UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Alexandre Silva Rodrigues

DIFMA: FIREWALL DISTRIBUÍDO COM ÊNFASE NA INTEROPERABILIDADE ENTRE APLICAÇÕES NO CONTEXTO DAS REDES ELÉTRICAS INTELIGENTES

Santa Maria, RS
2017

PPGEE/UFSM,RS RODRIGUES, Alexandre Silva Mestre 2017

Alexandre Silva Rodrigues

DIFMA: FIREWALL DISTRIBUÍDO COM ÊNFASE NA INTEROPERABILIDADE ENTRE APLICAÇÕES NO CONTEXTO DAS REDES ELÉTRICAS INTELIGENTES

Dissertação apresentada ao Curso de Programa de Pós-graduação em Engenharia Elétrica (PPGEE) , da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Engenharia Elétrica**.

Orientadora: Prof. Dra. (UFSM) Luciane Neves Canha

Santa Maria, RS

2017

Rodrigues, Alexandre Silva

DIFMA: firewall distribuído com ênfase na interoperabilidade entre aplicações no contexto das Redes Elétricas Inteligentes / por Alexandre Silva Rodrigues. – 2017.
81 f.: il.; 30 cm.

Orientadora: Luciane Neves Canha
Dissertação (Mestrado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-graduação em Engenharia Elétrica, RS, 2017.

1. Arquitetura DIFMA. 2. Firewall distribuído. 3. Redes Elétricas Inteligentes. 4. Segurança da informação. I. Canha, Luciane Neves. II.DIFMA: firewall distribuído com ênfase na interoperabilidade entre aplicações no contexto das Redes Elétricas Inteligentes.

E-mail: alexandrerodrigues@mail.ufsm.br

Alexandre Silva Rodrigues

DIFMA: FIREWALL DISTRIBUÍDO COM ÊNFASE NA INTEROPERABILIDADE ENTRE APLICAÇÕES NO CONTEXTO DAS REDES ELÉTRICAS INTELIGENTES

Dissertação apresentada ao Curso de Programa de Pós-graduação em Engenharia Elétrica (PPGEE) , da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Engenharia Elétrica**.

Aprovado em 16 de agosto de 2017:

Luciane Neves Canha, Dra. (UFSM)
(Presidente/Orientadora)

Paulo Ricardo da Silva Pereira, Dr. (UNISINOS)

Rafael Gressler Milbradt, Dr. (UFSM)

Santa Maria, RS

2017

DEDICATÓRIA

Primeiramente, quero dedicar essa conquista ao meu pai (in memoriam). Sua ausência física jamais irá apagar todas as lembranças que tenho de quando estavas aqui. A cada dia que passa, a saudade aumenta e com ela, a certeza que sempre estarás presente em minha memória. Quero dedicar a minha mãe, minhas irmãs e ao meu sobrinho. Essa vitória é de vocês também. Dedico, também, a todas as pessoas que acreditaram em mim e de alguma forma, contribuíram para que eu chegasse até aqui.

AGRADECIMENTOS

Primeiramente, quero agradecer a minha mãe e minhas duas irmãs por estarem ao meu lado, dedicando-me muito amor, apoio e carinho em todos os momentos. Só tenho a agradecer por ter uma família tão maravilhosa. Vocês representam muito em minha vida.

Agradeço ao minha orientadora Luciane Neves Canha, por todos os ensinamentos, pela disponibilidade que sempre tivestes para ajudar-me e pela oportunidade de fazer parte de projetos que contribuíram intensamente para minha formação acadêmica.

Agradeço ao colega de projeto Tiago Antonio Rizzetti pela disponibilidade e ensinamentos durante o andamento desse trabalho e demais desenvolvidos, desde a época em que eu cursava a graduação.

Agradeço aos amigos (bolsistas da sala 302 do CTISM) pela amizade e disponibilidade para ajudar sempre que precisei.

Agradeço também a todos amigos e familiares que estiveram ao meu lado.

Muito obrigado a todos.

“The eternal mistake of mankind is to set up an attainable ideal.”

(ALEISTER CROWLEY)

RESUMO

DIFMA: FIREWALL DISTRIBUÍDO COM ÊNFASE NA INTEROPERABILIDADE ENTRE APLICAÇÕES NO CONTEXTO DAS REDES ELÉTRICAS INTELIGENTES

AUTOR: ALEXANDRE SILVA RODRIGUES
ORIENTADORA: LUCIANE NEVES CANHA

O Sistema Elétrico de Potência (SEP), vem sendo aprimorado nos últimos anos, visando a implementação das Redes Elétricas Inteligentes (REI). Para isso, é essencial integrar tecnologias da informação e redes de comunicação bidirecionais aos equipamentos presentes no SEP. Entretanto, ao realizar esse tipo de avanço, expõe-se o SEP a novos tipos de ameaças cibernéticas e vulnerabilidades. Uma solução interessante para garantir a segurança em uma rede de comunicação de dados é impedir acessos não autorizados. Para isso, a utilização de um *firewall* é essencial. Através de regras eficientes em um *firewall* é possível controlar todo o tráfego de informação de uma rede ou dispositivo. Com a utilização de um *firewall* distribuído, cada dispositivo pode implementar suas próprias políticas de segurança, filtragem de pacotes e regras, não dependendo de uma filtragem centralizada. Entretanto, a heterogeneidade de equipamentos e dispositivos presentes na rede de comunicação de dados para REI apresenta um outro desafio: podem ser utilizados tipos de aplicações de *firewall* em um mesmo segmento da rede. Com isso, é necessário ter mecanismos para realizar a divulgação de regras e aplicação dessas em cada dispositivo. Nesses termos, o presente trabalho apresenta uma solução capaz de resolver essas prerrogativas: a Arquitetura DIFMA (*Distributed Firewall Multiple Applications*), a qual foi desenvolvida pelo autor desse trabalho. Essa arquitetura é composta por três módulos: DEMON (realiza o gerenciamento de dispositivos participantes da REI em grupos), RSIN (implementa uma rede de sobreposição para realizar a divulgação de regras que serão aplicadas nos dispositivos) e RIMA (realiza a interpretação de uma regra para uma determinada aplicação de *firewall* por meio da utilização de *plugins* específicos). Para avaliar a eficiência da Arquitetura DIFMA foram realizados testes de performance para sincronizar uma informação na rede de sobreposição e de criação de regras genéricas para aplicar nos dispositivos participantes de um determinado grupo. A interpretação dessa regras foi realizada por *plugins* desenvolvidos para as aplicações de *firewall* Iptables e UFW. Com base nos resultados obtidos durante esses testes, a Arquitetura DIFMA mostrou ser uma alternativa interessante e viável de ser implementada em um cenário real, visto que, os mecanismos de divulgação e interpretação de regras mostraram-se eficientes. Dessa forma, o risco de erros durante a geração da regra é reduzido, visto que, o operador não precisa preocupar-se com sintaxes específicas de cada aplicação de *firewall* que possa ser utilizada pelos dispositivos participantes de um grupo. Portanto, a Arquitetura DIFMA se destaca em relação a outras soluções encontradas na literatura por proporcionar uma solução integrada e escalável para a implementação de um *firewall* distribuído e possibilitando a interoperabilidade entre diferentes aplicações de *firewall*.

Palavras-chave: Arquitetura DIFMA. Firewall distribuído. Redes Elétricas Inteligentes. Segurança da informação.

ABSTRACT

DIFMA: DISTRIBUTED FIREWALL WITH EMPHASIS ON INTEROPERABILITY BETWEEN APPLICATIONS IN THE CONTEXT OF SMART GRIDS

AUTHOR: ALEXANDRE SILVA RODRIGUES
ADVISOR: LUCIANE NEVES CANHA

The Electric Power System (EPS) has been improved in recent years, aiming at the implementation of Smart Grids (SG). For this, it is essential to integrate information technologies and bidirectional communication networks at equipments present in the EPS. However, when making this type of advance, EPS is exposed to new types of cyber threats and vulnerabilities. An interesting solution to ensure security in the data communication network is prevent unauthorized access. For this, a use of a firewall is essential. Through efficient rules in a firewall it is possible to control all the information traffic of a network or device. With the use of a distributed firewall, each device can implement its security policies, packet filtering and rules, not depending a centralized filtering. However, the heterogeneity of equipment and devices present in the SG data communication network presents another challenge: diferents types of firewall applications can be used in the same segment of the network. With this, it is necessary to have mechanisms to perform the disclosure of rules and application of these in each device. In these terms, this work presents a solution capable of solving these prerogatives: the DIFMA Architecture (Distributed Firewall Multiple Applications), which was developed by the author of this work. This architecture is composed of three modules: DEMON (performs the management of SG participating devices in groups), RSIN (implements an overlay network to perform the disclosure of rules that will be applied to devices) e RIMA (perform the interpretation of a rule for a determined firewall application using specific plugins. To evaluate the efficiency of the DIFMA Architecture, performance tests were performed to synchronize information in the network of overlapping and creation of generic rules to apply to the participating devices of a given group. The interpretation of these rules was performed by plugins developed for the Iptables and UFW firewall applications. Based on the results obtained during these tests, DIFMA architecture proved to be an interesting and feasible alternative to be implemented in a real scenario, since the mechanisms of disclosure and interpretation of rules show to be efficient. In this way, the risk of errors during rule generation is reduced, since the operator does not have to worry about specific syntax of each firewall application that can be used by the devices participating in a group. Therefore, the DIFMA Architecture stands out in relation to other solutions found in the literature for providing an integrated and scalable solution for implementing a distributed firewall and enabling interoperability between different firewall applications.

Keywords: Architecture DIFMA. Distributed firewall. Smart Grids. Cybersecurity.

LISTA DE FIGURAS

Figura 2.1 – Fluxos de energia e de informações em uma REI 21
Figura 2.2 – Camadas de uma REI, em relação a sua área de abrangência. 23
Figura 2.3 – DHT circular. 25
Figura 2.4 – Criptografia com chaves pública. 29
Figura 2.5 – Criptografia com chave privada. 30
Figura 2.6 – Assinatura digital com SHA-1 e RSA. 31
Figura 2.7 – Processo de verificação de uma assinatura digital com SHA-1 e RSA. 31
Figura 2.8 – *Firewall* tradicional. 34
Figura 2.9 – *Firewall* distribuído. 35
Figura 4.1 – Visão geral da arquitetura DIFMA. 44
Figura 4.2 – Submódulos da aplicação DEMON *client*. 46
Figura 4.3 – Submódulos da aplicação DEMON *server*. 47
Figura 4.4 – Fluxograma de processos do módulo DEMON. 48
Figura 4.5 – Cenário de exemplo da utilização do módulo RIMA. 51
Figura 4.6 – Integração entre módulos da arquitetura DFMA. 52
Figura 5.1 – Tela da execução da aplicação DEMON client. 57
Figura 5.2 – Modelo físico do banco de dados da aplicação DEMON *server*. 58
Figura 5.3 – Página principal da interface web. 59
Figura 5.4 – Página que exibe as requisições ainda não verificadas. 60
Figura 5.5 – Tela da execução da aplicação RSIN *app*. 62
Figura 5.6 – Testes da comunicação em *multicast* através do uso da rede DHT. 65
Figura 5.7 – Tela da execução da aplicação RSIN *app* para gerar regra do cenário 1. 66
Figura 5.8 – Tela da execução da aplicação RSIN *app* para gerar regra do cenário 2. 67
Figura 5.9 – Tela da execução da aplicação RSIN *app* para gerar regra do cenário 3. 68
Figura 5.10 – Regras interpretadas pelo *plugin* desenvolvido para a Iptables. 68
Figura 5.11 – Regras interpretadas pelo *plugin* desenvolvido para UFW. 69
Figura 5.12 – Aviso de falha na verificação da assinatura de uma mensagem. 69
Figura A.1 – Página que exibe os grupos cadastrados na base de dados. 78
Figura A.2 – Página que possibilita criar um novo grupo. 79
Figura A.3 – Página que possibilita pesquisar os grupos em um dispositivo está cadastrado. 79
Figura A.4 – Página que exibe os grupos que um dispositivo participa. 80
Figura A.5 – Página que possibilita pesquisar os dispositivos cadastrados em um grupo. . . 80
Figura A.6 – Página que exibe os grupos que um dispositivo participa. 81
Figura A.7 – Página que exibe os certificados emitidos. 81

LISTA DE APÊNDICES

APÊNDICE A – FUNCIONALIDADES DISPONIBILIZADAS PELA INTERFACE WEB DA APLICAÇÃO DEMON SERVER 78

LISTA DE ABREVIATURAS E SIGLAS

AES	Advanced Encryption Standard
AMI	Advanced Metering Infrastructure
BT	Baixa tensão
CA	Certificate Autorithy
DEMON	Devices Manager On Network
DES	Data Encryption Standard
DHT	Distributed Hash Table
GPRS	General Packet Radio Service
DIFMA	Distributed Firewall Multiple Applications
HAN	Home Area Network
IGMP	Internet Group Management Protocol
IED	Intelligent Electronic Device
IP	Internet Protocol
LAN	Local Area Network
MLD	Multicast Listener Discovery
MT	Média tensão
PLC	Power Line Comunication
RAN	Regional Area Network
REI	Redes Elétricas Inteligentes
RIMA	Rules Interpreter Multiple Applications
RSIN	Rules Synchronizer In Network
SCADA	Supervisory Control and Data Acquisition
SHA	Secure Hash Algorithm
SEP	Sistema Elétrico de Potência
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
XML	eXtended Markup Language
WAN	Wide Area Network

SUMÁRIO

1 INTRODUÇÃO 13
1.1 JUSTIFICATIVA 13
1.2 OBJETIVOS 14
1.2.1 Objetivos específicos 14
1.3 PRINCIPAIS CONTRIBUIÇÕES 15
1.4 ESTRUTURAÇÃO DO TRABALHO 15
2 FUNDAMENTAÇÃO TEÓRICA 17
2.1 SEP ATUAL 17
2.2 REDES ELÉTRICAS INTELIGENTES 18
2.3 REDES DE COMUNICAÇÃO DE DADOS EM UMA REI 22
2.3.1 Redes de sobreposição 24
2.4 SEGURANÇA DAS INFORMAÇÕES 26
2.4.1 Criptografia 28
2.4.2 Assinaturas digitais 30
2.4.3 Certificados X.509 32
2.5 SEGURANÇA DA REDE DE COMUNICAÇÃO DE DADOS 32
2.5.1 *Firewall* 33
2.5.1.1 Localização de um firewall 34
2.5.1.2 Tipos de firewall 36
2.6 RESUMO DO CAPÍTULO 36
3 REVISÃO BIBLIOGRÁFICA 37
3.1 TRABALHOS RELACIONADOS A SEGURANÇA DAS INFORMAÇÕES EM REI 37
3.2 ORGANIZAÇÃO DE DISPOSITIVOS EM GRUPOS E COMUNICAÇÃO EM *MULTICAST* 39
3.3 FIREWALL DISTRIBUÍDO E INTERPRETAÇÃO DE REGRAS GENÉRICAS 40
3.4 RESUMO DO CAPÍTULO 41
4 METOLOGIA 43
4.1 ARQUITETURA DIFMA 43
4.2 DEVICES MANAGER ON NETWORK (DEMON) 45
4.3 RULES SYNCHRONIZER IN NETWORK (RSIN) 49
4.4 RULES INTERPRETER MULTIPLE APPLICATIONS (RIMA) 51
4.5 INTEGRAÇÃO ENTRE MÓDULOS DA ARQUITETURA DFMA 52
4.6 APLICAÇÃO PRÁTICA DA ARQUITETURA DIFMA 53
4.7 RESUMO DO CAPÍTULO 55
5 DESENVOLVIMENTO PRÁTICO 56
5.1 MÓDULO DEMON 56
5.2 MÓDULO RSIN 60
5.3 MÓDULO RIMA 63
5.4 TESTES E RESULTADOS 64
5.5 ANÁLISE DE RESULTADOS 70
6 CONSIDERAÇÕES FINAIS 71
REFERÊNCIAS 74
APÊNDICES 77

1 INTRODUÇÃO

A energia elétrica é utilizada para os mais diversos fins, seja nas residências quanto nas indústrias. Em situações onde o seu fornecimento é interrompido, evidencia-se o quanto ela é importante e necessita de sistemas capazes de automatizar o processo de restabelecimento da mesma. Além disso, a relação entre as concessionárias de energia elétrica e seus clientes ainda é restrita. Para resolver essas questões, diversas tecnologias têm surgido para facilitar o processo de distribuição de energia elétrica.

Nesse contexto, destaca-se o conceito de *Smart Grids* ou Redes Elétricas Inteligentes (REI), que apresentam uma série de vantagens em relação ao sistema convencional de energia elétrica, como por exemplo: interação entre dispositivos ativos no sistema elétrico de potência em tempo real, maior resiliência em casos de falhas no sistema, inserção de novas fontes de geração de energia (geração distribuída), automatização e melhor gerenciamento dos processos de geração, transmissão e distribuição da energia elétrica.

A implementação de uma REI requer uma rede de comunicação segura e bidirecional para interação entre os diversos dispositivos do sistema elétrico. Além disso, é importante ressaltar que essa comunicação necessita de um elevado índice de disponibilidade e confiabilidade, devido ao alto grau de criticidade das informações que nela podem trafegar. Nesses termos, a alteração ou falsificação de uma informação pode causar grandes prejuízos ao sistema elétrico ou interrupção de importantes serviços oferecidos por ele.

Dessa forma, a segurança das informações que trafegam em uma REI é essencial. Além disso, proteger a rede contra ações de agentes maliciosos e ataques externos é um requisito básico e um dos principais desafios para a implementação de uma REI.

1.1 JUSTIFICATIVA

A premissa básica para garantir a segurança das informações em uma rede de comunicação de dados é impedir a ação de atacantes externos nessa rede. Para isso, a utilização de um *firewall* na rede é essencial para analisar e controlar o tráfego de informações entre uma rede local e a Internet. Dessa forma, ele pode ser visto como uma barreira que realiza a filtragem dos pacotes que entram ou saem da rede e verifica se eles podem prosseguir ou serem descartados. Nesses termos, um *firewall* atua para bloquear um tráfego de informação suspeito ou malicioso

na rede de comunicação de uma REI, ou seja, um atacante pode utilizar a Internet para injetar informações na rede ou obter acesso remoto a um determinado dispositivo.

Em relação a localização do *firewall*, geralmente, ele é instalado no ponto de conexão entre a rede local e a Internet. Essa abordagem, em um ambiente de REI, pode apresentar alguns problemas, como por exemplo, a disponibilidade e escalabilidade do *firewall*. Dessa forma, uma alternativa interessante é utilizar um *firewall* distribuído. Ou seja, inserir um *firewall* em diversos segmentos da rede, por exemplo, em cada concentrador de informação ou em diversos dispositivos.

Entretanto, é necessário existir um mecanismo que possibilite a divulgação das regras a serem utilizadas. Além disso, por se tratar de uma rede de comunicação com grande heterogeneidade de equipamentos, diferentes tipos de *firewall* podem ser utilizados, o que dificulta esse processo, pois, cada tipo requer uma determinada sintaxe.

1.2 OBJETIVOS

O presente trabalho tem como objetivo prover uma arquitetura, denominada *Distributed Firewall Multiple Applications* (DIFMA), que possibilita descentralizar um *firewall*, aplicado a Redes Elétricas Inteligentes. Em outras palavras, nessa arquitetura, cada dispositivo pode atuar como um *firewall*, realizando a filtragem de todos pacotes que chegam ou que ele envia.

Para realizar a divulgação de regras utiliza-se uma rede de sobreposição, a qual facilita o processo de comunicação em *multicast*. As regras enviadas são interpretadas para o tipo de aplicação de *firewall* utilizada por cada dispositivo, ou seja, uma regra pode ser escrita de uma única forma, independentemente do tipo de *firewall* utilizado. Além disso, através dessa arquitetura é possível organizar os dispositivos em grupos, baseando-se em suas características de *hardware* e funcionalidades que exercem.

1.2.1 Objetivos específicos

- desenvolver uma ferramenta que permita a organização e gerenciamento de dispositivos baseados em suas características e funções que exercem;
- desenvolver uma ferramenta que possibilite que os dispositivos possam gerar requisições de certificados X509 e recebam seus certificados emitidos por uma *Certificate Autorithy* (CA);

- implementar uma rede de sobreposição capaz de realizar a troca de mensagens entre os dispositivos participantes de um grupo;
- implementar um mecanismo para interpretar regras e gerar a sintaxe correspondente ao *firewall* utilizado por um determinado dispositivo;
- realizar testes de verificação do tempo necessário para sincronizar informações entre os dispositivos participantes de um grupo de dispositivos;
- realizar testes de viabilidade da solução através da simulação de um cenário próximo a um ambiente de REI;
- escrita e publicação de resultados referentes a arquitetura proposta.

1.3 PRINCIPAIS CONTRIBUIÇÕES

A proposta deste trabalho visa apresentar a arquitetura de um *firewall* descentralizado, ou seja, diversos dispositivos participantes da rede de comunicação de dados de uma REI podem atuar como *firewall*, aplicando suas próprias regras para filtrar pacotes e assim, agregar maior segurança e escalabilidade para essa rede. Em virtude da grande heterogeneidade de equipamentos presentes em uma REI, os quais são capazes de enviar e receber dados, aplicar esse tipo de medida de segurança é essencial.

A principal contribuição deste trabalho em relação a outros existentes na literatura está relacionado a integração entre outros fatores essenciais para a implementação de um *firewall* distribuído, como por exemplo, o gerenciamento de dispositivos sob a forma de grupos, implementação de um mecanismo que possibilite a divulgação de regras para dispositivos que apresentam necessidades de regras de *firewall* semelhantes e interoperabilidade entre aplicações de *firewall*. Dessa forma, a arquitetura proposta é dividida em módulos, onde, cada módulo apresenta uma solução para esses desafios.

1.4 ESTRUTURAÇÃO DO TRABALHO

O presente trabalho está estruturado da seguinte forma:

- o capítulo 2 apresenta uma visão geral sobre o Sistema Elétrico de Potência atual e o conceito de Redes Elétricas Inteligentes, descrevendo seus benefícios e a necessidade de

integração dos equipamentos ativos nesse sistema com redes de comunicação de dados. Além disso,é apresentada uma visão geral sobre aspectos da segurança da informação e da rede de comunicação de dados no contexto das REI;

- o capítulo 3 apresenta alguns trabalhos que abordam a segurança das informações no contexto das REI, com ênfase na organização de dispositivos em grupos, formas de realizar comunicações em *multicast* e soluções de implementação de *firewalls* em uma rede de comunicação de dados;

- o capítulo 4 apresenta a proposta de uma arquitetura denominada *Distributed Firewall Multiple Applications* (DIFMA), descrevendo o seu funcionamento e suas principais aplicações;

- o capítulo 5 descreve os aspectos referentes a implementação da arquitetura DIFMA e os testes realizados para verificar sua eficiência e viabilidade;

- o capítulo 6 apresenta as considerações finais sobre a arquitetura DIFMA, descrevendo suas principais contribuições. Além disso, são abordadas melhorias futuras para essa arquitetura.

