AutAvailChain: Disponibilização Segura, Controlada e Automática de Dados IoT usando Corrente de Blocos

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

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

Resumo. *A centralização da confiança utilizada nos sistemas de compartilhamento de dados atuais limita o controle do usuário proprietário sobre os próprios dados. A intervenção individual do proprietário para autorizar o acesso a seus dados a cada demanda dificulta a acessibilidade a dados muito populares. Este artigo propõe AutAvailChain*[1]*, uma arquitetura baseada em corrente de blocos e redes definidas por software para prover o compartilhamento seguro, automático e distribuído de dados IoT. O protótipo desenvolvido utiliza a plataforma Hyperledger Fabric para implementar a corrente de blocos e um contrato inteligente. Os resultados mostram um desempenho satisfatório para atender, de maneira rápida e segura, dezenas de transações por segundo.*

1. Introdução

Soluções clássicas de armazenamento e disponibilização de dados delegam a autoridades centralizadas, como governos e empresas, as tarefas de compartilhamento, de controle de acesso e de garantia da privacidade dos seus dados [Lodderstedt et al. 2013]. Essas autoridades impõem altas taxas e termos que comprometem a privacidade dos dados dos proprietários para fornecer serviços de armazenamento e compartilhamento. Assim, o compartilhamento de dados pessoais com autoridades centralizadas implica na perda de controle e rastreabilidade do usuário sobre os próprios dados. A centralização da confiança em um serviço também cria um ponto único de falhas, tornando esse serviço suscetível a ataques de negação de serviço (*Denial of Service* – DoS). A tecnologia de corrente de blocos (*blockchain*) fornece as propriedades necessárias para garantir a segurança no compartilhamento de dados de forma distribuída e auditável, para que o usuário mantenha o controle sobre os dados.

Uma abordagem para a autorização de acesso a dados privados ou sensíveis é a distribuição de certificados para autenticação criptográfica de acesso. Nesse caso, o proprietário dos dados é responsável por emitir um certificado autorizando o acesso de seus dados privados para outra entidade usuária de dados que os queira ou necessite. Entretanto, em um cenário de dezenas de milhares de usuários de dados que queiram ou necessitem, torna-se inviável a emissão manual de certificados [Paillisse et al. 2019]. Portanto, é necessário um meio de automatizar o compartilhamento de dados e a atualização das permissões de acesso aos dados armazenados. Nesse cenário sem confiança mútua, o uso de contratos inteligentes provê a automatização necessária para receber uma demanda de

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

[1] AutAvailChain: Automatic and Secure Data Availability through Blockchain.

acesso a dados, verificar se as condições impostas por um usuário proprietário de dados são atendidas pelo usuário consumidor de dados e autorizar as permissões de acesso aos dados. Adicionalmente, a tecnologia de redes definidas por *software* (*Software Defined Networking* – SDN) oferece a flexibilidade necessária devido a sua capacidade de programação da rede para a implementação do controle de acesso aos dados armazenados.

Este artigo propõe, desenvolve e avalia uma arquitetura para compartilhamento seguro e automático de dados IoT através da tecnologia de corrente de blocos e de redes definidas por *software* em um cenário multi-domínio. A proposta garante a transparência dos direitos de acesso aos dados, fornecendo ao usuário a rastreabilidade sobre os próprios dados. Para isso, o artigo usa a propriedade da integridade e auditabilidade da corrente de blocos, que armazena as permissões de acesso de cada usuário aos dados IoT. Para limitar os possíveis danos do compartilhamento de dados por parte do servidor com terceiros, a proposta assume que os dados são criptografados antes de serem armazenados. A Internet das coisas gera grande volume de dados, os quais podem ser comercializados entre donos de dispositivos IoT e cientistas de dados que tenham interesse no treino e criação de modelos baseados em dados reais de dispositivos. Assim, a proposta assume que os proprietários de dados esperam ser recompensados por disponibilizá-los. Para atender isso, o artigo apresenta um mercado de dados na corrente de blocos propondo três tipos de transação que permitem que os usuários possam anunciar e adquirir dados emitindo transações. As transações propostas garantem a comercialização efetiva e automática dos dados IoT. Assim, a arquitetura não depende da disponibilidade do usuário para atualizar as permissões de acesso. O cenário da proposta considera grandes centros de dados interconectados através da tecnologia SDN. Um controlador SDN implementa o controle de acesso aos dados. O artigo implementa um protótipo da arquitetura proposta usando a plataforma de código-aberto Hyperledger Fabric [Androulaki et al. 2018] para a corrente de blocos, o controlador Ryu e o emulador de redes Mininet. Um contrato inteligente implementa as lógicas de transação propostas no artigo. Os resultados apresentam um desempenho de vazão de transação comparável a comércios eletrônicos a nível nacional ao controlar o acesso a dados IoT de maneira segura, automática e rápida.

O restante do artigo está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. A Seção 3 apresenta o modelo de atacante considerado para cada entidade da arquitetura proposta. A Seção 4 detalha a arquitetura proposta e apresenta os tipos de transação propostos para um compartilhamento automático e efetivo de dados. A Seção 5 apresenta o desenvolvimento do protótipo e avalia o desempenho da proposta. Os resultados de vazão de transação e tempo de acesso são medidos e analisados para avaliar a proposta. Por fim, a Seção 6 conclui o artigo, finalizando o que foi desenvolvido ao longo deste trabalho e apresentando as direções de trabalhos futuros.

2. Trabalhos Relacionados

O uso de corrente de blocos para solucionar problemas de redes de comunicação tem sido frequente na literatura, tais como atualização segura de dados [Boudguiga et al. 2017], fatiamento seguro de redes virtuais [Rebello et al. 2019b], orquestração segura de redes virtuais [Rebello et al. 2019a], entre outros. A corrente de blocos também tem sido usada para prover controle de acesso a recursos.

