Revisitando o Bootstrapping Seguro na IoT: Abordagens do TCP/IP e Oportunidades com NDN

Adriana Viriato Ribeiro[1], **André L. R. Madureira**[1], **Leobino N. Sampaio**[1]

[1]Instituto de Computação
Universidade Federal da Bahia (UFBA) – Salvador – BA – Brasil

{adrianavr,andre.romano,leobino}@ufba.br

Abstract. *Security in IoT applications relies on bootstrapping to establish secure communications between devices and applications. In TCP/IP architectures, the bootstrapping and network configuration processes are decoupled. In contrast, Named-Data Networking (NDN) natively integrates this functionality, simplifying management and ensuring security from the beginning of the connection. In this work, we compare the challenges and methodologies of bootstrapping in IoT, analyzing specificities of TCP/IP and NDN. In addition to the theoretical review, the paper also provides a proof of concept to evaluate the viability of bootstrapping in NDN, considering node ingress time and resource consumption. The results indicate low temporal overhead and bandwidth impact associated with the security bootstrapping steps, although factors such as congestion and packet loss can influence its performance. It was also observed that NDN still relies on TCP/IP security techniques, especially in remote bootstrapping scenarios.*

Resumo. *A segurança em aplicações IoT depende do bootstrapping para estabelecer comunicações seguras entre dispositivos e aplicações. Na arquitetura TCP/IP, os processos de bootstrapping e configuração de rede estão dissociados. Em contraste, as Redes de Dados Nomeados (do inglês, Named-Data Networking – NDN) integram essa funcionalidade nativamente, simplificando o gerenciamento e garantindo segurança desde o início da conexão. Neste trabalho, comparamos os desafios e metodologias de bootstrapping em IoT, analisando especificidades do TCP/IP e NDN. Além da revisão teórica, o artigo também traz uma prova de conceito para avaliar a viabilidade do bootstrapping em NDN, considerando tempo de ingresso de nós e consumo de recursos. Os resultados indicam baixa sobrecarga temporal e impacto na largura de banda atrelados às etapas de bootstrapping de segurança, embora fatores como congestionamento e perda de pacotes possam influenciar seu desempenho. Também se observou que a NDN ainda depende de técnicas de segurança desenvolvidas para TCP/IP, especialmente em cenários de bootstrapping remoto.*

1. Introdução

O processo de *bootstrapping* é composto por procedimentos essenciais que antecedem a operação de um sistema IoT, incorporando desde parâmetros de configuração básica (e.g., *gateway* e servidor DNS) até aspectos de segurança (e.g., troca de chaves criptográficas e âncoras de confiança). Esse procedimento varia conforme o escopo da rede, o contexto

da aplicação e o caso de uso específico, resultando em diferentes implementações. Nas redes IoT, o *bootstrapping* visa garantir a comunicação segura entre dispositivos, mediante o estabelecimento de relações de confiança e mecanismos de segurança, tais como criptografia de dados, autenticação, não repúdio e assinaturas digitais [Malik et al. 2019].

Na arquitetura TCP/IP, o *bootstrapping* limita-se à configuração básica necessária para a troca de dados entre dispositivos, utilizando protocolos como o *Dynamic Host Configuration Protocol* (DHCP). No entanto, quando uma rede exige comunicação segura, etapas adicionais são necessárias para estabelecer parâmetros de segurança, resultando em um processo complementar de *bootstrapping* de segurança. Nesses cenários, o dispositivo entra em operação somente após a conclusão dos procedimentos necessários para estabelecer os parâmetros de segurança na rede. Nesse sentido, estudos recentes [Malik et al. 2019, Li et al. 2019, Tao et al. 2020, Ramani et al. 2020, Yu et al. 2023a, Yu et al. 2023b, Sethi et al. 2025] têm investigado estratégias de *bootstrapping* de segurança tanto em redes TCP/IP quanto nas redes de dados nomeados (do inglês, *Named-Data Networking* – NDN).

Arquiteturas como a NDN tem despertado interesse da comunidade científica pois se propõem a prover segurança de forma nativa e porque os *bootstrapping* de rede e segurança ocorrem de forma integrada. No entanto, essa integração pode fazer com que a NDN introduza carga computacional adicional na rede e nos dispositivos. Consequentemente, foram desenvolvidas adaptações para redes IoT [Zhang et al. 2018, Hail 2019, Wang et al. 2021], visando otimizar o desempenho em dispositivos com recursos limitados. Apesar disso, até o presente momento, o impacto das etapas de *bootstrapping* de segurança da NDN nas redes IoT ainda não foi adequadamente investigado. Assim, este trabalho realiza uma análise teórica comparativa das soluções de *bootstrapping* em redes TCP/IP e NDN, além de apresentar uma prova de conceito com uma avaliação inicial do desempenho e viabilidade do *bootstrapping* de segurança da arquitetura NDN em redes IoT. Assim, este trabalho visa responder às seguintes questões de pesquisa:

- **Q1**: Como as funcionalidades e características da arquitetura NDN se comparam às soluções existentes para redes TCP/IP?
- **Q2**: Quais são as etapas fundamentais para a realização do *bootstrapping* de segurança em IoT, independentemente da arquitetura de rede adotada?
- **Q3**: É viável a integração dos processos de *bootstrapping* de rede e segurança?
- **Q4**: Qual é o impacto da incorporação das etapas do *bootstrapping* de segurança no comportamento temporal (desempenho) da rede?

Como resultado da análise das questões acima, este artigo apresenta as seguintes contribuições: (i) descreve as funcionalidades e características das arquiteturas IP e NDN; (ii) compara os processos de *bootstrapping* de segurança em ambas as arquiteturas, identificando suas similaridades e etapas fundamentais; e (iii) avalia o impacto no desempenho da rede, bem como a viabilidade da integração das etapas de *bootstrapping* em um procedimento unificado. A discussão de cada uma dessas questões está distribuída ao longo das seções do artigo, conforme descrição a seguir: a Seção 2 compara as arquiteturas TCP/IP e NDN, analisando as suas diferenças com relação ao *bootstrapping* de segurança (Q1); a Seção 3 analisa os procedimentos de *bootstrapping* em ambas as arquiteturas, destacando as etapas comuns (Q2); e a Seção 4 discute a viabilidade da unificação do *bootstrapping* de rede e segurança (Q3), utilizando para isto uma análise experimental do desempenho do *boostrapping* NDN para redes IoT, construído sob a forma de prova de conceito (Q4).

2. Fundamentos de Segurança em Redes de Dados Nomeados (NDN)

Nas redes TCP/IP, a entrega de dados é baseada no endereço IP, que atua como identificador e localizador de dispositivos. Esse modelo baseia-se na premissa de que cada nó deve ter pelo menos um endereço IP. Entretanto, outras arquiteturas, como a NDN, usam nomes hierárquicos em vez de endereços, permitindo o encaminhamento baseado no conteúdo entre produtores (detentores dos dados) e consumidores (interessados nas informações). Além da identificação de conteúdos, os nomes na NDN possuem diversas aplicações, abrangendo desde a identificação de usuários até a descrição de comandos para dispositivos IoT. Para isso, os nomes na NDN seguem a estrutura de *Uniform Resource Identifiers* (URIs), organizados hierarquicamente através de componentes separados por barras ($/$). Essa hierarquia proporciona vantagens no quesito escalabilidade do roteamento, permitindo a agregação eficiente de pacotes por prefixos [Sampaio et al. 2021, Yu et al. 2022b].

