quadratic growth, as described in Table 2. Also, we notice that the Dual_Loop contract exceeded the limit of gas cost and was not able to execute more than 256 iterations per loop. Moreover, it is worth noting that there is no significant difference in execution time between the Pure and the Standard groups.

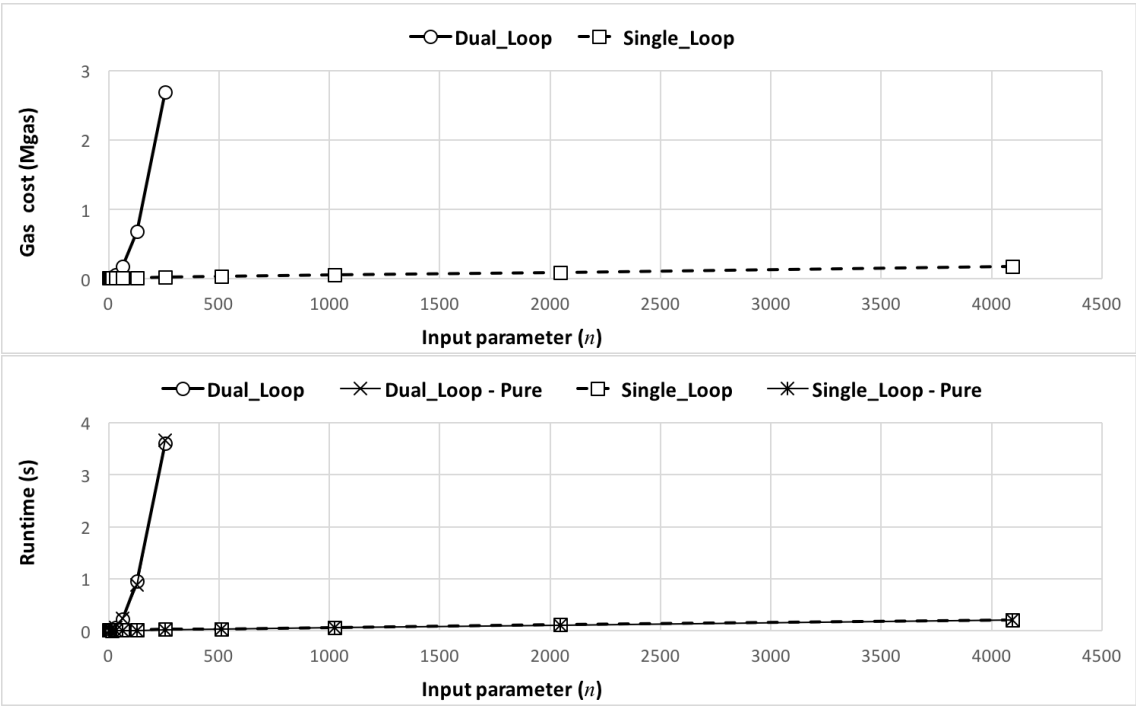


Figure 2. Gas cost and runtime of the Linear Operations contracts.

Figure 3 presents the runtime gas cost, which is a metric that expresses the ratio of gas consumption in time; it is given in Mgas per second (or Mgas/s). The average cost of the Single_Loop contract equals 0.87 Mgas/s, with a minimum of 0.66 Mgas/s and a maximum of 1.17 Mgas/s.

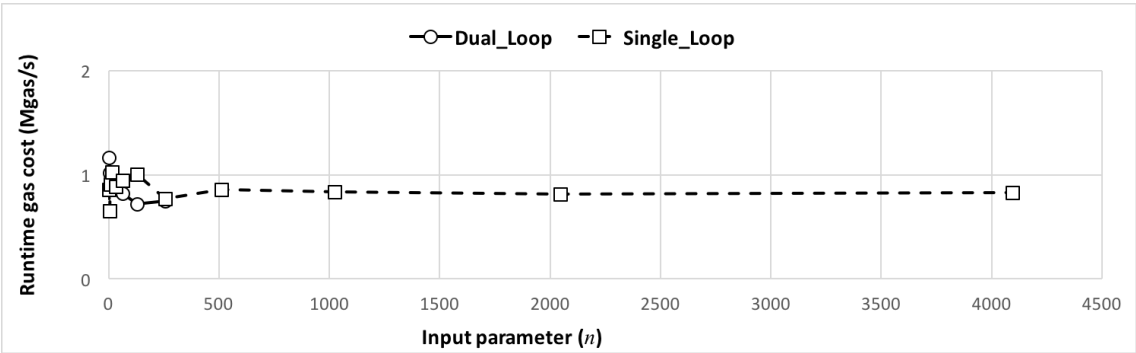


Figure 3. Runtime gas cost for Linear Operations contracts.

4.2. Data Manipulation

Figure 4 presents the gas cost and runtime results for the Data Manipulation contracts. Figure 5 identifies the cost per update for each one of these contracts. We observe that these algorithms also have a linear behavior in network processing and gas cost for the operations executed. We also notice that the contract that store the shorter data (Store_Char) consume more gas per update, but have better scalability.

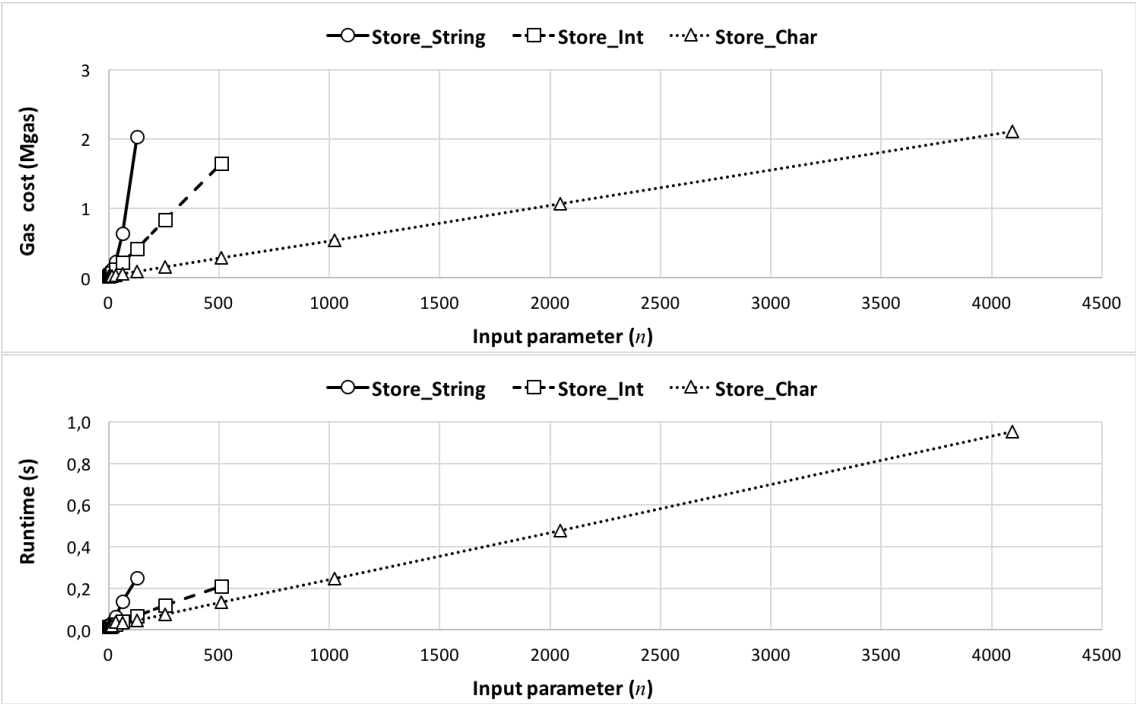


Figure 4. Gas cost and runtime of Data Manipulation contracts.

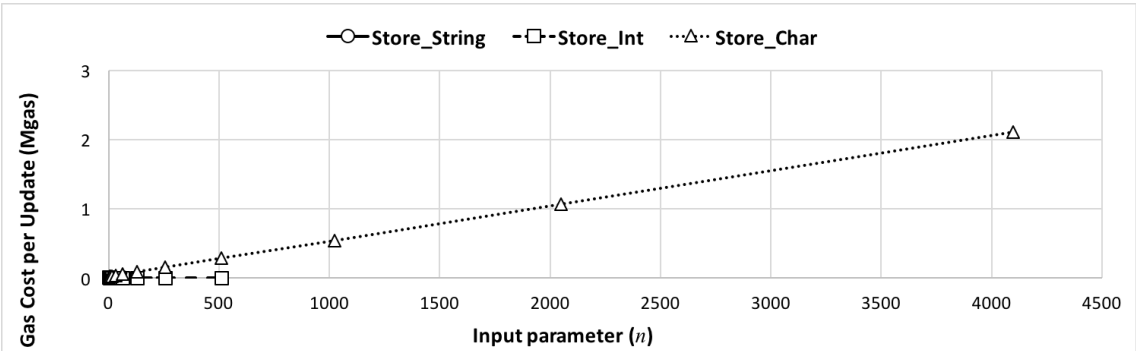


Figure 5. Gas cost per update.

The experiments shown that the Data Manipulation contracts have a high runtime gas cost (Figure 6). Its minimum cost (1.18 Mgas/s) is greater than the higher cost of the Linear Operation contracts (1.16 Mgas/s), reaching a maximum cost of 8.09 Mgas/s in StringChange operations.

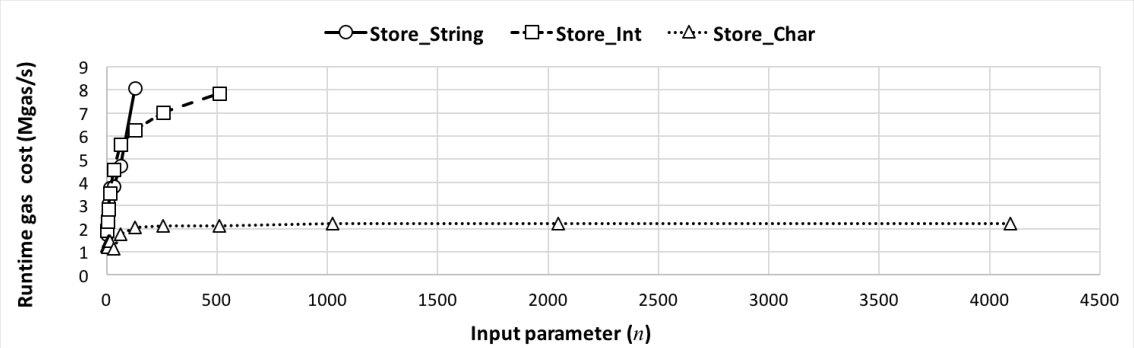


Figure 6. Runtime gas cost for Data Manipulation contracts.

4.3. Sorting Algorithms

As expected, the Sorting Algorithm contracts have the longest processing time, reaching the cost of 2.8 Mgas in 4 seconds of processing in BubbleSort ordering contracts. As Figure 7 shows, this same contract is only able to order vectors of up to 128 items because tests with more operations exceeded the gas limit, different from the much more complex QuickSort contract, which was able to perform all the orderings in the experiments. As in the Linear Operations class, the execution time of the Pure group implementations is similar to the one of the Standard implementations.

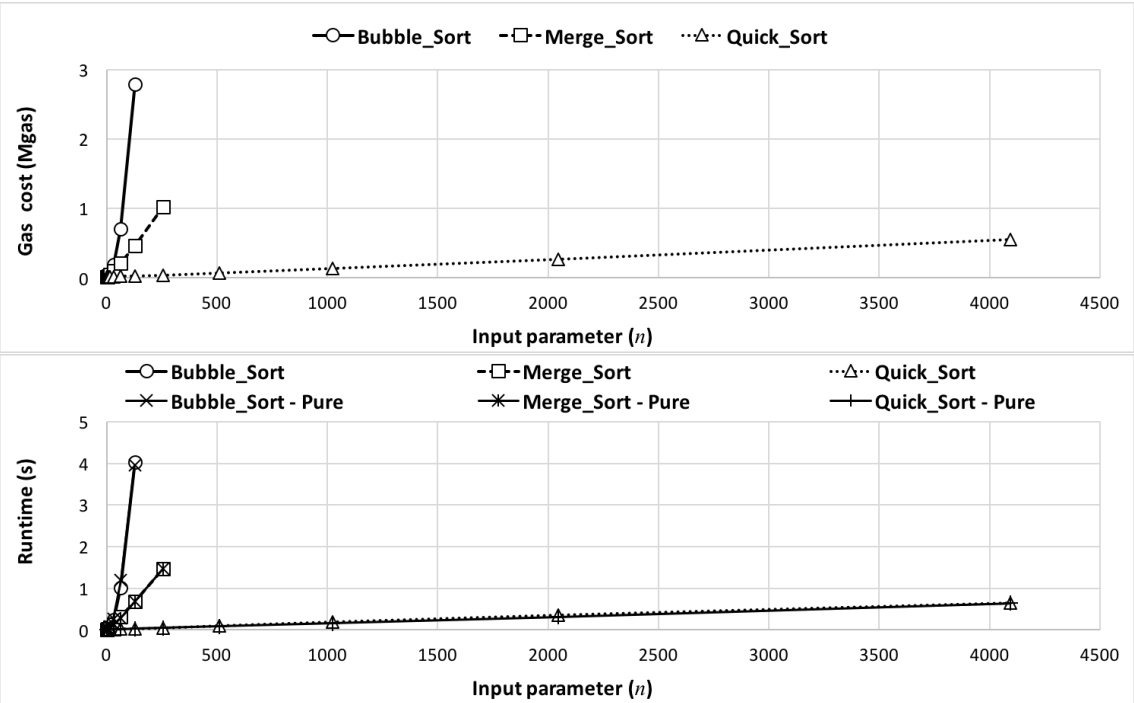


Figure 7. Gas cost and runtime for Ordering Algorithms contracts.

Although the gas cost and the processing time are higher than the ones observed in the other classes, this class has an average runtime gas of 0.6 Mgas/s. This cost is close to the one obtained with the Linear Operations algorithms and lower than the Data Manipulation class. These results are evidence that contracts that perform operations in the blockchain are much more costly, as Figure 8 shows.

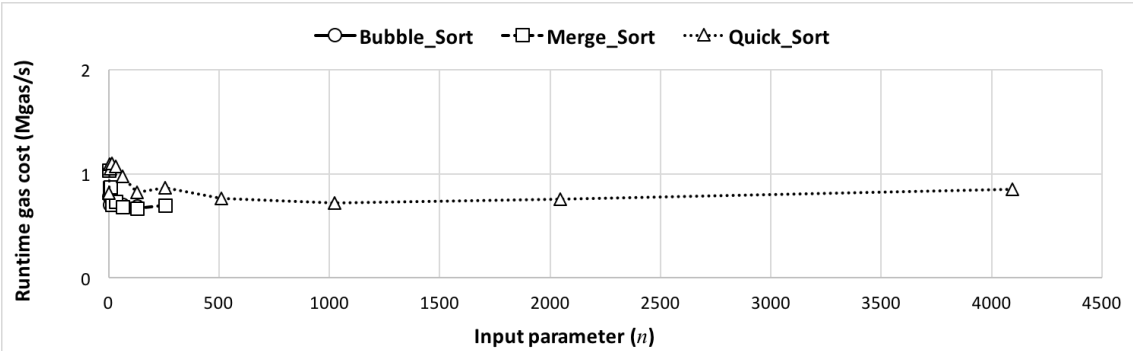


Figure 8. Runtime gas cost for Ordering Algorithms contracts.

The results of this study questioned the ability of a private network as a processing source and base for blockchain data storage, which, despite a solution for simple computational problems, brought a high data processing time in the tests performed and a high gas cost in operations that carry out large amounts of records in the blockchain.

5. Conclusions

In this paper, we have presented an experimental analysis of the processing cost of the Ethereum blockchain in a private network. From the tests performed, we have observed that algorithms that use the Ethereum blockchain for data processing are not efficient because some operations take too many time to process a request. For instance, it takes around four seconds to sort an array of 128 positions. While using it for linear data processing is not good, making use of the Ethereum blockchain to process data in parallel might be useful. However, further studies on this topic are necessary to confirm this hypothesis.

The process of designing and creating a smart contract must be carefully thought to decide which types of data are going to be used to store the information into the blockchain. From the tests performed using data manipulation, we have identified that requesting changes of a single character of a text variable is four times more expensive than requesting changes of integers. Furthermore, we have found that the cost for manipulating changes to a string of 128 characters is six times higher than to manipulate only two characters of text, costing more than two thousand gas for a single operation. Based on that, we can conclude that using the Ethereum blockchain to store a significant amount of data is not the ideal scenario, although we can use it to store thousands of small registers for the same price as a big one. As an overall conclusion, we can assume that the Ethereum blockchain presents a lack of scalability because the network struggled to handle a small number of complex operations. The source code used in the tests can be found in the LEDS repository provided in GitHub [Ochôa et al. 2019].

As future work, we intend to simulate cryptocurrency operations into the network, because they are supposed to be the core application for this network. After that, we are going to be able to compare if the complexity of the transactions used in this work relates to the expected by the Ethereum blockchain core design. Also, we plan to perform the same tests on the Ethereum public network to compare the gas and monetary costs.

Furthermore, we intend to perform an analysis of the gas costs for real-world applications to assess how costly these applications are in an Ethereum-based network.

Acknowledgments

This work was financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001 and by Fundação de Amaparo à Pesquisa e Inovação do Estado de Santa Catarina – Brasil (FAPESC) – Grant No. 2019TR169.

References

Aste, T., Tasca, P., and Di Matteo, T. (2017). Blockchain technologies: The foreseeable impact on society and industry. *Computer*, 50(9):18–28.

Blom, F. and Farahmand, H. (2018). On the scalability of blockchain-supported local energy markets. In *International Conference on Smart Energy Systems and Technologies*, SEST'18, pages 1–6.

Buterin, V. (2014). Ethereum: A next-generation smart contract and decentralized application platform. `https://etherscan.io/chart/address`. Accessed: 29-11-2018.

Chauhan, A., Malviya, O. P., Verma, M., and Mor, T. S. (2018). Blockchain and scalability. In *IEEE International Conference on Software Quality, Reliability and Security Companion*, QRS-C'18, pages 122–128.

Decker, C. and Wattenhofer, R. (2013). Information propagation in the bitcoin network. In *IEEE 13th International Conference on Peer-to-Peer Computing*, P2P'13, pages 1–10.

Dinh, T. T. A., Liu, R., Zhang, M., Chen, G., Ooi, B. C., and Wang, J. (2018). Untangling blockchain: A data processing view of blockchain systems. *IEEE Transactions on Knowledge and Data Engineering*, 30(7):1366–1385.

Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., and Tan, K.-L. (2017). Blockbench: A framework for analyzing private blockchains. In *ACM International Conference on Management of Data*, SIGMOD/PODS'17, pages 1085–1100.

Dwork, C. and Naor, M. (1992). Pricing via processing or combatting junk mail. In *Annual International Cryptology Conference*, CRYPTO'92, pages 139–147.

Etherscan (2018). Ethereum unique adress growth chart. `https://etherscan.io/chart/address`. Accessed: 29-11-2018.

EthStats (2018). Ethereum statistics. `https://ethstats.net/`. Accessed: 29-11-2018.

Kim, S. and Kim, J. (2018). POSTER: Mining with Proof-of-Probability in blockchain. In *Asia Conference on Computer and Communications Security*, ASIACCS'18, pages 841–843.

Miller, D. (2018). Blockchain and the Internet of Things in the industrial sector. *IT Professional*, 20(3):15–18.

Mohanta, B. K., Panda, S. S., and Jena, D. (2018). An overview of smart contract and use cases in blockchain technology. In *9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–4.

Ochôa, I. S., Piemontez, R. A., Martins, L. A., Leithardt, V. R. Q., and Zeferino, C. A. (2019). Experimental analysis of the processing cost of Ethereum blockchain in a private network: source code. `https://github.com/UNIVALI-LEDS/WBlockchain-2019`.

Pongnumkul, S., Siripanpornchana, C., and Thajchayapong, S. (2017). Performance analysis of private blockchain platforms in varying workloads. In *26th International Conference on Computer Communication and Networks*, ICCN'17, pages 1–6.

Saraf, C. and Sabadra, S. (2018). Blockchain platforms: A compendium. In *IEEE International Conference on Innovative Research and Development (ICIRD)*, ICIRD'18, pages 1–6.

Sunny King, S. N. (2012). PPCoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper*.

Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32.

Zheng, Z., Xie, S., Dai, H., Chen, X., and Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. In *IEEE International Congress on Big Data*, BigData Congress'17, pages 557–564.

Um Estudo Sobre o Misterioso e Arriscado Mundo dos Misturadores de Criptomoedas

Rodolfo da Silva Costa[1], **Rostand Costa**[2]

[1]Centro de Informática (CI) – Universidade Federal da Paraíba (UFPB)
R. dos Escoteiros, s/n – Mangabeira, João Pessoa – PB – Brasil

[2]Laboratório de Aplicações de Vídeo Digital (LAVID) – Universidade Federal da Paraíba (UFPB)
R. dos Escoteiros, s/n – Mangabeira, João Pessoa – PB – Brasil

rodolfo.informatica@gmail.com, rostand@lavid.ufpb.br

Abstract. *The new paradigm that emerged with cryptocurrencies in the last decade allows higher transparency of the amounts involved in a specific digital financial asset and allows that all transactions executed can also be visualized. To ensure anonymity, a significant number of cryptocurrency users have tried to hide the origin of their digital assets. In order to cope with this demand, new specialized services, such as cryptocurrency mixers or tumblers, have emerged. In this paper, we present a study on how these services work and evaluate their dynamics and efficacy on unlinking the origin from the destination wallets. Preliminary results point to a high risk business model that moves enormous amounts of money and is debatable from an ethics point of view.*

Resumo. *O novo paradigma que emergiu com as criptomoedas na última década permite uma maior transparência nas grandezas envolvidas em um determinado ativo financeiro digital e possibilita que todas as transações realizadas possam ser visualizadas. Para garantir o anonimato, uma quantidade expressiva de usuários de criptomoedas tem buscado esconder a origem de seus ativos digitais. Para atender tal demanda, surgiram serviços especializados, chamados cryptocurrency mixers (ou tumblers). O objetivo deste trabalho é realizar um estudo acerca do funcionamento de tais serviços para avaliar a sua dinâmica de funcionamento e sua eficácia na desvinculação da carteira de origem com a carteira de destino. Os resultados preliminares apontam para um negócio de alto risco e que movimenta valores bilionários através de uma ética discutível.*

1. Introdução

Surge em 2008 uma nova proposta de moeda eletrônica que viria para revolucionar a forma como valores monetários podem ser transferidos. Com o artigo intitulado *Bitcoin: Um sistema ponto-a-ponto de dinheiro eletrônico*, [Nakamoto 2008] descreve um "sistema de pagamento eletrônico baseado em provas criptográficas ao invés da confiança mútua, permitindo quaisquer duas partes dispostas transacionarem diretamente entre si sem a necessidade da intermediação de uma terceira parte". Utilizando o sistema apresentado, torna-se possível realizar transações financeiras através da internet, sem a necessidade de um banco ou qualquer outro intermediador.

Nakamoto propôs um livro caixa público e compartilhado que viria a chamar-se *Blockchain*. Este é utilizado para registrar de forma pública todas as transações entre as partes, representadas por endereços de carteira. Nestas transações, é possível identificar quais moedas foram transferidas de uma conta para outra. Desta forma, protegendo o sistema contra adulteração de contas, gastos duplos ou criação de novos ativos.

Esse novo paradigma permite transparência nas grandezas envolvidas em um determinado ativo financeiro digital. Por concepção, a blockchain permite que qualquer um possa analisar todas as transações já realizadas no sistema, além de consultar o saldo que cada carteira contém e qualquer outra relação que possa existir entre os endereços de carteiras.

O fortalecimento e o surgimento destas novas moedas estavam ocorrendo para além do mercado financeiro tradicional, deixando governos sem a arrecadação de impostos. Mas o uso de exchanges, onde é possível trocar moedas convencionais por moedas digitais, permitiu aos países realizarem a cobrança de tais impostos. Uma das primeiras iniciativas norte americanas de taxação das criptomoedas ocorreu em novembro de 2017 [1], quando o governo "verificou que a quantidade das declarações no imposto de renda não se alinhava com a popularidade emergente das moedas digitais".

A Receita Federal Brasileira, em 31 de Outubro de 2018, anunciou que exigirá das corretoras declarações mensais de todas as operações de vendas de criptomoedas[2]. Outros países também estão iniciando processos no sentido de cercear o anonimato das blockchains, a China [3], onde órgãos reguladores da internet "propõem que todas as operações com a ferramenta (blockchain) realizadas no país sejam identificadas".

Para auxiliar nessa regulação, governos e empresas necessitam de ferramentas que forneçam condições de analisar as blockchains e identificar transações e indivíduos. Surge então um mercado de desenvolvimento dessas ferramentas capazes de colaborar não apenas com questões de fiscais mas também no combate ao crime. Por outro lado, na perspectiva de recuperar a característica de privacidade desejada na origem da Bitcoin, nasce um tipo de serviço de troca e embaralhamento de criptomoedas chamado de *cryptocurrency mixers* ou *cryptocurrency tumblers*.

Nesse cenário, encontramos um ambiente polarizado onde um grupo de instituições buscam identificar os usuários de criptomoedas e o outro grupo busca garantir a privacidade e o anonimato dos mesmos. Sendo assim, o objetivo desta pesquisa é realizar uma análise do funcionamento dos serviços de mixagem de Bitcoins que buscam anular as ferramentas de análise das blockchains. Os objetivos específicos desta primeira fase são avaliar a eficiência destes mecanismos para apagar os rastros de origem e movimentação das moedas, os possíveis riscos envolvidos no serviço e quais as motivações para uso dos mesmos.

Durante o desenvolvimento da pesquisa, foram executados experimentos usando

[1] "Coinbase ordered to give the IRS data on users trading more than $20,000". https://techcrunch.com/2017/11/29/coinbase-internal-revenue-service-taxation/.

[2] "Receita aperta fiscalização no uso de criptomoedas". https://economia.estadao.com.br/noticias/mercados,receita-aperta-fiscalizacao-no-uso-de-criptomoedas,70002575506.

[3] "Operações com blockchain deixarão de ser anônimas na china". https://meiobit.com/391950/china-blockchain-regulacao-fim-anonimato/.

quatro diferentes serviços de mixagem de Bitcoins, nos quais foram rastreados os caminhos que as moedas percorreram e suas transações dentro das reservas das mixers. Descobriu-se uma gigantesca quantidade de moedas circulando pelas carteiras destes serviços e alguns riscos inerentes a sua utilização. Apesar das evidências encontradas nesta investigação, o segmento milionário das misturadoras continua quase que completamente ignorado pela academia. Isso explica a existência de poucos artigos científicos sobre o tema.

O presente documento está estruturado de forma que na Seção 2 é apresentado o funcionamento básico dos *mixers* de criptomoedas. Na Seção 3, é detalhado como os experimentos foram realizados, enquanto que a análise e discussão dos resultados é feita na Seção 4. Na Seção 5 apontamos alguns trabalhos relacionados com a privacidade do Bitcoin, os quais foram utilizados como referencial nesta pesquisa. Finalmente, na Seção 6 são apresentadas as nossas considerações finais.

2. Mixers de Criptomoedas

Como discutido em [Böhme et al. 2015], cada Bitcoin individual pode ser facilmente rastreado através de todas as transações nas quais foi usado e, portanto, até o início de sua circulação. Todas as transações de Bitcoin são legíveis por todos em registros armazenados por estruturas de dados amplamente replicadas.

Os serviços de mistura e mescla de criptomoedas funcionam trocando as moedas enviadas de seus clientes por outras moedas que estejam em suas reservas, devolvendo-as, após uma série de operações de transferências coordenadas, em novos endereços de carteira de forma a não possuir relação com a carteira original (Figura 1). O objetivo é fazer com que não seja mais factível identificar o endereço de destino das criptomoedas a partir do endereço de origem. Os mixers, ao custo de uma pequena taxa, prometem garantir a privacidade do detentor de ativos através do embaralhamento da movimentação entre a carteira de origem e a carteira de destino.

Para promover o anonimato para as carteiras finais, os serviços misturadores fornecem ao cliente um endereço temporário de entrada para o qual devem ser enviadas as moedas que serão embaralhadas/trocadas. As interfaces de utilização dos sistemas permitem algumas personalizações de configuração, tais como:

- Dividir o destino das moedas em mais de uma carteira;
- Escolher o tempo de atraso entre o envio e o recebimento das novas moedas;
- Valor da tarifa a ser paga pelo serviço, que pode variar de 0,5% a 5%;

Após o envio das moedas para o endereço de entrada da misturadora, a entidade aguardará o tempo programado para iniciar as transações de devolução das novas moedas. Cada prestador desses serviço utiliza algoritmos próprios para evitar que as moedas sejam retornadas para os proprietários originais. São utilizadas então moedas da reserva da misturadora, as quais foram recebidas de outros clientes ou investidores do serviço.

2.1. Mixers em Operação

Em buscas rápidas na Internet é possível encontrar a oferta de uma série de serviços misturadores de criptomoedas que estão em funcionamento[4]. Cada um deles possui suas

[4]Neste trabalho foram testados e analisados os quatro serviços de mistura mais referenciados: *Bestmixer.io*, *Blender.io*, *Helix Light Grams*, *BTC Blender*.

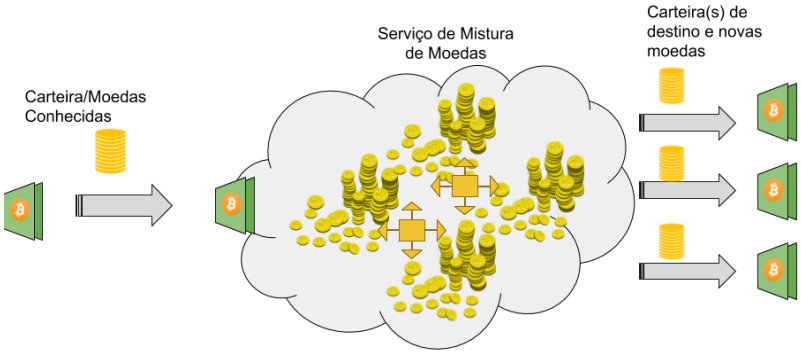


Figura 1. Serviço de Mistura de Moedas

características e particularidades envolvendo tarifas, tipos de moedas e algoritmos de embaralhamento. Muitos podem ser acessados através da Internet aberta, enquanto outros permitem acesso apenas na *deep web*, através da rede TOR, ou ambos.

Dentre as *mixers* pesquisadas, a **Bestmixer** destacou-se por trazer um novo nível de segurança para o segmento. O projeto foi apresentado no fórum Bitcointalk[5] em 16 de Março de 2018, prometendo o máximo de anonimidade para criptomoedas após a mixagem. Na ocasião, seus criadores afirmaram que a maioria dos serviços de mistura existentes anteriormente não garantiam o anonimato das transações e que seria possível identificar qual a carteira de saída a partir da transação de entrada. Eles haveriam disponibilizado para utilização uma ferramenta chamada *CAE-check*[6] ou *Coin Anonymisation Event-check* para comprovação pelos próprios clientes da ineficiência dos serviços.

É fato que após apontar as vulnerabilidades existentes nos mixers e entrar no mercado para fazer concorrência, a Bestmixer iniciou um novo ciclo de serviços mais eficientes em apagar rastros de moedas digitais. A equipe Bestmixer.io afirmou ter encontrado vulnerabilidade nos seguintes serviços existentes à época: Coinmixer.se, Bitmix.biz, Privcoin.io, Cryptomixer.io. Destes, apenas o Coinmixer foi descontinuado, os demais serviços continuam disponíveis após as falhas serem corrigidas.

3. Planejamento de Experimentos: Escopo e Dinâmica

Para este trabalho foram realizados experimentos utilizando mixers de criptomoedas com os objetivos de levantar dados, entender seu funcionamento e descobrir os possíveis riscos envolvidos. Na primeira etapa, realizou-se uma seleção de mixers, onde foram escolhidos 4 serviços, estes foram submetidos a testes de funcionalidades. Os dados resultantes dos testes, foram analisados na blockchain, e os relacionamentos entre as carteiras envolvidas foram rastreados; por fim, ocorreu a análise dos dados.

Durante a etapa de seleção dos serviços[7], foi utilizada uma lista contendo "9 me-

[5]https://bitcointalk.org/index.php?topic= 3140140.0

[6]Nós procuramos a ferramenta para estudos e testes mas a mesma foi removida do endereço web oficial (https://bestmixer.io/en/cae-check), bem como não pode ser encontrada via buscadores da internet. Quando questionados por e-mail sobre a ferramenta *CAE-check* em 27 de Novembro de 2018, os desenvolvedores responderam que "Quase todos os mixers com uma vulnerabilidade que encontramos, consertaram seus sistemas. Removemos essa funcionalidade do nosso site".

[7]Não foram encontrados textos acadêmicos que ranqueiam os serviços dos mixers.

lhores serviços misturadores de Bitcoin" como referência[8]. Apesar do título do texto mencionar nove serviços, o autor lista sete mixers como sendo os "melhores e mais populares": BestMixer, PrivCoin.io, Bitcoin Blender, CryptoMixer, Bitcoin Fog, Blender.io e MixTum.io e outros 3 serviços (Grams Helix, BTC Blender, Coinmixer) foram rotulados como sendo "não aprovados" e de "muito grande risco".

Os experimentos foram realizados com dois dos serviços considerados "melhores e mais populares"[8]: **Bestmixer** e **Blender.io**. Bem como, dois dos considerados de grande risco: **Helix Light Grams** e **BTC Blender**.

3.1. Realizando Transações de Mixagem

A configuração padrão para os testes possui os seguintes parâmetros: 0,001 Bitcoin é enviado para a mixer; as novas moedas são recebidas em 2 novos endereços de carteira; cada uma delas deveria receber um valor próximo a 50% do valor final; a taxa de serviço utilizada é a sugerida pela mixer, mas não superior a 2% do valor enviado; o tempo de atraso entre as confirmação do envio e o recebimento das novas moedas é de 2 horas para cada carteira de destino.

O serviço BTCBlender utilizou por padrão um valor randômico para a tarifa. No caso deste experimento, o valor foi 1%. Já a mixer Helix Light Grams não permitiu configuração da taxa de serviço, fixando o valor em 2,5%.

