SINFONIA: Gerenciamento Seguro de Funções Virtualizadas de Rede através de Corrente de Blocos

Gabriel Antonio Fontes Rebello, Igor Drummond Alvarenga, Igor Jochem Sanz e Otto Carlos Muniz Bandeira Duarte

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

{gabriel,alvarenga,sanz,otto}@gta.ufrj.br

Resumo. *A tecnologia de virtualização de funções de rede (Network Function Virtualization – NFV) provê um modelo de rede flexível e de baixo custo, que substitui as funções de rede proprietárias implementadas em dispositivos físicos por funções virtualizadas em software, que são executadas em máquinas de uso geral. As funções virtualizadas de rede são orquestradas em um ambiente multi-inquilino, distribuído, e sem confiança entre os pares e, portanto, são susceptíveis a ameaças de segurança. Este artigo propõe o SINFONIA, um sistema baseado em corrente de blocos que fornece segurança às redes virtualizadas, garantindo a auditabilidade, o não-repúdio e a integridade das operações de orquestração. O SINFONIA possui uma arquitetura modular que permite orquestrar as funções de rede de forma simples e ágil. Um protótipo para a Open Platform for Network Function Virtualization (OPNFV) foi desenvolvido com a implementação de uma corrente de blocos específica e um protocolo de consenso resistente a conluio. Os resultados mostram que o SINFONIA provê segurança com baixa sobrecarga ao orquestrador de nuvem, e que o desempenho permanece estável ao aumentar o número dos participantes do consenso.*

Abstract. *Network Function Virtualization (NFV) technology provides a flexible and low-cost network model that replaces proprietary hardware network functions with software-based virtualized network functions which run on general-purpose machines. Virtual functions are orchestrated in a distributed multi-tenant trustless environment and are, thus, susceptible to security threats. This article proposes SINFONIA, a blockchain-based system that provides security to virtualized networks, ensuring auditability, non-repudiation, and integrity of orchestration operations. SINFONIA provides a modular architecture which allows the orchestration of network functions in a simple and agile way. A prototype for the Open Platform for Network Function Virtualization (OPNFV) was developed with the implementation of a specific blockchain and a collusion-resistant consensus protocol. The results show that SINFONIA provides security with low overhead to the cloud orchestrator with stable performance as the number of consensus participants increases.*

1. Introdução

A tecnologia de virtualização de funções de rede (*Network Function Virtualization* – NFV) permite reduzir gastos e flexibilizar o gerenciamento e a operação das redes de comunicações através da substituição de recursos em *hardware* dedicado por funções virtualizadas em *software*, que são executadas em *hardware* de uso geral [Pattaranantakul et al. 2016].

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

Assim, os sistemas intermediários (*middleboxes*), como *firewalls*, sistemas de detecção e prevenção de intrusão, balanceadores de carga, entre outros, que hoje são implementados em *hardware* específicos de um fabricante, serão substituídos por funções em *software* virtualizadas que podem ser providas por diferentes fornecedores [Sekar et al. 2012]. As funções virtualizadas de rede (*Virtual Network Functions* – VNFs) permitem a introdução de uma cadeia de funções de rede no caminho da origem até o destino que fornece, sob demanda, um serviço adaptado para cada aplicação ou usuário.

Nas redes da próxima geração, baseadas na tecnologia NFV [ETSI 2014], o encadeamento de funções de rede (*Service Function Chaining* – SFC) [Halpern e Pignataro 2015] é o procedimento fundamental para fornecer controle e gerenciamento flexível do tráfego de um serviço ou de uma aplicação. A virtualização de funções de rede associada à tecnologia de redes definidas por *software* (*Software-Defined Networking* – SDN) provê a flexibilidade, a agilidade e o baixo custo desejado para as telecomunicações, mas traz novos desafios de segurança [Medhat et al. 2017]. Neste cenário, os inquilinos compartilham a mesma infraestrutura de nuvem, e cadeias podem envolver funções virtualizadas instanciadas em domínios de operadoras concorrentes. Desta forma, é necessário garantir que a cadeia de funções virtualizadas de rede seja construída de maneira confiável em um ambiente sem confiança entre os pares, sejam estes inquilinos ou domínios. O ambiente multi-inquilino e multi-domínio aumenta a possibilidade de ataques dentro da nuvem e dificulta a responsabilização ou uma possível indenização pelos provedores do serviço devido a um mau funcionamento. Além disso, os impactos de possíveis ataques tornam-se bem maiores, uma vez que ataques ao hospedeiro de funções de rede podem comprometer milhares de usuários simultaneamente.

O modelo FCAPS (*Fault, Configuration, Administration, Performance and Security)* é o padrão de gestão de redes de telecomunicação [Raman 1998] em ambientes centralizados e de provedor único. No entanto, o cenário de virtualização de funções de rede é um ambiente distribuído que não possui confiança entre os pares, de forma que o FCAPS não é diretamente aplicável. Neste cenário, diversos provedores oferecem serviços distribuídos através de múltiplas nuvens, trazendo novos desafios, tais como: i) selecionar as métricas que garantem a corretude do serviço fim-a-fim de uma cadeia de funções de rede; ii) identificar o provedor de nuvem, dentre os participantes de uma cadeia de funções de rede, responsável por uma falha; e iii) estabelecer a proveniência, o impacto e o tempo que uma determinada falha permaneceu indetectada. Para solucionar estes desafios, portanto, uma solução com o uso de corrente de blocos é apropriada.

Em trabalhos anteriores, os autores avaliaram o desempenho de cadeias de funções de rede na plataforma OPNFV [Sanz et al. 2018, Sanz et al. 2017] e avaliaram o uso de corrente de blocos na plataforma OPNFV para configuração e migração seguras de funções virtualizadas de rede [Alvarenga et al. 2018]. Este artigo propõe, desenvolve e avalia o SINFONIA[1] (*Secure vIrtual Network Function Orchestrator for Non-repudiation, Integrity, and Auditability*), um sistema para o gerenciamento ágil e seguro das operações de orquestração de cadeias de funções virtualizadas de rede através de corrente de blocos (*blockchain*). A proposta do SINFONIA estende, para um cenário multi-domínio, as premissas de configuração e de administração do modelo FCAPS. Ao mesmo tempo, o SINFONIA garante a transparência das operações de rede realizadas pelos diferentes provedores através do uso de correntes de blocos, pois registra de forma imutável todas as instruções de construção, modificação e remoção da cadeia. Assim, o SINFONIA provê os mecanismos necessários para um gerenciamento seguro de funções virtualizadas de rede, garantindo a auditabilidade das operações e a responsabilização dos autores

[1]Disponível em http://www.gta.ufrj.br/sinfonia.

pelas consequências da operação. O SINFONIA também garante a autenticidade, a integridade e o não-repúdio das instruções enviadas ao orquestrador da plataforma de nuvem. Através do uso de criptografia de chave assimétrica, aliada às características intrínsecas da estrutura de dados de corrente de blocos, assegura-se que nenhuma instrução é adulterada após ser enviada, o que permite confirmar sua proveniência. O sistema também garante a imutabilidade e irretroatividade do histórico de operações realizadas pelo orquestrador de nuvem. A combinação das propriedades de não-repúdio, imutabilidade e irretroatividade garante a auditabilidade de todo histórico de transações efetuadas, o que é essencial em um ambiente sem confiança entre os pares. Portanto, o SINFONIA provê a inquilinos e provedores as evidências da operação correta da orquestração de funções de rede que compõem a cadeia de funções de uma comunicação. Essas evidências são imprescindíveis para investigação em caso de um incidente de segurança.

