UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS –
FACULDADE DE CIÊNCIA DA COMPUTAÇÃO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

RAMON DA GAMA CORDEIRO

MECANISMO DE AUTENTICAÇÃO DESCENTRALIZADA DE USUÁRIOS EM REDES DEFINIDAS POR SOFTWARE

BELÉM-PA

Fevereiro / 2018

RAMON DA GAMA CORDEIRO

MECANISMO DE AUTENTICAÇÃO DESCENTRALIZADA DE USUÁRIOS EM REDES DEFINIDAS POR SOFTWARE

Trabalho de Conclusão de Curso apresentado no curso de Ciência da Computação da Universidade Federal do Pará, como requisito parcial para obtenção do grau de Bacharelado em Ciência da Computação.

Orientador: Dr. Antônio Jorge Gomes Abelém

Co-Orientadores: Leonardo Barbosa da Costa

BELÉM-PA

Fevereiro / 2018

RAMON DA GAMA CORDEIRO

MECANISMO DE AUTENTICAÇÃO DESCENTRALIZADA DE USUÁRIOS EM REDES DEFINIDAS POR SOFTWARE

Trabalho de Conclusão de Curso apresentado no curso de Ciência da Computação da Universidade Federal do Pará, como requisito parcial para obtenção do grau de Bacharelado em Ciência da Computação.

Aprovada em: __/__/____

BANCA EXAMINADORA

Prof. Dr. Antônio Jorge Gomes Abelém
Universidade Federal do Pará
Orientador

Prof. Dr. Denis Lima do Rosário
Universidade Federal do Pará

Dr. Billy Anderson Pinheiro

Agradecimentos

Agradeço a Deus que me deu forças para realizar este trabalho, especialmente nos momentos de luto, incerteza, tristeza e problemas pessoais.

Agradeço aos meus pais pela paciência e por acreditarem em mim.

Agradeço à Pamella que me motivou em alguns momentos e por sempre estar ao meu lado.

Agradeço ao Professor Antônio Abelém que acreditou no desenvolvimento deste trabalho.

Agradeço ao Leonardo Costa, Co-orientador e grande parceiro nesta empreitada que sem sua paciência e solicitude, este trabalho não seria possível. Especialmente porque a ideia deste trabalho foi dele. Meus sinceros agradecimentos.

Agradeço à Igreja Batista Reformada de Belém por ter me ajudado a renovar o ânimo para encarar as dificuldades, através das pregações e convívio com pessoas incríveis

Agradecimento sincero a todos que de alguma maneira contribuíram para este trabalho.

1

A realidade é como um mosaico, as partes podem ser feias em sí,
mas o todo é belo.
(Jonathan Edwards)

Resumo

O mecanismo de autenticação descentralizada permite que usuários acessem os recursos de uma rede definida por software (SDN) com segurança, disponibilidade e confiabilidade. O paradigma SDN tem sido desenvolvido ao longo dos últimos anos por pesquisadores, mas ainda existe o desafio de garantir acesso seguro e disponibilidade da rede. Muitos trabalhos propostos utilizam autenticação centralizada e apenas um controlador para gerenciar a rede.O trabalho propõe um mecanismo de acesso seguro em redes de múltiplos controladores, através da descentralização do processo de autenticação do usuário entre os gerentes do plano de controle com o intuito de impedir acesso indevido e garantir que os serviços estejam sempre disponíveis.

Palavras-chaves: autenticação descentralizada; autenticação SDN; segurança SDN; criptografia limiar; segurança da informação; controle de acesso.

Abstract

The decentralized authentication mechanism allows users to acess resources of a Software-Defined Network (SDN) with security, available and reliability. The SDN paradigm has grow in recent years through researchers' work, but still there is challenges of ensure security acess and availability of network. Many papers make use of centralized authentication with only one controller to manager the mesh. This paper proposes a mechanism of security acess in networks of multiples controllers, through of decentralization of user authentication process among managers of control plane in order to hinder improper acess and guarantee that the services are always available.

Keywords: decentralized authentication; SDN authentication; SDN security; threshold criptography; information security; acesso control.

Sumário

Lista de Abreviaturas

API - Application Programming Interface

AR - Acess Request

CS - Content Store

DDoS - Distributed Denial of Service

DNS - Domain Name System

ELASTICON - Elastic Distributed Controller Architecture

IEEE - Institute of Electrical and Electronics Engineers

FIB - Forwarding Information Base

IP - Internet Protocol

LAN - Local Area Network

LMI - Layer Management Interface

MAC - Media Access Control

NAC - Network Acess Control

OrBAC - Organization-Aased Access Control

PAE - Port Acess Entity

PCR - Plataform Configuration Register

PDP - Policy Decision Point

PEP - Policy Enforcement Point

PIT - Pending Interest Table

SDN - Software Defined Networks

S-NAC - Security Network Acess Control

TCP - Transmission Control Protocol

VLAN - Virtual LAN

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

Com o advento da Internet e crescimento de seu uso ao passar dos anos, houve a criação de uma sociedade denominada digital onde muitos dispositivos estão conectados e acessíveis de qualquer lugar. Porém, apesar de sua ampla utilização, as redes tradicionais possuem alguns elementos de alta complexidade em relação a sua administração, especialmente no que tange a flexibilidade e atualização. No geral, há uma dificuldade associada a mudanças na rede de acordo com políticas já definidas e reconfiguração desta para responder a falhas, atualizações e demandas que podem vir a ocorrer. Pois as redes são dinâmicas e gerentes de rede possuem dificuldades para responder aos eventos da rede Kreutz et al. (2013), devido a baixa flexibilidade e pouco ou inexistente programabilidade do modelo tradicional.

Normalmente, em uma rede tradicional existem equipamentos de diferentes fabricantes. Isto gera dificuldade na atualização de políticas da rede, pois ela deverá ser feita em cada um dos equipamentos e geralmente com a utilização de comandos de baixo nível, aumentando a complexidade de gerencia desses dispositivos Kreutz et al. (2013). Além disso, estas redes são verticalmente integradas; o plano de controle e de dados são agrupados Kreutz et al. (2015).

No entanto. A partir do grande avanço de tecnologias de comunicação, modelos distintos do paradigma tradicional surgiram e se tornaram cada vez mais relevantes, pois proporcionam soluções mais adequadas, permitindo alterar a função de alguns elementos da rede e a tarefa por eles empenhada. Além disso, esses novos modelos também possibilitam a redefinição de regras dinamicamente Kreutz et al. (2015).

Dentre estes modelos, o paradigma de Redes Definidas por Software (*Software-Defined Networking* - SDN) apresenta grande destaque na comunidade científica e empresarial atualmente. Nesta abordagem, há a quebra da integração vertical, que separa a lógica de controle de rede (plano de controle) dos roteadores e *switches* subjacentes

que encaminham o tráfego (plano de dados). Isto permite que o gerenciamento da rede se torne programável através de interfaces (APIs) Jain and Paul (2013), como por exemplo o protocolo OpenFlow em McKeown et al. (2008). A partir desta desagregação, os *switches* se tornam dispositivos de encaminhamento simples e a lógica de gestão é executada em um controlador centralizado, facilitando a implementação de novas políticas, assim como a reconfiguração e evolução da rede Kreutz et al. (2015).

Assim, permite que novos equipamentos adicionados a rede não necessitem ser configurados individualmente, uma vez que existe uma heterogeneidade de fabricantes de dispositivos e muitos destes utilizam comandos específicos de fornecedores (em alguns casos de baixo nível), introduzindo uma alta complexidade na programação e tornando-os difíceis de gerenciarRawat and Reddy (2017). Tais dificuldades são reduzidas neste novo paradigma.

Um outro importante benefício é o melhor desempenho da rede, em razão de permitir um controle centralizado a partir de uma visão global, que concede um gerenciamento eficiente do tráfego e aumento do desempenho da redeRawat and Reddy (2017). Além disso, como SDN separa o plano de dados do plano de controle, gera-se flexibilidade, boa abstração de rede, fácil administração e potencial de inovação Kreutz et al. (2015), pois as regras de tráfego não ficam condicionadas às definições de implementação em hardware de um *switch*.

1.1 Motivação e Justificativa

A medida que crescem as taxas de adoção de dispositivos em redes diferentes, como telefones celulares, tablets e sensores, as redes tornam-se cada vez mais vulneráveis à diversos tipos de ameaças Ali et al. (2015b). Dentre elas, pode-se citar: o acesso não autorizado a dispositivos; a modificação de dados; aplicações maliciosas; e ataques de negação de serviço. A fim de que seja possível evitar tais ameaça, é necessário que exista um mecanismo de autenticação seguro e políticas de controle do acesso devem ser aplicadas para que haja um monitoramento mais adequado dos diversos dispositivos que se conectam à rede, mantendo a confiabilidade, integridade e disponibilidade.

A necessidade de autenticação de usuários é um desafio presente nas redes tradicionais, bem como em SDN.Além da necessidade de autenticação de usuários, existe a deficiência de poucos trabalhos com proposta descentralizada de controle de acesso para o contexto SDN. Diante dos benefícios e características propostas pelo modelo em questão (SDN), a capacidade de visualizar o estado da rede em tempo real e controlar o comportamento da malha através da programação introduz possibilidades interessantes em relação a segurança em redes.O controlador pode ser utilizado como elemento fundamental no processo de autenticação em SDN. Como descrito em Ali et al. (2015a), esta utilização não é muito frequente no estado da arte. Mais comumente, utiliza-se um servidor centralizado de autenticação e desatrelado ao controlador.

Na literatura existem algumas propostas para realizar a autenticação em SDN,

porém a maioria não utiliza o controlador como um elemento ativo durante este processo na tentativa de uma entidade acessar os recursos da rede, e a estratégia de descentralização da funcionalidade de autenticação não está presente.

A abordagem de utilizar um servidor centralizado consiste em problema de segurança, visto que caso o servidor seja comprometido, a segurança da rede também será. Para mitigar esse problema, pode ser utilizado abordagem de autenticação descentralizada, onde vários elementos participam do processo, eliminando o ponto único de falha.

Diante do exposto nos parágrafos anteriores, a motivação deste trabalho é propor um mecanismo de autenticação descentralizado de forma a garantir mais segurançã, pois elimina o ponto único de falha, e a utilização de controladores como participantes ativos no processo de autenticação, visto que estes possuem uma visão global da rede.

1.2 Objetivos

Este trabalho tem como objetivo geral propor um mecanismo de autenticação descentralizado em redes definidas por software utilizando o criptossistema de chave pública ElGamal. A utilização do criptossistema Elgamal é justificada porque o mesmo tem sua segurança provada matematicamente. O trabalho tem como objetivo melhorar a segurança na autenticação de usuários em SDN utilizando descentralização do processo. Para alcançar o objetivo principal, os objetivos específicos são definidos:

- Descrever o mecanismo de autenticação descentralizado.
- Implementar mecanismo de autenticação distribuída utilizando criptossistema Elgamal limiar.
- Realizar simulações em topologia SDN.
- Obter os dados das simulações
- coletar, minerar e avaliar os dados das simulações
- Avaliar as amostras para chegar a conclusões acerca da eficiência deste mecanismo.
- Comparar os resultados obtidos com de outras propostas de autenticação para SDN.

1.3 Metodologia

No trabalho foi explorado o problema da autenticação em redes SDN, que não utilizam o controlador como um participante ativo neste processo, bem como a falta de descentralização da tarefa de autenticação. Assim, foi proposto autenticação descentralizado, e analisado o impacto desta proposta perante o cenário de uma rede SDN Elasticamente Distribuída.

Inicialmente, houve a obtenção de conhecimento para uma comparação mais adequada e consistente entre as estratégias através de estudos aprofundados no estado da arte em artigos científicos, os quais foram publicados em conferências e por organizações como é o caso da IEEE.

Houve a implementação do sistema de autenticação utilizando El Gamal limiar para descentralizar a chave privada, primeiramente utilizado um cliente na tentativa acessar recursos da rede, e em seguida vários clientes, ambos em malha Elastica Distribuída-Dixit et al. (2013), o segundo cenário visa verificar a escalabilidade.

Sucessivamente à implementação, diversas simulações foram ser feitas, para o levantamento de dados e validação destes resultados, pois houve a necessidade de testar as estratégias observadas e realizar uma profunda análise destas, levando-se em consideração diversas características como a tempo de autenticação, escalabilidade e segurança obtida através do uso da proposta.E assim comparando estes com os referidos nos trabalhos relacionados para, se possível, comprovar que estas melhorias podem otimizar, o padrão de segurança em SDN existentes sem grandes perdas de desempenho.

Os testes utilizaram o simulador Mininet para criar a topologia da rede e poder coletar os dados destas simulações e permitir posteriormente avalia-los. O controlador utilizado foi o POX que possui interface de programação em python. Este por sua vez realizará o processo de autenticação.

Após estes testes, gráficos foram gerados para uma análise mais precisa do esquema. Além disso, foi feita a detecção das deficiências da proposição em relação a outros existentes,então foi possível considerar os impactos do mecanismo de autenticação descentralizada na rede com a intenção de apresentar melhorias.

1.4 Organização do texto

O restante do documento estão dividido seguindo o ordenamento descrito abaixo:

- Capítulo 2: Referencial Teórico
- Capítulo 3: Trabalhos Relacionados
- Capítulo 4: Proposta
- Capítulo 5: Resultados e Avaliação
- Capítulo 6: Conclusão

1.5 Conclusão do capítulo

Neste capítulo foi explanado a introdução do trabalho, as motivações para realizalo bem como os objetivos e a metodologia. As seções apresentam a o trabalho e mostram

a direção que o trabalho seguiu para que os objetivos fossem alcançados, bem como as metologias utilizadas para alcançar estes objetivos.

