UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
MESTRADO EM ENGENHARIA ELÉTRICA

HUGO FERNANDO MAGALHÃES DE FIGUEIREDO

ARQUITETURA SCADA PARA SISTEMAS DE GERAÇÃO DISTRIBUÍDA BASEADA EM GATEWAY PARA COMUNICAÇÃO DNP3 SEGURA

DISSERTAÇÃO DE MESTRADO

PATO BRANCO

2021

HUGO FERNANDO MAGALHÃES DE FIGUEIREDO

ARQUITETURA SCADA PARA SISTEMAS DE GERAÇÃO DISTRIBUÍDA BASEADA EM GATEWAY PARA COMUNICAÇÃO DNP3 SEGURA

SCADA ARCHITECTURE FOR GATEWAY-BASED DISTRIBUTED GENERATION SYSTEMS FOR SECURE DNP3 COMMUNICATION

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica - PPGEE, da Universidade Tecnológica Federal do Paraná - UTFPR, Campus Pato Branco, como requisito parcial para obtenção do título de Mestre em Engenharia.

Orientador: Prof. Dr. Gustavo Weber Denardin

Co-orientador: Prof. Dr. Jean Patric da Costa

PATO BRANCO

2021

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Pato Branco

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

HUGO FERNANDO MAGALHAES DE FIGUEIREDO

ARQUITETURA SCADA PARA SISTEMAS DE GERAÇÃO DISTRIBUÍDA BASEADA EM GATEWAY PARA COMUNICAÇÃO DNP3 SEGURA

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Sistemas E Processamento De Energia.

Data de aprovação: 09 de Abril de 2021

Prof Gustavo Weber Denardin, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Carlos Henrique Barriquello, Doutorado - Universidade Federal de Santa Maria (Ufsm)

Prof Emerson Giovani Carati, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Giovanni Alfredo Guarneri, Doutorado - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 30/06/2021.

"Você é o único representante do seu sonho na face da terra".

Leandro Roque de Oliveira

AGRADECIMENTOS

A Deus, pela vida, saúde, e por me permitir superar todos os obstáculos encontrados durante a realização deste trabalho.

Ao meu orientador, professor Gustavo Weber Denardin, por todo suporte direcionado à execução do presente trabalho, bem como ao conhecimento repassado durante todo esse período.

À minha família, pelo suporte, carinho, dedicação e incentivo durante todo esse período.

A todos os professores do PPGEE pelos conhecimentos técnicos transmitidos ao longo do programa de pós-graduação e do projeto de pesquisa e desenvolvimento "PD 2866-0468/2017 - Gerador e inversor inteligente para conexão de sistemas fotovoltaicos em geração distribuída de energia".

Aos colegas do PPGEE, pelas discussões técnicas e momentos de descontração, em especial ao colega Matheus Kowalczuk Ferst, pela troca de experiências ao longo do desenvolvimento do setor de comunicações do projeto de P&D.

Agradeço também a ANEEL/COPEL, pelo apoio financeiro disponibilizado por meio da bolsa de estudos viabilizada pelo projeto PD 2866-0468/2017.

RESUMO

FIGUEIREDO, Hugo Fernando Magalhães. Arquitetura SCADA para sistemas de geração distribuída baseada em gateway para comunicação DNP3 segura. 2021. 117 f. Dissertação – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná, Pato Branco, PR, 2021.

Os sistemas SCADA aplicados à geração de energia passaram por várias atualizações ao longo dos anos. Especialmente no setor relacionado a redes de comunicação, o que motiva pesquisadores a elevar sua atenção a recursos capazes de prover segurança a esses sistemas. Sendo assim, este trabalho busca a ampliação da segurança de um sistema *Smart Grid* SCADA aplicando um *gateway* para comunicação segura baseada no protocolo TLS sobre a estrutura. O *gateway* proposto será capaz de fornecer a proteção adicional a um sistema *Smart Grid* SCADA composto pelo protocolo DNP3, visto que a maioria dos sistemas de supervisão utilizados nos dias atuais não possuem suporte a comunicação segura. Com essa configuração, o *gateway* está na rede da concessionária de energia. Assim, tanto o sistema SCADA da concessionária, quanto os gerenciadores de energia controlados por ela conectam-se ao *gateway*. Essa abordagem reduz a possibilidade de ataques e facilita a conexão de gerenciadores nas redes de usuários, que não possuem um IP externo de fácil acesso. Finalmente, nessa proposta a comunicação sem segurança fica restrita a uma conexão local, podendo ser protegida de outras formas. Testes de segurança são implementados por meio de um *plug-in* desenvolvido para o *software* Ettercap. Os resultados experimentais comprovam a eficiência do sistema proposto em impedir ataques do tipo *Man-In-The-Middle*.

Palavras-chave: Protocolo DNP3, Geração Distribuída, SCADA, Segurança, Redes Inteligentes.

ABSTRACT

FIGUEIREDO, Hugo Fernando Magalhães. SCADA architecture for gateway-based distributed generation systems for secure DNP3 communication. 2021. 117 p. Master's Dissertation – Post-graduate Program in Electrical Engineering, Universidade Tecnológica Federal do Paraná, Pato Branco, PR, 2021.

SCADA systems applied to power generation have undergone several updates over the years. Mainly in sectors related to communication networks, which motivates researchers to raise their attention to resources that can provide security to these systems. Therefore, this work seeks to increase the security of a SCADA Smart Grid system by applying a secure communication gateway based on the TLS protocol. The proposed gateway can provide additional protection to a SCADA Smart Grid system composed of the DNP3 protocol since most of the supervisory systems used today do not support data security coming from communication networks. With this configuration, the gateway is on the utility distribution company network. Thus, the utility's SCADA system and the energy managers controlled by it connect to the gateway. This approach reduces the possibility of attacks and provides a simple connection of managers on user networks, which do not have an accessible external IP. Finally, in this proposal, unsecured communication is restricted to a local connection and protected in other ways. The security tests were implemented by a plug-in developed for the Ettercap software. The experimental results prove the efficiency of the proposed system in preventing Man-In-The-Middle attacks.

Keywords: DNP3 Protocol, Distributed Generation, SCADA, Security, Smart Grids.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

GD	Geração Distribuída
SCADA	*Supervisory Control and Data Acquisition* (Sistema de Supervisão e Aquisição de Dados)
IHM	Interface Homem-Máquina
RTU	*Remote Terminal Unit* (Unidade Terminal Remota)
DER	*Distributed Energy Resources* (Recursos Energéticos Distribuídos)
SEP2.0	*Smart Energy Profile 2* (Perfil de Energia Inteligente 2)
VPN	*Virtual Private Networks* (Rede Privada Virtual)
CLP	Controlador Lógico Programável
SGE	Sistema de Gestão de Energia
DNP3	*Distributed Network Protocol Version* 3 (Protocolo de Rede Distribuída Versão 3)
MITM	*Man-In-The-Middle* (Homem-no-Meio)
DoS	*Denial of Service* (Negação de Serviço)
DDoS	*Distributed Denial of Service* (Negação de Serviço Distribuído)
FDI	*False Data Injection* (Injeção de Dados Falsos)
IEC	*International Electrotechnical Commission* (Comissão Eletrotécnica Internacional)
Module-OT	*Module for Operational Technology* (Módulo para Tecnologia Operacional)
SEDEA	*State Estimation-Based Dynamic Encryption and Authentication* (Criptografia e Autenticação Dinâmicas Baseadas em Estimativa de Estado)
PKI	*Public Key Infrastructure* (Infraestrutura de Chave Pública)
NAT	*Network Address Translation* (Tradução de Endereços de Rede)
DMZ	*DeMilitarized Zone* (Zona Desmilitarizada)
RAT	*Remote Access Trojan* (Cavalo de Troia de Acesso Remoto)
ARP	*Address Resolution Protocol* (Protocolo de Resolução de Endereços)
WPAD	*Web Proxy Auto-Discovery Protocol* (Protocolo de Descoberta Automática de *Proxy* da *Web*)

DNS	*Domain Name System* (Sistema de Nomes de Domínio)
BGP	*Border Gateway Protocol* (Protocolo de Borda do *Gateway*)
OPC	*Open Platform Communication* (Plataforma Aberta de Comunicação)
TLS	*Transport Layer Security* (Segurança da Camada de Transporte)
MTU	*Master Terminal Unit* (Unidade Terminal Mestre)
UART	*Universal Asynchronous Receiver/Transmitter* (Transmissor e Receptor Assíncrono Universal)
TCP/IP	*Transmission Control Protocol/Internet Protocol* (Protocolo de Controle de Transmissão/Protocolo da Internet)
I^2C	*Inter-Integrated Circuit* (Circuito Inter-Integrado)
OSI	*Open System Interconnection* (Interconexão de Sistemas Abertos)
UDP	*User Datagram Protocol* (Protocolo de Datagramas do Usuário)
API	*Application Programming Interface* (Interface de Programação de Aplicativos)
EPA	*Enhanced Performance Architecture* (Arquitetura de Desempenho Aprimorada)
PDU	*Protocol Data Unit* (Unidade de Dados de Protocolo)
CVE	*Common Vulnerabilities and Exposures* (Vulnerabilidades e Exposições Comuns)
MAC	*Media Access Control* (Controle de Acesso ao Meio)
DHCP	*Dynamic Host Configuration Protocol* (Protocolo de Configuração Dinâmica de Endereços de Rede)
ICS	*Industrial Control Systems* (Sistemas de Controle Industrial)
TSDU	*Transport Service Data Unit* (Unidade de Dados de Serviço de Transporte)
TPDU	*Transport Protocol Data Unit* (Unidade de Dados do Protocolo de Transporte)
LSDU	*Link Service Data Unit* (Unidade de Dados do Serviço de Enlace)
ASDU	*Application Service Data Unit* (Unidade de Dados do Serviço de Aplicação)
APDU	*Application Protocol Data Unit* (Unidade de Dados do Protocolo de Aplicação)
APCI	*Application Protocol Control Information* (Informações de Controle do Protocolo de Aplicação)

IIN	*Internal Indication* (Indicação Interna)
SGDA	Sistema de Gestão de Distribuição Avançado
DNP3-SA	*Distributed Network Protocol Version 3 Secure Authentication* (Protocolo de Rede Distribuída Versão 3 - Autenticação Segura)
DNPSec	*Distributed Network Protocol Version 3 Security Framework* (Estrutura de Segurança para o Protocolo de Rede Distribuída Versão 3)
EPRI	*Electric Power Research Institute* (Instituto de Pesquisa de Energia Elétrica)
TC57	*Technical Committee Number* 57 (Comitê Técnico 57)
DIFG	*Photovoltaic Inverter Data Identification Focus Group* (Grupo Voltado a Identificação de Dados de Inversores Fotovoltaicos)
SCL	*Substation Configuration description Language* (Linguagem de Configuração de Subestações)
LN	*Logical Node* (Nó Lógico)
PICOM	*Piece of Information for COMmunication* (Parte de Informação para a Comunicação)
SAS	Sistema de Automação de Subestações
LD	*Logical Devices* (Dispositivos Lógicos)
CDC	*Common Data Class* (Classe Comum de Dados)
UML	*Unified Modeling Language* (Linguagem de Modelagem Unificada)
FTP	*File Transfer Protocol* (Protocolo de Transferência de Arquivos)
SMTP	*Simple Mail Transfer Protocol* (Protocolo de Transferência de Correio Simples)
HTTP	*HyperText Transfer Protocol* (Protocolo de Transferência de Hipertexto)
POP3	*Post Office Protocol Version 3* (Protocolo de Correio Versão 3)
LDAP	*Lightweight Directory Access Protocol* (Protocolo de Acesso a Diretórios Leves)
IMAP	*Internet Message Access Protocol* (Protocolo de Acesso à Mensagem da Internet)
HTTPS	*HyperText Transfer Protocol Secure* (Protocolo de Transferência de Hipertexto Seguro)
CROB	*Control Relay Output Block* (Bloco de Saída do Relé de Controle)
JSON	*JavaScript Object Notation* (Notação de Objetos *JavaScript*)
CRC	*Cyclic Redundancy Check* (Verificação Cíclica de Redundância)

SUMÁRIO

1 INTRODUÇÃO

A demanda por fontes de energia capazes de fornecer benefícios superiores aos meios tradicionais incentivou a busca por métodos de geração de energia elétrica alternativos. Os métodos que surgiram podem ser implantados nos mais diversos locais, o que torna o sistema de geração distribuída (GD) um dos tópicos de pesquisa mais interessantes nos últimos anos (HUNG *et al.*, 2013).

