Um Sistema Seguro e Distribuído para o Provisionamento de Funções Virtuais de Rede como Serviço através de Corrente de Blocos

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

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

Resumo. *A virtualização nas redes de próxima geração, como 5G e 6G, se servem de modelos de provisionamento de serviços em cenários multi-domínios e multi-inquilinos. Nestes cenários, a orquestração de funções virtuais de rede (Virtual Network Function - NFV) e o cumprimento de acordo de níveis de serviço (Service Level Agreement - SLA) tornam-se susceptíveis a ameaças de segurança, uma vez que não há confiança entre os pares. Este artigo propõe um sistema baseado em corrente de blocos para o provisionamento ágil, seguro e distribuído de funções virtuais de rede em cenários de múltiplos domínios administrativos. A proposta utiliza contratos inteligentes para atender de maneira automática todas as etapas do ciclo de vida do gerenciamento de um acordo de nível de serviço. Os resultados da análise de desempenho de um protótipo desenvolvido mostram que o sistema garante o provisionamento seguro e ágil de VNFs, atendendo a centenas de requisições de criação de fatias por segundo.*

1. Introdução

As redes móveis de próxima geração apresentam modelos de conectividade para atender demandas específicas de diversos usuários. A tecnologia de virtualização de funções de rede (*Network Function Virtualization* - NFV) é fundamental para a realização desses modelos, já que permite o encadeamento flexível de funções virtuais de rede que atendam a requisitos de qualidade de serviço específicos às aplicações utilizadas pelos clientes. Apesar de permitir o provisionamento de serviços de maneira ágil e flexível, o encadeamento de funções de serviço (*Service Function Chaining* - SFC) apresenta diversos desafios de segurança [Medhat et al. 2017]. Cadeias de funções de serviço podem incluir funções virtuais de rede (*Virtual Network Functions* - VNFs) instanciadas em provedores de serviço concorrentes, o que dificulta a responsabilização e punição de falhas de operação e de comportamento malicioso entre os múltiplos domínios administrativos. Assim, é necessário garantir o provisionamento seguro de cadeias de serviço, identificando corretamente as falhas e o comportamento malicioso na rede. Nestes cenários em que não há confiança mútua, a corrente de blocos (*blockchain*) pode prover um registro confiável das operações de maneira imutável e distribuída, provendo transparência aos usuários e permitindo a correta identificação de comportamento malicioso [Pinno et al. 2017, Michelin et al. 2018, de Oliveira et al. 2020, Rebello et al. 2019b].

O cliente e o provedor de serviço estabelecem um acordo de nível de serviço (*Service Level Agreement* - SLA), que define níveis de desempenho que o provedor de serviço

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

deve entregar aos inquilinos. Este acordo de nível de serviço é fundamental para garantir ao usuário que o provedor de serviço aprovisionou corretamente os recursos e se oferece os serviços contratados. Entretanto, os inquilinos não possuem visibilidade das operações de gerenciamento de rede e não conseguem verificar e validar o serviço que está sendo oferecido, o que dificulta sobremaneira o processo de ressarcimento financeiro por descumprimento do acordo e torna o acordo desigual [Balachandran et al. 2020]. O acordo também pode conter restrições de uso do serviço e, neste caso, é necessário verificar se o usuário age conforme foi determinado no contrato. O uso de contratos inteligentes provê a automação e transparência necessária para a correta e confiável verificação e validação dos acordos de serviço de maneira distribuída.

Este artigo propõe um sistema baseado em corrente de blocos para garantir a segurança na orquestração de fatias de rede, através do registro imutável e transparente das operações. As contribuições do artigo são as seguintes:

- a concepção de um sistema seguro e rápido para prover transparência, não-repúdio e rastreabilidade das operações de orquestração de fatias de rede. O sistema permite a correta identificação e responsabilização de erros, falhas, mau uso e comportamento malicioso na rede. A proposta utiliza contratos inteligente para automatizar a punição de comportamento malicioso na rede;
- a proposta de um modelo que cumpre de maneira eficiente todos os requisitos do ciclo de gerenciamento de um SLA e garante a confidencialidade na criação da fatia de rede sem perder a transparência para os usuários envolvidos;
- o desenvolvimento e a implementação de um protótipo do sistema proposto através de contratos inteligentes desenvolvidos na plataforma utilizando Hyperledger Fabric. Os resultados da avaliação de desempenho do protótipo implementado mostram que o sistema atende de maneira ágil as demandas de inquilinos, atingindo em torno de 115 requisições por segundo.

O restante do artigo está organizado da seguinte forma. A Seção 2 apresenta o ciclo de gerenciamento de um SLA e detalha o modelo de atacante considerado para o sistema. A Seção 3 apresenta o sistema proposto, detalhando as etapas, os tipos de transação e a troca de mensagens para o provisionamento seguro de fatias de rede. A Seção 4 descreve e avalia o desempenho de um protótipo desenvolvido do sistema proposto. A Seção 5 apresenta o estado da arte e discute os trabalhos relacionados. Por fim, a Seção 6 conclui o artigo e apresenta direções para trabalhos futuros.

2. O Ciclo de Vida do Gerenciamento de um Acordo de Níveis de Serviço e o Modelo de Atacante

Um acordo de nível de serviço é um contrato estabelecido entre um inquilino e um provedor de serviço definindo o nível esperado que o provedor deve entregar em um serviço contratado pelo inquilino. Enquanto o acordo de nível de serviço (SLA) é apenas um contrato entre o usuário e o provedor de serviço, o papel do gerenciamento do acordo de nível de serviço é muito mais complexo, pois envolve o monitoramento dos serviços providos. Este artigo assume que o monitoramento dos serviços providos é realizado e que as comprovações da qualidade do serviços medida são registradas de forma transparente e imutável em uma corrente de blocos. Além disso, o artigo considera o ciclo de vida de gerenciamento de um acordo de nível de serviço proposto pela Sun

Microsystem Internet Data Center Group [Wu and Buyya 2010]. O ciclo de vida é divido em seis etapas [Maarouf et al. 2015]:

1. **Descoberta de provedor de serviço**: o inquilino escolhe o provedor de serviço responsável por fornecer a infraestrutura para executar os serviços requiridos;
2. **Definição do SLA**: o provedor de serviço e o inquilino acordam os parâmetros de qualidade de serviço e definem as punições em caso de descumprimento dos níveis de serviço;
3. **Estabelecimento do acordo**: as partes envolvidas estabelecem um modelo (*template*) definindo os níveis discutidos durante a etapa de definição do SLA. O provedor de serviço e o inquilino assinam o acordo, validando os níveis e punições;
4. **Monitoramento de violações do SLA**: o serviço entregue pelo provedor de serviço é testado para verificar o cumprimento dos níveis definidos na segunda etapa e acordado na terceira etapa;
5. **Terminação de SLA**: o SLA expira devido a tempo decorrido previamente acordado ou devido a infrações no cumprimento de contrato;
6. **Aplicação de penalidades do SLA**: o provedor é punido de acordo com as cláusulas definidas no contrato, caso descumpra os níveis acordados.