2 FUNDAMENTAÇÃO TEÓRICA

O presente capítulo apresenta uma visão geral sobre o Sistema Elétrico de Potência (SEP) e o conceito de Redes Elétricas Inteligentes (REI). Além disso, são abordados os conceitos de redes de comunicações de dados para REI e a segurança das informações trafegadas. Dessa forma, as seções seguintes estão divididas da seguinte forma: a seção 2.1 apresenta uma visão geral sobre o SEP atual, a seção 2.2 descreve o conceito de REI e seus benefícios, a seção 2.3 trata sobre redes de comunicação de dados para REI, descrevendo os principais meios físicos, divisão da rede em camadas e comunicação em *multicast*, a seção 2.4 aborda os principais requisitos e formas para garantir a segurança das informações trafegadas em uma rede de comunicação de dados, a seção 2.5 trata sobre segurança em redes de comunicação e a seção 2.6 apresenta um resumo do presente capítulo.

2.1 SEP ATUAL

O Sistema Elétrico de Potência (SEP) têm por função fornecer energia elétrica aos usuários (residenciais e industriais) com a qualidade adequada, no instante em que for solicitada (MCTI, 2014). Esse sistema pode ser visto como uma complexa infraestrutura, o qual deve atender padrões de confiabilidade e qualidade, modicidade tarifária e sustentabilidade social e ambiental. Esse sistema é composto, desde a sua concepção, em quatro partes: geração, rede de transmissão, rede de distribuição e usuários de energia elétrica.

A geração é centralizada em grandes usinas que, geralmente, estão localizadas longe dos maiores centros residenciais e industriais. Dessa forma, a transmissão de energia é realizada em longas distâncias, através da Rede Básica. Para que essa energia chegue até o consumidor final, são utilizadas as redes de distribuição em média tensão (MT) e baixa tensão (BT). Entretanto, nos últimos anos, a participação da geração distribuída vem crescendo consideravelmente (MCTI, 2014).

Em relação ao monitoramento, automação e controle desses sistemas, a geração e transmissão apresentam diversas iniciativas, tais como: chaves telecomandadas, medidores fasoriais e equipamentos de proteção e controle. Além disso, vale destacar a automação presente nas subestações de energia. Nesse cenário, os dispositivos enviam dados para as centrais de controle que processam essas informações e podem realizar ações remotamente. Para isso, utilizam-se

sistemas supervisórios, como por exemplo, o sistema SCADA (*Supervisory Control and Data Acquisition*), que é utilizado para supervisionar, controlar, otimizar e gerenciar os sistemas de geração e transmissão de energia elétrica (BAIG; AMOUDI, 2013). Dessa forma, consegue-se analisar o consumo, demanda e cargas dos consumidores. Além disso, é possível detectar falhas e realizar um rearranjo da topologia (LOPES et al., 2012).

No sistema de distribuição, nos últimos tempos, percebe-se algumas ações nesse sentido, como por exemplo, operações de controle e proteção. Entretanto, não existe um monitoramento em tempo real capaz de coletar informações de variáveis do sistema, como por exemplo, tensão fornecida ou correntes que circulam na rede (MCTI, 2014).

Dessa forma, o SEP vem sendo aprimorado com novas tecnologias, possibilitando diversas melhorias e novas funcionalidades. Nesse contexto, destaca-se o conceito de Redes Elétricas Inteligentes (*Smart Grids*), que apresentam uma série de vantagens em relação ao sistema convencional de energia elétrica (WANG; XU; KHANNA, 2011). A implementação desse conceito é o principal instrumento para modernizar o sistema elétrico, que há muitos anos não apresenta uma grande evolução (LOPES et al., 2012). Em relação a esse novo conceito, basicamente, ele pode ser visto como a interligação de um grande número de dispositivos e tecnologias que trará inúmeros benefícios a toda cadeia de provimento e consumo de energia elétrica (CGEE, 2012). Dessa forma, a seção seguinte apresentará uma visão mais detalhada sobre essa nova abordagem, seus benefícios e desafios de implementação.

2.2 REDES ELÉTRICAS INTELIGENTES

Segundo (CGEE, 2012), o conceito de REI pode ser visto como a integração de tecnologias digitais, recursos computacionais e redes de comunicação de dados ao SEP tradicional. Através dessa integração é possível realizar o monitoramento e gerenciamento da eletricidade ao longo da estrutura de tranmissão e distribuição até os consumidores finais. Esse conceito visa modernizar o SEP tradicional, proporcionando eficiência e melhor qualidade de fornecimento, diminuição dos custos e melhorias em aspectos ambientais (MCTI, 2014).

A implementação de uma REI trará diversos benefícios para os consumidores, concessionárias de energia e para o meio ambiente. Entre os principais benefícios para o consumidor final destacam-se (GHANSAH, 2009):

- medição inteligente: através da utilização de *Smart Meters* (medidores inteligentes) o

consumidor poderá ter informações sobre o seu consumo em tempo real. Essa informação pode ser disponibilizada em aplicativos para dispositivos móveis (*smartphone, tablet*) ou através de aplicações (*web* ou *software*) para computadores, apresentando informações detalhadas sobre o consumo, horários com maior consumo, entre outras.

- aplicação de tarifas diferenciadas: com a utilização de medidores inteligentes é possível aplicar tarifas diferentes com base nos períodos de consumo, ou seja, nos horários em que o consumo de energia é menor, pode-se aplicar uma tarifa reduzida em relação a horários de pico. Isso pode representar uma economia significante na fatura do consumidor;

- *self-healing*: uma possível falha no sistema elétrico que ocasiona uma interrupção do fornecimento de energia pode-se ser automaticamente detectada e corrigida em um intervalo de tempo menor, pois não necessita que o consumidor notifique a concessionária e talvez não seja necessário o deslocamento e a intervenção de uma equipe;

- venda de energia para a concessionária: o consumidor poderá gerar elétrica por meio da utilização de placas fotovoltaicas e vender para a concessionária, que poderá suprir a necessidade de energia em horários com maior demanda, sem ter que recorrer a fontes de geração que apresentam maiores custos para gerar energia. Com isso, a energia proveniente da geração distribuída dos consumidores, pode ter um valor menor para a concessionária e representar uma fonte de lucros para o consumidor que pode abater o valor da energia vendida em suas faturas.

Entre os benefícios para as concessionárias de energia elétrica pode-se citar (LAMIN, 2013):

- melhoria nos índices que quantificam a qualidade do serviço prestado: utilizando-se um sistema que ofereça mecanismos eficientes de medição e análise de dados é possível manter a qualidade dos serviços dentro das metas estimuladas pelos órgãos reguladores;

- redução das perdas não-técnicas e furtos de energia: através da medição eficiente das grandezas elétricas que circulam na rede, é possível identificar com maior facilidade os casos de perdas não-técnicas;

- redução de custos operacionais: com um sistema automatizado, capaz de detectar falhas e corrigi-las automaticamente, os custos de manutenções e deslocamento de equipes pode

ser reduzido. Além disso, oferecendo um serviço com melhor qualidade, possíveis multas por não cumprimento de metas ocorrerão com menor frequência;

- otimização da operação dos equipamentos ativos no sistema: com um sistema interligado e com possibilidade de operação remota, operações de manutenções podem ser melhor planejadas e executadas com maior rapidez. Além disso, o risco de danificação de equipamentos será menor.

- maior controle sobre a demanda de energia: a partir de uma base de dados mais detalhada sobre o consumo será possível realizar um melhor planejamento sobre a demanda de energia ao longo de diferentes intervalos de tempo;

- maior participação da geração distribuída, ou seja, os clientes poderão gerar energia e vender para as concessionárias onde a demanda for maior. Além disso, consumidores que geram energia podem a utilizar nesses horários e, dessa forma, reduzir a demanda nesses horários.

Em relação aos benefícios para o meio ambiente, deve-se destacar:

- a geração distribuída possibilitará o aumento do uso de fontes renováveis;

- redução da necessidade de construir novas usinas;

- redução da emissão de CO2.

Para que a implementação de uma REI possa ser concretizada, o primeiro passo é a comunicação entre todos os elementos existentes no SEP. Dessa forma, é necessária uma rede de comunicação bidirecional segura e eficiente que permita a troca de informação entre diversos tipos de equipamentos, enlaces, protocolos e tecnologias. Nesse cenário, uma REI é composta por diversos domínios lógicos, com agentes e dispositivos inteligentes que devem ser interligados. Nesses termos, pode-se definir cada domínio lógico da seguinte forma:

- geração: unidades geradoras de grandes parcelas de energia elétrica;

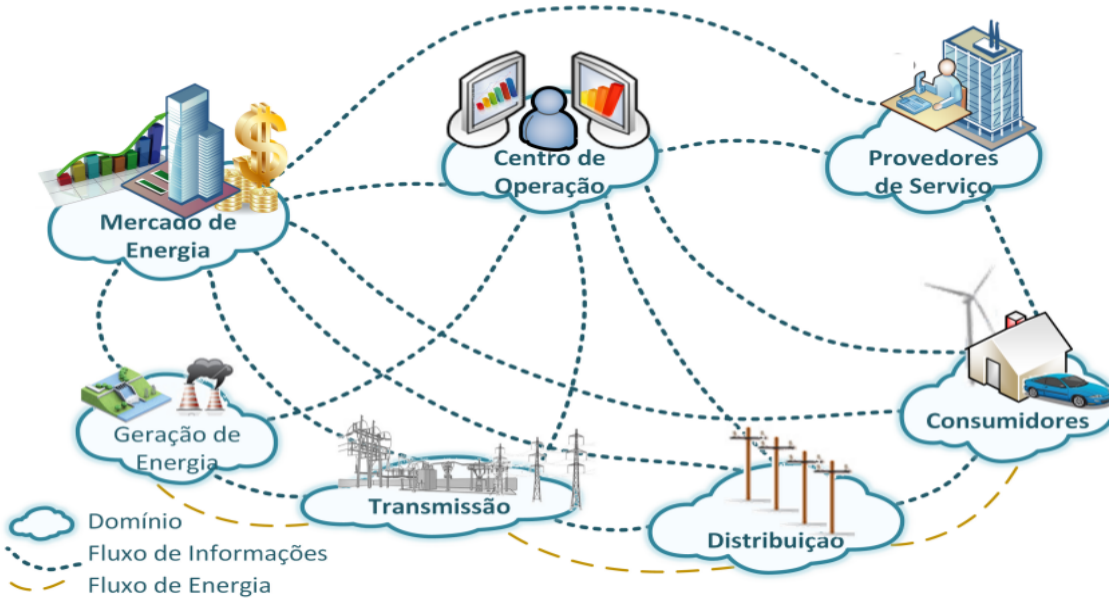
- transmissão: recursos de transmissão de eletricidade a longas distâncias;

- distribuição: relacionado aos distribuidores de eletricidade aos consumidores finais;

- consumidores: usuários finais da eletricidade que assumem papel de produtor e consumidor – prosumer ou prosumidor;
- mercado de energia: operadores e participantes do mercado de energia;
- operadores de rede: relaciona-se aos gerenciadores do fluxo de eletricidade.
- provedores de serviços: relaciona-se aos fornecedores de utilidades e serviços aos consumidores finais.

A Figura 2.1 ilustra esse cenário, apresentando o fluxo de energia e de informações, onde pode-se destacar a interligação lógica para a troca de informações entre os domínios citados anteriormente [LOPES, 2012].

Figura 2.1: Fluxos de energia e de informações em uma REI



Fonte: (LOPES et al., 2012).

Nesses termos, realizar a comunicação entre esses domínios, visto o grande números de variáveis envolvidas nesse processo, é necessária uma rede robusta e com diversas particularidades. Dessa forma, a seção seguinte apresenta uma visão geral sobre redes de comunicações de dados, com destaque para as topologias de rede e os principais meios físicos de transmissão de dados.

2.3 REDES DE COMUNICAÇÃO DE DADOS EM UMA REI

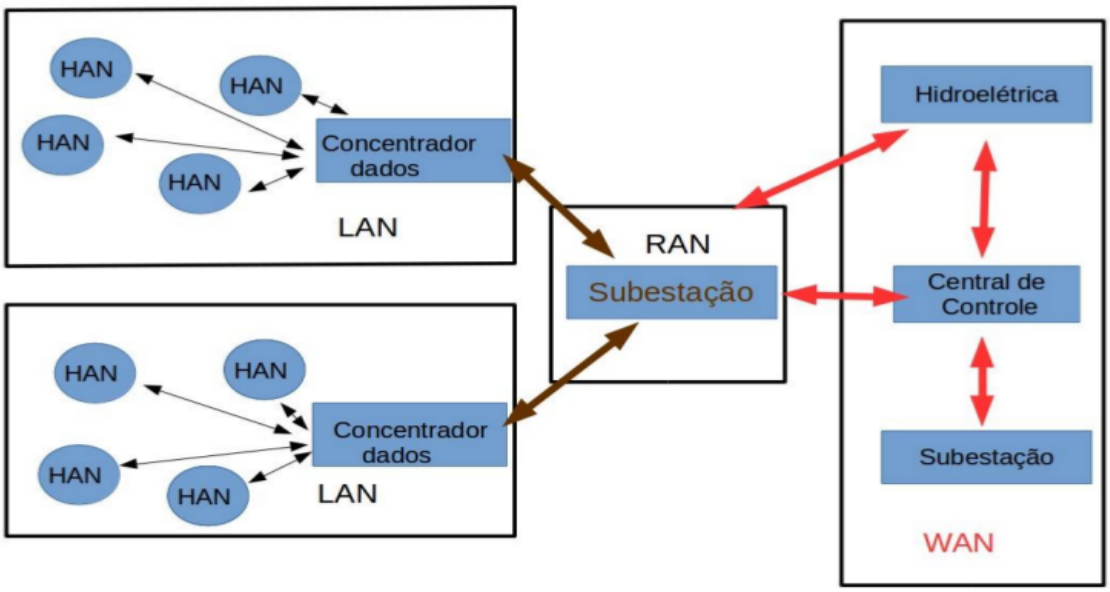
Conforme mencionado anteriormente, a implementação de uma REI requer uma rede de comunicação de dados robusta para que as informações possam ser trocadas de forma bidirecional entre equipamentos e demais atores (domínios lógicos) presentes nesse novo cenário. Para isso, uma alternativa interessante é dividir essa grande rede em partes menores, as sub-redes. Dessa forma, pode-se dividir em quatro camadas, em relação a sua abrangência: HAN (*Home Area Network*), LAN (*Local Area Network*), RAN (*Regional Area Network*) e WAN (*Wide Area Network*).

A HAN compreende os dispositivos presentes na residência do cliente. Nessa camada destaca-se a utilização de medidores inteligentes, os quais enviam informações referentes ao consumo de energia para a concessionária. Além disso, ele pode receber informações. Com isso, é possível que o cliente possa ter um controle sobre o seu consumo de energia elétrica. Um exemplo disso, consiste em desligar equipamentos em determinados horários em que a tarifa cobrada é mais cara. Nesse contexto, a concessionária poderá controlar a carga dentro de cada residência, limitar a demanda e evitar sobrecargas na rede. Nessa camada, o medidor inteligente será capaz de interligar a residência ao restante da REI. As informações enviadas por ele serão recebidas, geralmente, por um concentrador de dados (NARUCHITPARAMES; GÜNEŞ; EVRENOSOGLU, 2011).

Em relação a LAN e a RAN, essas camadas são responsáveis por coletarem informações de diferentes concentradores de dados. Basicamente, uma LAN abrange os concentradores de dados de uma pequena região, que pode ser um bairro. Enquanto isso, a RAN abrange uma área maior, ou seja, diversos concentradores de diferentes LAN's. Um exemplo disso, é uma rede dos clientes atendidos por uma subestação de distribuição (WANG; XU; KHANNA, 2011). Uma WAN recebe informações de dispositivos espalhados em uma grande área geográfica. Por exemplo, vários concentradores de dados e dispositivos presentes em diversas subestações podem enviar informações para uma central de controle (NARUCHITPARAMES; GÜNEŞ; EVRENOSOGLU, 2011).

A Figura 2.2 apresenta a organização da rede de uma REI com base nas camadas citadas anteriormente (EKANAYAKE et al., 2012).

Figura 2.2: Camadas de uma REI, em relação a sua área de abrangência.



Fonte: Acervo Pessoal.

Após o entendimento sobre camadas da rede de comunicação de dados de uma REI, em relação a sua abrangência, é interessante descrever as tecnologias que podem ser empregadas em cada camada para realizar a transmissão de dados. A tecnologia empregada vai refletir diretamente em aspectos técnicos, como por exemplo: latência (tempo necessário para transmitir uma informação) e disponibilidade da rede (KUROSE, 2013). Entre as principais tecnologias destacam-se (TANENBAUM, 2011) (EKANAYAKE et al., 2012):

- redes *mesh*: caracteriza-se por ser uma rede em malha, ou seja, todos os nós se comunicam, realizam o roteamento de pacotes e são auto-configuráveis.
- PLC (*Power Line Comunication*): tecnologia que utiliza os cabos condutores de eletricidade para trafegar dados, necessitando apenas de dispositivos de comunicação PLC e uma interface para os dispositivos a serem controlados;
- GPRS (*General Packet Radio Service*): utiliza os serviços de dados das operadoras de telefonia móvel;
- rádio frequência: as ondas de rádio são fáceis de gerar e podem percorrer longas distâncias. São amplamente utilizadas para comunicação, seja em ambientes fechados ou em locais abertos.

- fibra óptica: é usada para transmissão por longa distância nos *backbones* da rede e LANs de alta velocidade. Esse tipo de tecnologia apresenta altas velocidades de transmissões, confiabilidade e disponibilidade.

Nesses termos, para a comunicação entre equipamentos e seus concentradores, as tecnologias mais recomendadas são rádio frequência e rede *mesh*. A interligação entre as LANs e RANs pode ser realizada com rádio frequência ou fibra óptica. Para interligar as diversas RANs a WAN, a tecnologia mais indicada é a fibra óptica, pois, apresenta altas taxas de transferências e confiabilidade.

Com base nas topologias e meios de transmissão de dados, outro aspecto que deve ser analisado para implementar uma REI é a segurança das informações que irão ser trocadas, pois, elas são essenciais para que as novas funcionalidades possam apresentar os resultados esperados. Dessa forma, a seção 2.4 apresentará uma visão sobre os principais aspectos voltados a segurança das informações, os principais riscos e tecnologias que podem ser empregadas para garantir os requisitos necessários para esse tipo de aplicação.

Outro aspecto importante, quando trata-se de redes de comunicações de dados para REI, é a troca de informações entre dispositivos que participam de um determinado grupo (comunicação em *multicast*). Essa forma de comunicação pode ser utilizada quando deseja-se que diversos dispositivos recebam uma determinada informação. Para exemplificar essa abordagem, pode-se ilustrar um cenário onde diversos equipamentos atuam de forma similar, ou seja, enviam e recebem os mesmos tipos de informações. Nesse contexto, pode-se citar, como exemplo, os seguintes grupos de dispositivos: medidores inteligentes, medidores fasoriais, chaves telecomandas, entre outros. Dessa forma, a subseção seguinte apresenta uma solução para possibilita essa forma de comunicação

2.3.1 Redes de sobreposição

Para realizar o envio de informações em grupo, uma alternativa interessante é a utilização de redes de sobreposição, como por exemplo, uma rede DHT (*Distributed Hash Table*). Essa abordagem é descrita por (STOICA et al., 2001), que estabelece o seguinte conceito: uma rede DHT se baseia na utilização de tabelas contendo estruturas similares ao roteamento, as quais possuem ponteiros para uma estrutura de nós ativos à sua frente. Essa estrutura é utilizada para localização e indexação de informações.

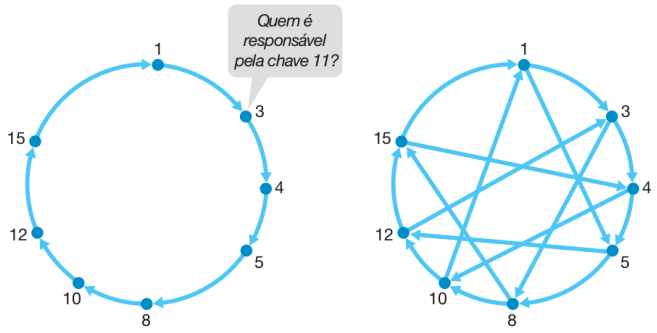
As redes DHT são amplamente utilizadas para implementar redes de compartilhamento

de arquivos. A principal característica desse tipo de rede é a sua descentralização. Isso permite que elas possam ser utilizadas em diferentes tipos de aplicações, como por exemplo: armazenamento de arquivos e formação de redes específicas para *multicast*, *anycast*, entre outros (URDANETA; PIERRE; STEEN, 2011) (STOICA et al., 2001).

Em função de sua estrutura distribuída, uma rede DHT possui características de elevada disponibilidade e tolerância a falhas, uma vez que suporta a indisponibilidade de diversos nós da rede sem apresentar comprometimento desta. A base de funcionamento de uma rede DHT, pode ser vista como um sistema composto por vários nós que implementam uma operação de pesquisa (k). Essa pesquisa retorna os dados associados a uma chave k. Os dados contêm o endereço de rede do nó responsável pela chave k. Alternativamente, uma DHT pode implementar a Operação Rota (k), que simplesmente encaminha uma mensagem para o nó responsável pela chave k (URDANETA; PIERRE; STEEN, 2011).

O princípio de funcionamento de uma DHT é a existência de um espaço identificador comum para os nós e as chaves, com cada chave k armazenada no nó com o mais próximo de identificador k de acordo com uma função de distância (URDANETA; PIERRE; STEEN, 2011). Para localizar o nó responsável por uma chave k, um nó encaminha o pedido de pesquisa para outro par, cujo identificador está mais perto de k (de acordo com a função de distância). Todos os nós devem manter *links* para um subconjunto de outros nós, formando assim uma rede de sobreposição. A solicitação de pesquisa é encaminhada por nós até que nenhum nó é encontrado, com um identificador mais perto da chave solicitada (URDANETA; PIERRE; STEEN, 2011). Para exemplificar essa busca, a Figura 2.3 apresenta uma DHT circular, composta por 15 nós (KUROSE, 2013).

Figura 2.3: DHT circular.