Dois outros aspectos se destacam na nomeação NDN: i) semântica expressiva e rica; e ii) flexibilidade de unicidade de nomeação. Primeiramente, os nomes podem conter semântica no nível da aplicação, permitindo que o nome expresse significado contextual (e.g., $/ufba/medicina/hospital/sala/B/sensor/temperatura/2930$ pode identificar um sensor específico na estrutura de um hospital universitário). Em segundo lugar, nem todos os nomes precisam ser globalmente únicos, de maneira equivalente ao funcionamento dos endereços privados na arquitetura TCP/IP. Já a distribuição de prefixos globais únicos na NDN deve ser coordenada por entidades competentes, de maneira similar à distribuição de domínios no IP realizada por Registros Regionais da Internet. Por exemplo, suponha um domínio `ufba.br`, gerido pela Universidade Federal da Bahia. Neste caso, a instituição é responsável por gerenciar todos os subdomínios associados a `ufba.br`, estando outros domínios sob administração de outras instituições ou entidades. Na NDN, o prefixo `/br/ufba` seria administrado da mesma forma, facilitando a verificação da integridade e autenticidade dos dados.

O processo de validação na NDN também possui paralelos com as redes TCP/IP. O controlador de um domínio NDN gera uma chave pública assinada por uma Autoridade Certificadora (do inglês, *Certificate Authority* – CA) externa e uma chave autoassinada, que atua como âncora de confiança (*trust anchor – TA*) [Yu et al. 2023b]. A autenticidade do conteúdo é garantida através de campos de segurança nos pacotes, como o *KeyLocator*, que identifica a chave pública utilizada para assinar cada pacote. O consumidor verifica este campo e solicita a chave correspondente, garantindo a integridade e confidencialidade do conteúdo. Essa chave é distribuída como um pacote de dados comum na NDN, cuja obtenção segue o mesmo processo de requisição-resposta utilizado pelos demais pacotes na rede . Assim, ao receber um pacote de dados, o consumidor solicita o pacote correspondente à chave indicada no *KeyLocator*, repetindo esse processo recursivamente até alcançar a âncora de confiança do domínio (Figura 1). Quando isto ocorre, o pacote e a sua chave associada são validados. Logo, a segurança deste processo de validação é lastreada na cadeia de certificados terminada na âncora de confiança, de forma similar ao que ocorre com certificados X.509 em redes TCP/IP [Ullah et al. 2021].

Contudo, uma cadeia de certificados válida não garante a legitimidade do produtor. Por exemplo, considere que um subprefixo `/br/ufba/medicina`, delegado para o Instituto de Medicina, assina o certificado de outro subdomínio, como `/br/ufba/computacao`, vinculado ao Instituto de Computação. Embora a cadeia

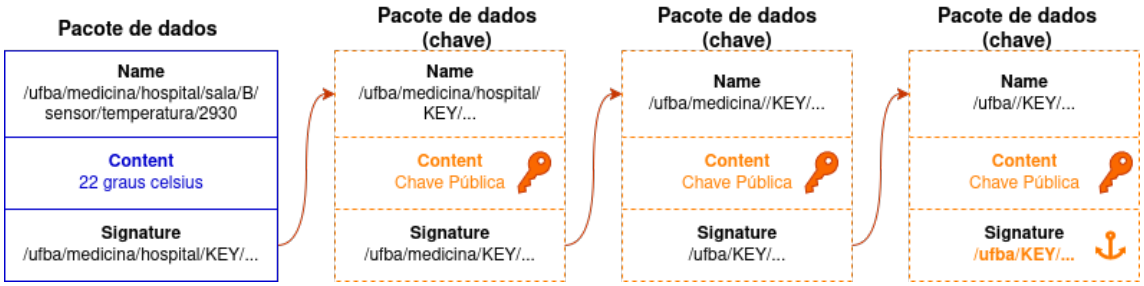


Figura 1. Cadeia de confiança de validação do conteúdo.

de certificados `/br/ufba/computacao` → `/br/ufba/medicina` → `/br/ufba` seja válida, isto não significa que a mesma esteja coerente com as políticas de segurança da instituição. Um exemplo de incongruência seria a exigência de que os certificados de cada instituto sejam assinados diretamente pelo administrador do domínio. Para evitar violações de políticas como a descrita anteriomente, a NDN adota esquemas de confiança (*trust schemas* – TS), que definem regras explícitas para autenticação de produtores dentro de um domínio. Consequentemente, um TS atua como insumo para validação de cadeias de certificados, pois descreve as políticas de confiança e a ordem de validação destas. A implementação desses esquemas, incluindo a definição de políticas, distribuição de certificados e configuração de âncoras de confiança, constitui uma etapa crucial do *bootstrapping* em NDN, conforme detalhado nas seções subsequentes.

Como considerações finais, ressaltamos que a NDN introduz um paradigma centrado no conteúdo, no qual nomes hierárquicos substituem endereços. Quando comparado ao TCP/IP na questão da segurança, ambas as arquiteturas dependem de infraestruturas de chave pública (do inglês, *Public Key Infrastructure* – PKI) para autenticação e garantia da integridade dos dados. No entanto, a NDN aprimora esse aspecto ao incorporar mecanismos como âncoras e esquemas de confiança, baseados no *KeyLocator* e cadeias de certificação, possibilitando a automação da validação e a autenticidade dos pacotes.

3. Processo de Bootstrapping de segurança

O processo de *bootstrapping* de segurança envolve etapas fundamentais, como autenticação mútua e troca de material criptográfico entre dispositivos, usuários e aplicações. Em redes TCP/IP, esse processo ocorre após o *bootstrapping* da rede e, frequentemente, requer acesso à Internet para a troca de informações essenciais. Em contrapartida, na NDN, o *bootstrapping* da rede já incorpora mecanismos básicos de segurança, garantindo requisitos como integridade do conteúdo e autenticidade do produtor nativamente. Nesta seção, esses processos serão discutidos e comparados, destacando seus benefícios e desafios.

3.1. Bootstrapping de segurança em redes TCP/IP

As redes TCP/IP visam a comunicação entre dispositivos, sem oferecer garantias de segurança, deixando esses aspectos a cargo dos desenvolvedores de aplicações. Para estabelecer comunicação, os nós devem ser configurados de forma estática ou dinâmica (e.g., DHCP). No entanto, quando a aplicação requer propriedades de segurança, um segundo *bootstrapping* faz-se necessário. Este procedimento, chamado de *bootstrapping* de segurança, é composto por três etapas essenciais: i) autenticação mútua, ii) distribuição de credenciais e iii) configuração de canais seguros de comunicação.

A autenticação mútua envolve a verificação das identidades do *bootstrapper* (provê o serviço de *bootstrapping*) e do *bootstrapee* (quem esta entrando na rede). A Tabela 1 apresenta soluções comuns para verificação de identidade, que empregam diversos mecanismos de autenticação, como algoritmos criptográficos e autenticação por senha, podendo estes ser combinados para garantia de mais propriedades de segurança. Em cenários IoT, a eficiência desses mecanismos se torna crucial, uma vez que muitas das soluções presentes na literatura (e.g., IPSec) possuem elevada demanda por recursos computacionais. Esta característica pode tornar uma implementação de segurança inviável, dada a limitação de recursos dos dispositivos IoT. Além de considerar o consumo de recursos, a viabilidade de soluções de segurança em IoT também depende da sobrecarga computacional (i.e., complexidade) dos protocolos adotados nesta etapa [Malik et al. 2019].

Tabela 1. Mecanismos de Autenticação de dispositivos.