As propriedades intrínsecas à corrente de blocos garantem a segurança e automação do sistema, pois se resumem a aplicações distribuídas que executam contratos inteligentes. Os ataques podem ter como alvo inquilinos, funções virtuais de rede (VNFs), a corrente de blocos em si e a rede. O modelo de atacante é similar ao definido por 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. Alvarenga *et al.* descrevem os ataques à corrente de blocos, a inquilinos e às funções virtuais de de rede e, por fim os ataques à rede, que são repetidos abaixo [Alvarenga et al. 2018]:

Os ataques à corrente de blocos impedem que uma transação ou um bloco legítimo sejam adicionados à corrente de blocos. Protocolos de consenso mitigam este tipo de ataque ao exigir que as informações sejam assinadas e difundidas para todos os participantes seguindo um algoritmo tolerante a comportamento malicioso. As transações possuem um *hash* assinado o que evita ataques que corrompem ou adulteram as transações.

Ataques a inquilinos ou VNFs com tentativa de obtenção de informações de configuração ou personificação do alvo, não são possíveis porque todas as transações são assinadas e as informações confidenciais são encriptadas. 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 é 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.

Os ataques à rede isolam 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. 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 arquitetura proposta foca na prevenção dos ataques à corrente de blocos e transações.

3. O Sistema Proposto

O objetivo do sistema proposto consiste em prover a transparência e o não repúdio das operações de orquestração multidomínio de cadeias de serviço ao registrar de maneira imutável e distribuída em uma corrente de blocos as etapas do gerenciamento de um acordo de nível de serviço (SLA). Assim, inquilinos podem facilmente detectar comportamento malicioso e não cumprimento de contratos, cobrando ressarcimento e indenizações dos provedores de serviço, que não podem negar a responsabilidade pelos erros e falhas cometidas. O sistema é completamente automatizado através de contratos inteligentes, que são aplicações distribuídas e autoexecutáveis, que registram todo o ciclo de vida do contrato de serviço estabelecido seguindo regras acordadas entre as partes envolvidas e geram multas compensatórias e punições para o infrator do contrato.

O sistema atende dois tipos de usuários na corrente de blocos: provedores de serviço e inquilinos. Os provedores de serviço (*Service Providers* - SP) criam cadeias de funções de redes virtuais para atender às demandas dos usuários inquilinos. As funções de redes virtuais são selecionadas através de pregões eletrônicos e incorporadas às cadeias através da tecnologia de virtualização de funções de redes. Os inquilinos são clientes que buscam provedores de serviço para receber os serviços providos pelo encadeamento de funções virtuais de rede. Cada função de rede virtual fornece um serviço parcial específico enquanto a cadeia completa de funções de redes oferece o serviço fim-a-fim que atende a uma demanda correspondente a um conjunto de indicadores-chave de desempenho (*Key Performance Indicators* - KPIs). A cadeia completa de funções virtuais de rede forma uma rede virtual específica, denominada fatia de rede, construída em cima de uma infraestrutura física de uso geral. A arquitetura para prover as diferentes fatias de rede deve ser flexível e ágil para atender a milhares de usuários que demandam serviços específicos sob demanda.

A arquitetura do sistema proposto, mostrada na Figura 1, garante a segurança na criação de fatias de rede utilizando dois componentes: orquestrador multidomínio (*Multi-domain Orquestrator* - MdO) e aplicativo distribuído (*Distributed Application* - DApp). O orquestrador fornece o ciclo de vida de gerenciamento de um serviço de rede composto e gerenciado em vários domínios administrativos [Rosa and Rothenberg 2018]. Inquilinos e provedores de serviço demandam, negociam e criam fatias de rede de maneira segura utilizando o orquestrador multidomínio. O orquestrador utiliza a infraestrutura dos provedores de serviço para criar as fatias de rede que atendam às demandas dos inquilinos. A arquitetura proposta inclui um monitor de referência que verifica os indicadores-chave de desempenho para validar o cumprimento dos acordos de níveis de serviço. O aplicativo distribuído executa contratos inteligentes que validam de forma automática o cumprimento dos acordos de níveis de serviço (SLA) estabelecido. Assim, o aplicativo distribuído registra todas as operações do ciclo de vida de gerenciamento de um acordo de nível de serviço de maneira imutável e distribuída na corrente de blocos, provendo transparência, não-repúdio e rastreabilidade.

Usuários inquilinos utilizam o aplicativo distribuído para encontrar provedores de serviço interessados em instanciar o serviço de redes demandado. Para encontrar os provedores, os inquilinos registram os indicadores-chave de desempenho de maneira pública na corrente como uma forma de anúncio de pregão. Os provedores de serviço utilizam técnicas de mapeamento transformando os indicadores-chave de desempenho em cadeias