Os envios foram realizados no período entre 04 de outubro e 16 de novembro de 2018, sendo uma transferência para os serviços Helix Light Grams e BTC Blender e duas transferências para Bestmixer e Blender.io. O resumo pode ser encontrado na Tabela 1.

Tabela 1. Configurações dos experimentos e saldos pós mistura

Data	Mixer	Taxa	Valor Enviado	Total Recebido	Total Consumido	% Consumido
04/10/2018	Bestmixer	1,937%	0,001	0,00096951	0,00003049	3,05%
07/11/2018	Bestmixer	1,933%	0,001	0,00084593	0,00015407	15,41%
31/10/2018	Blender.io	1,35%	0,001	0,0007865	0,0002135	21,35%
16/11/2018	Blender.io	1,35%	0,001	0,0007865	0,0002135	21,35%
31/10/2018	BTCBlender	1%	0,001	0	0,001	100%
31/10/2018	Helix	2,5%	0,001	0	0,001	100%

A etapa seguinte ao processo de testes dos serviços é a análise da blockchain. Nesta etapa analisou-se os endereços das carteiras de entrada das mixers e seguiu-se o fluxo de transações a partir daquele endereço, em busca dos endereços que pertenceriam às reservas do serviço. O caminho inverso também foi rastreado buscando encontrar a origem das novas moedas recebidas.

3.2. Rastreando as Transações de Embaralhamento

O rastreio dos caminhos tomados pelas moedas após a entrada na rede das mixers, assim como o rastreio reverso, a partir da carteira de destino, foi realizada por dois scripts[9], desenvolvidos em Python, utilizando o módulo *blockexplorer* da *api-v1-client-python*[10].

[8]disponível em https://cryptalker.com/best-bitcoin-tumbler/
[9]Disponível para download em: http://bit.ly/mixersscript
[10]Disponível em https://github.com/blockchain/

Para a execução dos scripts são enviados como argumentos de entrada um endereço de carteira **E**, um limite de transações **T** a ser explorado por carteira encontrada, e a quantidade de níveis a ser explorado **N**.

Durante a execução dos scripts, as transações encontradas são varridas na blockchain de forma recursiva, iniciando pela carteira **E**, buscando até **T** transações posteriores, quando varrendo a partir da carteira de entrada da misturadora, e até **T** transações anteriores, quando varrendo a partir das carteiras finais que receberam as novas moedas.

Para cada teste realizado nas mixers, os scripts foram executados três vezes, alterando o argumento de endereço **E** em cada execução. Uma execução para a carteira de entrada da mixer, outras duas execuções com os endereços que enviaram as moedas para a carteira final do cliente.

O valor utilizado por padrão para o parâmetro **T** foi 10. Desta forma, as 10 últimas transações daquela carteira seriam varridas recursivamente até chegar ao nível **N** que, durante as análises, variou entre 8 e 13. Essa variação ocorreu com a finalidade de encontrar endereços pertencentes aos núcleos dos *pools*. A estratégia adotada para chegar ao núcleo foi aumentar gradativamente o valor de **N** até encontrar endereços de carteira que fossem comuns tanto na varredura de entrada, quanto na varredura inversa, de saída.

Os scripts realizam consultas aos endereços na Blockchain à uma velocidade média de 1,15 endereço por segundo. Durante as varreduras chegou-se a números de carteiras variando entre 1.000 e 18.000 endereços. Essa variação está diretamente ligada a quantidade de transações que cada carteira encontrada fez e quantas carteiras foram utilizadas em cada transação.

Como resultado, os scripts criam um registro em documento de texto com valores separados por vírgula (.csv). Nas varreduras de entrada, seu conteúdo é formado pelo **endereço** da carteira encontrada; o seu **nível** ou graus de distância até a carteira de origem do cliente; um campo chamado **para** que registra para quais outros endereços a carteira enviou moedas, a **quantidade de transações** já realizadas pela carteira, a quantidade de BTCs que a carteira já **recebeu**, a quantidade de BTC que a carteira já **enviou** e seu **saldo** atual. Enquanto que as varreduras reversas registram os dados referentes as carteiras que enviaram moedas até o destino.

Os arquivos gerados foram utilizados para análise do funcionamento dos algoritmos das mixers. Grafos foram criados e são apresentados na Seção 4, com o objetivo de facilitar a visualização dos caminhos percorridos pelas moedas quando enviadas para os serviços de mistura e identificar padrões. Sua renderização foi realizada com o software *Cytoscape*[11], uma plataforma de código aberto para visualização de redes complexas e integração com qualquer tipo de atributo de dados.

4. Resultados e Análise

Como dito em [Böhme et al. 2015] os protocolos de mistura geralmente não são públicos, portanto difíceis de terem sua eficiência avaliada. Neste trabalho foram realizados os experimentos com mixers e a análise da blockchain resultante da mixagem. Algumas informações relevantes foram levantadas acerca da dinâmica operacional e do nível de eficiência dos serviços em pauta.

[11] versão 3.7.0 disponível em https://cytoscape.org/

O endereço de entrada dos serviços, apresenta um padrão: o uso de uma carteira com um endereço novo e que não possua nenhuma transação registrada na blockchain. Tal carteira "inicial" terá duas transações após completadas todas as etapas de mistura; i) uma de recebimento do valor depositado pelo cliente e ii) uma de envio para uma (ou mais) carteira(s) seguinte(s), também de posse e escolha da mixer.

Ao consultar na blockchain a carteira de entrada fornecida pelo serviço Helix Light Grams <13jnQqaJ...>[12], imediatamente encontra-se uma diferença para todos os demais serviços: no momento do envio das moedas para o endereço de entrada, a carteira já possuía 23 transações de recebimento, nenhuma de saída.

Os serviços *Bestmixer* e *Blender.io* realizaram a entregas das novas moedas nos endereços previamente configurados. Quanto aos valores devolvidos, são variáveis e uma parcela bem superior à taxa de serviço cobrada na entrada é retirada até a saída. As mixers alertam sobre as taxas de transação inerentes da tecnologia mas não fornecem valores precisos. Na Tabela 1 encontra-se os valores de cada teste.

Em uma constatação inicial, o risco de não devolução das moedas submetidas para embaralhamento é real e foi concretizado durante os testes. Apesar de seguir todas as recomendações estabelecidas, como "Não enviar mais que 1 transação", "Não enviar menos que 0,0001btc" e "não enviar mais que 43btc" etc, o serviço *Helix Light Grams* não devolveu nenhuma quantia para as carteiras de destino. O mesmo ocorreu com o serviço *BTCBlender*. Não há garantias nem a quem reclamar em caso de perda. Em geral são serviços apócrifos e desviculados de qualquer entidade real, física ou jurídica. Apesar disso, o grande volume e o alto valor das transações de embaralhamento sugerem que tal risco tem sido negligenciado por algumas classes de clientes, como pode ser visto na Tabela 4, a qual relaciona um sumário do volume transacionado nas mixers avaliadas.

Para os serviços Bestmixer e Blender.io, os quais concluíram a entrega prometida das moedas submetidas, foi possível efetuar todo o processo de análise da blockchain descrito na seção anterior. Para ambos os casos é possível identificar comportamentos comuns de seus algoritmos. No que tange a eficiência, a partir dos endereços de entrada das moedas não é possível encontrar a carteira de destino fazendo uso dos mecanismo de rastreamento da blockchain. Atualmente, os serviços mantém as moedas de entrada paradas e só as movimentam após as novas moedas serem entregues às carteiras finais dos clientes, desta forma impedido a relação direta entre as moedas de entrada e as de saída. Para as transações testadas no serviço Bestmixer, o tempo de permanência das moedas na carteira de entrada foi, respectivamente, 24h30m e 2h40m enquanto a Blender.io manteve as moedas por 1 dia 21h e 2 dias 18h.

Outro meio de tentar encontrar a carteira de destino a partir do endereço de entrada foi explicado pelos desenvolvedores do Bestmixer [Bestmixer 2018]. Este método foi utilizado pela ferramenta CAE explorando uma falha no algoritmo das mixers em não aleatorizarem o tempo de entrega das novas moedas às carteiras de seus clientes.

Na prática, a ferramenta verificou que as mixers, ao identificarem as transações para a carteira de entrada, aguardavam o número mínimo de confirmações e utilizavam o horário da última confirmação como parâmetro base para buscas de transações com

[12]Por questões de sigilo, os endereços de carteiras serão exibidos truncados, com apenas 8 caracteres iniciais

Tabela 2. Horários de confirmação e entrega

	Bestmixer 04/10/2018	Bestmixer 08/11/2018	Blender.io 31/11 e 01/12	Blender.io 16/11/2018
Horário 3ª Confirmação	11:59:50	00:06:29	22:39:17	15:37:25
Recebimento Carteira 1	13:49:09	01:58:09	00:40:04	17:38:04
Recebimento Carteira 2	15:57:07	04:03:08	00:40:05	19:38:04

valores equivalentes na blockchain, repetindo este mesmo processo em intervalos exatos de uma hora. Desta forma, as novas buscas poderiam revelar o endereço da carteira de destino.

Foi observado nas análises que os serviços Bestmixer e Blender.io aguardam três confirmações da transação de entrada na blockchain para validarem o envio de moedas por parte do cliente. Analisando os horários das terceiras confirmações das transações e dos horários de criação das transações de pagamento, vindas das mixers testadas, é possível identificar que a Bestmixer utiliza uma variável aleatória de tempo para efetuar os pagamentos de cada uma das carteiras de destino, enquanto que a Blender.io demonstra ter a variável de tempo aleatório apenas para a primeira carteira. Os dados podem ser encontrados na Tabela 2.

4.1. Análise do Processo de Embaralhamento

O trabalho de rastrear o caminho percorrido pelas moedas, da entrada nas mixers até o núcleo das redes misturadas e da chegada nas novas carteiras até o núcleo, foi automatizado como descrito na Seção 3.2. A análise dos dados levantados apontam para valores elevados de Bitcoins sendo repassados através dessas redes, carteiras que participam da rede de mais de uma mixer e uma rede complexa de carteiras transacionando entre si.

Tabela 3. 07 carteiras que mais receberam moedas e possuem relações com Bestmixer (Valores em BTC)

#	Endereço	Total Recebido	Total Enviado	Saldo
1	<1NDyJtNT...>	3.677.916,1613	3.666.614,1191	11.302,0423
2	<1N52wHoV...>	1.962.513,6623	1.961.374,6008	1.139,0615
3	<1J37CY8h...>	1.613.947,2360	1.613.221,8376	725,3984
4	<1DEcTtkr...>	919.343,8349	919.177,5658	166,2690
5	<1NYAd6fA...>	399.610,7258	399.375,4699	235,2560
6	<14cQRmVi...>	319.483,1280	319.313,7818	169,3461
7	<32RQLBAM...>	285.918,4089	285.738,1449	180,264

Ao avançar por níveis dentro das reservas das mixers é possível identificar carteiras que aparecem com frequência em centenas de transações e que já movimentaram valores bilionários em criptomoedas. Nas Tabelas 3 e 4 encontram-se dados dos 07 endereços de carteiras identificadas no núcleo das transações de mistura que mais movimentaram criptomoedas.

Nas transações de teste realizadas na Bestmixer, a carteira encontrada com maior destaque em número e volume de transações foi a <1NDyJtNT...>. Esta carteira recebeu suas primeiras moedas em 08 de Agosto de 2017 e em pouco mais de um ano já

movimentou valores acima de 20 bilhões de dólares, assumindo a cotação da CoinMarketCap[13] em 18 de Novembro de 2018, cujo valor da Bitcoin era de $5.594,97. De acordo com esse ranking mostrado na Tabela 3, as 8 primeiras carteira também movimentaram valores bilionários e as demais atingem cifras milionárias.

Tabela 4. 07 carteiras que mais receberam moedas e possuem relações com o Blender.io (Valores em BTC)

#	Endereco	Total Recebido	Total Enviado	Saldo
1	<1Kr6QSyd...>	5.952.280,2888	5.951.392,8943	887,3945
2	<12cgpFdJ...>	4.572.637,4094	4.569.547,8559	3.089,5534
3	<1NDyJtNT...>	3.682.989,1028	3.673.724,1191	9.264,9837
4	<17A16Qma...>	3.065.897,8365	3.058.860,6970	7.037,1396
5	<14wXrm49...>	2.018.062,1586	2.018.062,1586	0,0167
6	<1MEe2meb...>	1.969.409,9391	1.969.409,9391	0,00
7	<1N52wHoV...>	1.963.032,4378	1.961.830,1066	1.202,3312

Na Tabela 4, encontra-se as carteiras que mais movimentaram moedas identificadas nas transações com a Blender.io e o montante transacionado pelas carteiras é ainda maior. O endereço <1Kr6QSyd...>teve sua primeira transação registrada em agosto de 2016 e em dois anos e um mês de existência já movimentou $33 bilhões de dólares americanos em criptomoedas, ainda considerando a cotação de 18 de Novembro.

Um fato observado durante a análise é a existência de uma mesma carteira encontrada no centro das redes de embaralhamento das duas mixers, o endereço <1NDyJtNT...>, uma das carteiras com alto volume e valor de transações. As possibilidades analisadas são que esse endereço pode pertencer a um investidor/apoiador em comum, a uma grande exchange ou a uma combinação dos dois.

Mais um endereço de carteira que faz relação com transações nas duas misturadoras é o <1BestMix...>. Esta carteira pertence ao pool da Bestmixer, e é usada para receber e efetuar transferências de valores menores que 1BTC, em sua maioria. Uma consulta na blockchain e encontra-se milhares de envios de valores iguais a 0,00000888 saindo desse "hub" para diferentes carteiras.

Considerando as informações apresentadas anteriormente, em conjunto com a informação de que o serviço Bestmixer fornece gratuitamente uma API (*Application Programming Interface*)[14] para interação com o seu mixer, e que o mesmo sugere "Crie seu próprio serviço embaralhador de Bitcoin de graça", é possível que o serviço Blender.io utilize a rede da Bestmixer.

4.2. Visualização Gráfica da Mixagem

Após o levantamento de dados da blockchain explicado na Seção 3.2, as transações mapeadas foram consolidadas e utilizadas para criação de grafos dirigidos acíclicos (DAG), buscando representar os caminhos que as criptomoedas podem percorrer ao entrar na rede de carteiras de uma mixer.

[13]disponível em https://coinmarketcap.com/historical/20181118/
[14]em: https://bestmixer.io/en/api

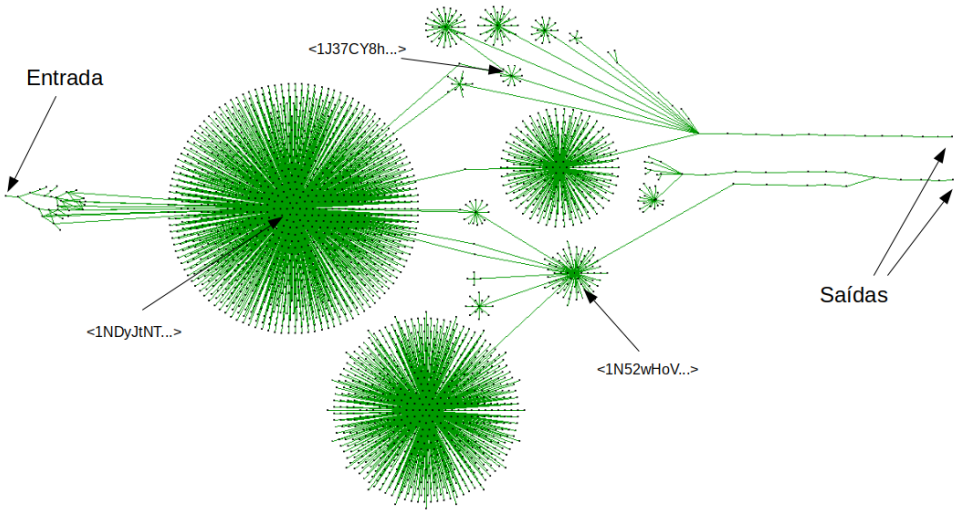


Figura 2. Grafo de Rastreamento da Bestmixer

Nos grafos apresentados nas Figuras 2 e 3, os vértices representam endereços de carteiras, enquanto as arestas representam transações entre elas. Os grafos foram organizados de maneira a representar a entrada na mixer no lado esquerdo e a saída, o recebimento das novas moedas, no lado direito.

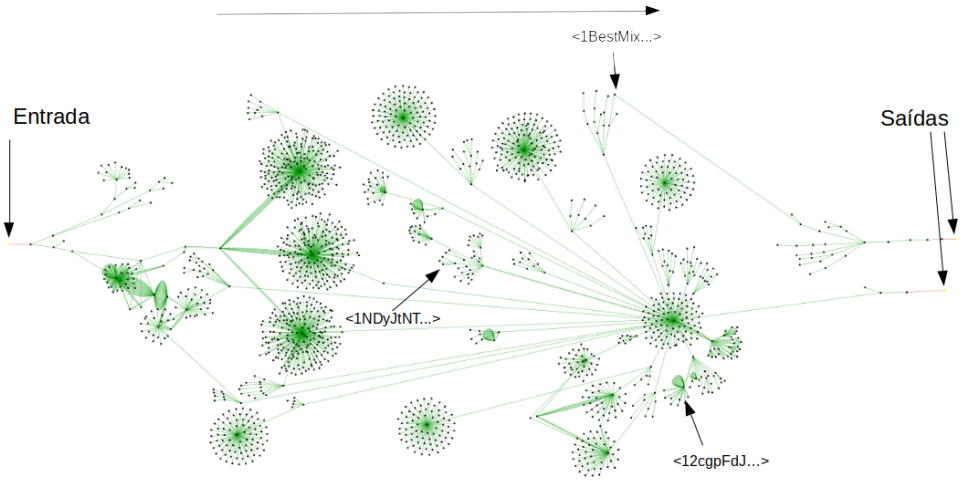


Figura 3. Grafo de Rastreamento da Blender.io

Na Figura 2, está representado o rastreamento realizado em uma das operações de mistura testadas com o serviço Bestmixer. Nela estão identificadas também três carteiras entre as que possuem maiores valores de movimentação de moedas durante o período de testes, conforme Tabela 3. Essas carteiras aparecem com frequência no rastreamento de ambas as transações realizadas com essa mixer. O endereço de carteira <1NDyJtNT...> foi encontrado nas duas operações de testes, 6 e 7 níveis após a entrada das moedas na mixer e, no rastreamento reverso, 12 níveis antes da carteira destino.

O comportamento padrão dos algoritmos durante a entrada é repassar moedas para

carteiras novas que não possuam muitas transações (normalmente 2), dividindo os valores da entrada até entregar a carteiras de destino ou chegar aos nós mais internos da rede de cada mixer, os quais possuem mais transações e mais moedas no saldo. Na saída foi possível identificar uma série de transferências sequenciais entre as carteiras, reduzindo gradativamente o valor das transferências até a chegada nos endereços finais.

A Figura 3, por sua vez, apresenta o grafo do rastreamento de uma operação com a mixer Blender.io. Para a geração desse grafo, foram rastreados e analisados 8 níveis a partir da carteira de entrada e, no rastreamento reverso, 9 níveis a partir da carteira de saída, para chegar a endereços de carteira que se relacionam e se interligam. Dois endereços chamam atenção ao serem localizados nessa representação por estarem presentes nos rastreamento das duas mixers, são as carteiras <1BestMix...> e <1NDyJtNT...>. Ao realizar uma comparação visual entre os grafos, analisando a complexidade de comunicações entre as carteiras apresentadas, a hipótese de adoção da API Bestmixer, por parte da Blender.io torna-se mais concreta.

4.3. Centralidade de Grau das Carteiras

Um padrão identificado nas redes de mistura aponta para carteiras com poucas transações em suas bordas enquanto, no núcleo, encontram-se algumas carteiras com elevado número de transações.

Para auxiliar na análise dos grafos, e identificar quais carteiras são influentes na rede, utilizou-se a ferramenta de medida de *Centralidade*. Como visto em [de Oriani e Paulillo et al. 2017] "existem três medidas de centralidade: i) centralidade de grau; ii) centralidade de proximidade; iii) centralidade de intermediação."

Tabela 5. Carteiras com maior centralidade de grau

Bestmixer					Blender.io				
#	Carteira	Entrada	Saída	Total	#	Carteira	Entrada	Saída	Total
1	<1NDyJtNT...>	24	6007	6031	1	<1BxVHak5...>	5	218	223
2	<1NyfNYAX...>	952	4	956	2	<1L26J8JG...>	219	1	220
3	<1AedYCVy...>	288	1	289	3	<3EN2aCf8...>	207	1	208
4	<14RcD35U...>	265	1	266	4	<38QyvVk8...>	205	1	206
5	<1KzemnJK...>	154	1	155	5	<3LZLpGi5...>	200	1	201
6	<1QCXuYkZ...>	153	1	154	6	<12ceuQRX...>	3	167	170

Na centralidade de grau, um ator (uma carteira Bitcoin, para este estudo) centraliza a relação com outros atores da rede. Esta medida consiste no número de ligações ou laços com os outros atores, simbolizando o quão influente é um ator em uma rede. De acordo com [Mascarenhas et al. 2018] O número de arestas incidentes a um nó *i* é denominado grau k_i. Em redes direcionadas, como as estudadas neste trabalho, considera-se para um nó *i* o grau de entrada k_i^{in} , de saída k_i^{out} ou total k_i.

Foram identificadas então as seis carteiras com maior centralidade de grau total de cada uma das redes, seus valores encontram-se na tabela 5.

Na rede Bestmix, a carteira com maior centralidade de grau total foi a <1NDyJtNT...> com 6031 interações. Esta carteira possui um elevado grau de saída quando comparado ao grau de entrada, sendo a carteira que registrou o maior número de

transferências de moedas durante a realização dos experimentos. No entanto, outros cinco endereços também apresentaram elevada centralidade de grau, porém com foco inverso à primeira. O grau de entrada dessas carteiras é muito superior ao de saída, o que indica que esses foram os endereços que mais receberam transferências durante os testes.

A rede Blender.io revelou um comportamento similar à rede Bestmix, quanto às carteiras mais influentes. Duas das carteiras com maior grau total, <1BxVHak5...> e <12ceuQRX...>, possuem elevado grau de saída e baixo grau de entrada, sendo esses endereços os que mais enviaram transferências para dentro da rede. As demais carteiras mais influentes, realizaram o papel inverso, com alto grau de entrada.

Ao analisar, na blockchain, os endereços de carteiras encontrados com alto grau de entrada, elucida-se que os mesmos funcionam como acumuladores, recebendo centenas de transferências de pequenos valores para formação de montantes elevados que serão redistribuídos em momento futuro.

5. Trabalhos Relacionados

A Bitcoin e a Blockchain são tecnologias que estão sendo pesquisadas, em diversas áreas, por toda a última década, a segurança é uma dessas áreas sendo exploradas. Em [Herrera-Joancomartí 2014] o autor discorre sobre os desafios da segurança da Bitcoin incluindo questões de privacidade e anonimato. É apresentada uma síntese sobre Análise de Blockchain e Análise de tráfego para identificação de transações. Como alternativa para a melhoria do anonimato na Bitcoin, o autor sugere a utilização de mixers disponíveis à época.

Em 2013, estudos acerca da utilização de misturadores de moedas surgem buscando entender seu funcionamento e suas possíveis falhas. [Moser et al. 2013] realizaram experimentos com duas das misturadoras de primeira geração: Bitcoin Fog e Bitlaundry, encontrando falhas na capacidade de anonimização da BitLaundry naquele ano através de rastreamento das transações na blockchain. Já em 2018 os próprios criadores da Bestmixer, mixer estudada neste trabalho, publicaram no fórum em [Bestmixer 2018], falhas das mixers existentes no período em que lançaram seu serviço.

Mais recente, em [van Wegberg et al. 2018], foi realizada uma pesquisa focada na lavagem de dinheiro utilizando Bitcoin. Os autores fizeram o levantamento financeiro e de usabilidade de cinco mixers e cinco exchanges, examinando dados de custos das transações desde as carteiras Bitcoin até os saques em dólares. Chegando a conclusão de que "é um conceito praticamente concebível e tem um alto grau de semelhança para ser integrado em esquemas atuais e futuros de lavagem de dinheiro".

Em [Khalilov and Levi 2018] é apresentado um survey sobre anonimato e privacidade em Bitcoins e outras criptomoedas. Os autores separam os trabalhos levantados durante o período de 03/01/09 até 01/01/17, em dois grandes grupos: por Resultados e por Métodos. Dentro da categoria Resultados, encontram-se os trabalhos que buscam fazer análises e ligações na blockchain a partir de endereços. Este é o grupo onde os autores encontraram mais trabalhos relacionados. Foram 27 pesquisas encontradas e outras 11 citadas. Dentre estes, outros trabalhos de estudo de mixers identificaram falhas nos serviços BitLaunder e Coinsplitter.

A respeito das análises de blockchain e grafos, em [Mascarenhas et al. 2018] há

um estudo sobre as transações na rede Ethereum. Nele é realizado uma análise da blockchain, onde são observadas as transações de carteiras pré-selecionadas realizadas entre os meses de maio e dezembro de 2017. O mapeamento dessas movimentações unidas aos cálculos dos graus dos nós avaliados, resultou em dados preliminares a respeito de padrões de comportamento na rede, visualmente representados em grafos

No sentido oposto ao de trazer anonimato para carteiras de criptomoedas, o trabalho de [Soares and Costa 2018] apresenta uma proposta de identificação de carteiras. Um serviço chamado *Address Name System* ou ANS. Nesta proposta apresenta-se dois módulos, um para registro de carteiras a ser utilizado "por parte do detentor de um endereço de carteira e de um certificado digital para registrar o mapeamento de endereço para entidade", e um módulo de consulta, correspondendo a um serviço que recebe parâmetros de entrada como um identificador de uma instância de DLT[15] e um endereço de carteira, e retorna ao usuário a identidade declarada, se houver uma.

6. Conclusões

Diante do novo paradigma financeiro pós surgimento da Bitcoin, que propôs uma nova forma de transferir valores entre usuários através da internet, utilizando criptografia e um livro caixa público, este trabalho realizou um estudo acerca do funcionamento de serviços de mistura ou embaralhamento de criptomoedas (*criptocurrency mixers* ou *tumblers*), complementaando e expandindo a pesquisa de [Moser et al. 2013], realizando a análise da nova geração de misturadores e seus algoritmos emergentes após o surgimento da Bestmix.io. Serviços estes que atraíram a atenção de muitos usuários que procuram anonimato e privacidade de suas carteiras com ativos digitais.

Foram executados testes com quatro serviços: Bestmixer, Blender.io, BTCBlender e Helix Light Grams, bem como o desenvolvimento de uma ferramenta própria de rastreamento das transações e carteiras envolvidas no embaralhamento das moedas digitais. Os resultados foram discutidos na Seção 4 e confirmam o risco de perca dos ativos envolvidos, pois duas (BTCBlender e Helix Light Grams) das quatro mixers testadas não devolveram nenhuma das moedas prometidas.

Ao analisar as duas mixers que concluíram a entrega das moedas, pode-se concluir que a mistura pode custar mais de 20% do valor de entrada enviado pelo cliente; nem todas as mixers entregam a segurança que prometem, a Blender.io pode conter uma falha conhecida e não garantir anonimidade de seus clientes; nem todas as mixers conseguem garantir o valor exato que será cobrado em todo o processo; a misturadora Bestmixer disponibiliza API para utilização de seus serviços e novas misturadoras estão utilizando sua infra-estrutura, como a Blender.io.

Isso posto, ao considerar os riscos de não recebimento das moedas, da possibilidade de falhas em garantir o anonimato, e de encargos totais que giram em torno de 20% do valor depositado, surgem alguns questionamentos como: A quem pode interessar um serviço com altos riscos e tarifas tão elevadas para conseguir anonimato? Apesar dos questionamentos, as mixers estão diariamente interagindo com diversos clientes, o que pode ser evidenciado pelas carteiras que movimentam os valores bilionários identificados.

[15] Sigla em inglês para Tecnologia Distribuída de Livro Razão, grupo de banco de dados o qual a Blockchain pertence.

A ética de tais serviços é claramente discutível pois, embora a primeira vista pareça resgatar o direito legítimo ao anonimato e sigilo financeiro, também favorece a execução de crimes. Como discutido em [van Wegberg et al. 2018] e [Böhme et al. 2015]“ a lavagem de dinheiro por Bitcoin pode evoluir para se tornar mais difícil de rastrear, particularmente quando os fundos são encaminhados através de misturadores”.

Como trabalho futuro destaca-se a necessidade de desenvolver uma nova ferramenta que possa identificar, a partir do endereço de entrada, em qual carteira serão depositadas as moedas de saída. Uma falha referente a falta de aleatoriedade no horário de entregas da moeda já havia sido explorada anteriormente por uma aplicação chamada *CAE-check* porém não é mais possível encontrar a aplicação, nem códigos-fonte, nem resultados de testes anteriores, e dados levantados levam a crer que ainda existam mixers vulneráveis.

Referências

Bestmixer (2018). Bestmixer.io the future of bitcoin mixing! technology is here. `https://bitcointalk.org/index.php?topic=3140140.msg33958885msg33958885`. [Acesso em: 10-10-2018].

Böhme, R., Christin, N., Edelman, B., and Moore, T. (2015). Bitcoin: Economics, technology, and governance. *Journal of Economic Perspectives*, 29(2):213–38.

de Oriani e Paulillo, L., Garcia, L., and Neto, M. (2017). *Governanças de Redes: Economia, Política e Sociedade*. Elsevier Editora Ltda.

Herrera-Joancomartí, J. (2014). Research and challenges on bitcoin anonymity. In *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance*, pages 3–16. Springer.

Khalilov, M. C. K. and Levi, A. (2018). A survey on anonymity and privacy in bitcoin-like digital cash systems. *IEEE Communications Surveys & Tutorials*.

Mascarenhas, J. Z., Vieira, A. B., and Ziviani, A. (2018). Análise da rede de transações do ethereum. In *Anais do I Workshop em Blockchain: Teoria, Tecnologias e Aplicações (WBlockchain-SBRC 2018)*, volume 1. SBC.

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

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Soares, M. and Costa, R. (2018). Auto identificação voluntária e verificável de participantes em aplicações baseadas em livros-razão distribuídos. In *Anais Estendidos do XVIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 99–112, Porto Alegre, RS, Brasil. SBC.

van Wegberg, R., Oerlemans, J.-J., and van Deventer, O. (2018). Bitcoin money laundering: mixed results? an explorative study on money laundering of cybercrime proceeds using bitcoin. *Journal of Financial Crime*, 25(2):419–435.

Demochain - Framework destinado a criação de redes blockchain híbridas para dispositivos IoT

Lorenzo W. Freitas[1], Carlos Oberdan Rolim[2]

[1]Departamento de Engenharias e Ciência da Computação
Universidade Regional Integrada do Alto Uruguai e das Missões – Santo Ângelo, RS – Brazil

lorenzofreitas@aluno.santoangelo.uri.br, ober@san.uri.br

Resumo. *O uso da tecnologia Blockchain no contexto da IoT (Internet of Things) está sendo cada vez mais explorado pela comunidade acadêmica e a indústria. No entanto, essa implantação pode ser custosa ou inviável pois a Blockchain pode exigir recursos computacionais que não são obtidos facilmente com o uso de dispositivos IoT. Assim, esse trabalho apresenta o framework Demochain cuja função é auxiliar no desenvolvimento de plataformas blockchains híbridas no contexto de IoT.*

1. Introdução

Uma importante área de pesquisa na Computação que está apresentando um rápido crescimento é a Internet das Coisas (IoT). Segundo Gartner (2017), o gasto total em dispositivos e serviços de IoT atingiu quase US $ 2 trilhões em 2017, e haverá mais de 20 bilhões de "coisas" conectadas em todo o mundo até 2020.

Entretanto, apesar de possuir a natureza distribuída, a maioria das soluções IoT ainda dependem de arquiteturas que seguem o modelo cliente servidor. Embora esse modelo arquitetural possa funcionar hoje, o crescimento da IoT sugere que novas arquiteturas deverão ser propostas no futuro [Caramés e Lamas 2017]. Uma alternativa que vêm sendo explorada pela comunidade acadêmica e a indústria é o emprego de blockchains devido a sua, capacidade de manter os registros imutáveis sem perder segurança, com algoritmos que tratam nodos maliciosos.

Com esse cenário e a falta de padronização dos dispositivos IoT devido a seus diferentes objetivos existe uma dificuldade na implementação e modelagem de uma rede blockchain personalizada para dispositivos IoT [Conoscenti et al, 2016]. Assim, o presente trabalho apresenta o Demochain, um framework voltado para a criação de redes blockchain híbridas para dispositivos IoT. Um dos diferenciais do Demochain é a sua capacidade de oferecer opções para mesclar e combinar diferentes níveis na arquitetura e também funcionalidades da blockchain pura (totalmente descentralizada), variando seus protocolos e criptografias, construindo assim, uma blockchain híbrida. Além disso, pode-se ressaltar que a maior contribuição do Demochain é a possibilidade de facilitar a modelagem e a prototipação de novas redes blockchain em ambientes variados.

2. Framework Proposto

O framework construído foi chamado de Demochain e foi pensado para diminuir a complexidade na conexão dos dispositivos. Portanto, vários conceitos utilizados em

blockchains tradicionais foram simplificados. Ele foi desenvolvido com a linguagem Go [Google 2009] e foi utilizada uma abordagem de orientação a objetos com o padrão Decorator em sua implementação [Schmager, Frank 2010]. A arquitetura que foi implementada é multicamadas, baseada no trabalho de Li e Zhang (2017). Nessa arquitetura, cada nodo na rede pode assumir dois papéis: ser um nodo centralizador (Edge Node), que serve para controlar o acesso dos nodos abaixo na camada, ou ser um nodo de alto nível (High-Level Node), também chamado de nodo filho, onde são adicionado blocos e executado o protocolo de consenso.

Com isso tem-se um modelo descentralizado com alguns níveis de centralização, porém não totalmente distribuído. A Figura 1 apresenta a arquitetura do Demochain.