No Brasil a principal tecnologia de GD difundida é a fotovoltaica, em virtude das diversas vantagens que possui, tais como o custo/benefício e a baixa manutenção. Além disso, o país conta com um clima subtropical, o que permite com que a geração de energia por meio dessa fonte seja realizável praticamente o ano todo (FIGUEIREDO *et al.*, 2018). A geração fotovoltaica é uma GD produzida por meio de recursos renováveis que tem recebido muita atenção na literatura. Além disso, a energia fotovoltaica encontra-se amplamente difundida no mercado, com diversas soluções aplicadas aos setores residenciais, industriais e comerciais (TOLEDO *et al.*, 2010).

O setor fotovoltaico, no entanto, está focado em melhorar e expandir a qualidade da rede elétrica. Nesse contexto, um dos principais tópicos de interesse refere-se ao sistema de supervisão e aquisição de dados, do acrônimo SCADA da literatura inglesa (*Supervisory Control and Data Acquisiton*). Esses sistemas podem fornecer o monitoramento e a supervisão de dispositivos e variáveis associadas ao sistema de energia, bem como implementar a tomada de decisões com base na leitura e gravação dos dados da rede (ALMALAWI *et al.*, 2016).

Com o crescimento da utilização de sistemas SCADA, foi necessário definir um padrão na troca de informações entre equipamentos da estrutura. Com isso surgiu o padrão IEEE 1815, também denominado protocolo DNP3 (*Distributed Network Protocol Version* 3), desenvolvido especialmente para trabalhar em aplicações industriais, especificamente na aquisição de dados e controle de dispositivos. Além disso, o DNP3 é composto por benefícios como a escalabilidade, confiabilidade e interoperabilidade de dispositivos voltados a segmentos de Redes Elétricas Inteligentes (Ramalho *et al.*, 2013).

O protocolo DNP3 não é a única solução desenvolvida para o controle de informações em sistemas de comunicação. No entanto, é um protocolo amplamente utilizado nas companhias elétricas para automações de subestações (GUIMARAES *et al.*, 2013). Além disso, o protocolo é utilizado na comunicação serial e TCP/IP (MAJDALAWIEH *et al.*, 2007).

A indústria de recursos energéticos distribuídos (DER, do inglês *Distributed Energy Resources*), principalmente a de geração de energia fotovoltaica, tem feito progressos consideráveis ao longo dos anos e com isso procedimentos tornaram-se essenciais como a tarifação dinâmica e a necessidade do controle da energia gerada. Desse modo, a conexão do sistema à Internet tornou-se fundamental para atender essas demandas, além de fornecer muitas vantagens, como a economia sobre a infraestrutura de rede, por exemplo.

Essa conexão permitiu o acesso remoto a inversores e outros dispositivos que fazem parte de um sistema de geração solar, permitindo assim que agentes fisicamente desconectados de várias plantas DER sejam capazes de interação, como o monitoramento e o controle necessário. No entanto, a exposição de sistemas de energia conectados na Internet expôs componentes presentes no sistema SCADA a diversas vulnerabilidades (Yang *et al.*, 2014), o que induziu a implementação de ferramentas de segurança para a proteção desses equipamentos. Um dos equipamentos regulares pertencentes a esses sistema é o inversor, que atualmente conta com vários protocolos de segurança implementados, como o SEP2.0 (*Smart Energy Profile 2*) (SGIP, 2015), baseado em HTTPS (*HyperText Transfer Protocol Secure*).

O protocolo SEP2.0 é indicado a sistemas de gerenciamento de energia de instalações e agregados, nos quais a instituição não esteja efetuando o monitoramento e controle direto de sistemas DER (SGIP, 2015). Entretanto, no setor relacionado a estruturas DER, em que o controle e monitoramento do SCADA seja efetuado pela concessionária é recomendado o uso do protocolo DNP3 (SGIP, 2015).

O DNP3 possui um formato de autenticação padrão (CREMERS *et al.*, 2019). No entanto, essa autenticação tem uma confiabilidade reduzida em comparação ao TLS (ROSBOROUGH *et al.*, 2019). Sendo assim, a falta de confidencialidade nos dados pode facilitar um ataque elaborado, que são os mais frequentes em sistemas SCADA, como será apresentado no decorrer dessa dissertação.

A manipulação de dados exclusivos de sistemas por meio de ameaças cibernéticas tornou-se um evento frequente e preocupante (CHALAMASETTY; MEMBER, 2016). Por exemplo, se um invasor obtiver o controle total da rede de comunicação de um sistema SCADA, todas as informações corporativas privadas estarão prontamente disponíveis para esse agente externo (SRIDHAR; MANIMARAN, 2010). Além disso, o invasor pode interferir no serviço fornecido, tornando-o instável ou indisponível.

Invasões em sistemas SCADA compostos por canais inseguros atingiram níveis consideráveis nos últimos anos (ALMALAWI *et al.*, 2016). Em 2011 foi realizado um relatório provindo da companhia de proteção cibernética McAfee, listando que entre as 200 estruturas analisadas no processo de avaliação de segurança, a grande maioria obteve invasões externas maliciosas (ZHU *et al.*, 2011). Além disso, durante a 3ª Conferência Latino-Americana de Segurança em Sistemas SCADA realizada em 2018, foi declarado que, segundo entrevistas feitas no Brasil, 55,27% das empresas não apresentam proteção contra invasões, ou contam com essa segurança, porém não atualizam periodicamente sua base de dados.

Os distúrbios gerados por meio de invasores em estruturas SCADA refletem diretamente na confiabilidade do sistema, no processo executado pelo *software* e na precisão do sistema (HILAL; NANGIM, 2017). Portanto, para evitar tais falhas de segurança, é recomendável implementar modelos de proteção em redes compostas por essa tecnologia.

Um ataque específico foi considerado o marco para a segurança das informações nos sistemas SCADA, a descoberta do *worm* Stuxnet em 2010 (LANGNER, 2013). O principal alvo do Stuxnet foram Controladores Lógicos Programáveis (CLPs), dispositivos que permitem a automação de processos eletromecânicos, como controlar máquinas e linhas industriais. Esse atacante tornou-se conhecido pela invasão do sistema de controle de uma centrífuga de enriquecimento de urânio no Irã, causando danos substanciais ao programa nuclear do país (CHEN; ABU-NIMEH, 2011). O *worm* Stuxnet não foi destinado exclusivamente a sistemas SCADA (LANGNER, 2011). No entanto, após o ataque do Irã em que o sistema afetado tinha uma arquitetura SCADA incorporada, ficou claro que tais sistemas são suscetíveis a invasões. Portanto, esse evento teve uma contribuição considerável no estímulo de buscas por fragilidades em sistemas SCADA.

Após o ataque do Stuxnet em 2010, os ataques cibernéticos direcionados a sistemas SCADA cresceram significativamente. Nos últimos anos, surgiu outro ataque relevante na Ucrânia, que foi denominado como BlackEnergy3 (CASE, 2016). No ataque

realizado na Ucrânia, os invasores usando e-mails de *spear-phishing* exploraram dados para sequestrar o Sistema de Gestão de Energia (SGE) SCADA (PULTAROVA, 2016), de modo a causar danos consideráveis, como por exemplo, quedas de energia (CASE, 2016).

O crescimento de GDs conectados à rede no sistema de energia elétrica convencional tem proporcionado uma variedade considerável de novos problemas, como falsas detecções de ilhamento, elevação de tensão, sobrecarga de dispositivos de rede, entre outros (SALEEM; CARTER, 2019). Esses problemas podem tomar proporções consideráveis quando os sistemas são submetidos a conexão com a Internet, devido a ataques cibernéticos.

Um sistema conectado à Internet, além de ter vulnerabilidades a eventos maliciosos, conhecidos como *malware*, *spyware* e vírus de computador, também é suscetível a outros possíveis ataques (MAGLARAS *et al.*, 2016). Essas ameaças podem ocorrer de diversas formas, tais como, *Man-In-The-Middle* (MITM), *Denial of Service* (DoS), *replay*, *False Data Injection* (FDI), entre outros (SALEEM; CARTER, 2019).

O avanço de ameaças em sistemas SCADA estimulou estudos específicos para aumentar a segurança dessas arquiteturas, principalmente quando compostas com o protocolo DNP3. Com esse intuito, surgiram discussões sobre duas possíveis soluções: aplicação de tecnologias criptográficas em ambas as extremidades do meio de comunicação ou melhorias de segurança aplicadas diretamente no protocolo (ROSBOROUGH *et al.*, 2019).

Uma das possíveis soluções para aumentar a segurança do protocolo DNP3 é a adição de segurança por meio do protocolo TLS (*Transport Layer Security*), como é sugerido pela empresa Automatak, detentora da biblioteca `OpenDNP3` (CRAIN, 2013). Tal protocolo é largamente utilizado em uma grande variedade de protocolos na Internet, sendo sua aplicação mais conhecida o HTTPS (LEVILLAIN *et al.*, 2015).

O TLS é um protocolo da camada de transporte que fornece comunicações seguras em conexões com a Internet (OPPLIGER, 2016). A principal finalidade do TLS é proteger os níveis inferiores de uma rede, disponibilizando integridade e privacidade aos dados trocados (DU *et al.*, 2001). Além da segurança na criptografia, o TLS também objetiva a flexibilidade, interoperabilidade e eficiência no uso de recursos (RISTIC, 2013).

A criptografia não irá impedir que um invasor tenha acesso aos dados criptografados do sistema, porém, caso consiga ele não poderá descriptografá-los ou alterá-los

(RISTIC, 2013). No entanto, para evitar ataques em que o atacante busca se passar pela vítima, o TLS conta com uma tecnologia chamada PKI (*Public Key Infrastructure*), que assegura que o tráfego chegue ao real destinatário (RISTIC, 2013). Portanto, a utilização do TLS é capaz de melhorar a proteção de um sistema SCADA utilizando o protocolo DNP3.

O presente trabalho propõe a amplificação de segurança de um sistema SCADA por meio do uso do protocolo TLS com autenticação de identidade nos canais de comunicação DNP3, com o uso de um *gateway* para a interface de dados internos DNP3 sem o uso de TLS a uma conexão externa DNP3 sob TLS. O protocolo TLS foi desenvolvido com o intuito de viabilizar uma comunicação segura em ambas as extremidades de uma conexão. A vantagem da inclusão do TLS é que ele age como uma subcamada do protocolo DNP3. Desse modo, os dados são transmitidos normalmente, a única diferença é que estarão encriptados com o protocolo TLS, permitindo assim uma melhoria na estrutura de segurança da informação.

Outra vantagem de se utilizar um *gateway* é a diminuição da superfície de ataque. No caso de uma concessionária de energia comandando inversores inteligentes por DNP3, espera-se que o *software* supervisório da concessionária seja o mestre (cliente) e os inversores como *outstations* (servidores). Ao utilizar o *gateway*, a presente proposta considera que tanto o supervisório, quanto os inversores se conectarão ao *gateway*. Assim, a análise de segurança será concentrada no *gateway* da concessionária e não nos inversores de clientes, o que irá diminuir consideravelmente a superfície de ataque.