CAPÍTULO 2

Referencial Teórico

2.1 Redes Definidas por Software

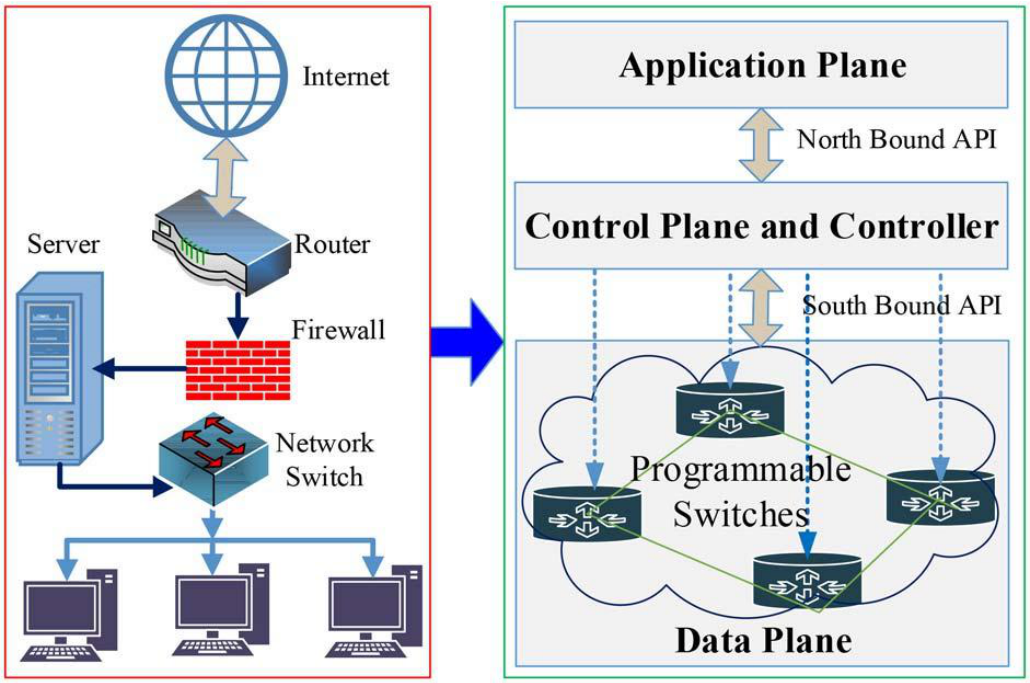
Redes definidas por software é um paradigma de redes de computadores criado com o intuito de permitir flexibilização e programabilidade se comparada ao modelo tradicional Rawat and Reddy (2017).

A Figura 1 apresenta uma comparação da arquitetura SDN com a tradicional. A estrutura SDN possui dois componentes principais que são: o controlador, que é responsável por realizar a gestão da rede; e os *Switches*, que são responsáveis pelo encaminhamento de pacotes, baseado nas instruções implementadas através do controlador Rawat and Reddy (2017). Este paradigma ganhou uma importância significativa para pesquisadores, administradores e empresas envolvidas com o setor devido a flexibilidade e programabilidade introduzida utilizando uma gama de linguagem de programação de alto nível Rawat and Reddy (2017).

A arquitetura consiste em três planos: dados, controle e aplicação.

Plano de Controle

O plano de controle é responsável pelo gerenciamento da rede, este controle é realizado pelo controlador SDN que é um software com capacidade de conhecimento global da rede, este controle é realizado de forma centralizada e dá a este elemento a possibilidade de escolher a melhor solução para a rede, visto que possui o conhecimento total da mesma Zhang et al. (2017). Além da administração da rede e do encaminhamento de pacotes, o



(a) Traditional Network Architecture (b) SDN Architecture

Figura 1: Comparação entre arquitetura tradicional e Arquitetura SDN. Fonte: Rawat and Reddy (2017)

controlador possui três interfaces: northbound, southbound e eastbound/westbound. O elemento de controle atua como intermediário entre o plano de aplicação e o plano de dados Wibowo et al. (2017).

Plano de Dados

Este plano consiste no encaminhamento dos pacotes que trafegam pela rede. No novo paradigma, o switch atua como elemento simples de encaminhamento de dados, onde as decisões são tomadas pelo controlador. O pacote normalmente é enviado ao controlador para que seja adicionado na tabela do switch o caminho, e então é seguido o caminho. O switch, em alguns casos, poderá realizar encaminhamento de pacotes sem passar pelo controlador desde que esta regra já tenha sido definida.

Plano de Aplicação

Este plano consiste em serviços e aplicações da rede que podem ser abstraídos Wibowo et al. (2017). Estes serviços são acessados através da API que provê consistência e padronização de acesso da rede.

Interface Sul

A interface sul é responsável pela comunicação entre controlador e ativos de encaminhamentoWibowo et al. (2017). Através desta, são enviados comandos aos switchs.

Interface Norte

A interface norte provê a comunicação entre as aplicações e o controlador Wi-

bowo et al. (2017). Basicamente, a fronteira é uma API que fornece a capacidade de programabilidade e e gerenciamento dinâmico da rede.

Interface Leste/Oeste

No presente, esta interface ainda não está padronizada, mas é responsável pela comunicação entre controladores ou domínios, nos quais cada domínio possui um controlador gerenciando a rede. A interface foi criada com o intuito de permitir escalabilidade e confiabilidade na rede Wibowo et al. (2017). O paradigma SDN possui um desafio ainda a ser superado que é o uso em escala de um único controlador. A interface tem o intuito de garantir estas duas propriedades importantes em redes. A figura 2 mostra a visão global da arquitetura SDN, bem como a conexão leste e oeste.

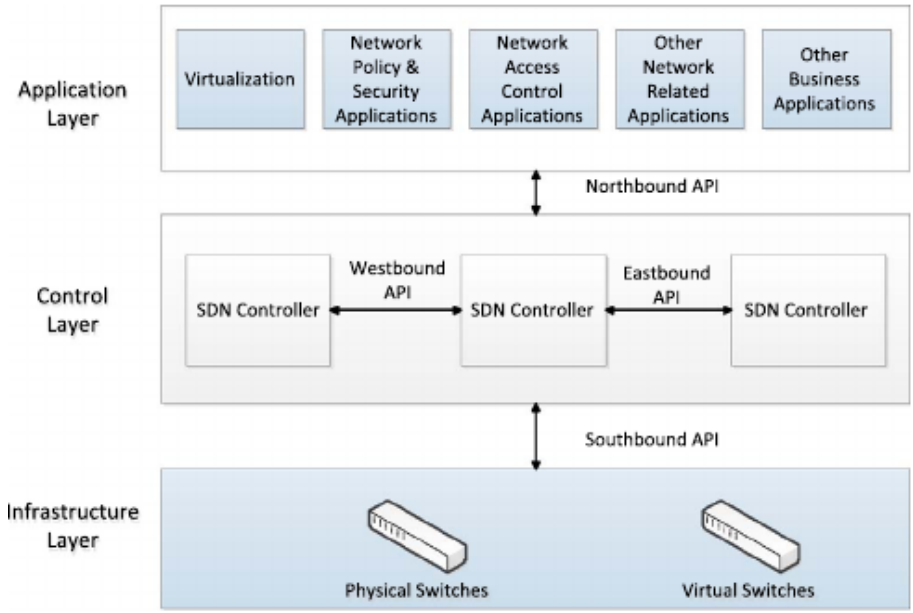


Figura 2: Visão global da Arquitetura SDN. Fonte: Wibowo(2017)

Existem na literatura alguns controladores, responsáveis por reproduzir o comportamento dos componentes na rede (terminais, tomadas de decisão, roteamento dos pacotes, FIB, PIT, CS etc.), assim como as estratégias utilizadas. Dentre estes controladores, há três estudados, são eles o POX, ONOS, RYU.

Ryu é uma framework SDN baseada em componentes, suporta vários protocolos de gerenciamento de redes, tais como OpenFlow, NetConfig e OF-Config.Atualmente Ryu suporta as versões do Openflow: de 1.0 à 1.5 Ryu (2018). O Onos é um dos controlados mais utilizados, possui API para linguagem Java, boa documentação e bom suporte por parte da comunidade. A missão do ONOS é prover um sistema operacional aberto de redes que utiliza os conceitos de SDN Ono (2018).

Pox atualmente se encontra na versão 0.2.0 (carp) e será utilizado, pois dentre as alternativas, possui API em Python (como o Ryu) e apresenta-se como um dos mais didáticos e prático para desenvolvimento de aplicações Sukhveer et al..

2.2 Redes SDN Distribuídas

O presente trabalho tem como foco apresentar um mecanismo de autenticação descentralizado para redes SDN e para tal fará proveito de redes com vários controladores, pois além de garantir escalabilidade e confiabilidade, como descrito, elimina o ponto único de falha, visto que se um controlador for comprometido, os outros podem garantir o funcionamento da autenticação normalmente.

A separação do plano de dados e de controle é uma vantagem na arquitetura SDN. Entretanto a utilização de um único controlador para a tarefa de controle da rede pode comprometer a disponibilidade num contexto de alta demanda, caso o controlador sofra problema de alta carga de trabalho. Este problema de escalabilidade pode ser superado utilizando topologias de múltiplos controladores, na qual a carga de trabalho é distribuída entre os diversos controladores, e caso ocorra indisponibilidade de algum deles, os serviços da rede continuarão a ser disponibilizados.

Diversos modelos tem sido propostos de arquitetura distribuída e como não há uma padronização, cada pesquisador ou equipe de laboratório tem sugerido seus modelos Wibowo et al. (2017). No trabalho Karakus and Durresi (2017) foi realizado a classificação dos tipos de topologia SDN desenvolvidas.

2.2.1 Padrão de Controladores Distribuídos Planos

Nesta topologia, os controladores estão dispostos igualmente, não há mestre nem escravo. Cada controlador possui um ou mais *switches* conectados a sí, essa disposição é intitulada de domínio. No que tange a visualização global da rede, existem duas abordagens. Na primeira, cada controlador possui gerência e percepção apenas dos dispositivos conectados nele, e vê o outro controlador como um nó Karakus and Durresi (2017).Na outra maneira, cada controlador tem a visão global de toda rede Karakus and Durresi (2017). Um exemplo deste padrão é a arquitetura Elasticon que será detalhada nas próximas seções. A figura 3 mostra o padrão de distribuído plano.

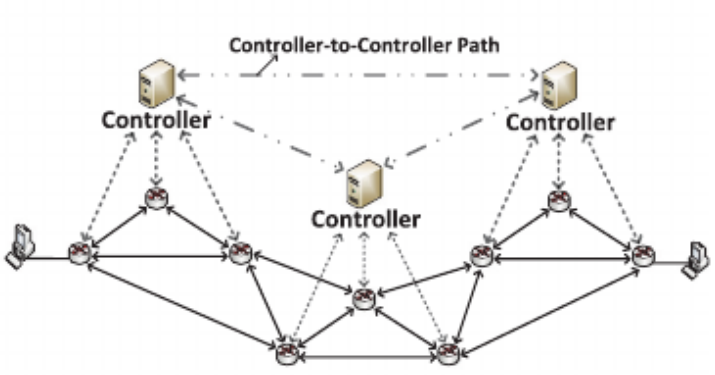


Figura 3: Padrão Distribuido Plano. Fonte: Karakus and Durresi (2017)

2.2.2 Padrão de Controladores Distribuídos Hierárquicos

O modelo hierárquico possui um controlador como principal que possui a visão global da rede e os controladores secundários. Os secundários possuem a perspectiva local, isto é, não detém entendimento total da rede, mas apenas do seu domínio e tratam das aplicações sob seu comando Karakus and Durresi (2017). A figura 4 mostra o padrão de controladores distribuídos hierárquicos

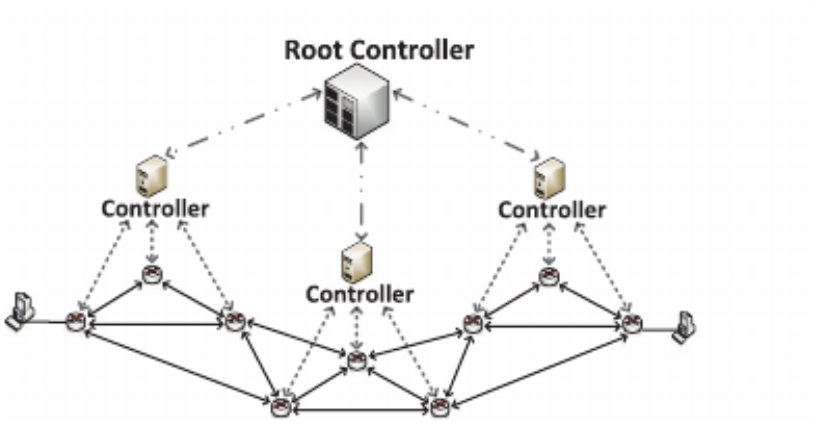


Figura 4: Padrão Distribuido Hierárquico. Fonte: Karakus and Durresi (2017)

2.2.3 Padrão de Controladores Distribuídos Híbrido

Esta abordagem difere das duas anteriores primeiramente nas tarefas dos planos. Os dispositivos do plano de dados tornam-se envolvidos por parte do controle e processamento da rede Karakus and Durresi (2017). As funções de controle são limitadas a estes dispositivos, apenas sendo concedido tarefas como: transferência de regras para outros dispositivos e deletar ou atualizar a tabela de fluxo Karakus and Durresi (2017). A figura 5 mostra o padrão de controladores distribuídos híbrido.

2.2.4 Redes Elásticas Distribuídas

Esta arquitetura tem como principio a inserção e remoção de controladores a medida que a necessidade da rede aumenta ou diminui. Entretanto, a ligação entre o controlador e switch é estaticamente configurada Dixit et al. (2013). Isto dificulta o desenvolvimento de uma arquitetura plural de plano de controle. Para solucionar este problema, o modelo de Redes Elásticas Distribuídas (Elastic Distributed Controller architecture - Elasticon) define uma piscina de controladores. Esta piscina pode ser ampliada ou reduzida de acordo com o tráfego e a carga de trabalho na rede.