Um protótipo do SINFONIA é implementado na nuvem *Open Platform for Network Function Virtualization* (OPNFV), que fornece a infraestrutura para a implementação da corrente de blocos, a execução de instruções de orquestração e o repositório de VNFs. Com o objetivo de atingir uma alta vazão do número de instruções de gerenciamento de cadeias de funções de rede e obter um consenso robusto a ataques de conluio, o protocolo de consenso *Practical Byzantine Fault Tolerance* (PBFT) [Castro et al. 1999] foi adaptado para utilização em correntes de blocos. O sistema SINFONIA é pioneiro na implementação de uma corrente de blocos modificada para atender às necessidades um cenário NFV. Os resultados mostram que é possível encadear funções de rede de forma ágil e segura e em uma infraestrutura de uso geral. Por fim, o SINFONIA implementa um protocolo de consenso que permite que novas operações realizadas na cadeia de funções virtualizadas de rede sejam inseridas de forma confiável sem a necessidade de uma autoridade central comum a todos os orquestradores.

2. Trabalhos Relacionados

O gerenciamento de cadeias de funções de rede é um procedimento crítico em relação à segurança, pois atua diretamente no encaminhamento da mensagem da origem ao destino, podendo afetar simultaneamente milhares de usuários. A adição de novas funcionalidades e o ambiente multi-inquilino aumentam as possibilidades de ameaças. Assim, é necessário garantir a correta operação e gerar evidências imutáveis das operações de gerenciamento tomadas sobre as funções virtualizadas, no encadeamento de funções de rede. Desta forma, é possível identificar possíveis problemas, apontar os responsáveis para futuras indenizações e garantir maior segurança neste novo modelo de redes de comunicação.

A literatura apresenta trabalhos que buscam prover auditabilidade e proveniência às operações de encadeamento. Dölitzscher e Clarke definiriam o conceito de auditabilidade de segurança como serviço (*Security Audit as a Service* – SAaaS) com o objetivo de detectar incidentes de segurança na nuvem [Doelitzscher et al. 2013]. Os autores desenvolveram uma arquitetura e uma linguagem para auditabilidade leve e em tempo real que é realizada conforme a dinamicidade da infraestrutura da nuvem. Rübsamen *et al.* propuseram um esquema para garantir a segurança e a privacidade de evidências coletadas na nuvem também para fins de auditabilidade [Rübsamen et al. 2016]. Essas evidências se resumem a *logs*, provas criptográficas, documentações, entre outras. Os autores implementam um protótipo de coleta de *snapshots* de máquinas virtuais em uma nuvem baseada em Openstack com o objetivo de detectar possíveis violações nessas imagens. Apesar de proverem auditabilidade, os trabalhos mencionados não garantem confiabilidade, pois assumem que a nuvem é uma entidade confiável e segura para armazenamento da proveniência de dados. Dessa forma, não há proteção contra possíveis modificações maliciosas por parte do provedor.

Outros autores têm proposto correntes de blocos[2] para garantir a rastreabilidade das transações em ambientes sem confiança entre pares. Zawoat e Hasan propuseram SECAP, um esquema baseado em corrente de blocos que armazena de forma segura uma árvore de proveniência de aplicações na nuvem [Zawoad e Hasan 2016]. No entanto, o esquema proposto se restringe apenas ao registro das mudanças de estados de aplicações. Bozic *et al.* propõem um esquema de orquestração de máquinas virtuais usando um sistema baseado em corrente de blocos como mediador de alterações no estado de execução destas máquinas [Bozic et al. 2017]. Na proposta, as instruções enviadas para o hipervisor de virtualização são registradas na corrente de blocos como transações. No entanto, os autores limitam-se a instruções de criação de máquinas virtuais e não possuem uma implementação da arquitetura proposta. Alvarenga *et al.* avaliaram o desempenho do uso de corrente de blocos na plataforma OPNFV para atualização e migração seguras de funções virtualizadas de rede sem revelar a identidade dos pares e, portanto, sem se preocupar com a segurança e a garantia de auditabilidade das operações de orquestração [Alvarenga et al. 2018]. A proposta deste artigo, por sua vez, é tornar seguras as operações de orquestração de VNFs sem a necessidade de uma autoridade centralizada.

O foco deste artigo é registrar de forma segura a proveniência de dados na nuvem. Assim, os mecanismos de consenso e corrente de blocos devem ser eficientes para que o ambiente virtualizado consiga prevenir ameaças sem comprometer a latência e o processamento da rede. Portanto, correntes de blocos baseadas em prova de trabalho ou tolerantes apenas a falhas desastrosas não atendem a estas demandas. Logo, o protocolo de consenso apropriado pertence à classe de protocolos tolerantes a falhas bizantinas (*Byzantine Fault Tolerance* – BFT), pois promovem o consenso com baixa latência para até algumas dezenas de participantes do consenso, e aceitam comportamento malicioso de até um terço dos participantes do consenso [Vukolić 2016, Miller et al. 2016, Cachin e Vukolić 2017]. O projeto Hyperledger Fabric [Cachin 2016] se propõe a implementar correntes de blocos BFT. Porém, hoje, o projeto não possui um protocolo BFT implementado. O protocolo de consenso em correntes de blocos do arcabouço Tendermint [Kwon 2014] apresenta uma falha de impasse em sua máquina de estados (*deadlock*). O protocolo HoneyBadgerBFT [Miller et al. 2016], desenvolvido pelas universidades de Berkeley e Cornell, é um protocolo ainda sem validação formal que não fornece garantia de ordenação temporal de transações, e portanto, não é adequado à proposta deste trabalho. O protocolo BFT-Smart [Sousa et al. 2017] é um outro protocolo de consenso BFT, porém Jason Yellick, desenvolvedor e responsável pelo Hyperledger Fabric na Linux Foundation, listou seis falhas fundamentais que inviabilizam o seu uso em correntes de blocos, entre elas, que o protocolo não realiza validação de transações e não permite que participantes de consenso comprovem a validade o bloco proposto [Yellick 2018]. Portanto, não é do conhecimento dos autores uma ferramenta de código aberto que atenda às necessidades de um ambiente virtualizado que previna ameaças sem comprometer a latência e o processamento da rede.