Finalmente, tal abordagem fornece uma vantagem adicional no que diz respeito ao sistema de tradução de endereços (NAT, do inglês *Network Address Translation*). A Tradução de Endereço de Rede é um processo específico que envolve remapear um único endereço IP em outro endereço IP, alterando as informações de rede e informações de endereço encontradas no cabeçalho IP dos pacotes de dados. As redes locais têm vários endereços IP privados que pertencem a dispositivos específicos na rede. Por meio de um sistema NAT, esses endereços privados são traduzidos em um endereço IP público quando são enviadas solicitações de saída dos dispositivos de rede para a Internet. Ainda, o NAT inicialmente só permite conexões de dentro da rede do cliente para a Internet, sendo necessário a configuração de um redirecionamento de portas ou DMZ (*DeMilitarized Zone*) caso seja necessário criar uma conexão da Internet (concessionária) para o inversor. Nesse sentido o *gateway* também reduz a complexidade da configuração no lado do cliente. Usualmente o cliente tem menor competência técnica e é mais passível a erro

nessa configuração, bem como seriam necessárias mais configurações de clientes (inversores) do que de servidores na concessionária. Com a abordagem proposta, ou seja, a inversão da conexão com a adição do *gateway*, os inversores inteligentes, que geralmente tem IPs privados na rede do cliente, irão se conectar com o *gateway* da concessionária. Assim, somente o *gateway* presente na rede da concessionária possuirá um IP público e disponibilizará a porta para conexão DNP3 a Internet.

Como o trabalho é baseado em sistemas *Smart Grid* é necessário também adequar o protocolo DNP3 ao padrão IEC 61850 (GROUP, 2018). O IEC 61850 foi disponibilizado em 2004 pelo *Technical Committee Number* 57 (TC57) da *International Electrotechnical Commission* (IEC) e refere-se a um padrão internacional que define protocolos de comunicação para dispositivos inteligentes em sistemas de energia elétrica. Alguns dos principais objetivos do padrão são estabelecer serviços básicos para transferência de dados, interoperabilidade e definir um modelo único para armazenamento de dados.

1.1 OBJETIVOS

1.1.1 Objetivo geral

Os principais objetivos do presente trabalho são adicionar segurança e diminuir a superfície de ataque de canais de comunicação de um sistema SCADA. Sendo assim, será desenvolvido um *gateway* capaz de efetuar troca de informações pelo protocolo DNP3 sem segurança a um canal de comunicação DNP3 sob TLS e ainda, capaz de proporcionar com que as extremidades da conexão, mestre e *outstation* se conectem diretamente ao *gateway*.

1.1.2 Objetivos específicos

Com o propósito de atingir o objetivo geral delimitado, os objetivos específicos baseiam-se em:

- Montar um sistema supervisório em uma arquitetura SCADA para Redes Elétricas Inteligentes.
- Implementar a comunicação DNP3 externa (*outstation* do sistema), composta pelo protocolo TLS.

- Implementar um *gateway* capaz de prover a segurança complementar na troca de dados com o *software* SCADA.
- Implementar a conexão DNP3 adequada aos perfis recomendados pelo padrão IEC 61850.
- Implementar um *software* capaz de realizar ataques a estruturas DNP3 SCADA para efetuar testes de segurança do sistema elaborado.

1.2 ORGANIZAÇÃO DO TRABALHO

No Capítulo 2 é realizada a revisão da literatura, na qual são abordados os principais tópicos necessários no desenvolvimento do trabalho. Inicialmente, é explicado como uma estrutura padrão é composta, discutindo brevemente sobre cada componente do sistema, contextualizando o tema para a área principal do trabalho: DNP3 e *Smart Grids*. Posteriormente, são discutidos os protocolos e elementos de segurança aplicados ao trabalho.

No Capítulo 3 são apresentadas as etapas de desenvolvimento da solução proposta. Sendo assim, a princípio é apresentado um diagrama de blocos contendo todos os itens necessários para o desenvolvimento da estrutura de geração distribuída. Em seguida, é comentado sobre os métodos utilizados para implementar cada etapa do trabalho. As etapas do trabalho são definidas pelos dispositivos utilizados na estrutura, como o sistema mestre, sub-mestre, *outstation* e o atacante, responsável por efetuar a validação da estrutura.

No Capítulo 4 são apresentados os resultados obtidos sobre a segurança do sistema desenvolvido. Inicialmente, é realizada uma análise no mesmo sistema de geração distribuída sem segurança para validação do sistema atacante desenvolvido. Em seguida, são apresentados todos os resultados obtidos após submeter uma estrutura SCADA ao sistema de segurança desenvolvido no trabalho.

No Capítulo 5 é apresentada a conclusão do trabalho. Portanto, esse Capítulo baseia-se em apresentar todo o desfecho sobre as etapas e estudos aplicados ao trabalho. O que permite uma discussão breve sobre o funcionamento do sistema, bem como a viabilidade e necessidade de uma estrutura de segurança aplicada a sistemas de geração de energia. Além disso, são apresentadas algumas sugestões para trabalhos futuros capazes

de ampliar a segurança de um sistema composto pela estrutura presente nesse trabalho. Em seguida, são apresentados os artigos publicados baseados no trabalho desenvolvido.

2 REVISÃO DA LITERATURA

No decorrer deste capítulo serão abordados os temas relevantes para o entendimento do trabalho desenvolvido. Dessa forma, serão revisados os principais componentes utilizados para desenvolver o trabalho, bem como incidentes e formatos de ataques voltados a esses sistemas, para compreensão de como funcionará a estrutura de segurança desenvolvida. Além disso, será realizada a revisão da literatura sobre autenticação e criptografia, formatos de segurança empregados para atender a solução proposta.

2.1 SCADA

O sistema de supervisão e aquisição de dados refere-se a uma estrutura capaz de monitorar e controlar plantas de diversos processos distintos (BOYER, 2009). Os principais setores que utilizam essa tecnologia são: sistemas de geração de energia elétrica, indústrias de petróleo ou gás, sistemas de irrigação, entre outros processos (KRUTZ, 2006). Esse tipo de arquitetura originou-se em sistemas isolados e concentrados, atuando somente em estruturas pequenas (BOYER, 2009).

Com o avanço de diversas tecnologias como sistemas de comunicação, melhorias e redução de valores de processadores, os sistemas SCADA obtiveram um avanço tecnológico considerável proporcionando as funcionalidades comuns disponibilizadas nesses sistemas atualmente. Além disso, com a inovação tecnológica as arquiteturas SCADA passaram a contar com estações de controle e dispositivos remotos (KRUTZ, 2006), tornando necessário a conexão dessas estruturas com a Internet para atender completamente essa melhoria. Sendo assim, diversos setores adotaram essas medidas, como é o caso de sistemas de energia elétrica.

Sistemas de energia elétrica também evoluíram e adaptaram-se à realidade e necessidade atual, o que possibilitou o desenvolvimento do conceito denominado *Smart Grid*, ou Rede Elétrica Inteligente. Esse formato é voltado a eficiência e sustentabilidade de energia realizado por meio dos benefícios provenientes da integração da tecnologia da informação a sistemas de geração, transmissão e distribuição.

As arquiteturas SCADA voltadas a *Smart Grids* são aplicadas ao controle, supervisionamento e gerenciamento de processos de geração, distribuição e transmissão de energia elétrica (GIANI *et al.*, 2011). Além disso, o uso de estruturas SCADA disponibilizam diversos benefícios, dentre eles a análise de falhas, a verificação sobre o consumo e demanda de energia, a balanço da carga utilizada pelos consumidores, a medição inteligente, a análise de carga nos transformadores, dentre outros (LOPES *et al.*, 2012).

Um diagrama representando os componentes de uma estrutura SCADA genérica desenvolvida para sistemas de GD é apresentado na Figura 1. Essa estrutura é composta por diversos elementos fundamentais, como IHM (Interface Homem-Máquina), MTU (*Master Terminal Unit*), rede de comunicação e RTU's (KRUTZ, 2006). O centro de controle é a base do sistema SCADA. No centro de controle é realizado o monitoramento e a operação do sistema. Normalmente, esse nível é composto por uma IHM e estações de trabalho. A IHM refere-se a uma aplicação demonstrada em tela, que simplifica o processo para o operador do sistema, tornando eficaz a compreensão de informações repassadas entre humano e máquina. Entretanto, nas estações de trabalho são realizados comandos de operação da planta e análises pelo operador em tempo real (BARBOSA *et al.*, 2012). A MTU é a unidade responsável por extrair, memorizar e processar os dados obtidos nas RTUs, a fim de posteriormente traduzir essas informações para serem utilizadas no contexto da estrutura. A RTU tem a responsabilidade de coletar e controlar informações na planta por meio de entradas e saídas analógicas e digitais compostas em dispositivos posicionados na planta. A rede de comunicação é responsável por transmitir as informações compostas na arquitetura SCADA de maneira rápida e precisa. Geralmente, o monitoramento e operação dos sistemas SCADA de GD é de responsabilidade das concessionárias de energia elétrica.

A RTU (*Remote Terminal Unit*) exibida na Fig. 1 é responsável, nesse contexto, por fornecer as informações de subestações, usinas de geração de energia e linhas de transmissão de energia. A RTU refere-se a um dispositivo eletrônico controlado por microprocessador que é capaz de conectar componentes a um sistema SCADA por meio de telemetria, com o uso de meios de conexão próprios para cada sistema. Além disso, esse elemento possui a capacidade de transferência dos dados auferidos para um sistema mestre na estrutura.

A rede de comunicação mudou consideravelmente ao longos dos anos. Inicialmente utilizava-se barramento de comunicação serial e os protocolos I^2C, UART, RS-232, RS-485, entre outros. Posteriormente com a necessidade de comunicações remotas surgiu

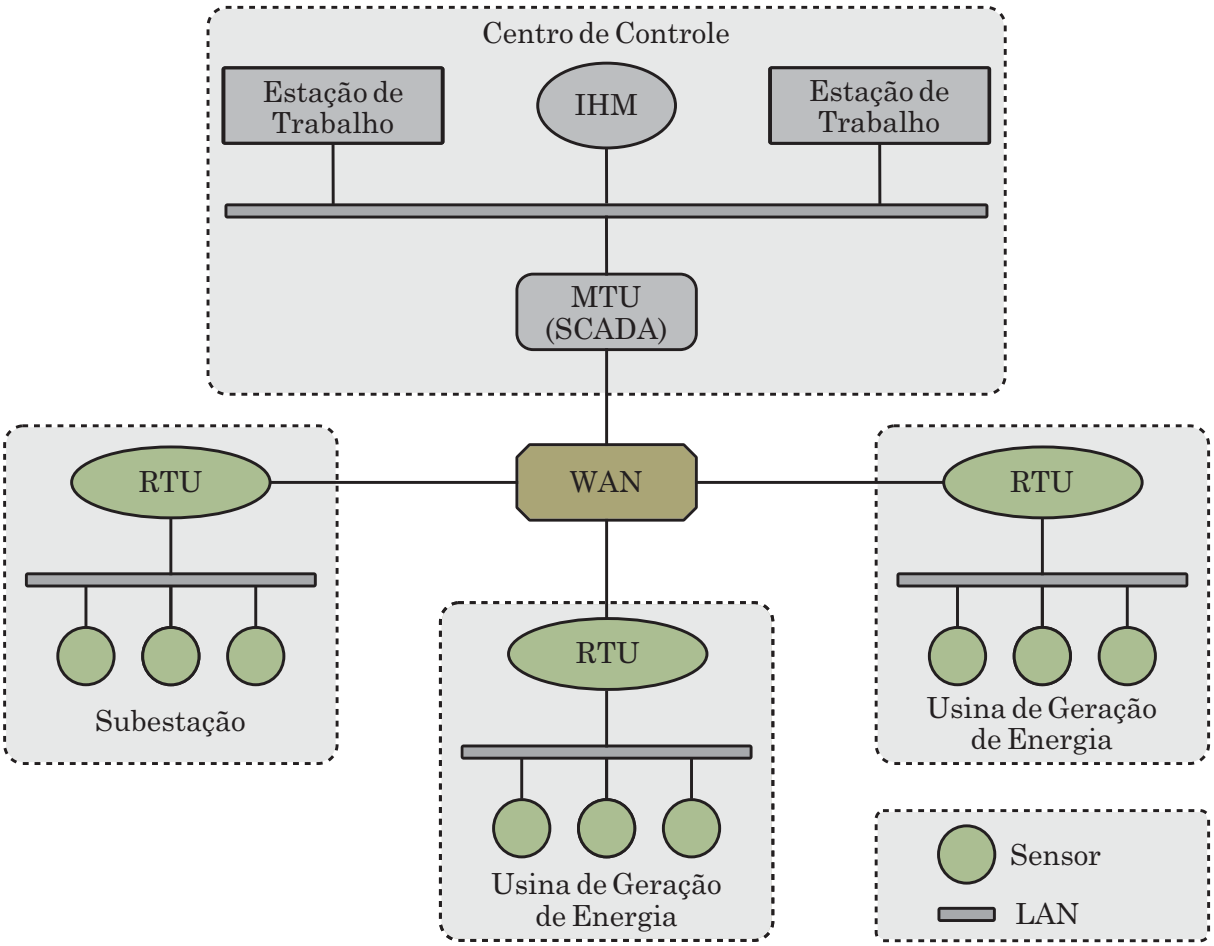

Figura 1: Arquitetura SCADA genérica para sistemas de geração distribuída.