Muitas soluções tem sido propostas para resolver o problema da escalabilidade e confiabilidade em SDN. Entretanto como já citado, muitos deles propõe ligação estática entre os dispositivos do plano de dados e controle, essa maneira de conexão torna-se um

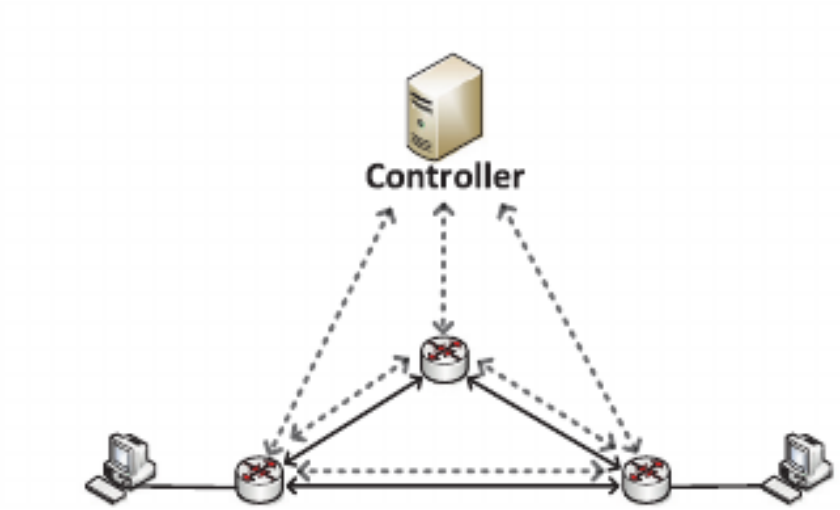


Figura 5: Padrão Distribuido Híbrido. Fonte: Karakus and Durresi (2017)

problema em redes reais que possuem variações de acordo com o momento Dixit et al. (2013). O tráfego e a carga da rede podem ser diferentes de dia e a noite, mudar de acordo com uma determinada época do ano e manter a conexão estática pode acarretar dois problemas: sobrecarga quando há poucos controladores ou subutilização dos mesmos Dixit et al. (2013).

No Elasticon, um switch é ligado à vários controladores (módulos), no qual dentre eles é eleito o mestre. Nesta disposição, todos os controladores possuem os dados dos dispositivos de encaminhamento Dixit et al. (2013). Cada módulo executa a função de um controlador centralizado, juntos realizam a eleição do responsável por um novo switch e realizam a migração deste, quando necessário, para outra gerência de controle Dixit et al. (2013).

A operação de transferência de switch e agregação/desagragação de controlador não é nativa, foi aproveitado uma operação do Openflow 1.3 que define os switchs como: mestre, escravo e igual. No processo de transferencia do switch, imaginemos que existem 2 controladores A e B. A é o mestre e B o escravo, para que seja realizada a operação com sucesso sem comprometer a segurança e a disponibilidade da rede, o controle B muda de estado para igual ao A, isto ocorre com A enviando mensagem para B informando o início do processo, B então, realiza role-request em direção a A, e este por sua vez faz o role-reply. Esta primeira etapa esta pronta, mas aqui todas as mensagens ainda são respondidas (coordenação do plano de dados) por A Dixit et al. (2013).

Na segunda etapa, o primeiro controlador envia um fluxo falso para o ativo, este tratará por primeiro o fluxo, e apagará a tabela de fluxo ligada ao primeiro controlador e passará o controle para o segundo, a partir de então o sucessor responderá as mensagens. Embora o fluxo de primário tenha sido apagado, este ainda continua mestre, pois para concluir a etapa 3, precisa responder todas as mensagens na fila e confirma-las ao ativo. Então, finalmente B realiza role-request e o switch torna A como escravo e B é o mestre Dixit et al. (2013).

2.3 Criptografia de Dados

Segundo BOTES and PENZHORN (1993) *apud* Portnoi (2005), criptografia é uma ciência que trata de tornar ininteligível a leitura do texto ou mensagem do remetente por entes que não possuem ou desconhecem o mecanismo de decodificação da mensagem ininteligível para inteligível. O processo de tornar uma mensagem ininteligível é denominado cifragem ou encriptação, enquanto que o processo inverso é chamado de decifragem ou decriptação. Texto cifrado ou cifra é o nome dado ao texto ininteligível gerado a partir do processo de cifragem. Texto plano é a mensagem, texto, informação sem codificação BOTES and PENZHORN (1993) *apud* Portnoi (2005) e Simmons (1979) *apud* Portnoi (2005).

Existem dois tipos de criptografia: simétrica (ou de chave secreta) e assimétrica (ou de chave pública). Na criptografia simétrica, existe somente uma chave que é utilizada no processo de cifragem e decifragem da mensagem, isto é, ambos os lados da comunicação devem possuir a mesma chave para que seja possível a troca de informações. Este tipo de criptografia apresenta alguns problemas, o primeiro deles está relacionado a segurança, pois para haver a comunicação é necessário inicialmente o compartilhamento da chave, e caso esta seja transmitida num meio inseguro de comunicação, a chave pode ser capturada por alguma entidade não autorizada. Outro problema desta abordagem é que para cada par de entidades de comunicação é necessário uma chave diferente, logo se um individuo deseja comunicar-se com n outros indivíduos, este necessitará guardar n chaves BOTES and PENZHORN (1993) *apud* Portnoi (2005) e Simmons (1979) *apud* Portnoi (2005).

A abordagem assimétrica utiliza duas chaves para cada entidade, a pública e a privada. A chave privada é de conhecimento apenas deste ser, e caso um outro indivíduo queira enviar mensagem criptografada para este primeiro, o segundo utilizará a chave pública, que é de conhecimento de todos, para cifrar a mensagem. Como somente a primeira entidade possui a chave privada, somente ele pode decriptar a mensagem criptografada e ter acesso ao texto plano. Essa abordagem garante maior segurança quando comparada à criptografia simétrica, pois a chave privada não será transmitida, não tendo o risco de ser capturada. Na criptografia assimétrica, a entidade de decriptação precisa conhecer apenas suas chaves pública e privada. É importante salientar que cada indivíduo possui um par de chaves, e caso outra entidade queira realizar comunicação segura com este, utilizará a chave pública do destinatário para criptografar informações BOTES and PENZHORN (1993) *apud* Portnoi (2005) e Simmons (1979) *apud* Portnoi (2005).

O mecanismo de autenticação proposto utiliza técnicas criptográficas para garantir a segurança durante o processo de autenticação, visando garantir a segurança das informações transmitidas pelas entidades da rede. As técnicas utilizadas serão introduzidas nesta seção, onde serão abordadas criptossistema ElGamal e ElGamal Limiar, e provas de conhecimento zero.

2.3.1 Criptossistema ElGamal

Segundo dos Santos Araujo (2008), o ElGamal é um criptossistema de chave pública que funciona de maneira probabilística. Foi concebido por Taher El Gamal e é baseado no problema de logaritmos discretos e Diffie-Hellman dos Santos Araujo (2008). Para melhor compreensão, será mostrado o passo a passo do criptossistema Elgamal, baseado na descrição em dos Santos Araujo (2008):

1. Primeiramente, há a definição de um número gerador g, bem como dois números primos representados por p e q.
2. Então, computa-se $p = 2q + 1$ para verificar que p é primo. Caso não seja, é necessário selecionar o valor novamente.
3. Um valor x pertencente ao conjunto de inteiros(Z) menor que q é selecionado aleatoriamente.
4. Computa-se $h = g^x mod p$, onde as chaves pública e privada serão respectivamente h e x.
5. Definida uma mensagem m, a cifragem é realizada a partir da seleção de um valor aleatório r que pertence ao conjunto Zq.
6. Após esta etapa, $\alpha = g^r mod p$ e $\beta = m * h^r mod p$ são computados, e o texto cifrado é definido pela tupla (α, β).
7. Por fim, a decifragem é feita a partir da computação de $m = \left(\frac{\beta}{\alpha^x}\right) mod\ p$

2.3.2 Interpolação Polinomial de Lagrange e Compartilhamento de Segredo de Shamir

Shamir (1979) definiu uma maneira de compartilhamento de segredo num grupo de entidades que não são confiáveis e possuem interesses conflitantes, de tal forma que a recuperação da informação por completa somente é obtida a partir do consenso de um subgrupo do total. Utilizando a interpolação polinomial de Lagrange, é possível realizar uma cifragem e dividir a chave em várias partes. Cada uma dessas partes não contém em sí nenhuma informação da chave por completo, então não necessariamente precisa ser protegida contra inspeção Shamir (1979).

Cada parte torna-se interessante, quando junto com um número limiar consegue recuperar a chave. Isto é chamado de limiar. Basicamente, para que a chave seja recuperada, é necessário que uma quantidade k dos n elementos em consenso queiram realizar a operação. Assim, basta utilizar o limiar (k,n) para recuperação.

O protocolo de compartilhamento proposto por Shamir é baseado na interpolação polinomial de Lagrange em que tendo k pontos bidimensionais de xe y com x' distintos

de tal forma que existe somente um polinômio $q(x)$ de grau $k-1$, de forma que existe $q(x_\mathrm{i}) = y_\mathrm{i}$ para cada elemento i. Para dividir a chave J em j_i partes, é definido o polinômio de grau k-1, na forma:

$$q(x) = a_0 + a_1 x + ... + a_{\mathrm{k\text{-}1}} x^{(\mathrm{k\text{-}1})}, \text{ tal que } a_0 = J$$

Esta equação é o polinômio gerador de cada parte, de tal modo que cada entidade i de 1 até n receberá uma parte, ou um ponto da polinomial de lagrange. Com um conjunto de k partes é possível gerar um valor que comporte-se similar ao J, caso k-1 entidades sejam utilizadas, simplesmente não será possível obter o comportamento da função que polinomial que foi gerada a partir de J. Outro ponto importante é que utiliza-se a aritmética modular, ao invés da real. Para isto, na computação da chave, é usado um primo p, a fim de garantir a maior exatidão da operação de recuperação. Outro fator para utilizar p é que no criptossistema de Elgamal, a cifra possui o dobro do tamanho da mensagem em texto plano, por conta das operações de multiplicação, e somente uma operação inversa, neste caso módulo, pode realizar a decriptação ElGamal (1985) A seguir é descrito o passo a passo.

1. Escolher um valor de p com muitos caracteres, e numericamente maior que J e n.
2. Escolher aleatoriamente os a_0, a_1 entre 0 e p aleatoriamente.
3. Cada parte da chave é calculada com *mod p*

Assim é possível dividir a chave J em $\mathrm{j_i}$ partes.

Com o objetivo de facilitar para o leitor, no capítulo 4, será demonstrado como é realizado a divisão da chave em múltiplas partes.

2.3.3 Criptossistema ElGamal Limiar

Este protocolo criptográfico constitui em uma versão limiar o algoritmo ElGamal descrito anteriormente, e considera a descentralização da chave privada entre os diversos membros de um determinado grupo o qual possui um tamanho n definido, e a decifragem é realizada apenas diante da participação do limiar descrito na seção 2.3.2.

O criptossistema de ElGamal Limiar é mais seguro que a versão centralizada. Isso se deve ao fato de uma possível entidade não confiável sozinha não ser capaz de decifrar o texto cifrado.

O funcionamento do protocolo ElGamal limiar, utilizado neste trabalho, será descrito a seguir.

1. Inicialmente, o gerador g e os números primos p e q são selecionados, assim como as chaves privada, x, e pública, h, como é definido na versão centralizada do El Gamal.

2. Um grupo S = 1, 2, 3, ..., n com n membros assim como um limiar t, são definidos.
3. Então, um polinômio de Lagrange definido por $f(z) = a0 + a1 * z + a2 * z^2 + ... + a(t-1) * z^{(t-1)}$ é selecionado aleatoriamente, sendo $a0 = x$.
4. Uma entidade confiável seleciona um identificador si e computa $xi = f(si)$, que é uma fração da chave, para cada um dos n membros do grupo. Após isso, cada um destes recebe uma fração da chave privada xi.
5. Define-se uma mensagem m.
6. O procedimento de cifragem é executado da mesma maneira que na versão centralizada do El Gamal, onde o texto cifrado é representado pela tupla (α, β).
7. A decifragem inicia com cada membro, identificado por si, computando $di = \alpha^{xi}$.
8. Por fim estas partes executadas são enviadas até uma entidade confiável, que computa $m = \left(\prod_{i \in S} di^{l(i,0,S)}\right)^{-1} \beta$ mod p ,

$$\text{sendo que: } l(i,0,S) = \left(\frac{\prod_{m \in S} -sm}{si - sm}\right) \text{modp.}$$

2.4 Provas de Conhecimento Zero

As provas de conhecimento zero, descritos em dos Santos Araujo (2008), constituem protocolos nos quais existem duas partes, sendo estas um provador e um verificador, onde o primeiro afirma afirma que conhece um determinado segredo e deve convencer o segundo acerca da veracidade deste, sem que as informações desta declaração sejam reveladas. Existem dois grandes grupos de provas de conhecimento zero que são os interativos e os não interativos. Nos interativos, o verificador interage com o provador para que seja possível averiguar que a declaração deste é verdadeira. Já nos não interativos, o verificador não necessita realizar uma interação com o provador para comprovar a veracidade. As provas de conhecimento zero utilizadas ao longo do trabalho pertencem ao tipo não interativo, e estão descritas nas subseções sequentes.

2.4.1 Teste de Equivalência de Textos Planos

O protocolo Teste de Equivalência de Textos Planos (Plaintext Equivalence Test - PET) da Costa et al. (2016) tem por objetivo verificar se dois textos cifrados possuem o mesmo texto plano, sem que as informações dos textos cifrados sejam reveladas. Seja um verificador que possua um par de chaves El Gamal definidas por h e x, sendo a primeira chave pública e a segunda privada, e também dois textos cifrados definidos por $(\alpha 1, \beta 1)$ e $(\alpha 2, \beta 2)$ gerados a partir da respectiva criptografia de dois textos planos m1 e m2 com h. Após receber $(\alpha 1, \beta 1)$ e $(\alpha 2, \beta 2)$, um verificador pode averiguar se os textos planos contidos nos textos cifrados são iguais, através dos seguintes passos:

- A partir dos parâmetros definidos anteriormente, um novo texto cifrado $(\alpha_{1,2}, \beta_{1,2}) = \left(\frac{\alpha 1}{\alpha 2}, \frac{\beta 1}{\beta 2}\right)$ é gerado
- Então, $(\alpha_{1,2}, \beta_{1,2})$ é decifrado.
- Caso o texto plano resultante seja 1, pode-se afirmar que $m1 = m2$, porém, se o resultado for diferente de 1, $m1$ é diferente de $m2$.