Este artigo propõe um sistema que assegura a proveniência das operações de orquestração, ao contrário dos artigos acima citados, que se concentram em registrar de forma segura a proveniência de dados na nuvem. Decidiu-se implementar uma corrente de blocos e um algoritmo de consenso próprios.

3. O Sistema SINFONIA Proposto

O objetivo do sistema SINFONIA é proteger e registrar em uma corrente de blocos todas as instruções de criação, remoção e alteração de máquinas virtuais, funções virtualizadas de rede

[2]Nakamoto introduziu o conceito de corrente de blocos como uma estrutura de dados distribuída para resolver o problema do gasto duplo em moedas digitais [Nakamoto 2008].

e cadeias de funções. Assim, o SINFONIA oferece a facilidade de todos os orquestradores, os inquilinos e os administradores da nuvem poderem verificar localmente um histórico imutável de instruções para identificar participantes maliciosos na rede. O cenário é composto pelos orquestradores, inquilinos e administradores da nuvem. O orquestrador de nuvem é a entidade que cria a cadeia de funções de rede através da plataforma de virtualização utilizada no centro de dados. O administrador da nuvem é a operadora responsável por um ou mais centros de dados na qual a virtualização é realizada e cada orquestrador implementa suas funções. Um domínio é o conjunto de centros de dados administrado por uma operadora. Um inquilino é um cliente do domínio que usufrui de um serviço provido pelo encadeamento de funções de rede.

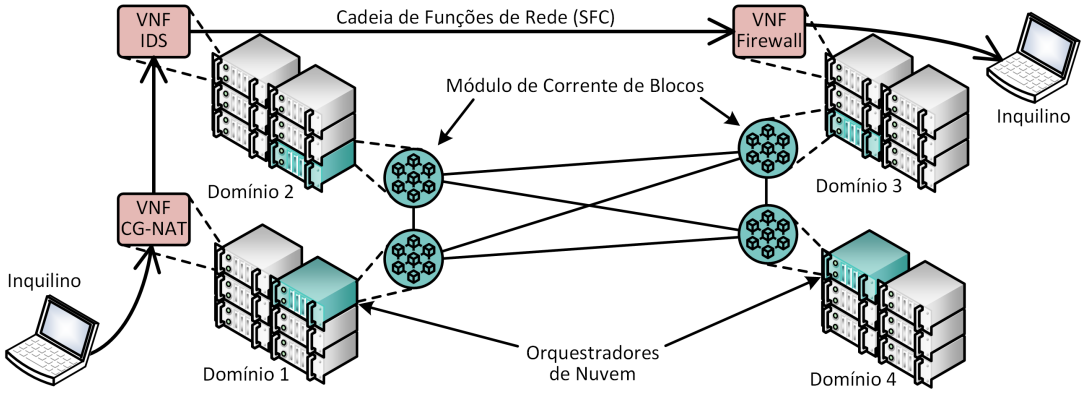


Figura 1: Exemplo de cenário de utilização do sistema SINFONIA, no qual uma cadeia de funções de rede possui VNFs localizadas em diferentes domínios de provedores de nuvem NFV. A corrente de blocos garante a imutabilidade dos registros de operações realizadas pelos orquestradores de nuvens entre diferentes domínios.

O cenário de comunicação entre grandes centros de dados implica que o ambiente NFV possua: i) número limitado de operadoras identificadas, uma vez que cada operadora possui contratos de serviço com diferentes inquilinos; ii) baixo número de falhas desastrosas (*crash failures*), devido à alta disponibilidade necessária aos grandes centros de dados; iii) alta vazão e baixo atraso na comunicação fim-a-fim, pois as funções são implementadas no núcleo da rede; e iv) tolerância a comportamento malicioso, para permitir um ambiente confiável entre operadoras e inquilinos concorrentes. Um cenário de encadeamento de funções para um serviço é ilustrado na Figura 1. Neste cenário, uma cadeia de funções de rede é composta por VNFs hospedadas em diferentes domínios de operadoras de telecomunicações. Cada operadora possui um domínio, que contém um único orquestrador de VNFs e máquinas de propósito geral que podem hospedar VMs e VNFs. O posicionamento geográfico das funções de rede pode afetar sua eficácia de acordo com sua função. Dessa forma, uma VNF IDS pode ser mais eficaz quando instanciada em um centro de dados especializado em detecção de intrusão e que realiza a correlação de informações e registros de detecção com outras VNF IDS. Uma VNF tradutora de endereços de Internet (*Carrier Grade* NAT – CG-NAT), por sua vez, necessita estar localizada no ponto de acesso mais próximo ao cliente. Estes requisitos de posicionamento de VNFs se tornam pontos críticos quando não há um centro de dados da operadora na localização desejada, necessitando o uso de recursos virtuais de outra operadora mais próxima. Neste ambiente de nuvem flexível, uma cadeia de funções de rede pode ter componentes instanciados em domínios de operadoras concorrentes, criando, portanto, um ambiente onde não há confiança entre os pares.

As instruções de orquestração de uma cadeia são assinadas por seus respectivos inquilinos e devem ser validadas antes de serem executadas. Uma instrução é uma chamada ao orquestrador de nuvem. Uma transação é uma instrução assinada pelo inquilino que a emite ou

uma resposta à instrução assinada pelo orquestrador que a emite. A cada instrução de escrita, isto é, criar, remover ou alterar uma função de rede virtualizada, uma cadeia de funções de rede ou uma rede virtual, corresponde uma transação de solicitação e uma transação de resposta assinadas e aceitas através de consenso. Desta forma, a presença de uma instrução de escrita assinada na corrente de qualquer orquestrador é garantia de que esta instrução foi realizada em um momento determinado, por um orquestrador identificável e solicitada por um inquilino em específico. Tanto as transações de requisição quanto as transações de resposta das instruções de escrita citadas são registradas na corrente de blocos. Em cada domínio, deve existir uma instância do SINFONIA cujo módulo de corrente de blocos contém uma cópia da corrente de blocos global. Desta forma, o SINFONIA permite que cada inquilino gerencie suas cadeias de funções de rede e que cada domínio orquestre a parte desta cadeia sob sua responsabilidade, escrevendo cada operação realizada por cada parte na corrente de blocos. O SINFONIA garante a todos os domínios e inquilinos participantes a capacidade de auditoria das operações realizadas.