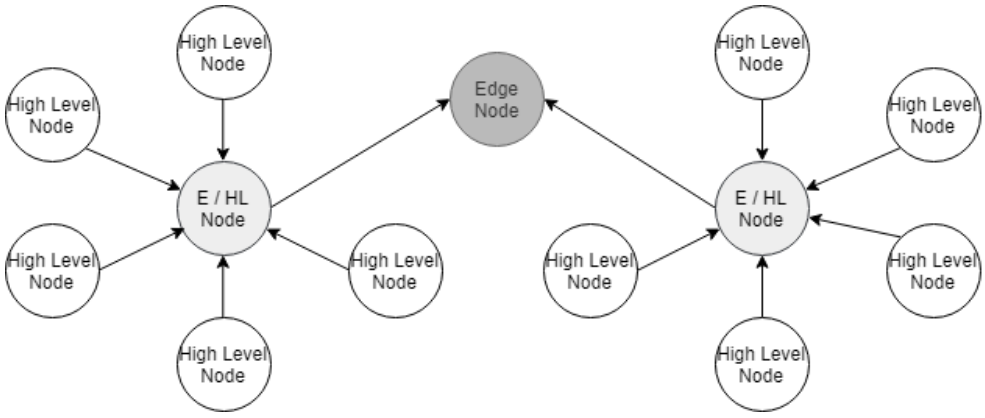


Figura 1. Arquitetura implementada.

Ao escolher o esquema de criptografia para o framework, foi levado em consideração não apenas a segurança fornecida de acordo com a carga computacional, mas também o consumo de energia, fazendo um trade-off entre segurança e consumo. Foram implementados quatro algoritmos de criptografia assimétrica: RSA, Ed25519, Secp256k1 e ECDSA. A identificação dos nodos ocorre utilizando a pilha TCP em sua forma tradicional. Portanto, um mesmo dispositivo pode estar rodando várias instâncias do Demochain, desde que em portas distintas.

No Demochain a blockchain é replicada para todos os nós. Entretanto o diferencial deste framework é que a execução do consenso é por nível. Protocolo consenso consiste em um mecanismo que determina as condições a serem alcançadas para concluir que um acordo foi alcançado em relação às validações dos blocos para ser adicionado ao blockchain [Zheng et al. 2017]. Foram implementados três protocolos de consenso: PoW (Proof of Work), PoS (Proof of Stake) e PBFT (Practical Byzantine Fault Tolerance).

2.1. Proof of Work (PoW)

Algoritmos de consenso PoW, ou Prova de Trabalho baseiam-se no fato de que um nó malicioso tem que executar muito trabalho do ponto de vista computacional para atacar a rede, e por isso, é menos provável que ele vá querer atacar. O trabalho realizado geralmente envolve fazer alguns cálculos até que uma solução seja encontrada, um

processo que é comumente conhecido como mineração. No caso da blockchain do Bitcoin, a mineração consiste em encontrar um número aleatório, chamado de número nonce que fará o hash SHA256 do cabeçalho do bloco para ter no início certo número de zeros. Portanto, os mineradores têm que demonstrar que eles realizaram certa quantidade de trabalho para resolver o problema. Uma vez resolvido o problema, é realmente fácil para outros nodos verificar se a resposta obtida é válida. No entanto, este processo de mineração faz a blockchain ineficiente em taxa de transferência, escalabilidade [M. Vukolić 2015], e também em termos de consumo de energia, o que não é desejável em uma IoT rede.

2.2. Proof of Stake (PoS)

PoS ou Prova de Participação é um mecanismo de consenso que requer menos recursos computacionais e potência do que PoW [BitFury Group 2015], por isso consome menos energia. Em uma blockchain baseado em PoS, assume-se que as entidades com mais participação na rede são os menos interessados em atacá-lo. Assim, os nodos precisam provar periodicamente que eles possuem certa quantidade de participação na rede (por exemplo, moeda ou quantidade de dados coletados a partir de sensores). Podemos comparar o PoS há uma loteria, onde é realizado um sorteio e escolhido um nodo que irá dizer se o bloco que está sendo enviado pela rede é válido ou não, por consequência, o nodo que tiver mais participação recebe mais "fichas" neste sorteio, e tem mais chances de validar o bloco. O nodo que realiza o sorteio pode ou não ser pré-definido.

2.3. Practical Byzantine Fault Tolerance (PBFT)

PBFT [M. Castro e B. Liskov 1999] ou Tolerância de Falhas Bizantinas Prática é um algoritmo de consenso que resolve o problema dos generais bizantinos, recomendado para ambientes assíncronos. PBFT assume que menos de um terço dos nós são maliciosos. Para cada bloco a ser adicionado à cadeia, um líder é selecionado para ser encarregado de encomendar a transação. Essa seleção deve ser apoiada por pelo menos 2/3 de todos os nós, que devem ser conhecidos pela rede.

2.4. Implementação do consenso

No Demochain o PoW funciona da maneira clássica, é imposta uma dificuldade e é realizada uma tentativa aleatória de encontrar uma Hash utilizando SHA256 que comece com o número de zeros da dificuldade. Na figura 2 temos um exemplo de Hash's geradas de acordo com a dificuldade do PoW, onde é possível verificar o número de zeros no início da Hash.

Dificuldade 1
012b46ed9cf86cab3fd08708ec3ee868528044e87931b2d00dd21c4fbae5d919

Dificuldade 2
001ca89c82993d99e3477a11c08687caa5c8fb6760cd93ef25de45dc2746e8b4

Dificuldade 3
000ca89c82993d99e3477a11c08687caa5c8fb6760cd93ef25de45dc2746e8b4

Figura 2. Dificuldade do Proof of Work.

Como o PoW processa os blocos apenas com os próprios recursos, sem depender da aprovação dos demais nodos, no Demochain utilizando o PoW é possível realizar a validação e geração de novos blocos independente da conexão com o seu Edge Node, portanto mesmo um dispositivo estando desconectado da rede, ao se reconectar ele consegue sincronizar os blocos validados, isso se ele estiver executando a mesma instância, pois esses blocos ficam armazenados em memória, entretanto caso sua execução seja interrompida e ele não estiver conectado na rede os blocos validados são perdidos, já que foi implementado apenas um arquivo físico de Blockchain que mantém o conteúdo atualizado conforme o conteúdo atual que está circulando na rede. Uma implementação com dois arquivos de Blockchain, um atualizado conforme a rede e outro com o conteúdo "local" é uma possível melhoria, conforme é sugerido em [Liang et al. 2017].

A razão dessa funcionalidade é dar liberdade ao desenvolvedor dizer quando deve ser o envio dos blocos, não precisando ser necessariamente logo após a validação dos mesmos, o que em um cenário de concorrência de criptomoedas não faz sentido, mas em redes privadas pode fazer, dependendo dos objetivos de cada um.

A implementação proposta para o PoS é baseada no trabalho de Dan Larimer (2014), também chamado de DpoS (Delegate Proof of Stake). A diferença do DPoS para o PoS "clássico" é que são delegados representantes para a realização do sorteio para a governança da rede com critérios predeterminados. Nesse caso a cada bloco gerado será solicitado um sorteio ao Edge Node, que irá direcionar o bloco ao vencedor para realizar a validação. Esse sorteio acontece no escopo de apenas um nível de rede, porém após ser validado o bloco é passado aos demais níveis. A figura 3 demonstra o funcionamento do PoS.

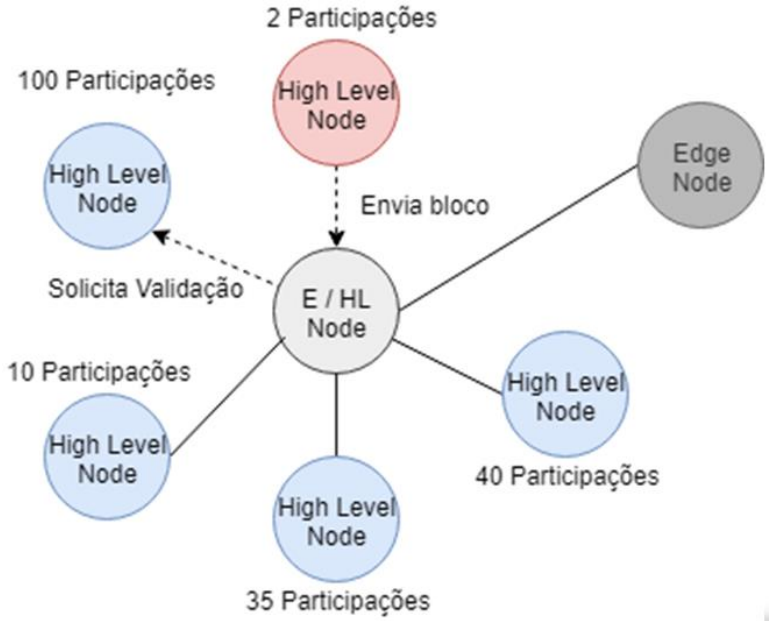


Figura 3. Funcionamento do PoS.

A implementação proposta para o PBFT também tem alterações. Assim como no PoS, toda validação deverá ser solicitado ao Edge Node da rede, porém ele irá enviar o bloco para todos os nodos do mesmo nível, ou seja, todos os High Level Nodes. Após o

mesmo irá contabilizar quem aprovou e quem desaprovou o bloco. Se pelo menos 2/3 de seus High Level Nodes aprovarem o bloco, então é decretado que o bloco é válido. Assim como o PoS o protocolo executa em um nível da rede, porém ao ser validado o bloco é passado aos demais níveis. A figura 4 demonstra o funcionamento do PBFT.

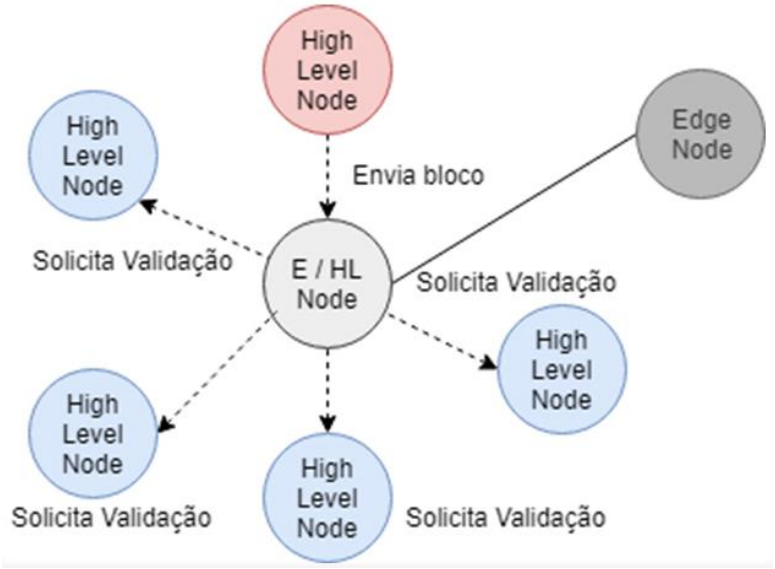


Figura 4. Funcionamento do PBFT.

Uma observação importante é que como os protocolos PoS e PBFT irão executar em apenas um nível da rede, uma preocupação que o desenvolvedor deve ter é a divisão dos Edge Nodes, pois quanto menos High Level Nodes vinculados a ele maior a chance de um nodo intruso realizar uma ação maliciosa, pois são protocolos probabilísticos.

Será suportado apenas um protocolo de consenso rodando em todos níveis de rede. Uma possível melhoria seria permitir cada High-Level Layer estar rodando um tipo de protocolo de consenso, ou mesmo não ter nenhum implementado, podendo apenas estar inserindo os blocos e os compartilhando sem protocolo de consenso, entretanto isso exigiria um esquema mais complexo de armazenamento e uma compressão dos dados validados de cada nível da rede, fazendo com que cada High-Level Layer tivesse uma blockchain diferente da outra em seu formato.

2.5. Mensagens

A troca de mensagens se dá de maneira simples, podendo ter um ou dois fluxos em cada conexão, um de recebimento e outro para envio. Caso o desenvolvedor defina uma rede não permissionada sempre será aberto os fluxos de recebimento e de envio, porém é possível definir também uma rede permissionada. Isso permite com que o desenvolvedor defina nodos apenas de backup das informações para eventuais falhas. Na figura 5 temos um exemplo ilustrativo de uma rede permissionada com o Demochain.

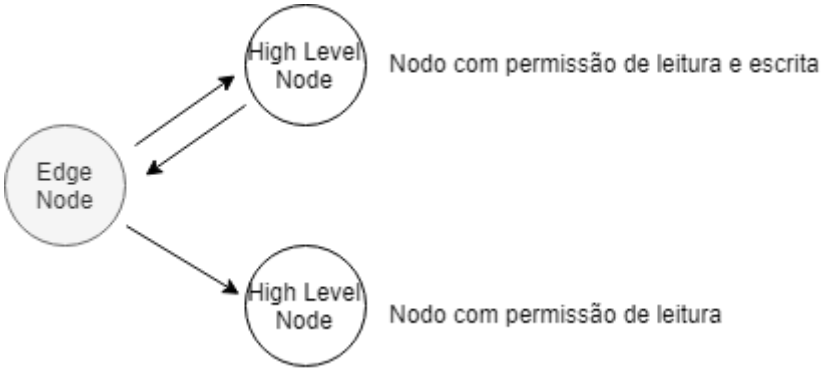


Figura 5. Rede permissionada com nodo de leitura.

Em uma primeira versão o demochain envia a Blockchain por inteira a cada mensagem trocada, o que a um longo prazo pode gerar gargalos na rede. Um controle mais complexo de troca de mensagens não foi implementado.

2.6. Estrutura dos blocos

A estrutura de blocos utilizadas pelo Demochain possui semelhanças e diferenças com as estruturas tradicionais utilizadas em criptomoedas. É importante ressaltar que normalmente em criptomoedas um bloco é formado de transações, sendo que toda transação tem um "recipient", podendo ou não ter um "sender". No Bitcoin por exemplo, em uma transação comum temos um "sender" que envia valor para o "recipient", porém quando um novo bloco é minerado é chamado de "Coinbase", pois não tem um "sender". Como esse framework não é voltado para criptomoedas foi utilizado um conceito mais simples somente com blocos, pois a princípio a validação que será aplicada nos dados coletados não deverá ser passada para outros Nodos. Abaixo temos a estrutura de bloco que o Demochain implementa:

- Index: Índice sequencial único que é incrementado a cada nova adição de blocos na blockchain.
- Timestamp: Carimbo de data e hora do momento da adição do bloco.
- Data: Dados que foram validados.
- Hash: Hash gerada para o bloco.
- PrevHash: Hash do bloco anterior.
- Consensus: Consenso utilizado para a validação do bloco.
- Nonce: Número Nonce é utilizado apenas para o consenso Proof of Work.
- Target: Target do nodo onde foi validado o bloco. Podemos dizer que corresponde ao "dono" do bloco.

2.7. Armazenamento

Para o armazenamento será gravado em um arquivo que servirá para manter os blocos já validados por todos os nodos. O arquivo será gravado com extensão ".bc" com um Json não identado com todos os dados da blockchain. Na figura 6 temos um exemplo de um

bloco gerado com a estrutura de bloco do Demochain, e na figura 7 temos o mesmo bloco armazenado no arquivo “.bc”.


```
[
   {
      "Index":0,
      "Timestamp":"2018-11-17 19:02:25.834476 -0200 -02 m=+3.635126501",
      "Data":"0",
      "Hash":"f1534392279bddbf9d43dde8701cb5be14b82f76ec6607bf8d6ad557f60f304e",
      "PrevHash":"",
      "Consensus":{
         "TypeConsensus":1,
         "Difficulty":3
      },
      "Nonce":"",
      "Target":"QmbiS1uqyVVXzV2pYdife8vJeZZRtW4TQiNEP4Dj1ckNiG"
   }
]
```


Figura 6. Bloco json identado.


```
[{"Index":0,"Timestamp":"2018-11-17 19:02:25.834476 -0200 -02
m=+3.635126501","Data":"0","Hash":"f1534392279bddbf9d43dde8701cb5be14b82f76ec6607bf8d6
ad557f60f304e","PrevHash":"","Consensus":{"TypeConsensus":1,"Difficulty":3},"Nonce":""
,"Target":"QmbiS1uqyVVXzV2pYdife8vJeZZRtW4TQiNEP4Dj1ckNiG"}]
```


Figura 7. Bloco json não identado.

A seguir serão apresentados os resultados de simulações de uso do framework com o objetivo de demonstrar a performance de cada protocolo de consenso sobre um mesmo desenho de arquitetura.

3. Resultados

Todos os códigos bem como testes realizados estão disponíveis em Freitas (2018). Os resultados foram obtidos com o uso de equipamentos para simular um ambiente IoT. A Tabela 1 demonstra a configuração do hardware utilizado para os testes do Demochain.

Tabela 1. Tabela de hardware para experimentação.

Código	Descrição	Processador	Memória RAM	Sistema Operacional
H01	Notebook	Intel Core i5 2x 2.20 GHz	4 Gigabytes	Windows 10 Pro
H02	Raspberry Pi	BCM2835	512 Megabytes	Raspbian Stretch Lite

Para testar o desempenho do framework foi utilizado as métricas descritas por S. Angelis (2017), que são comumente usadas para medir aplicações descentralizadas. A Tabela 2 descreve as métricas utilizadas.

Tabela 2. Tabela de métricas para experimentação.

Código	Nome	Descrição
M01	Taxa de validação (Throughput)	Mede o número de blocos gerados com sucesso por segundo.
M02	Latência (Latency)	Mede o tempo entre envio e confirmação do bloco relativo em milisegundos.
M03	Escalabilidade (Scalability)	Média em milissegundos das mudanças de latência, conforme é aumentado o número de nodos na rede.
M04	Tempo Total (Total Time)	Mede o tempo total em minutos de execução do programa de teste.

A fim de simular dados reais de ambientes IoT foi utilizado um dataset público [Ortiz, J. e Gottschlich, N. 2016]. A Tabela 3 descreve o dataset utilizado.

Tabela 3. Tabela de datasets para experimentação.

Código	Descrição	Referência
D01	Dados de consumo de eletricidade doméstica com 60.640 medições coletadas entre janeiro de 2007 e junho de 2007 (6 meses).	[Ortiz, J. e Gottschlich, N. 2016]

Para testar o framework foi definido cenários de experimentação, no qual para cada cenário foi criado um aplicativo de testes que importa o Demochain. Os cenários possuem dois objetivos principais, demonstrar exemplos de como utilizar o framework e analisar o seu desempenho. Nas subseções seguintes está descrito cada cenário, seu objetivo e seus resultados.

3.1. Cenário 1

O Objetivo deste cenário é comparar desempenho de geração de blocos, utilizando as métrica M01 e M04. Para isso foi implementado um nodo único para testar o protocolo de consenso Proof of Work com diferentes níveis de dificuldade utilizado o dataset D01 como entrada de dados. A cada execução do Proof of Work foi incrementada a sua dificuldade, variando de 1 a 5. A Tabela 4 demonstra os resultados obtidos deste cenário, e as figuras 8 e 9 demonstram os resultados retirados da tabela 4.

Tabela 4. Resultados da validação do PoW aplicando as métricas M01 e M04.

	Taxa de validação (blocos p/ segundo)	**Tempo Total (minutos)**
PoW 1	10 Blocos	434 min
PoW 2	8 Blocos	543 min
PoW 3	5 Blocos	869 min
PoW 4	3 Blocos	1.448 min
PoW 5	1 Bloco	4.344 min

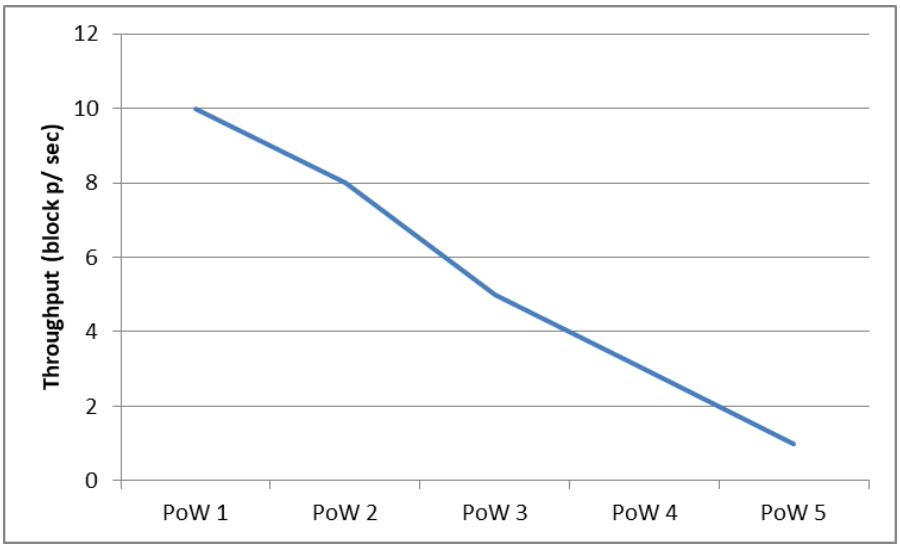


Figura 8. Aplicação da métrica M01 sobre o PoW.

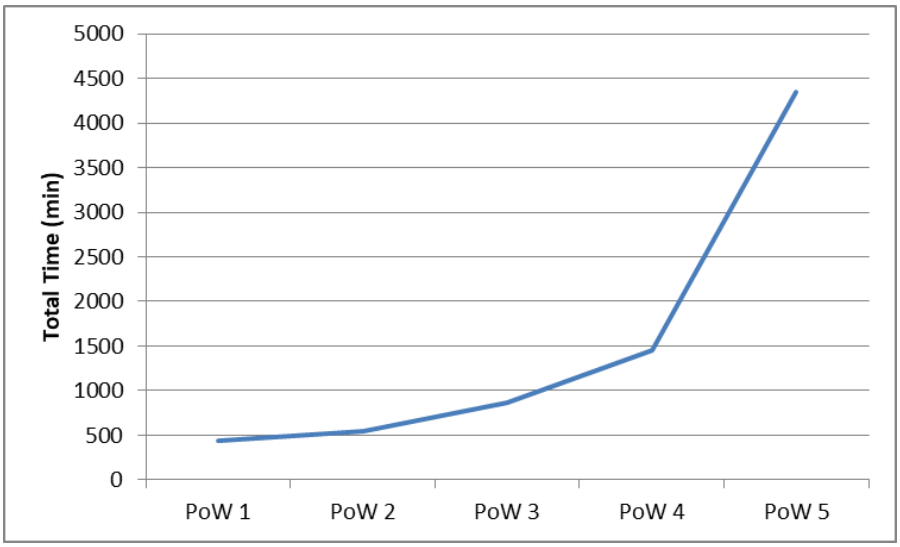


Figura 9. Aplicação da métrica M04 sobre o PoW.

Os resultados obtidos neste cenário demonstram que o Proof of Work tem um desempenho razoável na validação dos blocos até a dificuldade 3, portanto é uma boa alternativa para os nodos que não ficam conectados a todo momento, pois o seu processamento não depende da rede.

3.4. Cenário 2

O objetivo deste cenário é comparar desempenho de cada protocolo de consenso, utilizando as métricas M01, M02, M03 e M04. Para isso foi implementado um rede com até 15 nodos dispostos nos hardwares H01 e H02, utilizando a criptografia RSA, com o dataset D01 como entrada de dados, sendo que foi utilizado o Proof of Work apenas com dificuldade 3. Todos os nodos estavam rodando sobre um mesmo nível de rede, ou seja, todos respondiam para o mesmo Edge Node.

A Tabela 5 apresenta a aplicação da métrica M01 sobre os três protocolos de consenso, considerando a variação do número de nodos na rede, e a figura 10 demonstra com um gráfico de linhas os resultados desta tabela.

Tabela 5. Taxa de validação (Throughput) em blocos p/ Segundo.

	PoW 3	**PoS**	**PBFT**
1 Nodo	5 blocos	25 blocos	20 blocos
2 Nodos	4,7 blocos	21 blocos	15,7 blocos
5 Nodos	4,2 blocos	15 blocos	11 blocos
10 Nodos	3,5 blocos	10,5 blocos	6,5 blocos
15 Nodos	2,5 blocos	6,5 blocos	2,6 blocos

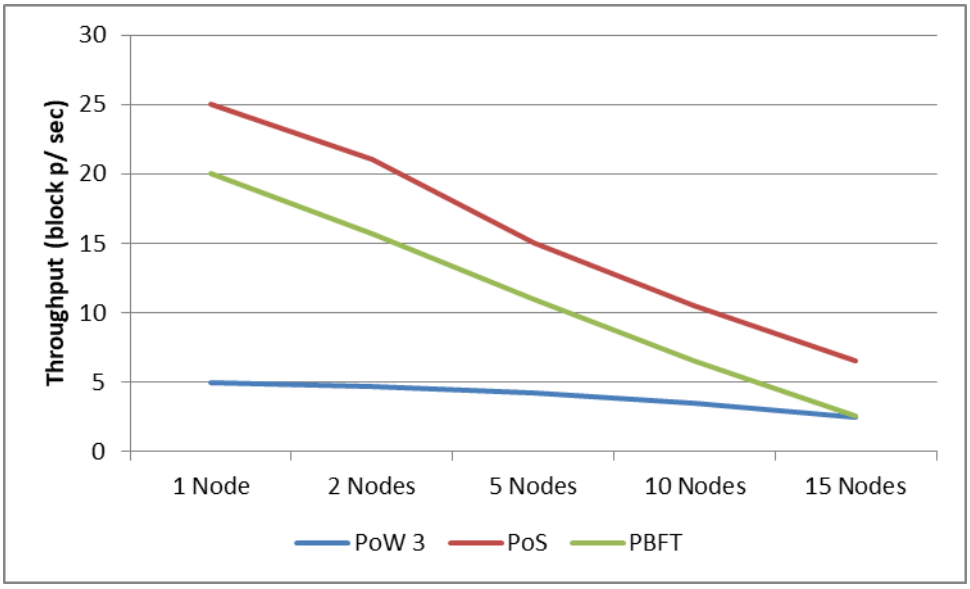


Figura 10. Aplicação da métrica M01.

A Tabela 6 apresenta a aplicação da métrica M02 sobre os três protocolos de consenso, considerando a variação do número de nodos na rede, e a figura 11 demonstra com um gráfico de linhas os resultados desta tabela.

Tabela 6. Latência (Latency) em milisegundos.

	PoW 3	**PoS**	**PBFT**
1 Nodo	200 ms	40 ms	50 ms
2 Nodos	212 ms	47 ms	64 ms
5 Nodos	238 ms	66 ms	90 ms
10 Nodos	285 ms	96 ms	154 ms
15 Nodos	400 ms	153 ms	384 ms

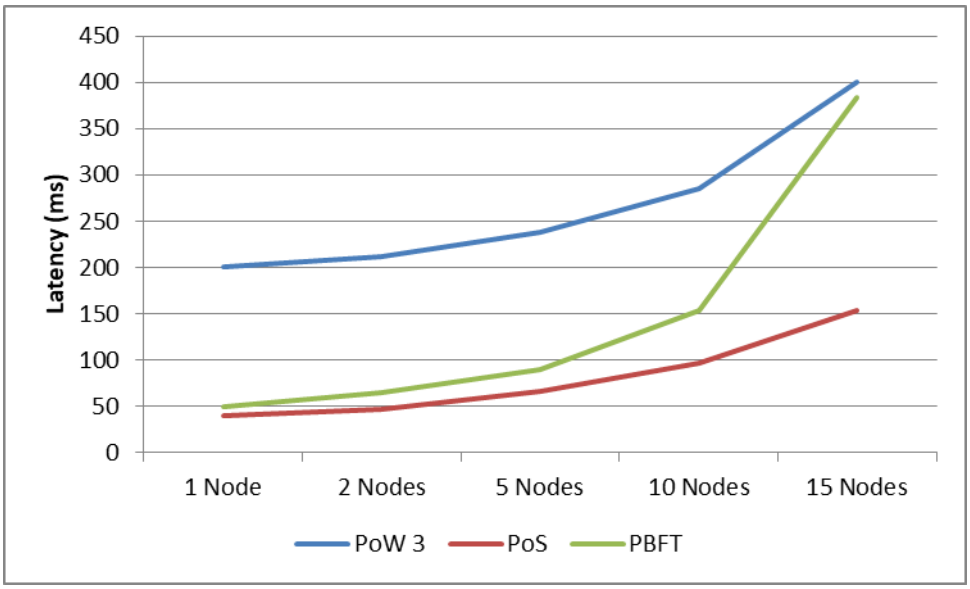


Figura 11. Aplicação da métrica M02.

A Tabela 7 apresenta a aplicação da métrica M03 sobre os três protocolos de consenso, considerando a variação do número de nodos na rede, e a figura 12 demonstra com um gráfico de barras os resultados desta tabela.

Tabela 7. Escalabilidade (milisegundos).

	PoW 3	**PoS**	**PBFT**
Média de Latência p/ Nodo	13,33 ms	7,53 ms	22,26 ms

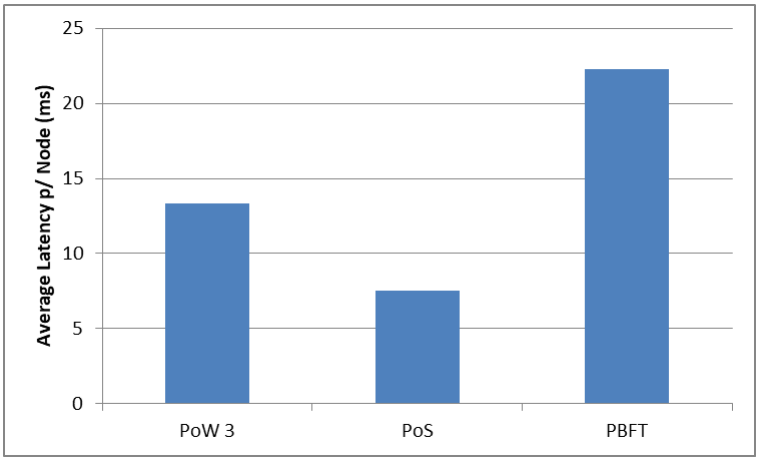


Figura 12. Aplicação da métrica M03.

A Tabela 8 apresenta a aplicação da métrica M04 sobre os três protocolos de consenso, considerando a variação do número de nodos na rede, e a figura 13 demonstra com um gráfico de linhas os resultados desta tabela.

Tabela 8. Tempo total da execução.

	PoW 3	**PoS**	**PBFT**
1 Nodo	869 min	174 min	217 min
2 Nodos	462 min	103 min	138 min
5 Nodos	207 min	58 min	79 min
10 Nodos	124 min	41 min	67 min
15 Nodos	116 min	44 min	111 min

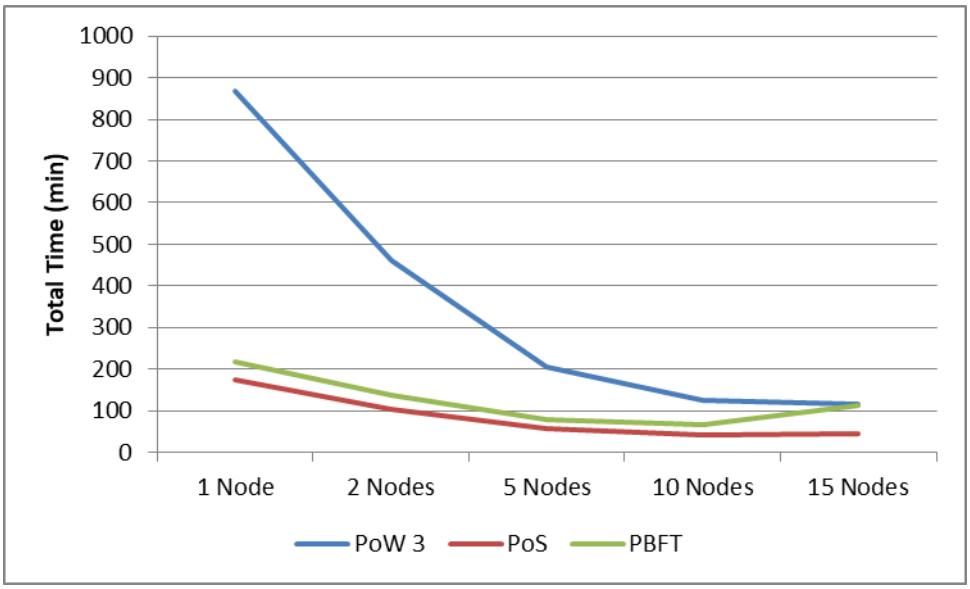


Figura 13. Aplicação da métrica M04.

Do ponto de vista de desempenho pode-se notar que o protocolo PoW é o menos performático, pois necessita de poder computacional para achar uma Hash específica com força bruta, porém é o mais escalável de todos já que a performance isolada de cada nodo não é influenciada pelos demais, apenas pelo seu próprio processamento, já para os protocolos PoS e PBFT a performance vai variar de acordo com a arquitetura modelada, já que a validação dos blocos é coletiva, depende da conexão com outros nodos, entretanto vale ressaltar que dependendo da disposição dos nodos, o nível de segurança varia, pois em um nível com apenas 2 nodos, por exemplo, caso um dos nodos se torne malicioso 50% do nível da rede está infectada, e como os outros níveis apenas "aceitam" os blocos deste nível infectado toda a rede pode ficar comprometida.

4. Conclusão

Esse artigo teve por objetivo apresentar o Demochain, um framework para a criação de redes blockchain híbridas para dispositivos IoT. O trabalho realizado até o momento demonstra que o framework proposto é capaz de auxiliar no desenvolvimento de redes blockchain híbridas para ambientes IoT. O estudo realizado e o framework desenvolvido propiciaram a abstração de diversos conhecimentos, possibilitando assim que trabalhos relacionados possam ser desenvolvidos utilizando este como base em diversos aspectos.

Assim, pode-se concluir que não existe uma solução única se tratando de redes blockchain para dispositivos IoT. No entanto, a adoção de um framework para auxílio no desenvolvimento abre uma ampla possibilidade de novas soluções cada vez mais performáticas em diferentes contextos.

Como trabalhos futuros pode ser apontado a implementação de novos protocolos de consenso nesta arquitetura híbrida, com a possibilidade de cada camada estar rodando o seu próprio consenso, além de novos controles de armazenamento e envio da blockchain, e também utilizar outras métricas de desempenho e segurança para testes com este framework, em diferentes ambientes IoT.

5. Refêrencias

Bitfury Group (2015). "Proof of Stake versus Proof of Work. White Paper". LINK DE ACESSO. Ultimo Acesso: 29/11/2018.

Castro, M. e Liskov, B. (1999). "Practical Byzantine fault tolerance".

Conoscenti, M., Vetrò, A. e De Martin, J. C. (2016). "Blockchain for the Internet of Things: A systematic literature review".

De Angelis, Stefano. (2017). "Assessing Security and Performances of Consensus algorithms for Permissioned Blockchains".