2.5 Conclusão do Capítulo

Neste capítulo foram apresentadas as bases para a proposta que será apresentada no capítulo 4. Fora explanado o conceito de redes definidas por software, distribuição do plano de controle nestas redes, criptografia e o sistema criptográfico de ElGamal que será utilizado no capítulo 4. A apresentação destes conceitos e trabalhos é importante para o entedimento do mecanismo de autenticação descentralizada que será demonstrado no capítulo mencionado neste parágrafo.

CAPÍTULO 3

Trabalhos Relacionados

3.1 Authflow: Mecanismo de Acesso e controle para Redes Definidas por Software

O modelo de autenticação proposto por Mattos and Duarte (2016) realiza a tarefa na camada 2 utilizando o endereço MAC. A arquitetura é composta pelo controlador, autenticador e o server Radius, quando uma entidade solicita acesso a rede é verificado junto ao autenticador se a entidade está autenticada. Caso a resposta seja não, será necessário realizar a tarefa.

3.1.1 Descrição

Inicialmente o host envia um pacote multicast com a mensagem start, o pacote deve conter a especificação IEEE 802.1x Mattos and Duarte (2016) o controlador redireciona para o autenticador que verifica no servidor radius se as informações são verdadeiras. Importante frizar que os dados são enviadas como mensagem do protocolo EAP é com este tipo de mensagem que o Radius realiza a validação. Caso a autenticação seja realizada com sucesso, o autenticador retorna uma mensagem de sucesso ao cliente e envia mensagem de autorização para o controlador, então a entidade tem acesso aos recursos da rede Mattos and Duarte (2016).

Importante salientar que o sistema permite a requisitantes acessar somente área de autenticação enquanto a validação da credencial não for realizada Mattos and Duarte (2016).

Neste modelo a tupla (mac, porta) é utilizada para verificar o estado do host e também o fluxo de pacotes.

3.1.2 Vantagens e Desvantagens

O projeto apresentado tem a vantagem de realizar a tarefa com baixa sobrecarga e garante controle de acesso refinado, visto que utiliza o endereço MAC, e não disponibiliza IP temporário Mattos and Duarte (2016). Também é possível definir acesso de acordo com o nível de privilégio de cada usuário, pois pode ser utilizado o protocolo MS-CHAP V2 e cada host pode acessar com usuário e senha, desta forma criar níveis na base de dados Mattos and Duarte (2016). A proposta permite que os controladores SDN utilizem a identidade do host como novo campo do fluxo para definir regras de encaminhamento Mattos and Duarte (2016).

Como desvantagem, o sistema proposto possui o fato de ser centralizado, caso um elemento malicioso ganhe o domínio do controlador pode alterar a regra de encaminhamento e desabilitar a autenticação ou encaminhamento de pacotes para o autenticador, pode ocorrer de o atacante ganhe o controle do autenticador e compromete o desenho de autenticação. Outro problema é que utiliza uma tecnologia específica, o banco de dados Radius, tornando novamente a rede refém de hardware específico e fabricantes não garantindo programabilidade na verificação e validação em sí. Esta ultima desvantagem pode ser vista como oposta a filosofia SDN.

3.2 Método de acesso confiável em Redes Definidas por Software

O trabalho apresentado por Liu et al. (2017) propõe um arquitetura complexa na tentativa de alcançar confiabilidade e segurança no processo de autenticação em Redes SDN.

3.2.1 Descrição

Liu apresenta uma arquitetura composta por 3 camadas: acesso a rede, avaliação de integridade, nível (mensuração) de integridade. A primeira delas consiste em autenticar entidades identificadas e comunicação com redes tradicionais. concede ao host acesso de acordo com o nível de privilégios da entidade, a partir das diferentes políticas de acesso Liu et al. (2017). A terceira é que realiza de fato a verificação da integridade da entidade.

A arquitetura também possui três entidades: o Requisitante de acesso (Access Resquestor - AR), ponto de execução da política (Policy enforcement Point - PEP) e o ponto de decisão da Política (Policy Decision Point - PDP) Liu et al. (2017).

O AR é o ente que deseja acessar a rede protegida, o PDP realiza o reconhecimento da credibilidade do AR e decide se este pode acessar a rede. O PEP realiza a certificação confiável com base nas informações repassadas pelo AR.

A proposta possui um mecanismo complexto de autenticação, é utilizado a criação

de uma conexão similar a TCP que realiza funções muito semelhantes a empregada no protocolo, mas sendo transmitido e verificado a autenticidade do usuário. E também, as três entidades são subdivididas de maneira que seja cada parte exerça sua função separadamente. O processo de autenticação se dá como descrito por Liu et al. (2017):

- No AR, existe cliente acessor que gera um valor de integridade para cada requisitante, seria o equivalente ao hash.O valor de integridade é salvo no PCR PCR (Plataform configuration register). O dispositivo de acesso (entidade do AR) utiliza o valor de integridade PCR junto com os dados do usuário para realizar autenticação.
- O cliente envia dados de autenticação para o servidor via Openflow e através dos switchs.
- O servidor de autenticação recebe os dados então verifica o valor integridade e os dados do usuário, então toma a decisão a respeito do acesso e envia essa decisão ao controlador.
- O controlador recebe a decisão do servidor e se for favorável insere o host na tabela de fluxo, então o switch realiza o bloqueio ou liberação de acordo com a inserção na tabela realizada pelo controlador.

.

3.2.2 Vantagens e Desvantagens

Uma das principais vantagens deste modelo é a realização de verificação da integridade da entidade solicitante ao acesso na rede, isto garante uma segurança extra no trabalho de Liu. Uma das desvantagens é o excesso de troca de mensagens entre os elementos que realizam a autenticação, isto pode causar sobrecarga na rede se o número de clientes for grande, ocorrendo DDOS passivo. Outro problema seria criar um protocolo similar ao TCP dentro deste pra garantir a troca dos dados de autenticação, embora traga segurança demanda criação de túnel de conexão dentro de outro, isto poderia facilmente ser resolvido substituindo o protocolo por Secure Socket Layler, simplificando a solução e garantindo a segurança.

3.3 Flownac

O protocolo proposto por Matias et al. (2014) tenta garantir o acesso através da escolha do tipo de serviço desejado pelo requisitante, isto é, o tráfego é associado ao serviço, e este tráfego deve ser devidamente identificado como único Matias et al. (2014). A proposta utiliza como base o sistem NAC de serviços

3.3.1 Descrição

A identificação consiste em validar e categorizar os quadros do usuário de acordo com o recurso desejado e realizar permissão ou negação para cada um deles (quadro) de acordo com o serviço associado Matias et al. (2014). E os requisitantes necessitam de autenticação para cada recurso da rede solicitado Matias et al. (2014).

O proposta realiza a identificação do host através de controle de acesso à porta. O mecanismo define uma porta virtual associada a cada endereço MAC, esta tupla permite acesso a somente um serviço por padrão (em NAC), mas o uso de VLAN disponibiliza o acesso simultâneo de diferentes recursos da rede. Embora todos as mensagens sejam recebidas por uma porta física, este mecanismo garante administração refinada de cada usuário Matias et al. (2014).

A arquitetura é composta por três elementos: o suplicante, autenticador (PEP) e o servidor de autenticação (PDP).

No contexto de SDN, o protocolo de autenticação sofre algumas alterações, pois como existe a separação de planos, é necessário realizar adaptação do mesmo para esta nova abordagem. O IEEE define o padrão 802.1X que possui três componentes PAE (com estado), PAC (sem estado) de acordo com Matias et al. (2014). Existe também a camada de gerenciamento de interface que comunica PAE e PAC, além de realizar o controle do status da porta.

Para SDN, os agentes necessários no processo são: Datapath SDN, função de autenticação da rede e o controlador.

O Datapath pode assumir perfeitamente a função de PAC, pois distingue dois tipos de tráfego. Este modelo de autenticação utiliza o protocolo EAPol sobre quadros ethernet, o datapath é capaz de diferenciar os quadros de autenticação dos demais através do tipo 0x888E explicitado no cabeçalho Matias et al. (2014). O autenticador realiza o recebimento e envio dos quadros EAPol com o usuário, e converte para o formato radius de maneira a permitir a comunicação com o servidor de autenticação Matias et al. (2014). O controlador SDN realiza a inserção de informações na tabela, sejam estas de autenticação ou de tráfego. Em termos gerais, o controlador atua como o PAE e controla o PAC (datapath) e as regras que serão seguidas por este . A comunicação entre PAE e PAC é realizada em OpenFlow que desempenha o papel do LMI.

3.3.2 Vantagens e Desvantagens

Talvez a principal vantagem do Flownac é operar em níveis mais baixos, não precisando de IP, regras de DNS. Isto pode garantir mais controle e segurança, visto que não é necessário entregar ao usuário endereço IP temporário. Por outro lado, a proposta não menciona uso de multiplos autenticadores, e esta questão pode ser um problemas, visto que caso um atacante consiga controle do autenticador ou do servidor o flownac estará comprometido.

3.4 Uma solução de acesso a rede combinando OrBAC e SDN

A proposta de Aschoff et al. (2017), utiliza também o S-NAC como base para realizar a autenticação. Desta forma, apresenta similaridades com outras propostas mostradas neste capítulo, mas neste modelo todos os elementos necessitam ser autenticados, garantindo mais segurança na rede. A política de acesso é definida de acordo com os dados inseridos nas tabela do switch.

3.4.1 Descrição

Um ou vários *switchs* estão no centro da rede sendo gerenciados pelo controlador. Os *switchs* funcionam como um elemento PEP de maneira que cada entidade, inclusive servidores necessitam de autenticação para acessar a rede e possuem múltiplas tabelas onde os fluxos são instalados Aschoff et al. (2017). No controlador é executado o Agente SNAC que realiza a troca de tráfego entre o suplicante e o autenticador, bem como manipulação das tabelas através do controlador Aschoff et al. (2017). O autenticador possui funções similares ao Radius IEEE 802.1X Aschoff et al. (2017). Por fim, os hosts possuem um ID que é utilizado em todo processo de autenticação Aschoff et al. (2017).

O controle de acesso é realizado através de uma combinação entre entradas da tabela e ações relacionadas, ambos são traduzidos em políticas definidas pelo administrador da rede. As políticas são definidas utilizando o modelo OrBAC. Este modelo permite amoldar o usuário, servidores e serviços. Desta maneira, os usuários acessam alguns serviços definidos previamente em detrimento de outros, como se houvesse níveis de permissão. Existe também um mecanismo de temporização de acesso, isto é,o usuário logado tem um tempo máximo de utilização do serviço na sessão até ser desligado da rede.

O protocolo é similar ao IEEE 802.1x e para o cliente, o processo de autenticação é semelhante. A diferença para o padrão do IEEE, além da utilização de switch com Openflow, controlador SDN é a utilização do servidor hostpad Aschoff et al. (2017).

3.4.2 Vantagens e desvantagens

A vantagem clara da proposta é o refinamento do acesso, usuários ou dispositivos da rede podem ter acesso permitidos ou negados a determinados recursos de acordo com as políticas definidas pela gerencia da rede. A desvantagem é que existe um autenticador e banco de dados centralizado, isto pode comprometer a segurança da rede caso o autenticador ou o banco de dados seja invadido.

3.5 Um protocolo seguro de autenticação em SDN baseado em criptografia de chave pública

O trabalho de da Costa et al. (2016) propõe um mecanismo de autenticação utilizando o o controlador como participante ativo do processo. No processo, o controlador possui uma chave privada. Para que haja autenticação, inicialmente o cliente deve comparecer a um local seguro afim de registrar sua chave de acesso numa base de dados, então a partir disto, o texto é cifrado pela chave pública h e o gerador g resultando em c que será guardado pelo controlador no índice `i`.

Então caso o cliente queira autenticar-se envia requisição com c e i, o switch envia para o controlador que verifica se o requisitante conhece o texto plano através da prova por ele enviada, então pega a cifra passada pelo usuário e o identificador.

Os passos para realizar autenticação são os seguintes, segundo da Costa et al. (2016) (texto na íntegra)

1. A entidade U solicita utilizar algum recurso da rede. Como não há regras nos *switches* que tratem pacotes enviados pela entidade U, este pacote é redirecionado para o controlador;

2. O controlador verifica se a entidade solicitante está autenticada atualmente na rede. Ao identificar que a entidade não está autenticada, o controlador solicita as informações necessárias por parte da entidade para a realização da autenticação: a credencial cifrada c, a prova de que a entidade conhece o texto plano confidenciado pelo texto cifrado c, e o valor i recebido pela entidade no momento do seu cadastro;

3. A entidade U (nessa ordem):

   - cifra sua credencial p utilizando a chave pública h do controlador. O texto cifrado resultante denotado por c;
   - utiliza p para computar a assinatura de Schnorr, resultando na tupla Y.
   - envia a tupla $\{c, i, Y\}$, a qual é redirecionada pelos switches para o controlador;

4. Após receber a tupla $\{c, i, Y\}$, o controlador:

   - verifica a prova contida na tupla Y , retornando um valor booleano B1.
   - o autenticador utiliza o valor i para eficientemente recuperar a credencial cifrada ci da base de dados, e usa a função PET com sua chave privada x para verificar se c e ci contém o mesmo texto plano. A função retorna um valor booleano *B2*.
   - Se *B1* e *B2* retornarem verdadeiro, então a autenticação é finalizada com sucesso. Caso contrário, a autenticação falha. Para finalizar, o controlador envia uma resposta para a entidade U.

3.5.1 Vantagens e Desvantagens

O modelo desta seção possui a vantagem de um processo de autenticação rápido, se comparado com propostas descentralizadas, visto que a operação toda é executada num único elemento. A prova de conhecimento garante que o requisitante de acesso a rede é de fato o usuário que realizou o cadastro, isto evita que caso no transporte, os dados sejam capturados o atacante não pode realizar a autenticação. Existe também o benefício de que o controlador participa do processo, desta forma os administradores preocupam-se com a segurança do controlador, e não de um sistema de autenticação. O trabalho de da Costa et al. (2016) é flexível e adaptativo e pode ser utilizado em diferentes cenários de rede.