Ouaddah *et al.* propõem o uso de contratos inteligentes para a implementação

de políticas de acesso em formato de código e implementação de tokens de autorização, que são assinaturas digitais do proprietário permitindo o acesso a um recurso [Ouaddah et al. 2016]. Na proposta, os tokens são individuais e só podem ser usados uma única vez. O uso de tokens apresenta desvantagens, como o alto tempo para conseguir acesso pela necessidade de consultar o proprietário a cada novo acesso. Pinno *et al.* propõem o ControlChain, uma arquitetura baseada em quatro correntes de blocos para autorização de acessos a dispositivos IoT [Pinno et al. 2019]. Esta arquitetura requer um grande espaço para armazenamento das correntes e pode gerar alto custo em plataformas que cobram por execução de contratos inteligentes, como o Ethereum.

Hu *et al.* propõem uma ferramenta para compartilhamento de dados focada na anonimidade [Hu et al. 2020]. Os autores separam o tempo em épocas e utilizam a corrente de blocos como um ponto de controle (*checkpoint*) das ações de leitura e escrita sobre um objeto em uma época. A proposta garante de forma robusta privacidade, mas requer uma complexidade e latência exagerada para consegui-la. Shafagh *et al.* propõem um sistema de compartilhamento de dados que separa o sistema de armazenamento em um plano de controle e um plano de dados [Shafagh et al. 2017]. O plano de controle contém a corrente de blocos, responsável por armazenar as permissões de acesso, e um sistema de gerenciamento e distribuição de chaves criptográficas. O plano de dados é formado por sistema de armazenamento distribuído, como as tabelas resumo distribuídas (*Distributed Hash Tables* - DHT). O trabalho, no entanto, não é implementado, além de não automatizar o acesso aos usuários.

Pailisse *et al.* propõem uma arquitetura para o controle de acesso entre domínios usando corrente de blocos [Paillisse et al. 2019]. Na arquitetura proposta, os autores utilizam a corrente de blocos para armazenar as permissões que os usuários possuem sobre um recurso e utilizam protocolo de separação localização/ID (*Locator/ID Separation Protocol* – LISP) para efetuar o controle de acesso ao recurso solicitado. As políticas de acesso são atualizadas pela corrente de blocos e armazenadas no plano de controle LISP para que roteadores possam consultar as permissões de acesso. Entretanto, a proposta dos autores não inclui uma recompensa aos proprietários por disponibilizar os dados. O controle e a comercialização dos dados pelos proprietários são propriedades desejáveis para a Internet das coisas. Além disso, os proprietários são os responsáveis por atualizar as políticas de acesso, o que não é uma solução escalável a nível de sistemas IoT.

Truong *et al.* propõem o Sash, um arcabouço para compartilhamento de dados IoT usando corrente de blocos, em que os proprietários dos dados podem comercializar os seus dados [Truong et al. 2019]. Os autores utilizam as propriedades da imutabilidade e da transparência da corrente de blocos para prover a auditabilidade das políticas de acesso. No Sash, os dados são criptografados e armazenados fora da corrente (*off-chain*) e os contratos inteligentes são utilizados para avaliar os pedidos de acesso. Entretanto, os autores deixam em aberto a forma como a distribuição de chaves criptográficas para decriptar os dados adquiridos é realizada, podendo ser um serviço estabelecido pelo próprio proprietário ou delegado para uma autoridade de chaves. O uso de uma autoridade distribuidora de chaves criptográficas apresenta problemas, como a centralização da rede e da confiança na autoridade, enfraquecendo a propriedade de descentralização, que é uma das principais propriedades da corrente de blocos. A proposta deste artigo, por sua vez, é de um sistema distribuído entre domínios em que o controle de acesso é feito de maneira

automática por controladores SDN. Os autores apresentam uma segunda proposta em que os dados não são criptografados antes de serem armazenados e o acesso é atualizado pelo servidor através de uma lista de controle de acesso (*Access Control List* – ACL) guardada na corrente de blocos. Nesse caso, o servidor poderia agir maliciosamente e disponibilizar os dados a terceiros, uma vez que não estão criptografados.

Ao contrário dos artigos citados, este artigo propõe um sistema simples e eficiente que seja seguro, distribuído e automático para um serviço de disponibilização e comercialização segura de dados IoT entre domínios usando redes definidas por software. A simplicidade do sistema está na implementação usando uma corrente de blocos, enquanto a eficiência está no rápido tempo de acesso aos dados e na vazão que o sistema apresenta, atingindo 65 transações por segundo.

3. Modelo de Atacante

O modelo "honesto, mas curioso" (*honest-but-curious*) é indicado na literatura para para modelar servidores de nuvem [Xu et al. 2013, Krämer et al. 2019, Xue et al. 2017], pois captura os ataques mais comuns de vazamento de dados em servidores. Este trabalho, portanto, utiliza o modelo "honesto, mas curioso" para especificar os ataques à nuvem que armazena os dados de um proprietário [Božović et al. 2012]. Neste modelo, a entidade comprometida por um atacante continua a seguir honestamente o protocolo do sistema. O interesse do atacante nesse caso não é comprometer a integridade dos dados ou o comportamento da entidade, mas sim obter e vazar informações sensíveis da organização. No contexto deste artigo, considera-se como ameaças principalmente intrusões ou ataques internos de funcionários, que têm como principal objetivo permanecerem não-descobertas e, portanto, raramente alteram o comportamento da entidade. Ainda, o artigo assume que os dados armazenados são criptografados utilizando a chave pública do proprietário. Assim, o conteúdo dos dados é inacessível ao servidor e possíveis vazamentos causam danos menos significativos.