Fernández Caramés, M., Fraga Lamas, D. P. (2017). "A Review on the Use of Blockchain for the Internet of Things".

Freitas, L. (2018). Github com códigos do Demochain. Disponível em https://github.com/LorenzoWF/Demochain. Ultimo acesso em 29/11/2018.

Gartner (2017). "Leading the IoT – Gartner Insights on how to lead in a connected world".

Google (2009). Golang site oficial. Disponível em https://golang.org. Último Acesso: 29/11/2018.

Larimer, Dan (2014). “DPOS Description on Bitshares”.

Li, C. e Zhang, L. J. (2017). “A blockchain based new secure multi-layer network model for Internet of Things”.

Liang, X., Shetty, S., Tosh, D., Kamhoua, C., Kwiat, K. e Njilla, L. (2017). “Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability”.

Ortiz, J. e Gottschlich, N. (2016). Base de dados “Household Power Consumption”. Disponível em https://data.world/databeats/household-power-consumption. Último Acesso: 29/11/2018.

Schmager, Frank (2010). “Evaluating the GO Programming Language with Design Patterns”.

Vukolić, M. (2018). “The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication”.

Zheng, Z., Xie, S., Dai, H. e Wang, H. (2017). “An overview of blockchain technology: Architecture, consensus, and future trends”.

II Workshop Blockchain: Teoria, Tecnologias e Aplicações
SBRC 2019
Sessão Técnica 2

Sensor Store: uma loja de dados de IoT usando Tangle

Caio Martins Barbosa[1,2], Vinícius Fernandes Soares Mota[1], Roberta Lima Gomes[1], José Gonçalves Pereira Filho[1]

[1]Departamento de Informática – Universidade Federal do Espírito Santo (UFES)
Av. Fernando Ferrari, 514, 29075-910 - Vitória - ES, Brasil

[2]Instituto de Tecnologia da Informação e Comunicação do Estado do Espírito Santo (PRODEST), Av. João Batista Parra, 465, 29050-925 - Vitória - ES, Brasil

caio.barbosa@outlook.com, {vinicius.mota, rgomes, zegonc}@inf.ufes.br

Abstract. *In an increasingly connected world, the connected devices that form the so-called Internet of Things (IoT) will become increasingly present in people's daily lives, whether monitoring or acting in the environment around them. The omnipresence of IoT will produce an excessive amount of data on the most diverse physical phenomena. By glimpsing a potential market, this work proposes an architecture for the commercialization of sensor data. Devices data will be published in Tangle, a Distributed Ledger Technologies (DLTs), which will store data immutably and allow payments in IOTA cryptocurrency without the charge of reward fees.*

Resumo. *Em um mundo cada vez mais conectado, os dispositivos conectados, que formam a chamada Internet das Coisas (IoT) se tornarão cada vez mais presentes no cotidiano das pessoas, seja monitorando ou atuando no ambiente ao seu redor. A onipresença da IoT produzirá uma quantidade excessiva de dados sobre os mais diversos fenômenos físicos. Vislumbrando um potencial mercado, este trabalho propõe uma arquitetura para comercialização de dados de sensores. Os dados dos dispositivos estarão publicados na Tangle, uma tecnologia de livros-razão distribuídos (DLTs) que armazenará os dados de forma imutável e permitirá a realização de pagamentos em criptomoeda IOTA sem cobrança de taxas de recompensa.*

1. Introdução

O século XXI será marcado pelo surgimento de uma *commodity* mais valiosa do que petróleo, os dados [The Economist 2017]. Empresas como Amazon, Apple, Google, Facebook e Microsoft disputam uma corrida em que o vencedor se transformará na *Standard Oil Company*[1] da era digital. Essas gigantes da computação sabem o valor e a importância estratégica dos dados para sobrevivência e diferenciação em um mercado

[1] https://pt.wikipedia.org/wiki/Standard_Oil_Company

altamente competitivo. Apple e Amazon são companhias que superaram a marca de 1 trilhão de dólares de valor de mercado [Streitfeld 2018]. Google e Facebook têm receitas baseadas principalmente em publicidade digital, visto que são líderes em anúncios exibidos de acordo com as buscas dos usuários ou com o tipo de conteúdo acessado.

Nesse mercado altamente rentável, os dados produzidos são oriundos das mais diversas fontes, incluindo de dispositivos IoT. Com a previsão da Cisco de 50 bilhões de dispositivos conectados até 2020 [Macaulay *et al*. 2015], a IoT, além de gerar novos e mais sofisticados serviços, também poderá gerar dividendos para os proprietários dos dispositivos. O Google poderá remunerar proprietários de veículos que forneçam localização do automóvel, a fim de compor os seus serviços de mapa de trânsito: Google Maps e Waze. A indústria farmacêutica ao desenvolver novas pesquisas será capaz de comprar dados de frequência cardíaca, pressão arterial, glicose no sangue, horas de sono, atividades físicas, entre outros dados de saúde, de forma a permiti-la superar um dos maiores desafios de pesquisas médicas, o recrutamento de participantes para o estudo.

Considerando os ilimitados cenários de aplicação de sensores, os autores acreditam que para aproveitar o potencial mercado de dados de IoT, mecanismos de comércio precisam ser desenvolvidos de forma que permitam conectar os consumidores com os proprietários dos dispositivos. Tais mecanismos deverão contemplar requisitos que atendam à natureza distribuída, de diferentes capacidades de processamento, armazenamento, comunicação heterogêneos, quando não restritas e, evidentemente, recursos de pagamento.

Este trabalho apresenta uma arquitetura, batizada de Sensor Store, para comercialização de dados de dispositivos utilizando tecnologias de livros-razão distribuídos (DLTs - *Distributed Ledger Technology*) para armazenamento distribuído e pagamentos sem taxas. Na prática, a Sensor Store facilita a conexão entre os consumidores (compradores) e anunciantes (vendedores) de ofertas de dados de sensores.

Devido às características dos dados gerados pela IoT, como tamanho do dado, quantidade e necessidade de escalabilidade, a Sensor Store utiliza a tecnologia Tangle para armazenamento e IOTA para pagamento das transações. A escolha dessa DLT como *backend* da Sensor Store possibilita aos vendedores de dados de IoT e à própria loja desconsiderar questões relativas ao local onde os dados estarão armazenados, ou seja, não há preocupação de manter o funcionamento (licenciamento, backup, atualização de *patchs* de segurança, etc.) de um banco de dados local ou em nuvem, pois os dados dos dispositivos estarão armazenados na Tangle. Além disso, ao vincular o endereço IOTA a um dispositivo, o vendedor é inserido imediatamente em um mercado de dados sem a necessidade de possuir um cartão de crédito, conta bancária ou em empresas de pagamentos on-line, como a PayPal[2]. E, diferentemente da Bitcoin – a mais famosa implementação de Blockchain – o pagamento não sofrerá aplicação de taxas para que ele seja efetivado.

[2] https://www.paypal.com

O restante do documento está organizado como segue. A Seção 2 apresenta dois tipos de tecnologias de livro-razão distribuído. A Seção 3 descreve alguns trabalhos relacionados. A Seção 4 é apresenta o conceito de funcionamento da loja com a Tangle, a arquitetura da loja, bem como o estado atual de implementação. Na Seção 5 dois possíveis cenários de uso da loja são introduzidos. A Seção 6 discute alguns problemas ocorridos durante a implementação. Por fim, a Seção 7 conclui o artigo, tecendo as considerações finais e as perspectivas de trabalhos futuros.

2. Tecnologias de DLTs

A utilização da tecnologia de livros-razão distribuídos (DLTs) se apresenta como um modelo alternativo para o armazenamento de informação uma vez que essa tecnologia se comporta como um tipo de banco de dados que está distribuído entre todos os membros da rede e quaisquer mudanças serão refletidas simultaneamente para todos participantes.

Um caso especial dessas tecnologias são as Blockchains, que implementam a ideia de livros-razão distribuídos por meio de um mecanismo organizacional baseado em encadeamento de blocos de transações. Uma outra tecnologia DLT de destaque é a Tangle que, diferentemente da Blockchain, é implementada em uma estrutura de dados de grafo acíclico direcionado. Ambas as tecnologias são descritas brevemente a seguir.

2.1 Blockchain

A tecnologia Blockchain surgiu em 2008 para apoiar a implementação da criptomoeda Bitcoin [Nakamoto 2008]. Apesar de a idealização original ter sido orientada para pagamentos eletrônicos, as Blockchains foram rapidamente identificadas como de aplicabilidade mais abrangente. Por exemplo, na implementação de gerenciamento de redes virtualizadas [Rebello *et al*. 2018], controle de registros eletrônico de saúde de paciente [Conceição et al. 2018], preservação de documentos digitais acadêmicos [Costa *et al.* 2018], entre outros.

Uma Blockchain opera como um livro-razão que pode registrar transações entre entidades sem um ente centralizador, garantindo assim confiabilidade distribuída no sistema [Wood 2014]. Esses registros são realizados de forma eficiente, permanente e podem ser facilmente verificáveis com base em criptografia. Normalmente, uma Blockchain é gerenciada por uma rede P2P (do inglês *peer-to-peer*), onde os participantes aderem ao protocolo da aplicação e validam os blocos que são criados ao longo da existência da cadeia. Uma vez que um bloco é registrado (inserido na cadeia) pelo coletivo de nós da rede, seus dados não podem ser alterados (imutabilidade) sem a devida alteração de todos os blocos subsequentes a ele.

Na tecnologia Blockchain cada bloco está ligado a apenas um bloco anterior a ele referenciando via um *hash pointer*. No caso de Blockchains como a do Bitcoin, os blocos são adicionados à medida que são validados por um processo matemático de tentativas sucessivas (denominado Prova de Trabalho, ou *Proof of Work*) para determi-

nar o valor que resolve o *hash* criptográfico do bloco de transações. O responsável por adicionar um novo bloco à cadeia recebe uma recompensa, que é um incentivo financeiro para que a processo seja continuamente realizado.

A Blockchain utiliza uma infraestrutura de chaves criptográficas assimétricas para registrar e validar as transações. A criptografia assimétrica prevê a existência de duas chaves uma pública e outra privada, a chave pública é usada para gerar um endereço público para as transações, algo equivalente ao número de uma conta em uma instituição bancária, já a chave privada permite acessar os recursos associados a uma chave pública.

A tecnologia Blockchain possui alguns princípios básicos, são eles segurança, privacidade, descentralização, incentivo à participação na rede e inclusão, que são descritos a seguir.

Atendendo ao princípio segurança, a tecnologia permite que as transações sejam armazenadas de forma segura, sendo virtualmente impossível alterá-las após a inclusão na cadeia e o sua confirmação. Para tanto, seria necessário, como no caso do Bitcoin, o controle de mais da metade do "poder computacional" da rede.

Já no quesito privacidade, uma chave pública não pode ser imediatamente associada a uma pessoa ou instituição, funcionando como um pseudônimo para o usuário. Sendo assim, a tecnologia oferece certo nível de privacidade (pseudo-anonimicidade).

No quesito descentralização, o poder sobre a rede formada pela tecnologia é distribuído entre os participantes da rede, não existe a figura de um órgão central que regula como as transações são realizadas, e as decisões na rede também são tomadas pelo grupo que a forma. Com isso, cada minerador possui uma parcela de "poder de decisão" sobre a rede, proporcional ao seu poder computacional.

No quesito incentivo à participação na rede, existe o papel do minerador. Em geral, os nós da rede empregam energia e poder computacional para validar as transações de um bloco. Mas no caso particular do minerador, ele atua na criação de novos blocos válidos. Para tanto, com base na Prova de Trabalho, uma vez que ele consegue resolver o problema matemático definido pela rede criando um bloco válido, e o seu bloco é inserido com sucesso na cadeia, como recompensa, o minerador recebe novas moedas pelo trabalho realizado.

Por fim, no quesito inclusão, a Blockchain proporciona inclusão econômica, pois permite que recursos sejam transferidos entre dois usuários independentes da posição geográfica dos envolvidos em uma transação. A rede trata as transações sem distinção, taxando cada transação de mesma forma.

2.2 Tangle

No contexto de IoT, em que dispositivos estão constantemente fazendo medições e gerando dados, criptomoedas como a Bitcoin são pouco viáveis para utilização principalmente devido às taxas associadas tenderem ser maiores do que valores a serem cobrados pelos dados dos sensores. Nesse sentido, Popov (2017) desenvolveu a IOTA, uma crip-

tomoeda voltada para a realização de micropagamentos sem cobrança de taxas apoiada na Tangle.

A tecnologia Tangle foi criada para atender as necessidades do projeto IOTA que tem como foco oferecer uma infraestrutura de microtransações para o universo da *Internet of Things.* Sua aplicabilidade, no entanto, não se restringe a pagamentos eletrônicos, mas envolve um escopo bem mais abrangente, tal como ocorre com a tecnologia Blockchain.

Conforme mencionado, diferentemente da Blockchain, em que a implementação é baseada uma lista encadeada de blocos de transações, os quais são adicionados à lista conforme são validados por meio de processo matemático, a Tangle é baseada na implementação de um Grafo Acíclico Direcionado (do inglês, *Directed Acyclic Graph* − DAG). Cada nó do grafo é uma transação e para adicionar uma nova transação na DAG, existe um processo matemático, semelhante à Blockchain, que termina por validar outros duas transações pré-existentes. Justamente devido a essa relação temporal entre as transações, garante-se que o grafo seja acíclico.

A adição de novas transações na Tangle é mais simples do que no Blockchain, sendo assim, ocorre que a validação demanda um esforço computacional bem menor que aquele da mineração na Blockchain, a qual envolve múltiplas transações. Por sua vez, a validação de uma transação pré-existente é simples e pode ser realizada em tempo constante.

As transações na Tangle podem transportar valores monetários ou dados, sejam eles enviados por um carro, monitor cardíaco ou aplicativo de telefone, etc.. O canal MAM, protocolo de comunicação de dados de segunda camada, adiciona a funcionalidade de emitir e acessar um fluxo de dados criptografados [Handy 2017]. Um publicador ao criar um canal MAM envia fluxo de dados por meio de transações para os ouvintes daquele canal, de modo similar a exibição de um vídeo do YouTube[3]. O protocolo de consenso da IOTA adiciona integridade a esses fluxos de mensagens e criptografia de dados. Não sendo possível por parte de terceiros (ouvintes) interferir nas mensagens, por exemplo, enviando *spam* pelo canal. Existem três modos de acesso ao canal: público, restrito e privado. O modo público é acessado por qualquer pessoa que conheça o endereço de publicação dos dados. O modo privado é restrito ao proprietário do canal. E o modo restrito possibilita a qualquer pessoa com o endereço e a chave de acesso consultar o fluxo de dados. A chave de acesso pode ser alterada a qualquer momento pelo proprietário, revogando o acesso aos dados aos ouvintes que não tenham a nova chave.

Na Tangle todos os participantes da rede desempenham a mesma função, isto é, todos os participantes emitem e validam transações. Enquanto que na tecnologia Blockchain isso não é observado. Tal distinção entre dispositivos pode levar a conflitos na mineração de blocos em virtude das recompensas previstas. Nesse caso, uma disputa entre mineradores pode transformar um paradigma de validação concebido originalmen-

[3] https://www.youtube.com/

te descentralizado em um paradigma centralizado, constituído por mineradores de maior capacidade de processamento.

3. Trabalhos Relacionados

Embora a utilização de Blockchain para diversos domínios de problema tenha sido proposta na literatura, poucos trabalhos avaliaram a utilização da Tangle como tecnologia de controle de transações. Os requisitos associados a um sistema de venda de dados de sensoriamento obtidos por dispositivos conectados tornam a Tangle uma tecnologia viável para atender a demanda de centenas de micro vendas ocasionadas pela geração desses dados.

Com uma proposta de negócio semelhante à da Sensor Store, o aplicativo Wibson [Travizano *et al.* 2018] também propõe um *marketplace* para que os usuários disponibilizem os seus dados pessoais para comercialização, de forma análoga aos dispositivos IoT da Sensor Store. Contudo, diferentemente da Sensor Store, o Wibson é baseado no uso da tecnologia Blockchain para prover a infraestrutura de loja descentralizada.

Mišura e Žagar (2017) propõem um modelo de mercado de dados de IoT centralizado tanto no catálogo de dados quanto na própria troca, em que os provedores de dados da IoT registram suas ofertas e ajudam os consumidores a encontrá-los. O modelo proposto apoia-se em uma arquitetura tradicional web: servidor de banco de dados e aplicação (web). Portanto, os dados dos dispositivos estariam armazenados em um banco de dados.

Wörner e Bomhard (2014) abordam uma solução distribuída para criação de um mercado de dados de IoT com pagamento direto em Bitcoin e troca de dados *peer-to-peer* através de Blockchain. Os autores propõem um repositório central para registro dos sensores. O comprador acessa o repositório para selecionar o sensor e pagar pelos dados em Bitcoin. Um websocket é mantido aberto entre comprador e o repositório aguardando a confirmação de pagamento e publicação dos dados do sensor na Blockchain. Como os dados são publicados diretamente na Blockchain sem criptografia, qualquer usuário pode acessá-los.

A IDMoB [Özyılmaz *et al.* 2018] introduz o conceito de uma loja de dados de IoT implementada como um contrato inteligente Ethereum e o Swarm[4] sendo usado como a plataforma de armazenamento distribuído. Ou seja, os autores propõem que o que local para registro dos sensores seja construído como um contrato inteligente Ethereum. Logo, o comprador de dados deverá pesquisar por dispositivos e realizar pagamentos usando funcionalidades de *smart contract*. Os dados de dispositivos não são publicados diretamente na Blockchain, eles são publicados nos arquivos Swarm. E os dados são publicados em intervalos de tempo superiores a 30 minutos.

[4] https://swarm-guide.readthedocs.io/en/latest/

Este trabalho se diferencia dos demais por propor o desenvolvimento de uma loja de IoT suportada pela Tangle, uma tecnologia de livro-razão distribuída em uma estrutura DAG. Além de armazenar os dados publicados pelos sensores, a tecnologia Tangle provê aos consumidores a realização de micropagamentos sem cobrança de taxa e o consumo de dados em tempo real diretamente do vendedor, pois a publicação de dados não demanda esforço computacional. Portanto, o consumidor pode "assinar" um feed e obter os dados em tempo real. Enquanto que na Blockchain a latência de publicação de dados seria muito maior em função do tempo necessário para empacotar as transações e validar o bloco, logo há atraso na publicação de dados.

4. Sensor Store

Considerando as previsões de crescimento da presença de dispositivos de IoT – sensores e atuadores – a proposta deste trabalho é o desenvolvimento de uma loja que possibilite a realização de comércio de dados de tais dispositivos. Os proprietários dos objetos (vendedores) divulgarão suas ofertas de dados na loja, os compradores farão solicitações de compra e a loja gerenciará os contratos. O objetivo da Sensor Store é ser o Mercado Livre[5] para dados de dispositivos, ou seja, um local para que compradores e vendedores, independentemente de serem pessoas físicas ou pessoas jurídicas, possam realizar comércio diretamente entre eles.

De acordo com objetivo apresentado, o papel da Sensor Store é definido por: i) possibilitar os vendedores cadastrarem seus dispositivos e, consequentemente, os anúncios de oferta de dados; ii) identificar o pagamento feito aos vendedores e iii) gerenciar os contratos de compra. A Figura 1 exemplifica o funcionamento básico da Sensor Store.

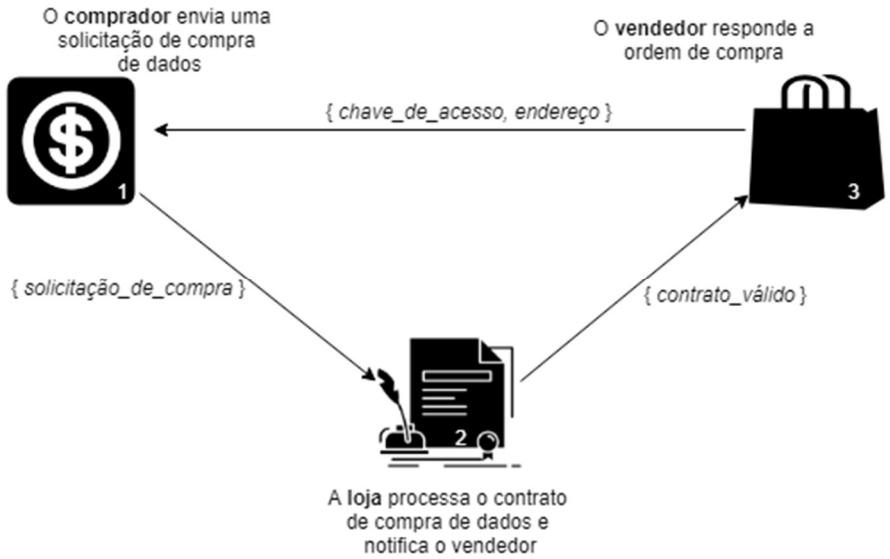


Figura 1. Representação do funcionamento básico da loja

[5] https://www.mercadolivre.com.br/

Em 1) o comprador submete uma solicitação de compra de um anúncio. No passo 2) a loja recebe a solicitação do comprador e passa a monitorar o pagamento a fim de comunicar ao vendedor da compra de anúncio e a existência de um contrato. Uma vez que o vendedor é notificado, no passo 3) ele enviará ao comprador o endereço e a chave de acesso aos dados do dispositivo.

Posto o funcionamento básico da loja, torna-se imperioso o esclarecimento de conceitos que foram abstraídos na representação da Figura 1. A loja é uma aplicação Web de divulgação de anúncios de ofertas de dados, cuja interação com os dispositivos e seus usuários (vendedores e compradores) é estabelecida por meio da tecnologia Tangle. Um comprador de dados ao selecionar uma oferta na Sensor Store executará uma transação de *tokens IOTA* para um endereço Tangle da carteira do vendedor. Por sua vez, a loja monitorará a rede Tangle para identificar a confirmação do pagamento por meio da mudança de estado de "pendente" para "confirmado". Dado que o pagamento seja confirmado, a loja controlará a vigência do contrato e notificará o vendedor por meio de uma transação (zero *IOTA*) a autorização de acesso do comprador (endereço Tangle). O vendedor, ao receber a mensagem de autorização, realizará uma transação (zero *iota*) para o endereço do comprador informando a chave de acesso e o endereço do canal MAM em que os dados serão publicados.

4.1. Arquitetura de Implementação

A proposta de arquitetura para a Sensor Store baseada na tecnologia Tangle é apresentada na Figura 2, onde se vê os seguintes componentes:

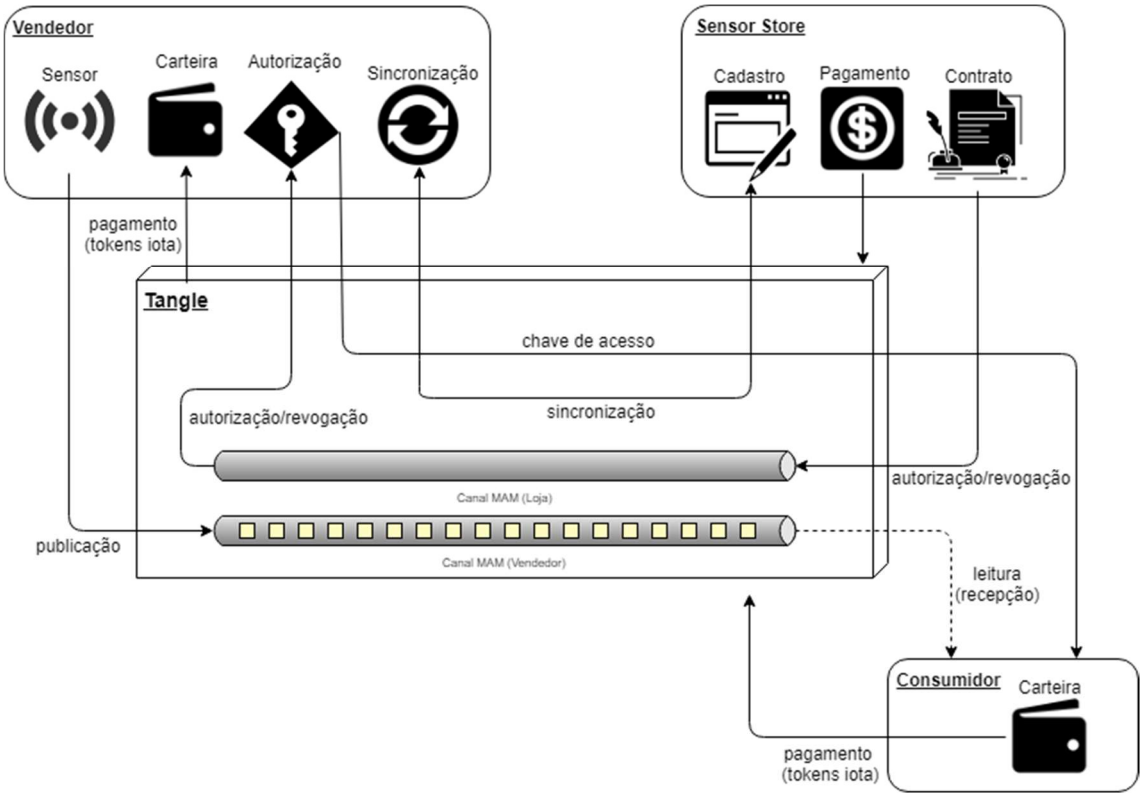


Figura 2. Arquitetura da Sensor Store

- **Tangle**: um *distributed ledger* capaz de realizar transações de pagamento entre o comprador e o vendedor, transações de envio de informações (mensagens) para os dispositivos, além de permitir aos dispositivos a publicação de dados de poluição do ar, temperatura corporal, pressão de pneus, velocidade do veículo, localização, entre outros dados de sensores.
- **Sensor**: é componente (dispositivo IoT) responsável por gerar as medições. Ele deverá ser capaz de publicar os dados na Tangle através do canal MAM do vendedor.
- **Carteira**: uma carteira eletrônica é responsável por armazenar a *seed* (chave privada) e os endereços (chaves públicas) dos usuários. É por meio das chaves públicas e privadas que os usuários (comprador e vendedor) recebem ou enviam *tokens iota* na solução.
- **Canal MAM do vendedor**: este componente é responsável por registrar as medições dos dispositivos na Tangle através de um canal MAM restrito. Os consumidores poderão ler os dados ao passo que as medições forem publicadas, pois um canal MAM tem característica de transmissão (*stream)* de dados.
- **Sincronização**: para que o dispositivo possa vender seus dados na loja, ele precisa ser cadastrado e ser capaz de perceber os eventos gerados pela loja em seu canal MAM privado. Para o dispositivo conseguir ler esses eventos, a loja inicia um processo de negociação para estabelecimento de comunicação similar ao *handshake*[6] do TCP.
- **Autorização**: durante a vigência do contrato, a loja publica em seu canal MAM o início e o término. O módulo de autorização deve ser capaz de perceber esses eventos e enviar a chave de acesso para um novo consumidor, bem como trocar a chave e enviá-la para os consumidores com contratos vigentes, quando um contrato expirar.
- **Canal MAM da Sensor Store**: meio de comunicação da loja com os dispositivos. A loja publicará no canal mensagens direcionadas de autorização ou revogação de acesso à medida que compras sejam realizadas ou contratos expirem.
- **Cadastro de dispositivos**: funcionalidade similar ao cadastro básico dos produtos como de qualquer outra loja. O vendedor deverá fornecer informações básicas sobre o dispositivo, tais como: nome, localização, a propriedade e a unidade de medida, além do modelo de comercialização (tempo ou cota de uso).
- **Pagamento**: os pagamentos entre vendedores e consumidores são realizados pela criptomoeda IOTA. A loja permitirá a seus usuários conectar suas carteiras e efetuarem os pagamentos.
- **Gestão de contratos:** este componente atua em dois momentos distintos: i) início de vigência do contrato: a atuação ocorre pelo monitoramento da

[6] https://en.wikipedia.org/wiki/Handshaking#TCP_three-way_handshake

transação na Tangle de transferência de *tokens iota* entre o comprador e o vendedor até que o estado da transação mude de "pendente" para "confirmado". Concretizada a confirmação de pagamento, este componente publicará no canal MAM da loja as informações de autorização de acesso do comprador; ii) fim de vigência do contrato: uma vez que o contrato expire, o gestor de contrato publicará no canal MAM da loja as informações de revogação de acesso do comprador.

4.2. Transações

Uma transação é a unidade básica de informação manipulada pela solução. Os seguintes tipos de transações são definidos:

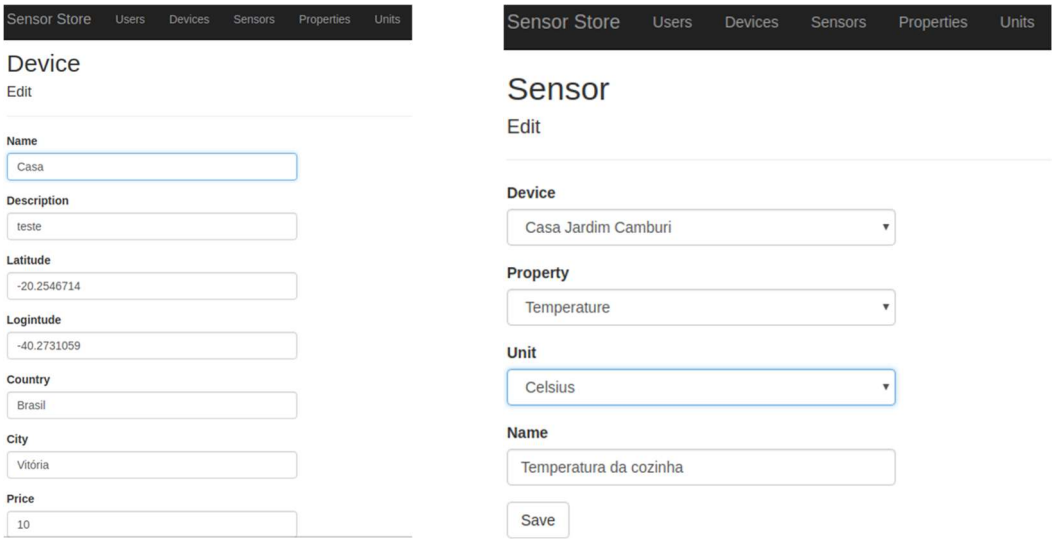
- **Pagamento:** transações de pagamentos transferirão *tokens iota* da carteira do comprador para o vendedor. Esse tipo de transação é nativa da Tangle.
- **Publicação:** a publicação de dados é realizada no canal MAM do tipo restrito do vendedor. Esse tipo de transação não exige transferência de *tokens iota*.
- **Autorização/Revogação:** este tipo de transação são mensagens enviadas da loja pelo componente Gestão de Contrato para os dispositivos. O envio de mensagens não gera transferência de *tokens iota*.
- **Sincronização:** as transações de sincronização são realizadas para que a loja informe o endereço do seu canal MAM e conheça o endereço do dispositivo ou quando ocorre a necessidade de trocar a chave de acesso ao canal MAM do dispositivo. Todas as transações geradas na sincronização não consomem *tokens iota*.
- **Chave de Acesso:** este tipo de transação é realizado no início de vigência de um contrato diretamente para o novo consumidor ou no término de vigência de contrato, pois será necessário trocar a chave de acesso e enviá-la para todos os consumidores com contrato em vigência. Essa transação não transfere *tokens iota*.

É importante ressaltar que apesar da Figura 2 conter uma transação "leitura (recepção)", trata-se apenas de uma representação para melhor entendimento da arquitetura. Não acontecendo efetivamente uma transação para ler dados de um canal MAM.

4.3. Estado atual da implementação

O desenvolvimento até o momento contempla uma versão inicial da Sensor Store para cadastro de usuários, cadastro de dispositivos e gestão de contrato. Além disso, uma versão do vendedor (dispositivo) que sincroniza com a loja e publica os dados na Tangle. Portanto, o foco inicial foi o desenvolvimento de funcionalidades que permitissem o cadastro e sincronização de dispositivos, gestão de contrato e a publicação de dados.

A Figura 3 (a) exibe a tela para cadastro de dispositivo. Informações básicas como nome, localização e valor são registradas. A Figura 3 (b) exibe a tela para cadastro de sensor, em que as informações básicas do sensor são registradas em conjunto com o dispositivo no qual o sensor está embutido, além de informações sobre a propriedade e a unidade de medida.



(a) (b)

Figura 3. Cadastro de dispositivo e sensor

5. Exemplos de caso de uso

Esta seção apresenta dois casos de uso em que a Sensor Store facilitaria a procura e compra de dados de sensores: *Healthcare* e transporte público. Embora simples, estes cenários servem para ilustrar o potencial da arquitetura proposta.

Considerando um cenário em que uma empresa startup[7] de plano de saúde – *healthtech* (junção de *health* "saúde" em inglês e *tech*, abreviação de *technology*) – deseja estabelecer-se no mercado de Vitória (ES). E que essa *healthtech* pela essência das empresas startups de fornecerem serviços inovadores tem um modelo de negócio em que o cálculo do plano de saúde leva em consideração características endêmicas, por exemplo, malária, febre amarela, câncer, etc.. Para o caso de Vitória, a startup tomou por referência o trabalho de Freitas *et al.* (2016), cujo estudo aponta grande número de morbidade e doenças respiratórias ocasionados pela poluição atmosférica.

Nesse cenário, a *healthtech* consultaria a Sensor Store em busca de dados de sensores de poluição atmosférica de material particulado fino (PM10), dióxido de enxofre (SO2) e ozônio (O3); além de registros de temperatura e umidade. E compraria esses

[7] https://pt.wikipedia.org/wiki/Startup

dados diretamente dos proprietários de sensores, sem a necessidade da empresa criar uma estrutura sensores espalhados pela cidade para atender o seu negócio.

O segundo cenário considera que uma companhia de transporte de passageiros de uma grande cidade deseja desenvolver um portal web de monitoramento de seus ônibus para disponibilizar informações mais precisas sobre as suas viagens, de forma a permitir aos seus usuários o acompanhamento de possíveis atrasos de veículos em função da situação do trânsito. Para isso, a companhia processará leituras de seus próprios veículos e de terceiros transmitidos ao vivo na Tangle. O acesso aos dados de terceiros se dará por compra de dados na Sensor Store. Uma vez que a companhia tenha recebido as chaves de acesso dos veículos de terceiros, ela poderá "se inscrever" no canal MAM de cada veículo e acessar o fluxo de dados, como se fosse um *feed* de RSS.