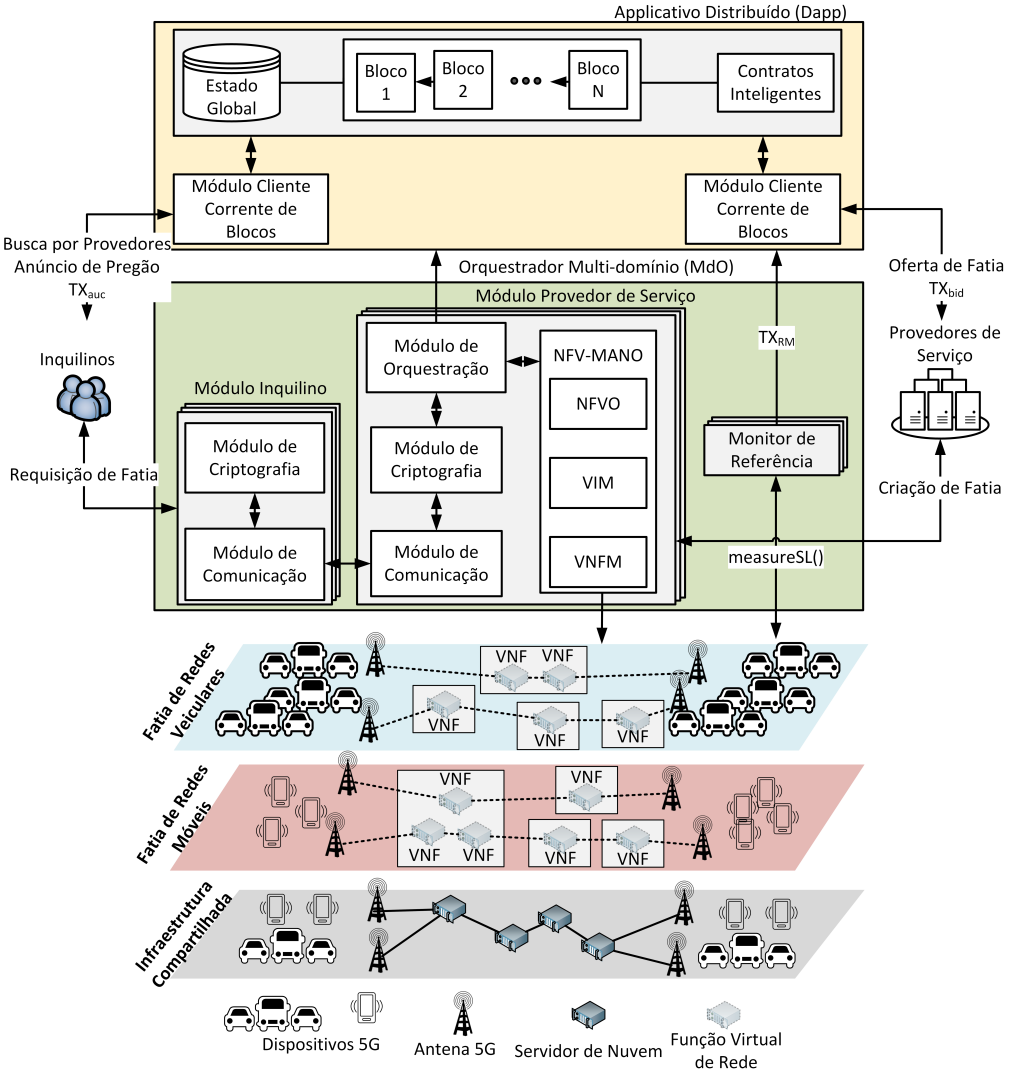


Figura 1. A arquitetura proposta do sistema distribuído de criação de fatias de rede. O orquestrador multidomínio acessa o módulo provedor de serviço para criar fatias de rede específicas, como para redes veiculares e redes móveis. O orquestrador registra as operações do ciclo de vida de um acordo de nível de serviço no aplicativo distribuído baseado em corrente de blocos para prover não-repúdio e transparência.

de serviço. Assim, os provedores de serviço que desejarem participar do pregão podem planejar uma fatia de rede que atenda os indicadores-chave e apresentar um lance no pregão. O lance do pregão é composto pelo SLA proposto, o preço do serviço oferecido, a restituição financeira do provedor e multa do inquilino em caso de descumprimento dos níveis de serviço. O inquilino verifica a corrente de blocos, seleciona o provedor de serviço que apresenta o melhor lance e cria uma chave simétrica para garantir a confidencialidade na troca de mensagens com o provedor selecionado.

Provedores de serviço utilizam o aplicativo distribuído para verificar as demandas e ofertar as fatias de rede. O registro das requisições de maneira distribuída na corrente de blocos permite a fácil e rápida verificação das requisições dos inquilinos. Após ser selecionado e ter estabelecido o contrato com o inquilino, o provedor utiliza o orquestra-

dor multidomínio para construir a cadeia de função serviço formada por VNFs de diversos fornecedores sediados em diferentes domínios. O módulo de orquestração registra na corrente de blocos os comandos de maneira criptografada utilizando o aplicativo distribuído, garantindo confidencialidade, não-repúdio e transparência ao usuário inquilino.

A arquitetura proposta também define um monitor de referência que se comunica com as cadeias de serviço criadas, realizando medidas ativas e passivas. Assim, o monitor executa avaliações em tempos aleatórios aos indicadores-chave de desempenho (KPIs) da cadeia alocada e registra os valores obtidos na corrente de blocos. O monitor de referência é executado em ao menos três instâncias: do lado do provedor, do cliente e de alguma terceira parte, de forma a medir os KPIs e até mesmo verificar uma adulteração de medida por alguma das partes. Um contrato inteligente verifica se as medidas correspondem ao SLA acordado previamente para tomar a decisão de punir ou não o provedor de serviço ou multar o inquilino. Diferentes KPIs poderão ser introduzidos ao longo do tempo através de atualizações nos contratos inteligentes, de forma muito mais ágil e com menor impacto de custos operacionais (OPEX) e de capital (CAPEX), comparado a um sistema que não utilize contratos inteligentes. Este artigo assume que o monitor é seguro e inviolável tanto pelo provedor, quanto pelo inquilino. A implementação deste monitor seguro, será objetivo de trabalho futuro explorando tecnologias como extensão de proteção de software da Intel (*Software Guard eXtension* - SGX).

3.1. Mensagens e Fluxo de Operações do Sistema Proposto

O sistema proposto é divido em três fases: (i) pregão eletrônico, (ii) orquestração de funções de serviço e (iii) verificação dos acordos de nível de serviço. As três partes do sistema cobrem todas as etapas do ciclo de vida e do gerenciamento de um acordo de nível de serviço (SLA). Todas as operações de todas as fases geram transações que são registradas na corrente de blocos para garantir transparência e não repúdio aos envolvidos.