Fonte: O Autor (2021).

a Internet e com isso a aplicação de outros protocolos, como o TCP (*Transmission Control Protocol*) e UDP (*User Datagram Protocol*), por exemplo.

Em sistemas de geração de energia elétrica, inicialmente utilizava-se protocolos proprietários como o DeviceNet e o Profibus, por exemplo. Entretanto, apesar desses protocolos ainda possuírem aplicações no mercado atual, a tendência de sistemas *Smart Grids* proporcionou um foco significativo no uso de protocolos abertos, como o Ethernet TCP/IP, Modbus TCP, DNP3, entre outros.

O protocolo DNP3, por exemplo, é composto por funções baseadas no modelo *Enhanced Performance Architecture* (EPA). No modelo DNP3 utiliza-se as camadas: aplicação, pseudo-transporte e enlace. Essa representação diminui o número de camadas, bem como o tamanho do *Protocol Data Unit* (da sigla em inglês, PDU) em cada nível. Sendo assim, esse modelo colabora, enviando e recebendo pequenas quantidades de dados com eficácia (FARUK, 2008).

Inicialmente a IEC apresentou o padrão IEC 870 para transmissão de dados de telemetria em estruturas SCADA. Essa recomendação foi baseada no modelo composto por sete camadas de funções elaborado para padronizar a troca de informações entre dois sistemas, denominado modelo OSI (*Open Systems Interconnection*) (EAST *et al.*, 2009). No entanto, logo em seguida foi desenvolvida a arquitetura EPA, definido pela IEC, especificamente para redes de sistemas SCADA. Esse modelo constava com apenas três camadas, pensando em reduzir a sobrecarga do modelo OSI de maneira adequada para atender sistemas SCADA.

Após algum tempo percebeu-se que o modelo EPA não suportava mensagens da camada de aplicação superiores ao comprimento máximo de um quadro da camada de enlace. Dessa forma, durante o desenvolvimento do modelo DNP3 esse problema foi resolvido, devido a permissão de fragmentação de mensagens na camada incluída, denominada pseudo-transporte (EAST *et al.*, 2009).

Outro conjunto de protocolos de comunicação amplamente utilizado em transmissão de informações para estruturas SCADA é o modelo TCP/IP. Esse formato origina-se da junção dos protocolos TCP e IP. Além disso, é composto por quatro camadas, aplicação, transporte, internet e interface de rede. A Figura 2 apresenta um comparativo entre a estrutura dos modelos comentados anteriormente OSI, TCP/IP, EPA e DNP3.

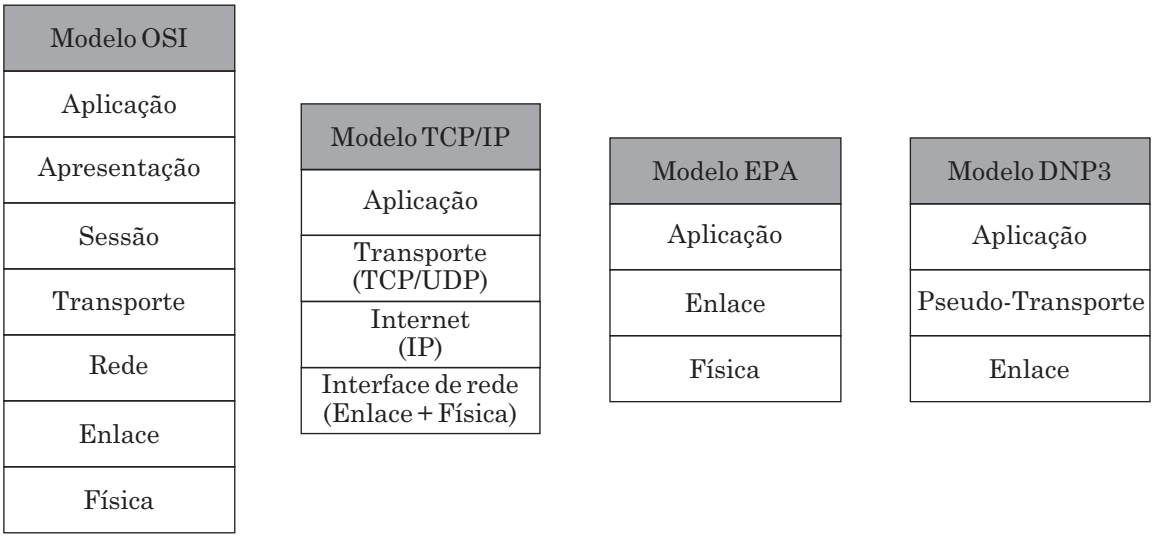

Figura 2: Camadas dos modelos OSI, TCP/IP, EPA e DNP3.

Fonte: O Autor (2021).

O modelo DNP3 não prevê a camada física, no entanto, é necessário aplicar o modelo sob uma camada física para obter a transmissão dos dados. Na concepção

inicial do modelo DNP3, o mesmo era baseado conforme demonstra a Figura 2. Porém, atualmente o DNP3 funciona sob o TCP/IP.

O modelo DNP3 é amplamente utilizado no gerenciamento de subestações de estruturas *Smart Grid*, com redes protegidas por VPN, o que permite uma preocupação reduzida em relação a segurança. No entanto, com a necessidade dos dados provenientes de clientes da concessionária a segurança tornou-se algo fundamental.

Estruturas SCADA voltadas a *Smart Grids* incorporam um índice elevado de sensoriamento, comunicação e controle distribuído em comparação a sistemas legados (GIANI *et al.*, 2011). Inicialmente, os sistemas SCADA possuíam uma comunicação local com barramentos dedicados e protocolos específicos, que continham uma segurança frágil, porém raramente explorada. Com o passar do tempo, protocolos baseados em TCP/IP se tornaram comuns em estruturas SCADA, o que possibilitou o aumento de vulnerabilidades nesses sistemas. Como o protocolo Modbus por exemplo, que inicialmente tinha como base as interfaces RS-232 e RS-485 e que devido a necessidade da ampliação da velocidade das taxas de transmissão e da conexão de dispositivos remotos passou a suportar também o protocolo TCP/IP. Porém, com o suporte ao TCP/IP, a vulnerabilidade a ataques maliciosos aumentou consideravelmente (KRUTZ, 2006).

Uma das principais limitações do protocolo IP no quesito segurança refere-se a sua ineptidão em autenticar um dispositivo na rede, ou seja, utilizando a informação do IP de origem é praticamente impossível determinar a identidade correta do respectivo dispositivo (POUW, 1999). Além disso, a garantia de que o conteúdo recebido não apresente alteração da sua versão de origem é insatisfatória, tal como a afirmação de que a privacidade das informações tenham sido mantidas (POUW, 1999).

2.2 INCIDENTES

Incidentes a estruturas SCADA que utilizam do protocolo TCP/IP têm se tornado comuns nos últimos anos. Sendo assim, diversos tipos de ataques distintos têm se mostrado eficientes em invasões a esses tipos de sistema e com isso *hackers* motivados por ideologias próprias ou financiados por empresas ou organizações criminosas têm concentrado esforços para danificar, inviabilizar e comprometer sistemas SCADA ao redor do mundo.

Os ataques focados em exploração de falhas do protocolo IP atuam simulando um dispositivo real, com intuito de coletar dados privados e explorar a confiança que os

dispositivos têm entre si (POUW, 1999). Além disso, os ataques também focalizam em intermediar as informações trocadas entre dispositivos e fazer alterações nos dados. Dessa forma, diversas invasões foram realizadas, como as listadas a seguir, que ocorreram em incidentes mundialmente conhecidos atingindo estruturas SCADA de sistemas de energia.

2.2.1 Stuxnet

O Stuxnet foi um *worm*, descoberto em junho de 2010, que ficou conhecido por atingir instalações nucleares em Natanz no Irã (MATROSOV *et al.*, 2010). Esse *worm* foi responsável por infectar mais de 60000 computadores em países como a Índia, China, Indonésia, Coréia do Sul, Azerbeijão, Malásia, Estados Unidos da América, Reino Unido, Alemanha, Finlândia, Austrália e Irã, região em que foram designadas mais da metade dessas invasões (FARWELL; ROHOZINSKI, 2011).

O Stuxnet tinha a capacidade de reprogramar as lógicas de controle e ocultar todas as mudanças feitas (FALLIERE *et al.*, 2011). Este agente malicioso explorava a vulnerabilidade de controladores projetados pela fabricante SIEMENS (LANGNER, 2011). Além disso, esse *worm* utilizava técnicas modernas comparadas com as formas de ataque conhecidas e foi reconhecido como o *malware* de maior complexidade registrado até o momento de sua descoberta (LANGNER, 2011).

O *worm* possuía um tamanho considerável, 500000 *bytes*, diferentemente dos habituais que tinham de 10000 a 15000 *bytes* (ROSENBERG, 2017). Além disso, atuavam especificamente em sistemas baseados em Windows (KNAPP; LANGILL, 2015).

O Stuxnet tratava-se de uma fração de código malicioso que agia em três etapas. Inicialmente, buscava por redes e dispositivos Microsoft Windows, replicando-se incessantemente durante a contaminação dos sistemas (ROSENBERG, 2017). Em seguida, investigava a presença do *software* Siemens STEP7. Com a evidência de que o *software* estava presente na máquina avançava-se a próxima etapa que focalizava comprometer os CLPs associados ao dispositivo (ROSENBERG, 2017).

Geralmente, as funcionalidades do Windows são inicializadas de acordo com a necessidade por meio de um arquivo DLL alocado no disco rígido, se fosse realizado o mesmo procedimento com arquivos maliciosos, esses seriam descobertos por um *software* antivírus (ROSENBERG, 2017). Por esse motivo o Stuxnet alocava o arquivo DLL composto pelo código malicioso de forma descriptografada somente na memória, como uma espécie de arquivo virtual com uma denominação específica (ROSENBERG, 2017).

Posteriormente, o *worm* reprogramava a API (*Application Programming Interface*) do Windows de modo que, no instante que uma aplicação tentasse inicializar uma função composta pelo nome especialmente criado para o Stuxnet, esse arquivo seria inicializado diretamente da memória para o disco rígido (ROSENBERG, 2017). Desse modo, o *worm* elaborava uma espécie de arquivo fantasma que não ficava armazenado no disco rígido e por causa disso era praticamente impossível encontrar o agente malicioso no sistema (ROSENBERG, 2017).