A desvantagem é que caso o atacante ganhe o domínio do controlador, conseguirá ter acesso a toda a rede, pois há somente um ponto de falha no processo.

3.6 Tabela comparativa dos trabalhos relacionados

Trabalho	Descentralizado	Servidor de Autenticação	Utiliza controlador processo de autenticação
Authflow	Não	Sim	Não
Método de acesso confiável em Redes Definidas por Software	Não	Sim	Não
Flownac	Não	Sim	Não
Uma solução de controle de acesso a rede combinando OrBAC e SDN	Não	Sim	Sim
Mecanismo de autenticação centralizada em redes definidas por software.	Não	Não	Sim

Figura 6: Tabela comparativa trabalhos relacionados.

3.7 Conclusão do Capítulo

Neste capítulo foram apresentados todos os trabalhos que propõe autenticação de usuários em SDN, bem como as vantagens e desvantagens de cada proposta. Vale ressaltar que as propostas apresentadas utilizam autenticação centralizada, e para 3 delas é utilizado

um servidor centralizado de autenticação. A utilização deste servidor centralizado torna-se um problema na perspectiva de segurança, pois concentra a falha em um único ponto.

CAPÍTULO 4

Proposta

Diante dos trabalhos desenvolvidos para garantir controle de acesso em SDN e a escassez de soluções descentralizadas, neste capítulo será apresentado um mecanismo de autenticação descentralizado com o intuito de prover mais segurança e disponibilidade dos recursos oferecidos. A proposta é baseada no modelo centralizado de da Costa et al. (2016) .

4.1 Proposta Descentralizada

Enquanto boa parte dos modelos descritos no capítulo 3 não utilizam o controlador como entidade ativa no processo, esta proposta visa utiliza-lo na autenticação. A ideia é utiliza-lo de maneira descentralizada,isto é, quando um controlador receber o pacote de autenticação, será escolhido por este outros controladores aleatoriamente para que realizem o cálculo parcial e entregue ao primeiro, que por sua vez, com os dados parciais realizará a computação final e decidirá por permissão ou negação dos serviços ao usuário.

Os controladores são adicionados ou removidos da rede dinamicamente, isto é, de acordo com a necessidade da rede baseado em Dixit et al. (2013). A utilização da arquitetura Elasticon foi estabelecida para montar um cenário de redes onde há utilização de vários controladores com o intuito de validar a proposta em contexto de descentralização do plano de controle. Essa abordagem tem impacto na autenticação, visto que para a computação parcial, cada controlador recebe parte da chave privada,e caso o tráfego da rede mude significativamente, será necessário a redistribuição da chave e definição de um novo limiar (número mínimo de entidades participantes no processo de acordo com o total).

Esta proposta utiliza criptografia de chave pública baseado no criptossistema

limiar de ElGamal (1985) para geração de chaves, onde é disponibilizado para os requisitantes a chave pública e o gerador,a fim de que os usuários gerem mensagens encriptadas antes de transferir seus dados de autenticação.

4.1.1 Visão geral

O mecanismo de autenticação proposto neste trabalho tem por objetivo trazer uma alternativa para acesso a recursos das redes SDN, garantindo acesso seguro e escalabilidade da mesma. Diversos outros modelos foram propostos, e muitos deles possuem bons mecanismos de autenticação, mas não há ainda a preocupação com autenticação de redes definidas por software de alta demanda, redes que possuem mais de um controlador. Um dos problemas atuais em SDN é justamente disponibilizar serviços para uma rede de larga escala, visto que o plano de controle e dados nestas malhas passam a sofrer problemas quando a cargar de trabalho aumenta em termos substanciais Dixit et al. (2013), como foi descrito no capítulo 2.

O trabalho oferece um mecanismo que aproveita-se de múltiplos controladores da rede para implementar autenticação, aproveitando-se de recursos já existentes na rede, não sendo necessário adicionar hardware, protocolos de redes ou softwares específicos de fabricante.

Em resumo, as entidades da proposta são: requisitantes e controladores. Um requisitante desempenha o papel de usuário. Ele cadastra-se no sistema de autenticação e posteriormente faz requisições para utilização dos serviços da rede, passando pelo processo de autenticação para isso. Os controladores são responsáveis por autenticar usuários. É gerado um par de chaves (pública e privada) para eles, que será usado no processo de autenticação. Cada controlador recebe uma parte da chave privada. A chave pública correspondente é usada pelo requisitante para criptografar mensagens de autenticação de maneira que os controladores possam autentica-los.

O controlador pode assumir diferentes papeis em momentos distintos. Desta forma, será definido aqui as nomenclaturas controlador receptor e controlador autenticador. O controlador receptor é responsável por receber uma mensagem de autenticação de um requisitante (contendo a sua credencial, e.g. , senha). Após recebê-la, ele realiza um cálculo inicial com essa mensagem e encaminha o resultado desse cálculo à um conjunto de controladores autenticadores. Um controlador autenticador é responsável por receber o resultado do cálculo inicial e realizar um cálculo parcial com esse resultado. O controlador autenticador então envia o resultado do cálculo parcial para o controlador receptor. Este, por sua vez, deve reunir um determinado número de cálculos parciais para a realização de um cálculo final, que finalmente definirá se o requisitante autenticou-se com sucesso na rede. Note-se que que ambos os tipos de controladores realizam também a tarefa do plano de controle de uma SDN.

A utilização de várias entidades de autenticação serve para garantir que, caso um controlador seja comprometido, nem o esquema, tampouco a segurança da rede será

comprometida. Tal característica baseia-se no fato de que no El Gamal limiar a decifragem de uma mensagem só ocorre se um número T de um conjunto com N controladores (onde T é menor que N) computarem o cálculo parcial da mensagem de autenticação enviada pelo requisitante. Em outras palavras, isso significa que para ocorrer a permissão de acesso, é necessário um número T de controladores da rede do total de N destes para validação de acesso.

4.1.2 Geração do par de chaves e cadastro dos requisitantes

No mecanismo proposto, o primeiro passo realizado é a geração do par de chaves dos controladores. Após a definição do número N de controladores que irão compor o plano de controle e de um limiar T, é gerado um par de chaves do El Gamal limiar para os controladores. A chave privada é então compartilhada entre os N controladores através do uso do esquema de compartilhamento de segredo de Shamir. Cada controlador tem conhecimento apenas de sua chave parte de chave privada e esta não deve ser divulgada para qualquer outra entidade. O comprometimento de T partes de chave pode levar ao comprometimento do sistema de autenticação da rede.

O compartilhamento da chave privada é realizado utilizando o esquema de compartilhamento de segredo de Shamir, este esquema utiliza interpolação polinomial de Lagrange dos Santos Araujo (2008), para gerar chaves parciais a cada uma das entidades. Inicialmente são gerados dois números aleatórios *a1* e *a2*. Estes necessitam ser números grandes (na proposta foi utilizado de 1024 bits). Estes dois valores são compartilhados entre todos os controladores autenticadores. Também será necessário o identificador de cada um dos controladores, a chave privada inteira e o primo *q*. A equação a seguir demonstra o processo de criação da chave parcial.

$$sec = (a2 * (id^2) + a1 * id + s) \mod q$$

a1 e a2: números aleatórios de 1024 bits gerados aleatoriamente.

id: identificador do controlador autenticador.

s: chave privada completa;

q: número primo gerado com a chave privada inteira.

O valor de *sec* (segredo parcial) é guardado por cada controlador autenticador e somente este sabe o segredo. É importante que na geração destas chaves os controladores envolvidos estejam conectadas apenas localmente umas com as outras, sem saída para internet. Mas como durante a execução da rede o valor do limiar pode mudar dinamicamente, uma possível solução é a criação de uma outra rede (e.g. subrede) em que apenas controladores autorizados à desempenhar o papel de autenticador comunicam-se entre sí para trocar os valores de $a1$ e $a2$, bem como a transmissão da chave privada completa.

Note-se que esta definição de transmissão dos elementos não é escopo fechado. O ideal, em casos de plano de controle centralizado ou plano de controle descentralizado de

tamanho fixo é utilizar a troca dessas informações antes de disponibilizar qualquer serviço a usuários de maneira a garantir maior segurança. Além disso, é importante frisar que a chave privada completa não fica salva em cada uma das entidades, após a geração das chaves parciais, os elementos irão guardar apenas suas respectivas chaves parciais.

Utiliza-se o compartilhamento de segredo de Shamir com o intuito de distribuição segura da chave privada, pois se a chave fosse apenas dividida o texto plano em partes e compartilhada com cada um dos elementos autenticadores, poderia gerar falha de segurança caso um elemento malicioso conseguisse acessar diferentes partes da chave e obter o todo, comprometendo a segurança. O modo de compartilhamento utilizado neste trabalho, gera um outro valor para cada entidade a partir da chave original. Isto garante que a chave privada original seja preservada e desconhecida caso ocorra ataque, além de outros benefícios que serão mostrados em seções a seguir.

Assim que o par de chaves for gerado e a chave privada distribuída entre os controladores autenticadores, requisitantes podem cadastrar-se no sistema de autenticação. O cadastro consiste na criação de um identificador e uma credencial para o requisitante. A credencial é então criptografada com a chave pública dos controladores. O texto cifrado resultante é armazenado em um banco de dados junto ao identificador do requisitante. Todos os controladores têm acesso livre a esse banco de dados. O identificador e a credencial são disponibilizados ao requisitante para que ele possa utiliza-los no processo de autenticação posteriormente. O cadastro pode ser realizado tanto de maneira presencial quanto através de um formulário online. Entretanto, o cadastro presencial é preferido devido a sua maior segurança.

Como é utilizado a descentralização da chave privada e esta é realizada de acordo com o El Gamal limiar, primariamente é vital definir o número de controladores na rede. E partindo da perspectiva que a proposta está utilizando a arquitetura Elasticon Dixit et al. (2013), seria custoso para a rede a geração do par de chaves dos controladores toda vez que houvesse mudança no número de controladores. Desta forma, afim de minimizar computações referentes a geração do par de chaves dos controladores, a mudança do limiar e sua consequente geração de um novo par de chaves somente serão realizados quando a carga de trabalho na rede aumentar significativamente, independente se novos controladores forem adicionados para gerência da rede em curta escala.

4.1.2.1 Interpolação Polinomial de Lagrange

Inicialmente, a chave privada está centralizada numa única entidade da rede, caso um usuário malicioso consiga o controle desta entidade, será possível obter a chave privada e consequentemente comprometer a segurança da rede. A interpolação de Lagrange permite obter pontos distintos de uma função e a partir destes conhecer o comportamento da função, neste aspecto, a interpolação é muito útil no trabalho, pois através dela é possível decriptar a mensagem do usuário, mesmo não possuindo a chave completa, utilizando o segredo de shamir de cada controladores de passo 2, como se fossem pontos distintos da função,visto que para gerar o segredo de cada um, foi utilizado um polinômio. Utilizando

o princípio descrito pelo matemático, é possível verificar a mensagem utilizando k controladores do total destes, que em outra seção foi denominado limiar. A utilização deste mecanismo garante mais segurança pra rede, uma vez que para realizar autenticação, é necessário que pelo menos um número k de entidades realizem o cálculo parcial, desta forma, se um dos controladores de passo 2 for comprometido não haverá risco na segurança,dado que para ter acesso são necessários mais alguns elementos que junto destes, realizarão o processo. Pois utilizando o principio da interpolação de Lagrange, só será possível obter a chave toda com o conjunto de k partes, neste caso não necessariamente a chave, mas o comportamento da função que permite decriptar a mensagem.

4.1.3 Requisição de autenticação e o cálculo inicial

Quando um switch recebe um pacote de um requisitante, verifica se o mesmo possui permissão de acesso a rede através das regras impostas pelos controladores. Caso não, envia para o controlador receptor a fim de tratar o pacote. Caso o pacote seja de requisição de acesso aos recursos da rede, o controlador receptor solicita os dados de autenticação do requisitante. Este então criptografa sua credencial com a chave pública dos controladores (conhecida por todos os requisitantes) e envia o texto cifrado resultante, juntamente com o seu identificador, para o controlador receptor. O controlador receptor recebe os dados, recupera do banco de dados a credencial cadastrada criptografada do requisitante (através do identificador enviado) e realiza um cálculo inicial. Esse cálculo consiste na divisão do texto cifrado enviado pelo requisitante pelo texto cifrado cadastrado. Isto é, seja o texto cifrado enviado composto por α_1, β_1 e o texto cifrado cadastrado composto por α_2, β_2, então o cálculo inicial consiste em $\alpha_1/\alpha_2, \beta_1/\beta_2$.

4.1.3.1 Cálculo Parcial

Após realizar o cálculo inicial, o controlador receptor envia o resultado desse cálculo para os controladores autenticadores junto com os identificadores de todos os controladores autenticadores que participarão do processo de autenticação. Quando os controladores autenticadores recebem os dados enviados pelo controlador receptor, eles realizam o cálculo parcial da autenticação. Primeiramente é realizado o cálculo do Lagrange, utilizando os identificadores. Levando em conta a ordem dos identificadores, será obtido o valor do Lagrange. No cálculo o identificador dos outros controladores autenticadores é multiplicado no numerador e cada um deles é subtraído do identificador do controlador em questão no denominador, como mostrado na função a seguir:

$$lagrange = \frac{x1 * x2}{(x0 - x1) * (x0 - x2)} \text{ , onde:}$$

x0 = Identificador do autenticador que obterá o lagrange.

x1, x2 = identificador dos outros autenticadores participantes do processo.

É importante frisar que caso o denominador resulte num valor não inteiro, será utilizado o inverso do valor, pois nas operações seguintes é utilizado operações com módulo, e o inverso possibilita o uso do resto congruente, a fim de garantir o cálculo final correto.