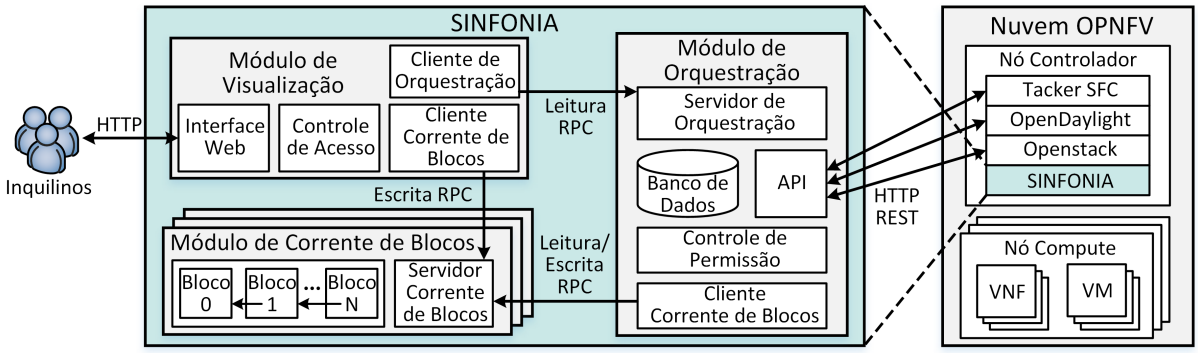


Figura 2: Arquitetura do sistema SINFONIA proposto. Os inquilinos acessam o sistema através de uma interface *web* e realizam operações de orquestração, que são assinadas e enviadas ao módulo de corrente de blocos em forma de transações. As transações validadas são incorporadas à corrente de blocos para serem lidas pelo módulo de orquestração. O módulo de orquestração envia requisições HTTP para o orquestrador da nuvem OPNFV, que então retorna o resultado da operação também como uma transação assinada.

A arquitetura do SINFONIA, mostrada na Figura 2, é dividida em três módulos: i) o módulo de visualização, responsável pela interface entre os inquilinos e a plataforma de nuvem que oferece serviços de NFV e SFC; ii) o módulo de orquestração, responsável pela execução das instruções enviadas pelos inquilinos da plataforma de nuvem através do módulo de visualização e iii) o módulo de corrente de blocos, responsável por validar a execução de instruções do módulo de visualização e repassá-la o módulo de orquestração.

O **Módulo de Visualização e de Controle de Acesso** tem o objetivo de tornar a orquestração de VNFs, SFCs, classificadores e redes simples e intuitiva para o inquilino. Este módulo é composto por quatro componentes principais: a interface *web*; o gerenciador de controle de acesso; o cliente de orquestração e o cliente de corrente de blocos. O primeiro componente é uma interface *web* amigável, que permite ao inquilino gerenciar seus serviços de NFV e SFC contratados, bem como emitir instruções. O segundo é um gerenciador de controle de acesso que aplica as políticas de acordo de nível de serviço (*Service Level Agreements* – SLAs) a cada inquilino, bem como restringe o acesso de cada inquilino a seus serviços contratados. O terceiro é um cliente de orquestração, que se comunica com o módulo de orquestração a fim de executar solicitações de leitura do estado de serviços na plataforma. O último componente é um cliente de corrente de blocos, que envia as instruções solicitadas ao módulo de corrente de blo-

cos a fim de executar operações de escrita de estado de serviços na plataforma. A comunicação dos componentes clientes é realizada através de chamadas de procedimento remoto (*Remote Procedure Call* – RPC), protegidas pelo protocolo TLS (*Transport Layer Security*).

O **Módulo de Orquestração** é responsável por executar as instruções recebidas do módulo de visualização e é composto por cinco componentes principais. O primeiro componente é um servidor de orquestração, que recebe as chamadas RPC do módulo de visualização. O segundo é um banco de dados que registra as informações de conta e serviços pertencentes a cada inquilino. O terceiro é um sistema de controle de permissão, que atua em conjunto com o banco de dados para verificar se um usuário é autorizado a executar a instrução recebida. O quarto, um cliente de corrente de blocos que se comunica com o módulo de corrente de blocos a fim de verificar a existência de instruções pendentes, bem como para registrar o resultado de uma instrução executada. Por fim, uma API (*Application Programming Interface*) para conexão com a plataforma OPNFV e efetivação das instruções autorizadas.

O **Módulo de Corrente de Blocos** atua como mediador de solicitações de escrita e é composto por um servidor de corrente de blocos e pela própria corrente de blocos. O servidor de corrente de blocos recebe chamadas RPC para escrita e consulta da corrente de blocos. A corrente de blocos é um repositório imutável de todas as instruções emitidas na plataforma. Cada instrução requisitada é assinada através da utilização de um par de chaves assimétrico pertencente a um inquilino, criando uma transação de instrução. A cada intervalo de tempo, da ordem de segundos, todas as transações são registradas em um bloco, associado à função resumo (*hash*) do bloco anterior e assinado pelo módulo de corrente de blocos com um par de chaves fornecido pelo gestor do serviço de nuvem. Dessa forma, é construída uma corrente imutável e íntegra. A combinação dessas funcionalidades permite a auditagem das requisições de uso dos serviços oferecidos pela plataforma, indispensável no caso da ocorrência de um incidente de segurança. Múltiplos módulos de corrente de blocos são executados simultaneamente e mantêm réplicas da corrente de blocos aceitas através de um protocolo de consenso adaptado ao caso da corrente de blocos.

3.1. A Corrente de Blocos Desenvolvida para SINFONIA

A corrente de blocos é uma estrutura de dados replicada cuja utilização garante a confiança e o funcionamento correto de um sistema distribuído sem a necessidade de uma autoridade central comum [Nakamoto 2008]. Uma corrente de blocos funciona como um livro-razão (*ledger*), ou registro (*log*) permanente, cujas entradas são blocos de transações ordenadas. A transação representa uma ação atômica a ser armazenada na corrente de blocos. Cada bloco na corrente é identificado por um valor resultante de uma função resumo criptográfica e possui tanto as transações realizadas em um determinado intervalo de tempo, como o identificador de seu antecessor. A exceção a esta regra é o bloco inicial, que não possui bloco anterior. O uso de uma função resumo criptográfica, cuja saída muda drasticamente após a alteração de qualquer *bit*, garante a imutabilidade de um bloco após a inserção de um sucessor e, consequentemente, da corrente. Dado que a corrente é replicada em todos os módulos de corrente de blocos, e que toda transação é assinada, a propriedade de não-repúdio entre clientes da corrente é garantida.