O Stuxnet não buscava se expandir utilizando a Internet, diferentemente de *worms* desenvolvidos anteriormente, que possuiam esse comportamento. Esse ataque direcionava-se normalmente a sistemas que não tinham conexões com a Internet, ou como são chamados na liteturatura inglesa, *air-gapped systems*. Entretanto, esse ataque geralmente eram espalhados por dispositivos USB utilizados em máquinas pertencentes as estruturas infectadas (FALLIERE *et al.*, 2011). No entanto, a mídia removível, atuou como o mecanismo de entrega de dados para explorar a rede afetada, e posteriormente seguia-se instalando em diversas versões do Windows, repetindo as explorações e instalações até que o invasor conseguisse atingir um sistema com acesso a Internet, para que pudesse assim, transmitir as informações aos servidores C^2 (Controle e Comando) do atacante (ASSANTE; LEE, 2015).

O Stuxnet utilizou pelo menos três explorações de dia zero. A CVE (*Common Vulnerabilities and Exposures*), uma iniciativa colaborativa composta por diversas organizações voltadas a tecnologia e segurança que auxilia na listagem de vulnerabilidades e explorações e apresentou as seguintes explorações de dia zero referentes ao *worm* Stuxnet:

- CVE-2010-2568 - Executa código arbitrário quando o usuário abre uma pasta com o arquivo .LNK ou .PIF criado com códigos maliciosos.
- CVE-2010-2729 - Executa código arbitrário quando o invasor envia uma mensagem de chamada de procedimento remoto especialmente criada.
- CVE-2010-2772 - Permite que usuários locais acessem um banco de dados *back-end* e obtenham privilégios no sistema Siemens Simatic WinCC e PCS7 SCADA.

Esse *worm* foi projetado para trabalhar em vulnerabilidades que eram desconhecidas até então (FALLIERE *et al.*, 2011). O propósito principal desse invasor era deteriorar fisicamente a arquitetura-alvo. Embora esse invasor tenha sido projetado para deteriorar sistemas somente no Irã, diversos países foram atingidos de maneira significa-

tiva, como na Índia, que teve satélites comprometidos (FARWELL; ROHOZINSKI, 2011). Desse modo, esse incidente ficou conhecido como o principal ataque da história a sistemas compostos por estruturas SCADA.

2.2.2 Dragonfly/HAVEX

O Dragonfly ou HAVEX refere-se a um RAT (*Remote Access Trojan*) capaz de viabilizar completamente o controle administrativo de uma estrutura infectada (VENKATACHARY *et al.*, 2017). O Dragonfly foi considerado um ataque muito sofisticado, e sua descoberta aconteceu em 2014 (LANGILL, 2014). Esse tipo de ataque focaliza suas invasões em arquiteturas compostas com os padrões OPC (*Open Platform Communication*) (KANG *et al.*, 2015a).

Os desenvolvedores do Dragonfly direcionaram o *malware* utilizando três vetores de ataque (LANGILL, 2014). Esse RAT agiu enviando mensagens de correio eletrônico não solicitadas contendo arquivos infectados para executivos e funcionários seniores, infectando sites comumente acessados por possíveis vítimas de modo que quando o visitante efetuasse o acesso fosse redirecionado para uma página composta pelo kit de exploração e dessa forma, fosse capaz de efetuar o propósito do RAT na máquina-alvo (LANGILL, 2014).

Esse RAT continha cargas úteis concentradas em atingir elementos específicos de sistemas de controle industriais (KANG *et al.*, 2015a). O HAVEX atuava analisando o ambiente para identificar os componentes do ICS (*Industrial Control System*), coletar as informações privadas e filtrá-las e posteriormente enviá-las ao servidor C^2 externo (ASSANTE; LEE, 2015). O HAVEX agiu corrompendo sistemas em diversos países, como França, Romênia, Grécia, Alemanha, entre outros (VENKATACHARY *et al.*, 2017). No entanto, o evento tornou-se mundialmente conhecido devido a invasão a empresas de energia nos Estados Unidos (VENKATACHARY *et al.*, 2017).

Outro vetor interessante no Dragonfly foi o ataque a fornecedores de ICS, comprometendo sites de suporte. Os atacantes agiram substituindo arquivos legítimos de instalação nesses sites por *softwares* compostos por elementos maliciosos (LANGILL, 2014).

Como resultado, o mesmo conteúdo transmitido pelo Dragonfly via e-mail, agora estava presente no pacote de instalação considerado confiável, pois era oriundo de uma fonte credível, mais precisamente o próprio fabricante do ICS (LANGILL, 2014). O

mecanismo de entrega do *malware* nesse cenário era basicamente a conexão com a Internet utilizando o protocolo HTTP (ASSANTE; LEE, 2015).

Até então acreditava-se que equipamentos como CLPs e RTUs de estruturas SCADA, quando desconectados da Internet eram imunes a ataques. No entanto, esse ataque pôde demonstrar que no momento de associação de itens externos aos dispositivos ICS, mesmo quando provenientes de fornecedores confiáveis era possível a inclusão de agentes maliciosos nessas estruturas (LANGILL, 2014).

Em sistemas ICS e SCADA, o Dragonfly agia incluindo uma linha de código para executar o arquivo `mbcheck.dll`, que continha o *malware* HAVEX (CAMPBELL, 2015). O arquivo malicioso tinha o objetivo de efetuar o roubo de dados privados (LU *et al.*, 2018), ação essa que é conhecida como ciber-espionagem. A maioria das infecções descritas ocorreram no nível de aplicação, setor em que encontravam-se os sistemas de supervisão industrial, composto por estações de trabalho e plataformas IHMs (ASSANTE; LEE, 2015).

O Dragonfly não sabotou nenhum sistema de ICS apesar de possuir a capacidade de até mesmo causar danos ou perturbações no fornecimento de energia em vários países na Europa (RESPONSE, 2014). A princípio, acredita-se que o *malware* foi focado somente na coleta de informações dos sistemas afetados.

2.2.3 BlackEnergy3

Essa forma de ataque referia-se a um *malware* baseado em *plug-in* voltado a atingir estruturas críticas (KHAN *et al.*, 2016). O BlackEnergy3 fornece ao invasor uma ferramenta simples que possui a capacidade de efetuar diversos ataques utilizando sintaxe e estrutura reduzida.

Esse tipo de ataque têm sido amplamente utilizado a ocasionar danos físicos a estruturas, como na Ucrânia em dezembro de 2015 (KHAN *et al.*, 2016). No ataque realizado na Ucrânia, os invasores exploraram informações relevantes de um sistema de energia para sequestrar o SGE SCADA, executando falsas ações para causar uma queda de energia (CASE, 2016).

O BlackEnergy3 infiltrava-se nas redes por meio de um anexo de e-mail do Microsoft Word infectado e também pelo KillDisk, *malware* destrutivo que era aplicado com o intuito de deteriorar os sistemas remotos de controle e monitoramento (PULTAROVA, 2016). O *malware*, no entanto, fazia parte de apenas uma etapa do plano. Uma empresa

de cibersegurança norte-americana, denominada SANS ICS descobriu que o BlackEnergy3 era responsável por permitir que os atacantes cegassem os operadores e com isso abrissem *gateways* nos sistemas, ao mesmo tempo o KillDisk impossibilitava o acesso remoto a partir do momento que a violação era detectada (PULTAROVA, 2016).

Os atacantes utilizaram *Virtual Private Networks* (VPNs) para acesso a rede ICS (WEERATHUNGA, 2017). Com isso, exploraram as ferramentas de acesso remoto presentes no sistema para emitir ações a partir de uma IHM SCADA remota.

Os invasores utilizaram como base uma versão modificada do KillDisk com o intuito de apagar o *master boot record* das estruturas, bem como os logs das organizações afetadas (WEERATHUNGA, 2017). Durante o ataque foi também alterado a *firmware* dos conversores Seriais-Ethernet, que faziam a conversão do protocolo IEC60870-5 101 para o IEC60870-5 104 (WEERATHUNGA, 2017).

A interrupção de energia foi ocasionada pela invasão ao sistema SCADA. Portanto, os *malwares* BlackEnergy3 e KillDisk, foram usados somente como ferramentas para efetuar o ataque (WEERATHUNGA, 2017). Sendo assim, as alterações feitas nos conversores Serial-Ethernet, bem como a negação de acesso ao *call center* da empresa devido as milhares de chamadas efetuadas pelo atacante foram usadas como base para atrasar a restauração do sistema (WEERATHUNGA, 2017). Portanto esse ataque configurou-se como uma invasão amplamente arquitetada, com diversos pontos concentrados para manter o ataque até conseguir o propósito desejado.

A maioria desses ataques ocorreu pelo acesso de computadores a Internet. O que demonstra que um sistema SCADA sem acesso direto a Internet é um formato relevante de prevenção à ciberataques.

2.3 FORMATOS DE CIBERATAQUES

Ataques em sistemas SCADA voltados a Redes Elétricas Inteligentes, podem comprometer a geração e transmissão de energia elétrica, bem como danificar equipamentos, colocar vidas em risco e deixar cidades sem energia elétrica. Esses ataques atuam com intuito de realizar alterações na rede de comunicação. Estas modificações são capazes de causar danos físicos severos no sistema (KALLURI; MAHENDRA, 2016; DO *et al.*, 2017).

Em sistemas SCADA voltados as *Smart Grids* os danos de ataques cibernéticos podem ocasionar a interrupção no fornecimento de energia, como relatado no ataque

BlackEnergy3 na Ucrânia e também coletar informações de estruturas privadas, principal propósito do RAT Dragonfly. Sendo assim, estudos pertinentes a esse tema são amplamente necessários o que faz com que esta seção seja imprescindível para entendimento de invasões a estruturas de GD. Portanto, esta seção aborda de forma não exaustiva alguns dos principais formatos de ciberataques voltados a sistemas de GD.

2.3.1 *Man-In-The-Middle*

O ataque *Man-In-The-Middle* atua no controle de informações trocadas entre duas plataformas (CONTI *et al.*, 2016). Isso fornece ao invasor a capacidade de mascarar a comunicação na rede. Desse modo, isso proporciona ao sistema de comunicação uma falsa impressão de que as extremidades estão se comunicando normalmente, aspecto que favorece para que o sistema se torne instável.

Essa camuflagem permite ao atacante detectar os dados enviados e realizar a transmissão de dados falsos (KIM; TONG, 2013). Além disso, esse modo viabiliza a exclusão de pacotes sem informar a rede de qualquer sinal de divergência nas informações. Por exemplo, quando o ataque é realizado em sistemas DER fotovoltaicos, o usuário malicioso atua na modificação das informações de distribuição e controle de energia, inserindo dados alterados em subestações, medidores, inversores, entre outros dispositivos (KANG *et al.*, 2015b).

Ataques *Man-In-The-Middle* podem causar danos consideráveis em uma rede elétrica (KIM; TONG, 2013). No entanto, entre todas as possíveis adversidades geradas por invasões a esses sistemas, é interessante destacar algumas delas, como as intermitências geradas na rede, a perda da taxa de resposta do sistema, o mau funcionamento dos dispositivos e até mesmo um possível apagão (LANGER *et al.*, 2016).

A Fig. 3 ilustra como um ataque realizado pelo modelo MITM funciona. Esse ataque age modificando a informação transmitida entre as duas vítimas. Assim, após o intruso completar seu objetivo, o ataque compromete o canal no qual os dados foram transportados, e todas as informações do sistema precisam ser passadas primeiro pelo intermediário, que nesse caso é o atacante.