Para o comportamento das organizações na corrente de blocos e para o proprietário dos dados, este trabalho considera o modelo de atacante Dolev-Yao [Dolev and Yao 1983]. O modelo Dolev-Yao é o que prevê os atacantes mais poderosos em um cenário inseguro [Cervesato 2001]. Neste modelo, o atacante pode ler, enviar e descartar uma transação endereçada à corrente de blocos, ou qualquer pacote da rede. O atacante pode se conectar passivamente à rede e capturar trocas de mensagens ou injetar, reproduzir, filtrar e trocar informações ativamente.

Ataques à corrente de blocos objetivam impedir que uma transação ou bloco legítimo sejam adicionados à corrente de blocos. O protocolo de consenso tolerante a falhas do sistema mitiga esse tipo de ataque, pois exige que o atacante deve controlar uma parcela significativa da rede para afetar o protocolo de consenso. Caso contrário, o emissor da transação/bloco pode verificar a qualquer momento sua presença na corrente de blocos. Os ataques que exigem corrupção ou adulteração de transações são impossíveis quando todas as transações incluem seu *hash* assinado correspondente.

Ataques a vendedores ou a compradores 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. 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 agente que sofreu o ataque pode facilmente substituir os pares de chaves roubados, restabelecendo a segurança e evitando mais danos. Este trabalho não aborda o caso em que um vendedor age de forma deliberadamente maliciosa ao, por exemplo, vender dados inexistentes ou que não correspondem ao informado na transação de venda. Em trabalhos futuros, pretende-se tratar este problema com um sistema de reputação distribuído [Malik et al. 2019].

Ataques à rede representam a tentativa de isolar um único alvo, impedindo assim que vendedores e compradores emitam transações ou que o controlador 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 organizações, clientes e proprietários. Este trabalho assume uma rede pública redundante, como a Internet, que interconecta todos os participantes.

4. O Sistema Proposto

A proposta do artigo consiste em um sistema simples, seguro, eficiente e distribuído para a comercialização automatizada de dados. As principais vantagens de um sistema com essas características são a robustez à ataques de negação de serviço e a baixa latência para liberação do acesso aos dados comprados. A robustez aos ataques de negação de serviço é possível porque o sistema é distribuído, sendo assim mesmo que um nó da rede seja afetado, outros continuam a fornecer o mesmo serviço. A baixa latência é implementada removendo a necessidade do procedimento com a intervenção do usuário vendedor para a liberação do acesso dos dados do servidor de armazenamento pelo vendedor devido a automatização do controle de acesso.

A automatização é implementada através do uso de contratos inteligentes e redes definidas por *software*. O cenário da proposta considera o compartilhamento entre grandes centros de dados interconectados através da tecnologia de redes definidas por *software*, como mostra a Figura 1. Além disso, o cenário considera que os dados fo-

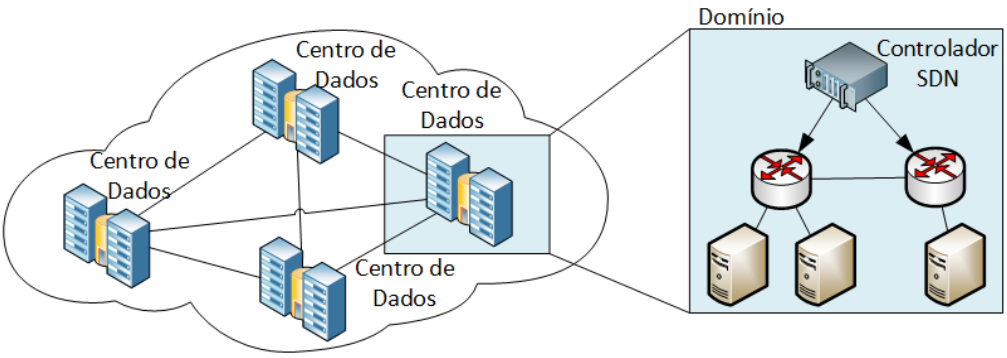

Figura 1. O cenário da proposta considera grandes centros de dados que possuem a tecnologia redes definidas por software (SDN). O protótipo considera um único controlador SDN e um domínio por centro de dados.

ram criptografados e estão armazenados em um servidor de armazenamento. O uso de redes definidas por *software* simplifica o gerenciamento de rede ao introduzir a programabilidade por desacoplar o plano de controle do plano de dados [Afolabi et al. 2018]. Os contratos inteligentes automatizam o processo de venda, pois ao comprar o dado de um servidor de armazenamento, o contrato inteligente associa a transação de anúncio à transação de compra, sem a necessidade de verificação do vendedor. Como as transações de compra e venda são associadas, o controlador SDN libera automaticamente o acesso aos dados através da leitura das transações da corrente de blocos.

O objetivo do sistema proposto é registrar os pedidos de acesso aos dados e autorizar o acesso automaticamente caso o usuário cumpra os requisitos estabelecidos. Essa transparência permite que o usuário tenha maior controle sobre os próprios dados. Para isso, o sistema usa a propriedade da auditabilidade da corrente de blocos para registrar quem possui acesso a um determinado recurso. A arquitetura do sistema é dividida em dois módulos, como mostra a Figura 2: i) módulo de corrente de blocos, armazena um registro das permissões de acesso de um usuário a um recurso; ii) módulo de controle de acesso, controla o acesso aos dados a partir de leituras de transações da corrente de blocos.