Possuindo os itens segredo de Shamir, dado do cliente e o primo p é possível realizar o cálculo parcial, obtido a partir de:

$$cp = (alfa^{si}) \mod p, \text{ onde:}$$

cp: resultado do cálculo parcial

alfa: valor alfa resultado do cálculo inicial realizado pelo controlador receptor.

si: parte de chave privada do controlador autenticador

p: número primo gerado na etapa de geração do par de chaves.

A entidade então envia para o controlador receptor o valor *mp* e o *lagrange* para que seja possível o cálculo final que será explanado na seção seguinte.

4.1.4 Computação Final

Depois que os controladores autenticadores realizam seus cálculos parciais, eles transferem ao controlador receptor os seus resultados obtidos. Em posse das informações geradas pelos controladores autenticadores e do resultado do cálculo inicial, o cálculo final é realizado pelo controlador receptor. Caso o resultado do cálculo final seja igual à 1, isso significa que a credencial transmitida pelo requisitante é igual à credencial cadastrada e então é permitido a ele acesso a rede. O cálculo final é realizado conforme os passos a seguir.

$$da = di^{l0} \mod p$$
$$db = dj^{l1} \mod p$$
$$dc = dk^{l2} \mod p$$

onde: di, dj, dk são os cálculos parciais de cada controlador autenticador e $l0$, $l1$ e $l2$ são seus respectivos Lagranges. Com esses dados é possível obter o resultado cf do cálculo final:

$$j = da * db * dc$$
$$cf = (j^{p-2} \mod p) * beta) \mod p, \text{ onde:}$$

j: valor intermediário. beta: valor beta resultado do cálculo inicial realizado pelo controlador receptor.

No cálculo final, o j foi utilizado apenas como valor intermediário, a fim de facilitar o entendimento, mas poderia ser facilmente inserido diretamente na função que gera resultado do cálculo final. Note-se que em vários momentos é utilizado *modp* ou *modq*, o uso deste artifício é para reduzir o tamanho dos valores gerados, pois no processo inteiro são utilizados valores com grande quantidade de caracteres, sem este artifício seria impossível realizar os cálculos.

Para melhor entendimento, foi criado uma topologia exemplo do mecanismo de autenticação.Na figura 6 é mostrado um cliente que tenta autenticar-se na rede que possui 6 controladores. Nesta topologia, poderia ser utilizado um limiar igual a 3, isto é, para que ocorra autenticação seria necessário pelo menos 3 controladores realizando o cáculo parcial, além do controlador de passo 1 que recebeu o pacote do usuário.

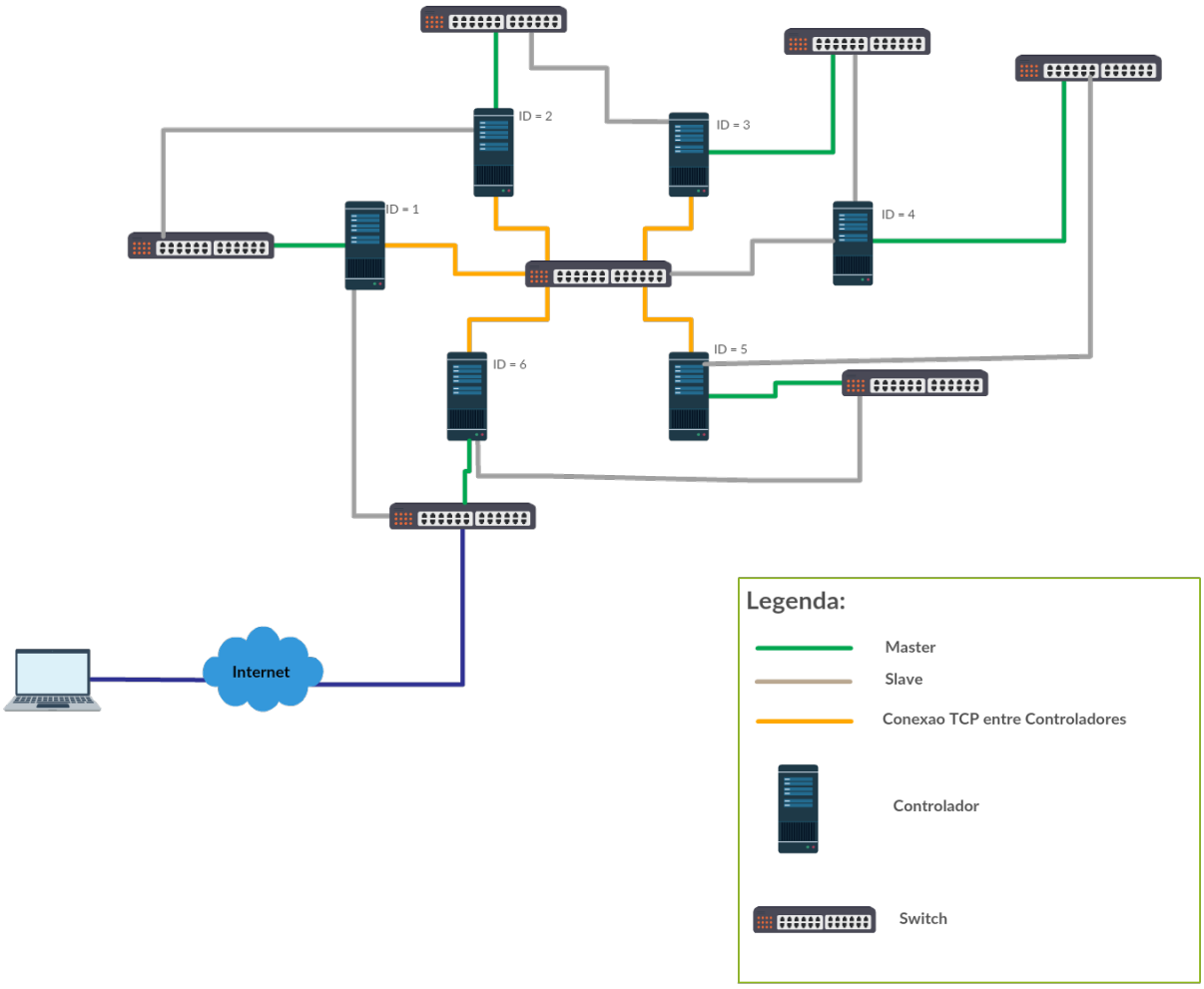


Figura 7: Topologia do esquema de autenticação aplicada na arquitetura Elasticon.

4.2 Resumo das etapas do mecanismo

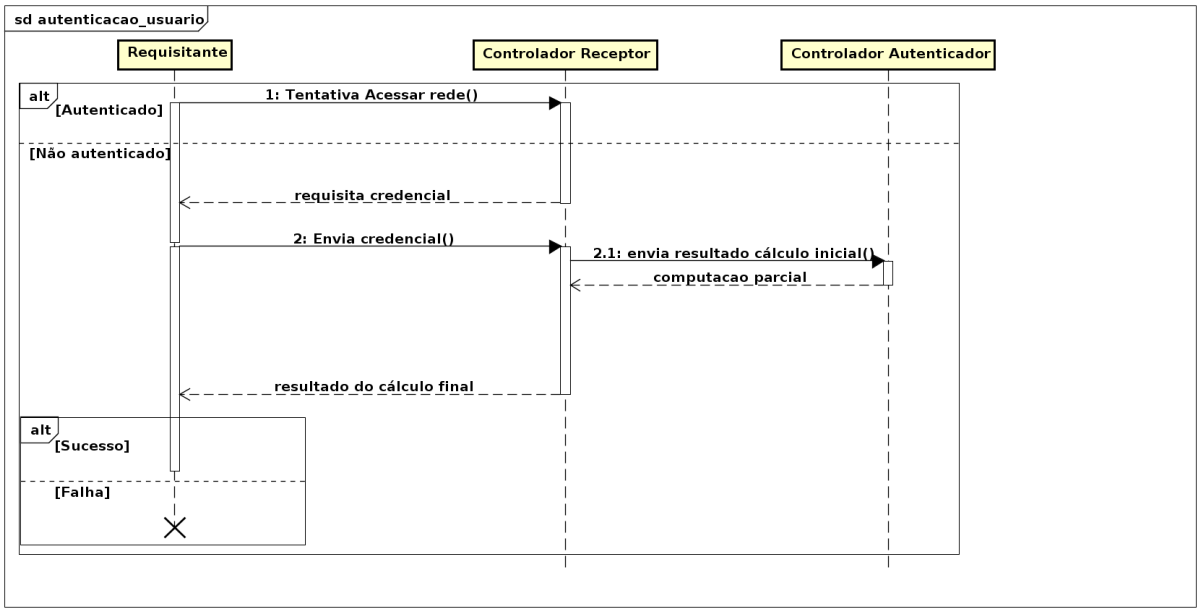


Figura 8: Diagrama da sequencia de autenticação

De maneira a sumarizar o mecanismo, a seguir é demonstrado enumeradamente cada um dos tópicos explicados anteriormente.

1. Divisão da chave privada entre os controladores, um número N de controladores.
2. Recebimento dos pacotes e tratamento dos mesmos.
3. Requisitantes que desejam autenticar-se enviam sua credencial criptografada, uma tupla α e β.
4. O controlador receptor realiza o cálculo inicial e escolhe 3 dos 5 controladores restantes para realizar o cálculo parcial da autenticação.
5. Cada um dos controladores autenticadores escolhidos recebe o resultado do cálculo inicial e o identificador dos controladores escolhidos.
6. Cada um realizará o cálculo de Lagrange com os identificadores recebidos.
7. É realizado o cálculo parcial por cada controlador autenticador;
8. Os controladores autenticadores enviam para o controlador requisitante o Lagrange e o cálculo parcial;
9. Com o Lagrange e o cálculo parcial de cada controlador autenticador, o controlador receptor realizar o cálculo final;
10. Caso o resultado seja igual a 1, significa que o requisitante autenticou-se com sucesso. Caso contrário, a autenticação falha.

4.3 Conclusão do Capítulo

Neste capítulo foi apresentado mecanismo de autenticação descentralizada de usuários em redes definidas por software, bem como o cenário utilizado para a implementação da proposta deste trabalho e o passo a passo do criptossistema de Elgamal aplicado na autenticação de usuários.

CAPÍTULO 5

Experimentos e Resultados

5.1 Descrição da Simulação e Cenários

Para realização validação da proposta foram realizados alguns testes em diferentes cenários, descritos a seguir.

5.1.1 Cenários de Teste

Neste cenário, foi utilizado seis controladores cada um ligado como *master* de um switch e *slave* de outro com o intuito de garantir caso o master seja comprometido ou haja sobrecarga o *slave* assume o controle do switch em questão. Um cliente tentando realizar a autenticação, como mostra a figura 7. A utilização de tantos controladores na autenticação é para seguir o principio de escolha dos participantes da autenticação (k,n) do El Gamal limiar, onde k é o número de entidades que realizaram a segunda etapa, e n é o número de autenticadores que contém uma parte da chave privada gerada no compartilhamento de segredo de shamir.

Inicialmente, será verificado como a rede comporta-se com apenas uma requisição. Este cenário será o ponto de partida, e servirá como base para avaliar os próximos cenários em que o número de clientes fazendo requisições, aumentará substancialmente.

O cenário 2 também possui 6 controladores, a diferença é que dessa vez são 10 clientes tentando realizar a autenticação.

Os cenários 3 e 4 serão similares ao cenário 2, a única diferença é que no cenário 3, ao invés e 10 *hosts*, serão 30 e no cenário 4 serão 50 hosts.

O incremento do número de requisitantes é para verificar se há sobrecarga do es-

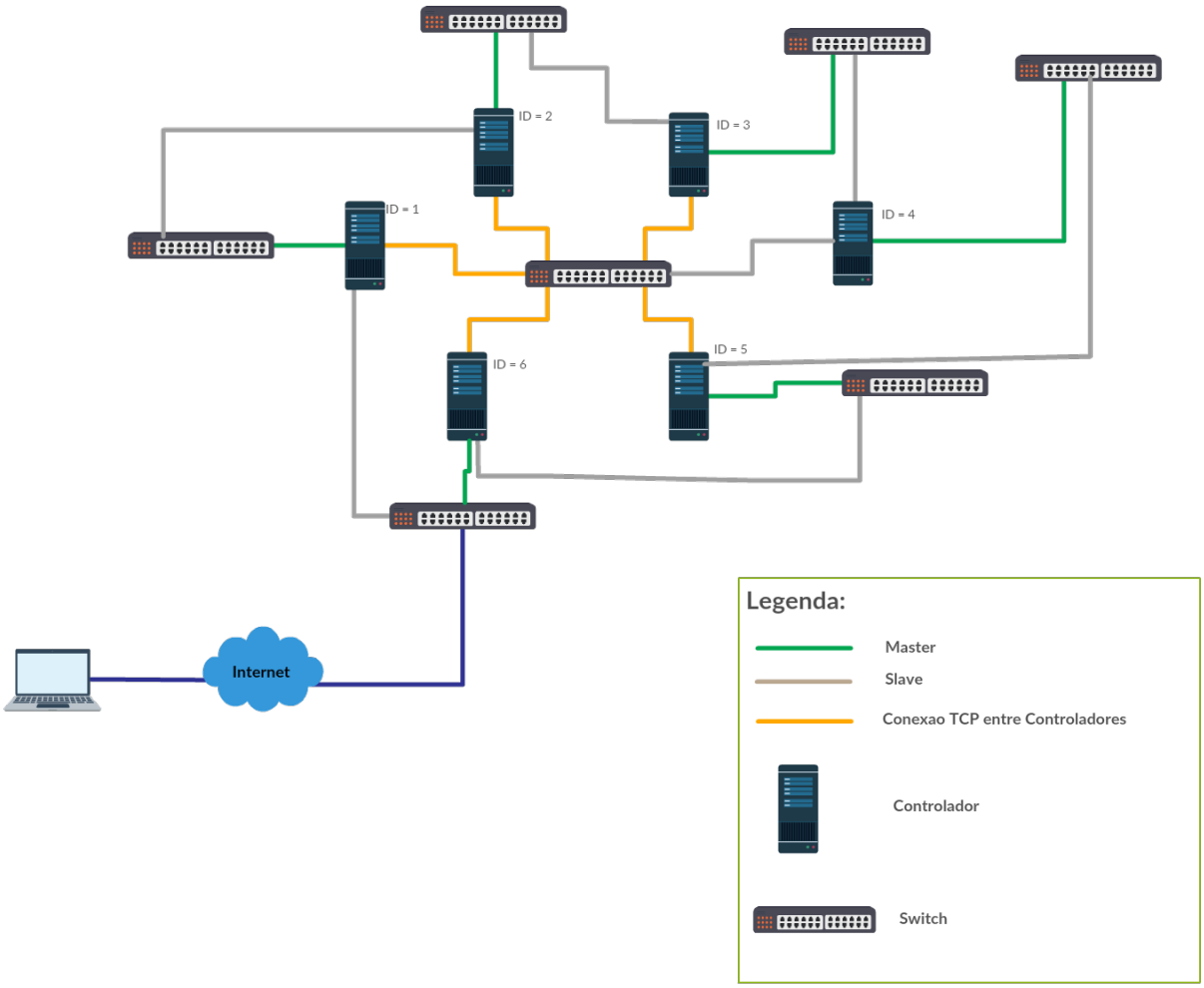


Figura 9: Topologia de autenticação com 1 cliente.

quema de autenticação sobre a rede, visto que são realizados vários cálculos matemáticos em cada entidade participante do processo de autenticação. O tráfego de dados provavelmente é superior ao que acontece em esquemas centralizados de autenticação, pois no segundo, um único dispositivo realiza a autenticação.Este acréscimo de transferência de informação, talvez em redes que possuem muitos serviços e uma grande quantidade de equipamentos conectados, causar redução significativa na largura de banda disponível.