Mecanismo	Implementação	Limitações	Exemplos
Chaves Pré-configuradas	Chaves previamente configuradas (*in-band* ou *out-of-band*).	Risco de comprometimento de chaves, maior para chaves simétricas.	IPsec com PSK [Frankel and Krishnan 2011]
Autenticação por PKI	Certificados emitidos por CA, provendo identidade aos nós.	Confiança em nó intermediário. Escalabilidade limitada. Alta complexidade.	mTLS [Rescorla and Dierks 2008]
Autenticação por PSK	Autenticação por senha ou *token* previamente configurados.	Vulnerável a ataques de força bruta (e.g., *phishing*, *keylogger*, *MITM*).	TLS-PSK [Tschofenig and Eronen 2005]
Prova Criptográfica	Identidade validada sem envio de informações sensíveis.	Exige etapa de configuração para distribuir chaves, ou se baseia em soluções de PKI.	OPAQUE [Bourdrez et al. 2022]
Autenticação Federada	Confiança baseada em entidade intermediária (similar a PKI).	Confiança em intermediário. Autenticação restrita à entidades da federação.	OIDC [Siriwardena 2020]

Após a autenticação mútua, inicia-se a distribuição de credenciais, cuja implementação varia conforme os requisitos de segurança da aplicação e a capacidade dos dispositivos. Por exemplo, a criptografia assimétrica é adequada para autenticação, integridade das mensagens e troca de chaves simétricas. Enquanto a criptografia simétrica é comumente utilizada para garantir a confidencialidade dos dados em trânsito de maneira eficaz (e.g., TLS). Após a distribuição de credenciais, realiza-se a configuração do canal seguro de comunicação, onde o *bootstrapper* e o *bootstrapee* negociam algoritmos criptográficos e estabelecem as chaves de sessão necessárias para assegurar uma comunicação segura [Li et al. 2019]. Exemplos de mecanismos de distribuição de credenciais e configuração de canal seguro estão sumarizados nas Tabelas 2 e 3.

Devido às limitações de recursos em IoT, a comunidade científica tem proposto protocolos mais leves para autenticação mútua [Rai et al. 2023, Malamas et al. 2025] e para as demais etapas do *bootstrapping* de segurança. Em [Kumar et al. 2022], é sugerido um conjunto de serviços de segurança que inclui um mecanismo leve de autenticação baseado em biometria e um *bootstrapping* em três camadas: local (serviços básicos), borda (segurança intermediária) e global (garantias mais robustas, com maior uso de recursos). Outra abordagem, descrita em [Tao et al. 2020], propôs um *bootstrapping* de segurança

Tabela 2. Soluções para Distribuição de Credenciais.

Solução	Implementação	Limitações	Exemplos
Derivação de chave	A chave de sessão do nó é derivada de uma chave compartilhada.	Risco de comprometimento de chave compartilhada, *phishing*, *MITM*.	ECDH [Nath and Sarkar 2020]
Distribuição centralizada de chaves	Servidor KDC distribui credenciais de sessão para cada nó.	Escalabilidade limitada. Ponto único de falha (servidor KDC).	Kerberos [Neuman and Ts'o 1994]
Certificados PKI	Servidor PKI distribui e renova certificados digitais.	Problema de revogação de certificados. Ponto único de falha (servidor PKI).	x.509 [Boeyen et al. 2008]
Tokens temporários	Autoridade emissora (AS) distribui tokens, com validade limitada.	Risco de *phishing*, *MITM*. Ponto único de falha (comprometimento do AS).	JWT [Jones et al. 2015]
Credenciais baseadas em atributos	Autenticação contínua, baseada em atributos verificáveis.	Alto consumo de recursos. Verificação e revogação de atributos complexos.	IRMA [Alpár et al. 2017]

no nível da camada de aplicação, capaz de realizar a troca das credenciais entre dispositivos sem necessitar de infraestruturas pré-existentes que possibilita a configuração de canais de comunicação seguros entre dispositivos (D2D) em redes IoT sem fio dinâmicas, com recursos limitados e presença de usuários maliciosos. [Danilchenko et al. 2019] segue uma outra vertente de investigação, com enfase na automatização da instalação de certificados de CA, através do protocolo TLS e de canais não seguros.

Tabela 3. Mecanismos de Configuração de Canal Seguro.

Proposta	Implementação	Limitações	Exemplos
Tunelamento	Trafego criptografado fim-a-fim através de túneis.	Complexidade (e.g., IPsec + NAT). Escalabilidade e ponto único de falha (e.g., servidor).	IPsec [Frankel and Krishnan 2011]
Protocolos de Transporte	Trafego criptografado fim-a-fim na camada de transporte.	Vulnerável à *phishing*, *MITM*. Pontos únicos de falha (e.g., servidores).	SSL/TLS [Rescorla 2018]
Protocolos de Aplicação	Trafego criptografado fim-a-fim na camada de aplicação.	Incompatibilidade entre aplicações. Cabeçalhos de protocolos inferiores desprotegidos.	DNS over HTTPS [Hoffman and McManus 2018]
Negociação dinâmica de chaves	Chave secreta negociada sobre um canal inseguro.	Alto consumo de recursos e baixo desempenho. Vulnerável a ataques *MITM*.	IKEv2 [Kaufman 2005]
Certificado Autoassinado	Identidade derivada de uma chave pública autoassinada.	Problema de revogação de identidades. Vulnerável à ataques de *spoofing*, *MITM*.	x.509 [Boeyen et al. 2008]
Segurança baseada em SDN	Plano de controle da rede define regras para canais seguros.	Ponto único de falha (controlador SDN). Escalabilidade limitada. Vulnerável à *DDoS*.	OpenFlow Security [Klöti et al. 2013]

Adicionalmente, algumas propostas para redes IoT já vêm sendo analisadas e padronizadas pela IETF há algum tempo, como o *Ephemeral Diffie-Hellman Over COSE*

(EDHOC) [Selander et al. 2024] e o *Nimble Out-of-Band Authentication for EAP* (EAP-NOOB) [Sethi et al. 2025]. O EDHOC é um protocolo leve de troca de chaves autenticadas, projetado para dispositivos altamente restritos, utilizando CBOR e COSE em substituição às estruturas convencionais (e.g., ASN.1, X.509). Contudo, ele ainda requer infraestrutura de chaves (e.g., PKI, PSK) e sincronização de tempo ou uso de *nonces*, o que limita o escopo de uso da proposta. Já o EAP-NOOB é uma adaptação do protocolo *Extensible Authentication Protocol* (EAP) para autenticação de dispositivos IoT sem o uso de canais *out-of-band*. Apesar de viabilizar o provisionamento dinâmico de dispositivos sem certificados, o protocolo demanda interação humana, pois utiliza autenticação mútua por geração de chaves temporárias. Assim, o EAP-NOOB não é adequado para cenários de *bootstrapping* remoto, nos quais dispositivos IoT podem estar fisicamente inacessíveis.

3.2. Bootstrapping de segurança em redes NDN

NDN adota um roteamento baseado em nome, comunicação *publish/subscribe*, armazenamento em cache, e segurança a nível de dados [Sampaio et al. 2021], garantindo integridade, autenticidade e não-repúdio do conteúdo [Mirajkar et al. 2024]. Para prover esses recursos, o *bootstrapping* na NDN necessita de quatro etapas principais: i) autenticação mútua, ii) instalação da âncora de confiança, iii) instalação do esquema de confiança e iv) atribuição de nome e certificado ao produtor [Yu et al. 2023b]. Assim como nas redes TCP/IP, o *bootstrapping* na NDN começa com a autenticação mútua entre o *bootstrapper* e o *bootstrapee*, permitindo a validação recíproca das identidades. O papel do *bootstrapper* nesse processo é assegurar que apenas entidades autorizadas possam integrar o domínio, enquanto o *bootstrapee* deve validar a autenticidade do *bootstrapper*. A verificação dessas identidades é realizada por mecanismos diversos [Li et al. 2019, Ramani et al. 2020, Yu et al. 2023a], conforme descritos na Tabela 1, podendo adotar princípios semelhantes aos empregados na arquitetura TCP/IP.