A fase de pregão eletrônico anuncia uma demanda de serviço e seleciona o provedor de serviço (SP) que, em seguida, criará a cadeia de funções de serviços de rede para atender ao inquilino através de um serviço fim-a-fim. Logo, um inquilino interessado em receber um serviço emite uma transação de requisição de pregão eletrônico, $TX_{\text{auc}_{\text{req}}}$, ao contrato inteligente informando os indicadores-chave de desempenho desejados e iniciando um pregão na corrente de blocos, definida como:

$$TX_{\text{auc}_{\text{req}}} = [TX_{ID_{\text{auc}}}|\text{KPI}|t_{\text{out}}], \tag{1}$$

onde o campo $TX_{ID_{\text{auc}}}$ é o identificador da transação que iniciou o pregão, o campo KPI é o conjunto dos indicadores-chave de desempenho desejado e t_{out} é o tempo de expiração do pregão. Os provedores de serviço podem facilmente verificar as ofertas de pregões disponíveis consultando a corrente de blocos e efetuar um lance no pregão com uma proposta como resposta, emitindo uma transação de oferta $TX_{\text{bid}_{\text{resp}}}$, definida como:

$$TX_{\text{bid}_{\text{resp}}} = [TX_{ID_{\text{auc}}}|V_b|t_P|C_P|PK_{SP_j}], \tag{2}$$

onde o campo $TX_{ID_{\text{auc}}}$ identifica o pregão, o campo V_b é o valor do lance ofertado pelo provedor SP_j, o campo t_P define um limiar mínimo de níveis de serviço, como vazão e latência, que o provedor de serviço j (SP_j) deve cumprir, o campo C_P define a multa

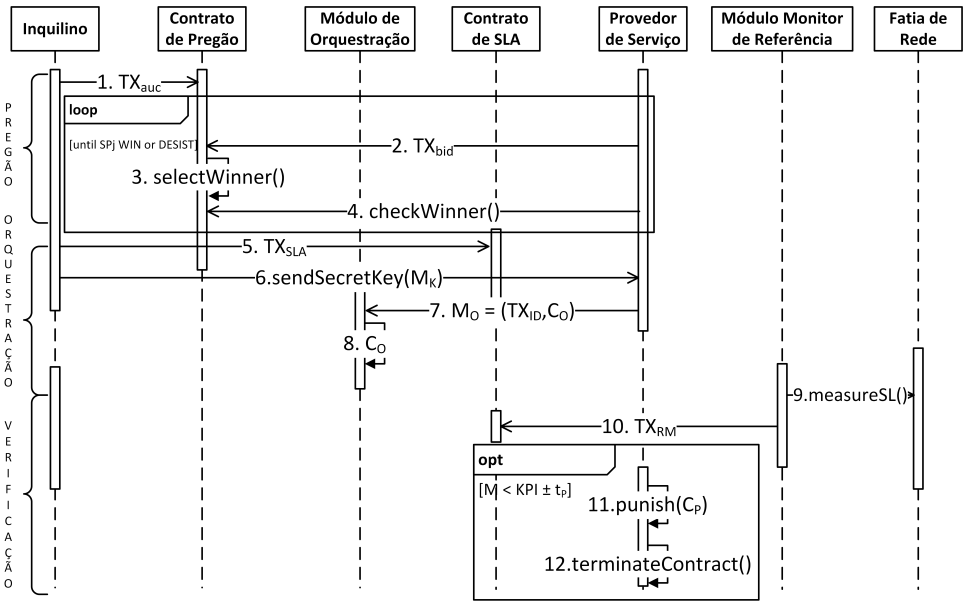


Figura 2. Diagrama de sequência do sistema proposto representando as etapas desde a busca por provedores de serviço até a verificação dos níveis de serviço.

ou punição financeira em caso de descumprimento de contrato e o campo PK_j é a chave pública do provedor. Ao emitir o primeiro lance, o contrato inteligente bloqueia um valor V_{stake} como garantia da real intenção de seguir com $TX_{\text{bid}_{\text{resp}}}$. O valor de V_{stake} é acordado entre os participantes na inicialização de corrente de blocos. O valor é devolvido ao provedor SP_j após a confirmação de TX_{orq} ou após outro participante vencer o pregão. Caso o provedor não honre com TX_{orq} prometido, V_{stake} é creditado ao inquilino T_i. Um contrato inteligente automatiza a verificação do vencedor, selecionando a melhor oferta ao inquilino e identificando o vencedor pela chave pública. A fase de orquestração de funções de serviço registra as operações de criação de cadeias de serviço para prover transparência de operações aos inquilinos e provedores de serviço. O provedor de serviço vencedor do leilão SP_j registra os níveis de serviço acordados emitindo uma transação TX_{SLA}, definida como:

$$TX_{\text{SLA}} = [TX_{ID_{\text{bid}}} | Sig_{T_i}], \tag{3}$$

o campo $TX_{ID_{\text{auc}}}$ identifica a transação que iniciou o leilão. Após emitir essa transação, SP_j envia ao módulo de orquestração uma mensagem $M_O = (TX_{ID_{\text{SLA}}}, C_O)$, contendo o identificador da transação $TX_{ID_{\text{SLA}}}$ e o comando de orquestração C_O. O módulo de orquestração verifica o registro da transação $TX_{ID_{\text{SLA}}}$ na corrente de blocos e repassa o comando C_O ao NFV-MANO para a criação da cadeia de serviço. Em seguida, o módulo encripta o comando C_O utilizando uma chave secreta SK_{ij} compartilhada com o inquilino T_i e gera a cifra $c_o = \text{Enc}(C_O)$. Por fim, o módulo emite uma transação de orquestração TX_{orq} para registrar os comandos na corrente de blocos. A transação de orquestração TX_{orq} é definida como:

$$TX_{\text{orq}} = [TX_{ID_{\text{SLA}}} | c_o], \tag{4}$$

onde o campo $TX_{ID_{\text{SLA}}}$ é o identificador da transação de SLA correspondente e o campo c_o é o comando utilizado para orquestração criptografado utilizando a chave secreta SK_{ij} compartilhada entre o inquilino e o provedor de serviço. O comando é armazenado de maneira criptografada para garantir a confidencialidade das funções utilizadas por T_i ao mesmo tempo em que garante a transparência aos agentes envolvidos. Ao emitir TX_{orq}, o provedor de serviço recebe o pagamento V_b pelo serviço oferecido. O valor V_b é bloqueado da conta do inquilino quando o inquilino emite a transação TX_{SLA} para evitar comportamento malicioso que resulte no não-pagamento do valor.