Teoricamente, o modo mais simples de efetuar um ataque MITM é invadindo uma rede de comunicação e redirecionando o rastreamento das vítimas por intermédio de um nó malicioso (ZHU *et al.*, 2011). Por exemplo, uma rede sem fio em que não há a necessidade de autenticação, utilizada amplamente nos dias atuais, são particularmente

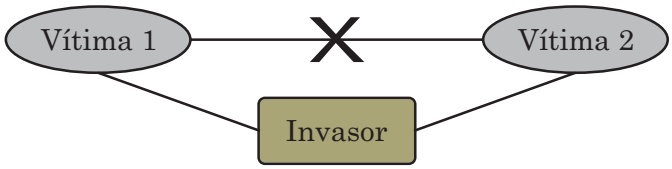

Figura 3: Ataque man-in-the-middle.

Fonte: O Autor (2021).

vulneráveis, pois qualquer usuário pode acessá-las, inclusive, um cliente mal intencionado e assim efetivar o ataque (RISTIC, 2013). Entretanto há outras maneiras de efetuar a invasão, como o ato de influenciar na interferência da infraestrutura de roteamento voltada a resolução de nomes de domínio e endereços de IP, entre outras opções, como as listadas abaixo:

- ARP *Spoofing*: O ARP (*Address Resolution Protocol*) é utilizado em redes locais para relacionar endereços MAC (*Media Access Control*) de dispositivos da rede a seu endereço IP correspondente. Portanto, um atacante que utilize como base esse formato tem a possibilidade de desvendar o endereço IP da máquina após o acesso a rede e redirecionar de modo desejado o tráfego para o dispositivo pretendido (RISTIC, 2013). Esse método de ataque é capaz de proporcionar a um usuário externo acesso a todos os dados transmitidos na rede atacada. No entanto, o ataque só pode ser realizado em redes que utilizam ARP e necessita que o atacante tenha acesso ao segmento de rede para realizar a invasão.

- WPAD *Hijacking*: O WPAD (*Web Proxy Auto-Discovery Protocol*) é utilizado pelos navegadores com intuito de restabelecer de modo automático a configuração do *proxy* HTTP. O WPAD se baseia em vários métodos, inclusive o DHCP (*Dynamic Host Configuration Protocol*) e DNS (*Domain Name System*). Para atacar o WPAD, um invasor inicializa um *proxy* na rede local e o divulga aos usuários que buscam o protocolo na rede.

- DNS *Hijacking*: O sequestro ou redirecionamento de DNS é uma forma de ataque na qual os invasores induzem a solução de forma incorreta sobre as consultas DNS e redirecionam todo o tráfego para um IP alternativo ao pretendido. Desse modo, o atacante que utiliza essa opção é capaz de redirecionar todo o rastreamento destinado a esse nome de domínio (RISTIC, 2013).

- DNS Cache *Poisoning*: O envenenamento de cache DNS explora a fragilidade no cache dos servidores DNS e possibilita ao atacante a injeção de informações invá-

lidas sobre nomes de domínio no cache. Desse modo, caso o ataque seja efetuado corretamente será possível afetar todos os usuários pertencentes ao servidor DNS afetado, de modo a completar o ataque em massa.

- BGP *Route Hijacking*: O BGP (*Border Gateway Protocol*) é um protocolo de roteamento utilizado nos principais roteadores da Internet, com o intuito de realizar a localização exata dos blocos de endereços IP (RISTIC, 2013). Sendo assim, invasores agem também utilizando essa base, pois se uma rota inválida for assumida por um ou mais roteadores, o tráfego integral de um bloco de endereços IP pode ser transferido para outro endereço, nesse caso o do atacante (RISTIC, 2013).

A execução de um ataque MITM em uma rede insegura é simples, pois o invasor pode adentrar na rede, entre duas extremidades e interceptar todas as mensagens trocadas. Posteriormente, o invasor consegue se passar por ambas as extremidades, devido ao fato de que no sistema não encontra-se nenhuma ferramenta que seja capaz de indicar qual o usuário que está enviando os dados. A adição do TLS a uma conexão possibilita a autenticação das extremidades utilizando criptografia, viabilizando assim um formato que amplia a segurança do sistema. No entanto, mesmo composto com o protocolo TLS, ainda é possível algumas formas de invadir o sistema.

A maioria das tentativas de ataques contra sistemas que possuem o protocolo TLS são formados pelo formato *Man-In-The-Middle* (RISTIC, 2013). Nesse cenário, as vítimas tentam estabelecer comunicação segura enviando chaves públicas entre si. No entanto, quando um invasor está presente na rede, essa conexão não é instituída. Então, nesse caso, o invasor retém as informações e retorna aos usuários outros dados, definidos pelo atacante e compostos por suas chaves públicas.

Depois que os dados manipulados pelo invasor são enviados, a vítima criptografa a mensagem pela chave do invasor e a envia para outra vítima. Mas o invasor intercepta os dados e os descriptografa usando uma chave privada. Em seguida, o agente malicioso envia para a vítima informações falsas compostas pela chave pública do usuário (CONTI *et al.*, 2016). Assim, isso faz com que o sistema funcione perfeitamente, sem que as vítimas reconheçam que estão sendo invadidas e que as informações trocadas entre os usuários foram alteradas. Entretanto, o protocolo TLS está em constante evolução para aperfeiçoamento de segurança, principalmente para minimizar a possibilidade desse tipo de ataque.

Algumas precauções são sugeridas para dificultar esses métodos de ataque MITM, como por exemplo:

- Uso de criptografia WEP/WAP forte em pontos de acesso;
- Utilização de VPNs;
- Uso de chaves públicas para autenticação.
- Uso de aplicações ou *softwares* confiáveis;

2.3.2 *Replay*

O ataque de repetição ou *replay* possui uma propriedade fundamental que se baseia na leitura e retransmissão de dados momentâneos compostos no sistema (CHEN *et al.*, 2017). Além disso, o invasor pode manipular os dados do sistema de modo a retransmitir diversas vezes um dado antigo gerado por um dispositivo afetado, fornecendo assim um comportamento irreal no sistema (IRITA; NAMERIKAWA, 2017).

O objetivo básico do invasor é fazer com que os dados alterados aparentem ser dados reais do sistema. Assim, durante um ataque *replay*, a confiabilidade dos dados transmitidos/recebidos no sistema não pode ser assegurada. O principal método para detectar tal ataque é identificá-lo no momento em que o ataque está sendo realizado. Se a identificação da invasão puder ser realizada, será possível efetuar as contra medidas (MO *et al.*, 2014).

O ataque de repetição pode ser aplicado para obter acesso a dados de redes protegidas, devido ao fato de se comportar como um dado real do sistema e possuir credenciais que aparentam ser válidas para a estrutura. Esse ataque pode ser aplicado em sistemas de geração distribuída para duplicar estados ou valores obtidos na geração de energia.

Uma opção usada para prevenção de ataques de repetição são utilizar chaves criptográficas únicas para cada pacote trocado entre as extremidades da conexão. Outra alternativa também baseia-se em medidores de tempo em todas as mensagens trocadas. Além disso, é possível pré-estabelecer que o sistema armazene dados semelhantes no cache e interrompa após um certo limite as mesmas requisições.

2.3.3 *False Data Injection*

Os ataques de injeção de dados falsos em sistemas SCADA estão sendo considerados como um cenário em constante evolução (LIU *et al.*, 2011). Este ataque foi desenvolvido para atacar os Sistemas de Gestão de Energia (SGE), principalmente em *Smart Grids* modernas (WANG *et al.*, 2017b). Um SGE inteligente é composto por estimativas de estado. O objetivo das estimativas de estado é assegurar que a rede elétrica esteja operando em condições pretendidas (YANG *et al.*, 2014). A condição essencial para que o ataque funcione corretamente baseia-se em conseguir passar no teste de estimativas de estado e evitar o alarme (Abdallah; Shen, 2016). Dessa forma, o ataque engana o centro de controle com valores dentro dos limites estabelecidos, o que induz ao mestre da estrutura possíveis tomadas de decisão errôneas.

Os ataques FDI provaram ser eficientes na modificação da estimativa de estado desses sistemas (ZHAO *et al.*, 2015). O invasor FDI trabalha com intuito de modificar a informação gerada pelo sistema SCADA, de modo a produzir efeitos negativos na estrutura (HUG; GIAMPAPA, 2012).

Se o sistema de controle não for capaz de identificar alterações na estimativa de estado, isso pode ocasionar o comprometimento da segurança do sistema, sendo assim capaz de gerar ações equivocadas pelo operador. No entanto, mesmo que a invasão possa ser detectada, parte da informação do sistema é perdida durante o ataque, o que faz com que o estimador de estado possa gerar valores incorretos e assim possa causar efeitos físicos ou econômicos indesejáveis na rede (LIANG *et al.*, 2016).

Depois que a invasão for concluída e todo o sistema for atingido, é impossível interromper o trabalho do invasor em grande parte do processo. Uma estimativa de estado falsa é capaz de enganar todas as funções de controle e operação de um SGE (LIANG *et al.*, 2016). Por causa disso, o ataque FDI é considerado uma das ameaças mais graves que podem ocorrer em uma rede inteligente (XU *et al.*, 2017). No entanto, segundo (Abdallah; Shen, 2016) uma forma efetiva de prevenir esse tipo de ataque baseia-se em garantir a integridade e confiabilidade dos dados transmitidos no sistema.

2.3.4 *Denial of Service* e *Distributed Denial of Service*

Ataques DoS e DDoS em sistemas DER são focados em causar a negação do serviço do dispositivo alvo, sendo assim após a invasão a unidade atingida será incapaz de efetuar a sua função no sistema. Essas invasões visam transformar os dados correntes

do sistema em obsoletos ou mesmo inacessíveis (DIOVU; AGEE, 2017b). Esses ataques podem causar falhas consideráveis nos sistemas, como a supressão do fornecimento de energia por exemplo (PANDEY; MISRA, 2016).

A Fig. 4 apresenta como os ataques DoS e DDoS funcionam em uma estrutura SCADA. Esses ataques podem ocorrer em um módulo RTU ou MTU na rede. No entanto, a unidade remota é o componente mais vulnerável do sistema SCADA, porque um sistema pode ser composto por várias RTUs monitoradas e controladas, que podem ser posicionadas em diferentes locais ao redor da planta (KALLURI; MAHENDRA, 2016). Portanto, supondo que um ataque DoS ou DDoS tenha ocorrido em uma ou mais RTUs. Desse modo, o atacante tem a liberdade de congestionar a rede e o sistema não é capaz de identificar esse problema, o que pode comprometer a comunicação. Devido à interrupção na comunicação, o nível de MTU também é comprometido, desempenhando de modo integral o objetivo do ataque.

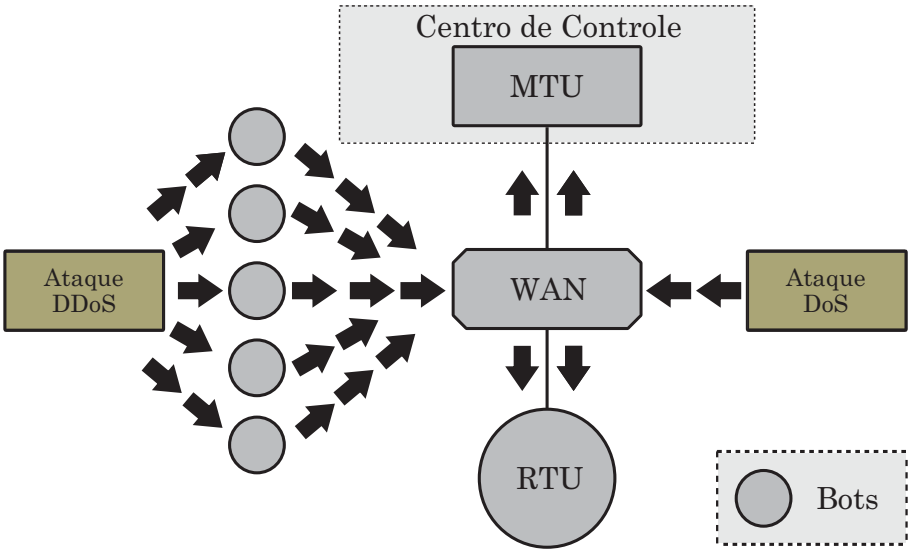

Figura 4: Ataque DoS e DDoS.