6. Discussões e limitações da arquitetura

A Tangle ainda é uma tecnologia em desenvolvimento e, como tal, apresenta certa instabilidade e inconsistência. O ambiente de teste da Tangle por vezes gera erros, sendo que muitos deles são pouco documentados. Um problema ocorrido com a publicação de dados no canal MAM era gerado por usar o mesmo *seed* ao iniciar o canal, o que obrigou a publicação de uma mensagem inicial. Caso contrário, ao se recuperar[8] a mensagem, um erro seria gerado.

O pioneirismo da Tangle também é afetado pela escassez de recursos que permitam interagir com ela, por exemplo, bibliotecas, *frameworks*, documentação, códigos-fontes de exemplos, etc.

O processo de revogação, em função da estrutura atual da Tangle, é uma limitação da arquitetura da loja, pois uma vez que um contrato tenha sido encerrado, o componente de gestão de contratos comunicará ao dispositivo por meio do canal MAM o encerramento. Logo, o dispositivo deverá trocar a chave de acesso do seu próprio canal MAM e informar à loja por meio do processo de sincronização sobre a alteração. A loja publicará então uma mensagem em seu canal com a nova chave de acesso encriptada com a chave pública dos compradores. Portanto, esse processo obriga aos compradores ficarem escutando o canal da loja para verificar as possíveis revogações de acesso.

7. Conclusões

Neste artigo, propomos a criação de um local para divulgação de ofertas e comercialização de dados de dispositivos IoT promovida por meio da Tangle, uma *distributed ledger technology* subjacente à loja que provê transferência de *tokens iota* (criptomoeda) e disseminação de mensagens (medições dos dispositivos). A loja, componente web da arquitetura, essencialmente cumpre o papel de ser um *hub* de anúncios de vendas de dados

[8] https://github.com/iotaledger/mam.client.js/issues/5

de dispositivos, similar ao que é o Mercado Livre. Nela os vendedores se registram, cadastram e vendem os seus anúncios para outros usuários, enquanto que a Tangle provê a moeda de pagamento (*IOTA*) e atua como local em que os dados (produto vendido) estão armazenados.

Portanto, mesmo que este trabalho seja a demonstração de conceitos básicos e de um protótipo em evolução para venda de dados de dispositivos IoT em troca de criptomoeda, a proposta apresenta-se como uma solução bastante promissora para conectar vendedores (proprietários de dispositivos) e compradores de dados (consumidores). Por outro lado, por ser uma tecnologia ainda em desenvolvimento, é necessário fazer uma avaliação de desempenho robusta da tecnologia, em cenários diversos e de maior complexidade, para poder avaliar melhor a tecnologia.

Como trabalhos futuros, queremos disponibilizar um protótipo da Sensor Store ao público, o que certamente ajudará no processo de avaliação da robustez da solução. Essa implementação incluirá um modo de conectar a carteira do comprador na loja de forma a permitir fazer o pagamento pela própria loja. Também objetiva-se desenvolver uma solução de *smart contract* em Blockchain com Ethereum. Um aspecto importante a ser considerado no futuro é a adição de novos recursos da Tangle para canal MAM. Há expectativa de que em uma nova versão, haverá a possibilidade de se criarem múltiplos canais. Tal característica simplificará o processo de revogação de acesso.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), do CNPq, da FAPES, do PRODEST e do projeto RNP FUTEBOL.

Referências

Conceição, A.F., Silva, F.S.C., Rocha, V., Locoro, A. e Barguil, J.M.M (2018) "Eletronic Health Records using Blockchain Technology", Workshop Blockchain 2018, Campos do Jordão-SP. Simpósio Brasileiro de Redes de Computadores, 2018.

Costa, R., Faustino, D., Lemos, G., Queiroga, A., Djohnnatha, C., Alves, F., Lira, J. e Pires, M. (2018) "Uso Não Financeiro de Blockchain: Um Estudo de Caso Sobre o Registro, Autenticação e Preservação de Documentos Digitais Acadêmicos", Workshop Blockchain 2018, Campos do Jordão-SP. Simpósio Brasileiro de Redes de Computadores, 2018.

Freitas, C.U., Leon, A. P., Junger, W. e Gouveia, N. (2016) "Poluição do ar e impactos na saúde em Vitória, Espírito Santo. Revista Saúde Publicação. vol.50, 4.

Handy, P. (2017) "Introducing Masked Authenticated Messaging", https://blog.iota.org/introducing-masked-authenticated-messaging-e55c1822d50e

Macaulay, J., Buckalew, L. e Chung, G., (2015) "Internet Of Things In Logistics".

Mišura, K. e Žagar, M. (2016) "Data marketplace for Internet of Things," in Proceedings of the 1st IEEE International Conference on Smart Systems and Technologies, SST 2016, pp. 255–260, Croatia, October 2016.

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

Popov, S. (2018), "The Tangle" – Version 1.4.3 disponível em: https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf.

Rebello, G.A.F., Alvarenga, I.D., Sanz, I.J. e Duarte O.C.M.B. (2018) "SINFONIA: Gerenciamento Seguro de Funções Virtualizadas de Rede através de Corrente de Blocos", Workshop Blockchain 2018, Campos do Jordão-SP. Simpósio Brasileiro de Redes de Computadores, 2018.

Streitfeld, D., (2018), "Amazon Hits $1,000,000,000,000 in Value, Following Apple", https://www.nytimes.com/2018/09/04/technology/amazon-stock-price-1-trillion-value.html.

The Economist, (2017) "The world's most valuable resource is no longer oil", https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data.

Travizano, M., Minnoni, M., Ajzenman, G., Sarraute, C., Della Penna, N. (2018), "Wibson: A decentralized marketplace empowering individuals to safely monetize their personal data" - disponível em: https://wibson.org/wp-content/uploads/2018/10/Wibson-Technical-Paper-v1.1.pdf

Wood, G. (2014). "Ethereum: A secure decentralised generalised transaction ledger". *Ethereum Project Yellow Paper.* https://github.com/ethereum/yellowpaper.

Wörner, D. e Von Bomhard, T. (2014) "When your sensor earns money: Exchanging data for cash with Bitcoin," inProceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 295–298.

Editora Científica Autônoma e Distribuída sobre Blockchain Privada

Bruno Evaristo[1], **Vagner Nascimento** [1], **André Defrémont**[1], **Billy Pinheiro**[1], **Antônio Abelém**[1]

[1]Grupo de Pesquisa em Redes de Computadores e Comunicação Multimídia (GERCOM)
Universidade Federal do Pará (UFPA)
Caixa Postal 470 – 66075-110 – Belém – PA – Brasil

{ebec2012,andre.def93}@gmail.com e {vagner,billy,abelem}@ufpa.br

Abstract. *The process of publication of scientific articles is made globally by publishers through the Internet and using systems of submission, review and publication. These systems are very expensive, financially as well as computationally, making scientific knowledge restricted to a declining audience. There are some initiatives that try to make the editing process open and aim to promote more democratic access to it. However, in spite of the effort of democratizing the access to scientific production, two challenges are evident: the first is the process of submission, review and publication, and the relationship between authors and publishers is unequal and the second is related to the infrastructure of a publisher who normally responds to the article's publication rights. So, for the sake of giving more governance to the intermediary (editor) over the publication process, we propose an Autonomous and Distributed Scientific Publisher (Distributed Autonomic Scientific Publisher - DASP), an autonomous scientific publisher that works over private(permissioned) blockchain, making it possible for every process to be audited and that ensures the authorial rights to the involved researchers on the publication of each article, lowering the cost for readers with the withdraw of intermediaries (publishers). proof of concept, it was developed a publisher environment with the basic processes for submission and review, to validate the proposal.*

Resumo. *O processo de publicações de artigos global através da Internet utilizando sistemas de submissão, revisão e publicação. Tais sistemas tem um elevado custo financeiro e computacional, tornando o conhecimento científico restrito a um público cada vez menor. Existem iniciativas que tentam tornar o processo de submissão e avaliação abertos e visam promover o acesso mais democráticos a ele. Entretanto, dois desafios ficam evidentes, o primeiro é a relação desigual entre autores e editoras durante os processos de submissão, revisão e publicação; o segundo está relacionado a infraestrutura de uma editora que normalmente responde pelos direitos de publicação do artigo. Neste sentido, visando dar maior governança ao processo de publicação de artigos, nós propomos a Editora Científica Autônoma e Distribuída (Distributed Autonomic Scientific Publisher - DASP), uma editora científica autônoma funcionando sobre uma blockchain privada(permissionada) que permite que todo o processo seja auditado, garantindo os direitos autorais aos pesquisadores envolvidos na publicação de cada artigo e diminuindo o custo para os leitores com a retirada de intermediários. Como prova de conceito foi desenvolvido um ambiente editorial com os processos básicos de submissão e revisão para validar a proposta.*

1. Introdução

O método científico permite que as explicações para os fenômenos sejam testados, analisados, discutidos e revisados por pares distintos. Esta validação é realizada através do processo de publicação de artigos, no qual a comunidade científica certifica se os resultados no artigo são suficientes para comprovar as hipóteses levantadas [Job et al. 2009].

Atualmente, o processo de publicações de artigos científicos é realizado de maneira global por grandes editoras como IEEE[1] e ACM[2] e outras através do uso da Internet e de sistemas de submissão, revisão e publicação como o JEMS[3] e EDAS[4]. Neles, cada autor envia seus artigos, os quais serão avaliados por revisores externos e, eventualmente publicados de forma impressa ou digital.

Em consonância com a crescente migração para o meio digital, o setor financeiro passou a criar e manipular digitalmente seus ativos, culminando com o aumento no interesse da indústria e da academia pelo desenvolvimento de criptomoedas e de sua tecnologia base, a *blockchain*. Esta permite que informações sejam armazenadas de forma descentralizadas, com tolerância a falha e auditável [Zheng et al. 2016].

As aplicações que rodam sobre uma rede *blockchain* são chamadas de *Decentralized Applications* (Dapps) e são formadas, principalmente, por partes lógicas executadas na Blockchain, denominadas de contratos inteligentes (*smart contracts*). Estes expressam a lógica de forma confiável e verificável, tornando-se uma boa solução para estabelecer confiança entre entidades distintas [Frantz and Nowostawski 2016].

Neste contexto, para prover serviços de revisões e publicações de artigos científicos de maneira transparente, propomos a Editora Científica Autônoma e Distribuída *(Distributed Autonomic Scientific Publisher - DASP)*. Nela, são consideradas características de editoras conhecidas e seu processo de gestão, com a diferença que na DASP propomos que a própria comunidade gerencie o processo. Assim, temos uma abordagem mais ampla e transparente, onde todas as políticas de consenso estão vinculadas aos contratos inteligentes, garantindo melhor gestão do processo de publicação.

Como contribuições científicas deste artigo, destacamos a adaptação do modelo de Organizações Autônomas Descentralizadas *(Decentralized Autonomous Organization - DAO)* para o contexto de publicações científicas, usando um esquema de comprometimento não iterativo para troca de chaves criptográficas dos autores e revisores com a DASP, mantendo-os anônimos. A outra contribuição é a infraestrutura fornecida pela DASP aos autores e revisores que permite escalabilidade, confiança, segurança e auditabilidade.

Este trabalho está estruturado da seguinte forma: A Seção 2 introduz as tecnologias empregadas na DASP; A Seção 3 discute os trabalhos relacionados; A Seção 4 descreve a DASP em detalhes; A Seção 5 apresenta a prova de conceito da DASP; e, por fim, a Seção 6 conclui esse trabalho e aponta os trabalhos futuros.

[1]https://www.ieee.org/
[2]https://www.acm.org/
[3]https://jems.sbc.org.br/
[4]https://edas.info/

2. Referencial Teórico

Nesta seção, apresentamos a base teórica para discutir o processo de gestão por parte das editoras nas publicações de artigos científicos. Assim como, as tecnologias presentes em redes *blockchain*, que viabilizam a criação de contratos inteligentes e o armazenamento distribuído de arquivos.

2.1. O processo de Publicação de Artigos

O processo de publicação de um artigo científico envolve o autor, o revisor, o editor, a editora e o leitor. Cada um desempenha um papel específico no sistema, podendo a mesma pessoa desempenhar diferentes papéis, porém, não concorrentemente [Ferraz and Navas 2016]. A seguir detalharemos as funções exercidas por cada participante em uma editora científica e descreveremos os modelos de revisão comumente usados por elas.

- **Autor**: Realiza a escrita do artigo e é responsável pela submissão do mesmo. Todo o trabalho científico e até mesmo a diagramação do conteúdo a um formato pré-estabelecido é de sua responsabilidade;
- **Revisor**: Avalia os artigos e atribue notas que determinam a aceitação do artigo. Não é raro que autores de um determinado períodico sejam convidados a compor o corpo de revisores do mesmo. Tal função é, normalmente, não remunerada, mas de reconhecido prestígo dentro do seu respectivo campo de atuação;
- **Editor**: Responsável por receber as revisões, organizar os artigos aceitos e gerar a publicação final;
- **Editora**: Organizam os editores e revisores, fazendo as chamadas para publicações, nas quais os autores submetem seus artigos. Dependendo da organização do evento, os autores e leitores são taxados de alguma forma para realizarem a publicação ou leitura do trabalho;
- **Leitor**: Acessa os artigos que foram publicados, podendo eventualmente pagar por isso.

Com o avanço das tecnologias de processadores de texto e diagramação, como, por exemplo, (LaTeX[5]), os trabalhos dos editores e das próprias editoras são minimizados, retirando todo o esforço manual do processo de diagramação por parte dos editores, restando a estes a escolha dos revisores e unificação das notas; e às editoras a escolha dos editores.

O processo de publicação é iniciado com a submissão do artigo seguida da distribuição destes para os revisores previamente cadastrados, alguns sistemas já fazem esse trabalho automaticamente, como o *JEMS*, substituindo o fator humano que pode influenciar no processo. Na sequência, cada revisor recebe uma notificação de que uma revisão foi solicitada podendo aceitar ou negar. Esta dinâmica entre editoras, revisores e autores pode variar já que existem inúmeros modelos de revisões propostos por pesquisadores. Entre os principais modelos, podemos citar:

- **Halfblind**: Somente os revisores conhecem os nomes dos autores do artigo.
- **Fullblind**: Nem o autor, nem o revisor sabem a identidade um do outro.

[5]https://www.latex-project.org/

- **Open peer review**: A identidade dos autores e dos revisores são reveladas, tornando conhecido todos os envolvidos no processo;
- **Triple-blind**: A identidade dos autores, revisores e editores é oculta;
- **Crowdsourced peer review**: Permite que a comunidade em geral contribua para o processo de revisão;
- **Open pre-review manuscripts**: Os trabalhos ficam acessíveis (via Internet) antecipadamente, ou em sincronia com qualquer procedimento formal de revisão por pares. Normalmente, esse tipo de revisão é usado em repositórios institucionais.

Um levantamento abordando a avaliação por pares lista a saturação dos revisores em relação aos métodos existentes e alguns vícios de avaliações, já que no meio acadêmico, um grupo de estudo normalmente reconhece o trabalho ou a área de atuação que está envolvida um outro grupo, o que pode acabar gerando vantagens avaliativas comparado com grupos acadêmicos de estudos não tão conhecidos [Mulligan et al. 2012].

2.2. Blockchain

A tecnologia *blockchain* funciona como um livro de registros distribuído, permitindo a troca de recursos digitais que não são emitidos ou controlados por uma autoridade central. Cada transação é assinada com uma chave PGP[6], garantindo a autenticidade, e o conjunto de transações são organizadas em blocos ligados por uma árvore binária formada com os *hash* das últimas transações [Greve et al. 2018]. Desta forma, as informações armazenadas na *blockchain* são consideradas de alta integridade [Tschorsch and Scheuermann 2016].

São acordos automatizados nos quais a execução é realizada por meio de ambientes de desenvolvimento [Cachin 2016], como *Chaincode*, no *Fabric* [7] e no *Ethereum* [8]. Esses contratos, nada mais são que condições ou regras criadas pelo próprio usuário. Assim, quando as premissas são alcançadas, uma condição pré-estabelecida (programada) é executado automaticamente.

Os contratos inteligentes são códigos auto-executáveis usados para configurar as redes baseado em *blockchain*, habilitam as Organizações autônomas descentralizadas (Decentralized Autonomous Organization - DAO), que são organizações de decisões descentralizada, conhecidas também como iniciativas de fundo de centro vazio, onde cada membro da aplicação tem seu poder de votação baseado em regras pré-estabelecidas e flexíveis via contratos inteligente, que rodam de forma autônoma e transparente em uma *blockchain* [DuPont 2017].

2.3. InterPlanetary File System (IPFS)

Fala-se de aplicações descentralizadas e distribuídas, porém é necessário fornecer mecanismo de armazenamento em larga escala. É nesse contexto que o IPFS surge, como uma alternativa de conectar dispositivos computacionais através de um sistema de arquivos, ou seja, um banco de dados distribuído [Benet 2014].

O IPFS cria uma estrutura simples e flexível na rede através de nós conectados, gerando uma árvore de dados com *hashes*. Após a instalação do IPFS, o usuário gera um

[6]https://www.openpgp.org/about/
[7]https://hyperledger-fabric.readthedocs.io/en/release-1.3/chaincode.html
[8]http://www.ethdocs.org/en/latest/contracts-and-transactions/contracts.html#what-is-a-contract

primeiro nó, o qual permite ler e interagir com outros *hashes*, formando assim a árvore de *hashes*. O IPFS também gera um *daemon*, acessível via *CLI* e *interface web*, onde é possível gerenciar, adicionar e organizar as identidades e suas conexões correspondentes.

3. Trabalhos Relacionados

Nesse cenário detalhamos abaixo cada característica citada acima e por fim geramos uma tabela comparativa entre as propostas que contribuem no processo de publicação.

- **Gestão de Processo**: Essa característica demonstra os mecanismos administrativos que regem o processo editorial, baseado nas pesquisas relacionados aos modelos clássicos do *Institute of Electrical and Electronics Engineers* (IEEE)[IEEE 2013], Sociedade Brasileira de Computação (SBC)[SBC 2015] e *Association for Computing Machinery* (ACM) [Shurkin, Joel 2017], fica nítido o modelo de governança centralizada, em contrapartida as propostas de ciência aberta tem como premissa descentralização da gestão, nossa proposta segue essa premissa, onde não se tem uma entidade central administrando o processo.
- **Banco de Dados**: O armazenamento das informações em um processo editorial tradicional é centralizada em repositórios mantidos pelas editoras. Até mesmo iniciativas visando descentralização, como *EUREKA* [EurekaPlatform 2018] precisam de um servidor central organizando as informações. Nossa proposta (DASP) tem como premissa o uso de um banco de dados distribuído (IPFS) assim como a proposta *Open Science Network* (OSN)[Partners 2018]
- **Auditoria**: Registros de informações de artigos, processos de revisão, e históricos em geral, são extremamente difíceis de serem encontrados nos modelos clássicos, demostrando a ausência de transparência no processo. Nesse ponto, *Decentralized Research Platform* (DEIP)[DEIP 2018] e a DASP surgem com a idéia do uso de *blockchain* como mecanismo de registro público, uma vez que partimos da idéia que qualquer membro pertencente a comunidade que fará uso da solução, poderá ter acesso a qualquer tipo de informação de histórico em geral, fornecendo transparência e confiança entre os entes envolvidos no processo.
- **Direitos Autorais**: Normalmente os lucros e dividendos, majoritariamente, ficam com as editoras por conta da organização e gestão do processo. Na contramão desse cenário, EUREKA, OSN e DEIP fornecem mecanismos próprios de taxação, porém usando a rede *Ethereum*[9], que é uma rede pública, sem um atrelamento direto entre os entes virtuais e os entes físicos, dificultando a segurança. Tendo isso em vista, na DASP o autor tem o poder de decidir sobre o futuro da sua produção, onde pode deixar o acesso totalmente público ou cobrando alguma taxa mínima, através de uma *blockchain* permissionada, sem intermediários (editora). A ideia é ser justo com quem mais trabalha em todo processo de submissão e revisão.
- **Inovação**: Observamos que os sistemas editoriais tradicionais apresentam dificuldade em expor transparência. Partindo da ideia de descentralização na governança e nos mecanismos estruturais, EUREKA, DEIP e OSN surgem com o princípio de inovar os mecanismos de publicação, trazendo a possibilidade de classificação de pesquisas entre os cientistas, além de focar na facilitação da disseminação de trabalhos acadêmicos. Visando o cenário inovador dessas propostas, a DASP surge

[9]https://www.ethereum.org/

com a idéia de humanizar ainda mais o processo, com a idéia de aproximar o autor do revisor, possibilitando o compartilhamento de discussões sobre as revisões através do uso de revisões secundárias, além disso a DASP, fornece a possibilidade de gerar instâncias da aplicação, permitindo criar processos editoriais de comunidades específicas ao instanciar a DASP.

Características	Sistemas Editoriais Clássicos			Propostas de Ciência Aberta (Blockchain)			
	IEEE	ACM	SBC	EUREKA	DEIP	OSN	DASP
Gestão de Processo	Centr.	Centr.	Centr.	Descen.	Descen.	Descen.	Descen.
Banco de Dados	Centr.	Centr.	Centr.	Centr.	Descen.	Descen.	Descen.
Auditória	Não	Não	Não	Sim	Sim	Sim	Sim
Direitos Autorais	Desigual	Desigual	Desigual	Desigual	Desigual	Desigual	Igualitário
Inovação	Nada Relac.	Nada Relac.	Nada Relac.	Descen.	Descen.	Descen.	Descen.

Tabela 1. Comparação das propostas.

A Tabela 1 apresenta uma comparação das características apresentadas anteriormente nesta seção, mostrando os ambientes computacionais voltados à governança de sistemas editoriais clássicos e as propostas de editoras que visam inovar no processo de governança, tomando como base a transparência, descentralização do processo e uma relação mais justa entre os atores envolvidos.

É possível identificar que as propostas centralizadas causam um desequilíbrio, com custo alto para leitores e as organizações que acessam seus repositórios. Em contrapartida, várias propostas surgiram, como, por exemplo, o EUREKA o qual seria uma aplicação de revisão de artigos ligada a plataforma de submissão *ScienceMatters* – Já o DEIP fornece apoio em pesquisas iniciais, através de angariação de recursos e da transformação de projetos de pesquisa em ativos digitais contáveis via *tokenização*. O OSN é uma rede pública (*ethereum*) de pesquisa distribuída que usa a tecnologia *blockchain* para uma transmissão transparente de informações de valor e um realinhamento significativo de incentivos.

Neste contexto, a DASP, além de gerir todo um processo editorial, é capaz de fornecer plataformas de divulgação de artigos acadêmicos, usando uma blockchain permissionada, ainda não explorada pelas novas soluções, relacionando mecanismos de taxação igualitária entre os entes envolvidos no processo e fomentando a relação entre a comunidade visando as variadas áreas que as constituem.

4. Editora Científica Autônoma e Distribuída (DASP)

Visando essas novas propostas de gestão, a DASP surge como uma alternativa para aperfeiçoar a governança no processo de publicação, promovendo a disrupção ao aplicar gerenciamento distríbuído e autônomo entre as entidades envolvidas em todo o processo. Para isso, a proposta utiliza como base o conceito das DAO. Os envolvidos mantêm os papéis de revisor e autor, porém tudo controlado pela DASP, sem intervenção humana, possibilitando o próprio autor tenha alguma recompensa com a publicação, seja financeiro ou através de algum outro bônus que poderá ser decidido entre a comunidade envolvida.

Ao instanciar a DASP, os revisores serão convidados, depois os próprios autores que tiverem artigos aceitos podem virar revisores. Os membros da comunidade podem dar notas para as revisões, assim aumenta ou diminui a probabilidade daquele revisor

receber outro artigo para revisar, funcionando como um método de retroalimentação na comunidade. Além de todos esses mecanismos, podemos avaliar a não revisão de um artigo aceito por um revisor, o que lhe acarretaria uma nota baixa na plataforma.

A DASP organiza as relações entre os atores da comunidade através de contratos inteligentes executados na rede. A partir deles os membros da comunidade podem assumir o papel de autor, leitor, revisor e revisor dos revisores. Tudo vai depender do quanto a comunidade acredita no individuo para desempenhar um papel específico. Sendo assim, a ideia central da DASP é aumentar a relevância da comunidade e não apenas criar um *software*, entendemos que o *software* é importante para prova de conceito neste artigo, mas estamos propondo um modelo de relação que prestigie a comunidade e não as editoras.

4.1. Arquitetura

A Figura 1 mostra a arquitetura de componentes da DASP usando o padrão de projeto *Model-View-Controller* (MVC) [de Lemos et al. 2013]. Na camada modelo estão presentes os módulos que irão se comunicar com a tecnologia de rede (*Hyperledger*) e armazenamento (IPFS) que serão utilizados. A camada de comunicação é composta pelos gerenciadores(usuário) usando notações de objetos em *JavaScript* e *HyperText Markup Language* (HTML). Na camada controle estão as partes relacionadas as funcionalidades responsáveis pela gerência da DASP, sendo ela dividida em quatro componentes que são apresentados a seguir:

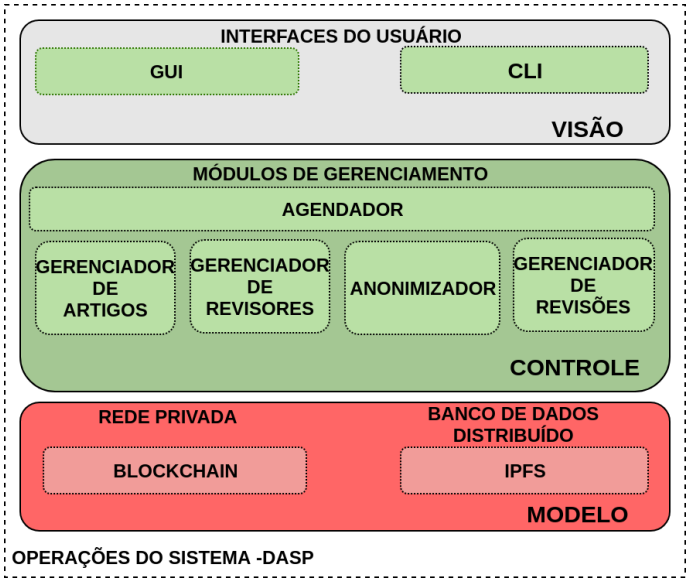


Figura 1. Arquitetura da DASP

- **Agendador**: Responsável pela definição de tempo e gerenciamento da comunicação entre os demais módulos.
- **Gerenciador de Artigos**: É responsável pela organização de submissão de arquivos e verificar se o documento está sendo enviado para o ente correto.
- **Gerenciador de Revisores**: É responsável pela escolha de revisores de forma randômica, mas considerando as condições de escolhas do revisor, como a quantidade de pontos ou *tokens* que o revisor possui e o número de revisões já realizadas. Sendo esta escolha realizada pela aplicação, sem qualquer intermediário.

- **Anonimizador**: Quando uma submissão é realizada, um conjunto de chaves temporárias é gerado para o autor. O sistema distribui a solicitação de revisão para os revisores, que compõem a comunidade, esses revisores geram através da própria *interface* do sistema um novo endereço *blockchain*, ou seja, uma chave pública e uma chave privada, e este novo endereço que é desconhecido para todos os entes, passa a ser uma outra carteira, só que temporária, que permanecerá enquanto o processo de revisão estiver ativo. Ao final do processo de revisão, a chave do revisor e do autor são associadas as chaves temporárias, anexando o processo de revisão ao histórico do revisor e a submissão é relacionada ao autor. Tal abordagem utiliza o conceito de comprometimento, no qual gera-se um valor secreto que posteriormente é utilizado para comprovar a autoria [Di Crescenzo et al. 2001].
- **Gerenciador de Revisão**: É a função responsável pela verificação da qualidade da revisão, que será determinada pela própria comunidade. Neste processo encontra-se a função lógica avaliativa em conjunto, entre o autor, a revisão em si e a comunidade, o método avaliativo é observado através de discussões do quanto o avaliador foi coerente em suas avaliações, e destina-se *tokens* ou pontos, que serão uma espécie de pagamento pela avaliação ter sido julgada por todo o conjunto citado anteriormente, como justa e altamente argumentativa.

A Figura 2 apresenta o diagrama de atividades da DASP, mostrando como funciona o processo de submissão e avaliação do artigo usando a aplicação, que para executar estas tarefas, fará uso de contratos inteligentes que serão assinados por cada um dos envolvidos, desta forma a DASP sabe exatamente o que cada ator fez no processo, permitindo a auditoria. Cada um dos atores terão uma chave pública e uma chave privada com a qual serão identificados no sistema, garantindo assim a privacidade e o não repúdio (não é possível dizer que não foi ele que fez determinada ação).

O processo é iniciado quando o autor submete o artigo que é instanciado no IPFS, onde é gerado um *Hash*, o arquivo então é enviado para a DASP, que está funcionando sobre uma rede privada usando a tecnologia *Hyperledger* Fabric[10] baseado no conceito de empresa de centro vazio (DAO), ou seja, não existe uma instituição controlando a aplicação, sendo a gestão realizada pelos entes da comunidade.

A partir desse momento, a DASP recebe o arquivo e realiza a escolha dos revisores, sendo que nessa etapa o revisor sorteado tem a opção de aceitar ou não a revisão do artigo. Na sequência, a DASP entra no período de revisão. Após a submissão das revisões pelos revisores, será gerada a resposta se a revisão foi aceita ou rejeitada. Em caso de aceite, o artigo é publicado e a revisão por escolha do autor é tornada pública ou não. Caso o artigo seja rejeitado, é enviado ao autor a revisão, o qual pode tornar pública ou não a revisão. Se esta for pública, a revisão estará passível de revisão pela comunidade.

5. Prova de conceito: DASP

A DASP surge à medida que movimentos voltados a ciência aberta se amadurecem e ganham visibilidade em comparação ao modelo clássico existente, que centraliza e restringe a relação entre os entes que participam da comunidade de pesquisa. Na contramão desse modelo, fazemos uso da *blockchain*, uma tecnologia que surgiu na indústria com objetivos de prover maior confiança entre os pares, excluindo intermediários, e além disso, propor

[10]https://www.hyperledger.org/projects/fabric

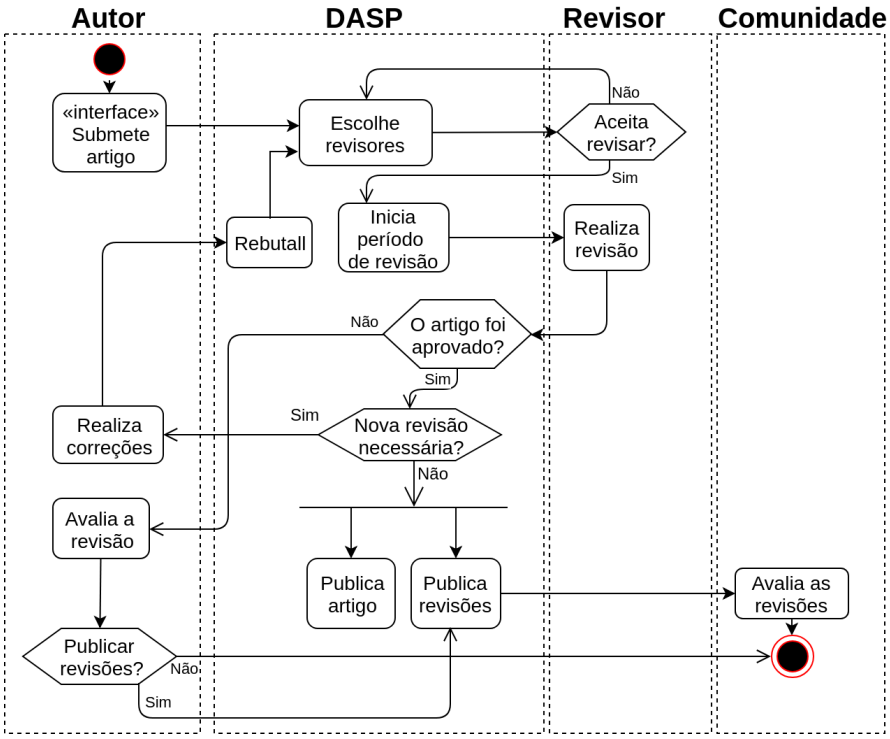


Figura 2. Diagrama de Atividades da DASP

formas de explorar e desenvolver aplicações que possam estabelecer modelos inovadores de relações na comunidade acadêmica.

A Figura 3 apresenta como os diferentes componentes da DASP se comunicam. Com intuito de fundamentalizar o uso do conceito da DAO para a solução, pois este provê um modelo de negócio autônomo e distribuído, no qual a comunidade se auto governa, sem nenhum ente regulador, gerente, ou governo controlando as ações que serão tomadas baseando-se em consenso de regras imutáveis, escritas na *blockchain*, na forma de contratos inteligentes.

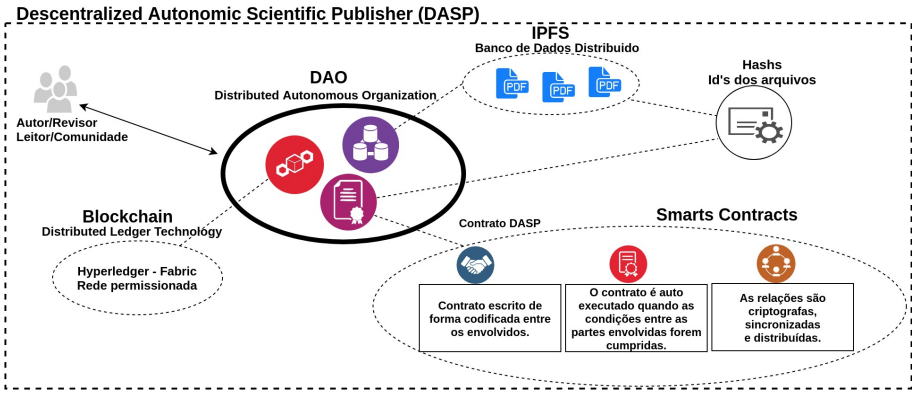


Figura 3. Componentes para Implementação da DASP

A plataforma *Hyperledger*, como demostrada na Figura 3, é utilizada como base para a criação da DASP. Neles são executados os passos de instruções de envio do arquivo, submissão, revisão do artigo, tempo de publicação e revisão, ou seja, todo o processo que

envolve os prazos (tempos) e interações entre os entes, junto com as especificações em relação a cada decisão da comunidade, do revisor ou do autor em específico.