As cadeias de funções de rede em cenários com múltiplos domínios administrativos geralmente utilizam funções instanciadas em domínios concorrentes. A fase de orquestração proposta inclui um controle de acesso para evitar abuso de recursos dos domínios. Os provedores de serviço registram na corrente de blocos ao entrar na rede quotas de recursos que cada um dos demais domínios pode utilizar para orquestrar serviços. Assim, cada vez que um provedor utiliza a infraestrutura de outro domínio para sediar uma função de rede, o contrato inteligente de orquestração retira automaticamente o valor utilizado de recursos da quota destinada a esse domínio.

A fase de verificação dos níveis de serviço registra na corrente de blocos as medidas de desempenho da cadeia de funções de rede efetuadas e obtidas em tempos aleatórios. Um contrato inteligente recebe as medidas e as compara com o SLA previamente registrada com a transação da Equação 3 e verifica se alguma medida é menor que o limiar t_P registrado. O contrato pune automaticamente o provedor de serviço em C_P caso o provedor descumpra os níveis de serviço acordados ou emite uma multa para o inquilino no caso deste ser o infrator. Um valor superior a C_P é bloqueado na conta dos envolvidos durante a orquestração para evitar o não-pagamento da multa. A transação enviada pelo monitor de referências ao contrato inteligente é definida como:

$$TX_{\text{RM}} = [TX_{ID_{\text{SLA}}}|M], \tag{5}$$

onde o campo $TX_{ID_{\text{SLA}}}$ é o identificador da transação em que os níveis de serviço estão registrados e o campo M é o conjunto de medidas dos KPIs da cadeia de funções.

O sistema proposto define três contratos para processar as transações de cada fase definida acima. O contrato de pregão SC_A registra as transações TX_{auc} e TX_{bid} da fase do pregão eletrônico. O contrato de orquestração SC_O registra a transação TX_{orq} e efetua o controle de acesso entre os domínios. O contrato de SLA SC_{SLA} define as transações TX_{SLA} e TX_{RM}. A Figura 2 mostra o diagrama de sequência completo do sistema proposto, detalhando as etapas. O sistema proposto possui o seguinte fluxo de operações:

1. um inquilino T_i interessado em adquirir uma cadeia de funções de serviço inicia um pregão RA_{T_i} emitindo uma transação ao contrato SC_A com os KPIs desejados;
2. os SPs interessados em prover o serviço ao inquilino T_i ofertam no pregão RA_{T_i} emitindo uma transação de lance TX_{bid} ao contrato SC_A;
3. o contrato SC_A seleciona o menor lance no pregão RA_{T_i} como vencedor e identifica a chave pública PK_{SP_j} do provedor de serviço vencedor SP_j;
4. o provedor SP_j verifica se venceu o pregão e decide por emitir outro lance ou desistir do pregão RA_{T_i};
5. o inquilino T_i verifica que SP_j venceu o pregão RA_{T_i} e decide se vai prosseguir com o provedor vencedor ou escolherá outro provedor. Caso concorde com os

parâmetros t_P e C_P definidos, T_i emite uma transação TX_{SLA} assinada registrando o limiar t_P acordado, a multa C_P acordada com o provedor SP_j e a assinatura de T_i provando que as duas partes concordam com os parâmetros;

6. o inquilino T_i gera uma chave secreta SK_{ij} e a encripta utilizando a chave pública PK_{SP_j} do SP vencedor do pregão SP_j, gerando a cifra $c \leftarrow \text{Enc}_{PK_{SP_j}}(SK_{ij})$. O inquilino T_i assina a cifra $\sigma \leftarrow \text{Sign}_{SK_{ij}}(c)$ garantindo a autenticação e a envia ao provedor de serviço SP_j. A partir deste ponto, toda a comunicação entre T_i e SP_j é encriptada utilizando SK_{ij}, provendo confidencialidade à comunicação;
7. o provedor SP_j envia uma mensagem assinada ao módulo de orquestração contendo o identificador de transação de SLA $TX_{ID_{SLA}}$ e o comando C_O para criar uma cadeia de funções de serviço que atenda aos KPIs;
8. o módulo de orquestração envia o comando C_O para NFV-MANO para a criação da fatia e criptografa o comando C_O usando a chave secreta SK_{ij}, gerando uma cifra $c_o \leftarrow \text{Enc}_{SK_{ij}}(C_O)$. O módulo de orquestração emite uma transação de orquestração TX_{orq} registrando c_O na corrente;
9. após o inquilino se conectar à cadeia de serviços, um monitor de referência consulta em tempos aleatórios a fatia de rede alocada ao inquilino T_i para verificar os níveis de serviço;
10. o monitor de referência registra na corrente de blocos os níveis de serviço medidos na fatia de rede, provendo total transparência aos envolvidos;
11. se o conjunto de medidas feito pelo monitor de referência for menor que o limiar t_P definido pelo contrato, o contrato inteligente pune o provedor SP_j de acordo com o custo C_P acordado com T_i;
12. o contrato é desativado, seja por limite de tempo ou por violação das regras acordadas entre T_i e SP_j.

As etapas do sistema cobrem todos os requisitos do ciclo de vida de gerenciamento de um SLA apresentados na Seção 2. As etapas 1, 2, 3 e 4 do sistema cobrem a fase de descoberta de provedores de serviço, uma vez que é nessa fase que o inquilino T_i descobre o provedor SP_j que irá prover o serviço. A etapa 2 e 5 cobrem a definição de SLA e o estabelecimento de acordo, já que nessas etapa s o inquilino e o provedor de serviço, T_i e SP_j assinam os níveis de serviço e o modelo de punição por descumprimento de contrato. Nas etapas 9 e 10, o monitor de referência verifica os níveis de serviço, cobrindo a etapa de monitoramento de violações de SLA. A etapa 11 aplica as penalidades do SLA, enquanto a etapa 12 termina o SLA.