O Módulo de Corrente de Blocos registra os anúncios e demandas de aquisição de dados no sistema e é composto pela corrente de blocos [Nakamoto 2008], um registro imutável do histórico de transações emitidas no sistema, e o estado global. O estado global é um banco de dados com todas as transações feitas no sistema, funcionando como uma versão resumida da corrente de blocos, indexando as transações para um processo de busca mais rápido. Os usuários do sistema, vendedores ou compradores de dados, aces-

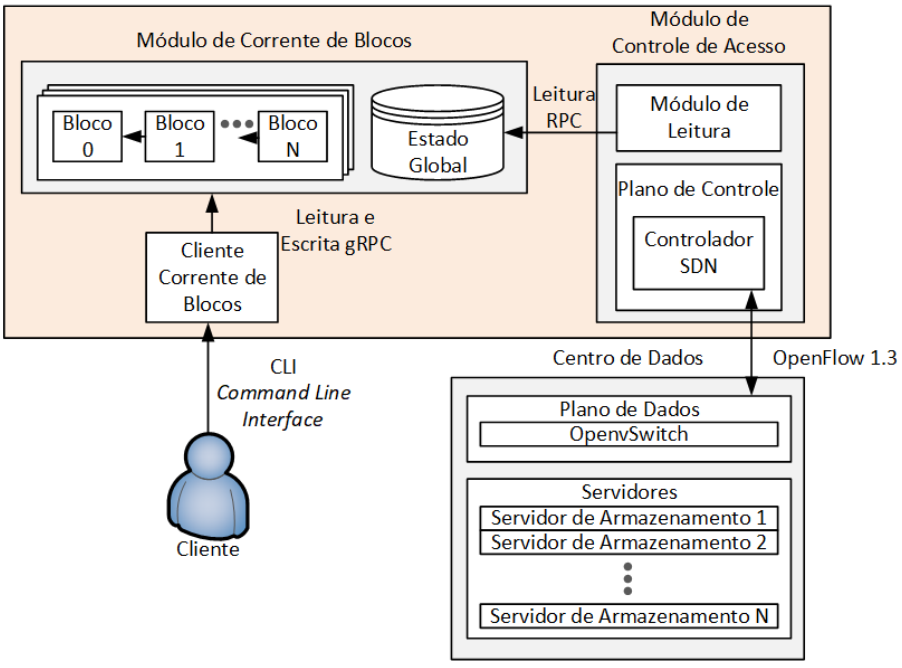

Figura 2. Arquitetura do sistema proposto. O cliente comprador (vendedor) anuncia dados próprios (adquire dados) se comunicando com a corrente de blocos. O Módulo de Controle de Acesso lê as transações de compra de dados e muda as permissões de acesso do comprador sobre o recurso.

sam o cliente de corrente de blocos através da interface de linha de comando (*Command Line Interface* – CLI) para enviar ou obter informações sobre os conjuntos de dados disponíveis. A comunicação entre o cliente corrente de blocos e o módulo de corrente de blocos é feita através de chamadas de procedimento remoto de propósito geral (*general-purpose Remote Procedure Call* - gRPC).

O Módulo de Controle de Acesso gerencia as permissões de acesso a um recurso e é composto pelo plano de controle e o módulo de leitura. O módulo de leitura é responsável por ler a corrente de blocos através de chamadas de procedimentos remotos (RPC - *Remote Procedure Call*). As informações adquiridas pelo módulo de leitura são fornecidas para o plano de controle atualizar as permissões de acesso atuais. A atualização permite que os usuários possam conectar ao servidor de armazenamento dos dados que foram comprados através da corrente de blocos. O controlador armazena, baseado nos endereços IP de origem e destino, quais usuários têm acesso ao servidor de armazenamento. Caso um usuário sem permissão tente estabelecer conexão ao servidor, o controlador verifica na corrente de blocos através do módulo de leitura se as permissões concedidas foram atualizadas. Caso estejam desatualizadas, o controlador atualiza a versão local das permissões dos servidores, atendendo as transações a serem processadas. Se a versão for atual ou se o usuário não possuir permissões após a atualização, o controlador insere na sua tabela uma ação de descartar novos pacotes recebidos do usuário.

A corrente de blocos na arquitetura proposta funciona como um mercado de dados, em que os proprietários e interessados podem anunciar e procurar por dados IoT. Para isso, o sistema define dois tipos de usuários: vendedores e compradores. Vendedores detêm os direitos e podem comercializar os seus dados, pedindo uma recompensa por eles. Compradores interessados podem procurar através de buscas (*query*) por anúncios da corrente de blocos para adquirir dados. Apesar da separação em dois tipos de usuário, um participante pode assumir os dois papéis, tanto anunciando seus dados, como comprando dados de outros proprietários. O uso de contratos inteligentes permite a implementação de um sistema de tokens que funcionam como ativos de uma organização. As organizações podem adquirir dados mediante pagamento de uma quantidade de tokens e acordar fora da corrente (*off-chain*) um pagamento equivalente em moeda [Truong et al. 2019].

A arquitetura define três tipos de transação para a comercialização dos dados: i) transação de anúncio; ii) transação de compra e iii) transação de resposta. Transações de anúncio são emitidas a partir de um proprietário interessado em disponibilizar e comercializar seus dados. O proprietário submete os dados a serem comercializados em um servidor de armazenamento capaz de suportar e processar grande quantidade de dados [Truong et al. 2019] e emite uma transação de anúncio assinada. As transações são assinadas pelo emissor usando criptografia assimétrica para garantir a autenticidade e a integridade da transação. A transação de anúncio deve conter uma breve descrição dos tipos de dados sendo ofertados, por exemplo dados de sensores médicos, e o preço para os dados serem adquiridos. A transação de anúncio TX_{ad} é definida como:

$$TX_{ad} = [TXID_{ad}|Sig_{ow}|IP_{sp}|Pr|Org_{ow}|DTD], \quad (1)$$

onde $TXID$ é o identificador da transação, Sig_{ow} é a assinatura digital do proprietário, IP_{sp} é o endereço IP do servidor de armazenamento, Pr é o preço dos dados, Org_{ow} é a organização proprietária dos dados e DTD é um campo contendo uma descrição dos

tipos de dados que o proprietário está anunciando[2].