Partindo das explicações de funcionamento do sistema, o IPFS tem como função a DASP o armazenamento dos arquivos submetidos, ou seja, quando o autor usa a aplicação (DASP) para enviar um arquivo, esse arquivo é inserido no IPFS, que automaticamente gera um *hash* que será o identificador daquele arquivo, é através desse identificador (ID) que o revisor irá localizar o arquivo que será visualizado e, posteriormente irá entrar em processo de revisão.

Como na aplicação existe a possibilidade de várias revisões no mesmo trabalho, seja pela escolha primária que é realizada de forma randômica dos revisores ou, através da revisão exposta à comunidade, quando é pedida pelo autor, funciona como uma espécie de segunda opinião, onde é gerado um novo ID para o arquivo, já que o mesmo passou por modificações.

A estrutura da aplicação está disponível publicamente no endereço (https://gitlab.com/gercom/dasp-hyperledger). Para testar a aplicação basta utilizar o ambiente de teste do *Hyperledger* chamado de **Hyperledger Composer Playground** que é um ambiente de prototipagem e teste. Através do endereço (https://composer-playground.mybluemix.net/login), efetue o *download* do GitLab o arquivo de nome ”dasp-net@0.0.1.bna”e carregar o arquivo no *playground*, conforme a Figura 4.

Após a implantação da rede de negócio da DASP no *playground*, é necessário conectar a aplicação DASP a *blockchain*, onde será possível visualizar os arquivos e efetuar os testes descritos nesta prova de conceito. A Figura 4, mostra a aplicação DASP pronta para efetuar os testes e criar autor e revisor atráves da DASP(Admin), é nesse momentos que os atores poderão interagir, baseado nas permissões definidas no contrato inteligente.

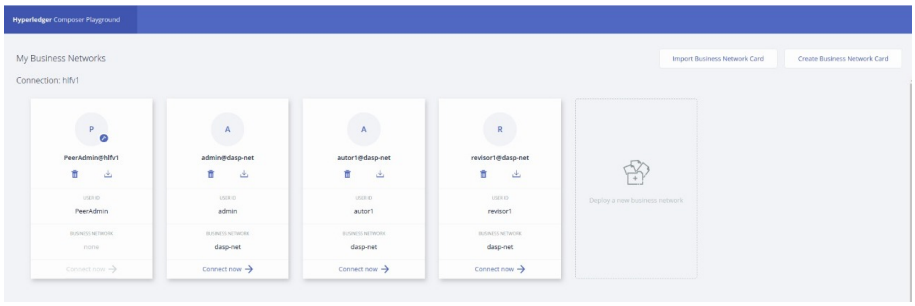


Figura 4. DASP conectada

Após a criação dos atores, é possível iniciar o processo de submissão do artigo por parte do autor, a Figura 5 mostra o registro de submissão dos autores, contituído de detalhes como o tipo de revisão, o *hash* de identificação do artigo, o ID do autor.

A Figura 6, mostra o registro dos artigos submetidos, como o tipo de revisão, se foi aceito ou não por algum revisor e os IDs de identificação de autor e revisor. Nesse processo, o acesso ao artigo é definido via contrato inteligente, então só os entes autorizados pelo contrato que seguem o fluxo editorial poderão ter acesso ao trabalho para efetuar a revisão, e posteriormente reenviar o resultado de avaliação do trabalho.

A Figura 7, mostra os registro de artigos aceitos pelos revisores, onde cada hash

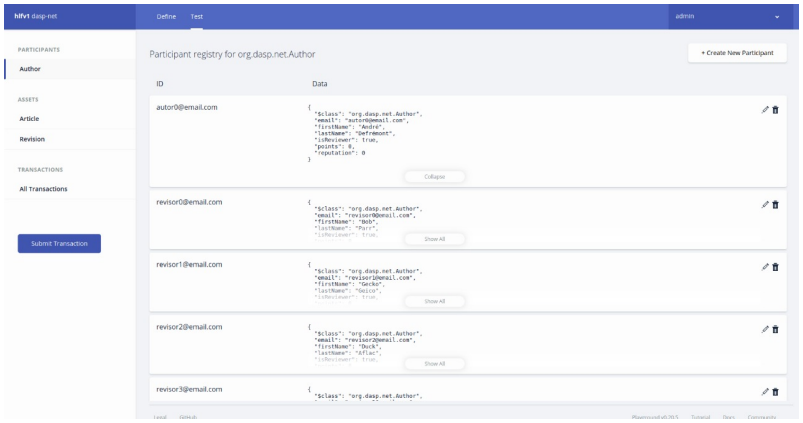


Figura 5. Registro de Submissão dos Autores

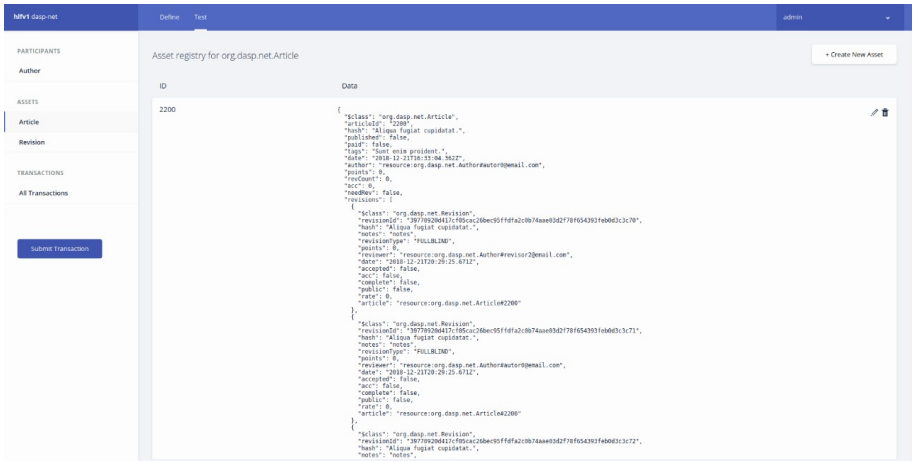


Figura 6. Registro de Artigos Submetidos

de transação e ID do autor fica vinculado ao *ID* do revisor, e as modificações no artigo ficam registradas na rede *blockchain*, além do registro das informações como o resultado da avaliação do artigo submetido e as políticas de visualização das revisões (ex: se serão públicas ou privadas a revisão do trabalho à comunidade).

As Figura 8 e 9 mostram as funcionalidades, que seriam os tipos de transações que são usados no processo, como:

- Bootstrap: Gera automaticamente atores e revisores para teste;
- CreateRevision: É respónsavel pela criação de uma nova revisão, é usada quando o autor realizar uma submissão de artigo;
- NewArticle: É usada em caso de uma nova submisão de artigo;
- NewAutor: Quando o autor deseja criar um *ID* na aplicação;
- NewHash: Ferramenta que gera um novo *hash*, quando o artigo sofrer algum tipo de modificação, esse mecanismo normalmente é usado quando revisor realiza a avaliação do atrabalho e reenvia o resultado do trabalho, pois toda vez que o trabalho passa por algum tipo de modificação o hash é modificado automaticamente, sem perder o vínculo entre as partes envolvidas;
- PublishRevision: Opção ativada quando a revisão é publicada ao autor;

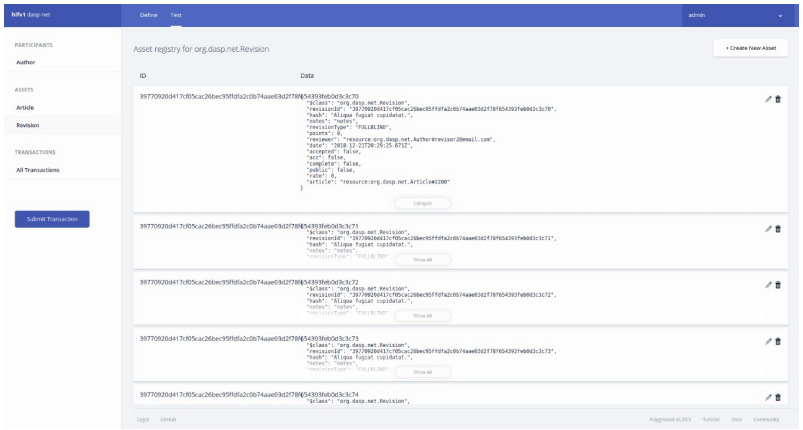
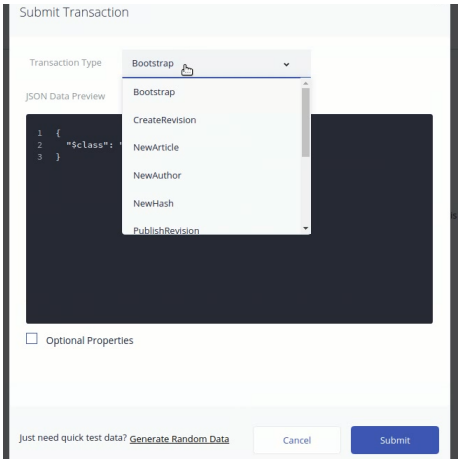


Figura 7. Registro de Trabalhos Aceitos pelos Revisores



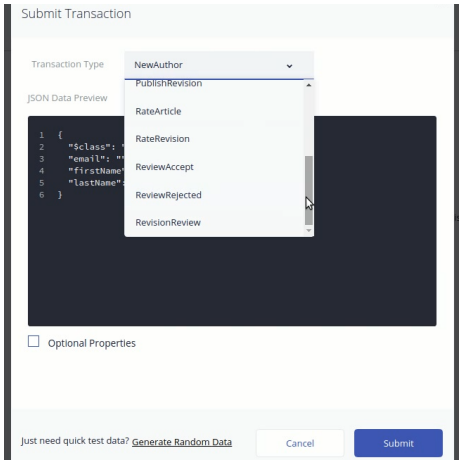


Figura 8. Funções da DASP

Figura 9. Funções do DASP

- RateRevision: É a opção onde se determina a avaliação, seguido por comentários acerca do trabalho;
- ReviewAccept: Determina se o artigo foi aceito, baseado na revisão;
- ReviewRejected: Determina se o artigo foi Recusado, baseado na revisão;
- RevisionReview: Aqui temos a opção de revisão secundária, ou revisão da revisão, onde o próprio DASP, baseado em termos de qualidade da revisão, determina se será necessário ou não uma nova revisão.

A Figura 10 mostra o registro geral de todas as transações realizadas no decorrer do processo, podemos a partir desse registro gerar auditorias e relatórios do que foi feito na aplicação de modo geral.

Em relação aos testes realizados, não realizamos testes de efetivos de avaliação e eficiencia, porém tivemos boas respostas da ferramenta, relacionados ao tempo de interação entre os entes e o uso da rede privada, agregando os mecanismo de segurança da mesma. Entendemos que o futuro das propostas citadas nos trabalhos relacionados será a migração de uma rede pública para as permissionadas, por motivos de segurança na interação da rede entre os entes e o acesso as informações.

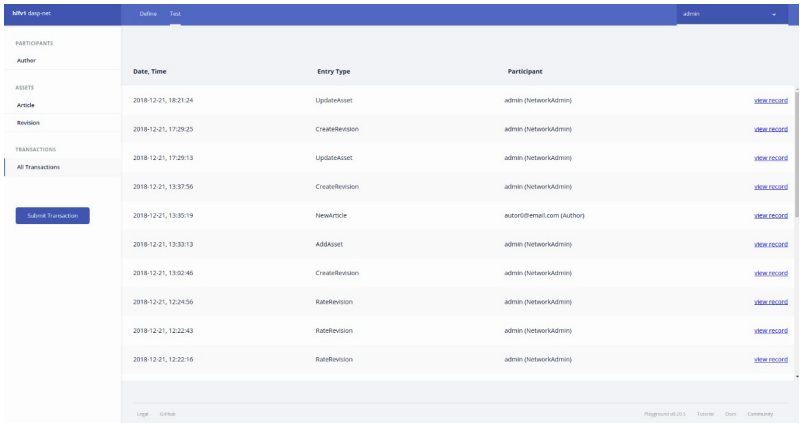


Figura 10. Registro das Transações Realizadas

6. Considerações Finais e Trabalhos Futuros

Os processos atuais de submissão, as entidades e nuances envolvidas na revisão por pares, acaba gerando um problema em relação a qualidade e o tempo das revisões, e acima de tudo, a ausência de controle, por parte dos autores, sobre os direitos autorais do seu próprio trabalho.

Nesse sentido a DASP, é proposta para autonomizar o ente editora, responsável pela gestão de todo o processo de revisão e publicação. A ideia por trás dessa alternativa é permitir maior autonomia e confiança entre as entidades que realmente fazem o processo existir, retirando o entidade intermediário (editora), promovendo uma comunicação *Peer-to-peer*(p2p), baseado em regras de consenso entre os membros da comunidade.

Desta forma, umas das principais mudanças seria como o autor, o revisor, o leitor e a comunidade passariam a se relacionar no processo, tendo em vista que todos teriam maior autonomia de opinião no processo, por exemplo, caso o autor se sinta desconfortável com a primeira revisão, ele poderá solicitar uma revisão secundária por parte de outros membros da comunidade. Só esse ponto já é uma iniciativa totalmente diferente comparada as atuais. Essa abordagem acaba dando maior incentivo aos revisores a terem mais atenção aos trabalhos, isso não seria apenas uma mudança de comportamento, mas também de respeito ao pesquisador e o grupo que ele estiver envolvido.

Como trabalhos futuros, iremos implementar um *front-end* para a aplicação juntamente com funcionalidades mais intuitivas, aperfeiçoar modelos de revisões de acordo com os mecanismos de consenso existentes e desenvolver novas soluções que facilitem o uso de instâncias por parte de usuários que desejem expandir a rede de interações com DASP.

Referências

Benet, J. (2014). Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*.

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

de Lemos, M. F., Oliveira, P. C., Ruela, L. C., da Silva Santos, M., Slveira, T. C., and de Sousa Reis, J. C. (2013). Aplicabilidade da arquitetura mvc em uma aplicação web (webapps). *RE3C-Revista Eletrônica Científica de Ciência da Computação*, 8(1).

DEIP, W. (2018). Decentralized research platform. Url: https://deip.world/. Ultimo Acesso: 2018-12-20.

Di Crescenzo, G., Katz, J., Ostrovsky, R., and Smith, A. (2001). Efficient and non-interactive non-malleable commitment. In Pfitzmann, B., editor, *Advances in Cryptology — EUROCRYPT 2001*, pages 40–59, Berlin, Heidelberg.

DuPont, Q. (2017). Experiments in algorithmic governance: A history and ethnography of "the dao," a failed decentralized autonomous organization. *Bitcoin and Beyond: Cryptocurrencies, Blockchains and Global Governance. Routledge*.

EurekaPlatform (2018). Eureka, a peer-to-peer scientific data publishing platform, powered by blockchain, to promote open science. https://eurekatoken.io/whitepaper_v3_1.pdf. Ultimo Acesso: 2018-12-21.

Ferraz, E. d. C. and Navas, A. L. G. P. (2016). *Publicação de artigos científicos: recomendações práticas para jovens pesquisadores*. Associação Brasileira de Editores Científicos.

Frantz, C. K. and Nowostawski, M. (2016). From institutions to code: Towards automated generation of smart contracts. In *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, pages 210–215.

Greve, F., Sampaio, L., ABIJAUDE, J., COUTINHO, A., VALCY, I., and Queiroz, S. (2018). Blockchain e a revolução do consenso sob demanda. *Livro de Minicursos do SBRC*, 1:1–52.

IEEE, F. (2013). ieee faqs. Url: http://www.ieee.org.br/ieee2013/wp-content/uploads/2013/01/revista_final.pdf/. Ultimo Acesso: 2018-03-26.

Job, I., Trindade, A., and Mattos, A. M. (2009). Processo de revisão pelos pares: Por que são rejeitados os manuscritos submetidos a um periódico científico? *Movimento (ESEFID/UFRGS)*, 15(3):35–55.

Mulligan, A., Hall, L., and Raphael, E. (2012). Peer review in a changing world: An international study measuring the attitudes of researchers. *Journal of the American Society for Information Science and Technology*, 64(1):132–161.

Partners, N. (2018). Open science network. Url: https://www.osn.global/. Ultimo Acesso: 2018-11-05.

SBC (2015). Sbc. Url: http://seer.ufrgs.br/index.php/reic/about/submissions#privacyStatement Ultimo Acesso: 2018-05-09.

Shurkin, Joel (2017). Acm history. Url:https://www.acm.org/about-acm/acm-history Ultimo Acesso: 2018-05-09.

Tschorsch, F. and Scheuermann, B. (2016). Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Comm. Surveys Tutorials*, 18(3):2084–2123.

Zheng, Z., Xie, S., Dai, H.-N., and Wang, H. (2016). Blockchain challenges and opportunities: A survey. *Work Pap*.

Controle de acesso a serviços usando Smart Contracts

Marcos Vinicius M. R. de Souza[1], **Hélio Crestana Guardia**[1]

[1]Depto. de Computação – Universidade Federal de São Carlos (DC-UFSCar)
São Carlos – SP – Brazil

marcos.souza@dc.ufscar.br, helio@dc.ufscar.br

Abstract. *Advances in ubiquitous computing and in the Internet of things (IoT) bring new challenges to access control systems, such as not having to rely on a centralized trusted party, and the ability to use contextual data. Such characteristics may be required to enable IoT services in smart homes and cities, and in electronic health systems. In these environments, service-based device to device interactions reinforce the need for flexible and scalable authenticity and confidentiality mechanisms. This work presents an access control model to address such challenges using a blockchain infrastructure. In the proposed model, services and contextual data are defined using smart contracts, and blockchain logs are used for reliable information management. Assets are associated with permissions, and their transfers prove access control rights in service invocation requests. A reference implementation on Hyperledger shows the viability of the model.*

Resumo. *Avanços na computação ubíqua e na Internet das Coisas (IoT) trazem novos desafios aos sistemas de controle de acesso, como a não dependência de unidades centrais seguras, e a necessidade de identificar dados de contexto. Estas características podem ser necessárias para serviços em IoT, como em casas e cidades inteligentes e em sistemas eletrônicos de saúde. Nesses ambientes, autenticidade e confidencialidade devem ser providas de forma flexível e escalável. Este trabalho apresenta um modelo de controle de acesso que atende estes desafios baseando-se na infraestrutura de blockchain. Nesse modelo, serviços e dados contextuais são definidos via smart contracts e registros na blockchain provêem gerenciamento seguro de informações. Assets são associados às permissões e suas transferências provam direitos de acesso a serviços. Uma implementação de referência do modelo sobre Hyperledger mostra sua viabilidade.*

1. Introdução

O acesso à informação comumente deve ser controlado, a fim de garantir privacidade e segurança. Diversos modelos podem ser adotados para definir o controle de acesso e suas políticas e regras, destacando-se o RBAC (*Role-Based Access Control*), baseado em papéis, e o ABAC (*Attribute-Based Access Control*), baseado em atributos.

RBAC [Ferraiolo e Kuhn 1992] baseia-se na relação entre papéis assumidos por um usuário e os recursos que ele pode acessar. Esse modelo se enquadra em situações em que as permissões de acesso são definidas pelo cargo ocupado pelo usuário. Já no ABAC [Hu et al. 2014], o acesso é concedido àqueles usuários que possuírem os atributos necessários para acessar cada recurso. Esse modelo propicia uma granularidade mais fina nas regras de acesso, ao custo de maior complexidade de gerenciamento das regras.

Apesar de o modelo ABAC permitir uma granularidade mais fina, sua complexidade de implementação e manutenção dificulta sua adoção. Assim, o modelo RBAC é mais adotado, incluindo extensões para atender necessidades adicionais de controles de acessos. A sensibilidade ao contexto (*context-awareness*), por exemplo, é considerada essencial na computação ubíqua [Kuhn, Coyne e Weil 2010] e extensões no modelo RBAC buscam viabilizar sua adoção.

Em sistemas computacionais, contexto é definido como qualquer informação que possa ser usada para caracterizar estado, situação, ou atividade de uma entidade, ou ainda do ambiente no qual a entidade opera [Dey 2001, Toninelli et al. 2006]. Informações de contexto podem incluir, entre outros aspectos, tempo, localização, condições gerais do ambiente, intenções do usuário, estado da rede e estado de execução. Os modelos de controle de acesso que utilizam tais informações de contexto são classificados como CAAC (*Context Aware Access Control*).

O desenvolvimento das tecnologias mobile, da internet das coisas e da computação ubíqua de forma geral, gerou um aumento na quantidade de formas e locais possíveis para acessar informações, consequentemente, aumentando a variedade de situações nas quais as informações serão acessadas. No entanto, uma informação não estará segura se não for visualizada apenas pela pessoa certa, no momento certo e nas circunstâncias certas.

A preocupação em tratar variáveis de contexto no controle de acesso, trouxe várias propostas sobre o assunto. Exemplos disso são trabalhos como [Cleeff, Pieters e Wieringa 2010], com um estudo sobre LBAC (*Location-Based Access Control*), e [Bertino e Kirkpatrick 2011] com os GEO-RBAC (*Geo-spatial RBAC*). Ambos trabalhos mostram que modelos de controle de acesso com uso de localização têm sido usados para melhorar a segurança em TI. Ao estabelecer localizações específicas para atuação de sistemas e usuários, atacantes terão mais dificuldade em comprometer um sistema.

[Cleeff, Pieters e Wieringa 2010] ainda apresentam metas para os LBAC, sendo: o princípio da separação de deveres (*principle of separation of duty*), princípio do privilégio mínimo (*principle of least privilege*), capacidade de responsabilização (*accountability*), capacidade de manutenção (*maintainability*) e usabilidade (*usability*). No entanto, tais metas podem ser visadas por outros modelos de controle de acesso.

O avanço das tecnologias com dispositivos móveis ainda trouxe mais preocupações, como a discutida por [Gupta, Kirkpatrick e Bertino 2012] que ressaltam que a localização relativa de um usuário pode ser mais importante do que sua localização absoluta, pois a posição do usuário em relação à outras pessoas é uma potencial ameaça a segurança das informações.

Com a mesma preocupação com localização relativa, [Choi et al. 2013] propõem um modelo de controle de acesso baseado em proximidade. Chaves e assinaturas parciais são propagadas por dispositivos Bluetooth e o leitor precisa de todas essas chaves parciais e da composição das assinaturas para obter acesso. Para isto, ele deve estar na área determinada como confiável, que é delimitada pelos dispositivos.

Novas abordagens para controle de acesso também têm incluído o uso de blockchain, como visto em [Zyskind, Nathan e Pentland 2015], que apresenta uma forma de administrar o controle de acesso sem a necessidade de confiança em terceiros. Para tanto, implementam um serviço que armazena de maneira distribuída dados assinados e, even-

tualmente, cifrados. MedRec [Azaria et al. 2016] e FairAccess [Ouaddah, Elkalam e Ouahman 2016] são exemplos de frameworks desenvolvidos para modelos de controle de acesso baseados em blockchain.

Essas abordagens visam a aproveitar princípios e práticas já testados no ambiente das criptomoedas (*crypto currencies*), trazendo para o controle de acesso a possibilidade de uma arquitetura de sistema descentralizada, distribuída, auditável, transparente e robusta, incrementando a segurança e privacidade. Essas características tornam o uso das tecnologias de blockchain promissor nos ambientes de Internet das Coisas (*IoT - Internet of Things*), com aplicação em serviços de cuidados de saúde e *Smart Cities* [Hashemi et al. 2016], por exemplo.

Esse trabalho apresenta uma nova proposta de uso da tecnologia blockchain para prover controle de acesso na interação entre dispositivos usando invocações de serviço. De maneira geral, a abordagem proposta procura agregar características da rede como rastreabilidade, eliminação do terceiro confiável e imutabilidade. Na seção 2 apresentam-se considerações sobre blockchains e alguns sistemas que utilizam essa infraestrutura para controle de acesso. A seção 3 apresenta um novo modelo de controle de acesso a serviços baseado em blockchains. Já na seção 4 são descritos aspectos da implementação do modelo proposto e, na seção 5 são discutidos resultados da aplicação do modelo. Por fim, na seção 6 são apresentadas as conclusões sobre a proposta e alguns aspectos de trabalhos futuros.

2. Controle de acesso e blockchain

A estrutura de blockchain foi primeiramente apresentada como um registro público e descentralizado para transações financeiras com o Bitcoin [Nakamoto 2008]. A blockchain apresentada com o Bitcoin utiliza um conceito de transações focadas na transferência de valores entre contas. Outras plataformas blockchain expandiram o conceito de transações de acordo com suas aplicações. Ethereum [Wood 2014], por exemplo, apresentou uma plataforma blockchain mais generalizada, expandindo as transações para contratos complexos que passaram a ser chamados de *smart contracts*. Dessa forma, transações passam a ter descrições algorítmicas (*smart contract*), representando operações computacionais que serão feitas no blockchain [Wood 2014, English, Auer e Domingue 2016]. *Smart contracts* passaram a ser implementados em outras plataformas blockchain, como no Hyperledger Fabric do projeto Hyperledger, onde são chamados de *chaincodes* [Cachin et al. 2016].

Ao criar uma arquitetura descentralizada, distribuída, auditável, transparente e robusta, com potencial para prover elevada segurança e privacidade, a estrutura de blockchain passou a atrair interesses em outras áreas de aplicação, como no gerenciamento e controle de acesso de informações.

[Zyskind, Nathan e Pentland 2015], implementam um protocolo que transforma um blockchain em um gerenciador de controle de acesso automatizado que não requer confiança em terceiros. Diferente do Bitcoin, as transações no sistema não são financeiras, mas são usadas para carregar instruções, como armazenar, consultar e compartilhar dados. Os autores discutem ainda possíveis futuras extensões ao blockchain para aproveitá-lo como solução para problemas de confiança em computação na sociedade.

[Azaria et al. 2016] apresentam MedRec, um sistema de gerenciamento de regis-

tros descentralizado para manusear Registros Médicos Eletrônicos (RMEs), usando tecnologia blockchain. A estratégia aproveita propriedades únicas de blockchain, fazendo MedRec gerenciar autenticação, confidencialidade, capacidade de responsabilização e compartilhamento de dados. MedRec ainda possui uma estrutura modular que o integra com provedores existentes e com soluções de armazenamento local de dados.

[Azaria et al. 2016] ainda incentivam as partes médicas interessadas a participar da rede como "mineradores" de blockchain, e que obtenham em troca acesso aos dados, agregados e anônimos, como recompensa pela mineração, por sustentar e assegurar a rede por prova de trabalho (*Proof of Work*). Incentivam também que pacientes e provedores liberarem esses metadados anonimamente.

[Ouaddah, Elkalam e Ouahman 2016] fornecem um modelo de referência para um framework proposto para sistemas *IoT*, com Objetivos, Modelos, Arquitetura e Mecanismo (camadas OM-AM). Assim, apresentam o FairAccess, um framework de gerenciamento de autorizações descentralizado, com pseudônimos e preservação da privacidade, que permite aos usuários possuírem e controlarem seus dados. Então, adaptam o blockchain como um gerenciador de controle de acesso descentralizado. Diferente das transações financeiras do bitcoin, o FairAccess apresenta novos tipos de transações que são usadas para conceder, receber, delegar e revogar acesso.

Considerando as características de uma infraestrutura de blockchain, contudo, novos mecanismos de segurança para controle de acesso podem ser providos sobre essa plataforma, potencialmente combinando consenso e consistência dos dados, dinamismo de geração de informações para tratamento de aspectos de localização e flexibilidade no gerenciamento de múltiplos recursos lógicos.

3. Modelos para controle de acesso baseado em blockchain

No cenário de computação distribuída considerado neste trabalho, investiga-se a comunicação via solicitações de serviço entre dispositivos que queiram compartilhar recursos perante certas condições e outros que queiram ter acesso a estes recursos e que, para tal, precisam satisfazer as condições estabelecidas.

Tendo em vista as flexibilidades do modelo de controle de acesso baseado em papéis e usando dados de contexto em cenários de *IoT*, ou em qualquer situação de interação direta entre dispositivos, bem como o modelo de autenticação distribuída provido pelos sistemas de blockchain, esse trabalho propõe a criação de um sistema de autenticação que integre essas plataformas.

Neste cenário, adotou-se o uso de uma plataforma blockchain por sua capacidade de estabelecer uma relação de confiança distribuída, entre nós que não possuem confiança pré-estabelecida entre si, nem com uma terceira parte centralizada. Considerando sistemas como o Hyperledger e o Ethereum, uma infraestrutura de blockchain também permite registrar serviços e operações que se deseja realizar de forma verificável, e que podem ser criados, armazenados e acessados de maneira distribuída.

Para tratar dados de contexto e a autenticação de forma distribuída, propõe-se o uso de smart contracts para a especificação de códigos dos serviços de forma confiável entre múltiplos pares, sem o uso de uma entidade central confiável. As operações previamente indicadas nos smart contracts e solicitadas pelos participantes são armazenadas no

blockchain e podem ser verificadas. As verificações sobre as operações para registro e alteração do estado atual da rede são feitas por meio de consenso entre os nós participantes.

Blockchain é utilizado para o registro de recursos, de políticas de acesso, de dados de contextos e de transações, bem como para obtenção de acesso, atuando também como registro (*logging*) auditável das operações. Para o desenvolvimento de modelos de controle de acesso neste trabalho foi utilizada a plataforma blockchain Hyperledger Fabric, pela sua versatilidade na construção da rede blockchain e na execução de transações, sua não dependência de criptomoedas e seu suporte para criação e execução consensual de *smart contracts*.

3.1. Hyperledger Fabric

Hyperledger Fabric é um dos projetos Hyperledger com desenvolvimento hospedado pela *The Linux Foundation*. Trata-se de uma implementação de estrutura blockchain com a finalidade de atuar como base para aplicações, com funcionalidades como consenso na execução de códigos e serviços de associação entre membros, aos moldes de tecnologia *plug-and-play* (conectar e usar).

No Hyperledger Fabric, os *smart contracts* são chamados de *chaincodes* e são utilizados para compor toda a lógica da aplicação do sistema e suas transações. Suas especificações podem ser implementadas como códigos executáveis com funções para manipular os dados de entrada, o estado e os registros na rede.

As transações são as operações sobre a rede. No Hyperledger, toda transação tem uma entrada com dados novos e/ou dados já registrados, e resulta na criação, leitura, ou atualização de informações para registrar na blockchain, ou ainda apagá-las. As transações são definidas como funções contidas em *chaincodes*. A real aplicação das transações só ocorre caso haja consenso entre os *peers* de endosso, definidos na criação do *chaincode*, quanto ao resultado da execução.

A rede é composta por nós lógicos nos papéis de **cliente**, **endorsing peer** e **ordering peer**. Nós atuando como clientes são responsáveis por criar propostas de execução de transações, aguardar assinaturas de endosso e propagar ao *ordering service* as transações assinadas. *Endorsing peers* ficam em execução permanente, recebendo propostas de transação e simulando a execução dessas propostas, assinando propostas endossadas, realizando *commits* e mantendo seus *PeerLedgers*. *Ordering peers* são responsáveis pelo *ordering service*; eles garantem a ordem cronológica das transações, coletam propostas assinadas, verificam o endosso e redistribuem as transações válidas, ordenando *commits*.

Com o sistema Hyperledger, trabalha-se com os conceitos de *Participant* e *Asset*. *Participants* são os atores que interagem com a rede, usados para definir as classes dos participantes na comunicação. *Assets* são os objetos utilizados na interação dos *participants* com as transações, que podem ser criados, lidos, atualizados ou deletados. Neste trabalho, utilizam-se os termos **requerente** (*Requester*) e **proprietário de recurso** (*ResourceOwner*) para identificar os *Participants* do sistema de acesso.

Usando a infraestrutura do sistema Hyperledger, este trabalho apresenta duas estratégias para o gerenciamento de permissões, uma baseada em *smart contracts* e outra que utiliza *assets* como indicadores de permissões. Em ambos os casos, as permissões são verificadas em invocações de serviço (*smart contracts / chaincodes*).

3.2. CAAC usando smart contracts

Na primeira proposta de controle de acesso, *assets* são definidos para representar os recursos (*Resource*), as políticas de acesso (*AccessPolicy*), dados de contexto (*Context*) e permissões de acessos (*Access*).

O modelo aqui proposto trata interação entre nós numa rede via invocação de serviço. Os usuários de serviços são os usuários da rede blockchain, enquanto os prestadores de serviços são os *peers* que executam as operações (*chaincodes*). As transações definem os requisitos necessários e as regras de execução do serviço (também definidos nos *chaincodes*). Em caso comum neste modelo, um usuário de serviço solicitando acesso deve prover seus parâmetros e informações (dados de contexto) ao fazer a solicitação. Caso os parâmetros do requerente atendam os requisitos definidos (políticas de acesso) o serviço é executado e retorna os dados apropriados.

Três transações foram definidas como base para implementar esse modelo de controle de acesso: *AddResource*, *ComposeContext* e *RequestAccess*.

AddResource é a transação pela qual um Usuário (*ResourceOwner*) registra um recurso ao qual pretende conceder acesso perante determinadas condições. As entradas são as informações do recurso e as definições da política de acesso, gerando registro de um objeto *Resource* com uma política de acesso (*AcessPolicy*) vinculada.

ComposeContext é a transação que um requerente deve fazer para registrar suas atuais variáveis de contexto. Esta transação tem as informações do requerente como entrada e gera um objeto do tipo *Context* que é tanto usado nas requisições de acesso, quanto permanece registrado para futuras verificações e auditorias.

RequestAccess é a transação em que um requerente solicita o acesso a um determinado recurso. O requerente provê o *Asset* contendo seus dados de contexto validados e o recurso que pretende acessar como entrada para a transação. As verificações se o requerente satisfaz as regras de acesso são definidas nas operações dos *smart contracts* (*chaincodes*). O código da transação verifica se o contexto atual do requerente satisfaz as condições estabelecidas na política de acesso do recurso. Se as condições não são satisfeitas, nega o acesso. Caso as condições forem satisfeitas, a transação emite uma permissão de acesso para o requerente e cria um objeto *Access* para manter o registro daquela ação. O acesso ocorre pela verificação da existência desse objeto *Access* e o requerente tem permissão (*READ* e *UPDATE*) até que o objeto seja removido.

3.3. CAAC com smart contracts baseado em transferências de recursos

Na segunda proposta de uso de blockchain para controle de acesso, assets são associados às permissões para execução de serviços registrados na rede pelos nós / dispositivos servidores.