O sistema gera uma nova chave secreta para cada operação de orquestração. Assim, se um provedor de serviço SP_j mantém n inquilinos, SP_j possui um conjunto de chaves secretas $s = (SK_1, SK_2, ..., SK_n)$ para se comunicar com cada um dos n inquilinos. Essa propriedade garante que, caso uma chave qualquer SK_i seja revelada, as demais não serão afetadas por esse vazamento.

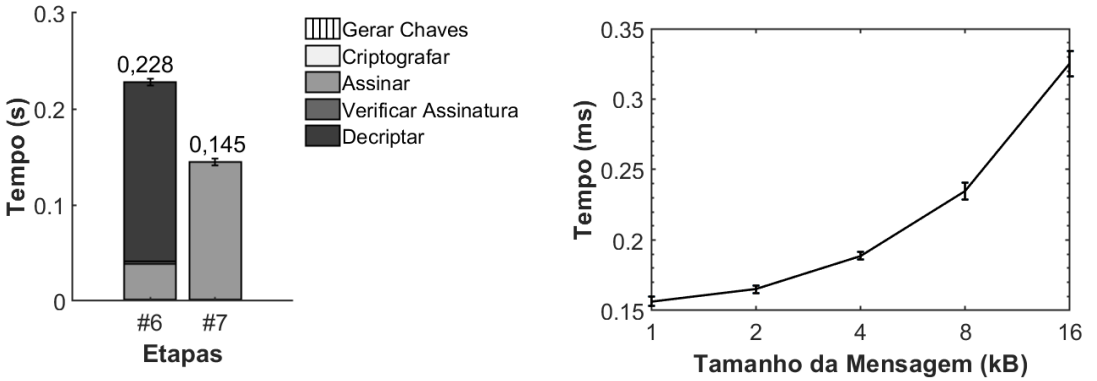
4. Desenvolvimento e Avaliação de Desempenho de um Protótipo do Sistema

Um protótipo do sistema proposto foi desenvolvido[1] utilizando a plataforma de código aberto Hyperledger Fabric 2.0 [Androulaki et al. 2018] para

[1]A implementação está disponível em https://github.com/GTA-UFRJ-team/NFVIaaS-Distributed-Orchestration

o desenvolvimento de corrente de blocos permissionadas utilizando o consenso Raft [Ongaro and Ousterhout 2014]. Embora a plataforma não apresente um protocolo BFT, o Hyperledger Fabric foi escolhido, uma vez que o aspecto organizacional e empresarial do Hyperledger Fabric se adéqua ao cenário de múltiplos domínios administrativos da proposta [Camilo et al. 2020b, de Souza et al. 2020]. Apesar do protótipo utilizar o Hyperledger Fabric, o sistema proposto é agnóstico a uma corrente de blocos específica e pode ser implementado em outras plataformas que suportam contratos inteligentes. Um computador i7-8700 CPU 3.20 GHz com 32 GB RAM e 6 núcleos físicos de processamento executa os nós da rede como contêineres Docker. Um contrato inteligente autoexecutável escrito em Go implementa as lógicas de transação propostas. O módulo de criptografia foi implementado em Python 2.7 utilizando a biblioteca pyCryptodome para as operações criptográficas. O módulo utiliza o sistema criptográfico de chave pública Rivest-Shamir-Adleman (RSA) com tamanho de chave de 2048 bits. O sistema de assinaturas digitais foi implementado utilizando o esquema de assinaturas probabilísticas do padrão #1 de criptografia de chave pública (*Public Key Cryptography #1 Standart Probabilistic Signature Scheme* - PKCS#1-PSS). A criptografia simétrica foi implementada utilizando o padrão de criptografia avançado (*Advanced Encryption System* - AES) com modo de contador (*Counter Mode* - CTR) como modo de operação. Os resultados apresentam intervalo de confiança de 95%.

O primeiro experimento avalia o tempo acrescido pelo sistema no provisionamento de cadeias de VNF. Esse tempo é a duração total das etapas 6 (*sendSecretKey*) e 7 (M_O) propostas pelo sistema. O experimento envia uma chave de 32 bytes na mensagem 6 e uma instrução de 64 bytes na mensagem 7. A Figura 3a mostra que o atraso adicional das duas etapas propostas é de cerca de $0,5$ segundo, demonstrando um acréscimo de tempo desprezível aos participantes envolvidos. As etapas de decriptar e assinar são responsáveis pelo maior acréscimo, enquanto encriptar, gerar chave e verificar assinatura possuem atraso desprezível comparadas com as outras etapas. O acréscimo introduzido pelo sistema é insignificante frente ao atraso no envio da mensagem ao considerar uma conexão de rede do inquilino com nuvem via Internet.



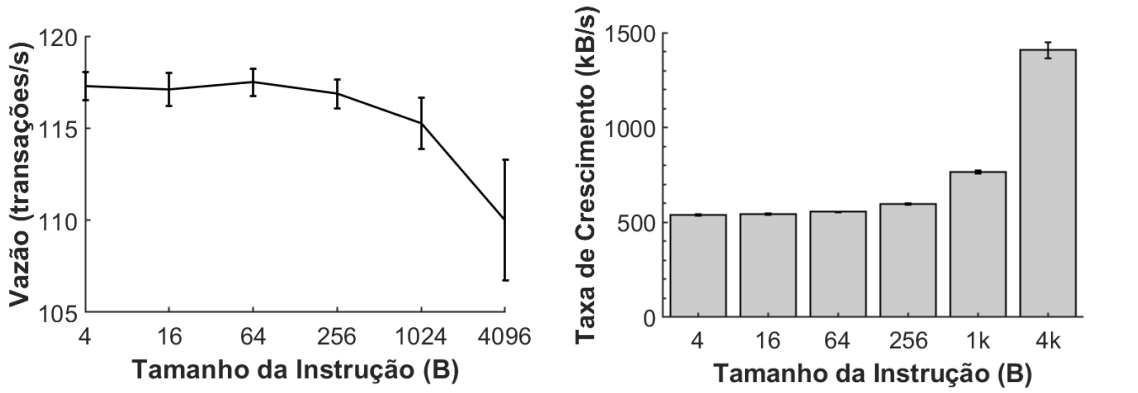
(a) Avaliação do acréscimo de tempo das etapas 6 e 7 do sistema proposto, divididas pelas operações criptográficas.