Cenário 1

Foram realizas 50 simulações utilizando a topologia da figura 7, e após coletar os dados foi possível obter o seguinte resultado:

O gráfico da figura 8 apresenta o tempo de autenticação de cada uma das tentativas realizadas de um cliente.

O teste consistiu em criar a topologia no mininet, disponibilizando todos as entidades na rede, e em seguida inserindo o código de cliente num terminal do *xterm*. O

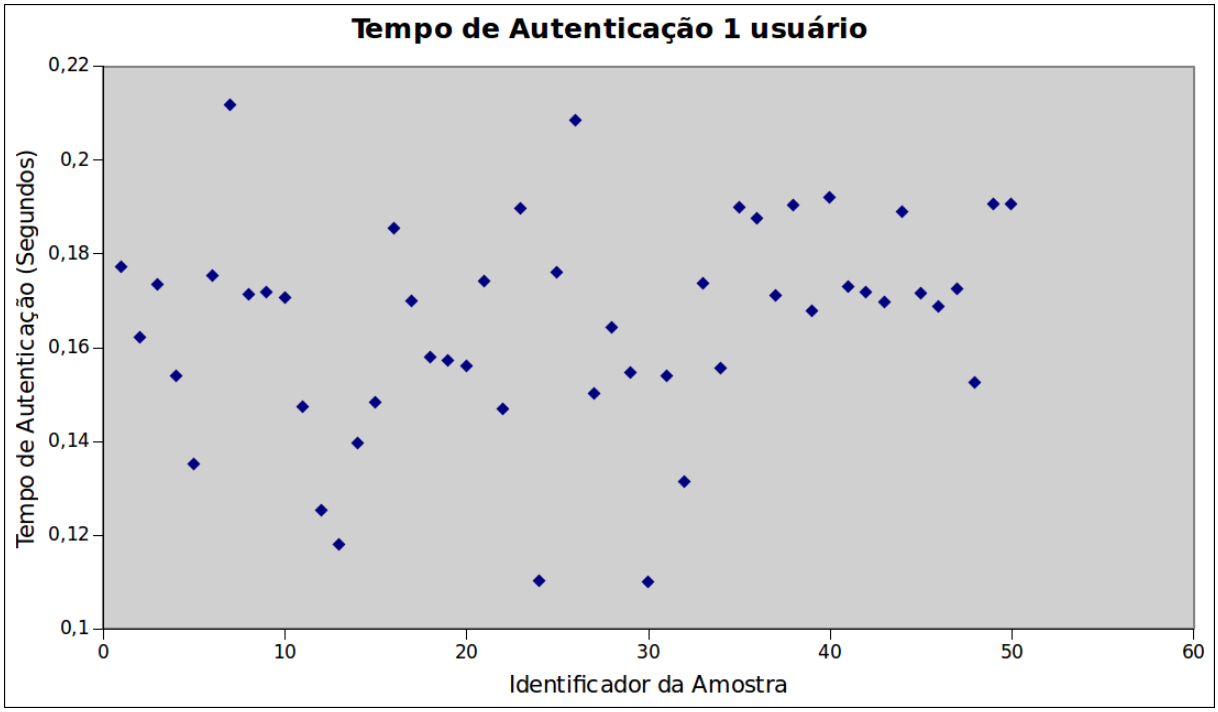


Figura 10: Tempo de autenticação das amostras de 1 usuário

cliente envia a credencial para o controlador receptor e inicia a contagem do tempo de autenticação. O controlador recebe realiza a verificação do alfa e beta recebidos com os que estão registrados, realiza a divisão de cada um de acordo com seu respectivo par (alfa com alfa e beta com beta) em seguida envia o resultados do alfa para os controladores controladores selecionados. Os controladores controladores realizam a computação parcial de devolvem para o controlador receptor, este computa os cálculos parciais com o beta recebido do cliente. Caso o resultado seja 1, o controlador envia uma flag positiva para o cliente. Assim que recebe esta flag a contagem de tempo de autenticação para e este dado é salvo. A média é de 0.165093565 e o desvio padrão 0.02233588797. O processo descrito foi realizado 50 vezes, gerando os resultados do gráfico da figura 8.

Cenários 2

No cenário dois foram utilizados 10 clientes tentando realizar autenticação assim como no cenário 1. A topologia é similar, a única diferença é o número de estações.

O gráfico da figura 9 apresenta a média de autenticação para cada uma das 10 estações. Cada ponto no gráfico representa a média em segundos (ordenada) do cliente (abscissa). Para as amostras, foi coletado a média $0,38785$ e desvio padrão $0,01779$. Nota-se que em relação ao cenário 1, houve um aumento em torno de

Também foi coletado o intervalo de confiança, nesta análise foi verificado que o tempo médio de autenticação das amostras 1 e 2 entre as 10 estão discrepantes em relação as outras.

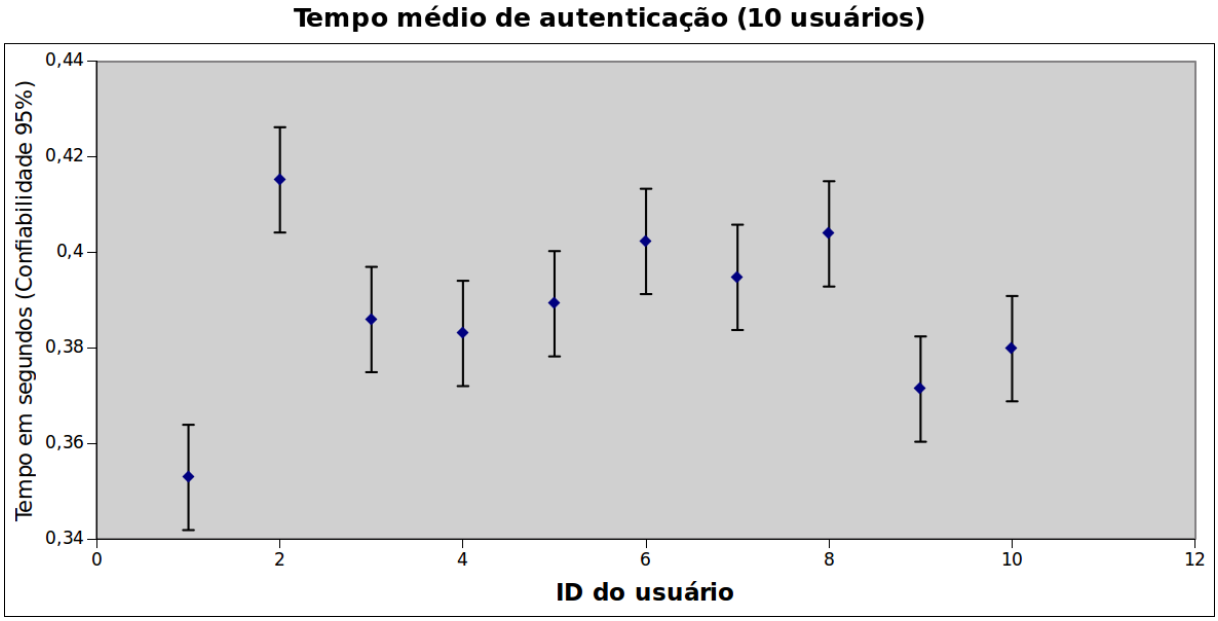


Figura 11: Tempo médio de autenticação de clientes no cenário com 10 estações *hosts*

Cenários 3

No cenário foi utilizado 30 estações tentando realizar a autenticação. O gráfico da figura 10 apresenta os resultados obtidos nos testes.

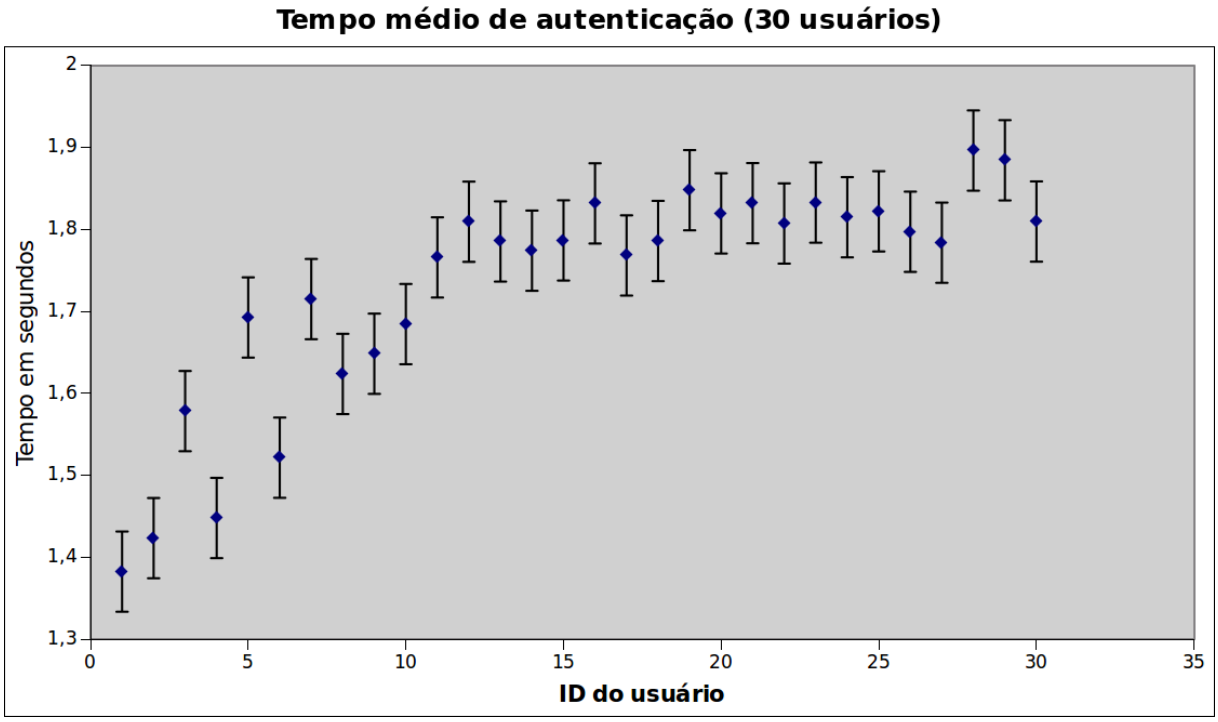


Figura 12: Tempo médio de autenticação de clientes no cenário com 30 estações

Nos testes do cenário 3, a média de autenticação foi de $1,73243$ segundos com desvio padrão de $0,13676$

Cenários 4

No cenário 4 foram utilizados 44 *hosts* e coletados os dados demonstrados no gráfico da figura 11.

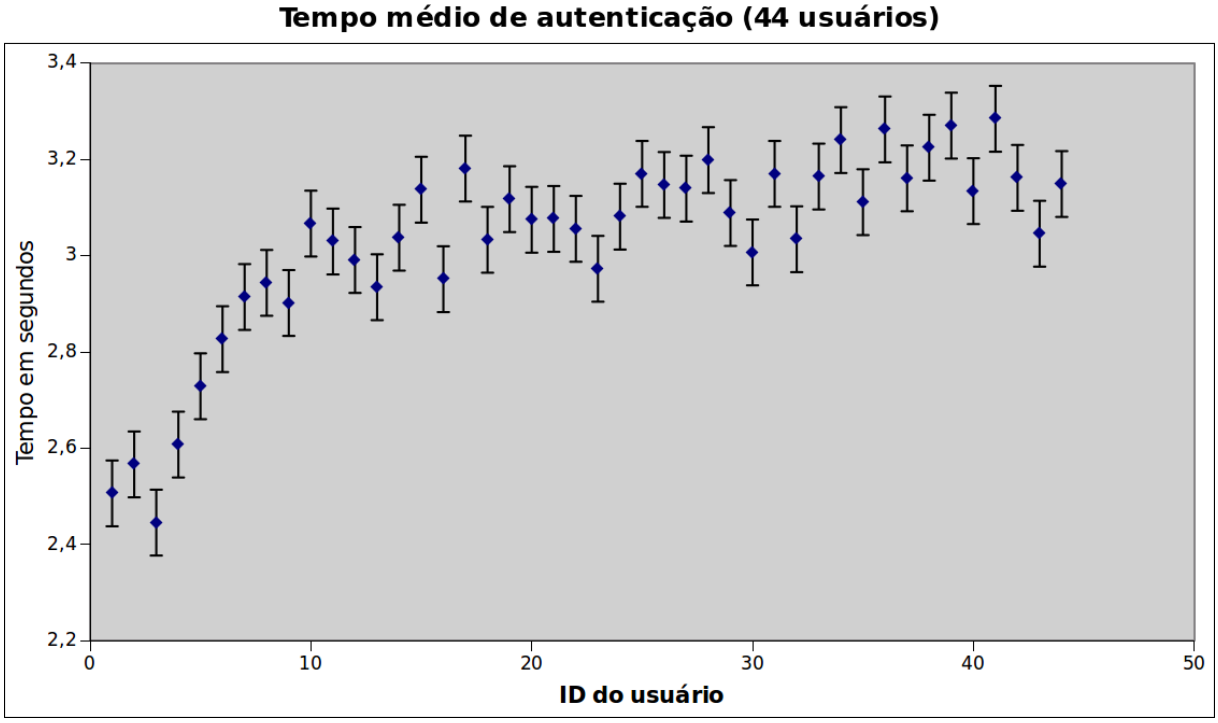


Figura 13: Tempo médio de autenticação de clientes no cenário com 44 estações

Neste cenário a média é $3,04551$ segundos e o desvio padrão $0,2314$ segundos.