Fonte: (KUROSE, 2013)

Conforme pode ser visualizado na Figura 2.3, o nó 3 deseja determinar qual o par no DHT responsável pela chave 11 para inserir ou requisitar uma informação. Utilizando a rede de sobreposição, o par 3 pergunta para seu vizinho mais próximo (par 4) “quem é o responsável pela chave 11”. Quando um par receber essa mensagem, como ele conhece o identificador de seu sucessor, ele pode determinar se é responsável pela chave em questão. Caso ele não seja o responsável, encaminha a mensagem para o seu sucessor. Esse processo continua até que a mensagem chegue ao par 12 (mais próximo de 11). Assim, o par 12 envia uma mensagem para o par 3, informando que é responsável pela chave 11 (KUROSE, 2013).

2.4 SEGURANÇA DAS INFORMAÇÕES

Para tornar possível a implementação de uma REI é essencial que os equipamentos presentes no SEP estejam conectados e troquem informações. Em virtude dessa premissa básica, a segurança da rede de comunicação de dados e das informações que nela trafegam representam um grande desafio, pois, qualquer vulnerabilidade existente na rede de comunicação de dados pode ser utilizada por *hackers* ou usuários mal-intencionados para a realização de ataques ou interceptação e alteração de dados.

Tais ações podem ser realizadas propositalmente para obter benefícios, chamar atenção, prejudicar alguém ou apenas por curiosidade. Nesses termos, garantir a segurança das informações é imprescindível (TANENBAUM, 2011).

Quando se trata de segurança de informações trocadas em uma rede de comunicação, alguns requisitos mínimos devem ser levados em consideração, como por exemplo: confidencialidade, integridade e disponibilidade, os quais formam a conhecida e consolidada tríade denominada CID (STALLINGS, 2014). Entretanto, ressalta-se que outros requisitos devem ser observados, como por exemplo, a autenticidade e responsabilização (não-repúdio) (STALLINGS, 2015).

O conceito de confidencialidade pode ser visto como a privacidade de uma informação, ou seja, apenas o remetente e destinatário podem conhecer o conteúdo de uma informação transmitida. Caso algum intruso obter acesso a essa informação durante a sua transmissão, ela não pode ser entendida (KUROSE, 2013). Um exemplo disso, em uma REI, é o envio de informações referentes a medição do consumo de energia de um consumidor, a qual deve ter sua privacidade mantida. Se um intruso obter acesso a essas informações, que poderão ser enviadas simultaneamente, ele pode ter acesso sobre o consumo do cliente e assim, identificar aspectos

pessoais sobre ele, como por exemplo: horários que costuma sair e chegar em sua residência (GHANSAH, 2009).

O requisito de integridade está relacionado ao conteúdo da informação, ou seja, evitar que ela seja modificada ou destruída indevidamente (KUROSE, 2013). Como exemplo para esse requisito, pode-se citar novamente as informações de medições de um cliente. Se um intruso interceptar essa informação, ele pode modificar os dados lidos pelo medidor inteligente e gerar um prejuízo financeiro ao consumidor. Outro exemplo é os dados enviados para atuadores presentes na rede elétrica. Nesse caso, um atacante pode alterar a ação que deve ser realizada, ocasionado erros de operação dos equipamentos (BAIG; AMOUDI, 2013) (GHANSAH, 2009).

Quanto a disponibilidade, ela pode ser relacionada com o acesso e utilização rápida e confiável de uma informação ou sistema (STALLINGS, 2015). Para ilustrar a importância desse requisito pode-se citar o sistema de aquisição de dados em uma central de controle. Caso essa aplicação fique indisponível, possíveis anomalias na rede elétrica podem não serem detectadas e assim, ocasionar grandes danos a equipamentos ou um *blackout* em todo o sistema elétrico (BAIG; AMOUDI, 2013).

Sendo um dos requisitos adicionais para garantir a segurança de uma informação, a autenticidade é propriedade de um emissor ser genuíno, confiável e que possa ser verificado. Em outras palavras, ele deve ser quem diz que é (STALLINGS, 2014). Esse requisito pode ser utilizado como uma forma de garantir que as informações transmitidas por um determinado equipamento foi enviada por ele mesmo, ou seja, não foi interceptada, modificada e reenviada para o destino. No cenário descrito anteriormente, em que um intruso modificava informações do consumo de um cliente, após a detectar a falsa de autenticidade do transmissor (no caso um atacante que reenviou a mensagem), essa informação deveria ser descartada.

Outro requisito é a responsabilização que pode ser visto como atribuição das ações de uma entidade sejam a ela, assim, garantindo a irretratabilidade de uma determinada ação, por exemplo, envio de uma determinada informação (STALLINGS, 2014). Um exemplo disso é responsabilizar um atacante que realizou alterações de informações, conforme descrito em cenários exemplificados anteriormente, ou seja, identificar a origem do ataque, que pode ser através de seu endereço IP (*Internet Protocol*).

Para garantir esses requisitos existem diversas técnicas, como por exemplo, criptografia, assinaturas digitais e certificados, que serão descritas nas subseções seguintes.

2.4.1 Criptografia

O conceito de criptografia pode ser visto como tornar os dados contidos em uma mensagem ininteligíveis a um intruso, ou seja, o remetente disfarça os dados de modo que um intruso não consiga obter nenhuma informação dos dados interceptados. Dessa forma, apenas o destinatário poderá interpretá-los (KUROSE, 2013). Para isso, o emissor deve cifrar a mensagem que deseja enviar (conhecida como texto plano ou texto claro), utilizando uma chave criptográfica, e enviar para o destinatário. Quando esse receber a mensagem (texto cifrado), ele deve realizar o processo inverso, ou seja, decifrar o texto criptografado para obter a informação original. No caso de um terceiro interceptar o texto criptografado e não conhecer a chave necessária para descriptografar a mensagem, ele não conseguirá obter a informação correta. Para criptografar uma determinada informação utiliza-se algoritmos criptográficos de chaves simétricas ou assimétricas (TANENBAUM, 2011).

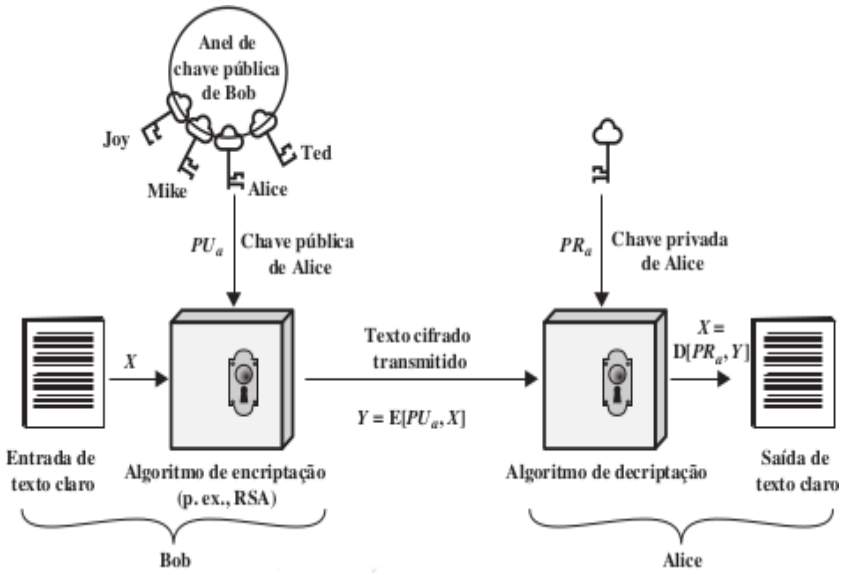
No caso de algoritmos de chave simétrica, utiliza-se a mesma chave para criptografar e descriptografar uma mensagem. O funcionamento desses algoritmos está relacionado a uma cifra de bloco, que recebe um texto plano de tamanho fixo e retorna um bloco de texto cifrado de tamanho igual. Entre os principais algoritmos desse tipo destacam-se: *Data Encryption Standard* (DES), o *Triple* DES (DES triplo) e *Advanced Encryption Standard* (AES) (STALLINGS, 2015). O mais conhecido e utilizado atualmente é o algoritmo AES, que utiliza blocos de 128 bits e pode operar com chaves com tamanho de 128, 192 e 256 bits (KUROSE, 2013).

Apesar de sua eficiência, algoritmos de chave simétrica apresentam problemas de gerenciamento da chave utilizada, pois, caso um atacante descobrir a chave utilizada para criptografar as informações, ele obterá acesso a todos os dados que forem criptografados com ela. Isso se torna mais grave quando se utiliza esse algoritmo em aplicações que envolvem diversos dispositivos, pois tornaria toda a aplicação insegura. Para esses casos é necessário um mecanismo de gerenciamento de chaves eficiente, capaz de atualizar a mesma em todos os dispositivos que a utilizam (TANENBAUM, 2011).

Outra opção é utilizar algoritmos de chave assimétrica ou criptografia de chave pública. Esses utilizam chaves diferentes para criptografar e descriptografar uma mensagem. Ou seja, a base de funcionamento desses algoritmos é a utilização de um par de chaves (pública e privada).O algoritmo de chave assimétrica mais conhecido e utilizado chama-se RSA, e utilizam chaves maiores ou iguais a 1024 bits (STALLINGS, 2014).

Para ilustrar a forma como se realiza a criptografia e descriptografia utilizando chaves assimétricas, a Figura 2.4 apresenta um cenário em que o emissor (Bob) pretende enviar uma mensagem criptografada, utilizando a chave pública do destinatário (Alice) (STALLINGS, 2015).

Figura 2.4: Criptografia com chaves pública.

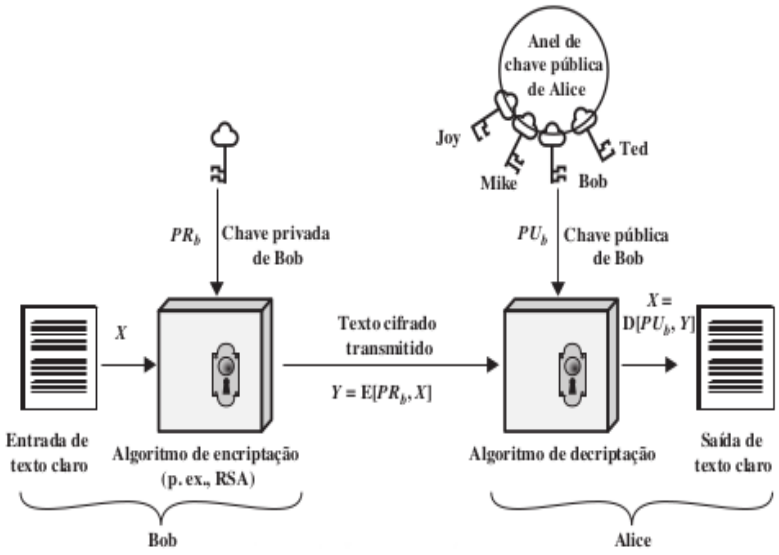


Fonte: (STALLINGS, 2015)

Conforme pode ser observado na Figura 2.4, primeiramente, utiliza-se a chave pública de Alice para criptografar e depois, a chave privada de Alice. Dessa forma, se uma mensagem for criptografada com a chave pública do destinatário, ela será descriptografada somente com a chave privada do mesmo. Nesse caso, se o emissor quiser enviar para diversos, ele deve conhecer a chave pública de todos eles. Quando se aplica essa forma de criptografia garante-se a confidencialidade da informação, pois, somente o destinatário poderá obter os dados que foram enviados.

A outra forma apresentada é criptografar com a chave privada do emissor. Nesse caso, a mensagem deve ser descriptografada com a chave pública do mesmo. Dessa forma, é garantida a autenticidade do emissor. O processo para realizar esse tipo de criptografia pode ser visualizado na Figura 2.5. Nesses termos, esse método é utilizado para gerar assinaturas digitais, que será detalhado na subseção seguinte.

Figura 2.5: Criptografia com chave privada.



Fonte: (STALLINGS, 2015)

2.4.2 Assinaturas digitais

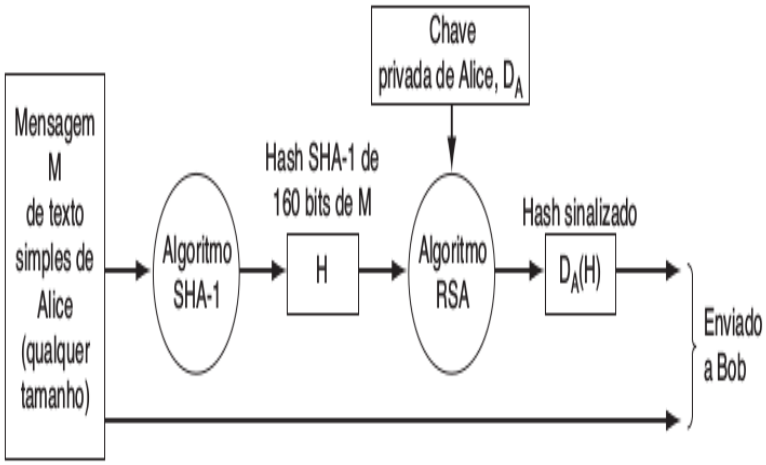
Quando se pretende garantir a autenticidade de uma mensagem, é aplicado o conceito de assinatura digital que consiste em gerar uma *hash* do conteúdo que se deseja enviar,através de uma função de *hash*, que aceita uma mensagem de tamanho variável M como entrada e produz um valor de *hash* de tamanho fixo h = H(M). A principal propriedade dessa função é que uma mudança em qualquer bit ou bits em M resulta em uma mudança no código de *hash*. Entre as principais funções de *hash* pode-se citar a *Secure Hash Algorithm* (SHA-1) (STALLINGS, 2015). Após gerar a *hash*, essa é criptografada com a chave privada do emissor. Dessa forma, a assinatura digital está construída e pode ser enviada junto a mensagem cifrada para o destinatário.

Quando o destinatário receber a informação assinada, ele pode verificar a autenticidade e integridade da informação recebida. Para isso, ele deve descriptografar a assinatura digital recebida com a chave pública do emissor. Posteriormente, deve ser gerada uma nova *hash* do conteúdo recebido (após realizar a descriptografia para obter a mensagem original). Após, deve-se realizar a comparação das duas *hash*. Se elas forem iguais, existe a garantia que informação não teve sua integridade afetada e teve sua autenticidade comprovada (KUROSE, 2013)

(EKANAYAKE et al., 2012).

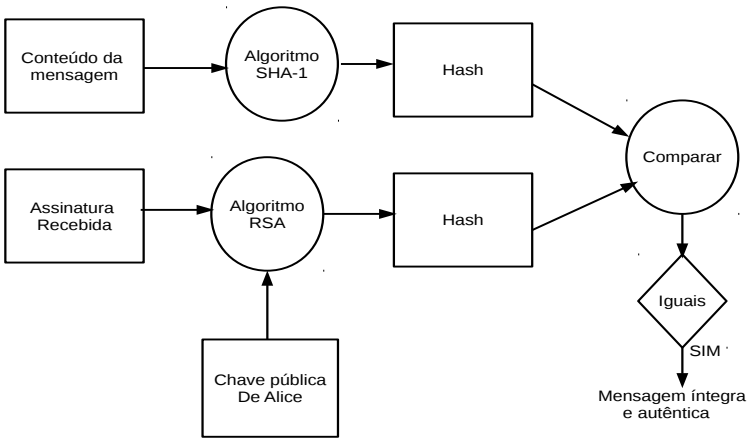
Nesses termos, a Figura 2.6 ilustra o processo de assintatura de uma mensagem e a Figura 2.7 ilustra o processo necessário para verificar uma assinatura.

Figura 2.6: Assinatura digital com SHA-1 e RSA.



Fonte: (TANENBAUM, 2011)

Figura 2.7: Processo de verificação de uma assinatura digital com SHA-1 e RSA.



Fonte: Adaptado de (KUROSE, 2013)

2.4.3 Certificados X.509

Para que a criptografia com chave assimétrica possa ser utilizada em uma troca de mensagens, ambos participantes devem conhecer as chaves públicas dos demais. Levando em consideração a forma de realizar a troca de chaves públicas entre eles, pode-se utilizar o conceito de certificados, que pode ser visto como uma chave pública mais um identificador do proprietário da chave. Essas informações devem ser assinadas por uma *Certificate Authority* (CA), a qual deve confiável por ambos (STALLINGS, 2015).

Nesses termos, quando se pretende obter um certificado, o usuário deve gerar uma requisição contendo suas credencias e sua chave pública. Assim a CA pode emitir um certificado para ele após a verificação das informações apresentadas. Após a geração de um certificado, ele pode ser publicado para os demais usuários. Caso outro usuário precise da chave pública do emissor de uma mensagem, ele pode solicitar o certificado. Além disso, com essa abordagem, consegue-se garantir que o usuário que enviou a mensagem é autorizado a usar o certificado por uma CA confiável (STALLINGS, 2014).

O padrão X.509 é um formato de certificados utilizado universalmente e define uma estrutura para a provisão de serviços de autenticação pelo diretório X.500 aos seus usuários. Cada certificado contém a chave pública de um usuário e é assinado com a chave privada de uma CA confiável (STALLINGS, 2014). O formato de um certificado X.509 possui diversos elementos, como por exemplo: versão do X.509, número serial do certificado, algoritmo de *hash* utilizado para assinar o certificado, nome do emissor, data de validade, dados do dono do certificado (nome, chave pública, identificador) e assinatura do certificado (assinado pela chave privada da CA) (TANENBAUM, 2011).

2.5 SEGURANÇA DA REDE DE COMUNICAÇÃO DE DADOS

Além de garantir a segurança das informações trocadas entre os equipamentos e demais participantes de uma REI, garantir a segurança da rede de comunicação é extremamente importante. Por se tratar de uma rede com grande heterogeneidade, isto é, diversos tipos de equipamentos, fabricantes, protocolos e diferentes tecnologias, o número vulnerabilidades que podem ser exploradas cresce consideravelmente.

Esse cenário se agrava mais ainda em decorrência do acesso de operadores e provedores de serviço aos equipamentos instalados nas residências dos clientes ou na rede de distribuição.

Esses acessos devem acontecer de forma remota e muitas vezes, através de sub-redes diferentes ou através da Internet. Qualquer vulnerabilidade existente no meio de comunicação, aplicações, protocolos ou tecnologias empregadas entre um destes agentes e o sistema de energia poderá acarretar consequências ao sistema de energia.

Quanto maior as vulnerabilidades presentes, assim como as consequências destas, maior será o risco de sua utilização. Um ataque num sistema desse porte pode ocasionar grandes prejuízos, como por exemplo: deixar cidades sem energia elétrica, danificar equipamentos, ocasionar acidentes fatais na operação e manutenção do sistema elétrico, entre outros.

Para impedir o acesso de agentes não autorizados a rede de comunicação de uma REI, uma alternativa interessante é a utilização de um *firewall*. Dessa forma, consegue-se filtrar os pacotes que entram e saem da rede de comunicação. Assim, ações de atacantes externos a essa rede podem ser impedidas.

2.5.1 *Firewall*

A utilização de um *firewall*, geralmente, é essencial para qualquer organização que interliga uma rede local a Internet, visando a prevenção de danos a sua rede, através da implantação de uma determinada política de segurança (STEPANEK, 2001).

Um *firewall* pode ser visto como um sistema, composto por um conjunto de hardware e software, o qual é capaz de atuar como defesa entre uma rede local e a Internet. A base de seu funcionamento está relacionada ao controle de tráfego entre as sub-redes de uma rede privada, ou seja, todo o tráfego de entrada e saída é controlado por ele, que pode autorizar, negar e registrar tudo o passa por ele (MORAES, 2011). Assim, um *firewall* permite que o administrador de rede controle o acesso entre o mundo externo e os recursos da rede que administra, gerenciando todo o fluxo de tráfego (KUROSE, 2013).

Dessa forma, um *firewall* apresenta os seguintes objetivos (KUROSE, 2013) (STALLINGS, 2015):

- todo o trafego que entra ou sai de uma rede passa pelo *firewall*: isso é obtido bloqueando fisicamente todo o acesso a rede local, exceto por meio do *firewall*. Dessa forma, o *firewall* realiza a filtragem de cada pacote e em seguida, determina a ação a ser tomada: deixar o pacote seguir ou descartar;

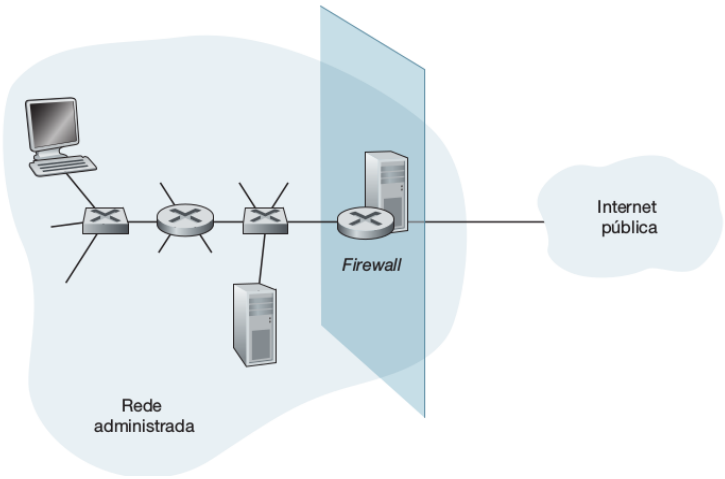
- somente o trafego autorizado, como definido pela política de segurança, poderá passar;

- o próprio *firewall* é imune a intrusão: se ele não for projetado ou instalado corretamente, pode comprometer a segurança de toda a rede.

2.5.1.1 Localização de um firewall

Segundo (KHOSROSHAHI; SHAHINZADEH, 2016), um *firewall* é inserido no ponto de conexão entre uma rede local (LAN) e a Internet para estabelecer um enlace controlado e definir um perímetro de segurança externo, conforme pode ser visualizado na Figura 2.8. Essa abordagem é amplamente utilizada em redes corporativas.

Figura 2.8: *Firewall* tradicional.



Fonte: (KUROSE, 2013)

Entretanto, em uma rede de comunicação para REI, utilizar um *firewall* centralizado apresenta alguns problemas em relação a disponibilidade e escalabilidade. Por tratar-se de uma rede que requer elevada disponibilidade, um *firewall* centralizado poderia ser o principal alvo de um atacante. Dessa forma, após realizar um ataque de negação de serviço no *firewall*, a segurança de toda a rede estaria comprometida ou tornaria indisponíveis alguns serviços oferecidos por ela (KHOSROSHAHI; SHAHINZADEH, 2016).