Após a autenticação mútua, o *bootstrapee* pode solicitar ao controlador da zona o envio de seu certificado autoassinado (âncora de confiança) [Zhang et al. 2017]. Nos casos em que essa etapa ocorre *out-of-band*, os dispositivos não precisam realizar a solicitação do certificado, pois ele já está previamente instalado. Com as entidades autenticadas e a âncora de confiança instalada, o *bootstrapee* inicia a instalação do esquema de confiança. O esquema de confiança é obtido por meio de requisição-resposta, da mesma forma que ocorre com os demais pacotes NDN. Após essa instalação, o nó possui os requisitos necessários para validar pacotes de dados no domínio [Yu et al. 2023a]. Contudo, para ser capaz de produzir conteúdo, são executadas as etapas de atribuição de nome e emissão de certificado à entidade: o *bootstrapee* solicita o nome e o certificado, conforme as diretrizes do esquema de confiança, restrições de segurança e semântica da aplicação.

O *bootstrapping* na NDN tem sido explorado através de diferentes estratégias, tais como as baseadas em TCP/IP e suas adaptações para ambientes IoT, além da automação por meio da semântica de nomeação. Assim, em [Yu et al. 2023b], foi proposto um protocolo de *bootstrapping* remoto para dispositivos e usuários na NDN, utilizando autenticação mútua com tecnologias da arquitetura TCP/IP, como PKI e autenticação federada. A automação das etapas subsequentes foi realizada através do *Name Authentication and Assignment Protocol* (NAAP), no qual, após a autenticação mútua, a entidade acessa o repositório do GitLab do controlador da zona para obter a âncora e o esquema de confiança do domínio, além de atribuir nomes baseados nos identificadores da Internet atual, como

nomes DNS para servidores e endereços de e-mail para usuários. Já em [Li et al. 2019], os autores desenvolveram o *Secure Sign-On Protocol* (SSP), um protocolo de *bootstrapping* voltado para dispositivos com recursos limitados. O protocolo incorpora estratégias para prevenção e detecção de ataques, utilizando chaves simétricas e assimétricas pré-configuradas para autenticação mútua e obtenção da âncora de confiança. Para dispositivos com recursos limitados, o protocolo permite que o controlador da zona gere os pares de chaves e envie a chave privada criptografada junto ao certificado. Além disso, uma versão adaptada do protocolo foi desenvolvida para dispositivos com maior capacidade.

Em [Ramani et al. 2020], os autores propõem o NDNViber, uma estratégia inovadora para *bootstrapping* de dispositivos IoT com recursos limitados, fisicamente inacessíveis. O NDNViber realiza o *bootstrapping out-of-band* por meio da transmissão de pacotes utilizando modulação de vibrações de ondas sonoras, o que permite a configuração simultânea de múltiplos dispositivos sem interferir no meio de transmissão de dados. Essa abordagem utiliza a semântica do esquema de nomeação para automatizar a troca de mensagens de segurança e utiliza o protocolo NDNCert [Zhang et al. 2017] para a verificação automatizada da cadeia de certificados. Assim, o NDNViber proporciona uma solução robusta para a autenticação e configuração segura de dispositivos IoT, mesmo em cenários onde os recursos são limitados e a acesso físico aos dispositivos é limitado.

3.3. Comparação do *Bootstrapping* de Segurança em Arquiteturas TCP/IP e NDN

O *bootstrapping* de segurança em arquiteturas TCP/IP e NDN apresenta tanto semelhanças quanto diferenças. A Tabela 4 apresenta uma comparação dos principais aspectos desses processos. O principal ponto de distinção entre as abordagens é que, conforme discutido anteriormente, o *bootstrapping* de segurança na NDN está intrinsecamente integrado ao *bootstrapping* da própria rede, uma vez que a arquitetura fornece segurança nativamente. Esse aspecto inicial está relacionado a dois pontos relevantes de investigação para redes NDN: i) suporte nativo limitado a integridade e autenticidade, e ii) desafios na implementação de mecanismos de autenticação. O primeiro ponto de investigação esta relacionado à ausência de suporte nativo a outras propriedades de segurança para além da integridade e autenticidade em redes NDN. Por exemplo, embora a arquitetura permita a implementação de estratégias para assegurar a confidencialidade dos dados [Brito et al. 2024], essa propriedade não é intrínseca ao modelo.

Já o segundo ponto está vinculado aos desafios na implementação de mecanismos de autenticação mútua que exigem uma terceira parte (e.g., PKI). Este problema é consequência da integração entre os *bootstrappings* de rede e segurança na NDN, uma vez que, nesse estágio, o dispositivo ainda não possui acesso à rede e às aplicações, comprometendo a adoção de protocolos de autenticação usuais. Por exemplo, em [Yu et al. 2023b], utiliza-se PKI na etapa de autenticação, mas é assumido que os nós já estão conectados à infraestrutura da Internet antes de ingressarem no domínio da aplicação NDN. Como alternativa para esse cenário, os dispositivos podem ser pré-configurados pelos fabricantes com informações sobre CASs confiáveis. No entanto, essa abordagem depende de terceiros e pode não ser viável para a maioria dos dispositivos, especialmente em ambientes com restrição de recursos e conectividade limitada. Outra distinção vinculada a ambas as arquiteturas é a maneira como a segurança é implementada em nível de rede. No TCP/IP, a segurança na comunicação é alcançada a partir da proteção dos canais de comunicação fim-a-fim. Já na NDN as propriedades de segurança (i.e., autenticidade,

não-repúdio, integridade) são alcançadas através da criptografia de cada pacote, individualmente. Assim, a NDN não requer o estabelecimento de canais seguros de comunicação, mas necessita de autenticação e distribuição de credenciais, de forma similar ao TCP/IP.

Tabela 4. Comparativo entre o *bootstrapping* em redes TCP/IP e NDN.

Características do *Bootstrapping*	Subcritério	TCP/IP	NDN
Propriedades	Autenticidade	✓	✓
	Não-repúdio	✓	✓
	Integridade	✓	✓
	Confidencialidade	✓	✗
Etapas	Autenticar nós	✓	✓
	Distribuir credenciais	✓	✓
	Definir canal seguro	✓	✗
	Criar nome e certificado	✗	✓
Automação	de rede	✓	✓
	de segurança	✗	✓
Integração	de rede e segurança	✗	✓

Por fim, na NDN a semântica de nomeação é usada para automatizar o *bootstrapping*, enquanto nas redes TCP/IP, a automação ocorre por meio de ferramentas na camada de aplicação. Apesar das diferenças, as arquiteturas compartilham etapas similares, como a distribuição de credenciais, âncoras de confiança e políticas de segurança. Assim, o processo de *bootstrapping* pode ser generalizado em três etapas: i) autenticação mútua, ii) distribuição de credenciais e iii) configuração de canais de comunicação ou pacotes seguros. A Tabela 5 sumariza alguns trabalhos relacionados que propõem soluções de *bootstrapping* de segurança em arquiteturas TCP/IP e NDN. Cada uma dessas alternativas se baseia em um conjunto de premissas, mecanismos de autenticação e distribuição de credenciais distintas. No entanto, algumas propostas de *bootstrapping* de segurança em NDN não apresentam avaliação de desempenho para verificação da viabilidade de sua aplicação em cenários reais. Neste caso, como o *bootstrapping* de segurança é o próprio *bootstrapping* da rede, o impacto do seu desempenho não está restrito às entidades e aplicações que têm requisitos de segurança bem estabelecidos. Portanto, torna-se essencial verificar a viabilidade desse tipo de alternativa através da execução de análises de desempenho dessas propostas. Assim, a próxima seção apresenta uma prova de conceito do *bootstrapping* em NDN avaliando o impacto da quantidade de produtores nos tempos de execução das etapas do *bootstrapping* e no *overhead* de mensagens adicionados à rede.