(b) Crescimento do tempo de criptografia utilizando AES256 em relação ao tamanho da mensagem.

Figura 3. Avaliação de desempenho do módulo de criptografia do sistema proposto.

O segundo experimento do módulo de criptografia avalia o tempo para criptografar

uma mensagem em relação ao tamanho da mensagem utilizando criptografia de chave simétrica AES256. O experimento verifica o acréscimo de tempo da criptografia na troca de mensagens entre o provedor de serviço e o inquilino. Como as mensagens trocadas não possuem tamanho fixo, o experimento varia o tamanho da mensagem entre 1 kB e 16 kB, de forma a considerar o envio de configurações para cadeias mais complexas de VNFs. A Figura 3b mostra que para uma mensagem de 16 kB, o acréscimo de tempo é menor que 0,35 ms, valor insignificante em relação ao envio da mensagem. Ademais, considerando o valor máximo padrão de segmentos TCP como 536 bytes, espera-se que a encriptação de grande parte das mensagens resulte em um acréscimo menor que 0,15 ms.



(a) Impacto do tamanho da instrução registrada em relação à vazão de transações.

(b) Crescimento da corrente de blocos em relação ao tamanho da instrução armazenada.

Figura 4. Avaliação de desempenho do aplicativo distribuído proposto.

A Figura 4 ilustra os resultados do contrato inteligente proposto, que implementa as lógicas de transação e o aplicativo distribuído em conjunto com a corrente de blocos. O primeiro experimento verifica o impacto do tamanho das instruções de orquestração na vazão de transações. Provedores de serviço registram as instruções de orquestração na corrente de blocos através do módulo de orquestração, como na etapa 8 do sistema. O experimento considera que operações mais complexas, como instanciação de funções em diversos domínios, implicam instruções mais longas. O cenário considera 8 clientes na corrente de blocos que emitem 100 transações de orquestração cada, agindo como provedores de serviço registrando os comandos de instanciação de VNFs. A Figura 4a mostra que o contrato proposto registra com facilidade centenas de instruções de criação de fatias de rede mesmo com instruções mais complexas. A taxa de transações atinge 117 transações por segundo até 256 B. Isso acontece porque o sistema passa a fragmentar as mensagens a partir de 512 B, resultando em uma queda na vazão de transações.

O segundo experimento do aplicativo distribuído avalia o crescimento da corrente de blocos variando os tamanhos das instruções a serem registradas. Assim como no experimento anterior, o cenário inclui 8 clientes que emitem 100 transações cada, agindo como provedores de serviço registrando comandos de orquestração na corrente de blocos. A Figura 4b mostra que o crescimento da corrente de blocos é estável com instruções de até 256 B, quando passa a crescer de maneira mais significativa. Exceto para casos específicos, comandos de orquestração em plataformas populares não possuem mais que dezenas de bytes [Rebello et al. 2019a]. Dessa maneira, a corrente de blocos cresce de maneira pouco significativa aos usuários.

5. Trabalhos Relacionados

A escolha dos provedores de serviços para fornecer a infraestrutura necessária para suportar as VNFs é a primeira etapa na orquestração de cadeias de serviço. Uma forma eficiente dos inquilinos descobrirem provedores de serviço é através de pregões e mercados eletrônicos. Gu *et al.* desenvolvem um mecanismo de leilão eletrônico para a provisão de cadeias de função de serviço (*Service Function Chaining* - SFC) em centro de dados [Gu et al. 2016]. Na proposta dos autores, os provedores de serviço são leiloeiros que vendem cadeias de serviço a usuários que agem como licitantes. Zhang *et al.* propõem um mecanismo de leilão estocástico para prover precificação de cadeias de serviço sobre demanda nos provedores de serviço [Zhang et al. 2017]. Os mecanismos de leilão propostos são centralizados, o que dificulta a auditoria de inquilinos no processo. A corrente de blocos [Nakamoto 2008] pode ser utilizada para prover um registro imutável e distribuído em leilões e mercados eletrônicos [Camilo et al. 2020a]. Além disso, a tecnologia de contratos inteligentes permite a automatização na escolha de um vencedor do pregão de maneira transparente aos inquilinos.

Diversos trabalhos aplicam a tecnologia de corrente de blocos para garantir segurança em ambientes de redes virtuais com múltiplos domínios administrativos. Alvarenga *et al.* propõem o uso de corrente de blocos para garantir segurança no gerenciamento de configurações e na migração de VNFs [Alvarenga et al. 2018]. Os autores propõem dois tipos de transação para definir e atualizar a configuração de uma VNF, mas não incluem a orquestração e a verificação de cumprimento de SLA na proposta. Rebello *et al.* apresentam o BSec-NFVO, um sistema baseado em corrente de blocos para garantir a segurança na orquestração de redes virtuais [Rebello et al. 2019a, Rebello et al. 2019c]. O BSec-NFVO armazena os comandos de orquestração de VNFs na corrente de blocos garantindo aos inquilinos transparência das operações dos provedores. Entretanto, a proposta dos autores não inclui etapas importantes no provimento de VNFs, como a descoberta de provedores de serviço, o controle de acesso durante a orquestração de VNFs e o monitoramento de SLA. Rosa e Rothenberg apresentam um arcabouço para orquestração de serviços multi-domínios utilizando aplicativos distribuídos baseados em corrente de blocos [Rosa and Rothenberg 2018]. Os autores apresentam um caso de uso em que a corrente de blocos é utilizada para armazenar as permissões de acesso de cada domínio. Um contrato inteligente (*smart contract*) verifica as permissões de acesso armazenadas para garantir o controle de acesso na orquestração de VNFs. O controle de acesso proposto, no entanto, não é implementado pelos autores, além da proposta não considerar todas as etapas do ciclo de vida de um SLA.