Para o sistema SINFONIA, foi desenvolvida uma corrente de blocos e um modelo de transações particulares adaptados ao cenário NFV. Cada módulo da corrente de blocos do SINFONIA é hospedado em um domínio, que possui uma cópia local da corrente na qual pode-se verificar publicamente todas as transações desde sua criação. A estrutura da corrente de blocos do SINFONIA é ilustrada na Figura 3. A principal diferença da estrutura da corrente de blocos do SINFONIA com a estrutura tradicional é que um bloco é dividido em assinatura do conteúdo

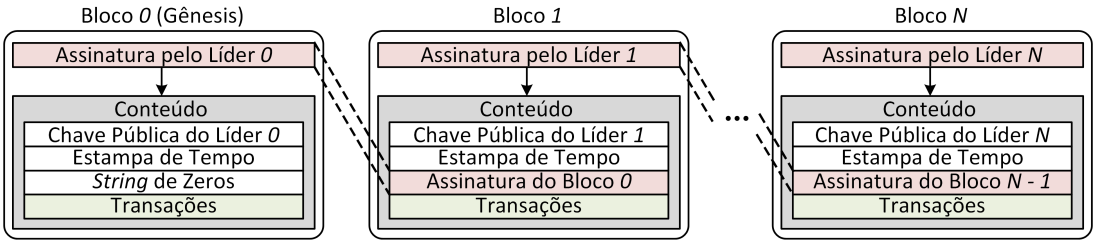


Figura 3: A estrutura da corrente de blocos desenvolvida para o SINFONIA. A assinatura do líder do bloco atual conserva as mesmas propriedades de uma função resumo (*hash*) criptográfica e também garante a autenticidade do bloco.

e conteúdo, e, na seção de conteúdo, é armazenada a chave pública do módulo de corrente de blocos que propôs o bloco e a assinatura do bloco anterior. A inclusão de um esquema de assinatura na corrente de blocos visa proporcionar a auditabilidade do processo de consenso, através da identificação clara do módulo de corrente blocos que propôs o bloco registrado. Ao disponibilizar a chave pública do assinante no conteúdo do bloco, o esquema de assinatura utilizado conserva as mesmas propriedades de segurança de uma função de resumo criptográfica, de forma que é adequado para utilização em corrente de blocos sem perda das propriedades originais. Durante a inicialização do sistema, um banco de dados relacional local contendo índices é construído para facilitar a busca por conteúdo armazenado na corrente de blocos. Este banco de dados é atualizado a cada novo bloco. Cada cliente da corrente de blocos, isto é, inquilinos e orquestradores, possui um par de chaves assimétricas que serve como identificação e permite assinar uma transação. Os módulos de corrente de blocos nunca emitem transações, mas cada módulo de corrente de blocos possui um par de chaves assimétricas exclusivamente para assinatura de blocos e das mensagens de consenso.

Dois tipos de transação são definidas na arquitetura proposta: i) transação de instrução; e ii) transação de resposta. Transações de instrução são emitidas a partir de uma requisição de um inquilino, e são utilizadas para registrar o pedido de escrita na cadeia. Transações de resposta são emitidas apenas pelo orquestrador de nuvem e registram o resultado da instrução solicitada. Uma transação de instrução é assinada pelo inquilino que a propõe e uma transação de resposta é assinada pelo orquestrador que executa a instrução. A Figura 4 mostra a estrutura de cada tipo de transação. Toda transação possui um campo de cabeçalho e um campo de conteúdo. O cabeçalho de cada transação é o mesmo para os dois tipos e contém a assinatura do conteúdo da transação gerada por seu emissor. Desta forma, é possível garantir tanto a autenticidade, pela identificação do assinante, quanto a integridade da transação, pois a assinatura utilizada combina criptografia de chave assimétrica e uma função resumo do conteúdo. Os subcampos de conteúdo comuns aos dois tipos de transação são: i) tipo, que define a categoria da transação; ii) remetente, que identifica o inquilino ou o orquestrador que a emitiu através de sua chave pública; e iii) estampa de tempo, que define o momento em que a transação foi emitida. Uma transação de instrução possui ainda o subcampo de instrução, que define a instrução a ser executada pelo orquestrador. Uma transação de resposta possui três subcampos de conteúdo adicionais: i) transação de origem, que identifica a transação de instrução correspondente àquela transação de resposta através de seu cabeçalho; ii) resultado, que define a resposta do orquestrador à instrução solicitada; e iii) erro, que define e identifica possíveis erros no processo de orquestração. Toda transação de resposta deve possuir uma transação de instrução correspondente, garantindo que a comunicação entre inquilino e orquestrador foi realizada.

A validação de uma transação pelo módulo de corrente de blocos consiste na verificação da: i) assinatura da transação segundo a chave pública do remetente; ii) existência dos sub-

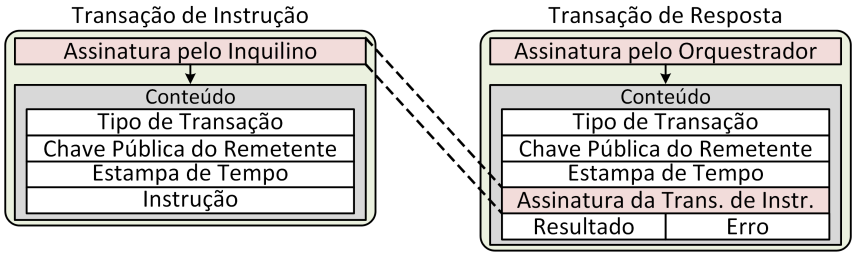


Figura 4: Os dois tipos de transações possíveis na corrente de blocos do SINFONIA: transação de instrução emitida pelo inquilino e transação de resposta emitida pelo orquestrador da nuvem. A transação de resposta é associada à transação de instrução correspondente através da assinatura pelo inquilino.

campos de acordo com o tipo de transação; iii) estampa de tempo de acordo com um limite pré-acordado entre participantes do consenso, evitando que transações futuras ou antigas sejam executadas e iv) existência da transação na corrente de blocos, evitando transações duplicadas. Para transações de instrução, a semântica do subcampo de instrução também é verificada de acordo com seus valores permitidos, evitando que transações arbitrárias sejam validadas. Uma transação inválida é descartada imediatamente. A validação de cada transação é realizada localmente em cada módulo de corrente de blocos. Um participante do consenso vota pela aceitação de um bloco se e somente se cada transação de um bloco é validada com sucesso.

3.2. O Algoritmo de Consenso do SINFONIA

O SINFONIA implementa um protótipo do protocolo de consenso *Practical Byzantine Fault Tolerance* (PBFT), que foi adaptado para o caso do consenso em correntes de blocos. O protocolo PBFT foi selecionado pois promove alto desempenho para um número limitado e conhecido de até algumas centenas de participantes, tornando-o ideal para a aplicação em um cenário NFV. Para prova de conceito do SINFONIA e sua avaliação, é implementado exclusivamente o caso de operação padrão do PBFT. Portanto, não foram consideradas trocas de líder, a atualização de um nó após a falha e as demais situações de contorno do PBFT na implementação do protótipo do SINFONIA. Deve ser ressaltado que as situações excepcionais não implementadas não afetam o objetivo deste artigo, que objetiva uma prova de conceito e a avaliação do desempenho em condições normais. A implementação própria da corrente de blocos e do um protocolo de consenso ainda permite maior entendimento e controle sobre os experimentos realizados, facilitando o isolamento da métrica avaliada.