4. Prova de Conceito – Bootstrapping NDN

Na arquitetura TCP/IP, a camada de rede é responsável apenas pela comutação de pacotes, delegando a segurança às camadas superiores. Já na NDN, os requisitos de segurança são integrados diretamente à camada de rede, tornando o *bootstrapping* de segurança parte nativa do processo. No entanto, essa abordagem pode impactar o tempo de entrada de novos nós e o desempenho da rede. Para avaliar esses efeitos, foi desenvolvida uma prova de conceito que implementa o *bootstrapping* em NDN, analisando seu impacto em função da quantidade de nós. Foram simulados cenários com 3 a 101 nós distribuídos aleatoriamente em uma área de $100 \times 100\ m^2$, considerando uma rede *wireless adhoc* estática,

Tabela 5. Soluções de bootstrapping de segurança em redes TCP/IP e NDN.

Arquitetura	Bootstrapping	Características
TCP/IP	LwM2M [Sethi et al. 2025]	- Confiança inicial baseada em credenciais pré-instaladas - Assume que o cliente possui as informações necessárias para confiar no servidor de bootstrapping - Baseia-se em uma arquitetura RESTful e pressupõe que a conectividade de rede foi estabelecida - Modos de bootstrapping: de fábrica, a partir de smartcard, iniciado pelo cliente ou pelo servidor
	EAP-NOOB [Sethi et al. 2025]	- Os dispositivos precisam ser pré-configurados pelos fabricantes - Lista pré-instalada de servidores de bootstrapping e certificados das âncoras de confiança
	BRSKI [Sethi et al. 2025]	- Depende de certificados pré-instalados pelo fabricante, serviço de Internet e conectividade local - Foca na autenticação remota de dispositivos que dependem do IEEE 802.1AR - Destina-se a dispositivos sem restrições, mas os autores afirmam que ela é escalável
	LPWAN [Sethi et al. 2025]	- O bootstrapping depende de soluções proprietárias e pressupõe a pré-instalação de credenciais - Refere-se a tecnologias de camada de enlace que são severamente limitadas - Diversos protocolos são utilizados no processo de bootstrapping (Ex: LoRaWAN, Sigfox e NB-IoT)
NDN	Cornerstone [Yu et al. 2023b]	- Depende de informações previamente cadastradas em uma Web PKI e de listas de acesso - Foca na autenticação remota e automatizada de dispositivos - Os usuários conseguem validar e produzir conteúdos
	NDN Sign on [Li et al. 2019]	- Solução de boostrapping em NDN para cenários de smart homes - Foca em dispositivos com restrição de recursos - Solução adaptativa que escolhe mecanismos de segurança com base na capacidade dos dispositivos
	Intertrust [Yu et al. 2022a]	- Solução de bootstrapping entre diferentes zonas, organizada em duas etapas: autenticação e autorização - Possibilita consumo de dados entre dispositivos e usuários de domínios distintos

gerada pelo BonnMotion 3.0.1 [Bothe and Aschenbruck 2020]. O primeiro nó, designado como controlador da zona, atuou como *bootstrapper*, armazenando o certificado raiz do domínio (âncora de confiança) e coordenando as etapas de *bootstrapping*. Os demais nós foram organizados em pares produtor-consumidor, onde produtores hospedavam aplicações NDN e geravam conteúdo, enquanto consumidores os requisitavam.

A implementação seguiu um modelo genérico de *bootstrapping* NDN, composto por um conjunto mínimo de etapas essenciais (Figura 2. Em geral, a autenticação do *bootstrapper* se baseia em mecanismos como Web PKI ou em informações previamente instaladas pelo fabricante. Na Figura 2, o processo é iniciado com a autenticação do *boostrapee* enviando uma requisição de autenticação para o *bootstrapper* (I1) que, ao receber essa requisição, solicita os parâmetros de autenticação (I2) do *bootstrapee*. Os parâmetros são enviados (D2) e é encaminhada uma resposta a respeito da autenticação (D1), que pode ser aprovada ou reprovada. Em seguida, é realizada a instalação da âncora de confiança a partir da solicitação do certificado (I3) e do seu recebimento (D3). Na penúltima etapa ($SCHEMA$), o *bootstrapper* obtém a versão mais recente do esquema de confiança da rede. As entidades podem enviar pacotes de interesse $SCHEMA/SUBSCRIBE$ (I4) para monitorar atualizações e, quando detectadas, o controlador responde com o novo esquema de confiança (D4). Para obter a versão mais recente, a entidade envia um pacote de interesse $SCHEMA/CONTENT$ (I5) e recebe o conteúdo do esquema de confiança propriamente dito (D5). Já na última etapa ($SIGN$), ocorre a assinatura do certificado do *bootstrapee*. Primeiro, o *bootstrapee* solicita a assinatura ao *bootstrapper* (I6), que responde pedindo o certificado autoassinado por meio de um pacote KEY (I7). Após recebê-lo (D7), o *bootstrapper* assina o certificado e o devolve (D6), permitindo que a entidade o armazene e forneça este certificado a outros nós NDN.

Na avaliação de desempenho, a etapa de autenticação mútua ($AUTH$) foi desconsiderada por envolver diferentes estratégias com demandas computacionais variáveis, o que iria comprometer a análise estatística dos resultados. Além disso, as estratégias de autenticação utilizadas em NDN são as mesmas utilizadas na arquitetura atual, portanto, o impacto seria semelhante em ambas. Adicionalmente, a instalação da âncora de confiança

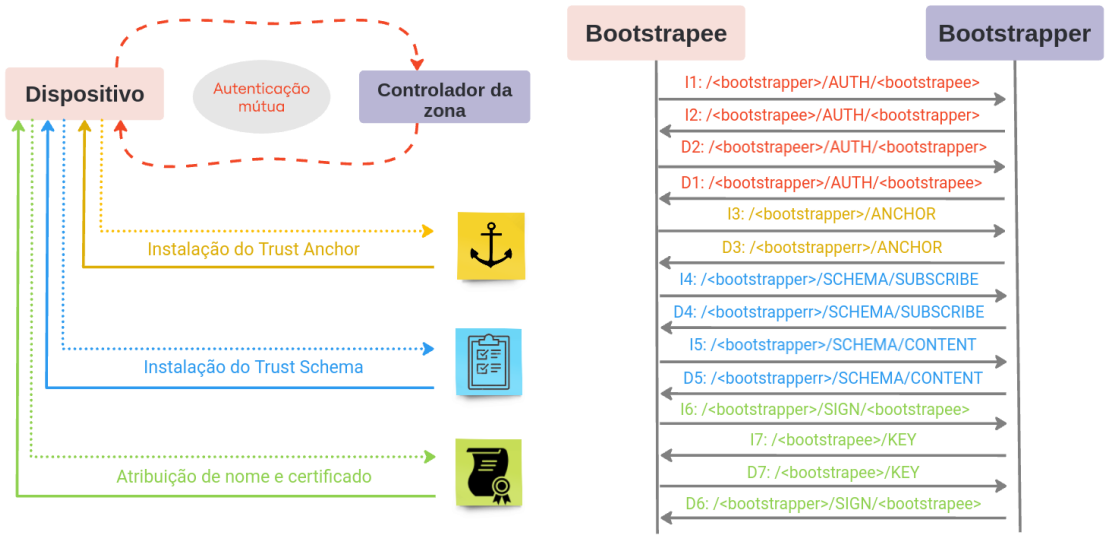


Figura 2. Sequência de etapas do *bootstrapping* NDN.

($ANCHOR$) também foi desconsiderada pois, na maioria dos casos, ocorre *out-of-band* ou envolve apenas a solicitação da âncora de confiança e o recebimento do certificado.

4.1. Parâmetros e Métricas de Avaliação

Os experimentos foram realizados no ndnSIM 3.30.1, simulador oficial do projeto NDN. A simulação envolveu dispositivos sem fio em uma rede *ad hoc* baseada em IEEE 802.11g. Cada cenário foi replicado 10 vezes para análise estatística, com cálculo da média, desvio padrão e intervalo de confiança de 95%. A Tabela 6 apresenta os parâmetros utilizados, enquanto os demais foram mantidos conforme os padrões do ndnSIM.