Balachandran *et al.* propõem o EDISON, um sistema para autenticação e controle de acesso baseado em corrente de blocos para assegurar o gerenciamento e orquestração de redes definidas por software (*Software Defined Networking* - SDN) [Balachandran et al. 2020]. O EDISON utiliza contratos inteligentes para controlar o acesso de inquilinos a elementos de redes, além de chaves de sessão para garantir confidencialidade e sigilo para frente (*forward secrecy*) na comunicação entre as entidades. A proposta garante segurança ao encriptar todo o tráfego entre o inquilino o elemento de rede e transparência ao registrar os pacotes na corrente de blocos, mas requer um grande espaço de armazenamento para suportar a corrente de blocos completa. Ademais, a proposta considera que os inquilinos e os provedores de serviço se conhecem previamente, o que não cumpre os requisitos de ciclo de vida de um SLA.

Ao contrário dos artigos previamente citados, este artigo propõe um sistema eficiente e rápido para prover a orquestração de VNFs de maneira distribuída e segura considerando todas as etapas de um ciclo de vida de um SLA. O sistema registra todas as operações dos provedores na corrente de blocos para prover transparência completa aos inquilinos. Um contrato inteligente automatiza o processamento do pregão proposto de maneira distribuída, eliminando a necessidades de intermediários. O uso de chaves simétricas para criptografar o tráfego entre o inquilino e os provedores garante a confidencialidade de maneira ágil e robusta.

6. Conclusão

O fatiamento de redes fornece serviços fim-a-fim customizados a inquilinos através do encadeamento de funções virtuais de rede instanciadas em múltiplos domínios concorrentes sem confiança mútua. Nestes cenários, é importante identificar e responsabilizar comportamentos maliciosos de provedores e inquilinos, seja na instanciação da cadeia de serviço ou no cumprimento do acordo de níveis de serviço. Este artigo propõe um sistema baseado em corrente de blocos para garantir a segurança na orquestração de fatias de rede de maneira rápida e distribuída em ambientes com múltiplos domínios administrativos. O sistema proposto cumpre todas os requisitos de ciclo de vida de gerenciamento de um acordo de níveis de serviço, além de registrar as etapas de orquestração na corrente de blocos, provendo transparência, rastreabilidade e não-repúdio das operações. Assim, inquilinos podem facilmente consultar a corrente de blocos e verificar o cumprimento dos acordos. Os resultados da avaliação de desempenho de um protótipo desenvolvido mostram que o sistema registra as operações de orquestração de maneira rápida, atingindo taxas acima de 100 transações por segundo. Além disso, o esquema criptográfico utilizado garante a confidencialidade de maneira ágil sem perder a transparência das operações entre os usuários.

Em trabalhos futuros, é prevista a integração do protótipo desenvolvido com uma plataforma de virtualização de funções de rede, além da proposta de um monitor seguro utilizando Intel SGX.

Referências

Alvarenga, I. D., Rebello, G. A. F., and Duarte, O. C. M. B. (2018). Securing configuration management and migration of virtual network functions using blockchain. In *2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9.

Androulaki, E., , et al. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, page 30.

Balachandran, C., A. C, P., Ramachandran, G., and Krishnamachari, B. (2020). EDISON: A Blockchain-based Secure and Auditable Orchestration Framework for Multi-domain Software Defined Networks. In *2020 IEEE Blockchain*, pages 144–153.

Camilo, G. F., Rebello, G. A. F., de Souza, L. A. C., and Duarte, O. C. M. B. (2020a). AutAvailChain: Automatic and Secure Data Availability through Blockchain. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–6.

Camilo, G. F., Rebello, G. A. F., de Souza, L. A. C., and Duarte, O. C. M. B. (2020b). A secure personal-data trading system based on blockchain, trust, and reputation. In *2020 IEEE Blockchain*, pages 379–384.

de Oliveira, M. T., Reis, L. H., Medeiros, D. S., Carrano, R. C., Olabarriaga, S. D., and Mattos, D. M. (2020). Blockchain reputation-based consensus: A scalable and resilient mechanism for distributed mistrusting applications. *Computer Networks*, 179:107367.

de Souza, L. A. C., Antonio F. Rebello, G., Camilo, G. F., Guimarães, L. C. B., and Duarte, O. C. M. B. (2020). DFedForest: Decentralized Federated Forest. In *2020 IEEE Blockchain*, pages 90–97.

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

Gu, S., Li, Z., Wu, C., and Huang, C. (2016). An efficient auction mechanism for service chains in the NFV market. In *IEEE INFOCOM 2016*, pages 1–9.

Maarouf, A., Marzouk, A., and Haqiq, A. (2015). Practical modeling of the SLA life cycle in Cloud Computing. In *2015 ISDA*, pages 52–58.

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 Communications Magazine*, 55(2):216–223.

Michelin, R. A., Dorri, A., Steger, M., Lunardi, R. C., Kanhere, S. S., Jurdak, R., and Zorzo, A. F. (2018). Speedychain: A framework for decoupling data from blockchain for smart cities. In *Proceedings of the 15th EAI MobiQuitous*, pages 145–154.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Disponível em https://bitcoin.org/bitcoin.pdf. Acessado em 17 de abril de 2021.

Ongaro, D. and Ousterhout, J. (2014). In Search of an Understandable Consensus Algorithm. In *USENIX ATC 14*, pages 305–319, Philadelphia, PA. USENIX.

Pinno, O. J. A., Gregio, A. R. A., and De Bona, L. C. E. (2017). ControlChain: Blockchain as a Central Enabler for Access Control Authorizations in the IoT. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6.

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 *2019 IEEE International Conference on Communications (ICC)*, pages 1–6.

Rebello, G. A. F. et al. (2019b). Correntes de blocos: Algoritmos de consenso e implementação na plataforma hyperledger fabric. In *38º JAI do XXXIX Congresso da Sociedade Brasileira de Computação (CSBC 2019)*.

Rebello, G. A. F. et al. (2019c). Providing a sliced, secure, and isolated software infrastructure of virtual functions through blockchain technology. In *2019 HPSR*, pages 1–6.

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

Wu, L. and Buyya, R. (2010). Service Level Agreement (SLA) in Utility Computing Systems.

Zhang, X., Huang, Z., Wu, C., Li, Z., and Lau, F. C. M. (2017). Online Stochastic Buy-Sell Mechanism for VNF Chains in the NFV Market. *IEEE JSAC*, 35(2):392–406.