A sequência de cinco fases do caso de operação padrão do protocolo *Practical Byzantine Fault Tolerant* (PBFT) [Castro et al. 1999] foi adaptada para uma sequência de três fases, ilustradas na Figura 5: i) pré-preparo, na qual o líder do consenso cria um bloco assinado contendo seu novo conjunto de transações e o envia a todos os participantes; ii) preparo, em que cada participante valida as transações localmente e informa sua decisão a todos os demais membros do consenso através de mensagens assinadas; e iii) registro, em que, ao receber mais de dois terços de mensagens assinadas com os respectivos pareceres, cada participante informa aos demais membros do consenso de seu resultado final através de mensagens assinadas. Na fase de registro, todas as mensagens de preparo recebidas por um participante são incluídas na mensagem final como um comprovante do consenso.

Na adaptação do PBFT para corrente de blocos, as fases de requisição e de resposta estão dissociadas da sequência de fases do protocolo de consenso, ocorrendo de forma assíncrona enquanto as demais fases são executadas. A fase de requisição é representada pelo envio de transações dos clientes para um módulo de corrente de blocos. Se este módulo não for

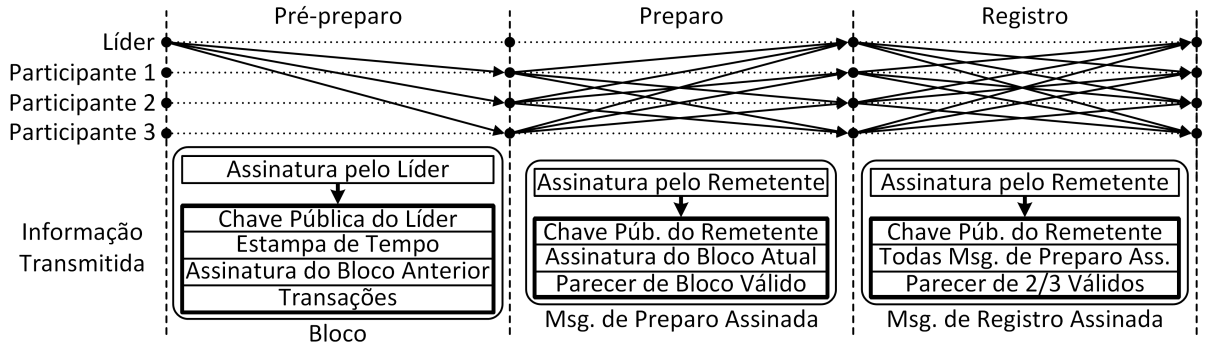


Figura 5: Sequência das 3 fases do caso de operação padrão do protocolo *Practical Byzantine Fault Tolerant* (PBFT) adaptado para corrente de blocos: Pré-preparo; Preparo; e Registro. Após a confirmação de registro, o novo bloco é incorporado à corrente em todos os módulos de corrente participantes.

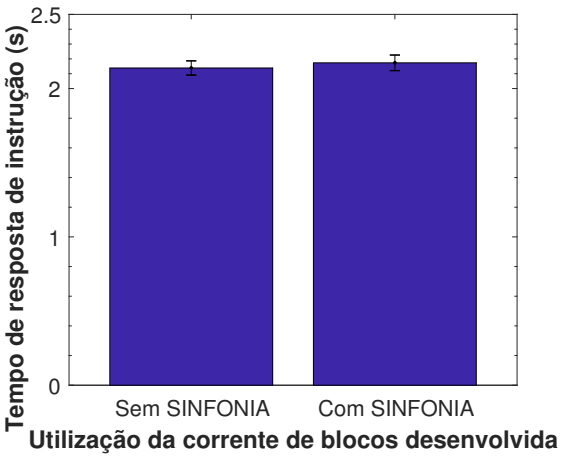
o atual líder do consenso, ele encaminha a transação para o líder. O líder, por sua vez, propõe um bloco que contém um lote de transações recebidas até o início do próximo consenso [Castro e Liskov 2002]. A fase de resposta é representada como uma consulta do cliente à corrente de blocos. Como cada transação é uma operação atômica, um cliente pode verificar a qualquer momento a inclusão de sua transação através da leitura da corrente de blocos.

4. Teste e Avaliação de Desempenho do Protótipo SINFONIA

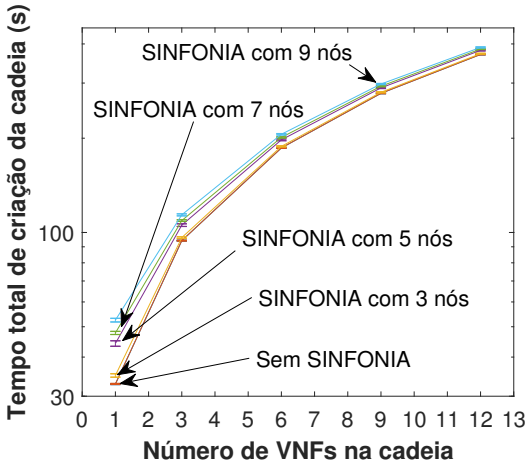
O SINFONIA utiliza a plataforma de código aberto para virtualização de funções de rede *Open Platform for Network Function Virtualization* – OPNFV), versão Danube 3.0. A OPNFV é compatível com arquitetura de virtualização de rede do ETSI, possui o controlador de redes definidas por *software* OpenDaylight e oferece facilidades de orquestração, gerenciamento e virtualização de funções de rede. O encadeamento de funções de rede é feito segundo a arquitetura IETF RFC 7665 [Halpern e Pignataro 2015] e segue a especificação do cabeçalho de funções de serviço (*Network Service Header* – NSH) [Quinn e Elzur 2017]. A avaliação do protótipo do SINFONIA tem como objetivo medir três aspectos da utilização do sistema: i) a sobrecarga gerada pela utilização da corrente de blocos e pela validação das transações no ambiente OPNFV; ii) o desempenho do modelo de transação proposto e do algoritmo de consenso desenvolvido; e iii) o custo de armazenamento introduzido pelas réplicas da corrente de blocos.

Para a avaliação de sobrecarga do sistema, o SINFONIA foi instalado em um servidor Intel 16-core Xeon E5-2650 2 GHz com 32 GB de memória que atua como nó controlador de um ambiente OPNFV instanciado no laboratório do GTA/UFRJ. Os nós de processamento (*compute*) do ambiente consistem em três servidores Intel 8-Core Xeon CPU X5570 2,93 GHz com 96 GB de memória (nó 1), Intel 8-Core i7-6700 CPU, 3.40 GHz com 64 GB de memória (nó 2) e Intel 8-Core i7-2600 CPU, 3,40 GHz com 32 GB de memória (nó 3). Todas as máquinas são interligadas em LAN através de um comutador topo de bastidor (*top of rack*) por interfaces de rede de 1 Gb/s que englobam as cinco redes VLANs necessárias para a instalação da nuvem OPNFV: pública, privada, de gerenciamento, de armazenamento e de *Preboot Execution Environment*. A interface web foi comandada por chamadas HTTP a partir de um computador pessoal. Todas as assinaturas no protótipo são realizadas através de pares de chave Rivest-Shamir-Adleman (RSA) de 2048 bits e de acordo o esquema padrão de assinatura *Public Key Cryptography Standard #1 Probabilistic Signature Scheme* (PKCS#1-PSS) utilizando a biblioteca de criptografia PyCryptodome[3].