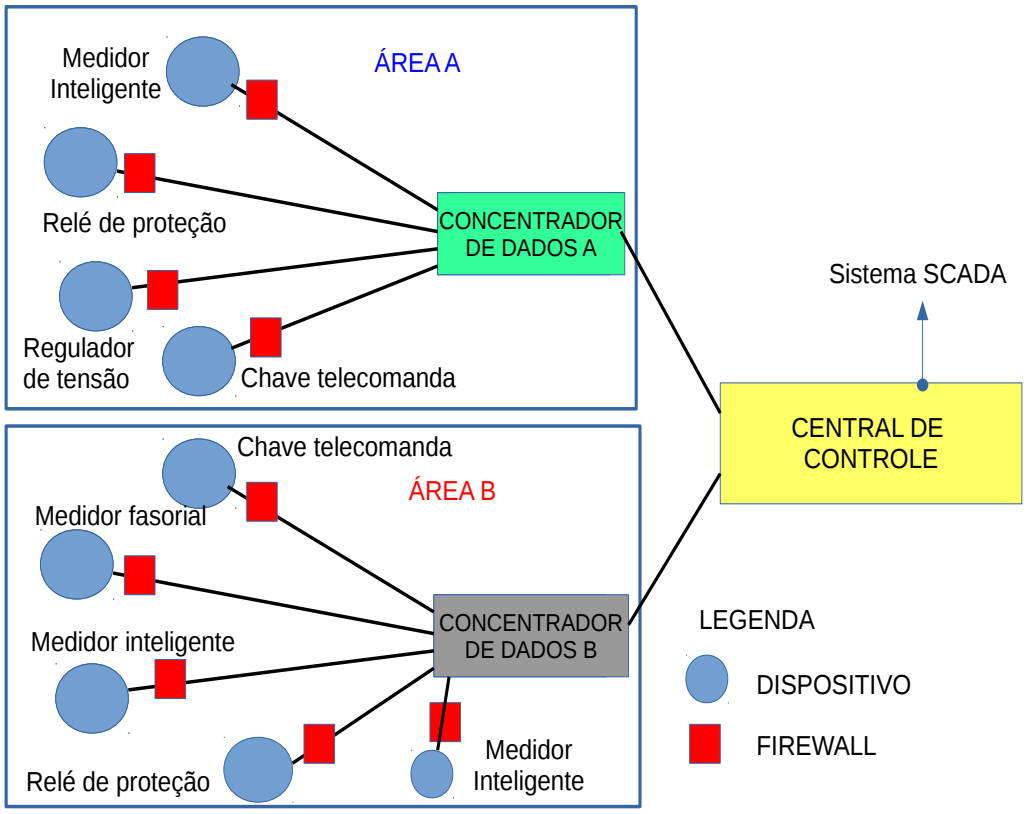
Além disso, por ser uma rede com um grande número de dispositivos, torna-se necessário dividir essa rede em sub-redes menores. Para isso, utiliza-se concentradores de informações. Assim, todas essas sub-redes devem estar interligadas a um único ponto (*firewall*).

Para resolver essas questões, uma alternativa interessante é a utilização de *firewalls* distribuídos. Nessa abordagem, diversos dispositivos podem realizar a filtragem de pacotes na

rede. Assim, cada dispositivo pode realizar essa ação de forma independente (JUNIOR, 2012).

A Figura 2.9 apresenta um cenário em que diversos dispositivos atuam como um *firewall* e estão interligados por meio da Internet. Esse cenário é típico de uma rede de comunicação para REI, onde diversos tipos de dispositivos (por exemplo: medidores inteligentes e equipamentos de proteção, monitoramento e controle) participantes de uma sub-rede enviam dados para um concentrador de dados. Esses concentradores, por sua vez, formam uma nova sub-rede, na qual se comunicam com uma central de controle e aquisição de dados.

Figura 2.9: *Firewall* distribuído.



Fonte: Acervo Pessoal

Conforme pode-se visualizar na Figura 2.9, cada dispositivo pode realizar a sua filtragem de pacotes. Dessa forma, essa filtragem se torna mais eficiente e escalável. Além disso, um *firewall* distribuído oferece uma maior proteção contra os ataques internos, ou seja, dispositivos participantes de uma rede local. Em uma rede de comunicação para REI, essa é uma

preocupação que deve ser prevista (JUNIOR, 2012).

2.5.1.2 Tipos de firewall

Quanto aos tipos de *firewalls*, eles podem ser divididos em três categorias, as quais estão relacionadas aos seus funcionamentos: filtros de pacotes tradicionais, filtros de estado e *gateways* de aplicação (KUROSE, 2013).

O primeiro tipo aplica um conjunto de regras a cada pacote que chega ou sai. Com base nessas regras, ele decide o tratamento que o pacote terá, ou seja, deixar prosseguir ou descartar. Essas regras são analisadas através de informações que constituem o cabeçalho do pacote de rede, como por exemplo: endereço IP (da origem e destino) e porta de origem e destino. Nesses termos, se determinado endereço IP ou porta corresponde a uma regra, o pacote analisado terá tratamento com base na ação relacionada a ela. Caso não exista uma correspondência, deve ser utilizada a política padrão, que pode ser liberar tudo o que não for proibido ou descartar tudo o que não for liberado. (STALLINGS, 2014).

Diferentemente de um firewall filtro de pacotes, no segundo tipo (filtro de estado) são rastreadas as conexões TCP para tomar decisões durante uma filtragem. Dessa forma, os pacotes são examinados com base em comunicações anteriores, ou seja, estabelecimento de novas seções ou encerramento dessas. O terceiro tipo de *firewall*, o *gateway* de aplicação tem funcionamento um pouco mais avançado, analisando dados da aplicação e não apenas cabeçalhos IP, TCP e UDP (KUROSE, 2013).

2.6 RESUMO DO CAPÍTULO

O Sistema Elétrico de Potência vem sendo aprimorado no decorrer do tempo, visando a convergência para a implementação das REI. Para isso, é essencial integrar novas tecnologias aos equipamentos presentes no SEP. Dessa forma, a implementação de redes de comunicação de dados é essencial. Por se tratar de um ambiente com grande fluxo de troca de informações, as quais apresentam um elevado grau de criticidade, garantir a segurança dessas informações representa um grande desafio. Além disso, deve-se garantir a segurança da própria rede de comunicação de dados. Dessa forma, o capítulo seguinte apresenta alguns trabalhos que abordam esses aspectos.

3 REVISÃO BIBLIOGRÁFICA

Esse capítulo apresenta alguns trabalhos relacionados a segurança das informações em REI. Dessa forma, as seções seguintes estão divididas da seguinte forma: a seção 3.1 aborda questões referentes a preocupação com a segurança em uma REI, a seção 3.2 descreve a importância de organizar dispositivos correlatos em grupos, possibilitando a comunicação em *multicast*, a seção 3.3 aborda aspectos relacionados a um *firewall* distribuído e a interpretação de regras escritas de forma genérica. Na seção 3.4 é apresentado um resumo sobre o presente capítulo.

3.1 TRABALHOS RELACIONADOS A SEGURANÇA DAS INFORMAÇÕES EM REI

A incorporação das tecnologias da informação e redes de comunicação bidirecionais no segmento de distribuição do SEP irão proporcionar um conjunto de novas aplicações, entre elas a infraestrutura de medição e atuação remota denominada *Advanced Metering Infrastructure* (AMI) (BROWN, 2008). Ao realizar esse tipo de avanço, expõe-se o sistema a novos tipos de ameaças cibernéticas, aumentando consideravelmente a vulnerabilidade do sistema como um todo. Possíveis falhas que podem ocorrer no sistema, devem representar um impacto mínimo aos consumidores, concessionárias e ao SEP como um todo. Para tornar possível a implementação de uma Rede Elétrica Inteligente é essencial, entre outras questões, que os equipamentos da AMI estejam conectados e troquem informações de forma segura. Essas trafegam na rede de comunicação e são críticas para operação do sistema.

Qualquer vulnerabilidade existente na rede de comunicação de dados pode ser utilizada por atacantes, internos ou externos, para a realização de ataques cibernéticos que podem visar a interceptação de dados, alteração de informações ou congestionamento de enlaces, dispositivos ou serviços de forma a tornar a sua utilização inviável. Um atacante, poderia por meio de acesso remoto, ou até mesmo adulteração de informações de controle e monitoramento, levar à indisponibilidade do sistema elétrico. A abrangência, dependendo dos elementos comprometidos, pode variar de poucos consumidores e até mesmo grandes *blackouts* envolvendo grandes áreas geográficas, como por exemplo, diversas cidades. Tais ações podem ser vulnerabilidades exploradas deliberadamente por um atacante motivado a causar danos ou, inadvertidamente, por usuários ou tráfegos de dados legítimos, em função de correlação de eventos e/ou falhas de

software (STALLINGS, 2014). Sendo assim, garantir a segurança das informações é imprescindível (NARUCHITPARAMES; GÜNEŞ; EVRENOSOGLU, 2011).

Uma questão que deve ser observada é que a ideia de REI não abrange somente o SEP tradicional, mas também áreas particulares de cada consumidor. A *Home Area Network* (HAN) permite comunicação entre os dispositivos consumidores dentro da área do usuário e, que irão comunicar-se com o sistema, de forma a interagir com ele, especialmente no que se refere ao controle de demanda. Dispositivos poderão, dentro de parâmetros aceitáveis, modificar horários de funcionamento em função de aspectos de tarifação dinâmica, por exemplo (RIVERA; ESPOSITO; TEIXEIRA, 2013). O elemento que serve de fronteira de comunicação entre os segmentos do SEP tradicionalmente conhecidos e a HAN é justamente o medidor inteligente (YAN et al., 2013). Essa integração eleva a importância dos mecanismos de segurança e controle a outro patamar. Neste contexto, a partir da HAN poderão ser exploradas vulnerabilidades que, se não contidas poderão até mesmo comprometer o SEP. Um claro contraste em relação aos mecanismos empregados hoje pelas concessionárias, onde basicamente a segurança existente deve-se ao fato do isolamento da rede de comunicação das demais redes e, especialmente, da Internet.

Nesses termos, a segurança das informações em um ambiente de REI é essencial e vem sendo amplamente discutida. Dessa forma, diversos trabalhos existentes na literatura buscam propor soluções capazes de garanti-la. Segundo (CLEVELAND, 2008), a maior parte desses trabalhos abordam as questões referentes ao gerenciamento e distribuição de chaves criptográficas, autenticação e criptografia de informações. Embora, tais aspectos sejam extremamente importantes, eles não podem solucionar todos os riscos e ameaças existentes nesse tipo de rede de comunicação. Para ilustrar um cenário que necessita que outras medidas sejam adotadas, (CLEVELAND, 2008) cita um medidor inteligente capaz de realizar conexões remotas. Nesse caso, se um atacante obter acesso a rede, ele pode enviar comandos maliciosos para esse dispositivo ou até mesmo, desconectar ele da rede.

Nesses termos, deve ser levado em consideração a proteção contra agentes externos ou atacantes que obtenham acesso de forma deliberada ou acidentalmente a rede de comunicação de uma REI. Para isso, a utilização de um *firewall* e de mecanismos de detecção de ataques é essencial. Seguindo essa linha, pode-se destacar os trabalhos realizados por (TONG et al., 2016) e (WANG; YI, 2011). O primeiro analisa as principais ameaças a um sistema de medição inteligente e apresenta a proposta de uma arquitetura distribuída de detecção de ataques, levando

em consideração as limitações de implementação desse tipo de mecanismo, como por exemplo, recursos computacionais dos equipamentos limitados e heterogeneidade de redes e protocolos de comunicação.

Em (WANG; YI, 2011) é apresentado um sistema de detecção de intrusões e resposta denominado *Smart Tracking Firewall*. Esse sistema se baseia em uma arquitetura de comunicação sem fio, através da topologia de uma rede *mesh* para interligar os dispositivos presentes em uma REI. Nesse sistema, cada nó possui um módulo do *Smart Tracking Firewall*, onde são implementados dois agentes de segurança: um agente de detecção de intrusão e um agente de resposta de intrusão. Além disso, cada nó de malha mantém duas listas: lista negra e lista cinza. Quando um determinado nó detecta um ataque proveniente de um nó malicioso, ele rejeita os seus pacotes e notifica os seus vizinhos. Ao receber essa notificação, cada vizinho adiciona o nó suspeito de ser um atacante a sua lista cinza. Se receber mais notificações (de outros vizinhos), o suspeito é confirmado como um nó malicioso (atacante). Então, ele é movido para a lista negra, na qual constam os dispositivos detectados e confirmados como agentes invasores (WANG; YI, 2011).

3.2 ORGANIZAÇÃO DE DISPOSITIVOS EM GRUPOS E COMUNICAÇÃO EM *MULTICAST*

Com base nos trabalhos apresentados por (WANG; YI, 2011) e (TONG et al., 2016) pode-se perceber a importância de utilizar mecanismos descentralizados para impedir a ação de atacantes, isto é, cada dispositivo pode atuar de forma independente, sem necessitar de um agente ou mecanismo centralizado. Entretanto, esses trabalhos não abordam aspectos referentes a organização dos dispositivos em grupos, bem como, a divulgação e aplicação de políticas e regras de segurança, baseando-se apenas em detecções de intrusões.

Em um cenário de REI, a ideia de organizar dispositivos é uma alternativa interessante, pois, possibilita realizar a divisão de uma rede grande em partes menores, nas quais, todos os dispositivos apresentam características e necessidades de níveis de segurança semelhantes. Com isso, consegue-se implementar mecanismos eficientes para aplicar medidas contra ataques, os quais podem ser detectados por abordagens semelhantes as citadas nesses trabalhos.

Seguindo essa linha, existem diversos protocolos e bibliotecas que possibilitam implementar uma rede DHT. O trabalho apresentado por (RHEA et al., 2005) descreve a implementação de uma biblioteca denominada OpenDHT, apresentando seus principais recursos e

interfaces, como por exemplo:

- *routing*: fornece acesso geral ao nó DHT responsável pela chave de entrada (denominado *bootstrap*) da rede e para cada nó ao longo do caminho de roteamento da rede DHT. Em termos gerais, o *bootstrap* é um ponto de referência para novos nós ingressarem na rede;
- *lookup*: fornece acesso geral ao nó DHT responsável por determinada chave;
- *storage*: oferece suporte as operações *put* (*key, value*) e *get* (*key*), roteando-as para o nodo DHT responsável por uma determinada chave.

Nesses termos, a biblioteca OpenDHT pode ser vista, em virtude das funcionalidades que oferece, como uma solução eficiente e que facilita o processo de criar uma rede DHT, seja para realizar o compartilhamento de dados ou o sincronismo de informações. O trabalho realizado por (DUAN; LI, 2007) vai ao encontro dessa afirmativa, destacando que a biblioteca OpenDHT opera em um conjunto de nós de infraestrutura sem exigir que as aplicações se preocupem em implementar a rede DHT. Segundo o autor, isso é um diferencial em relação a outras bibliotecas existentes.

(DUAN; LI, 2007) ainda afirma que essa biblioteca é muito flexível, possibilitando adicionar funcionalidades específicas de aplicações em cada um dos nós da rede DHT, mas cada aplicativo deve implementar sua própria DHT. Além disso, ele ressalta a flexibilidade na troca de informações entre os nós, pois, cada nó na implementação da OpenDHT mantém parte do armazenamento total do DHT em seu disco local e responde as operações *put* e *get*, podendo encaminhar mensagens para os demais membros da rede DHT.

3.3 FIREWALL DISTRIBUÍDO E INTERPRETAÇÃO DE REGRAS GENÉRICAS

Com base nos trabalhos relacionados a criação de uma rede DHT, percebe-se que essa abordagem destaca-se como uma solução eficiente para realizar o envio de informações para dispositivos participantes de um grupo. Além disso, pode-se relacionar essa forma de comunicação com o conceito de *firewall* distribuído. Dessa forma, pode-se realizar a integração dessas soluções para desenvolver uma arquitetura de segurança mais robusta, ou seja, utilizar o mecanismo de sincronismo de informações disponibilizada pela rede DHT para realizar a divulgação de regras para dispositivos que atuam como um *firewall* na rede de comunicação de uma REI.

Apesar de essa abordagem apresentar diversas vantagens, é necessário que exista um mecanismo que facilite o processo de criação e interpretação dessas regras, visto que, por ser

tratar de uma rede com grande variedade de dispositivos, diferentes aplicações de *firewall* podem ser utilizadas. Existem três formas de resolver essa prerrogativa:

- escrever regras para cada tipo de aplicação, divulgando-as para os dispositivos de forma ampla, ou seja, todas as regras escritas. Nesse caso, o dispositivo escolheria a regra de acordo com a aplicação que ele utiliza;
- escrever regras para cada tipo de aplicação e enviar separadamente, ou seja, manter os dispositivos organizados em grupos, classificados pelo tipo de *firewall*;
- escrever as regras de forma genérica e divulgar para todos os dispositivos. Nesse caso, o dispositivo interpreta a regra de acordo com a sua aplicação de *firewall*.

Com base nessas possibilidades, a terceira apresenta uma maior escalabilidade, visto que, o processo de escrita de uma regra é simplificado e representa menores possibilidades de erros durante a execução dessa tarefa. Além disso, simplifica-se o processo de divulgação das regras, necessitando um menor processamento e diminuindo o fluxo de informações na rede. Partindo dessa premissa, pode-se verificar em (MONTEIRO; VERDE; SOUTO, 2006) um XML Schema, denominado XSPSL, o qual realiza a especificação de políticas de segurança para redes de computadores. Nesses temos, o XSPSL foi desenvolvido utilizando uma linguagem declarativa baseada em objetos, a *Security Policy Specification Language* integrado com a linguagem *eXtended Markup Language* (XML).

A principal atribuição dessa proposta é possibilitar a escrita de regras genéricas que podem ser interpretadas e aplicadas em firewalls de diferentes fabricantes. Com isso, o administrador da rede pode especificar uma política de segurança em XML, sem necessidade de conhecimento de sintaxes específicas. Entretanto, (MONTEIRO; VERDE; SOUTO, 2006) não especifica em seu trabalho a forma como essas regras devem ser divulgadas.

3.4 RESUMO DO CAPÍTULO

Esse capítulo apresentou os principais aspectos referentes a segurança das informações em um ambiente de REI. Após o entendimento desses aspectos, foi abordado o conceito de *firewall* distribuído, o qual pode ser visto como uma alternativa interessante para garantir a segurança em diversos segmentos da rede de comunicação de uma REI. Além disso, foi abordado, na seção 3.2, uma solução para realizar a comunicação em *multicast* entre os dispositivos

participantes de uma REI.

Dessa forma, a integração dessa abordagem com os aspectos discutidos anteriormente, pode-se estabelecer uma proposta robusta e que proporciona alta escalabilidade para uma rede de comunicação de dados, principalmente, quando refere-se a medidas que buscam garantir a segurança dessa rede. Além disso, é possível destacar a interoperabilidade entre diferentes aplicações de *firewall*.

Nesses termos, o capítulo seguinte apresenta uma arquitetura que contempla os principais aspectos abordados no presente capítulo, ou seja, essa arquitetura implementa o conceito de *firewall* distribuído, utilizando uma rede DHT para realizar a divulgação de regras entre participantes de um grupo de dispositivos correlatos, as quais são escritas de forma genérica e podem ser interpretadas por diferentes aplicações de *firewall*.

4 METOLOGIA

Esse capítulo apresenta a visão geral de uma arquitetura desenvolvida para possibilitar a implementação de um *firewall* distribuído em ambientes de REI. Essa arquitetura é composta por três módulos, que atuam de forma complementar. Nesses termos, a seção 4.1 descreve essa arquitetura e nas seções seguintes (seções 4.2, 4.3 e 4.4) serão abordados cada um desses módulos. Na seção 4.5 é realizado uma descrição sobre a integração dos módulos que compõem a arquitetura. Além disso,a seção 4.6 apresenta uma descrição sobre os requisitos necessários para que a Arquitetura DIFMA possa ser aplicada em um determinado dispositivo e quais aspectos relacionados a segurança das informações podem ser solucionados por meio dessa arquitetura. Por fim, na seção 4.7, é apresentado um resumo sobre esse capítulo.

4.1 ARQUITETURA DIFMA

A arquitetura *Distributed Firewall Multiple Applications* (DIFMA) foi proposta, pelo autor desse trabalho, com o objetivo de facilitar a implementação do conceito de *firewall* distribuído, com ênfase na implementação em redes de comunicação de dados para REI. Dessa forma, cada equipamento ou dispositivo presente nessa rede, como por exemplo: medidores inteligentes, concentradores de dados, medidores fasoriais, sensores de grandezas elétricas e atuadores podem implementar suas próprias regras de filtragem de pacotes. Com isso, é possível proporcionar uma maior segurança do ponto de vista a ataques externos ou ataques provindos da rede local.

Basicamente, através de regras eficientes em um *firewall* consegue-se aplicar políticas de segurança e controlar todo o tráfego de informação que entra ou sai de uma rede ou de um dispositivo. Assim, é possível bloquear comunicações externas que não sejam oriundas de um agente ou dispositivo confiável. Entretanto, em uma rede de comunicação para REI existem outros riscos, como por exemplo, tentativas de ataques provindos de dispositivos presentes na rede interna. Um exemplo disso é um usuário mal-intencionado que pode tentar um ataque a um dispositivo utilizado pela concessionária de energia. Nesse cenário ele pode aproveitar-se por estar na mesma rede e começar a enviar um grande fluxo de dados para esse dispositivo com a intenção de realizar um ataque de negação de serviço.

Nesses termos, se utilizar apenas um ponto de *firewall*, localizado na borda da rede, ou

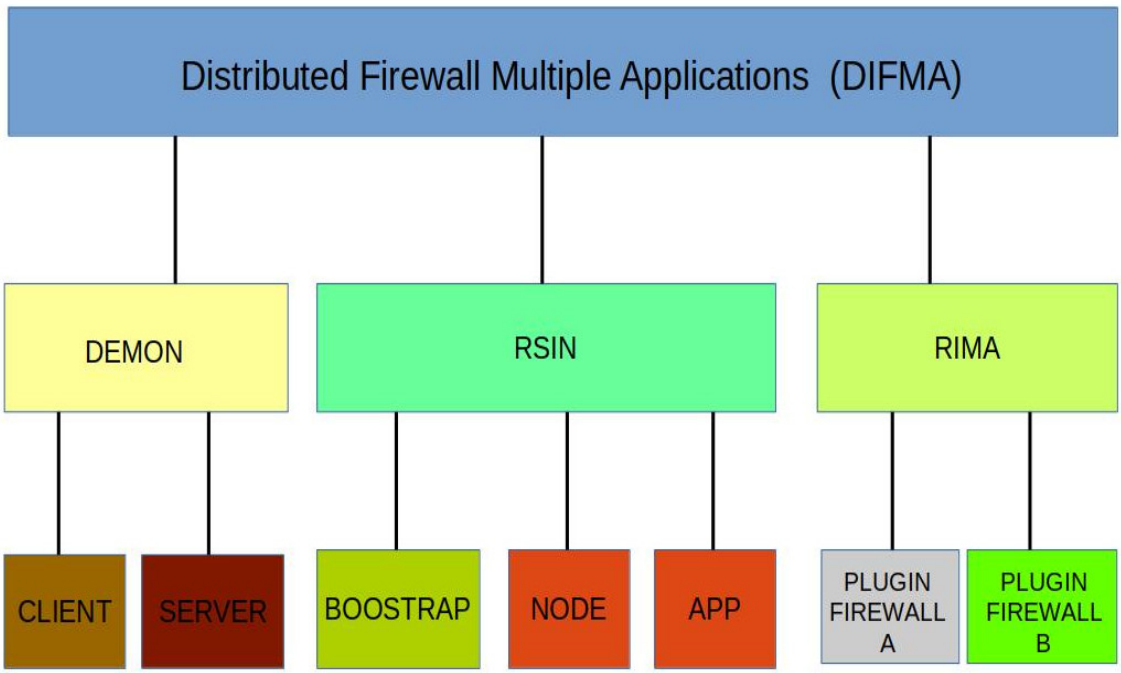
seja, conexão da rede interna com a Internet, esse tipo de ataque é mais difícil de ser mitigado. Com a utilização de um *firewall* distribuído, cada dispositivo pode implementar suas próprias políticas de filtragem de pacotes e regras, não dependendo de uma filtragem centralizada.