Fonte: O Autor (2021).

As principais diferenças entre os ataques DoS e DDoS estão no método de realização dessas invasões. O ataque DoS é ocasionado por apenas um invasor. No entanto, esse invasor é destinado a enviar vários pacotes, causando assim a sobrecarga do dispositivo de destino e impossibilitando assim o seu uso (RIZZETTI *et al.*, 2015).

Um exemplo interessante sobre esse ataque refere-se a uma insegurança obtida durante uma etapa do desenvolvimento do sistema de energia distribuído, utilizado no presente trabalho. O inversor durante a etapa em questão aceitava somente uma conexão, devido a disponibilidade de acesso a uma única porta Modbus. Portanto, caso esse sis-

tema fosse disponibilizado dessa forma, um invasor poderia conectar ao dispositivo antes do gerenciador de energia, fazendo com que a conexão com o gerenciador fosse negada, categorizando assim, um ataque DoS. Entretanto, com a inclusão de formatos de autenticação, essa conexão não seria aceita, como é o caso do Modbus sob TLS, que atualmente é fundamentado em aplicação de formatos de autenticação para uma proteção superior da conexão (MODBUS, 2018).

Por outro lado, no modo DDoS, o ataque é realizado por múltiplas fontes (*bots*) simultaneamente, o que pode amplificar os danos causados e assim tornar a proteção ainda mais difícil de ser realizada (DIOVU; AGEE, 2017a). Devido a característica de redes inteligentes serem submetidas a conexão com à Internet, a afirmação de que qualquer sistema SCADA está vulnerável a ataques DDoS é incontestável (YADAV *et al.*, 2016; FOGLIETTA *et al.*, 2018).

O ataque DDoS visa prejudicar a capacidade de tomada de decisão que ocorre na rede de comunicação. O ataque é projetado para diminuir o desempenho da rede, fazendo com que os modos de operação selecionados por clientes autorizados na estrutura não sejam aceitos (WANG *et al.*, 2017a).

O ataque DoS, assim como o MITM, é baseado em diversos formatos distintos. Alguns dos tipos de ataque DoS comumente realizados são listados abaixo:

- SYN *Flood*: Esse método visa a sobrecarrega direta na camada de transporte e indireta na camada de aplicação do modelo OSI, que permite uma sobrecarga na CPU do dispositivo-alvo. Além disso, esse ataque afeta diretamente comunicações TCP durante o processo de *three-way handshake*. Nesse formato o invasor envia diversas requisições (SYN) para inicializar uma nova conexão e o sistema-alvo disponibiliza novas conexões sem finalizar as estabelecidas anteriormente. Logo, isso induz o aumento do consumo da CPU, agindo até tornar os serviços indisponíveis.

- UDP *Flood*: Esse formato visa a sobrecarga da conexão de rede por meio da inundação devido a diversos envios de datagramas UDP a portas aleatórias do host-alvo. Nesse formato o invasor envia uma quantidade de pacotes maior que o link da rede-alvo pode suportar, gerando assim, uma inundação na rede e inviabilizando o uso.

- *Smurf*: Esse ataque visa a sobrecarga da rede interna por intermédio do envio de pacotes via *broadcast*. Esse invasor gera uma falsa solicitação, contendo o IP de origem falsificado (composto pelo IP da vítima). A maioria dos endereços internos

da rede responde de forma padrão o *broadcast* e se a rede constar com um número considerável de dispositivos, o computador da vítima será inundado rapidamente pelas respostas da solicitação.

Alguma das principais recomendações para prevenir ataques DoS baseiam-se em:

- Investimento em uma largura de banda maior;
- Uso de firewall no gerenciamento de conexões;
- Utilização de um *proxy* reverso.

2.4 PROTOCOLO DNP3

O DNP3 foi desenvolvido como um protocolo proprietário pela empresa Harris Controls Division no início dos anos 90, para a realizar a comunicação entre dipositivos de aquisição de dados e controle de industrias do setor de eletricidade (CLARKE *et al.*, 2004). Posteriormente, o protocolo foi disponibilizado para o *DNP3 User Group*, que permitiu o uso de terceiros, tornando-o um protocolo aberto.

O padrão DNP3 foi projetado para efetuar transferência de pacotes relativamente pequenos de modo confiável, transmitidos em uma sequência determinística (CLARKE *et al.*, 2004). Além disso, o protocolo apresenta diversos benefícios, como a arquitetura em camadas, a interoperabilidade, entre outros.

O DNP3 foi concebido especialmente para aplicações SCADA (CLARKE *et al.*, 2004). Os sistemas SCADA são destinados a diversos setores atualmente, como em *Smart Grids*, porque possibilitam a troca e o gerenciamento de dados dessas estruturas (LU *et al.*, 2011). Além disso, arquiteturas SCADA voltadas a Redes Elétricas Inteligentes podem fornecer diversos benefícios, como identificação de falhas em sistemas de medição em tempo real, controle de carga em transformadores, solução de balanço energético em tempo real, entre outros. No entanto, a principal preocupação de sistemas SCADA utilizando o protocolo DNP3 talvez seja a falta de segurança, pois a maioria dos dispositivos com suporte ao protocolo não possuem autenticação de identidade, criptografia de dados e controle de acesso (JIN *et al.*, 2011).

Uma estrutura clássica de um sistema SCADA DNP3 classifica os dispositivos em *Master* e *Outstation* (IGURE *et al.*, 2006). O DNP3 proporciona a estação mestre a

capacidade de solicitar dados das *outstations* utilizando comandos de função de controle pré-estabelecidos na estrutura (FARUK, 2008). Com isso, a *outstation* responde a solicitação da unidade mestre com os dados desejados (MAJDALAWIEH *et al.*, 2007). Além disso, os dispositivos *oustation* tem a possibilidade de enviar informações via resposta não solicitada, quando as mensagens transmitidas forem relevantes para o sistema (CLARKE *et al.*, 2004).

O DNP3 refere-se a um protocolo de transmissão de dados capaz de ser realizado por uma comunicação serial ou IP. O DNP3, assim como todo protocolo, estipula um conjunto de regras para estabelecer a comunicação entre os dispositivos.

O protocolo DNP3 suporta uma variedade de configurações de rede distintas, tais como a topologia ponto-a-ponto, multi-ponto, hierárquica e concentradora de dados. Desse modo, a Figura 5 disponibiliza o formato comum de configuração desses quatro modelos.

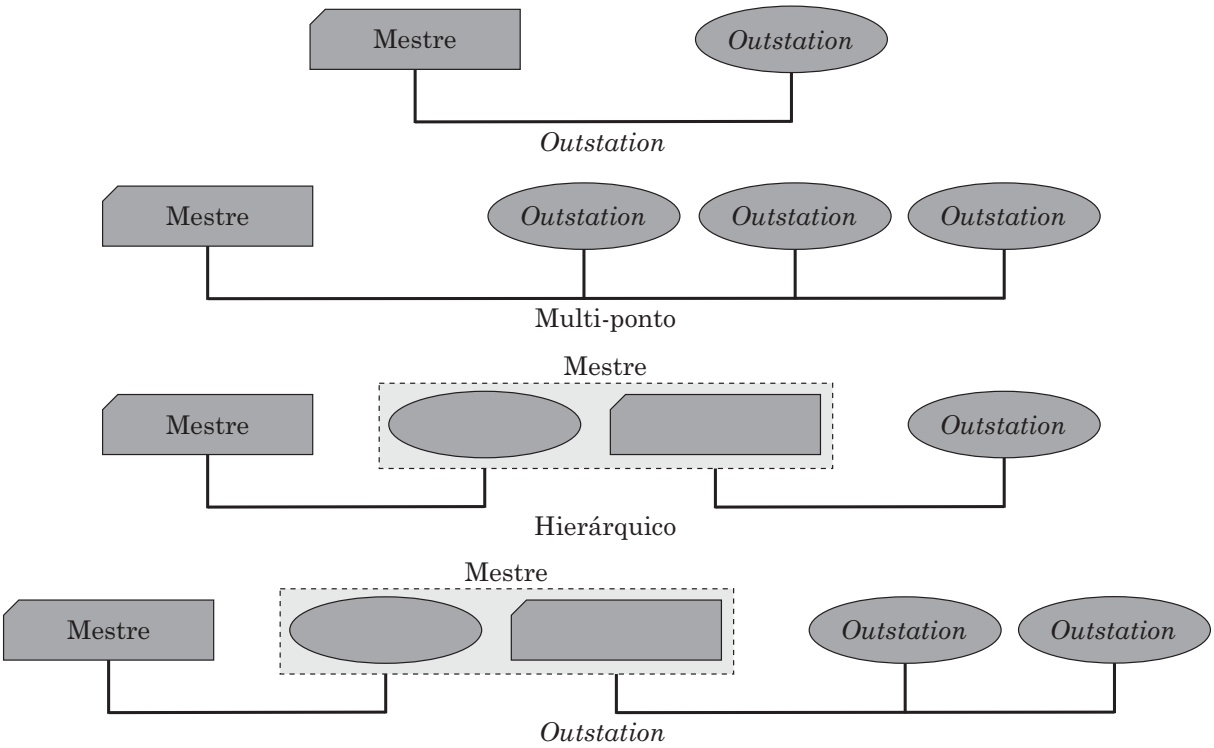

Figura 5: Configurações de rede DNP3.

Fonte: O Autor (2021).

Em uma configuração ponto-a-ponto, um dispositivo *master* e um *outstation* compartilham uma conexão dedicada (EAST *et al.*, 2009). Entretanto, há uma opção em que o mestre pode comunicar com várias *outstations* distintas, essa configuração é denominada multi-drop ou multi-ponto (EAST *et al.*, 2009). Nesse formato, cada estação

externa recebe todas as solicitações do dispositivo mestre, mas cada *oustation* retorna somente mensagens endereçadas ao *master* (IGURE *et al.*, 2006).

Na configuração hierárquica, há a possibilidade do mesmo dispositivo atuar como um *oustation* para um segmento, mesmo sendo mestre em outro, intitulando esse elemento de *sub-master* (EAST *et al.*, 2009). A quarta configuração apresentada na Figura 5 refere-se a topologia concentradora de dados, formato que é baseado na junção do modelo multi-ponto com o hierárquico.

O DNP3 aceita três modos simples de comunicação entre uma estação de controle (unidade mestre) e dispositivos externos (*outstation*) (CLARKE *et al.*, 2004). Na transação *unicast*, o dispositivo mestre envia uma solicitação a um *oustation* endereçado, que retorna uma mensagem de resposta. Por exemplo, o mestre requisita uma leitura de corrente ou uma ação (desarmar o disjuntor), a estação remota retorna o valor de leitura da corrente e uma confirmação composta com a informação que o disjuntor foi disparado ou houve uma falha (JIN *et al.*, 2011).

Na transição de transmissão, o dispositivo *master* envia uma solicitação a todos os dispositivos remotos da rede, como por exemplo, o envio de uma ação para redefinir todos os sensores de corrente e os *oustations* não respondem à mensagem de difusão (EAST *et al.*, 2009). O terceiro modo baseia-se nas respostas não solicitadas, provenientes dos *oustations* (JIN *et al.*, 2011). Essas informações são geradas no momento que há alguma atualização ou alerta de um dado crucial na estrutura, como por exemplo quando uma leitura de corrente excede o limite estipulado (EAST *et al.*, 2009), ou quando um estado de um determinado componente da rede altera de ligado para desligado (JIN *et al.*, 2011).

As três camadas do protocolo DNP3 são posicionadas acima de uma camada física, nível responsável pela transmissão de informações em mídias físicas, como rádio, satélite, fio e fibra óptica (CLARKE *et al.*, 2004). O nível físico é capaz de fornecer cinco serviços: envio de dados, recebimento de dados, conexão, desconexão e atualização de estado (EAST *et al.*, 2009).