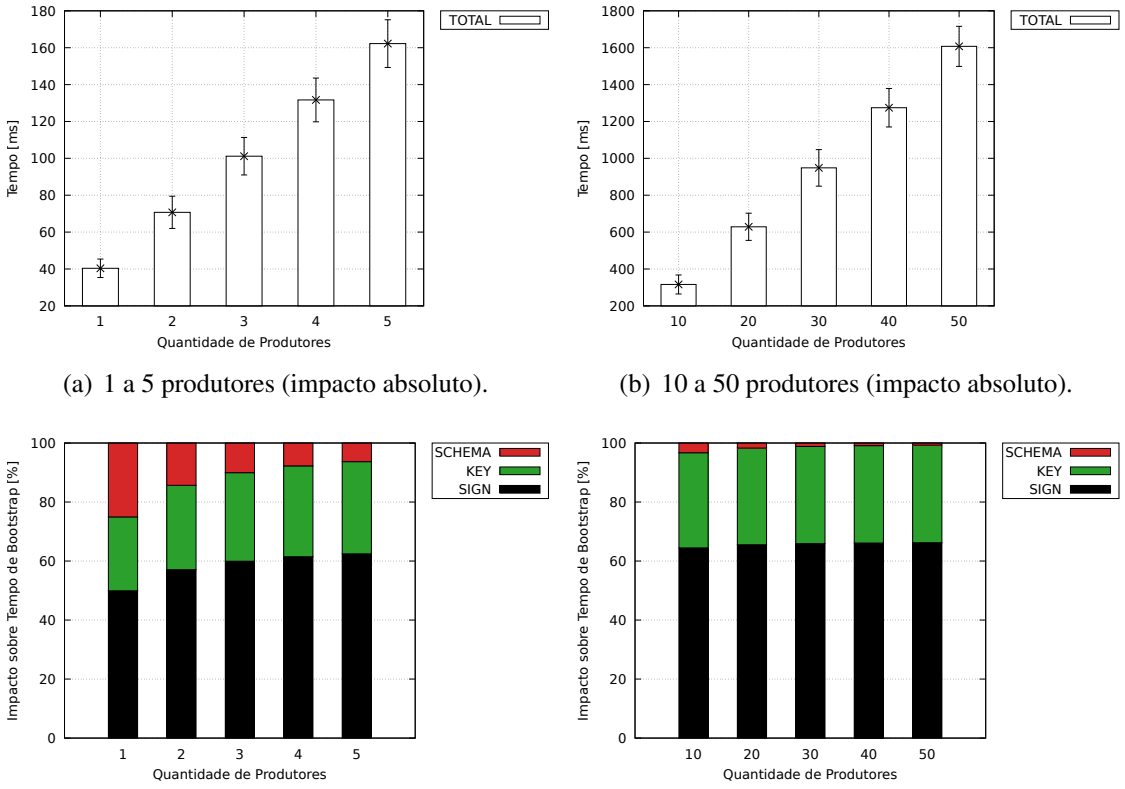
Tabela 6. Parâmetros de Simulação.

Geral	Duração de cada Simulação	10 s
	Número de Simulações	10
Link	Tecnologia	IEEE 802.11g
	Capacidade	24 Mbps
Rede	Modelo de perda de pacote	Nakagami LogDistance
	Atraso de propagação	ConstantPropagationSpeed
Carga da Rede	Modelo de carga	Constant bitrate (CBR)
	Frequência de pacotes	200 interesses/seg
	Tempo de vida do interesse	1 s

Para avaliar o desempenho, as seguintes métricas foram definidas: i) impacto de cada etapa no tempo total de *bootstrapping* (%); ii) quantidade de pacotes de dados transmitidos (%) em cada etapa; iii) impacto das etapas no consumo de largura de banda (%); e iv) impacto do tamanho médio dos pacotes de dados (%). Para cada métrica, também foi avaliado o valor absoluto: tempo de cada etapa (ms), quantidade de dados transmitidos ($pkts$), largura de banda consumida (KB) e tamanho médio dos pacotes (B).

4.2. Resultados e Discussões

Os resultados apresentados na Figura 3 indicam que todas as etapas do *bootstrapping* de segurança tem impacto significativo sobre o tempo total do processo[1]. No entanto, observa-se que, à medida que o número de produtores na rede aumenta – conforme ilustrado nas Figuras 3(c) e 3(d) – as etapas de assinatura ($SIGN$) e de obtenção dos certificados autoassinados de cada produtor (KEY) exercem maior influência sobre o tempo total do que a etapa de recepção do esquema de confiança ($SCHEMA$), possuindo os seguintes valores de desvio padrão $\pm 2{,}895\%$, $\pm 2{,}312\%$, e $\pm 1{,}626\%$, respectivamente. Esse comportamento pode ser explicado pelo fato de que a recepção do esquema de confiança envolve apenas a troca de duas mensagens: um pacote de interesse e um pacote de dados. Em contraste, no protocolo de *bootstrapping* de segurança (Figura 2), cada mensagem $SIGN$ exige duas mensagens KEY. Como consequência, esses pacotes tornam-se mais suscetíveis a interferências no meio compartilhado da rede e a atrasos decorrentes do enfileiramento na saída de cada nó, impactando o tempo necessário para a conclusão do processo em cada produtor. Apesar disso, em uma rede sem congestionamento, estima-se que o *bootstrapping* de um novo nó possa ser concluído em aproximadamente $40ms$.



(a) 1 a 5 produtores (impacto absoluto).

(b) 10 a 50 produtores (impacto absoluto).

(c) 1 a 5 produtores (impacto normalizado).

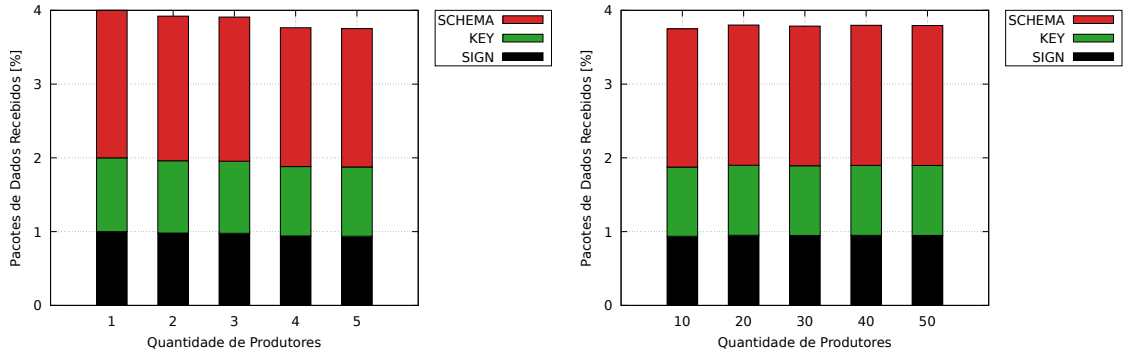
(d) 10 a 50 produtores (impacto normalizado).

Figura 3. Tempo de cada etapa do *bootstrapping* de segurança NDN.

A Figura 4 apresenta a quantidade de pacotes de dados transmitidos entre os produtores, o controlador de zona e os consumidores. Observa-se o aumento no tráfego à

[1]Os tempos de processamento das etapas foram desconsiderados devido à limitação do ndnSIM, que, como todo simulador de eventos discretos, é incapaz de mensurar o tempo de execução dos algoritmos.

medida que o número de produtores cresce, sendo este majoritariamente composto por solicitações de dados de aplicação ($PAYLOAD$). Esta requisições de dados representam cerca de 96% do trafégo total, enquanto que o *bootstrapping* NDN é responsável apenas por 4% do trafego restante (Figuras 4(a) e 4(b)), com desvio padrão de $\pm 3{,}638\%$. No contexto do *bootstrapping*, a maior parte dos pacotes transmitidos está relacionada à recepção do esquema de confiança. Isto ocorre pois, embora a etapa $SCHEMA$ tenha menor impacto temporal que $SIGN$ e KEY, ela exige um volume maior de pacotes. No entanto, de forma geral, o impacto do *bootstrapping* na quantidade total de pacotes transmitidos é pouco significativo (4% do tráfego total, com valores de desvio padrão de $\pm 1{,}605\%$, $\pm 1{,}062\%$, e $\pm 0{,}597\%$ para as etapas $SCHEMA$, KEY e $SIGN$, respectivamente).