No Hyperledger Fabric, existe um tipo de transação que são as *deploy transactions*, que permitem a criação dinâmica de novos *smart contracts* (*chaincodes*). Isso possibilita a criação de um sistema de controle de acesso dinâmico, já que novas lógicas de operação podem ser incluídas por esses *chaincodes*.

Um modelo de controle de acesso que varie com o tempo, traz a necessidade de adaptação no tratamento das informações de contexto, já que novas informações de

contexto podem ser modeladas e pode haver novas lógicas de validação. Assim, propõe-se que a posse de recursos (*Assets*) equivalha à satisfação dos requisitos de permissão necessários para solicitar a execução de um serviço.

Para isso, tanto é possível que um *Asset* único, represente todas informações de contexto de um cliente, quanto utilizar múltiplos *Assets* para representar um contexto, cada um referindo-se à prova de uma informação de contexto diferente. Ao representar diferentes informações de contexto com *Assets* independentes, torna-se possível a introdução de modelagens de novas variáveis de contexto já durante a operação da rede.

Neste modelo, ao invés de um *Requester* possuir um *Asset* do tipo *Context*, obrigatório para provar seu contexto, esse *Requester* passa a ser proprietário (*owner*) de vários *Assets*, cada um comprovando uma informação de contexto diferente. A associação de permissões com *Assets* deve ser realizada pelos proprietários de recursos compartilhados na forma de serviço. Diferentes (infinitos) *assets* podem ser criados e gerenciados por uma infraestrutura de blockchain, assim como há mecanismos para transferência desses *assets* de forma válida e auditável entre os participantes.

Nesse sentido, apresenta-se a ideia de um sistema com chaincodes adicionados em tempo de execução e *Assets* utilizados como "moeda de troca" a ser gasta diretamente na obtenção de uma informação (ou invocação de qualquer serviço / *chaincode*).

Neste cenário, um usuário recebe *Asset(s)* que prova(m) que ele está em um contexto, ou satisfaz requisito(s) de contexto e, portanto, tem as permissões necessárias para a execução do serviço. Diferentes mecanismos, inclusive externos, podem ser utilizados para a atribuição desses *Assets* de forma confiável. Assim, a transação de requisição requer que a posse desse(s) *Asset(s)* tenha sido transferida para o provedor do serviço desejado e que o registro dessa transferência na blockchain seja fornecido como parâmetro da invocação do serviço. Essa transferência indica que o cliente do serviço satisfaz as condições necessárias, ou seja, tem as permissões requeridas. Esta alternativa visa evitar buscas constantes por permissões salvas e o reúso indevido destas permissões.

Nesta abordagem, o proprietário de um recurso implementa um *chaincode* especificando como as informações do seu recurso serão acessadas, como o contexto do requerente será avaliado para concessão do *Asset* necessário para o acesso, e como esse *Asset* será gasto na requisição do acesso. Todos esses requisitos, validáveis pelos mecanismos de transferências de *Assets*, passam a ser embutidos numa avaliação lógica executada na parte inicial do código de cada *chaincode* de serviço.

Resumidamente, antes de executar a lógica associada a um serviço, é preciso que o código de um chaincode inclua a verificação da transferência dos *Assets* necessários, do solicitante para o dono do serviço.

Os identificadores das transferências realizadas, e devidamente salvas nas cadeias de informação do Hyperledger, são passados pelo cliente na solicitação de execução do serviço desejado. Retomando as operações do servidor, vê-se que ele será capaz de validar que os *Assets* foram transferidos e pode prestar o serviço.

4. Implementação

Na operação dos dois modelos de controle de acesso propostos neste trabalho, o seguinte cenário é previsto:

- Há um conjunto de nós interligados em rede, disponíveis para execução das atividades de manutenção da blockchain (neste caso, do Hyperledger): dois ou mais *endorsing peers* para execução do serviço e ao menos um *ordering server*;
- Um nó que deseja participar da rede provendo serviço deve registrar-se na rede, criando uma identidade e registrando seus serviços em algum "*channel*" do Hyperledger;
- Um cliente que deseja usar qualquer serviço provido na rede também deve criar uma identidade e conectar-se a um "*channel*" do Hyperledger.

Para o modelo de CAAC usando *smart contracts*, as operações de clientes e servidores são as seguintes:

- Ao cadastrar um recurso, um servidor deve cadastrar uma política de acesso vinculada, que deverá ser cumprida para geração da permissão.
- Um cliente que deseja executar um serviço deve cadastrar seu contexto e solicitar a permissão. Isso é feito através do serviço *RequestAccess*, que verifica se o contexto do cliente satisfaz a política de acesso do serviço, e gera um registro de permissão. Esse registro será verificado para a realização do acesso.

De maneira geral, o fluxo de controle de acesso a um serviço é definido de acordo com a Figura 1.

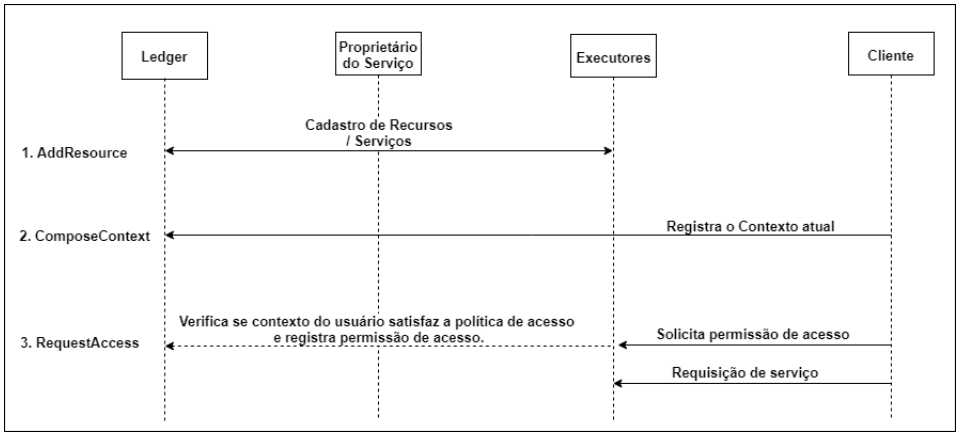


Figura 1. Fluxo de operações no CAAC com *smart contracts*.

Para o modelo CAAC com *smart contracts* baseado em transferências de recursos, as operações de clientes e servidores são as seguintes:

- Na fase inicial de qualquer serviço provido deve existir uma verificação de requisitos que correspondem a transferência(s) de *Asset(s)*.
- Um cliente que deseja um serviço deve satisfazer os critérios que levam ao recebimentos dos Assets necessários. Assim, solicitar permissão significa obter os *Assets* apropriados.
- Para solicitar um serviço, um cliente transfere os *Assets* associados aos privilégios necessários para o dono do serviço ou entidade responsável.

De maneira geral, o fluxo de controle de acesso a um serviço é definido de acordo com a Figura 2.

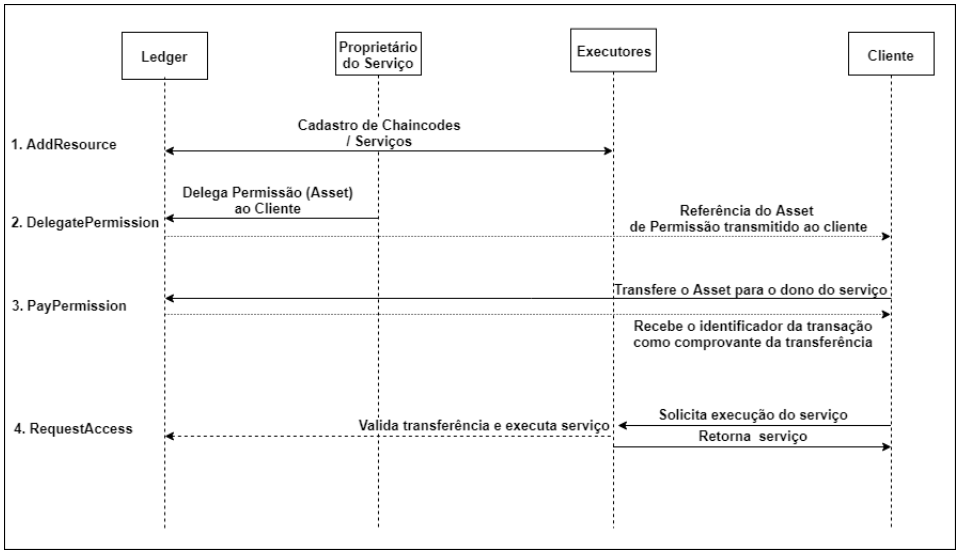


Figura 2. Fluxo de operações no CAAC com *smart contracts* baseado em transferências de recursos.

Vale observar que os códigos executados pelo Hyperledger são todos implementados como *chaincodes*. Assim, há os *chaincodes* de serviço, associados às operações de transferências e criações de *Assets*, por exemplo, disponíveis na biblioteca / API do Hyperledger, ou criados pela aplicação que vai usar essa infraestrutura. Usando os serviços do Hyperledger também é possível criar dinamicamente novos chaincodes, que serão os serviços providos por nós participantes do sistema e cujas execuções serão solicitadas pelos clientes que satisfizerem os requisitos de permissão.

Para demonstrar a implementação dos *chaincodes* que viabilizam o modelo de controle de acesso proposto, foram criadas funções em *JavaScript* utilizando a ferramenta Hyperledger Composer, que determinam o funcionamento das etapas de comunicação.

Para viabilizar os modos de controle de acesso previstos, serviços específicos (*chaincodes*) também foram desenvolvidos na plataforma Hyperledger.

No modelo CAAC com *smart contracts*, os *chaincodes* são:

- **addResource**: cadastra um novo recurso e uma política de acesso associada a ele, como no Algoritmo 1.
- **composeContext**: cadastra as informações de contexto do usuário, como no Algoritmo 2.
- **requestAccess**: verifica se o contexto do usuário satisfaz a política de acesso do recurso e gera o registro da permissão, como no Algoritmo 3.

Para o modelo CAAC com smart contracts baseado em transferências de recursos foram criados os seguintes chaincodes:

- **addResource**: cadastra um novo serviço, como no Algoritmo 4.
- **delegatePermission**: gera *Asset* de permissão para um cliente, como no Algoritmo 5.
- **payPermission**: cliente transfere *Asset* de permissão para o proprietário ou gestor do serviço, como no Algoritmo 6.
- **requestAccess**: verifica se as permissões foram corretamente transferidas e invoca serviço, como no Algoritmo 7.

Algoritmo 1: AddResource

Entrada: $resourceId, address, accessPolicy, owner$
Saída: registro de $Resource$ e atualização de $ResourceOwner$

```
1 início
2     Cria Resource com resourceId ;
3     Resource.addrees = address;
4     Resource.accessPolicy = accessPolicy;
5     Resource.owner = owner;
6     owner.resources.push(Resource);
7 fim
8 retorna add(Resource) e update(owner)
```

Algoritmo 2: ComposeContext

Entrada: $contextId, role, location, proximity, ..., owner$
Saída: registro de $Context$ e atualização de $Requester$

```
1 início
2     Cria Context com contextId ;
3     Context.role = role;
4     Context.location = location;
5     Context.proximity = proximity;
6     ...                                   // Quaisquer variáveis de contexto definidas
7     Context.owner = owner;
8 fim
9 retorna add(Context) e update(owner)
```

Algoritmo 3: RequestAccess

Entrada: $Resource, accessId, Requester$
Saída: registro de $Access$ e atualização de $Requester$

```
1 início
2     if Requester.Context satisfaz Resource.accessPolicy then
3         Cria Access com accessId ;
4         Access.Resource = Resource;
5         Access.permissionedRequester = Requester;
6         owner.accesses.push(Access);
7 fim
8 retorna add(Access) e update(owner)
```

Algoritmo 4: AddResource

Entrada: $resourceId, address, owner$
Saída: registro de $Resource$ e atualização de $ResourceOwner$

```
1 início
2     Cria Resource com resourceId ;
3     Resource.addrees = address;
4     Resource.owner = owner;
5     owner.resources.push(Resource);
6 fim
7 retorna add(Resource) e update(owner)
```

Algoritmo 5: DelegatePermission

Entrada: $permissionId, resourceId, requesterId$
Saída: registro de $Permission$

```
1 início
2     Cria Permission com permissionId ;
3     Permission.resourceId = resourceId;
4     Permission.permissionedRequester = requesterId;
5     Permission.owner = requesterId;
6 fim
7 retorna add(Permission)
```

Algoritmo 6: PayPermission

Entrada: $permissionId, provider$
Saída: atualização de $Permission$ e $Requester$
1 **início**
2 $Permission.owner = provider$;
3 $Permission.permissionedRequester.recibo = transactionId$;
4 **fim**
5 **retorna** $update(Permission)$ *e* $update(Permission.permissionedRequester)$

Algoritmo 7: RequestAccess

Entrada: $resourceId, Requester$
Saída: JSON de resposta do serviço
1 **início**
2 $tranx$ = busca('selectPayment', $recibo : Requester.recibo, resourceId : resourceId, userId : Requester.userId$);
3 **if** *tranx* **then**
4 Executa serviço solicitado $Resource.resourceId$;
5 Gera $response$ com dados de resposta do serviço;
6 Formata $response$ como JSON;
7 **fim**
8 **retorna** $response$

Na operação dos chaincodes de serviço criados, diferentes serviços do Hyperledger foram utilizados:

- getFactory().newResource(parâmetro): cria um *Asset*.
- getFactory().newConcept(parâmetro): cria *Concept* (Estrutura conceitual que pode ser definida no modelo. Por exemplo, uma estrutura para JSON de retorno de serviço).
- getAssetRegistry(parâmetro): busca registro de Assets no *ledger*.
- getParticipantRegistry(parâmetro): busca registro de Participants no *ledger*.
- add(parâmetro): registra informação no *ledger*.
- update(parâmetro): atualiza valor de registro no *ledger*.
- query(parâmetro): usado para implementação de buscas específicas no *ledger*. Esse tipo de mecanismo foi particularmente utilizado na verificação da transferência de *Asset*, através do identificador da transação de transferência.

5. Resultados

As propostas de controle de acesso a serviços apresentadas neste trabalho são baseadas no mecanismo de *smart contracts*, na transferência de recursos e em aspectos de autenticação providos por uma infraestrutura de blockchain. Sobre essa plataforma, implementada sobre o sistema Hyperledger Fabric, provê-se controle de acesso sensível ao contexto em que se encontram os nós comunicantes, como pode ser necessário em cenários de *IoT*, por exemplo.

No primeiro modelo apresentado, buscou-se a criação de um modelo CAAC seguindo a mesma linha de desenvolvimento apresentada na bibliografia com os trabalhos relacionados. Portanto, utiliza-se uma estrutura baseada no modelo RBAC e adiciona-se novas restrições sobre as políticas e permissões de acesso de acordo com variáveis de contexto.

Este modelo foi apresentado utilizando-se um *Asset* que representa o recurso a ser acessado, com uma política de acesso que dita as variáveis de contexto para o acesso.

Outro *Asset* foi utilizado para representar o papel (*role*) do usuário e suas informações de contexto. Assim, utilizou-se um *smart contract* (*chaincode*) para verificar se o papel e as informações de contexto do usuário satisfazem as condições da política de acesso do recurso.

No segundo modelo apresentado, idealizou-se um mecanismo baseado em interação por serviços com controle de permissões de forma distribuída e segura. Tipicamente, num sistema distribuído utiliza-se: *broker*, para fazer os registros e resolução de endereços; provedor, para registrar e oferecer os serviços; e cliente, que realiza a descoberta dos serviços e provedores, via *broker*, e invoca os serviços diretamente aos provedores.

Levando isto em consideração, no modelo apresentado o *brokering* é feito a partir do *ledger* com os registros distribuídos; o provedor é representado pelos peers que cadastram *chaincodes* e serviços e que gerenciam as permissões, além dos *endorsing peers*; já os clientes são os *peers* que interagem com o *ledger* para validações e consultas, e interagem com os mecanismos de execução (*endorsing* e *ordering*).

A infraestrutura utilizada conta com o uso do Hyperledger e do conjunto de serviços que viabilizam o modelo de permissões elaborado. Assim, utiliza *Assets* como moeda de troca para obtenção de acesso. Um usuário recebe um *Asset* de permissão caso satisfaça as políticas de acesso e, posteriormente o utiliza, transferindo-o para o provedor, em troca do acesso.

Dessa forma, o modelo possui os seguintes benefícios:

- *brokering* distribuído no *ledger*;
- mecanismo de consenso (*endorsement*);
- mecanismo de permissões flexível, com quantos requisitos forem desejáveis por serviço, através dos *smart contracts* (*chaincodes*);
- permite a transferência de permissões como moeda de troca;
- evita reúso de permissões;
- pode usar validações externas e circunstanciais para dados de contexto;
- e garantia de autenticidade e consistência através das assinaturas e chaves cripto-assimétricas.

6. Conclusões

A avaliação das propostas de controle de acesso com informação de contexto (CAAC) usando a plataforma Hyperledger mostrou a flexibilidade e simplicidade dos modelos. Utilizando o CAAC com *smart contracts*, a permissão de acesso é gerada a partir da verificação de um *Asset* de contexto em relação a uma política de acesso vinculada ao recurso. A permissão é gerada na forma de um registro na rede blockchain, que precisa ser verificado para obter o acesso.

Já no CAAC com *smart contracts* baseado em transferências de recursos, cada requisito para permissão é associado a um *Asset* a ser transferido ao dono do serviço como comprovação dos requisitos necessários. Informações de contexto, que podem ser validadas por elementos externos ao sistema, podem estar associadas a atribuição dos *Assets* necessários, inclusive de maneira dinâmica, como em situações que requerem presença e proximidade.

As execuções com o Hyperledger mostraram que a verificação dos requisitos de acesso podem estar embutidas na própria lógica dos serviços disponibilizados na rede.

Considerando que os mecanismos de verificação de autenticidade e garantia de confidencialidade do Hyperledger estão de acordo com a lógica definida nos sistemas de blockchain, as transações previstas nessa proposta atendem os requisitos de segurança envolvidos com as comunicações. Providas as autenticidades das informações transmitidas, as propostas de controle de acesso desenvolvidas podem ser facilmente implementadas e embutidas nas lógicas de quaisquer serviços que queiram utilizá-las. Para isso basta especificar requisitos como *Assets* e suas transferências como validações de seus cumprimentos.

Os mecanismos de consenso e tolerância a falhas provido pelo Hyperledger na execução dos serviços e atualização do *ledger* provêm uma segurança adicional ao modelo vislumbrado.

A proposta do modelo de segurança apresentado atende também as metas de segurança apresentados por [Cleeff, Pieters e Wieringa 2010]:

- há uma clara separação de deveres (*SoD*), provendo-se o gerenciamento das permissões aos serviços diretamente aos seus provedores;
- ao definir os *assets* necessários como permissões, usadas exclusivamente para o acesso a um serviço, atende-se o requisito de mínimo privilégio;
- o uso dos registros no livro razão da blockchain permite a responsabilização dos atores de forma praticamente inviolável;
- a possibilidade de gerar novos *assets* para concessões de privilégios de acesso, ou mesmo mudar os requisitos na prestação do serviço, permite o manutenção e revogação de direitos de acesso;
- ao basear suas operações em requisições existentes em uma infraestrutura de blockchain, a proposta atende também a requisitos de usabilidade.

Aspectos de escalabilidade e operação das transações em tempo real, precisam ser melhor investigados nos modelos propostos. Isso vale especialmente para plataformas de blockchain nas quais o *proof-of-work* demanda processamento significativo, assim como as operações de verificação completa das operações em relação aos blocos de registro.

Referências

AZARIA, A. et al. Medrec: Using blockchain for medical data access and permission management. In: *2016 2nd International Conference on Open and Big Data (OBD)*. [S.l.: s.n.], 2016. p. 25–30.

BERTINO, E.; KIRKPATRICK, M. S. Location-based access control systems for mobile users: Concepts and research directions. In: *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*. New York, NY, USA: ACM, 2011. (SPRINGL '11), p. 49–52. ISBN 978-1-4503-1032-1.

CACHIN, C. et al. Architecture of the hyperledger blockchain fabric*. In: . [S.l.: s.n.], 2016.

CHOI, S. et al. Secure and resilient proximity-based access control. In: *Proceedings of the 2013 International Workshop on Data Management & Analytics for Healthcare*. New York, NY, USA: ACM, 2013. (DARE '13), p. 15–20. ISBN 978-1-4503-2425-0.

CLEEFF, A. v.; PIETERS, W.; WIERINGA, R. Benefits of location-based access control: A literature study. In: *Proceedings of the 2010 IEEE/ACM Int'L Conference on*

Green Computing and Communications & Int'L Conference on Cyber, Physical and Social Computing. Washington, DC, USA: IEEE Computer Society, 2010. (GREENCOM-CPSCOM '10), p. 739–746. ISBN 978-0-7695-4331-4.

DEY, A. K. Understanding and using context. *Personal Ubiquitous Comput.*, Springer-Verlag, London, UK, UK, v. 5, n. 1, p. 4–7, jan. 2001. ISSN 1617-4909.

ENGLISH, M.; AUER, S.; DOMINGUE, J. Block chain technologies & the semantic web: A framework for symbiotic development. In: *Computer Science Conference for University of Bonn Students, J. Lehmann, H. Thakkar, L. Halilaj, and R. Asmat, Eds*. [S.l.: s.n.], 2016. p. 47–61.

FERRAIOLO, D.; KUHN, R. Role-based access controls. In: BALTIMORE, MARYLAND: NIST-NCSC. *Proceedings of 15th NIST-NCSC National Computer Security Conference*. [S.l.], 1992. v. 563.

GUPTA, A.; KIRKPATRICK, M.; BERTINO, E. A formal proximity model for rbac systems. In: *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*. [S.l.: s.n.], 2012. p. 1–10.

HASHEMI, S. H. et al. World of empowered iot users. In: *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*. [S.l.: s.n.], 2016. p. 13–24.

HU, V. C. et al. Guide to attribute based access control (abac) definition and considerations (draft). *NIST special publication*, v. 800, n. 162, 2014.

KUHN, D. R.; COYNE, E. J.; WEIL, T. R. Adding attributes to role-based access control. *Computer*, v. 43, n. 6, p. 79–81, June 2010. ISSN 0018-9162.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. November 2008.

OUADDAH, A.; ELKALAM, A. A.; OUAHMAN, A. A. Fairaccess: a new blockchain-based access control framework for the internet of things. *Security and Communication Networks*, v. 9, n. 18, p. 5943–5964, 2016. ISSN 1939-0122. SCN-16-0184.

TONINELLI, A. et al. A semantic context-aware access control framework for secure collaborations in pervasive computing environments. In: ____. *The Semantic Web - ISWC 2006: 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 473–486. ISBN 978-3-540-49055-5.

WOOD, G. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, v. 151, 2014.

ZYSKIND, G.; NATHAN, O.; PENTLAND, A. S. Decentralizing privacy: Using blockchain to protect personal data. In: *Proceedings of the 2015 IEEE Security and Privacy Workshops*. Washington, DC, USA: IEEE Computer Society, 2015. (SPW '15), p. 180–184. ISBN 978-1-4799-9933-0.

Proof-of-Stake baseado em Tempo Discreto

Yoshitomi Eduardo Maehara Aliaga, Diego Fernandes Gonçalves Martins, Marco Aurélio Amaral Henriques

[1]Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
13083-852 – Campinas, SP, Brasil

{ymaehara,diegofgm,marco}@dca.fee.unicamp.br

Abstract. *In this work, we present a new consensus mechanism for blockchains based on Prood-of-Stake, which makes easier the participation of nodes in the mining process. The new mechanism uses rounds based on a discrete time counter, where only participants who have passed the round challenge can generate a block. Therefore, it can guarantee a better chance of participation to each node because it does not require expensive hardware setup to mine new blocks.*

Resumo. *Neste trabalho apresentamos um novo mecanismo de consenso para blockchains baseadas em Proof-of-Stake, o qual facilita a participação de nós no processo de mineração. O novo mecanismo usa rodadas baseadas em um contador de tempo discreto, onde apenas participantes que tenham passado no desafio da rodada podem gerar um bloco. Portanto, ele pode garantir uma melhor chance de participação para cada nó, já que não exige caras instalações de hardware para a mineração de novos blocos.*

1. Introdução

As *blockchains* públicas têm atraído muita atenção principalmente depois da popularização das criptomoedas. A *blockchain* pode ser vista como um livro razão, onde as transações inseridas são imutáveis, após decorrido um tempo de confirmação. Aplicações que usam *blockchains* tornam-se cada dia mais populares e várias propostas inovadoras têm surgido para cartórios digitais, gestão de identidades, armazenamento confiável de informações críticas, criptomoedas, entre outras.

Uma das partes mais importantes de uma *blockchain* é o mecanismo de consenso, que permite que partes desconhecidas entre si e até mesmo concorrentes cheguem a um acordo sobre o estado atual e passado dos dados armazenados. Portanto, o consenso em *blockchain* viabiliza a criação de novos blocos de forma distribuída sem exigir validação por uma terceira parte confiável mesmo que existam participantes não confiáveis. Bashir [Bashir 2017] define o mecanismo de consenso como um conjunto de passos que são dados por todos ou pela maioria dos nós para entrar em consenso a respeito de um estado proposto ou valor. Em outras palavras este algoritmo permite que dois ou mais processos concordem sobre um determinado valor ou estado de uma rede. Um mecanismo de consenso pode ser visto como uma solução ao Problema dos Generais Bizantinos [Lamport et al. 1982] que consiste em resolver o dilema de atingir um consenso entre os participantes (que neste caso são os generais) com um objetivo comum. Embora existam generais traidores, os quais possuem objetivos opostos ao consenso e que tentarão

atrapalhar o processo, a ideia é que os generais leais (se forem maioria) atinjam o objetivo comum, sem que os traidores consigam impedí-los.

Dentre as propostas para mecanismos de consenso em *blockchains*, a mais conhecida é *Proof-of-Work* (PoW) [Nakamoto 2009]. Trata-se do mecanismo utilizado na mais popular das criptomoedas: *Bitcoin*. Porém, apesar de resolver o problema de consenso para uma criptomoeda de sucesso, ele traz problemas associados ao uso intensivo de poder computacional e ao alto consumo de energia. Este problema está diretamente ligado à segurança da *blockchain*, pois incentiva a formação de grupos de poderosos participantes (com acesso a instalações de alto poder computacional), que acabam dominando o processo de geração de novos blocos. Torna-se mais fácil então que se forme um grupo que supere os 50% de todo o poder computacional de mineração e comprometa a integridade da *blockchain*. Estes nós poderiam trabalhar em cadeias paralelas de blocos, chamadas de *forks*, revertendo transações já realizadas e colocando em risco a confiança no sistema. Esta exigência de alto poder computacional do PoW acaba dificultando a descentralização da rede e desmotivando a entrada de novos participantes.

Além disso o PoW é extremamente ineficiente quanto ao consumo de energia. Todos os participantes da rede utilizam alto poder computacional de suas máquinas mineradoras para vencer o desafio de geração do próximo bloco no menor tempo possível. Entretanto, a maior parte de toda essa energia é desperdiçada, pois apenas um participante de fato irá vencer a corrida e ganhar o direito de gerar um dado bloco n. Todos os demais participantes têm que abandonar suas tentativas de minerar o mesmo bloco e recomeçar do zero para tentar vencer a corrida pela mineração do bloco $n + 1$.

Por este motivo, foram propostos outros mecanismos de consenso tais como *Proof-of-Stake* [King 2013], *Proof-of-Activity* [Bentov et al. 2014], *Delegated-Proof-of-Stake* [Bitshare 2016], entre outros, como uma forma de contornar as desvantagens presentes no *Proof-of-Work*. Uma discussão mais detalhada e uma comparação entre estes mecanismos pode ser encontrada no trabalho de Maehara et al. [Maehara et al. 2018]

Neste texto apresentamos uma proposta de mecanismo de consenso baseado na ideia de sorteio presente no PoS. Trata-se de uma abordagem simplificada de um algoritmo *Proof-of-Stake*, onde novos blocos só podem ser gerados dentro de uma rodada esperada. Por meio do uso de um esquema síncrono, outros participantes podem calcular a rodada esperada de um bloco recebido da rede e compará-la com a rodada utilizada pelo participante criador do bloco. Assim é possível decidir se o bloco deve ser aceito, caso sua rodada seja a esperada, ou ignorado, caso contrário.

As seções seguintes irão destacar as principais características do mecanismo PoS, apresentar a proposta de um PoS baseado em tempo discreto, detalhar seus principais parâmetros e mostrar alguns resultados teóricos e práticos resultantes de uma implementação de prova de conceito.

2. Proof of Stake (PoS)

O *Proof-of-Stake* é um mecanismo de consenso em que o sistema faz uma escolha do nó que poderá criar um novo bloco por meio de um sorteio, cuja probabilidade de sucesso (de ser sorteado) pode ser influenciada pela sua quantidade de moedas.

Assim como no *Bitcoin*, o participante deve calcular o *hash de prova*, o qual é

resultante do cálculo de uma função $hash$ de boa qualidade sobre elementos do cabeçalho do bloco a ser criado. Uma vez calculado o *hash de prova*, este é comparado com um objetivo definido pelo sistema de modo a refletir um certo grau de dificuldade. Caso seja menor do que este objetivo, ocorre o sucesso do sorteio e o participante ganha o direito de criar o novo bloco e auferir os ganhos decorrentes de tal criação. A Eq. 1 define a condição que deve ser testada.

$$hash_de_prova < objetivo \tag{1}$$

Quanto maior o objetivo, menor é a dificuldade de satisfazer esta inequação.

Vasin [Vasin 2013] destaca que, para a geração de um bloco, é necessária uma prova de que o nó possua uma certa quantidade de moedas. Antes de se candidatar para gerar um bloco, um nó deve enviar moedas a si mesmo, para provar a propriedade das mesmas. A quantidade de moedas de um nó pode ser multiplicada pelo objetivo, aumentando assim a probabilidade de que tal nó venha a ter sucesso no sorteio. Daí advém o nome Proof-of-Stake.

Existem na literatura duas formas básicas de combinação do $stake$ com o objetivo: na criptomoeda *Peercoin* [King 2013] é usada a idade da moeda de um participante como modificador do *stake*, ou seja, o participante que possuir moedas por mais tempo possui maior probabilidade de gerar o próximo bloco.

Já no consenso utilizado em *Blackcoin* [Vasin 2013], apenas a quantidade de moedas é usada na modificação do objetivo. Este processo aumenta a possibilidade de criar grupos mais privilegiados na escolha para gerar novos blocos, o que pode ser um problema para atingir uma maior descentralização na rede, caso não sejam tomados cuidados para evitar uma grande influência do $stake$ de cada nó no processo.

Uma outra forma de se alterar o valor do objetivo é multiplicando-o pelo tempo decorrido desde a criação do último bloco. Desta forma, caso o nível de dificuldade usado no objetivo esteja alto e nenhum nó consiga sucesso no sorteio, em uma etapa seguinte o valor do tempo a ser multiplicado pelo objetivo irá aumentar, aumentando assim as chances de que algum nó seja sorteado.

Algumas características do mecanismo PoS:

- normalmente não possui recompensa por gerar um bloco, mas os mineradores recebem as taxas das transações contidas no mesmo [GreenfieldIV 2017];
- há um desafio grande em rastrear de forma confiável as mudanças de saldo de cada participante [Bano et al. 2017];
- este mecanismo diminuiu o gasto de energia em comparação com PoW, já que não incentiva e não depende de poder computacional elevado;
- pode existir um problema conhecido como "*nothing-at-stake*", onde os mineradores não têm nada a perder ao minerar em dois ou mais ramos (*forks*) ao mesmo tempo, com o objetivo de recolher as taxas de vários blocos gerados em ramos distintos;
- é mais caro realizar uma fraude em PoS que em PoW, já que para tentar possuir 51% das chances de ser sorteado (isto é, garantir o sorteio), é preciso possuir e usar muitas moedas; para isso o atacante que não as tenha poderá gerar uma demanda pelas mesmas, causando imediatamente sua valorização e dificultando a obtenção das que faltam;

- o PoS permite aumentar uma eventual concentração de riquezas, já que quem possui mais moedas acaba tendo uma chance maior de ser escolhido para criar novos blocos e ganhar as taxas por isso, ficando ainda mais rico.

As principais moedas que utilizam este mecanismo de consenso são NxT [Community 2014], BlackCoin [Vasin 2013] , PeerCoin [King 2013], Ethereum (Slasher [Buterin 2014] & Casper [Buterin and Griffith 2017]) e Cardano (Ouroboros)[Kiayias et al. 2017] .

3. Proposta de um novo mecanismo de consenso baseado em PoS

Antes de detalharmos a proposta, iremos definir alguns conceitos e parâmetros.

3.1. Definição de Parâmetros

Os mecanismos baseados em PoS, quando comparados àqueles baseados em PoW, trazem ganhos em relação ao consumo de energia e facilitam a participação de qualquer nó interessado em criar novos blocos. Entretanto, eles também trazem alguns desafios, tais como: como definir se um nó foi sorteado em um determinado instante? Como detetar e limitar o crescimento de cadeias secundárias geradas a partir de *forks*? Como aumentar a descentralização na geração de novos blocos? A seguir serão discutidos o sorteio do nó em uma rodada e o controle de *forks*, explicando como são tratados no novo mecanismo proposto.

3.1.1. Sorteio do nó em uma rodada

Como em todo mecanismo de sorteio, é necessário verificar se um determinado nó participante realmente foi sorteado. Em um ambiente distribuído é essencial que todos os outros participantes possam verificar quem de fato foi sorteado em uma determinada rodada. Nesta proposta, uma rodada é um intervalo de tempo representado por um número inteiro (número de intervalos de tempo transcorridos desde um dado instante). Cada nó participante deve tentar gerar o próximo bloco da cadeia blockchain sempre dentro de uma rodada. Se não tem sucesso no sorteio em uma dada rodada, precisa aguardar pela próxima para fazer uma nova tentativa.