Entretanto, a heterogeneidade de equipamentos e dispositivos presentes na rede de comunicação de dados para REI apresenta um outro desafio: podem ser utilizados tipos de aplicações de *firewall* em um mesmo segmento da rede (LEE, 2010). Com isso, é necessário ter mecanismos para realizar a divulgação de regras e aplicação dessas em cada dispositivo.

Para solucionar tais paradigmas, a arquitetura DIFMA pode ser utilizada. Essa arquitetura foi desenvolvida a partir de três premissas básicas: organizar dispositivos em grupos com base em suas características de *hardware* e funcionalidades, envio de informações em *multicast* (enviar uma mensagem para os dispositivos participantes de um determinado grupo) e facilitar a divulgação de regras para um grupo, independentemente da aplicação de *firewall* utilizada pelos dispositivos.

Nesse contexto, a arquitetura DIFMA foi dividida em módulos que exercem essas funções de forma independente e complementar. A Figura 4.1 apresenta uma visão geral dessa arquitetura, demonstrando os seus módulos.

Figura 4.1: Visão geral da arquitetura DIFMA.



Fonte: Acervo Pessoal.

De acordo com a Figura 4.1, a arquitetura DIFMA apresenta três módulos: DEMON (*Devices Manager On Network*), RSIN (*Rules Synchronizer In Network*) e RIMA (*Rules Interpreter Multiple Applications*). O módulo DEMON é responsável por realizar o gerenciamento de dispositivos participantes da REI, realizando a organização desses em grupos em virtude de suas características, possibilitando a geração de chaves criptográficas e gerenciamento de certificados X509. Esse módulo disponibiliza duas aplicações: *client* e *server*. A aplicação DEMON *client* deve ser executada em cada dispositivo para realizar a geração de um par de chaves RSA, gerar uma requisição de certificado X509 e configuração dos grupos que esse dispositivo participará. Após esses passos, o dispositivo envia essas informações (requisição e grupos que pretende participar) para a aplicação DEMON *server*, a qual é executada em uma central de controle da concessionária, onde elas devem ser verificadas por um operador do sistema. A aplicação DEMON *server* é responsável por realizar o cadastro dos dispositivos e grupos em uma base de dados, gerar e enviar certificados para dispositivos autorizados. O módulo RSIN implementa uma rede de sobreposição para realizar a divulgação de regras que serão aplicadas nos dispositivos. O módulo RIMA realiza a interpretação de uma regra para uma determinada aplicação de *firewall*. Para isso, adiciona-se um *plugin* desenvolvido para essa aplicação. Assim, esse módulo é bastante escalável, pois, diversos *plugins* podem ser desenvolvidos caso pretenda-se utilizar novas aplicações de *firewall*.

Nesses termos, a arquitetura DIFMA é bastante robusta e a partir de seus módulos apresenta diversas funcionalidades. Para facilitar uma compreensão mais detalhada dessa arquitetura, as sessões seguintes apresentam maiores detalhes sobre cada módulo apresentado na Figura 4.1.

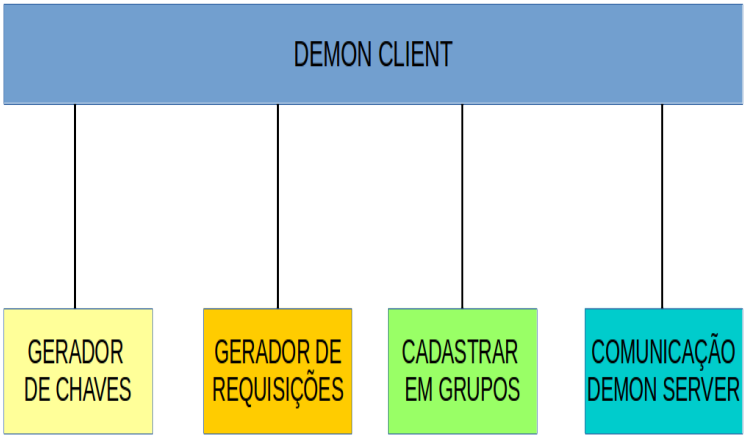
4.2 DEVICES MANAGER ON NETWORK (DEMON)

O módulo DEMON foi desenvolvido com a finalidade de facilitar o gerenciamento dos dispositivos participantes de uma REI. A sua concepção relaciona-se a premissa de que dispositivos com características semelhantes, em relação ao seu *hardware* e tarefas que realizam, apresentam as mesmas necessidades de regras de *firewall* para realizar a filtragem de pacotes. Assim, a criação de um grupo por parte desse módulo atende aos seguintes parâmetros: nome do grupo e papel (função) que o dispositivo executará.

Dessa forma, quando um novo dispositivo for ser inserido na rede, ele deve ser previa-

mente configurado por um operador do sistema, por exemplo, um funcionário da concessionária de energia elétrica. Para isso, o primeiro passo é determinar as características do equipamento e as funções que ele executará. Através dessas informações é possível inseri-lo em grupos existentes, nos quais, outros dispositivos semelhantes já fazem parte. Para ilustrar esse cenário, utiliza-se um medidor inteligente como exemplo, o qual pode ser atribuído ao grupo de medidores inteligentes (*Smart Meters*) e atuará como cliente. Esses medidores, geralmente, enviarão informações para um concentrador de dados, que participa do grupo chamado medidores inteligentes, exercendo o papel de concentrador. Nesse contexto, a aplicação DEMON *client* deve ser utilizada pelo operador que configurará o dispositivo. A Figura 4.2 apresenta as funções disponibilizadas por essa aplicação, desenvolvidos sob a forma de submódulos.

Figura 4.2: Submódulos da aplicação DEMON *client*.



Fonte: Acervo Pessoal.

Conforme a representação demonstrada na Figura 4.2, a primeira função da aplicação DEMON *client* é gerar chaves, ou seja, em sua primeira execução, a aplicação gera automaticamente um par de chaves (pública e privada) de 2048 bits, as quais são utilizadas para realizar criptografia de chave assimétrica através do algoritmo RSA. Após gerar o par de chaves, uma requisição de certificado X509 deve ser gerada, através do submódulo gerador de requisições. Para isso, a aplicação solicita que o operador insira as seguintes informações sobre o dispositivo que está sendo configurado, as quais constarão no certificado que será obtido posteriormente:

- C (*Country*): país do possuidor do certificado;

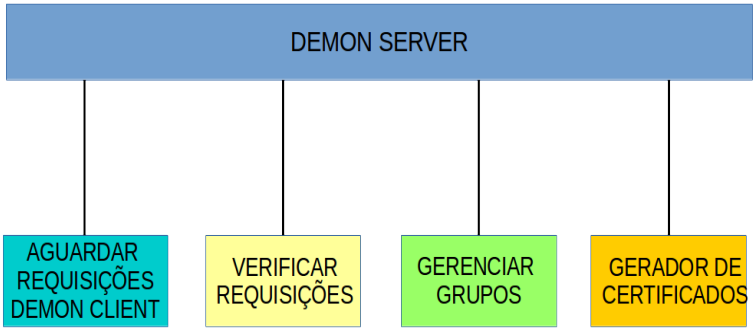
- ST (*State*): unidade federativa (estado) do possuidor do certificado;
- L (*Locality*): cidade do possuidor do certificado;
- O (*Organization*): organização ou empresa que possuidor do certificado está vinculado;
- CN (*Common Name*): nome que identifica o possuidor do certificado.

Com base nessas informações, a requisição é gerada. A próxima etapa é informar sobre grupos que o dispositivo participará e seus respectivos papéis nesses grupos. Essa função é realizada pelo terceiro submódulo (cadastrar em grupos). Para exemplificar os grupos de dispositivos que podem ser encontrados em um ambiente de REI pode-se citar: medidores inteligentes, concentradores de dados, medidores fasoriais, equipamentos de proteção, reguladores de tensão, entre outros.

Por fim, a aplicação solicita que o endereço IP atribuído ao dispositivo seja informado. Todas essas informações devem ser enviadas para a aplicação DEMON *Server*, o qual é executado em uma central de controle da concessionária. O envio dessas informações é realizada por meio do submódulo Comunicação DEMON *server*. Após o envio, a aplicação entra no modo de espera até receber o certificado X509. Quando ele receber, o certificado é salvo e o processo de configuração é encerrado. Esses processos são ilustrados na Figura 4.4.

Conforme mencionado anteriormente, a aplicação DEMON *server* é responsável por receber requisições e gerar certificados para os clientes (dispositivos) participantes da rede de comunicação de uma REI. Essa aplicação também é constituída por submódulos, que exercem diferentes funções, os quais são representados na Figura 4.3.

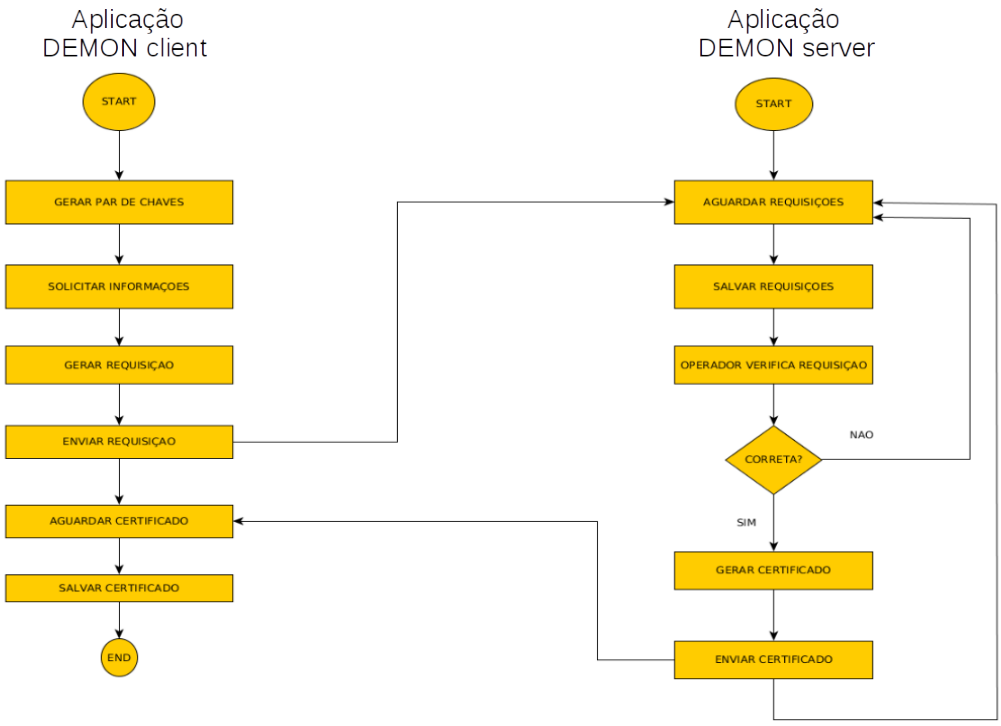
Figura 4.3: Submódulos da aplicação DEMON *server*.



Fonte: Acervo Pessoal.

De acordo com a Figura 4.3, o primeiro submódulo é responsável por aguardar as requisições provindas dos clientes, as quais são salvas em uma base de dados, para posteriormente serem verificadas por um operador do sistema. A verificação é disponibilizada pelo submódulo Verificar Requisições, o qual exibe em uma interface *web* todas as requisições que aguardam uma verificação. Nessa interface são exibidas as informações do dispositivo e também, sobre os grupos que ele deseja participar. Com base nessas informações, o operador pode gerar um certificado para o dispositivo ou negar, caso as informações estejam incorretas ou ele não tenha permissões para obter o certificado em questão. Em caso de sucesso na verificação, as informações referentes aos grupos que o dispositivo participará são salvas na base de dados para posteriores utilizações por outros módulos que compõem a arquitetura DIFMA. Além disso, é gerado o certificado e enviado para o dispositivo através do submódulo Geração de certificados. Nesses termos, a Figura 4.4 apresenta um fluxograma de funcionamento do módulo DEMON, onde todos os processos envolvidos desde a configuração inicial de um dispositivo até o recebimento do certificado X509 são representados.

Figura 4.4: Fluxograma de processos do módulo DEMON.



Fonte: Acervo Pessoal.

4.3 RULES SYNCHRONIZER IN NETWORK (RSIN)

A base de funcionamento desse módulo está relacionada a uma rede de sobreposição, através da utilização de uma rede DHT. A escolha por utilizar uma rede DHT para realizar uma comunicação *multicast* está relacionada com as abstrações da topologia da rede que podem ser realizadas, ou seja, uma rede DHT apresenta uma maior compatibilidade que protocolos específicos para realizar uma comunicação multicast, como por exemplo: o protocolo IGMP (*Internet Group Management Protocol*) e MLD (*Multicast Listener Discovery*). Dessa forma, uma rede DHT pode ser aplicada em dispositivos que utilizam endereços IPv4 e IPv6, diferentemente dos protocolo IGMP e MLD. Outro aspecto que evidencia a vantagem em utilizar uma rede DHT é a indexação de informações de forma distribuída na rede, ou seja, ao realizar o envio de uma informação para a rede, ela fica armazenada em diversos dispositivos. Com isso, caso um novo dispositivo ingresse na rede, uma informação, que já tenha sido indexada, pode ser facilmente sincronizada (novo dispositivo obtém uma informação divulgada antes de seu ingresso).

Nesse contexto, a principal finalidade do módulo RSIN é facilitar o envio e sincronismo de regras na arquitetura DIFMA. Em outras palavras, ele pode ser visto como um mecanismo capaz de enviar uma informação (regra) para todos os dispositivos participantes de um determinado grupo. Essa abordagem possibilita que esses dispositivos recebam uma determinada regra simultaneamente, reduzindo os tempos de transmissão, caso ela tivesse que ser enviada separadamente para cada um. Além disso, esse módulo permite que um dispositivo ao ingressar na rede receba todas as regras enviadas anteriormente, mantendo-se atualizado e sincronizado com os demais dispositivos que participam do mesmo grupo.

Assim, ele é composto por três aplicações distintas: RSIN *bootstrap*, RSIN *node* e RSIN *app*. A primeira aplicação é responsável por receber solicitações de ingresso na rede DHT e verificar se os dispositivos solicitantes possuem permissão para ingressar. Caso o dispositivo seja autorizado, a aplicação RSIN *bootstrap* realiza os procedimentos necessários para possibilitar o ingresso do dispositivo na rede DHT, como por exemplo: divulgar para os demais participantes da rede sobre o novo dispositivo e informar ao ingressante sobre os participantes da rede.

Nesses termos, a primeira etapa a ser executada pela aplicação RSIN *node* é comunicar-se com o *bootstrap*. Assim, o dispositivo realiza o processo de autenticação e posteriormente, ingresso na rede. Para comprovar sua autenticidade, ou seja, que ele é autorizado a ingressar na rede, o dispositivo envia o seu certificado X509, o qual foi emitido por uma CA confiável

por ambos, no caso, pela CA que gera os certificados no módulo DEMON. Após verificar a autenticidade, a aplicação RSIN *bootstrap* insere o dispositivo na rede.

A próxima etapa, após ingressar na rede de sobreposição, é abrir processos para receber informações destinadas aos grupos que o dispositivo participa. Para isso, ele busca os grupos em ele foi cadastrado. Essas informações são salvas em uma base de dados local pelo módulo DEMON. Dessa forma, é aberto um processo que executa a função "*listen*", disponibilizada pela biblioteca que implementa a rede de sobreposição, passando como parâmetro o nome do grupo e seu respectivo papel, os quais são utilizadas para gerar uma *hash* de 160 bits e formam o identificador de uma determinada sala.

Para exemplificar, vamos utilizar como base um medidor inteligente que atua como cliente do grupo denominado Smart_Meter. Nesse caso, ele deve escutar as informações enviadas para Smart_Meter:clientes. Ao inicializar o processo de escuta nessa sala, ele recebe as informações que já foram enviadas anteriormente. Além disso, se por algum motivo, o dispositivo ficar ausente da rede, quando ele voltar, recebe as informações que tenham sido enviadas nesse intervalo de tempo. Para realizar o envio de novas regras para os grupos de dispositivos, o operador do sistema deve utilizar a aplicação RSIN *app* que possibilita escrever uma regra de forma genérica, ou seja, utiliza um padrão que pode ser facilmente interpretado pelos *plugins* desenvolvidos para o módulo RIMA. Além disso, é possível realizar a revogação de um determinada regra por meio da aplicação RSIN *app*. Essa funcionalidade pode ser utilizada em situações que uma regra não seja mais necessária ou tenha sido escrita de forma errada pelo operador do sistema.

Em relação aos aspectos de segurança dessas informações, é essencial garantir a integridade e autenticidade das regras divulgadas, pois, caso o conteúdo delas sejam alterados por um invasor que obteve acesso a rede de forma deliberada, pode representar um grande risco ao funcionamento da rede de comunicação de dados. Além disso, deve-se garantir que a regra foi enviada por uma entidade confiável, no caso, por um operador do sistema. Para isso, todas as mensagens enviadas devem ser assinadas digitalmente. Ao receber uma regra, a aplicação RSIN *node* realiza a verificação da assinatura e, caso seja comprovada autenticidade e integridade da informação recebida, realiza uma chamada a um *plugin* do módulo RIMA, que interpretará e fará a aplicação da regra.

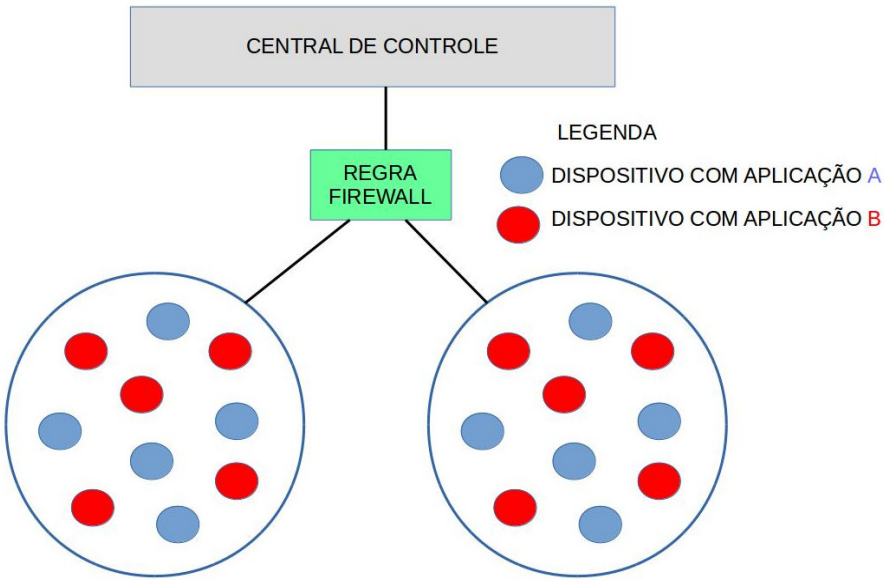
4.4 RULES INTERPRETER MULTIPLE APPLICATIONS (RIMA)

Para atender os requisitos de interoperabilidade de dispositivos e aplicações de *firewall* utilizadas em uma rede de comunicação de dados para REI, o módulo RIMA oferece uma solução para que diversas aplicações de *firewall* possam ser utilizadas na arquitetura DIFMA. A essência do seu funcionamento é a utilização de *plugins* para cada tipo de aplicação de *firewall*.

Nesses termos, para utilizar uma determinada aplicação de *firewall*, é necessário apenas adicionar um *plugin*, sem necessidade de modificações no código fonte dos demais módulos que compõem a arquitetura DIFMA. Além disso, o processo de envio das regras é facilitado, pois, não precisa que seja escrita uma regra para cada tipo de aplicação. Dessa forma, o *plugin* é responsável por receber uma regra genérica e interpretá-la de acordo com a aplicação de *firewall* para a qual foi desenvolvido.

Para exemplificar o funcionamento do módulo RIMA, vamos utilizar como base um cenário em que os dispositivos participantes de um grupo utilizam duas aplicações de *firewall* diferentes e estão divididos em duas sub-redes, mas respondem pelo mesmo identificador, conforme descrito na sessão anterior. Assim, para enviar uma regra, o operador do sistema escreve ela de forma genérica e envia através da rede criada pelo módulo RSIN. Dessa forma, a Figura 4.5 demonstra esse cenário.

Figura 4.5: Cenário de exemplo da utilização do módulo RIMA.



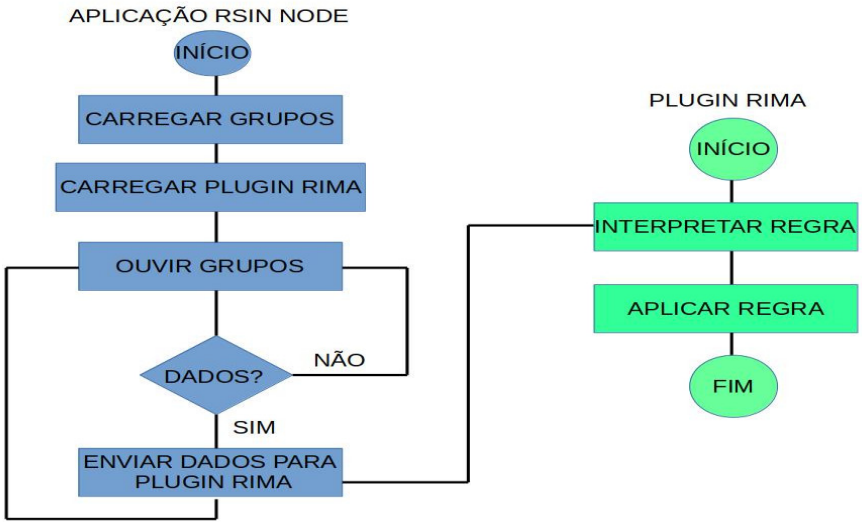
Fonte: Acervo Pessoal.