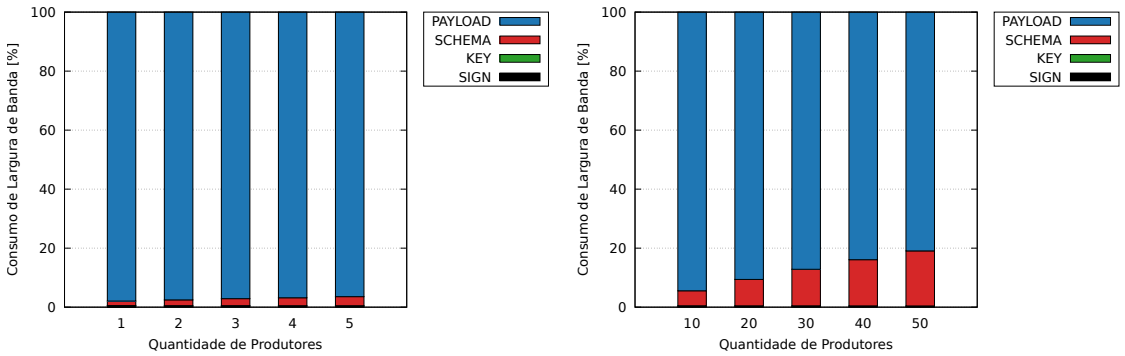


(a) 1 a 5 produtores (impacto normalizado). (b) 10 a 50 produtores (impacto normalizado).

Figura 4. Quantidade de pacotes transmitidos na rede.

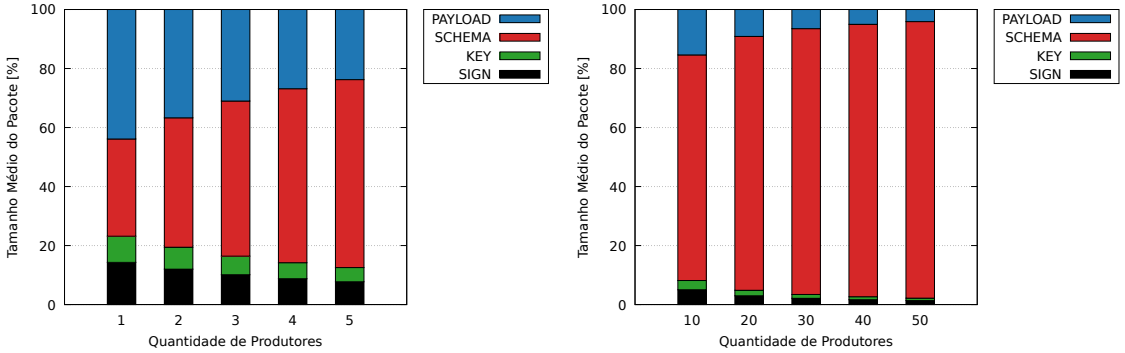
A Figura 5 apresenta o consumo de largura de banda associado à transmissão de conteúdos ($PAYLOAD$) e ao *bootstrapping*, considerando os valores de desvio padrão de $\pm 2{,}914\%$, $\pm 1{,}824\%$, $\pm 1{,}245\%$, e $\pm 0{,}629\%$ para o $PAYLOAD$, $SCHEMA$, KEY, e $SIGN$, respectivamente. Como o número de pacotes enviados cresce proporcionalmente à quantidade de produtores (Figura 4), observa-se um aumento correspondente no consumo de banda, predominantemente influenciado pela demanda das aplicações dos consumidores. Além disso, à medida que mais produtores são adicionados, o esquema de confiança aumenta de tamanho, pois inclui regras de validação para cada novo conteúdo produzido. Embora existam abordagens capazes de mitigar o crescimento do esquema de confiança através da generalização de regras, como as baseadas em nomeação estritamente hierárquica, tais abordagens dependem de premissas específicas, podendo não ser compatíveis com aplicações não projetadas para esses modelos. O crescimento no número de requisições do esquema de confiança por cada produtor também contribui para o aumento do consumo de banda. Esse comportamento é evidenciado nas Figuras 5(a) e 5(b) demonstra a relação entre o número de produtores, o tamanho do esquema de confiança, quantidade de requisições do esquema de confiança e o impacto na largura de banda.

Para identificar qual componente exerce maior influência no consumo de largura de banda, foi calculado o tamanho de cada pacote do *bootstrapping*, apresentado na Figura 6. Os resultados indicam que a maioria dos pacotes mantém um tamanho constante à medida que o número de produtores aumenta na rede, exceto o esquema de confiança ($SCHEMA$), considerando os valores de desvio padrão de $\pm 2{,}180\%$, $\pm 1{,}875\%$, e $\pm 1{,}184\%$ para as etapas $SCHEMA$, KEY e $SIGN$, respectivamente. Esse comporta-



(a) 1 a 5 produtores (impacto normalizado). (b) 10 a 50 produtores (impacto normalizado).

Figura 5. Consumo de largura de banda dos pacotes transmitidos na rede.



(a) 1 a 5 produtores (impacto normalizado). (b) 10 a 50 produtores (impacto normalizado).

Figura 6. Tamanho médio dos pacotes de dados.

mento sugere que o crescimento do esquema de confiança e sua atualização frequente nos produtores são os principais responsáveis pelo aumento do consumo de banda.

Diante desse cenário, duas abordagens podem ser adotadas para otimizar o processo de *bootstrapping*: (i) reduzir o número de solicitações do esquema de confiança pelos produtores e (ii) controlar o crescimento do esquema. A primeira pode ser parcialmente implementada por meio do mecanismo de cache da NDN, o que diminui o impacto das solicitações para o mesmo conteúdo. No entanto, a eficácia dessa solução depende de diversos fatores, como o tamanho e o posicionamento dos caches na rede, o algoritmo de substituição de cache, a frequência, o comportamento temporal e o tamanho das solicitações do esquema de confiança, e a velocidade de acesso à memória do cache. A segunda abordagem pode ser realizada a partir de algoritmos que generalizem as regras de validação para múltiplos produtores, eliminando a necessidade de definir regras individuais para cada conteúdo. Para isso, é crucial definir antecipadamente o esquema de confiança dos conteúdos, e as expressões regulares (ER) das regras genéricas. Contudo, o uso de ER e a generalização de regras reduzem a expressividade dos nomes NDN, comprometendo não apenas a semântica, mas também outras propriedades da rede, que são diretamente dependentes da nomeação (e.g., roteamento, agregação de pacotes, cache).

5. Conclusão

Este artigo comparou o *bootstrapping* de segurança em TCP/IP e NDN, adotando uma abordagem teórica, com análise das funcionalidades e estrutura arquitetural, e prática, com a implementação de uma PoC (Q1). Observou-se que as arquiteturas apresentam etapas em comum: autenticação mútua e distribuição de credenciais (Q2). Na NDN, a autenticação mútua baseia-se na validação de uma âncora de confiança, semelhante às CAs no TCP/IP, enquanto que a distribuição de credenciais equivale ao compartilhamento de chaves assimétricas em redes TCP/IP. No entanto, as arquiteturas diferem em requisitos de segurança e suporte à automação: A NDN oferece automação nativa no *bootstrapping* de segurança, mas não provê confidencialidade com o *bootstrapping* de segurança nativo.