Transações de compra são emitidas a partir de interessados em adquirir dados anunciados na corrente de blocos. Os clientes compradores procuram por dados na corrente de blocos a partir de buscas por transações de anúncio. O sistema permite buscas específicas, como buscas pelo tipo de dado, ou buscas por todas as transações de anúncio. O interessado em comprar os dados anunciados deve emitir uma transação de compra referenciando o identificador da transação de anúncio correspondente e informando o endereço IP do comprador para que o controlador possa liberar o acesso. A transação de compra também deve incluir o valor a ser pago pelos dados. Caso o valor oferecido seja menor que o pedido na transação de anúncio correspondente ou o comprador não tenha saldo suficiente para a compra, a transação não é válida. A transação de compra TX_{buy} é definida como:

$$TX_{buy} = [TXID_{buy}|TXID_{ad}|Sig_{buyer}|IP_{buyer}|Pay|Org_{src}], \quad (2)$$

onde $TXID$ é o identificador da transação, $TXID_{ad}$ é o identificador da transação de anúncio correspondente, Sig_{buyer} é a assinatura digital do comprador, IP_{buyer} é o endereço IP do comprador, Pay é o valor a ser pago pelos dados e Org_{src} é a organização que está efetuando o pagamento.

Transações de resposta são emitidas pelo proprietário automaticamente após um de seus dados receber uma transação de compra. A transação de resposta envia ao comprador a chave que decripta os dados. A chave é criptografada usando a chave pública do comprador para garantir que o servidor de armazenamento não tenha acesso aos dados decriptados e compartilhe com terceiros. A transação de resposta TX_{res} é definida como:

$$TX_{res} = [TXID_{res}|TXID_b|OK_{Enc\{PK_b\}}], \quad (3)$$

onde $TXID_{res}$ é o identificador da transação, $TXID_b$ é o identificador da transação de compra correspondente e $OK_{Enc\{PK_b\}}$ é a chave que decripta o objeto criptografada com a chave pública do comprador.

As transações são enviadas para execução dentro do módulo de corrente de blocos antes de serem adicionadas a um bloco. A execução, ilustrada no diagrama de sequência *Unified Modeling Language* (UML) da Figura 3, busca verificar se a transação é válida ou não. Esse processo consiste na verificação pelo contrato inteligente se: i) o valor a ser pago Pay é maior ou igual ao preço dos dados Pr pedido pelo proprietário; ii) se a organização possui tokens suficientes para efetuar a compra. As transações que atendem os requisitos anteriores são validadas e as que não atendem são marcadas como inválidas. Caso a transação seja válida, o valor pago de tokens é descontado de Org_{src} e enviado para Org_{ow} que pode usar os tokens adquiridos para comprar dados de outras organizações.

5. Desenvolvimento do Protótipo e Resultados

Um protótipo da proposta foi desenvolvido[3] utilizando a plataforma de código aberto Hyperledger Fabric 2.0 [Androulaki et al. 2018] para a corrente de blocos, o emu-

[2]A abreviação *ad* é de *advertisement*, *ow* é de *owner*, *sp* é de *service provider* e DTD é de *data type description*.

[3]A implementação está disponível em https://github.com/GTA-UFRJ-team/blockchain-marketplace

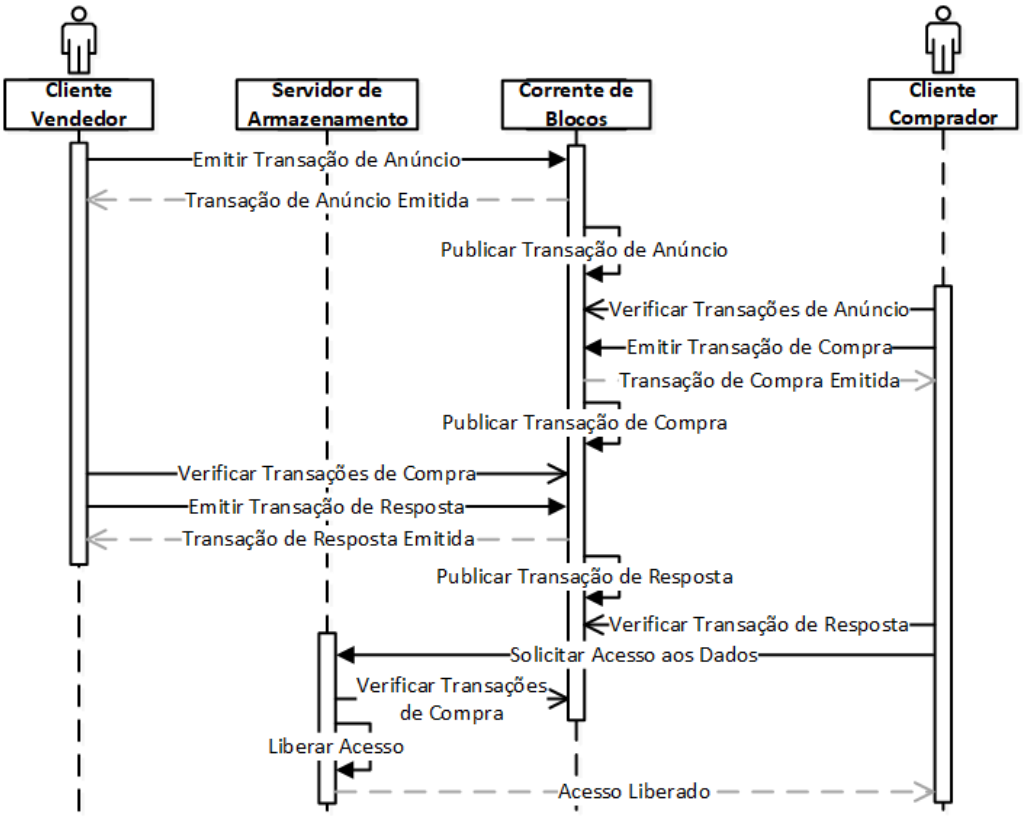

Figura 3. Diagrama de sequência do sistema proposto representando uma venda de dados entre duas organizações. Ao final do processo descrito no diagrama, a venda dos dados está registrada publicamente na corrente de blocos e o cliente comprador possui acesso aos dados vendidos.