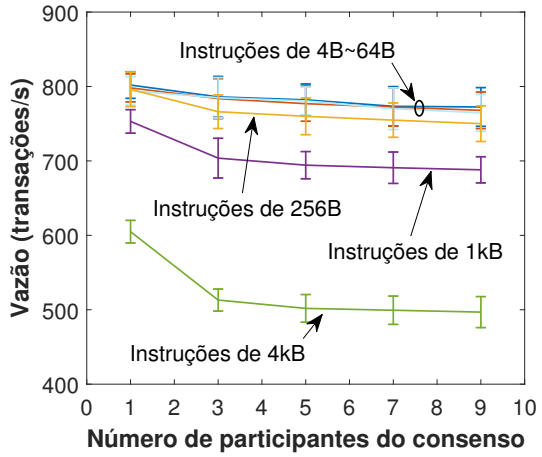
[3]A biblioteca está disponível em https://github.com/Legrandin/pycryptodome.



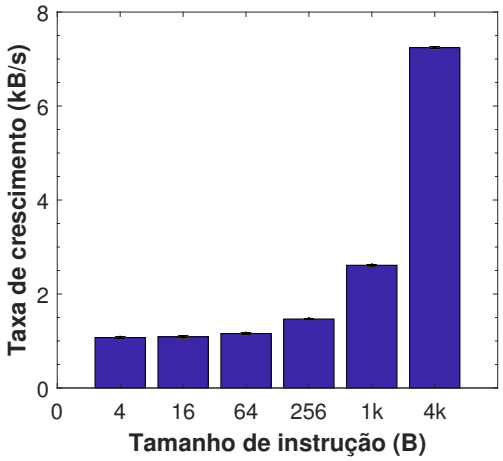
(a) Sobrecarga de comunicação introduzida pela cadeia de blocos sem consenso. Tempo de resposta de uma instrução para os casos sem (à esquerda) e com (à direita) a corrente de blocos desenvolvida.



(b) Tempo de criação de uma cadeia de funções de rede em relação ao número de VNFs para os casos com e sem consenso. Os nós do SINFONIA representam os módulos de corrente de blocos participantes do consenso de cada domínio.



(c) Impacto na vazão de transações por segundo ao aumentar o número de participantes do consenso e o tamanho da instrução. O protótipo do SINFONIA é capaz de processar até 803,3 transações por segundo.



(d) Taxa de crescimento da corrente de blocos ao emitir 100 transações por segundo. A taxa se mantém estável no tempo e atinge aproximadamente 100 kB por segundo para instruções de até 64 B.

Figura 6: Resultados da avaliação do protótipo do sistema SINFONIA. Os resultados mostram que a ferramenta é capaz de prover segurança com baixa sobrecarga no gerenciamento de cadeias de funções.

A primeira avaliação mede a sobrecarga do tempo de processamento entre uma requisição HTTP para a criação de uma cadeia de funções de rede, feita por um inquilino através da interface web, até a sua resposta, isto é, até a confirmação de que a instrução será executada. O objetivo é avaliar o atraso na comunicação entre inquilino e plataforma de nuvem ao utilizar o sistema SINFONIA. Foram comparados cenários sem corrente de blocos e com corrente de blocos sem a realização de consenso, considerando apenas a validação das transações. A Figura 6(a) mostra que, com um intervalo de confiança de 95%, o atraso adicional gerado pela utilização da corrente de blocos é de cerca de 0,06 segundo, ou de 3%, demonstrando que a sobrecarga de comunicação devido ao uso de corrente de blocos não é significativa. Em termos de memória utilizada, o espaço adicional de armazenamento necessário à assinatura das transações e do conteúdo de um bloco, bem como suas etiquetas de cada campo, é de 637 B

para transações de requisição, de 655 B para transações de resposta, e de 859 B para o bloco. Este resultado indica uma alta sobrecarga para instruções de poucos caracteres e demonstra o preço a se pagar pela garantia de autenticidade e integridade das transações. No entanto, estes valores são plausíveis em ambientes de nuvem, no qual considera-se que os recursos de rede, memória e disco são suficientemente grandes. Além disso, a média de transações em um bloco, medida no máximo da vazão de transações por segundo do protótipo, é de 3808 transações, indicando que a sobrecarga de assinatura de um bloco não é significativa.

A Figura 6(b) mostra o tempo de criação de uma cadeia de funções de rede quando o número de VNFs que a compõem cresce. Todas as VNFs foram instanciadas no mesmo nó de processamento para minimizar a sobrecarga gerada pela plataforma OPNFV [Sanz et al. 2017]. Os resultados mostram uma dependência direta entre tempo de encadeamento e o número de VNFs a serem encadeadas. Isto pode ser explicado pelo fato de que, antes do encadeamento, o ambiente OPNFV cria cada VNF a partir de uma imagem em máquina virtual, e este processo é feito sequencialmente. Em comparação ao caso apenas com validação de transações, ou seja, sem realização de consenso, a introdução da corrente de blocos com consenso PBFT gera um atraso de até 20 segundos, para o caso de 9 participantes. Em contrapartida, o tempo médio de criação de uma VNF é de 30,1 segundos. Isto demonstra que a sobrecarga gerada pelo consenso nas operações de orquestração é inversamente proporcional ao tamanho da cadeia e que, consequentemente, o termo dominante para cadeias de funções longas é seu tempo de instanciamento. Portanto, a corrente de blocos é melhor aplicada em cadeias mais longas, onde mais VNFs estão envolvidas no encadeamento.

Para a avaliação de desempenho da arquitetura proposta, os módulos de corrente de blocos e o módulo de orquestração foram instanciados como processos isolados na mesma máquina física com um núcleo dedicado e 4 GB de memória RAM para cada processo. A máquina hospedeira dos processos é um servidor Intel 32-core Xeon CPU E5-2650 2.00 GHz com 192 GB de memória RAM. A Figura 6(c) mostra o impacto da variação do tamanho das instruções de gerenciamento na vazão do sistema. Os tamanhos foram selecionados considerando o tamanho típico das chamadas ao controlador OPNFV, que são comandos em Linux que utilizam um *byte* para codificar um caractere. Chamadas ao controlador do OPNFV incluem a instrução e seus termos opcionais que podem apontar para um modelo de configuração disponibilizado no controlador, por exemplo, "`opnfv create-vnf --config=vnf.conf`". Exceto por casos específicos na documentação, como o envio de conteúdo de arquivo por linha de comando, uma instrução não possui mais de dezenas de caracteres. Para avaliar a vazão, são consideradas apenas as transações de instrução dos inquilinos, uma vez que os dois tipos de transação possuem tamanhos similares. Os resultados indicam que a vazão do protótipo, em transações por segundo, permanece estável com o aumento no número de participantes do consenso para todos os tamanhos de instrução, exceto quando comparado ao caso sem realização de consenso, no qual não existe troca de mensagens na rede. A vazão também se mantém estável com o aumento do tamanho das instruções para instruções de até 64 B de tamanho. O limite superior da vazão, de até 803,3 transações por segundo, é imposto pelo valor mínimo entre a capacidade de processamento do líder, que inicia a rodada de consenso de um bloco, e a capacidade de processamento dos participantes, que devem validar o bloco durante o consenso.