Conforme pode ser visto na Figura 4.5 e com base na descrição do módulo RSIN, ao enviar uma nova regra na rede de sobreposição, todos os dispositivos participantes de um determinado grupo receberão essa informação. Quando isso acontece, ela deve ser processada de forma diferente, ou seja, cada tipo de aplicação de *firewall* utiliza uma sintaxe específica para aplicar regras. Através da utilização de um *plugin*, o módulo RIMA possibilita interpretar essa regra. Nesse contexto, para escrever a regra, o operador utiliza uma sintaxe padrão, comum a todos os *plugins*. Um aspecto que deve ser ressaltado é a integração entre os módulos RSIN e RIMA, os quais atuam de forma complementar. Essa interação será detalhada na sessão a seguir.

4.5 INTEGRAÇÃO ENTRE MÓDULOS DA ARQUITETURA DFMA

Conforme descrito nas sessões anteriores, os módulos que compõem a arquitetura DIFMA atuam de forma complementar. A primeira relação que deve-se destacar é entre os módulos DEMON e RSIN, ou seja, o certificado X509 obtido por um dispositivo através da aplicação DEMON *client* é utilizado pela aplicação RSIN node, bem como, o credenciamento em grupos realizado pela aplicação do primeiro módulo. Assim, quando a aplicação RSIN *node* é inicializada, o identificador dos grupos que o dispositivo participa é carregado, para que se possa receber informações destinadas a esse grupo, conforme pode ser visualizado na Figura 4.6.

Figura 4.6: Integração entre módulos da arquitetura DFMA.



Fonte: Acervo Pessoal.

Conforme pode ser observado na Figura 4.6, os módulos RSIN e RIMA também estão diretamente relacionados. Durante a inicialização da aplicação RSIN *node*, o *plugin* desenvolvido para o tipo de aplicação de *firewall* que o dispositivo utiliza é carregado. Quando a aplicação RSIN *node* recebe uma informação, ela chama um *plugin* do módulo RIMA para processar a mesma. Assim, o *plugin* realiza a interpretação da regra e posteriormente, aplica ela.

4.6 APLICAÇÃO PRÁTICA DA ARQUITETURA DIFMA

A Arquitetura DIFMA foi desenvolvida para ser utilizada em equipamentos capazes de operarem em uma rede de comunicação por meio da pilha de protocolo TCP/IP. Outro aspecto que deve ser observado é a necessidade de atualização de *firmware* no dispositivo, ou seja, a Arquitetura DIFMA é um conjunto de aplicações que devem ser compiladas e incorporadas ao *firmware* de um determinado dispositivo. Dessa forma, em casos que não seja possível realizar uma atualização de *firmware*, recomenda-se a utilização de um equipamento intermediário, o qual possa estender a capacidade computacional do dispositivo em termos de *software*. Essa situação pode ser exemplificada por dispositivos legados presentes ou que estão sendo incorporados ao SEP, pois muitos utilizam um *firmware* fechado, com funcionalidades previamente definidas.

Nesses termos,pode-se destacar os seguintes equipamentos: medidores inteligentes e IED (*Intelligent Electronic Device*), como por exemplo: relés de proteção, chaves telecomandadas, reguladores de tensão, entre outros. Em muitos casos, esses equipamentos oferecem suporte apenas a comunicação com um sistema SCADA por meio de uma rede TCP/IP e protocolos de comunicações específicos. Entretanto, muitas vezes, não é possível garantir a segurança das informações trafegadas na rede de comunicação de dados e aplicar políticas de segurança (LOPES et al., 2012).

Nesses termos, a possibilidade de agregar um *firewall* em diversos pontos da rede de comunicação pode garantir uma maior segurança ao SEP de forma mais ampla, ou seja, protegendo um maior número de dispositivos contra acessos indevidos e ataques de negação de serviço, pode-se impedir que pontos mais críticos do SEP sejam protegidos. Dessa forma, um *firewall* é uma alternativa para garantir segurança nas seguintes situações:

- aplicação de políticas de segurança e controle de fluxo de informações: por meio de regras

de *firewall* é possível estabelecer uma política restritiva, bloqueando todas comunicações e liberando apenas as desejadas. Com isso, é possível permitir que um determinado equipamentos receba ou envie informações para dispositivos especificados nas regras aplicadas no *firewall*. Um exemplo disso, é liberar um IED para comunicar-se apenas com o sistema SCADA da concessionária de energia elétrica. Nesse caso, caso um atacante envie informações para esse equipamento, elas seriam descartadas;

- limitar o acesso remoto a um equipamento: muitas vezes é necessário que um equipamento seja acessado remotamente para a realização de atualização de *software* ou configurações específicas. Entretanto, o acesso remoto a um dispositivo pode ser usado para efetivação de um ataque na rede de comunicação de dados. Um exemplo disso é o acesso remoto a um IED onde o atacante, após acessar um dispositivo, pode enviar informações e comandos para outros equipamentos presentes na rede. Através de uma limitação de acesso remoto é possível impedir esse tipo de situação.

Nesse contexto, ao incorporar as aplicações que compõem a Arquitetura DIFMA ao *firmware* de um dispositivo, é possível aplicar medidas que garantam a segurança a nível de *firewall*. Além disso, é possível utilizar o certificado e o par de chaves gerados pelo módulo DEMON para a realização de outras técnicas de segurança, como por exemplo, criptografia e assinatura digital para evitar ataques onde um atacante intercepta e/ou modifica uma determinada informação (ataque *man-in-the-middle*). Entretanto, essas funcionalidades não fazem parte do escopo da arquitetura DIFMA.

Outro aspecto que deve ser observado é a quanto a topologia da rede e meios físicos utilizados para implementar uma rede de comunicação em uma REI. No decorrer nos últimos anos, alguns segmentos já apresentam algumas implementações nesse sentido. Com isso, é possível utilizar essas implementações para realizar a comunicação dos dispositivos com a aplicação que realiza a divulgação de regras (RSIN *app*). Nesses termos, podem ser utilizadas redes sem fio ou comunicação via PLC. Entretanto, é necessário que os equipamentos ofereçam suporte aos requisitos necessários (compatibilidade de bibliotecas, aplicações de *firewall*, atualização de *firmware* ou interação com equipamentos intermediários) para execução das aplicações que compõem a Arquitetura DIFMA. Nesses termos, esses requisitos limitam o escopo de aplicação da arquitetura proposta. Em equipamentos que não atendem a esses requisitos ou não utilizam a pilha de protocolos TCP/IP, a Arquitetura DIFMA não se aplica e deve-se utilizar outras técnicas de segurança,como por exemplo: segurança na camada física, criptografia e listas de

controle de acesso, as quais não fazem parte do escopo desse trabalho.

Quanto aos dispositivos que atendem aos requisitos para utilizar a Arquitetura DIFMA, recomenda-se a utilização dessa arquitetura nos equipamentos que representam uma maior criticidade quanto a preocupação com a segurança. Nesse caso, pode-se citar os medidores inteligentes, visto que eles são a fronteira entre a rede de comunicação do SEP e a rede local dos consumidores. Nesses termos, o medidor inteligente pode comunicar-se diretamente com dispositivos que estão constantemente conectados a Internet e podem ser alvos de um atacante. Outro equipamento muito crítico é o concentrador de dados, em razão do grande volume de informações que deve tratar. Um ataque nesse tipo de dispositivo pode afetar um grande segmento da rede de comunicação de uma REI.

4.7 RESUMO DO CAPÍTULO

O presente capítulo apresentou uma visão geral sobre a arquitetura DIFMA, descrevendo os seus módulos, a forma como eles atuam, os requisitos necessários para a sua implementação e os problemas que podem ser solucionados por meio da utilização de um *firewall* distribuído. Após o entendimento da arquitetura DIFMA e suas funcionalidades, o próximo capítulo abordará os aspectos de sua implementação, como por exemplo, as tecnologias e bibliotecas utilizadas. Além disso, serão apresentados testes realizados e resultados obtidos para validar essa arquitetura.

5 DESENVOLVIMENTO PRÁTICO

Conforme mencionado no capítulo anterior, a arquitetura DIFMA é composta por três módulos. Dessa forma, nas seções 5.1, 5.2 e 5.3 serão descritos os aspectos de implementação de cada um desses módulos. A seção 5.4 apresenta os testes e resultados realizados para validar a Arquitetura DIFMA e na seção 5.5 é realizada uma análise dos resultados obtidos.

5.1 MÓDULO DEMON

O módulo DEMON foi desenvolvido para possibilitar a geração de chaves assimétricas, facilitar o gerenciamento de certificados X509 e grupos de dispositivos correlatos. Para implementar tais funcionalidades foram desenvolvidas duas aplicações (*client* e *server*), utilizando-se a linguagem de programação C e compiladas para sistemas operacionais Linux através do compilador GCC, versão 5.4.0. Para realizar as operações relacionadas a geração de chaves e requisições de certificados utilizou-se a biblioteca *Openssl*, a qual é um projeto de código aberto e disponibiliza um conjunto de ferramentas para realizar a criptografia de informações (OPENSSL, 2017).

Além disso, esse módulo possui uma interface web que permite realizar, de forma facilitada, a emissão de um certificado para um dispositivo solicitante, bem como, realizar buscas sobre grupos e dispositivos que os compõem. Essa interface *web* foi desenvolvida com a utilização da linguagem PHP, a qual pode ser vista como uma linguagem de *script* de código aberto e utilizada, especialmente, para o desenvolvimento *web*. Para realizar a parte gráfica dessa aplicação, utilizou-se a linguagem de marcação HTML (*HyperText Markup Language*) e um mecanismo para adicionar estilos conhecido como CSS (*Cascading Style Sheets*) (HTML, 2017) (CSS, 2017). Como essa aplicação *web* deve realizar operações relacionadas a requisições e emissão de certificados X509, utilizou-se a biblioteca *phpseclib*, que disponibiliza funções para esses fins (PHPSECLIB, 2017).

Para explicar o funcionamento e implementação desse módulo, primeiramente, será descrita a aplicação DEMON *client*, a qual realiza a geração do par de chaves assimétricas (RSA) e posteriormente, solicita que o operador insira manualmente algumas informações, como por exemplo, as informações que constarão em seu certificado X509, endereço IP e porta que será utilizada pelo módulo RSIN e grupos que o dispositivo participará, bem como, os papéis que

o dispositivo exercerá nesses grupos. Vale ressaltar que essa aplicação deve ser executada logo após a instalação do dispositivo, ou seja, ela é o primeiro passo a ser realizado para que o dispositivo possa ingressar na rede de sobreposição e por consequente, ter acesso as regras de *firewall* divulgadas por meio do módulo RSIN.

Para ilustrar o funcionamento dessa aplicação, a Figura 5.1 apresenta uma tela de sua execução, a qual é realizada via terminal de comando do sistema operacional Linux. Nela nota-se as ações que ela realiza, como por exemplo: mensagem informando ao operador que o par de chaves RSA foi gerada, informações que ela solicita ao operador e ações realizadas após obter as informações necessárias, no caso, enviar a requisição para o servidor e aguardar o recebimento do certificado.

Figura 5.1: Tela da execução da aplicação DEMON client.

```
alexandre@alexandre-linux ~/codigos_RASP/client_DEMON $ ./client_DEMON
 ____  _____ __  __  ___  _   _
|  _ \| ____|  \/  |/ _ \| \ | |
| | | |  _| | |\/| | | | |  \| |
| |_| | |___| |  | | |_| | |\  |
|____/|_____|_|  |_|\___/|_| \_|

   Devices Manager On Network

Pair Keys not found. Generating a new pair
A request certificate should be generate. For this, provide the following informations:
Enter the Country: BR
Enter the Province: Rio Grande do Sul
Enter the City: Santa Maria
Enter the Organization: UFSM
Enter the Common: Smart_Meter_01
TABLE MY_GROUPS_NODE CREATED!
Enter the node IP: 172.16.56.196
Enter the node PORT: 9876
Enter the number of groups: 1
Enter the name group: Smart_Meters
Enter the paper: client
Sending request to server
 REQUEST SEND WITH SUCESS
Waiting for the Certificate
```

Fonte: Acervo Pessoal.

Para realizar o envio dos dados para o servidor, a aplicação DEMON *client* utiliza um *socket* TCP. Para facilitar esse envio, as informações são encapsuladas em único pacote, utilizando-se uma biblioteca denominada Binn, a qual permite serializar diversas informações em um único pacote, no caso, o pacote a ser enviado (BINN, 2017). Após realizar o envio dos dados para o servidor, a aplicação entra em modo de espera, ou seja, mantém o *socket* aberto até receber o certificado ou mensagem informando que ele não pode ser emitido. Para garantir que a resposta foi enviada pelo DEMON *server* e que ela está íntegra (não foi modificada), o pacote

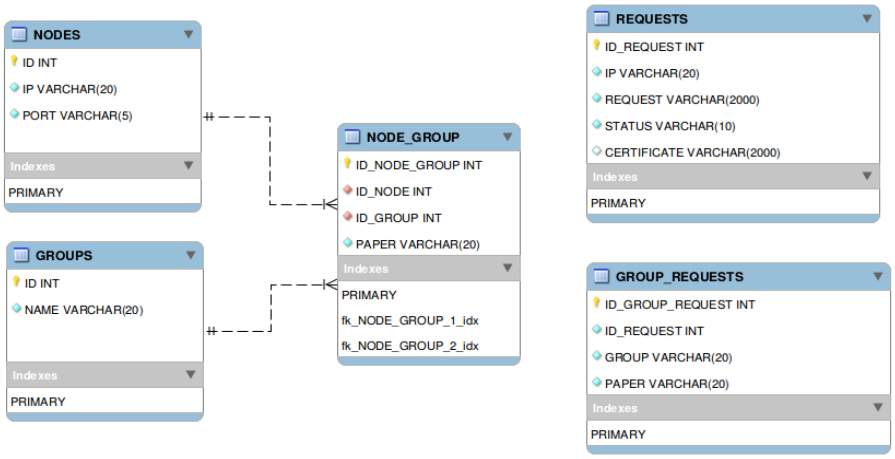
que o dispositivo recebe contém uma assinatura digital. Assim, caso a resposta seja interceptada ou alterada por um atacante, a aplicação DEMON *client* pode descartar o pacote recebido e reenviar a solicitação de certificado para a aplicação DEMON *server*. Em caso de sucesso no recebimento do certificado, ele é salvo localmente no dispositivo. Além disso, deve-se destacar que as informações referentes a grupos e papéis que o dispositivo foi cadastrado são salvas em uma base de dados no dispositivo.

Conforme mencionado anteriormente, a aplicação DEMON *server* é a responsável por possibilitar, ao operador do sistema, a verificação de requisições de certificados X509 e suas respectivas emissões. Essa aplicação deve ser executada nos servidores localizados nas centrais de controle do sistema, como por exemplo, no *datacenter* de uma concessionária de energia elétrica.

A aplicação DEMON *server* pode ser vista como uma interface *web* que exibe informações sobre as requisições enviadas pelos clientes (executando a aplicação DEMON *client*) e informações referentes ao gerenciamento de grupos e dispositivos. Para exibir essas informações são realizadas buscas no banco de dados utilizado pela aplicação.

A criação do banco de dados é realizado automaticamente por meio de uma aplicação escrita na linguagem C e que deve ser executada antes de acessar a interface *web*. Nesses termos, ao executar essa aplicação, é solicitada a criação de uma senha para o usuário que administrará a interface *web*. Posteriormente, o banco de dados é criado, utilizando o modelo físico apresentado na Figura 5.2.

Figura 5.2: Modelo físico do banco de dados da aplicação DEMON *server*.



Fonte: Acervo Pessoal.

Com base no modelo entidade-relacionamento apresentado na Figura 5.2, é possível obter-se as seguintes informações: requisições de certificados que ainda não foram analisadas, certificados emitidos, nome dos grupos cadastrados no sistema, dispositivos participantes de um determinado grupo, grupos que um determinado dispositivo participa e seus respectivos papéis nesses grupos. Nesses termos, para que esse banco de dados seja populado, ou seja, receba informações, a aplicação DEMON *server* mantém uma porta aberta para receber as conexões provindas dos clientes. Ao receber uma requisição, ela é salva no banco de dados.

Em relação a interface *web*, ao ser acessada pelo operador do sistema, é exigida a autenticação. Após inserir suas credenciais, o usuário é direcionado para a página principal que exibe as funcionalidades disponíveis, conforme pode ser visualizado na Figura 5.3. Nessa seção, será apresentada os detalhes referentes a opção de verificar as requisições que ainda não foram respondidas. As demais funcionalidades serão descritas no Apêndice A.

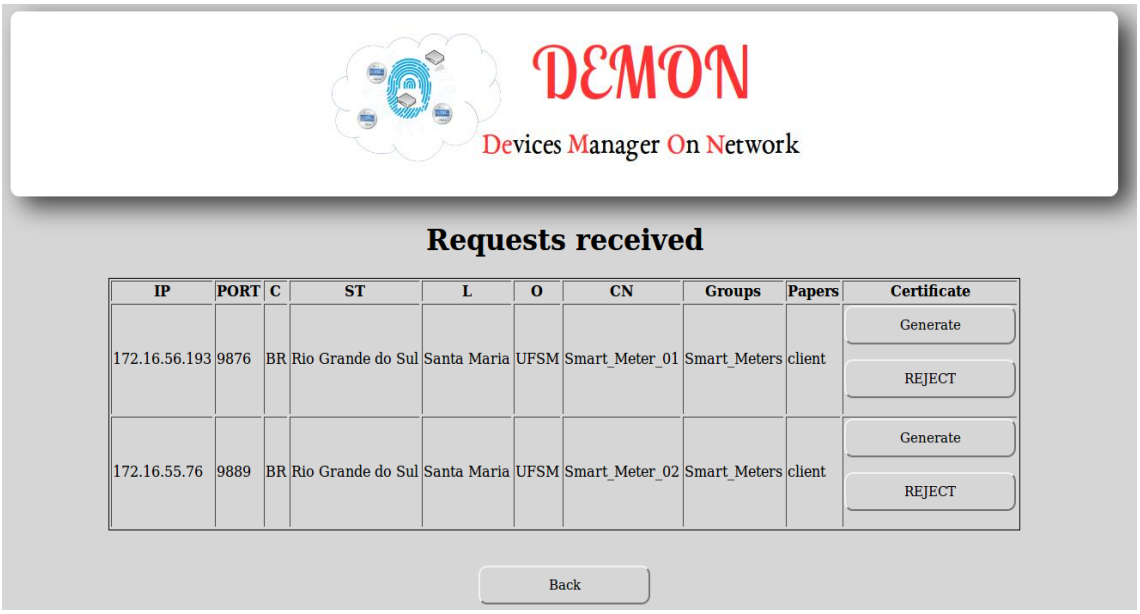
Figura 5.3: Página principal da interface web.

Fonte: Acervo Pessoal.

Para analisar as requisições pendentes, o operador ao clicar na aba *List Open Requests* é direcionado para uma nova página, a qual exibe informações referentes a cada requisição. Essas informações são extraídas da requisição com o auxílio de uma função disponibilizada pela biblioteca *phpseclib*. Assim, o operador consegue verificar a identidade do requerente do

certificado e grupos que ele deseja ser cadastrado. A Figura 5.4 ilustra essa página, na qual são apresentadas requisições enviadas por dois dispositivos.

Figura 5.4: Página que exibe as requisições ainda não verificadas.



Fonte: Acervo Pessoal.

Conforme pode-se observar na Figura 5.4, ao lado das informações sobre cada requisição existem dois botões. O primeiro deles (*Generate*) possibilita criar o certificado e enviá-lo para o dispositivo requisitante e deve ser utilizado quando as informações estão corretas e o dispositivo é autorizado a ingressar nos grupos solicitados. O segundo botão (*REJECT*) deve ser utilizado em casos que o certificado não pode ser emitido, em virtude de inconsistência nas informações apresentadas ou falta de autorização para participar dos grupos solicitados. Ao escolher a ação a ser realizada e clicar no respectivo botão, a aplicação DEMON *server* abre um *socket* para enviar a resposta para o dispositivo (que no caso, manteve o *socket* aberto para esperar a resposta do servidor).

5.2 MÓDULO RSIN

Após um dispositivo obter o seu certificado por meio do módulo DEMON, ele pode solicitar o ingresso na rede de sobreposição e assim, receber as regras destinadas aos grupos

que ele participa. Essa rede de sobreposição, também denominada rede DHT, é implementada por meio da biblioteca OpenDHT. Essa biblioteca foi escrita na linguagem de programação C++ e disponibiliza funções que permitem realizar a autenticação e inserção de dispositivos na rede de sobreposição, envio de mensagens em *multicast* e receber informações destinadas a um determinado grupo. Para realizar operações voltadas a verificação de certificados, autenticidade e integridade de uma informação, a biblioteca OpenDHT utiliza funções disponibilizadas pela biblioteca Gnutls (OPENDHT, 2017).

Conforme discutido anteriormente, para que um dispositivo possa ingressar na rede DHT, ele precisa se comunicar com algum dispositivo que já esteja nessa rede. Para simplificar esse processo, na arquitetura DIFMA utiliza-se uma aplicação (RSIN *bootstrap*) que possibilita o ingresso de novos dispositivos. Essa aplicação pode ser executada nos concentradores de dados ou nos servidores de uma concessionária de energia elétrica. Vale ressaltar que após um dispositivo ingressar na rede DHT, ele pode permitir o ingresso de novos dispositivos. Entretanto, por questões relacionadas ao gerenciamento e controle da rede, optou-se por restringir essa operação a apenas os dispositivos que utilizam a aplicação RSIN *bootstrap*. Nesses termos, a aplicação RSIN *bootstrap* ao receber uma solicitação de ingresso, realiza procedimentos para verificar se o dispositivo possui autorização para ingressar na rede. Para isso, ela verifica o certificado enviado pelo solicitante. O processo de envio da solicitação e do certificado é estabelecido conforme a implementação da biblioteca OpenDHT (OPENDHT, 2017).

Entretanto, foram necessários alguns ajustes no código fonte da biblioteca OpenDHT. A principal modificação foi na função que carrega o certificado que deve ser enviado ao dispositivo que atua como *bootstrap*. Originalmente, a biblioteca utiliza uma função que gera um novo par de chaves e um novo certificado (OPENDHT, 2017). Assim, essa função foi modificada para carregar as chaves e certificados gerados por meio do módulo DEMON. Outro aspecto que deve ser ressaltado é a forma como as informações são enviadas na rede DHT. Para facilitar a montagem da mensagem que representa uma regra escrita por meio da aplicação RSIN *app*, utilizou-se a biblioteca Binn. Dessa forma, é possível montar o pacote a ser enviado com base em uma chave e seu respectivo valor.