lador de rede Mininet [Lantz et al. 2010] e o controlador Ryu[4]. O Hyperledger Fabric é uma plataforma para a implementação de corrente de blocos permissionada entre organizações. A plataforma Hyperledger está disponível gratuitamente, com boa documentação, facilmente programável e sem nenhuma moeda e custo associado. O aspecto organizacional do Hyperledger se ajusta ao cenário multi-domínio da proposta, em que empresas comercializam os dados. Um computador Intel i7-2600 CPU 3.40 GHz com 32 GB RAM e 8 núcleos de processamento cria os nós da rede de corrente de blocos como contêineres Docker. Os resultados apresentam intervalo de confiança de 95%. Estipulou-se um número de transações por bloco da corrente igual a 100, como usado em trabalhos anteriores sobre avaliação de desempenho da plataforma Hyperledger Fabric [Gorenflo et al. 2019, Thakkar et al. 2018]. Os nós na arquitetura do Fabric, representam as entidades que participam da corrente de blocos. A arquitetura do Hyperledger Fabric apresenta três tipos de nós: clientes, pares e ordenadores. Na arquitetura do Hyperledger Fabric, clientes representam usuários e emitem transações que precisam ser executadas por pares endossadores (*endorsers*), que são responsáveis por verificar a validade da transação. Caso a transação seja válida, o cliente recebe as transações assinadas pelos pares endossadores e envia a transação com as assinaturas dos endossadores para nós ordenadores, que executam um protocolo de consenso e ordenam as

[4]Disponível em https://osrg.github.io/ryu/

transações em um bloco. O Fabric utiliza um tipo especial de nó chamado par âncora para divulgar os blocos ordenados. O protótipo utilizado nos experimentos descritos em seguida foi configurado usando cinco nós ordenadores e o protocolo de consenso Raft [Ongaro and Ousterhout 2014]. A Figura 4 mostra a arquitetura do Hyperledger Fabric com o cenário utilizado.

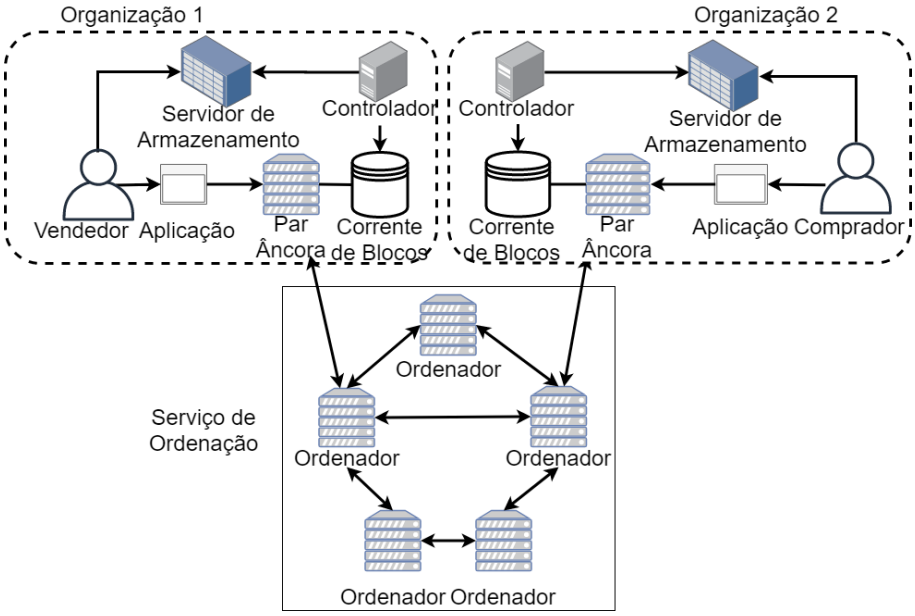

Figura 4. Arquitetura de uma corrente de blocos permissionada no Hyperledger Fabric com a adição do controlador e do servidor do cenário proposto. Usuários, vendedores e compradores de cada organização utilizam aplicações para emitir transações que são ordenadas em um bloco pelo serviço de ordenação usando um protocolo de consenso. Os blocos ordenados são enviados aos pares âncoras para atualização da corrente de blocos. O controlador lê as transações e controla o acesso aos dados no servidor de armazenamento. Os usuários usam o servidor de armazenamento para armazenar os dados anunciados.

Um contrato inteligente escrito como um código autoexecutável em Go é executado em todos os pares, eliminando uma entidade centralizada de confiança e implementando as lógicas de transação[5] descritas na seção anterior, além de um sistema de tokens e contas de organizações. Esses tokens funcionam como moedas que as organizações podem usar para a comercialização de dados. O valor real desses tokens pode ser discutido fora da corrente (*off-chain*) entre as organizações. O contrato implementa uma fila de transações pendentes, que armazena *strings* em formato JSON contendo o endereço IP do comprador IP_{buyer}, o endereço IP de onde os dados estão armazenados IP_{sp} e o identificador da transação de compra $TXID_{buy}$. A cada TX_{buy} recebido, o contrato atualiza a fila, adicionando uma transação pendente a ser processada pelo controlador de rede definida por software. A implementação da fila é um artifício usado para diminuir o tempo que o módulo de leitura leva para obter as informações que atualizam as permissões de acesso, evitando assim a leitura completa da corrente de blocos. Ao receber um pacote de um endereço IP desconhecido, o controlador chama uma função do módulo de leitura para coletar as *strings* armazenadas na fila e atualizar as permissões de acesso.