A Figura 6(d) mostra a taxa de crescimento da corrente de blocos para diferentes tamanhos de instrução. Uma taxa fixa de 100 transações por segundo foi utilizada para simular um ambiente onde as transações representam o número de pedidos de encadeamento pelos inquilinos. Os resultados mostram que, para instruções com até 64 B, a taxa de crescimento da corrente de blocos é da ordem de 100 kB/s por participante do consenso e cresce significativamente para

instruções acima de 256 B. A invariância na taxa de crescimento para transações com menos de 64 B ocorre devido à sobrecarga gerada pelas assinaturas, que produz um tamanho mínimo de transação e de bloco. Em todos os casos, a taxa de crescimento é estável, demonstrando que a corrente de blocos cresce linearmente. Medidas utilizando taxas de 10 e 500 transações por segundo tiveram resultados semelhantes de acordo com suas ordens de grandeza.

5. Conclusão

A tecnologia de virtualização de funções de rede provê serviços fim-a-fim encadeando funções virtualizadas entre diferentes infraestruturas de nuvem sem garantia de confiança entre os pares. Neste cenário, é imprescindível identificar a ocorrência de falhas e responsabilizar agentes maliciosos, que podem comprometer a segurança de milhares de usuários que usufruem do serviço simultaneamente. Esse artigo propõe o SINFONIA, um sistema baseado em corrente de blocos (*blockchain*) que provê a segurança necessária para orquestração de funções de rede em um ambiente multi-domínio e multi-inquilino. O sistema SINFONIA permite a construção segura de cadeias de funções virtualizadas de rede através da auditabilidade de todas as operações de gerenciamento de uma cadeia. Uma corrente de blocos e um modelo de transações particulares adaptados ao cenário NFV foram desenvolvidos e implementados. O caso padrão de operação de um protocolo de consenso tolerante a falhas bizantinas foi implementado, por promover um consenso com baixa latência e ser robusto a comportamento malicioso de até um terço dos participantes do consenso. Um protótipo do SINFONIA foi implementado na versão Danube 3.0 da *Open Plataform for Network Funtion Virtualization* – (OPNFV) que é conforme as especificações de NFV do ETSI. Os resultados mostram que o atraso resultante da utilização da corrente de blocos sem consenso não é significativo, cerca de 3% em média. Ainda, a vazão do sistema mantém-se estável com o aumento do número de participantes do consenso e com o aumento do tamanho da instrução emitida para instruções de até 64 B. Como trabalhos futuros, pretende-se desenvolver e formalizar o protocolo de consenso implementado e estender o mecanismo de corrente de blocos para garantir consistência na presença de falhas bizantinas entre controladores de redes definidas por software.

Referências

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

Bozic, N., Pujolle, G. e Secci, S. (2017). Securing virtual machine orchestration with blockchains. Em *2017 1st Cyber Security in Networking Conference*.

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

Cachin, C. e Vukolić, M. (2017). Blockchains consensus protocols in the wild. *arXiv preprint arXiv:1707.01873*.

Castro, M. e Liskov, B. (2002). Practical byzantine fault tolerance and proactive recovery. *ACM Trans. Comput. Syst.*, 20(4):398–461.

Castro, M., Liskov, B. et al. (1999). Practical byzantine fault tolerance. Em *OSDI*, volume 99, páginas 173–186.

Doelitzscher, F., Ruebsamen, T., Karbe, T., Knahl, M., Reich, C. e Clarke, N. (2013). Sun behind clouds-on automatic cloud security audits and a cloud audit policy language. *International Journal on Advances in Networks and Services*, 6(1-2):1–16.

ETSI (2014). ETSI GS NFV-MAN 001: Network functions virtualisation; management and orchestration. Relatório técnico.

Halpern, J. e Pignataro, C. (2015). Service Function Chaining (SFC) architecture. RFC 7665, RFC Editor. http://www.rfc-editor.org/rfc/rfc7665.txt. Acessado em 18 de dezembro de 2017.

Kwon, J. (2014). Tendermint: Consensus without mining. https://tendermint.com/static/docs/tendermint.pdf. Acessado em 21 de março de 2018.

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

Miller, A., Xia, Y., Croman, K., Shi, E. e Song, D. (2016). The honey badger of BFT protocols. Em *ACM Conference on Computer and Communications Security*, páginas 31–42. ACM.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. http://bitcoin.org/bitcoin.pdf. Acessado em 18 de dezembro de 2017.

Pattaranantakul, M., He, R., Meddahi, A. e Zhang, Z. (2016). SecMANO: Towards network functions virtualization (NFV) based security management and orchestration. Em *IEEE Trustcom/BigDataSE/ISPA*, páginas 598–605.

Quinn, P. e Elzur, U. (2017). Network service header. Internet-Draft draft-ietf-sfc-nsh-12, IETF Secretariat. http://www.ietf.org/internet-drafts/draft-ietf-sfc-nsh-12.txt. Acessado em 18 de dezembro de 2017.

Raman, L. (1998). OSI systems and network management. *IEEE Communications Magazine*, 36(3):46–53.

Rübsamen, T., Pulls, T. e Reich, C. (2016). *Security and Privacy Preservation of Evidence in Cloud Accountability Audits*, páginas 95–114. Springer International Publishing, Cham.

Sanz, I. J., Alvarenga, I. D., Lopez, M. A., Mauricio, L. A. F., Mattos, D. M. F., Rubinstein, M. e Duarte, O. C. M. B. (2017). Uma avaliação de desempenho de segurança definida por software através de cadeias de funções de rede. Em *XVII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais – SBSeg 2017*.

Sanz, I. J., Mattos, D. M. F. e Duarte, O. C. M. B. (2018). SFCPerf: An automatic performance evaluation framework for service function chaining. Em *IEEE/IFIP Network Operations and Management Symposium – NOMS 2018*. A ser publicado.

Sekar, V., Egi, N., Ratnasamy, S., Reiter, M. K. e Shi, G. (2012). Design and implementation of a consolidated middlebox architecture. Em *9th Symposium on Networked Systems Design and Implementation (NSDI)*, páginas 323–336, San Jose, CA. USENIX.

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

Vukolić, M. (2016). *The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication*, páginas 112–125. Springer International Publishing, Cham.

Yellick, J. (2018). Hyperledger Fabric developer mailing list. https://lists.hyperledger.org/pipermail/hyperledger-fabric/2018-March/003029.html. Acessado em 9 de março de 2018.

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