Nesses termos, a aplicação RSIN *app* ao ser executada solicita ao usuário (no caso, um operador do sistema) as informações referentes a regra que deve ser enviada. Dessa forma, a partir das informações inseridas é formada uma regra genérica que pode ser facilmente interpretada pelo *plugins* desenvolvidos para o módulo RIMA. A Figura 5.5 apresenta uma tela que

ilustra a criação de uma nova regra.

Figura 5.5: Tela da execução da aplicação RSIN *app*.

```
alexandre@alexandre-sala302:~/workspace/RSIN$ ./RSIN
 ____  ____  ___ _   _
|  _ \/ ___|_ _| \ | |
| |_) \___ \| ||  \| |
|  _ < ___) | || |\  |
|_| \_\____/___|_| \_|

Rules Synchronizer In Network
Enter the name of the group that desire applied a rule: Smart_Meters:client
Enter the OPERATION: ADD
Enter the FLOW: OUTPUT
Enter the source IP: ***
Enter the destiny IP: 10.6.6.6
Enter the source PORT: ***
Enter the destiny PORT: 666
Enter the PROTOCOL: ***
Enter the destiny ACTION: ACCEPT
Sending Rule to group: Smart_Meters:client
Rule sent with Sucess!
```

Fonte: Acervo Pessoal.

Conforme pode ser visualizado na Figura 5.5, a aplicação RSIN *app* fornece uma forma interativa de escrever as regras que pretende-se aplicar nos dispositivos presentes na rede de comunicação de uma REI. A primeira informação que ela solicita ao usuário é o nome do grupo em que deseja-se aplicar a regra. Posteriormente, é solicitado os campos necessários para gerar a regra genérica. Caso algum campo não seja necessário, utiliza-se os caracteres ***. Do ponto de vista dos campos solicitados, eles podem ser definidos da seguinte forma:

- operation: estabelece o tipo de operação que deseja-se realizar, ou seja, adicionar uma nova regra (utiliza a sintaxe ADD) ou revogar uma regra já existente (utiliza a sintaxe DEL);
- flow: refere-se ao sentido do fluxo dos pacotes que a regra visa filtrar, ou seja: pacotes que chegam no dispositivo (INPUT) ou pacotes que saem do dispositivo (OUTPUT);
- source IP: se refere ao endereço IP de origem do pacote a ser filtrado;
- destiny IP: se refere ao endereço IP de destino do pacote a ser filtrado;

- source Port: se refere a porta de origem do pacote a ser filtrado;
- destiny Port: se refere a porta de destino do pacote a ser filtrado;
- protocol: possibilita filtrar os pacotes com base em um determinado protocolo;
- action: define a ação que deve ser realizada caso um pacote filtrado coincida com a regra, ou seja, ele pode ser aceito (ACCEPT) ou descartado (DROP).

Após receber todas as informações solicitadas, a aplicação RSIN *app* realiza a montagem do pacote, atribuindo cada campo a sua respectiva chave. Concluída essa etapa, a regra genérica é enviada para o grupo de dispositivos informado. Para isso, a aplicação ingressa na rede DHT e insere a informação na rede por meio da função *put* da biblioteca OpenDHT, passando como parâmetro o nome do grupo e a regra genérica. Vale ressaltar, que a informação é assinada para permitir os dispositivos possam verificar a sua autenticidade e integridade (OPENDHT, 2017). Por se tratar de uma informação que não requer confidencialidade, optou-se por não criptografar as mensagens.

Para que seja possível receber as regras enviadas para um determinado grupo, cada dispositivo cria um processo para esperar informações destinadas a esse grupo por meio da função *listen* disponibilizada pela biblioteca OpenDHT (OPENDHT, 2017). Ao receber uma informação assinada, o dispositivo realiza a verificação da assinatura. Caso seu conteúdo esteja íntegro, a informação (no caso a regra genérica) deve ser processada. Para isso, a informação recebida é passada para o *plugin* correspondente ao tipo de aplicação de *firewall* utilizada pelo dispositivo.

5.3 MÓDULO RIMA

O módulo RIMA possibilita interpretar uma regra genérica recebida por meio do módulo RSIN. Esse módulo se destaca por permitir a interpretação de regras para diversas aplicações de *firewall* sem que seja preciso modificar o código fonte da aplicação que recebe as informações. Sob o ponto de vista de sua implementação, ele pode ser visto como uma extensão do módulo RSIN. Em outras palavras, o módulo RIMA não é composto por uma aplicação, e sim, um conjunto de *plugins* que estendem as funcionalidades da aplicação RSIN *node*. Em relação a essa integração com o módulo RSIN, os *plugins* são carregados pela aplicação RSIN *node* durante a sua inicialização. Esse carregamento ocorre de forma dinâmica, ou seja, quando a

aplicação RSIN *node* é inicializada, verifica-se o tipo de aplicação de *firewall* utilizada pelo dispositivo e carrega-se o seu respectivo *plugin*.

Nesses termos, para interpretar uma regra para uma determinada aplicação de *firewall*, utiliza-se um *plugin* específico, o qual deve ser construído, utilizando como base, a estrutura apresentada na sessão anterior (conforme pode ser visualizado na Figura 5.5). Sendo assim, esses *plugins* devem implementar uma função que receba uma variável do tipo *string* e extraia as informações contidas nela, ou seja, a *string* é formada a partir de um objeto criado com a biblioteca Binn e contém diversos valores atribuídos a suas respectivas chaves. Essas chaves são padronizadas e nomeadas da seguinte forma: *flow*, *ip_src*, *ip_dst*, *port_src*, *port_dst*, *protocol* e *action*.

Após extrair as informações que compõem uma regra genérica, o *plugin* devem realizar a interpretação dessa regra, ou seja, criar uma regra com base na sintaxe da aplicação de *firewall* para a qual foi desenvolvido. Para isso, deve ser realizada a associação dos campos da regra genérica com os campos específicos de cada sintaxe. Posteriormente, o *plugin* realiza a aplicação da regra por meio de uma chamada de sistema.

Um aspecto que deve ser destacado é a interoperabilidade entre diversas aplicações de *firewall*, ou seja, quando houver necessidade de utilizar uma determinada aplicação, basta apenas desenvolver ou adicionar um *plugin* correspondente. Dessa forma, foram desenvolvidos, inicialmente, dois *plugins* para as seguintes aplicações de *firewall*: Iptables e UFW.

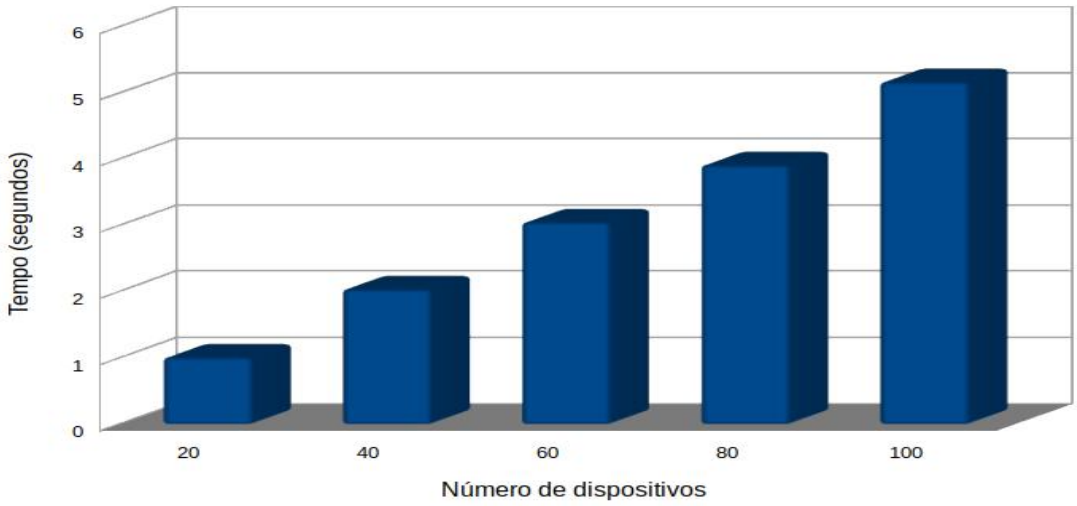
5.4 TESTES E RESULTADOS

Para verificar o desempenho da rede DHT, implementada por meio do módulo RSIN, em relação ao tempo necessário para sincronizar uma informação, foi desenvolvido um ambiente de testes onde um dispositivo enviava uma informação para os demais participantes da rede DHT. Além disso, nesse cenário optou-se por variar o número de dispositivos utilizados. Com isso, pode-se analisar o comportamento da rede em relação a escalabilidade, isto é, a medida que novos participantes sejam inseridos, não deseja-se que a rede apresente um aumento na latência acima de níveis desejáveis.

Para realizar esses testes utilizou-se um emulador de redes denominado Mininet (MININET, 2017), o qual foi instalado em servidor com sistema operacional Ubuntu Server 16.04 com as seguintes configurações de *hardware*: 64 GB de memória RAM e processador *Intel Xeon* E5-2640 v4 2.4GHz . Dessa forma, foram utilizados cenários com 20, 40, 60, 80 e 100 disposi-

tivos atuando como receptores. Nesses termos, a Figura 5.6 apresenta os resultados obtidos nos testes descritos anteriormente. Os tempos apresentados referem-se ao intervalo decorrido desde o início do processo de sincronismo e confirmação de recebimento da mensagem enviada, ou seja, receptor notifica o emissor que recebeu os dados enviados.

Figura 5.6: Testes da comunicação em *multicast* através do uso da rede DHT.



Fonte: Acervo Pessoal.

Após realizar os testes para verificar o tempo necessário para sincronizar uma informação na rede DHT, foi desenvolvido um novo ambiente de testes. Nesse novo ambiente utilizou-se dois dispositivos (Raspberry Pi 3) executando a aplicação RSIN *node* para receber as regras geradas a partir da aplicação RSIN *app*, sendo que em um dispositivo , foi utilizado o *plugin* para a aplicação de *firewall* Iptables, e no outro, para a aplicação UFW. As aplicações RSIN *app* e RSIN *bootstrap* foram executadas em computador com sistema operacional Linux, com processador Intel Core I5 e com 6 GB de memória RAM.

Nesses termos, para verificar a eficiência da aplicação RSIN *app* quanto a geração de regras genéricas e a capacidade dos *plugins* desenvolvidos em interpretar essas regras, utilizou-se como base os seguintes cenários:

- Cenário 1: escrever uma regra para bloquear um determinado serviço ou protocolo;
- Cenário 2: escrever uma regra que permita que um dispositivo possa ser acessado por um determinado *host*, utilizando uma determinada porta e um protocolo específico;

- Cenário 3: escrever uma regra que bloqueie as comunicações originadas de um determinado endereço IP.

O cenário 1 representa uma situação em que se deseja impedir que um dispositivo possa ser acessado por um determinado serviço, como por exemplo, o SSH, que permite acesso remoto a um *host*. Tratando-se de dispositivos presentes no sistema elétrico, pode-se citar o exemplo de um medidor inteligente, que pode ser utilizado por atacante como um *backdoor* e assim, executar comandos remotos ou comunicações com outros equipamentos participantes na rede, como por exemplo, enviar um grande fluxo de informações para um concentrador de dados ou IED com o objetivo de realizar um ataque de negação de serviço. Nesse caso, para gerar a regra genérica deve ser informada a porta utilizada pelo serviço que deseja-se restringir, nesse caso, a porta 22 (porta padrão do SSH). A Figura 5.7 apresenta a tela de execução da aplicação RSIN *app* para criar a regra para esse cenário.

Figura 5.7: Tela da execução da aplicação RSIN *app* para gerar regra do cenário 1.

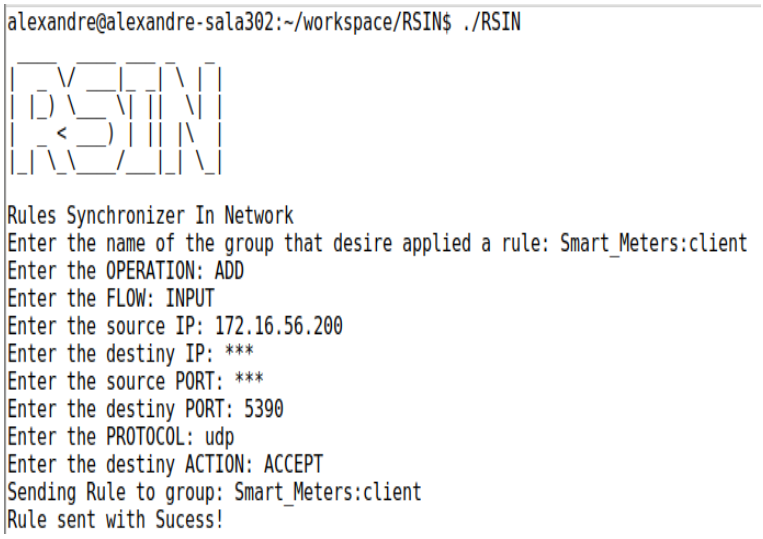
```
alexandre@alexandre-sala302:~/workspace/RSIN$ ./RSIN

 ____  ____  ___ _   _
|  _ \/ ___|_ _| \ | |
| |_) \___ \| ||  \| |
|  _ < ___) | || |\  |
|_| \_\____/___|_| \_|

Rules Synchronizer In Network
Enter the name of the group that desire applied a rule: Smart_Meters:clien
Enter the OPERATION: ADD
Enter the FLOW: INPUT
Enter the source IP: ***
Enter the destiny IP: ***
Enter the source PORT: ***
Enter the destiny PORT: 22
Enter the PROTOCOL: tcp
Enter the destiny ACTION: DROP
Sending Rule to group: Smart_Meters:client
Rule sent with Sucess!
```

Fonte: Acervo Pessoal.

O cenário 2 ilustra uma situação típica em uma rede de comunicação para REI, onde um sistema SCADA comunica-se com os dispositivos através de uma determinada porta, utilizando o protocolo UDP. Ao aplicar esse tipo de regra, em uma política restritiva, é possível restringir o acesso ao dispositivo por meio dessa porta, ou seja, apenas o *host* autorizado poderá se

comunicar, evitando que um atacante a utilize para a realização de uma comunicação com objetivos maliciosos. Para gerar a regra genérica para esse cenário deve ser informado o endereço IP do *host* de origem, a porta e protocolo utilizado. Nesse cenário, utilizou-se o endereço IP 172.16.56.200 (*host* que executa o sistema SCADA), por meio da porta 5390 e protocolo UDP. A Figura 5.8 apresenta a tela de execução da aplicação RSIN *app* para criar a regra para esse cenário.

Figura 5.8: Tela da execução da aplicação RSIN *app* para gerar regra do cenário 2.



Fonte: Acervo Pessoal.

O cenário 3 pode ser visto como uma situação em que deseja-se impedir comunicações originadas de um endereço IP específico, como exemplo, um *host* utilizado por um atacante para realizar um possível ataque de negação de serviço a um equipamento presente no sistema elétrico (por exemplo, um concentrador de dados). Além disso, é possível restringir a comunicação de dispositivos de uma determinada rede por meio de uma range de endereços IP's. Com isso, é possível impedir que dispositivos que façam parte de uma determinada rede acessem equipamentos que participam de uma outra rede. Isso pode ser útil quando tem-se uma segmentação de uma rede em sub-redes, por exemplo, medidores inteligentes não podem acessar um equipamento de proteção remotamente. Nesse cenário, utilizou-se como exemplo o endereço IP 172.16.56.221. A Figura 5.9 apresenta a tela de execução da aplicação RSIN *app* para criar a regra para esse cenário.

Figura 5.9: Tela da execução da aplicação RSIN *app* para gerar regra do cenário 3.

```
alexandre@alexandre-sala302:~/workspace/RSIN$ ./RSIN
 ____  ____ ___ _   _
|  _ \/ ___|_ _| \ | |
| |_) \___ \| ||  \| |
|  _ < ___) | || |\  |
|_| \_\____/___|_| \_|

Rules Synchronizer In Network
Enter the name of the group that desire applied a rule: Smart_Meters:client
Enter the OPERATION: ADD
Enter the FLOW: INPUT
Enter the source IP: 172.16.56.221
Enter the destiny IP: ***
Enter the source PORT: ***
Enter the destiny PORT: ***
Enter the PROTOCOL: ***
Enter the destiny ACTION: DROP
Sending Rule to group: Smart_Meters:client
Rule sent with Sucess!
```

Fonte: Acervo Pessoal.

Para ilustrar as regras interpretadas por cada um dos *plugins* desenvolvidos, a Figura 5.10 apresenta a tela de execução da aplicação RSIN *node* no dispositivo que utiliza o *firewall* Iptables.

Figura 5.10: Regras interpretadas pelo *plugin* desenvolvido para a Iptables.

```
root@raspberrypi:~/home/pi/RSIN_NODE# ./RSIN_NODE
 ____  ____ ___ _   _
|  _ \/ ___|_ _| \ | |
| |_) \___ \| ||  \| |
|  _ < ___) | || |\  |
|_| \_\____/___|_| \_|

Rules Synchronizer In Network
Plug-in to IPTABLES loaded!

Processing rule to IPTABLES...

RULE INTERPRETED: iptables -A INPUT --dport 22 -p tcp -j DROP

Rule applied with sucess!

Processing rule to IPTABLES...

RULE INTERPRETED: iptables -A INPUT -s 172.16.56.200 --dport 5390 -p udp -j ACCEPT

Rule applied with sucess!

Processing rule to IPTABLES...

RULE INTERPRETED: iptables -A INPUT -s 172.16.56.221 -j DROP

Rule applied with sucess!
```

Fonte: Acervo Pessoal.

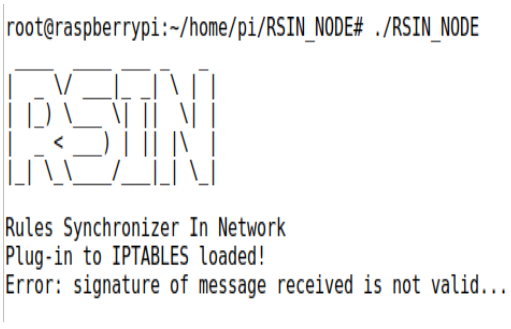
Quanto as regras interpretadas pelo *plugin* desenvolvido para a aplicação de *firewall* UFW, elas podem ser visualizadas na Figura 5.11.

Figura 5.11: Regras interpretadas pelo *plugin* desenvolvido para UFW.

Fonte: Acervo Pessoal.

Após verificar o comportamento dos *plugins* ao interpretar as regras genéricas, optou-se por realizar um teste onde foi simulada a interceptação e modificação de uma regra enviada pela aplicação RSIN *app*. Para isso, foi realizada uma modificação no código fonte da aplicação RSIN *node*, de forma que, ao receber uma informação (regra), o dispositivo alterava os valores contidos na mensagem original e reencaminhava para os demais participantes do grupo. Sendo assim, a mensagem não obteve sucesso nas verificações de integridade e autenticidade realizadas pela aplicação RSIN *node*, conforme pode-se observar na Figura 5.12, que apresenta um aviso informando o erro ocorrido (falha na assinatura) e ação a ser realizada (descartar mensagem).

Figura 5.12: Aviso de falha na verificação da assinatura de uma mensagem.



Fonte: Acervo Pessoal.

5.5 ANÁLISE DE RESULTADOS

Conforme pode ser observado na Figura 5.6, a utilização de uma rede DHT como alternativa para realizar o sincronismo de informações é uma solução viável, visto que ela requer um curto intervalo de tempo para enviar uma informação a diversos dispositivos. Além disso, pode-se considerar que é possível alcançar uma ótima escalabilidade na arquitetura proposta nesse trabalho, em virtude do discreto crescimento do tempo para sincronizar uma informação, com base nos diferentes cenários utilizados no ambiente de testes realizados. Esse crescimento pode ser explicado pelo mecanismo de confirmação de recebimento das informações enviadas, pois nesse caso, diversos dispositivos enviam informações para um único receptor que, necessariamente, realiza o tratamento de todas elas. Vale ressaltar ainda, que durante a execução dos testes, não foram percebidas eventuais falhas de comunicação e perdas de pacotes.

Visto que os resultados obtidos para divulgar uma informação para um determinado grupo mostraram-se satisfatórios, a rede de sobreposição criada por meio do módulo RSIN viabilizou a divulgação das regras genéricas geradas a partir da aplicação RSIN *app*. Dessa forma, após gerar uma nova regra e iniciar o processo de divulgação, não foram constatadas ocorrências de problemas no envio ou recebimento de mensagens.

Em relação a criação de novas regras, a aplicação RSIN *app* simplifica essa tarefa ao operador do sistema, visto que não é necessário preocupar-se com sintaxes específicas de cada aplicação de *firewall*. Conforme pode ser observado nas Figuras 5.10 e 5.11, as regras foram interpretadas corretamente. Dessa forma, a integração do módulo RSIN com os *plugins* desenvolvidos para o módulo RIMA mostrou-se eficiente e capaz de garantir a interoperabilidade entre múltiplas aplicações de *firewall*.

Outro aspecto importante é o mecanismo de verificação de autenticidade e integridade de uma informação recebida, o qual é integrado a aplicação RSIN *node* por meio de uma função disponível na biblioteca OpenDHT. Dessa forma, é possível assinar e verificar as assinaturas de todas as mensagens enviadas ou recebidas (OPENDHT, 2017). Além disso, é importante ressaltar a importância do módulo DEMON, o qual possibilitou organizar de forma simplificada os dispositivos em grupos, proporcionar a geração de chaves criptográficas e implementar a infraestrutura de uma CA convencional.

6 CONSIDERAÇÕES FINAIS

Prover uma rede comunicação de dados segura que permita amplo emprego dos conceitos relativos a REI é um grande desafio. Esse tem sido um intenso tema de pesquisa, conforme pode-se observar nas referências utilizadas nesse trabalho. Em virtude da grande heterogeneidade de equipamentos, tecnologias e protocolos que podem ser empregados para prover a comunicação entre os dispositivos presentes no SEP, a escalabilidade e interoperabilidade são aspectos que devem ser levados em consideração. Além disso, a segurança das informações e da rede de comunicação de dados são imprescindíveis para garantir que as REI possam ser implementadas.