[5]A transação de resposta TX_{res} será implementada em trabalhos futuros.

O módulo de leitura foi implementado em Python 2.7 para facilitar a integração com o controlador, que também é escrito em Python. As permissões de acesso de um usuário a um servidor são armazenadas em uma estrutura chave-valor no controlador, em que a chave é o endereço IP do servidor de armazenamento e valor é uma lista que armazena os endereços IP que podem acessar o servidor de armazenamento correspondente à chave. O controlador verifica se o IP origem do pacote está na lista correspondente ao IP destino.

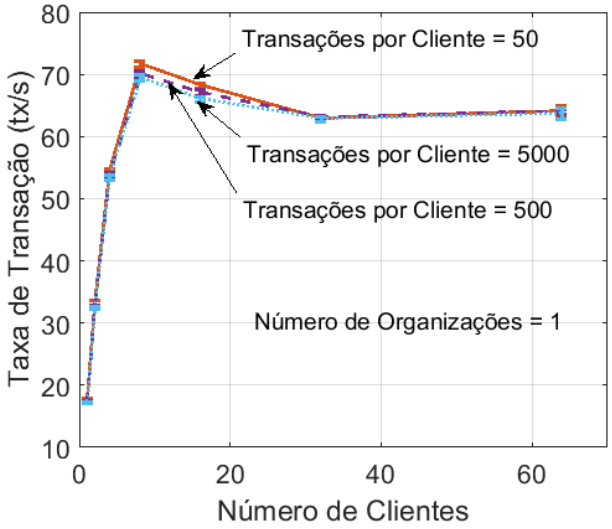

Figura 5. Resultado em que o número de clientes cresce em uma única organização. Cada cliente emite um número de transações de anúncio e o tempo decorrido para a emissão é medido para o cálculo da vazão.

1) Vazão de Transação por Número de Clientes em Uma Organização: o primeiro experimento consiste em aumentar o número de clientes em uma única organização e medir a vazão de transação. Esse experimento simula um cenário em que uma organização domina as operações de comercialização dos dados na corrente de blocos. No experimento, a rede de corrente de blocos é formada por duas organizações com dois pares cada. O experimento verifica a vazão de transação enquanto aumenta o número de clientes que emitem 50, 500 e 5000 transações aos pares endossadores concorrentemente. A taxa de transação corresponde a razão entre o número total de transações emitidas e o tempo decorrido para todos os clientes emitirem todas as transações. A Figura 5 mostra que a taxa de transação cresce inicialmente até estabilizar com o aumento do número de clientes. Isso acontece porque um baixo número de clientes emitindo transações limita a vazão na taxa de transação que esses clientes emitem. A vazão se estabiliza por volta de 65 transações por segundo. Esse resultado comprova que o serviço proposto pode ser aplicado de modo satisfatório a nível nacional, uma vez que o MercadoPago tem um valor médio de 30 transações por segundo[6].

2) Vazão de Transação por Número de Clientes em Diversas Organizações: o segundo experimento consiste em aumentar o número de clientes proporcionalmente

[6]MercadoPago apresentou 227 milhões de vendas no terceiro trimestre de 2019. Disponível em: https://ideias.mercadolivre.com.br/sobre-mercado-livre/mercado-livre-cresce-368-em-vendas-e-atinge-us-76-bilhoes-em-volume-de-pagamentos-com-mercado-pago-no-3o-tri/

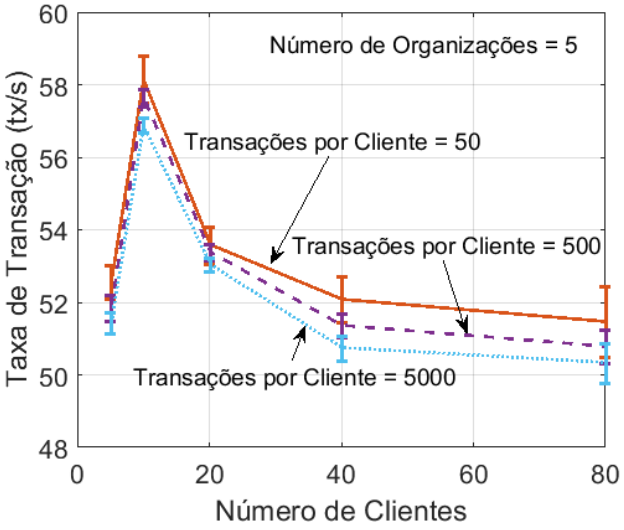

Figura 6. Resultado em que o número de clientes cresce com valores iguais nas cinco organizações. Cada cliente emite um número transações e o tempo decorrido para a emissão é medido para o cálculo da vazão.

entre organizações e medir a vazão de transações. O experimento simula o cenário em que nenhuma organização difere majoritariamente na quantidade de operações de comercialização na corrente de blocos. Neste experimento, a corrente de blocos no Hyperledger Fabric é formada por cinco organizações e o número de clientes em cada organização é aumentado igualmente, de 1 até 16 clientes por organização. A Figura 6 mostra a vazão de transação em relação ao número total de clientes na rede. Assim como no primeiro experimento, a vazão também estabiliza com o aumento do número de clientes. A taxa de transação se estabiliza por volta de 55 transações por segundo. Esse resultado comprova que, mesmo com aumento do número de organizações, a taxa de transação por segundo continua com valores satisfatórios a nível nacional. A taxa de transação por segundo deve aumentar quando os contêineres estiverem distribuídos em mais de uma máquina.