A rodada para o próximo bloco da cadeia é encontrada por meio da quantidade de tempo, desde a chegada do último bloco aceito, dividida por um intervalo fixo chamado de $TIMEOUT$ (Eq. 2). O valor do $TIMEOUT$ é global e todos os participantes podem calcular a rodada esperada para um novo bloco a partir da rodada registrada em um bloco já criado. Observamos que o único valor variável na equação da rodada é o tempo atual. Ou seja, o valor da rodada só irá mudar após transcorrido um intervalo de tempo igual a $TIMEOUT$ desde o último cálculo da rodada pelo nó em questão.

$$\begin{aligned} rodada &= último_bloco.rodada \\ &+ \left\lfloor \frac{tempo_atual() - último_bloco.tempo_chegada}{TIMEOUT} \right\rfloor \end{aligned} \tag{2}$$

onde:

- ***tempo_atual*()** representa a data e hora atual no padrão UTC,

- ***último_bloco.tempo_chegada*** representa o tempo de chegada do bloco anterior,
- ***último_bloco.rodada*** representa a rodada do bloco anterior,
- $TIMEOUT$ representa a duração de cada rodada.

Como já foi mostrado pela Eq. 1, cada participante deve calcular um *hash de prova* e o resultado do mesmo deve ser inferior a um dado objetivo. Supõe-se o uso de uma função *hash* de boa qualidade, na qual as saídas não demonstram correlação entre si, mesmo se as entradas forem correlacionadas. Nesse caso, tal função *hash* pode ser usada como gerador de números pseudoaleatórios ou como um mecanismo de sorteio. Se este *hash* de prova usa como entrada a identidade (fixa) do nó participante e outros dados que não podem ser alterados por ele, então conseguir vencer o objetivo é equivalente a vencer um sorteio.

O *hash de prova* usado nesse novo mecanismo está baseado na função de *hash* criptográfico SHA-256 e é calculado pela Eq. 3.

$$hash_de_prova = H(último_bloco.hash \| rodada \| identificador_do_nó) \quad (3)$$

onde:

- $H()$: é a função *hash* SHA-256,
- ***último_bloco.hash***: é o *hash* do último bloco da cadeia,
- ***rodada***: é o valor da rodada atual,
- ***identificador_do_nó***: é a identidade do nó baseada em sua assinatura.

O cálculo do *hash* de prova funciona então como uma espécie de sorteio baseado nas três informações de entrada. Dependendo do valor definido para o objetivo (Equação 1), é provável que o nó não consiga um *hash* de prova pequeno o suficiente e tenha que fazer nova tentativa, o que só poderá ocorrer em uma próxima rodada, já que os demais elementos de entrada da função são fixos. Isto impede que nós com grande poder computacional possam fazer mais tentativas de sorteio que os nós mais simples, uma vez que todos os nós só poderão calcular um *hash* de prova a cada intervalo de tempo.

O valor do parâmetro $TIMEOUT$ deverá ser escolhido cuidadosamente, pois não poderá ser tão pequeno que privilegie os nós mais rápidos (e facilite a criação de *forks*), nem tão grande que prejudique o desempenho da rede.

3.1.2. Controle de forks

Diferentemente dos mecanismos baseados em prova de trabalho, onde minerar blocos em duas ou mais cadeias gera um alto custo, nos mecanismos PoS este custo é desprezível, pois limita-se ao cálculo de um *hash*. Por isso, mecanismos de consenso PoS permitem que um participante crie blocos em diferentes cadeias e não apenas na cadeia mais longa, o que resulta na geração de *forks* indesejáveis.

Considerando que o *stake* de um determinado participante seja igualmente distribuído entre as cadeias, ele possui, nesse caso, a mesma probabilidade de sucesso no sorteio para todas as cadeias consideradas. Como consequência disso, ele pode contribuir,

sem custos adicionais, para que uma cadeia secundária, em um determinado momento, fique mais longa que a cadeia principal, causando efeitos indesejáveis para a rede. Na literatura esse problema é conhecido como *Nothing at Stake*, ou seja, um participante não tem nada a perder em tentar minerar ao mesmo tempo em várias cadeias diferentes.

Segundo [Buterin and Griffith 2017] é necessário criar mecanismos para punir os participantes que desrespeitam a geração de blocos para a cadeia mais longa. Nesta linha, ele propõe um algoritmo chamado *Casper*, que será o substituto do atual algoritmo baseado em PoW no sistema distribuído *Ethereum*. No *Casper*, a cada 100 blocos são fixados *checkpoints*, nos quais os nós participantes têm que votar para definir qual é a cadeia principal entre cada par de *checkpoints* consecutivos, criando os *enlaces majoritários* entre *checkpoints*. Desde que dois terços dos participantes sejam honestos, é possível saber com precisão qual é a cadeia principal a seguir, pois neste caso o protocolo garante a finalização apenas de *checkpoints* não conflitantes. Além disso, *Casper* penaliza com perda de moedas os nós participantes que não sigam a regra.

Na nova proposta, a penalidade é mais branda e estará na menor probabilidade de sorteio para um participante que deseja criar um bloco em uma cadeia secundária. É permitido ao participante criar blocos em qualquer *fork* da rede, porém a probabilidade de geração de blocos no ramo principal é consideravelmente maior quando comparada com outras cadeias. Assim, é pouco provável que uma cadeia que perdeu a disputa em um determinado *fork* em um tempo passado assuma a condição de cadeia principal. A proposta para diferenciar a probabilidade de mineração por cadeia é definir o objetivo como na Eq. 4.

$$\begin{aligned} objetivo &= stake \times 2^{tamanho_do_hash - dificuldade - penalidade} \\ &= stake \times 2^{256-D-P} \end{aligned} \tag{4}$$

Este objetivo também deve ser compatível com o *hash* de prova, isto é, deve ser representável por, no máximo, o número de bits da função de *hash* utilizada. O parâmetro $dificuldade$ ($D < 256$) controla a dificuldade de sucesso no sorteio, com impactos na velocidade de convergência do algoritmo.

Já o parâmetro $penalidade$ ($P < 256 - D$) define o nível de aumento da dificuldade para se minerar nas cadeias não principais (menores). Caso seja criado um bloco na cadeia principal, fazemos $P = 0$. Com valores elevados para P ($P \simeq 256 - D$), esta proposta de controle de *forks* equivale na prática a proibir a geração de blocos em cadeias secundárias.

Este controle de *forks* pode dar uma maior agilidade ao mecanismo, uma vez que permite que os nós tentem minerar blocos em cadeias secundárias, ainda que com uma probabilidade menor de sucesso. A motivação para esta estratégia reside no fato de que é possível que nós em pontos distantes da rede tenham visões diferentes sobre quais cadeias são secundárias e qual é a principal, devido a atrasos de propagação de blocos. Assim, mesmo que um nó considere uma dada cadeia como secundária, ele tem permissão para tentar criar um novo bloco para a mesma (com menor probabilidade de sucesso) na expectativa de ser bem sucedido futuramente, caso esta cadeia secundária seja, na verdade, a principal, de acordo com a visão dos demais nós da rede. Se esta estratégia não for bem controlada, no entanto, ela pode contribuir para um aumento dos *forks* e uma maior dificuldade da rede em descobrir qual é a cadeia principal.

3.2. Proof-of-Stake baseado em Tempo Discreto (PoSBTD)

Esta seção detalha a proposta do novo mecanismo, chamada de PoS baseado em Tempo Discreto (PoSBTD) , pois a criação de blocos só ocorre em intervalos de tempo fixos (rodadas).

Apesar de estarmos tratando de um mecanismo baseado em PoS, no qual cada nó coloca seu $stake$ para influenciar sua chance de sucesso no sorteio, limitaremos nossas análises iniciais ao caso em que todos os nós têm um mesmo valor de $stake = 1$. Assim, eliminamos a diferenciação que nós mais ricos poderiam impor aos mais pobres e igualamos as chances de qualquer nó ter sucesso no sorteio (observe que a obrigatoriedade de se minerar somente um vez por rodada também iguala as chances entre nós poderosos e nós mais simples).

O mecanismo proposto pode ser dividido em três processos principais que trabalham de forma concorrente, como descrito a seguir.

3.2.1. Processo de mineração de novos blocos

O processo de mineração consiste em o nó participante verificar se obtém sucesso na comparação do *hash de prova* com o objetivo na rodada atual. Em caso de sucesso, ele calcula e divulga (por meio de *broadcast*) um novo bloco para cada uma das cadeias que conhece; caso contrário, ele aguarda a chegada de um novo bloco calculado por outro nó ou uma nova rodada para fazer uma nova tentativa (Fig. 1).

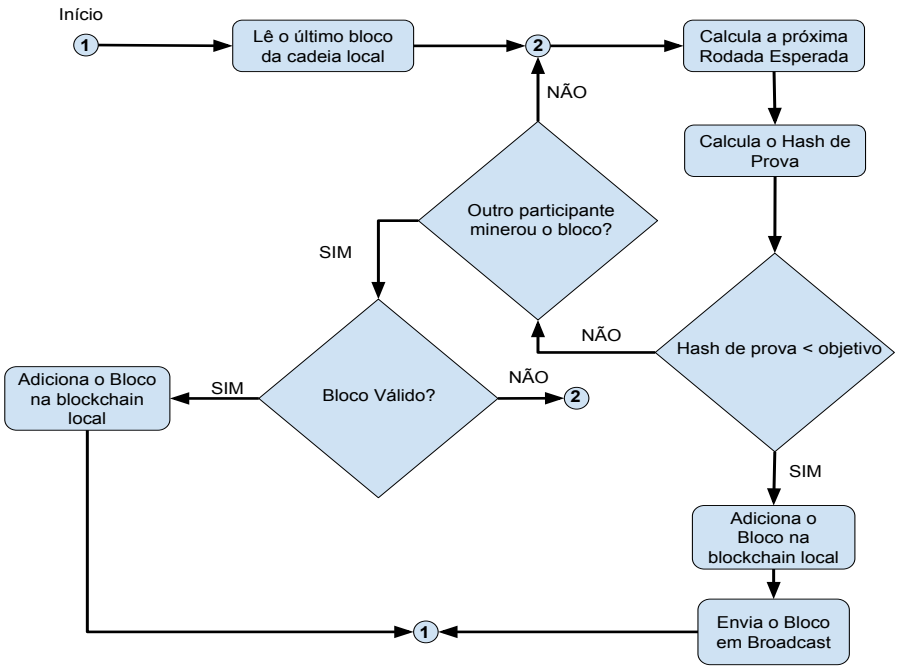


Figura 1. Processo de mineração de novos blocos

3.2.2. Processo de escuta de novos blocos

O processo de escuta é responsável por receber novos blocos criados por outros participantes e validá-los, buscando sua inserção na *blockchain* local (Fig. 2). Ele fica esperando uma mensagem com um novo bloco e, quando ela chega, é verificado se a cadeia local e este bloco estão sincronizados (isto é, se o bloco se encaixa no final da cadeia local ou se há alguma lacuna entre ele e a cadeia). Caso exista tal lacuna, o nó chama o processo de sincronização para completar sua cadeia local, o que pode também trazer novos *forks* para a mesma. Se não há lacuna, a consistência do bloco é verificada por meio do cabeçalho do bloco, da rodada esperada e da confirmação se o nó que criou e propagou o bloco de fato passou no teste do objetivo. Caso qualquer uma destas verificações falhe, o bloco é descartado e o processo volta ao início. Caso contrário, ele é aceito e a anexado à *blockchain* local.

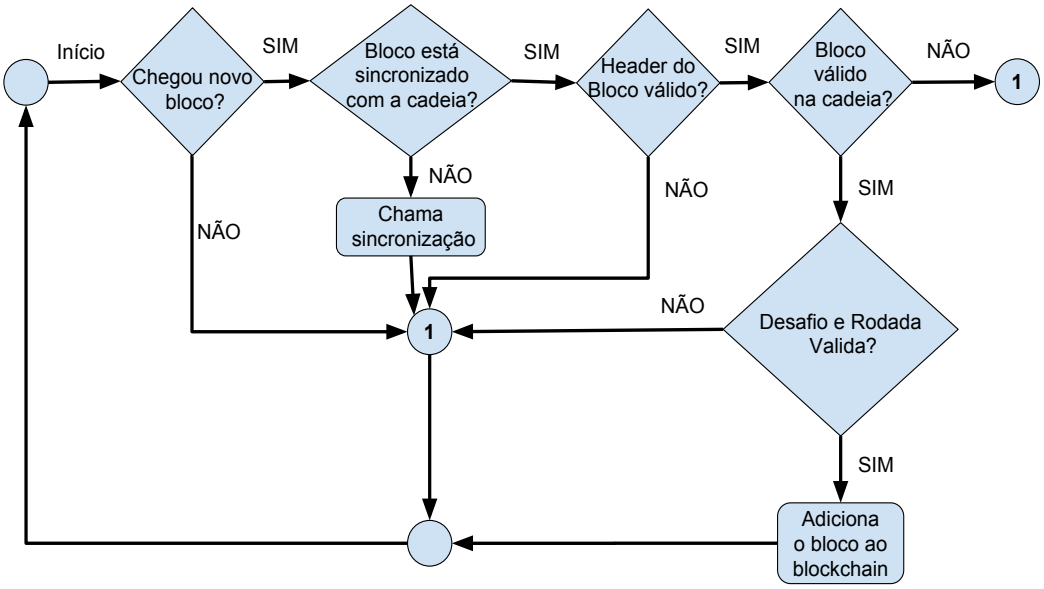


Figura 2. Processo de escuta de novos blocos

3.2.3. Processo de sincronização de blocos

O processo de sincronização consiste em buscar novos blocos de outros participantes para obter uma visão mais completa da *blockchain* (Fig. 3). Ao ser executado, o processo contata os *peers* do nó e solicita um a um os blocos que faltam, a partir do bloco com maior índice recebido, para preencher a lacuna que foi identificada no processo de escuta de novos blocos. Ao receber cada um dos blocos, o nó faz as verificações de consistência dos cabeçalhos e das rodadas antes de adicioná-los na *blockchain* local. Caso algum bloco não seja da cadeia principal, isso significa que existe um *fork* em algum ponto da cadeia. Esse *fork* é localizado e uma troca da cadeia principal pode ocorrer.

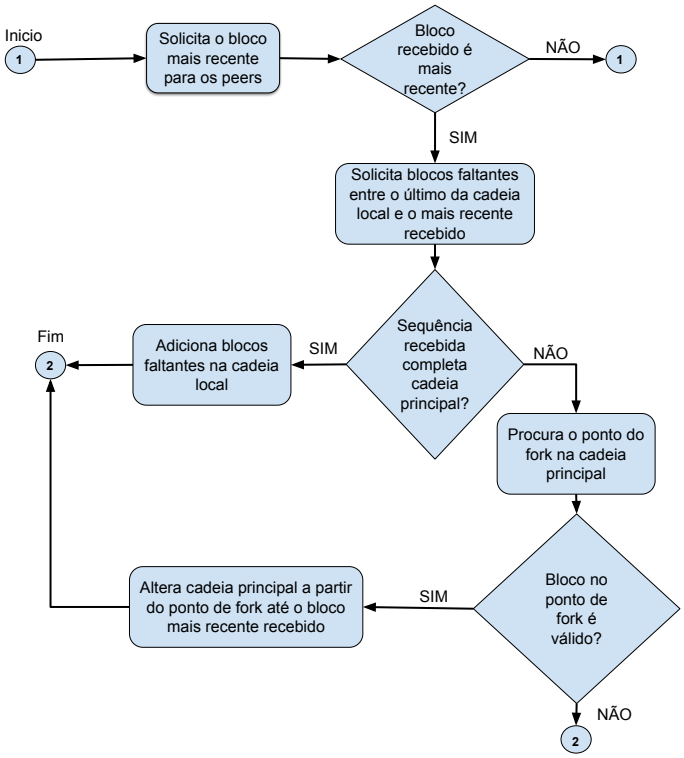


Figura 3. Processo de sincronização de novos blocos

4. Testes e Resultados

4.1. Resultados teóricos

Inicialmente faremos uma análise teórica do comportamento do mecanismo proposto.

Considerando que será utilizado um *stake* unitário para todos os nós, a Eq. 5 apresenta a probabilidade de ocorrência de um evento S_1 (sucesso no sorteio em um nó, isto é, sucesso em atingir o objetivo por meio do *hash de prova*).

$$p[S_1] = \frac{2^{256-D-P}}{2^{256}} \tag{5}$$

Usando o resultado da Eq. 5 podemos encontrar o número esperado de rodadas até ocorrer sucesso no sorteio em um nó (Eq. 6).

$$Nr_{S_1} = \left\lceil \frac{1}{p[S_1]} \right\rceil \tag{6}$$

Já o valor esperado para múltiplos nós depende da probabilidade de sucesso S_n, que é o sorteio bem sucedido em pelo menos um dos n nós participantes (Eq. 7).

$$p[S_n] = 1 - \prod_{i=1}^{n} (1 - p[S_1]) = 1 - (1 - p[S_1])^n \tag{7}$$

De forma similar pode-se encontrar o número esperado de rodadas para se ter algum sorteio (*hash*) bem sucedido quando se tem mais de um nó (Eq. 8).

$$Nr_{S_n} = \left\lceil \frac{1}{p[S_n]} \right\rceil \tag{8}$$

Com base nestas expressões foi avaliado, de forma teórica, o comportamento de uma *blockchain* controlada pelo novo mecanismo de consenso proposto.

A Fig. 4 mostra que a probabilidade de sucesso em uma dada rodada aumenta com o número de nós participantes na *blockchain*, mas diminui muito rapidamente quando cresce o nível de dificuldade. Foi adotada uma penalização P nula, já que é considerada a criação de bloco no ramo principal da *blockchain*.

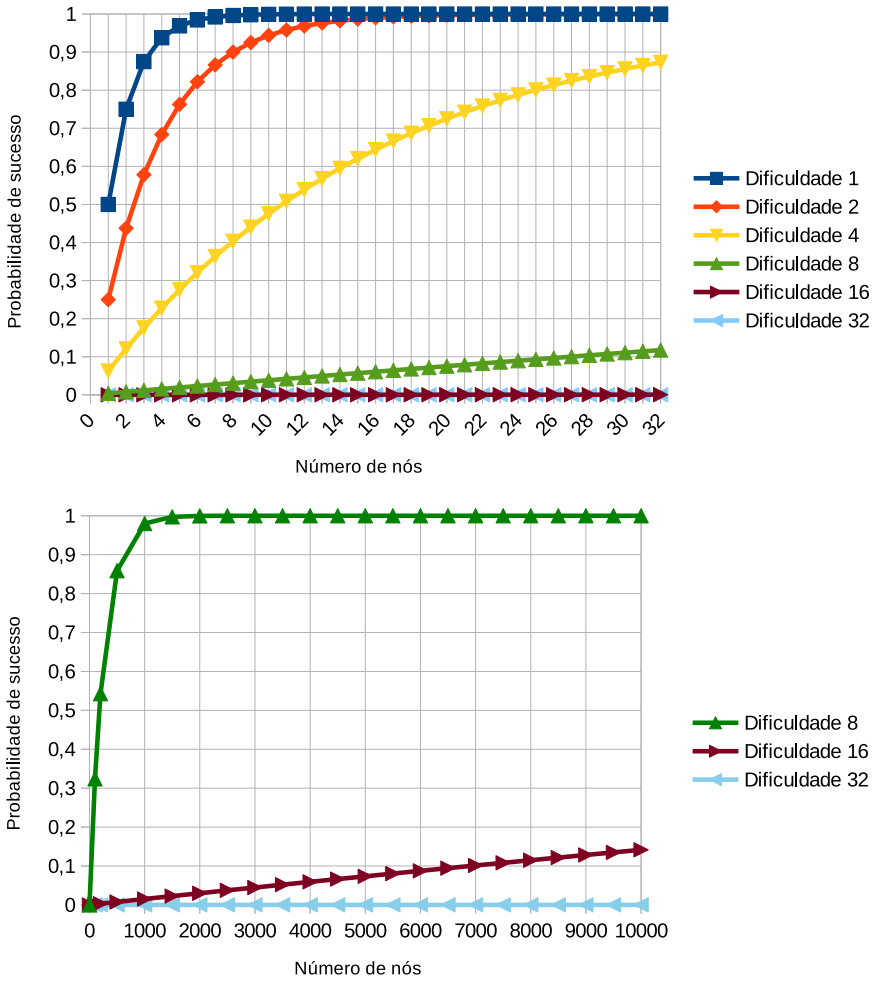


Figura 4. Probabilidade de sucesso no sorteio em função do número de nós

Na prática, os níveis de dificuldade devem ser ajustados de maneira a não permitir que seja muito provável qualquer nó ter sucesso no sorteio (aumentaria o número de *forks*), mas também não pode tornar esta probabilidade muito pequena, pois isso resultaria em atrasos muito grandes no processo de mineração. É possível notar que mesmo para uma rede similar em tamanho à rede do *Bitcoin* (cerca de 10.000 nós), os níveis de dificuldade 16 e 32 parecem ser muito fortes, não permitindo que a probabilidade de sucesso no sorteio de algum nó atinja os 15%. Pode parecer pouco, mas 15% de 10.000 são 1.500 nós tendo sucesso no sorteio, o que pode resultar em 1.500 *forks* a serem resolvidos. É por este motivo que se faz necessário um controle bem rigoroso do nível de dificuldade a fim de evitar que o número de *forks* aumente.

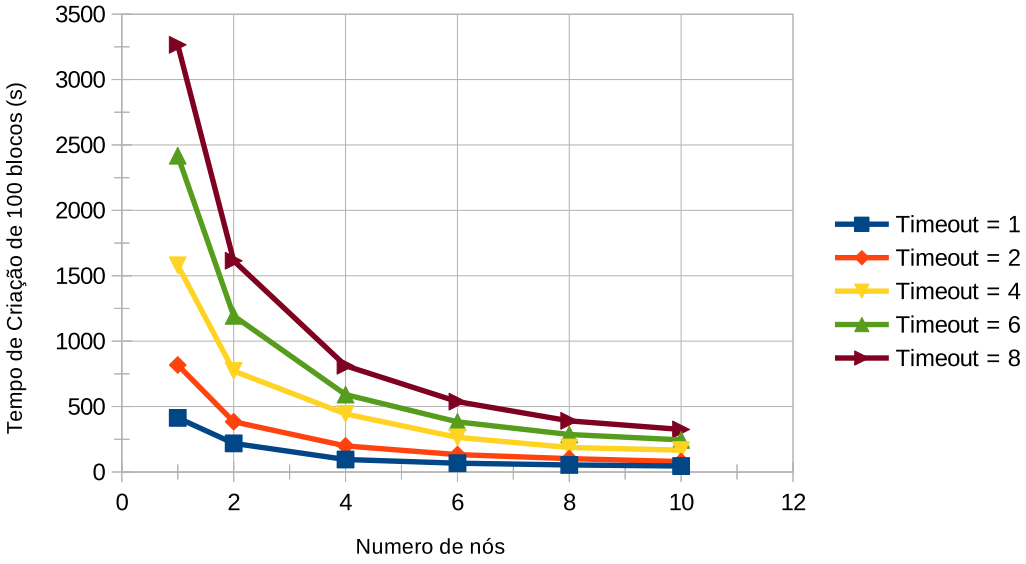


Figura 5. Tempo médio de criação de 100 blocos em função do número de nós para dificuldade 2 e P=0

4.2. Resultados práticos

Devido ao tempo necessário para cada teste e à dificuldade de gerenciar um número grande de nós que formam a *blockchain*, optamos por limitar o tamanho da rede em 10 nós e o $TIMEOUT$ em 8 segundos nesta avaliação inicial. Avaliações com outras configurações também foram efetuadas e os resultados foram coerentes com o esperado e com os aqui apresentados.

Os testes foram executados em um servidor de 16 *GB* de memória RAM e processador Intel *Xeon* de 2,4 *GHz* com um core. Os nós mineradores foram criados no simulador de redes *Mininet* em sua versão 2.3.0 usando a linguagem Python. Na simulação, todos os nós foram criados no mesmo servidor, não sendo utilizado link externo para comunicação com nós remotos. Nesses testes, foi adotado um nível de dificuldade $D = 2$ bits e penalidade $P = 0$.

A Fig. 5 mostra os tempos médios obtidos em cinco execuções do sistema para criar 100 blocos. A principal informação que se visualiza nas curvas para diferentes $TIMEOUTs$ é o tempo cada vez menor para criação de 100 blocos à medida que o tamanho da rede aumenta, como já era previsto teoricamente. A figura confirma também o crescimento esperado dos tempos de criação em função do aumento do $TIMEOUT$. Este crescimento do tempo é cada vez menor à medida que o sistema fica maior, dada a maior probabilidade de se obter sucesso no sorteio em poucas rodadas, o que diminui parcialmente o impacto do $TIMEOUT$. Resultados muito similares foram obtidos para $D = 4$ e $P = 0$, havendo apenas uma mudança de escala de tempo.

Na figura 6 temos uma comparação entre os tempos teóricos e práticos para a criação de 100 blocos para uma quantidade de nós variando de 1 até 10. É possível observar uma diferença maior entre os valores à medida que aumenta o número de nós. Essa diferença se deve principalmente ao fato de que o valor estimado pela Eq. 8 para

o número de rodadas está arrendondado para o primeiro inteiro acima. o que traz um erro maior quando menos rodadas são necessárias para se ter sucesso em um sorteio. Por outro lado, quanto mais nós estiverem participando dos sorteios, maior é a chance de algum deles conseguir obter sucesso antes do tempo estimado, já que se trata de uma estimativa baseada em probabilidades. Estes são os dois principais fatores responsáveis pela maior diferença entre os tempos para vários nós.

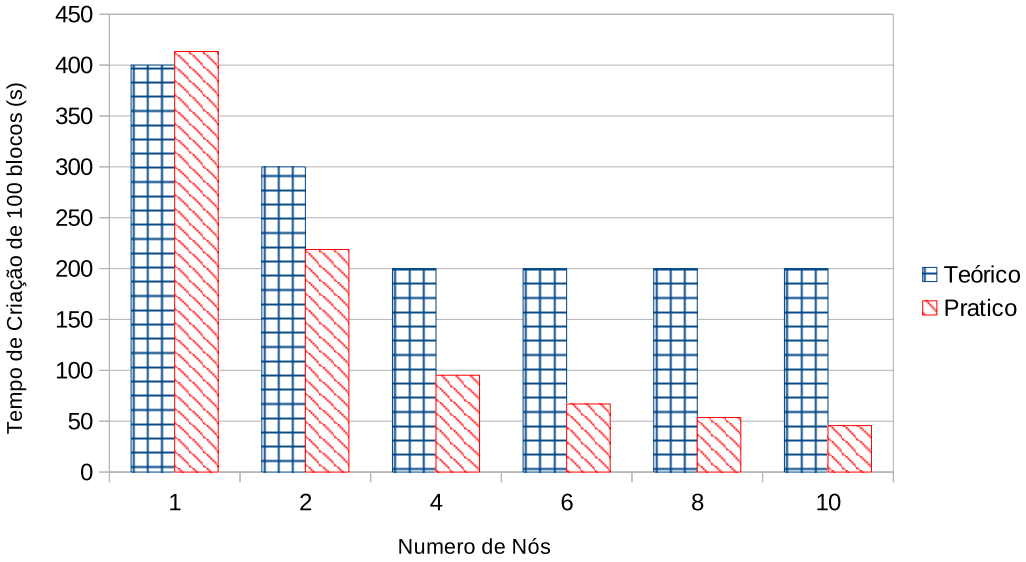


Figura 6. Comparação entre os tempos estimados e medidos para criação de 100 blocos com dificuldade 2 e $TIMEOUT = 1$

5. Segurança do PoSBTD

Ao contrário dos mecanismos baseados em PoW, nos quais há um custo muito alto para se produzir blocos fora da cadeia principal, naqueles baseados em PoS este custo é desprezível e um controle mais rigoroso sobre forks precisa ser exercido. Várias propostas de sistemas baseados em PoS fazem uso da ideia de se ter blocos especiais, chamados *checkpoints* que são pontos de controle sobre os quais os demais nós concordam que são definitivos e imutáveis. Entretanto, esta ideia de se usar *checkpoints* é bastante complexa e este é um dos motivos para não termos ainda um sistema de grande porte (de criptomoedas ou não) baseado nestas ideias do PoS.

No PoSBTD proposto, temos um controle de forks bem simples: os nós podem criar nós em cadeias secundárias, mas com uma probabilidade de sucesso bem menor do que a que têm na cadeia principal. Ainda são necessárias melhores avaliações desta proposta, mas acreditamos que seja uma forma simples e eficaz de evitar a proliferação de *forks*.

Outro problema de esquemas PoS é a Reversão de longo alcance, ou seja, um conjunto significativo de nós trabalhar em uma cadeia secundária e num dado momento propor a troca da cadeia principal pela secundária, revertendo muitas transações passadas. Para este caso também consideramos que o mecanismo proposto oferece uma boa robustez pela sua forma de construção, que exige a construção de blocos dentro de intervalos de tempo bem demarcados.

Assim como ocorre em outros mecanismos PoS ou até mesmo PoW, boa parte da segurança em PoSBTD advém do uso de funções $hash$ de boa qualidade, consideradas criptograficamente seguras para o intertravamento de blocos e para o sorteio dos nós autorizados a criar um novo bloco. Apesar de as transações que ocupam o corpo de um bloco não terem sido objeto das discussões neste trabalho, assume-se que elas estariam protegidas por assinaturas digitais tão seguras quanto possível e por uma Árvore de Merkle baseada em funções de $hash$ seguras, exatamente como é feito em outras blockchains, como a do *Bitcoin*, por exemplo.

6. Conclusões e Trabalhos Futuros

Os protocolos baseados em PoS são promissores para resolver problemas ligados ao consenso de *blockchain* públicos. Os grandes desafios estão ligados principalmente aos problemas de concorrência das cadeias geradas a partir de *forks* e também à dificuldade de rastreamento do *stake* possuído e utilizado por um determinado participante. Algumas propostas de sistemas baseados em PoS apareceram nos últimos anos na literatura, mas, devido principalmente à complexidade envolvida, nenhuma delas ainda está em produção em sistemas de maior porte.

O mecanismo proposto neste trabalho traz uma solução ao problema de grandes diferenças de oportunidades na definição do criador de um bloco. A obrigatoriedade de espera pela próxima rodada para se fazer uma nova tentativa de criação elimina as vantagens de poder computacional que um nó possa ter em relação a outros. Além disso, ele indiretamente resolve a questão do grande consumo de energia característico do PoW e propõe um controle simples de *forks*. Um protótipo do sistema com o mecanismo de consenso foi implementado e testado, apresentando resultados compatíveis com um modelo teórico previamente desenvolvido.

Um importante trabalho futuro é o desenvolvimento de um esquema automático de ajuste de nível de dificuldade para poder preservar o tempo de criação de blocos dentro de um intervalo previsível. Um outro ponto para desenvolvimento futuro seria um sistema de controle de *forks* com penalização adaptativa que dependesse basicamente do tamanho da diferença entre a cadeia principal e uma cadeia secundária específica.

A construção de um sistema para formação de grupos de validadores ou comitês para controlar a inserção de blocos na *blockchain* poderia trazer benefícios para a rede, como já proposto em vários outros trabalhos relacionados, já que liberaria os demais nós das tarefas de validação. Entretanto, alguns detalhes precisam ser melhor avaliados: a liberação dos nós comuns da tarefa de validação é realmente vantajosa? A escolha dos membros do grupo deve ser equiprovável e com certa rotatividade entre os mesmos? Os benefícios deste tipo de implementação compensariam os custos advindos do mesmo?

Acreditamos também que poderiam ser avaliadas possibilidades de utilização efetiva do $stake$ de cada nó em uma das suas possíveis variantes, mas de forma a não afetar de forma significativa as chances de sucesso no sorteio de um nó. É possível que o uso de $stake$ facilite o aumento da taxa de criação de blocos sem aumento dos *forks*, algo que precisa também de uma investigação mais aprofundada.

Agradecimento: à CAPES pelo financiamento parcial desta pesquisa (Processo núm. 88882.329365).

Referências

Bano, S., Sonnino, A., Al-Bassam, M., Azouvi, S., McCorry, P., Meiklejohn, S., and Danezis, G. (2017). Sok: Consensus in the age of blockchains. *ArXiv e-prints*.

Bashir, I. (2017). *Mastering Blockchain*. Packt Publishing Ltd., 1 edition.

Bentov, I., Lee, C., Mizrahi, A., and Rosenfeld, M. (2014). Proof of activity: Extending bitcoin's proof of work via proof of stake. *SIGMETRICS Perform. Eval. Rev.*, 42(3):34–37.

Bitshare (2016). Delegated proof-of-stake consensus a robust and flexible consensus protocol. `https://bitshares.org/technology/delegated-proof-of-stake-consensus/`. (acessado em 10/09/2017).

Buterin, V. (2014). Slasher: A punitive proof-of-stake algorithm. `https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm/`. (acessado em 27/09/2018).

Buterin, V. and Griffith, V. (2017). Casper the friendly finality gadget. *ArXiv e-prints*.

Community, N. (2014). Nxt whitepaper. [Online]. `https://bravenewcoin.com/assets/Whitepapers/NxtWhitepaper-v122-rev4.pdf`. Whitepaper.

GreenfieldIV, R. (2017). Explaining how proof of stake, proof of work, hashing and blockchain work together. `https://medium.com/@robertgreenfieldiv/explaining-proof-of-stake-f1eae6feb26f`. (acessado em 18/02/2018).

Kiayias, A., Russell, A., David, B., and Oliynykov, R. (2017). Ouroboros: A provably secure proof-of-stake blockchain protocol. *Advances in Cryptology – CRYPTO 2017*.

King, S. (2013). Primecoin: Cryptocurrency with prime number proof-of-work. [Online]. `http://primecoin.io/bin/primecoin-paper.pdf`. Whitepaper.

Lamport, L., Shostak, R., and Pease, M. (1982). The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401.

Maehara, Y., 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 *XVIII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais.*, Natal - RN.

Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system. [Online]. `https://bitcoin.org/bitcoin.pdf`. Whitepaper.

Vasin, P. (2013). Blackcoin's proof-of-stake protocol v2. [Online]. `https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf`. Whitepaper.

Realização

SBC Sociedade Brasileira de Computação

LARC

Organização

.inf INSTITUTO DE INFORMÁTICA UFRGS

Apoio Institucional

RNP 30 anos

Patrocínio

Diamante

CAPES

CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico

cgi.br Comitê Gestor da Internet no Brasil

nic.br Núcleo de Informação e Coordenação do Ponto BR

BEDU TECH

Ouro

SAP

Prata

HUAWEI

DATACOM

acm sigcomm

@dyLNET TELECOM

BRDigital Telecomunicações

Bronze

FAPERGS

Google