Nesses termos, a arquitetura DIFMA se caracteriza por possibilitar a implementação de um *firewall* descentralizado, ou seja, diversos dispositivos participantes da rede de comunicação de dados de uma REI podem atuar como *firewall*. Essa abordagem apresenta diversos benefícios em relação a utilização de um único *firewall* na rede. Um exemplo disso é a proteção contra ataques provindos da rede interna. Em alguns casos, dependendo da topologia da rede (uma rede *mesh* por exemplo), um determinado dispositivo (por exemplo, um medidor inteligente ou um IED de proteção) poderia ser alvo de ataques que não seriam bloqueados por um *firewall* localizado na borda da rede. Nesse caso, após o atacante obter acesso a esse dispositivo, poderia utilizar a rede de comunicação de dados para atacar outros dispositivos. Através da utilização de um *firewall* distribuído esse tipo de situação pode ser evitada, pois, consegue-se aplicar políticas de segurança mais restritivas, limitando o fluxo de informações que um equipamento pode enviar ou receber.

Outro exemplo que realça a importância de utilizar um *firewall* distribuído é proteger o medidor inteligente quanto a acessos remotos, visto que ele possui comunicação com diversos dispositivos localizados na HAN. Dessa forma, um atacante poderia obter acesso a esses dispositivos e assim, utilizar o medidor inteligente como um *backdoor* para realizar comandos remotos na rede de comunicação de uma REI. Através da utilização de *firewalls* em diversos pontos da rede é possível agregar maior proteção e restringir determinados tipos de acessos e fluxos de informações suspeitos.

Para viabilizar a implementação de um *firewall* distribuído em um ambiente tão diversificado como uma REI, a arquitetura DIFMA provê uma forma eficiente para realizar a divulgação de regras que devem ser utilizadas em cada dispositivo. Por meio da implementação de uma

rede DHT é possível enviar as regras para diversos dispositivos que participam de um grupo. Assim, minimiza-se os tempos necessários para realizar a divulgação de regras. Outro aspecto importante, quanto a utilização de uma rede de sobreposição, é o seu funcionamento que abstrai aspectos referentes a topologia da rede, o que pode ser visto como um diferencial a protocolos específicos para a comunicação em *multicast*.

A organização e gerenciamento de dispositivos são abordados na arquitetura DIFMA por meio do módulo DEMON. Esse módulo permite criar grupos de dispositivos correlatos, ou seja, que possuem características semelhantes e exercem funções específicas. Além disso, o módulo DEMON atua como uma CA convencional, permitindo a geração de chaves criptográficas e gerenciamento de certificados no padrão X509. Através da utilização de chaves criptográficas e certificados é possível garantir os requisitos de autenticidade e integridade durante a divulgação de uma regra na rede DHT.

Em relação a divulgação das regras, elas devem ser geradas por um operador do sistema. Para facilitar esse processo, a arquitetura DIFMA se destaca por oferecer um mecanismo que possibilita gerar uma regra de forma genérica e que possa ser interpretada para diversas aplicações de *firewall*. Dessa forma, além de oferecer maior praticidade, o risco de erros durante a geração da regra é reduzido, visto que, o operador não precisa preocupar-se com sintaxes específicas de cada aplicação de *firewall* que possa ser utilizada pelos dispositivos participantes de um grupo. Para isso, necessita-se apenas que seja desenvolvido um *plugin* que realize essa interpretação. Dessa forma, ao adicionar um plugin capaz de processar uma regra recebida, não é necessário modificar o fonte da aplicação que recebe as informações (RSIN *node*).

Durante os testes realizados para verificar o desempenho da arquitetura DIFMA, foram geradas regras genéricas (através da aplicação RSIN *app*) e interpretadas para as aplicações Iptables e UFW, através dos *plugins* desenvolvidos. Com base nos resultados obtidos, percebeu-se que essa abordagem demonstra ser uma alternativa interessante e viável de ser implementada em um cenário real. Além disso, a rede DHT mostrou-se muito eficiente para realizar o sincronismo de informações (regras geradas), apresentando resultados satisfatórios, seja, no tempo necessário para enviar uma informação a diversos dispositivos participantes de um grupo ou na possibilidade de expansão da rede. Dessa forma, pode-se afirmar que a arquitetura DIFMA oferece escalabilidade a rede de comunicação de dados em uma REI e contribui de forma satisfatória para manter a interoperabilidade entre as aplicações de *firewall*. Nesses termos, pretende-se expandir ainda mais as funcionalidades da arquitetura DIFMA. Entre as possíveis melhorias

pode-se destacar:

- aprimorar os *plugins* existentes: os *plugins* desenvolvidos para realizar os testes apresentados nesse trabalho focaram em interpretar regras direcionadas a filtrar os pacotes que chegam ou saem de um *host*. Dessa forma, pretende-se possibilitar a interpretação de regras mais avançadas, tratando regras utilizadas para fazer redirecionamento de pacotes e inspeção do estado de pacotes (*firewall* do tipo filtro de estado);

- desenvolver novos *plugins*: para realizar os testes optou-se desenvolver *plugins* para as aplicações Iptables e UFW por serem muito utilizadas em dispositivos que utilizam sistemas operacionais Linux e possuírem um sintaxe simples e fácil de ser utilizada. Dessa forma, será realizada um pesquisa sobre outras aplicações de *firewall* disponíveis e que possam ser utilizada em dispositivos presentes em uma REI. Após essa pesquisa, serão desenvolvidos *plugins* para tais aplicações;

- desenvolver novos módulos para a arquitetura DIFMA, como por exemplo, um módulo que possibilite detectar ataques e aplicar regras para impedi-los. Além disso, pode-se implementar uma abordagem semelhante a apresentada em (WANG; YI, 2011), classificando-se os ataques detectados em duas classes: lista cinza e lista negra. Assim, quando o modular detectar um possível ataque, ele notificaria os demais participantes da rede sobre o ocorrido. Ao receber notificações confirmando que trata-se de um ataque, seria gerada uma regra para bloquear esse ataque.

REFERÊNCIAS

BAIG, Z. A.; AMOUDI, A.-R. An analysis of smart grid attacks and countermeasures. **Journal of Communications**, v.8, n.8, p.473–479, 2013.

BINN. acessado em 16/07/2017, `https://github.com/liteserver/binn`.

BROWN, R. E. Impact of smart grid on distribution system design. **Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE**, p.1–4, 2008.

CGEE. Centro de Gestão e Estudos Estratégicos (Org.). **Redes elétricas inteligentes: contexto nacional.**, 2012.

CLEVELAND, F. M. Cyber security issues for advanced metering infrasttructure (AMI). **Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE**, p.1–5, 2008.

CSS. acessado em 16/07/2017, `https://www.w3schools.com/css/default.asp`.

DUAN, X.; LI, J.-S. Overlay Network Testing by OpenDHT. **Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on**, v.1, p.245–248, 2007.

EKANAYAKE, J. B. et al. **Smart grid**: technology and applications. New Delhi: John Wiley & Sons, 2012.

GHANSAH, I. Smart grid cyber security potential threats, vulnerabilities and risks. **California Energy Commission, PIER Energy-Related Environmental Research Program, CEC-500-2012-047**, 2009.

HTML. acessado em 16/07/2017, `https://www.w3schools.com/html/default.asp`.

JUNIOR, E. d. C. P. An Architecture for Self-adaptive Distributed Firewall. **XVI SIMPÓSIO BRASILEIRO EM SEGURANÇA DA INFORMAÇÃO E SISTEMAS COMPUTACIONAIS**, 2012.

KHOSROSHAHI, A. H.; SHAHINZADEH, H. Security Technology by using Firewall for Smart Grid. **Bulletin of Electrical Engineering and Informatics**, v.5, n.3, p.366–372, 2016.

KUROSE, J. F. **Redes de computadores e a Internet**: uma abordagem top-down. 6.ed. São Paulo: Pearson Education do Brasil, 2013.

LAMIN, H. **ANÁLISE DE IMPACTO REGULATÓRIO DA IMPLANTAÇÃO DE REDES INTELIGENTES NO BRASIL. 2013. 300 p**. 2013. Tese (Doutorado em Ciência da Computação) — Tese. Curso de Engenharia Elétrica, Departamento de Engenharia Elétrica, Unb, Brasília.

LEE, A. Guidelines for Smart Grid Cyber Security. **NIST Interagency/Internal Report (NISTIR)-7628**, 2010.

LOPES, Y. et al. Smart Grid e IEC 61850: novos desafios em redes e telecomunicações para o sistema elétrico. **XXX Simpósio Brasileiro de Telecomunicações**, 2012.

MCTI. Ministério da Ciência, Tecnologia e Inovação. **Redes Eletricas Inteligentes: Diálogos Setoriais Brasil-União Européia**, 2014.

MININET. acessado em 16/07/2017, `http://mininet.org/`.

MONTEIRO, P.; VERDE, J. V.; SOUTO, P. Um XML Schema para Especificação de Políticas de Segurança. **1ª Conferência Ibérica de Sistemas e Tecnologias de Informação**, 2006.

MORAES, A. F. d. **Firewalls**: segurança no controle de acesso. 1.ed. São Paulo: Érica, 2011.

NARUCHITPARAMES, J.; GÜNEŞ, M. H.; EVRENOSOGLU, C. Y. Secure communications in the smart grid. **Consumer Communications and Networking Conference (CCNC), 2011 IEEE**, p.1171–1175, 2011.

OPENDHT. acessado em 16/07/2017, `https://github.com/savoirfairelinux/opendht/wiki/API-Overview`.

OPENSSL. acessado em 16/07/2017, `https://www.openssl.org/`.

PHPSECLIB. acessado em 16/07/2017, `https://github.com/phpseclib/phpseclib`.

RHEA, S. et al. OpenDHT: a public dht service and its uses. **ACM SIGCOMM Computer Communication Review**, v.35, n.4, p.73–84, 2005.

RIVERA, R.; ESPOSITO, A. S.; TEIXEIRA, I. Redes elétricas inteligentes (smart grid): oportunidade para adensamento produtivo e tecnológico local. **Revista do BNDES, Rio de Janeiro**, n.40, p.43–83, 2013.

STALLINGS, W. **Segurança de computadores**: princípios e práticas. 2.ed. Rio de Janeiro: Elsevier, 2014.

STALLINGS, W. **Criptografia e segurança de redes**: princípios e práticas. 6.ed. São Paulo: Pearson Education do Brasil, 2015.

STEPANEK, R. Distributed firewalls. **Seminar on Network Security, Helsinki University of Technology, Finland**, 2001.

STOICA, I. et al. Chord: a scalable peer-to-peer lookup service for internet applications. **ACM SIGCOMM Computer Communication Review**, v.31, n.4, p.149–160, 2001.

TANENBAUM, A. S. **Redes de Computadores**. 5.ed. São Paulo: Pearson Prentice Hall, 2011.

TONG, W. et al. A Survey on Intrusion Detection System for Advanced Metering Infrastructure. **Instrumentation & Measurement, Computer, Communication and Control (IMCCC), 2016 Sixth International Conference on**, p.33–37, 2016.

URDANETA, G.; PIERRE, G.; STEEN, M. V. A survey of DHT security techniques. **ACM Computing Surveys (CSUR)**, v.43, n.2, p.8, 2011.

WANG, W.; XU, Y.; KHANNA, M. A survey on the communication architectures in smart grid. **Computer Networks**, v.55, n.15, p.3604–3629, 2011.

WANG, X.; YI, P. Security framework for wireless communications in smart distribution grid. **IEEE Transactions on Smart Grid**, v.2, n.4, p.809–818, 2011.

YAN, Y. et al. A survey on smart grid communication infrastructures: motivations, requirements and challenges. **IEEE communications surveys & tutorials**, v.15, n.1, p.5–20, 2013.

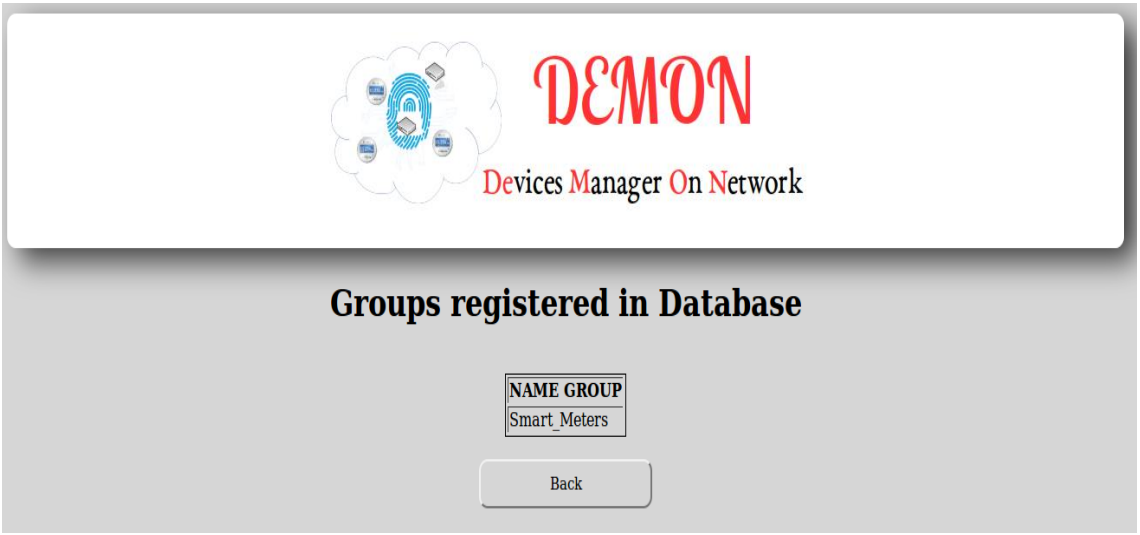
APÊNDICES

APÊNDICE A – Funcionalidades disponibilizadas pela interface web da aplicação DEMON server

A interface *web* da aplicação DEMON *server* disponibiliza diversas funcionalidades que permitem, ao operador do sistema, realizar o gerenciamento de grupos e dispositivos presentes na rede de comunicação de uma REI. Entre essas funcionalidades pode-se destacar: listar os grupos cadastrados na base de dados, criação de novos grupos de dispositivos, consultas sobre os dispositivos que participam de um determinado grupo, consulta sobre quais grupos que um dispositivo participa e consultas sobre os certificados emitidos.

Para que um dispositivo possa ser inserido um determinado grupo, deve-se verificar se o grupo que o dispositivo deseja participar existe. Para isso, o operador deve acessar o botão *List Groups* na página principal da interface *web* da aplicação DEMON server. Assim, ele é direcionado para uma página que lista todos os grupos cadastrados na base de dados, conforme pode ser visualizado na Figura A.1.

Figura A.1: Página que exibe os grupos cadastrados na base de dados.



Fonte: Acervo Pessoal.

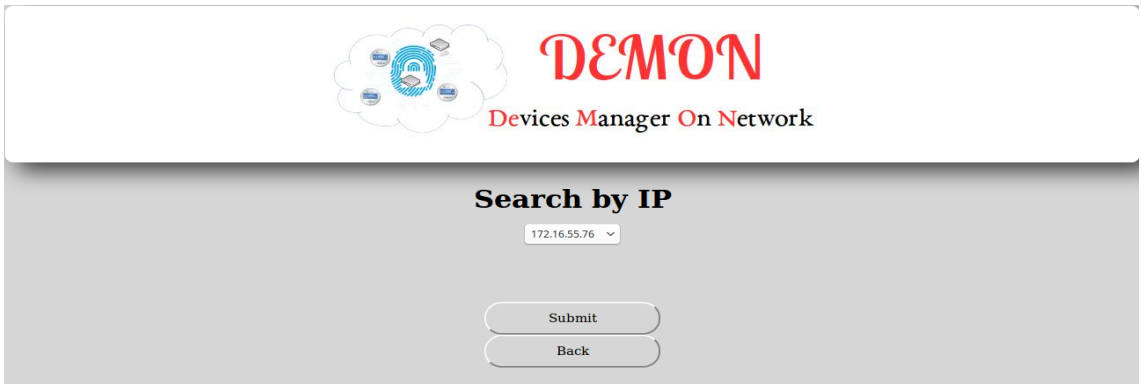
Caso o grupo desejado não exista, ele pode ser criado. Para isso, o operador deve acessar o botão *New Group*. Ao realizar essa ação, ele é redirecionado para a página que solicita o nome do grupo a ser criado, conforme pode ser visualizado na Figura A.2.

Figura A.2: Página que possibilita criar um novo grupo.

Fonte: Acervo Pessoal.

Para realizar uma consulta sobre os grupos que um determinado dispositivo participa deve ser utilizado o botão *Search by* IP na página principal. Dessa forma, o usuário é redirecionado para uma página que exibe todos os dispositivos cadastrados na base de dados, conforme pode ser visualizado na Figura A.3.

Figura A.3: Página que possibilita pesquisar os grupos em um dispositivo está cadastrado.

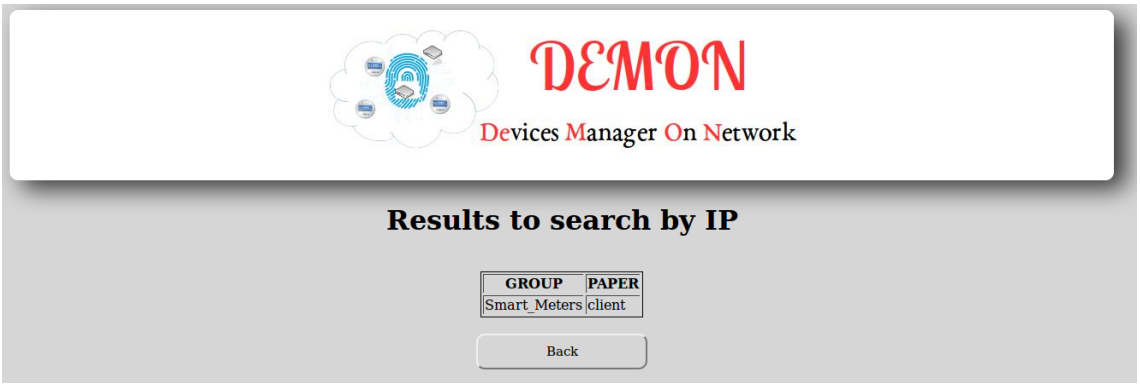


Fonte: Acervo Pessoal.

Após escolher o dispositivo que deseja-se pesquisar (através de seu endereço IP), basta clicar no botão *Submit*. Assim, o usuário será redirecionado para a página que apresenta os grupos que o dispositivo participa e respectivos papéis que exerce, conforme pode ser visualizado

na Figura A.4.

Figura A.4: Página que exibe os grupos que um dispositivo participa.



Fonte: Acervo Pessoal.

Outra opção disponível é pesquisar os dispositivos cadastrados em um determinado grupo. Para isso, o operador deve clicar no botão *Search by Group*. Ao realizar essa ação, ele é direcionado para uma página que possibilita escolher o nome do grupo que deseja pesquisar, conforme ilustrado na Figura A.5.

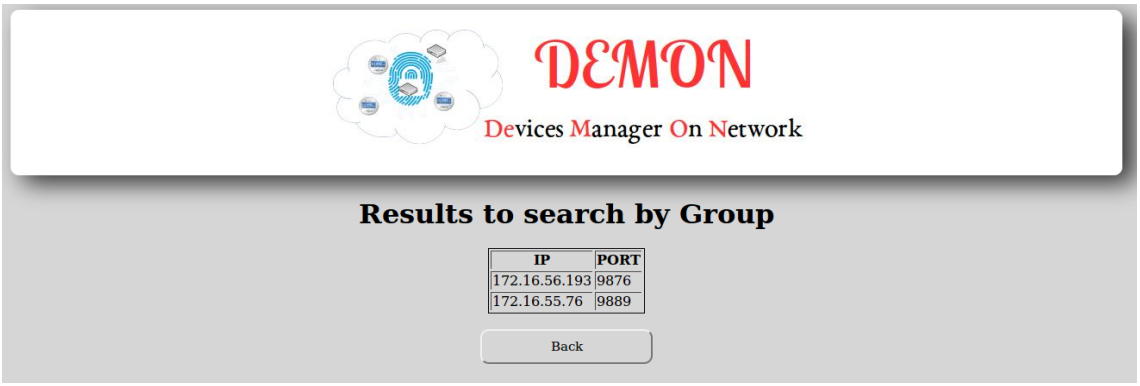
Figura A.5: Página que possibilita pesquisar os dispositivos cadastrados em um grupo.

Fonte: Acervo Pessoal.

Após escolher o grupo que deseja pesquisar, basta clicar no botão *Submit*. Assim, o usuário é redirecionado para a página que apresenta os dispositivo cadastrados no grupo pesquisado, apresentando os seus respectivos endereços IP e portas utilizadas para executar o módulo

RSIN, conforme pode ser visualizado na Figura A.6.

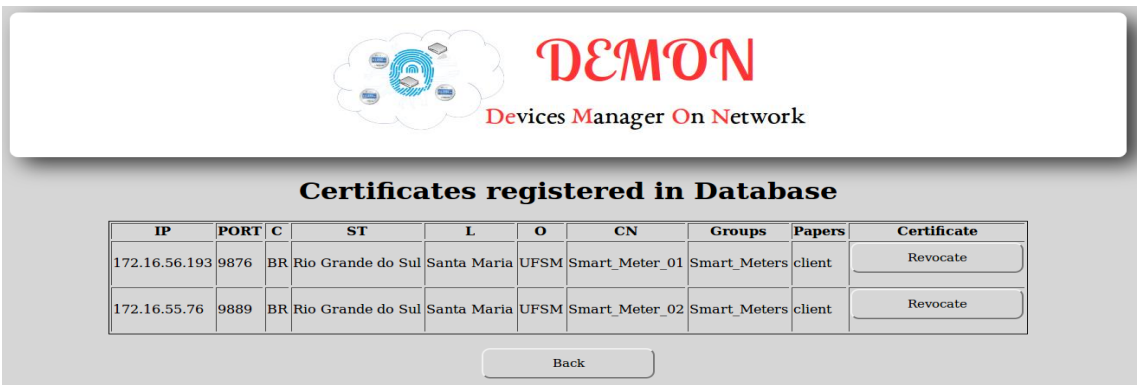
Figura A.6: Página que exibe os grupos que um dispositivo participa.



Fonte: Acervo Pessoal.

Além de permitir a realização de consultas referentes ao relacionamento entre dispositivos e grupos, a interface web da aplicação DEMON *server* possibilita listar todos os certificados emitidos, ou seja, gerados a partir das requisições provindas dos clientes (dispositivos que utilizam a aplicação DEMON *client*). Para visualizar essas informações, o operador deve clicar no botão *List Certificate*s na página principal da interface web. Ao realizar essa ação, ele é direcionado para a página que apresenta os detalhes sobre os certificados emitidos, conforme pode ser visualizado na Figura A.7.

Figura A.7: Página que exibe os certificados emitidos.



Fonte: Acervo Pessoal.