3) Tempo de Acesso aos Dados: o terceiro experimento tem como objetivo avaliar o tempo de acesso aos dados após a confirmação da compra na corrente de blocos. O experimento consiste em criar uma rede Mininet com três hospedeiros (*hosts*), um comutador (*switch*) e um controlador SDN. O hospedeiro h_1 simula um servidor que armazena os dados à venda que foram anunciados na corrente de blocos. Um cliente comprador emite uma transação de compra TX_{buy} passando o endereço IP do hospedeiro h_2, que faz uma requisição a h_1 e espera a resposta. O tempo da primeira requisição é maior porque o controlador verifica e processa as transações pendentes da fila. A média do tempo de resposta obtida é de 0,473 segundos. Esse resultado comprova que o tempo de acesso aos dados adquiridos é extremamente rápido e imperceptível, atendendo satisfatoriamente a interação com o cliente comprador.

6. Conclusão

O artigo apresentou uma arquitetura baseada em redes definidas por *software* e corrente de blocos que permite a comercialização de dados de forma segura, automática

e distribuída entre usuários do sistema. Um contrato inteligente foi concebido para controlar de forma segura e automática a negociação e comercialização entre o proprietário vendedor e o comprador dos dados. Um protótipo que roda na plataforma Hyperledger foi desenvolvido. Os resultados comprovam que a implantação do sistema proposto atende a demanda no cenário nacional, alcançando acima de 50 transações por segundo mesmo quando há um grande número de usuários emitindo transações simultâneas com um servidor físico de teste. A extensão do cenário de teste para múltiplos servidores físicos e com maior capacidade de processamento pode aumentar de modo significativo a taxa de transação por segundo.

Como trabalhos futuros é prevista a implementação de um sistema de reputação para evitar comportamentos maliciosos como a venda de dados corrompidos ou exclusão dos dados do servidor após a compra e a implementação em um aglomerado de servidores.

Referências

Afolabi, I., Taleb, T., Samdanis, K., Ksentini, A., and Flinck, H. (2018). Network slicing and softwarization: A survey on principles, enabling technologies, and solutions. *IEEE Communications Surveys Tutorials*, 20(3):2429–2453.

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.

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.

Božović, V., Socek, D., Steinwandt, R., and Villányi, V. I. (2012). Multi-authority attribute-based encryption with honest-but-curious central authority. *International Journal of Computer Mathematics*, 89(3):268–283.

Cervesato, I. (2001). The Dolev-Yao intruder is the most powerful attacker. In *16th Annual Symposium on Logic in Computer Science—LICS*, volume 1.

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. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 455–463.

Hu, Y., Kumar, S., and Popa, R. A. (2020). Ghostor: Toward a secure data-sharing system from decentralized trust. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pages 851–877, Santa Clara, CA. USENIX.

Krämer, M., Frese, S., and Kuijper, A. (2019). Implementing secure applications in smart city clouds using microservices. *Future Generation Computer Systems*, 99:308–320.

Lantz, B., Heller, B., and McKeown, N. (2010). A network in a laptop: Rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, New York, NY, USA.

Lodderstedt, E., McGloin, M., and Hunt, P. (2013). OAuth 2.0 threat model and security considerations. IETF. RFC 6819. Disponível em http://www.rfc-editor.org/rfc/rfc6819.txt. Acessado em 15 de abril de 2020.

Malik, S., Dedeoglu, V., Kanhere, S. S., and Jurdak, R. (2019). TrustChain: Trust management in blockchain and IoT supported supply chains. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 184–193.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Disponível em https://bitcoin.org/bitcoin.pdf. Acessado em 15 de abril de 2020.

Ongaro, D. and Ousterhout, J. (2014). In search of an understandable consensus algorithm. In *2014 USENIX Annual Technical Conference (USENIX ATC 14)*, pages 305–319, Philadelphia, PA. USENIX.

Ouaddah, A., Abou Elkalam, A., and Ait Ouahman, A. (2016). FairAccess: a new blockchain-based access control framework for the internet of things. *Security and Communication Networks*, 9(18):5943–5964.

Paillisse, J., Subira, J., Lopez, A., Rodriguez-Natal, A., Ermagan, V., Maino, F., and Cabellos, A. (2019). Distributed access control with blockchain. In *IEEE International Conference on Communications - ICC 2019*, pages 1–6.

Pinno, O. J. A., Grégio, A. R. A., and De Bona, L. C. (2019). ControlChain: A new stage on the iot access control authorization. *Concurrency and Computation: Practice and Experience*. e5238 cpe.5238.

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

Rebello, G. A. F., Camilo, G. F., C. Silva, L. G., B. Guimarães, L. C., C. de Souza, L. A., Alvarenga, I. D., and Duarte, O. C. M. B. (2019b). Providing a sliced, secure, and isolated software infrastructure of virtual functions through blockchain technology. In *HPSR 2019*, pages 1–6.

Shafagh, H., Burkhalter, L., Hithnawi, A., and Duquennoy, S. (2017). Towards blockchain-based auditable storage and sharing of iot data. In *Cloud Computing Security Workshop*, ACM CCSW '17, page 45–50.

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

Truong, H. T. T., Almeida, M., Karame, G., and Soriente, C. (2019). Towards secure and decentralized sharing of IoT data. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 176–183.

Xu, J., Chang, E.-C., and Zhou, J. (2013). Weak leakage-resilient client-side deduplication of encrypted data in cloud storage. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pages 195–206.

Xue, K., Xue, Y., Hong, J., Li, W., Yue, H., Wei, D. S., and Hong, P. (2017). RAAC: Robust and auditable access control with multiple attribute authorities for public cloud storage. *IEEE Transactions on Information Forensics and Security*, 12(4):953–967.