Os resultados experimentais indicaram que a inclusão de mecanismos de segurança no *bootstrapping* aumenta o tempo de integração de novos nós, sem prejudicar significativamente a operação da rede (Q3 e Q4). O *bootstrapping* na NDN gera mais mensagens do que protocolos legados, como o DHCP, mas o impacto é pequeno em relação ao tráfego da rede. Além disso, o impacto das etapas de segurança é limitado em redes estáveis e é possível otimizar o processo reutilizando credenciais previamente estabelecidas nos casos de acesso por dispositivos previamente inicializados. Adicionalmente, dispositivos IoT estão sujeitos à limitações que podem afetar o *bootstrapping*: (i) a geração de chaves criptográficas é frequentemente prejudicada pela baixa entropia dos sistemas, e (ii) há um *trade-off* entre o nível de segurança desejado, a complexidade algorítmica das soluções e o consumo de recursos computacionais em ambientes IoT.

6. Agradecimentos

Os autores agradecem o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq - 200404/2022-9 e 306571/2025-0), da Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB - TIC0004/2015) e do Air Force Office of Scientific Research (AFOSR - FA9550-23-1-0631).

Referências

Alpár, G., van den Broek, F., Hampiholi, B., Jacobs, B., Lueks, W., and Ringers, S. (2017). Irma: practical, decentralized and privacy-friendly identity management using smartphones. In *10th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2017)*. Accessed: 2023-11-15.

Boeyen, S., Santesson, S., Polk, T., Housley, R., Farrell, S., and Cooper, D. (2008). Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280.

Bothe, A. and Aschenbruck, N. (2020). BonnMotion 4 – Taking Mobility Generation to the Next Level. In *2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC)*, pages 1–8.

Bourdrez, D., Krawczyk, H., Lewi, K., and Wood, C. A. (2022). The OPAQUE Asymmetric PAKE Protocol. Internet-Draft draft-irtf-cfrg-opaque-09, Internet Engineering Task Force. Work in Progress.

Brito, I. V. S., Schramm, K., and Sampaio, L. N. (2024). D-NAC: Controle de acesso distribuído para redes de dados nomeados. *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*.

Danilchenko, V., Theobald, M., and Cohen, D. (2019). Bootstrapping security configuration for iot devices on networks with tls inspection. In *2019 IEEE Globecom Workshops (GC Wkshps)*, pages 1–7. IEEE.

Frankel, S. and Krishnan, S. (2011). IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap. RFC 6071.

Hail, M. A. (2019). Iot-ndn: An iot architecture via named data netwoking (ndn). In *2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pages 74–80.

Hoffman, P. E. and McManus, P. (2018). DNS Queries over HTTPS (DoH). RFC 8484.

Jones, M. B., Bradley, J., and Sakimura, N. (2015). JSON Web Token (JWT). RFC 7519.

Kaufman, C. (2005). Internet Key Exchange (IKEv2) Protocol. RFC 4306.

Klöti, R., Kotronis, V., and Smith, P. (2013). Openflow: A security analysis. In *2013 21st IEEE International Conference on Network Protocols (ICNP)*, pages 1–6.

Kumar, T., Ylianttia, M., and Harjula, E. (2022). Securing edge services for future smart healthcare and industrial iot applications. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6. IEEE.

Li, Y., Zhang, Z., Wang, X., Lu, E., Zhang, D., and Zhang, L. (2019). A secure sign-on protocol for smart homes over named data networking. *IEEE Communications Magazine*, 57(7):62–68.

Malamas, V., Kotzanikolaou, P., Nomikos, K., Zonios, C., Tenentes, V., and Psarakis, M. (2025). Ha-caap: Hardware-assisted continuous authentication and attestation protocol for iot based on blockchain. *IEEE Internet of Things Journal*.

Malik, M., Dutta, M., and Granjal, J. (2019). A survey of key bootstrapping protocols based on public key cryptography in the internet of things. *IEEE Access*, 7:27443–27464.

Mirajkar, R. R., Shinde, G. R., Mahalle, P. N., and Sable, N. P. (2024). NDN Security: Cryptographic Approaches for Safeguarding Content-Centric Networking against Threats. *Journal of Electrical Systems*, 20(3s):1516–1541.

Nath, K. and Sarkar, P. (2020). Efficient elliptic curve diffie-hellman computation at the 256-bit security level. *IET Information Security*, 14(6):633–640.

Neuman, B. and Ts'o, T. (1994). Kerberos: an authentication service for computer networks. *IEEE Communications Magazine*, 32(9):33–38.

Rai, V. K., Tripathy, S., and Mathew, J. (2023). Lpa: A lightweight puf-based authentication protocol for iot system. In *2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1712–1717. IEEE.

Ramani, S. K., Podder, P., and Afanasyev, A. (2020). Ndnviber: Vibration-assisted automated bootstrapping of iot devices. In *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6. IEEE.

Rescorla, E. (2018). The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446.

Rescorla, E. and Dierks, T. (2008). The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246.

Sampaio, L. N., Freitas, A. E. S., Araújo, F. R., Brito, I. V. S., and Ribeiro, A. V. (2021). Revisitando as ICNs: Mobilidade, Segurança e Aplicações Distribuídas através das Redes de Dados Nomeados. In *XXXXXX*, XXXXX.

Selander, G., Mattsson, J. P., and Palombini, F. (2024). Ephemeral Diffie-Hellman Over COSE (EDHOC). RFC 9528.

Sethi, M., Sarikaya, B., and Garcia-Carrillo, D. (2025). Terminology and processes for initial security setup of IoT devices. Internet-Draft draft-irtf-t2trg-security-setup-iot-devices-04, Internet Engineering Task Force. Work in Progress.

Siriwardena, P. (2020). *OpenID Connect (OIDC)*, pages 129–155. Apress, Berkeley, CA.

Tao, Y., Xiao, S., Hao, B., Zhang, Q., Zhu, T., and Chen, Z. (2020). Wire: Security bootstrapping for wireless device-to-device communication. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–7. IEEE.

Tschofenig, H. and Eronen, P. (2005). Pre-Shared Key Ciphersuites for Transport Layer Security (TLS). RFC 4279.

Ullah, S. S., Hussain, S., Gumaei, A., and AlSalman, H. (2021). A secure NDN framework for Internet of Things enabled healthcare. *Computers, Materials & Continua*, 67(1):223–240.

Wang, X., Wang, X., and Li, Y. (2021). Ndn-based iot with edge computing. *Future Generation Computer Systems*, 115:397–405.

Yu, T., Ma, X., Xie, H., Jia, X., and Zhang, L. (2023a). On the security bootstrapping in named data networking. *arXiv preprint arXiv:2308.06490*.

Yu, T., Ma, X., Xie, H., Kocaoğullar, Y., and Zhang, L. (2022a). Intertrust: establishing inter-zone trust relationships. In *Proceedings of the 9th ACM Conference on Information-Centric Networking*, pages 180–182.

Yu, T., Ma, X., Xie, H., Kutscher, D., and Zhang, L. (2023b). Cornerstone: Automating remote ndn entity bootstrapping. In *Proceedings of the 18th Asian Internet Engineering Conference*, pages 62–68.

Yu, T., Zhiyi, Z., Newberry, E., Afanasyev, A., Pau, G., Wang, L., and Zhang, L. (2022b). Names to rule them all: Unifying mobile networking via named secured data. Technical report, Technical Report NDN-0072.

Zhang, Z., Afanasyev, A., and Zhang, L. (2017). Ndncert: universal usable trust management for ndn. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*, pages 178–179.

Zhang, Z., Lu, E., Li, Y., Zhang, L., Yu, T., Pesavento, D., Shi, J., and Benmohamed, L. (2018). Ndnot: a framework for named data network of things. In *Proceedings of the 5th ACM Conference on Information-Centric Networking*, ICN '18, page 200–201, New York, NY, USA. Association for Computing Machinery.