O gráfico mostra que o tempo médio de autenticação vai aumentando para as amostras até a amostra 11 em que o tempo passa a apresentar um pouco mais de regularidade, isto é as médias estão mais próximas.

5.2 Análise dos resultados

O gráfico da figura 12, mostra a média de tempo de autenticação em cada um dos cenários. É possível perceber que com o aumento do número de clientes, o tempo de autenticação também aumenta, este aumento ocorre de maneira significativa o que pode resultar em esperas prolongadas caso num determinado momento uma grande quantidade de requisitantes realize a etapa de autenticação.

No primeiro cenário, o tempo de autenticação está dentro de um parâmetro razoável, mas a medida que são incrementados clientes esse tempo de autenticação também aumenta como esperado, devido ao maior número de requisições sendo tratadas.

No segundo cenário o número de requisições aumentou 10 vezes em relação ao primeiro e a média de tempo de autenticação é o dobro do cenário anterior.

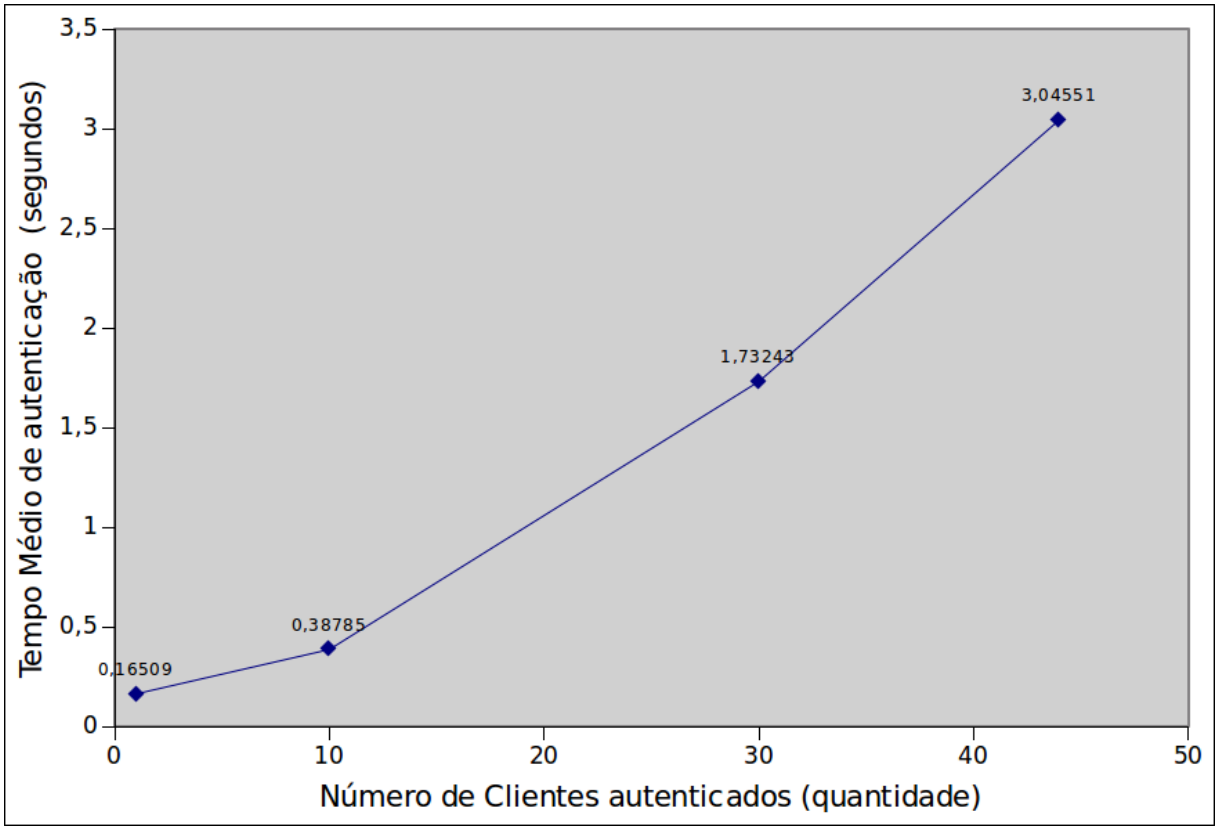


Figura 14: Tempo médio de autenticação de clientes no cenário com 44 estações

O terceiro cenário possui tempo médio de autenticação 10 vezes maior que o cenário inicial, mas com 30 vezes a quantidade de requisição deste.

O quarto cenário possui tempo média de autenticação 18 vezes maior que o primeiro. Essa média é para um aumento de 44 vezes o número de requisições do cenário 1.

O aumento significativo no tempo de autenticação de acordo com o acréscimo de requisições pode causar tempo de espera longo em cenários de redes com 1000, 10000 ou mais requisições por segundo.Evidentemente que casos onde ao mesmo instante são realizados essa grande quantidade de requisições de acesso não são tão comuns. O mais comum é uma rede que possui uma grande quantidade de requisitantes, mas que realizam acesso a rede em diferentes momentos, onde não há essa sobrecarga num instante t de tempo.

Um dos possíveis fatores para o aumento significativo do tempo de autenticação, além do número de clientes, é o fato que o controlador utilizado opera sobre eventos. Isto significa que não há concorrência de processos. O tratamento sequencial dos dados pode acarretar em aumento significativo do tempo de autenticação dos requisitantes em cenários de maior requisição. Na concorrência também há, mas como as requisições são tratadas simultaneamente, o tempo de processamento é menor que no tratamento de dados processo a processo.

A utilização do Pox se deu por conta de que algumas bibliotecas do relatório

compilado por da Costa et al. (2016) foram escritas em Python daí a necessidade de utilizar controlador escrito na mesma linguagem ou que disponibiliza API nesta linguagem de programação.

5.3 Conclusões do Capítulo

Neste capítulo foi exposto diferentes cenários de testes do mecanismo de autenticação descentralizada de usuários em redes definidas por software, bem como os resultados obtidos nos experimentos. O capítulo seguinte apresenta a avaliação da proposta e dos experimentos realizados.

CAPÍTULO 6

Conclusões e Trabalhos Futuros

6.1 Conclusões Gerais

A utilização de redes definidas por software vem crescendo ao longo dos anos, e diversos desafios surgiram. Um deles é o controle de acesso de usuários a uma rede deste novo paradigma. Inicialmente surgiram propostas de autenticação de usuários centralizada. Mas, a centralização do controle de acesso possui um único ponto de falha, e caso a entidade de controle de acesso seja comprometida, usuários indevidos podem ater acesso a rede.

A utilização de um controlador gerenciando a rede também mostrou-se problemática em caso de excessiva carga de trabalho na rede, então começou a ser proposto em diversos trabalhos a descentralização do plano de controle a fim de garantir escalabilidade e disponibilidade da rede, caso um dos controladores fique disponível ou sofra sobrecarga.

O trabalho apresenta um mecanismo de autenticação descentralizado com o intuito de prover confiabilidade e disponibilidade da rede. A descentralização da chave privada entre vários controladores garante mais segurança em relação a trabalhos que utilizam guarda centralizada da chave, além de não comprometimento da autenticação caso um dos controladores não seja confiável. O trabalho proposto contribui para a segurança de redes e também para a escalabilidade, visto que utiliza um cenário de múltiplos controladores e aproveita esta característica para fornecer uma alternativa de controle de acesso de usuários.

Os testes apresentaram aumento significativo no tempo de autenticação a medida que foi inserido mais usuários no processo de autenticação. Os resultados obitidos poderiam ser melhores utilizando controlador *multithread*.

Embora o resultado do tempo de autenticação não tenha sido satisfatório, o ob-

jetivo de apresentar uma proposta de autenticação distribuída foi alcançado. O trabalho apresenta um caminho inicial de mais segurança no controle de acesso em redes definidas por software. Comparado aos trabalhos apresentados no capítulo 3, o esquema proposto neste trabalho possui a vantagem de descentralizar a autenticação e a chave privada entre várias entidades, enquanto que alguma soluções utilizam o servidor Radius para manter as chaves e validar a autenticação em uma entidade. Caso este servidor não seja confiável a autenticação será comprometida.

6.2 Trabalhos Futuros

Utilizando este trabalho como base, futuramente é possível um esquema de autenticação aprimorado utilizando algum controlador que implemente multithreads como onos ou floodlight para melhorar o desempenho do tempo de autenticação. Também é possível implementar a confirmação de confiabilidade do cálculo final, isto é, a computaão parcial e o cálculo final realizado pelo controlador de passo 1 seria verificado entre os outros participantes do processo de autenticação. Após o resultado final o cliente recebe a permissão acesso de todos os participantes do processo de autenticação e não mais do controlador de passo 1 apenas. A abordagem de utilizar confirmação do cálculo final garantiria que o controlador de passo 1 não burle o processo de autenticação. O passo a passo seria:

1. Controlador de passo 1 recebe a mensagem de autenticação do cliente
2. realiza o processo de verificação da credencial e envia para os outros controladores realizarem a computação parcial
3. Cada controlador de passo 2 devolveria ao controlador de passo 1 a computação e aos outros controladores de passo 2 para atestar que o processo foi feito de maneira íntegra.
4. O controlador que realiza o cálculo final e envia para os outros controladores participantes da autenticação.
5. Caso todas as etapas de autenticação foram realizadas de maneira íntegra, o usuário recebe a permissão ou negação de todos os controladores que realizadam as etapas de autenticação deste usuário.

Referências Bibliográficas

Onos project, 2018. URL `https://onosproject.org/mission`.

Building sdn agile, 2018. URL `https://onosproject.org/mission`.

Syed Taha Ali, Vijay Sivaraman, Adam Radford, and Sanjay Jha. A survey of securing networks using software defined networking. *IEEE transactions on reliability*, 64(3): 1086–1097, 2015a.

Syed Taha Ali, Vijay Sivaraman, Adam Radford, and Sanjay Jha. A survey of securing networks using software defined networking. *IEEE transactions on reliability*, 64(3): 1086–1097, 2015b.

Rafael Aschoff, Daniel Rosendo, Marcos Machado, Alexandre Santos, and Djamel Sadok. A network access control solution combining orbac and sdn. In *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on*, pages 483–489. IEEE, 2017.

JJ BOTES and WT PENZHORN. Public-key cryptosystems based on elliptic curves. communications and signal processing, 1993. In *Proceedings of the 1993 IEEE South African Symposium*, 1993.

Leonardo da Costa, Airton Ishimori, and Antônio Abelém. Um protocolo seguro de autenticação em SDN baseado em criptografia de chave pública. Technical report, Universidade Federal do Pará, 08 2016.

Advait Dixit, Fang Hao, Sarit Mukherjee, TV Lakshman, and Ramana Kompella. Towards an elastic distributed sdn controller. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 7–12. ACM, 2013.

Roberto Samarone dos Santos Araujo. *On Remote and Voter-Verifiable Voting*. PhD thesis, Technische Universitat, 2008.

Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

Raj Jain and Subharthi Paul. Network virtualization and software defined networking for cloud computing: a survey. *IEEE Communications Magazine*, 51(11):24–31, 2013.

Murat Karakus and Arjan Durresi. A survey: Control plane scalability issues and approaches in software-defined networking (sdn). *Computer Networks*, 112:279–293, 2017.

Diego Kreutz, Fernando Ramos, and Paulo Verissimo. Towards secure and dependable software-defined networks. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pages 55–60. ACM, 2013.

Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76, 2015.

Jing Liu, Yingxu Lai, Zipeng Diao, and Yinong Chen. A trusted access method in software-defined network. *Simulation Modelling Practice and Theory*, 74:28–45, 2017.

Jon Matias, Jokin Garay, Alaitz Mendiola, Nerea Toledo, and Eduardo Jacob. Flownac: Flow-based network access control. In *Software Defined Networks (EWSDN), 2014 Third European Workshop on*, pages 79–84. IEEE, 2014.

Diogo Menezes Ferrazani Mattos and Otto Carlos Muniz Bandeira Duarte. Authflow: authentication and access control mechanism for software defined networking. *Annals of Telecommunications*, 71(11-12):607–615, 2016.

Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

Marcos Portnoi. Criptografia com curvas elipticas. Master's thesis, Universidade de Salvador, 2005.

Danda B Rawat and Swetha R Reddy. Software defined networking architecture, security and energy efficiency: A survey. *IEEE Communications Surveys & Tutorials*, 19(1): 325–346, 2017.

Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

Gustavus J Simmons. Symmetric and asymmetric encryption. *ACM Computing Surveys (CSUR)*, 11(4):305–330, 1979.

Kaur Sukhveer, Japinder Singh, and Navtej Singh Ghumman. Network programmability using pox controller.

Franciscus XA Wibowo, Mark A Gregory, Khandakar Ahmed, and Karina M Gomez. Multi-domain software defined networking: research status and challenges. *Journal of Network and Computer Applications*, 87:32–45, 2017.

Ziyao Zhang, Liang Ma, Kin K Leung, Franck Le, Sastry Kompella, and Leandros Tassiulas. How better is distributed sdn? an analytical approach. *arXiv preprint arXiv:1712.04161*, 2017.