Os sistemas SCADA atuais utilizam o DNP3 em redes IP. O DNP3 *User Group* definiu que deve-se manter a configuração de três níveis quando os sistemas forem baseados em IP (EAST *et al.*, 2009). Sendo assim, quando o protocolo DNP3 for utilizado em redes IP, recomenda-se aplicar as três camadas diretamente na pilha acima das camadas TCP/IP ou UDP/IP (EAST *et al.*, 2009), como é demonstrado na Figura 6.

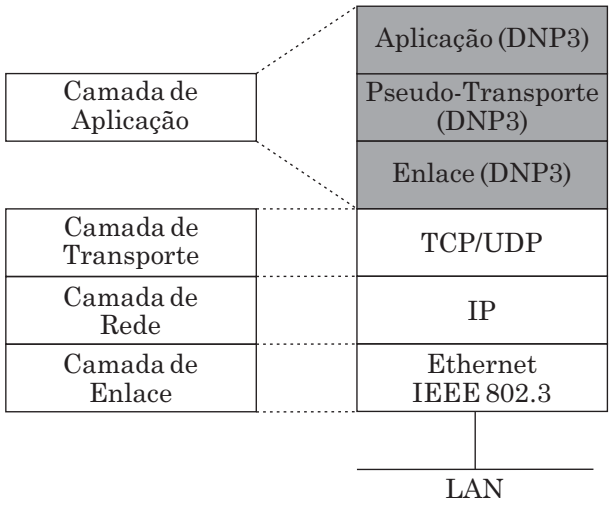

Figura 6: Modelo DNP3 sob TCP/IP.

Fonte: O Autor (2021).

O protocolo DNP3 possui as camadas de aplicação, pseudo-transporte e enlace. O nível pseudo-transporte é uma junção das camadas de rede e transporte do modelo OSI, visto que esses níveis são suportados de modo limitado pelo DNP3 (CLARKE *et al.*, 2004). A seguir será comentado brevemente sobre cada camada do modelo DNP3.

2.4.1 Camada de aplicação

A camada de aplicação é responsável por especificar as informações de solicitação e resposta do DNP3 (CLARKE *et al.*, 2004), e também estabelecer as funções do *master* e *oustation* na estrutura (EAST *et al.*, 2009). Além disso, é a camada mais alta do protocolo e faz a função de executar a interface com os níveis mais baixos, com o intuito de enviar os dados ao dispositivo desejado, de modo a atingir a transmissão fim-a-fim dos dados transferidos (CLARKE *et al.*, 2004).

Na camada de aplicação realiza-se a solicitação de dados às *oustations* pelo dispositivo mestre, bem como as respectivas respostas dos dispositivos externos ao *master* (EAST *et al.*, 2009). A camada de aplicação inicialmente constitui os dados em blocos de tamanho gerenciável, que são denominados ASDUs (*Application Service Data Units*) (CLARKE *et al.*, 2004). Além disso, nesse nível é realizado o desenvolvimento do APDU (*Application Protocol Data Unit*), que é formado pela combinação do cabeçalho com o dado ASDU (CLARKE *et al.*, 2004). O cabeçalho é denominado APCI (*Application Protocol Control Information*).

O tamanho do cabeçalho possui 2 ou 4 *bytes*, o que diferencia se o dado transmitido é uma solicitação (2 *bytes*) ou resposta (4 *bytes*). Entretanto, quando é enviado um comando, ou ocorre uma solicitação pelo usuário que não necessite de dados, o pacote conta somente com o cabeçalho e nenhuma ASDU (CLARKE *et al.*, 2004).

Quando há a necessidade de várias APDUs, cada uma é chamada de fragmento (EAST *et al.*, 2009). O número de fragmentos para descrever uma ASDU não é limitado, porém o comprimento permitido para cada fragmento é restrito a um valor máximo de 2048 *bytes* (CLARKE *et al.*, 2004).

A Figura 7 apresenta o modelo do cabeçalho da camada de aplicação. O elemento controle de aplicação exerce uma função análoga ao mesmo elemento presente na camada pseudo-transporte, porém em um nível superior. Duas *flags* são inclusas de modo a caracterizar o primeiro e o último fragmento de uma informação, bem como o número de sequência para requisição de remontagem. Após isso uma *flag* é usada para solicitar a confirmação do recebimento do fragmento (EAST *et al.*, 2009).

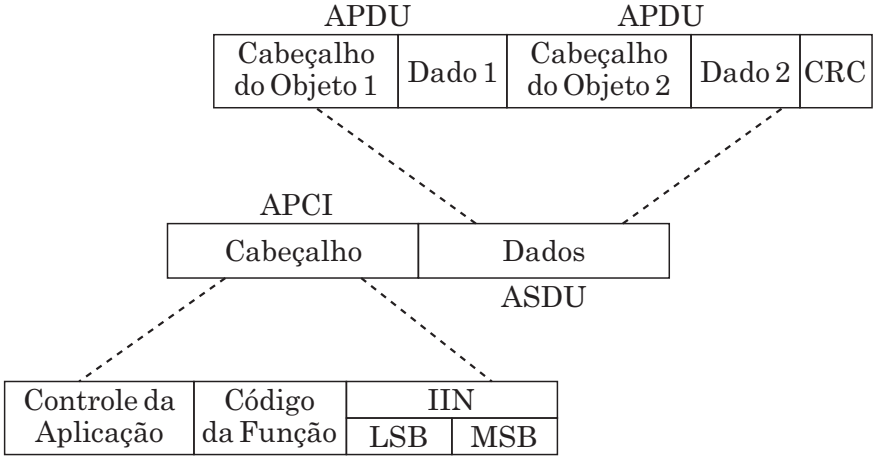

Figura 7: Mensagem DNP3 da camada de aplicação.

Fonte: Adaptado de (CLARKE et al., 2004).

O elemento `código da função` é responsável por transmitir o objetivo da mensagem (CLARKE *et al.*, 2004). Esse setor é destinado a aplicar as solicitações e respostas, contudo suas funções são modificadas conforme o tipo de mensagem enviada. Os códigos de função são divididos em 5 funções para solicitações: de transferência, de controle, de congelamento, de configuração e funções de sincronização (EAST *et al.*, 2009). No entanto, para mensagens de resposta, os códigos de funções são três: confirmação, resposta ou resposta não solicitada.

Os códigos de função estão presentes tanto no mestre, quanto no *outstation*. Alguns exemplos de códigos de função mestre são restabelecer o *link* remoto, requisitar *status* atual do *link*, redefinir o processo do usuário e função de teste (EAST *et al.*, 2009). Entretanto, os códigos de função para os *oustations* são confirmação positiva, *status* do *link*, mensagem não aceita e sem serviço de *link*.

Os cabeçalhos das mensagens de resposta são compostos por dois *bytes* a mais que o de solicitação, que são constituídos pelo campo IIN (*Internal Indication*), que é responsável por transferir dados úteis a respeito da *oustation* para o dispositivo mestre (CLARKE *et al.*, 2004). Cada *bit* referente ao elemento IIN concerne uma informação específica, como por exemplo: reinicialização do dispositivo, sincronização de tempo necessária, código de função não implementado, parâmetro inválido e objetos solicitados desconhecidos (EAST *et al.*, 2009).

Após o cabeçalho são posicionados os objetos de dados, responsáveis por transmitir representações codificadas de dados (EAST *et al.*, 2009), vide Figura 7. Dessa forma, diversos objetos de dados são determinados de modo a proporcionar que dispositivos processados em plataformas distintas possam transferir informações e ações de modo eficiente (CLARKE *et al.*, 2004). Nos objetos de dados podem ser enviadas diversas informações distintas, como por exemplo: contadores, entradas e saídas analógicas e binárias (JIN *et al.*, 2011).

2.4.2 Camada de pseudo-transporte

A camada de pseudo-transporte do modelo DNP3 é reponsável por fragmentar e remontar as informações transmitidas (IGURE *et al.*, 2006). Essa camada é composta pela união da camada de rede e de transporte baseadas no modelo OSI. As funções da camada de rede coordenam o roteamento e o controle de fluxo de pacotes sobre a rede. Logo, as funções baseadas nessa camada são responsáveis por conceder uma troca de dados inteiros, de forma transparente de ponta-a-ponta, incluindo a desmontagem e montagem de mensagens, bem como a correção de erros (CLARKE *et al.*, 2004).

Esse nível permite que os dados de aplicação compostos com mais de um *frame* da camada de enlace utilizem vários *frames* (CLARKE *et al.*, 2004). Além disso, a camada de pseudo-transporte acrescenta ao quadro um *byte* composto pelas *flags* FIR, FIN e um *número de sequência*, como é demonstrado na Figura 8. As *flags* FIR e FIN

são responsáveis por sinalizar o primeiro e o último quadro de um dado fragmentado, respectivamente (EAST *et al.*, 2009).

O *número de sequência* é incrementado para cada quadro seguinte e permite remontar as mensagens para serem processadas pela camada de aplicação (EAST *et al.*, 2009). O *número de sequência* possui um comprimento de 6 *bits*. Além disso, os dados de sequência auxiliam na identificação quando há a ocorrência de quadros perdidos (CLARKE *et al.*, 2004).

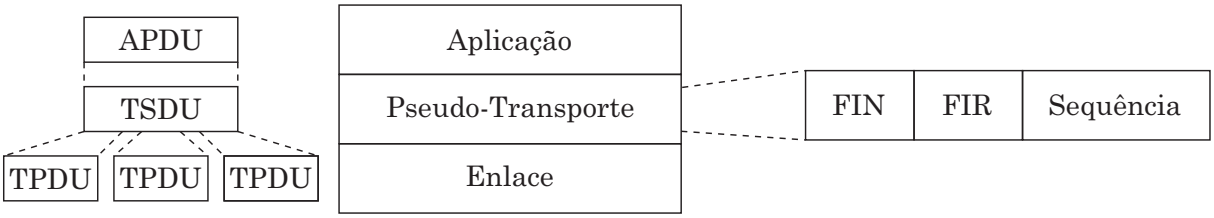

Figura 8: Nível pseudo-transporte.

Fonte: O Autor (2021).

A camada de transporte utiliza os dados do usuário, a TSDU (*Transport Service Data Unit*) e divide-os em um ou mais TPDUs (*Transport Protocol Data Units*) e posteriormente, envia cada uma dessas unidades a camada de enlace para a transmissão (CLARKE *et al.*, 2004). Dessa forma, os TPDUs convertem-se nos dados do usuário na camada de enlace, formando assim os LSDUs (*Link Service Data Units*) (IGURE *et al.*, 2006).

2.4.3 Camada de enlace

A camada de enlace é responsável por estabelecer e manter uma comunicação de forma confiável através do meio físico. A comunicação estabelecida envolve a troca de dados entre remetente e destinatário (CLARKE *et al.*, 2004).

A unidade de dados da camada de enlace é denominada *frame* ou quadro. Um *frame* desse nível possui um comprimento fixo de 10 *bytes* para o cabeçalho e uma seção de dados composta por informações dos níveis de pseudo-transporte e de aplicação (EAST *et al.*, 2009). O comprimento máximo da seção de dados ou LSDUs, como também são chamados, possuem um comprimento de 250 *bytes* de informações dos níveis superiores, além do código CRC, totalizando assim, 292 *bytes* (CLARKE *et al.*, 2004).

A Figura 9 apresenta o modelo do cabeçalho da camada de enlace. O elemento início ou *start* possui um valor conhecido de dois *bytes* (0x0564), o que possibilita ao

destinatário conhecer onde começa a transmissão DNP3 (EAST *et al.*, 2009). O campo tamanho é responsável por disponibilizar o número de *bytes* restantes no *frame*, com exceção aos *bytes* que fazem parte do CRC (CLARKE *et al.*, 2004).

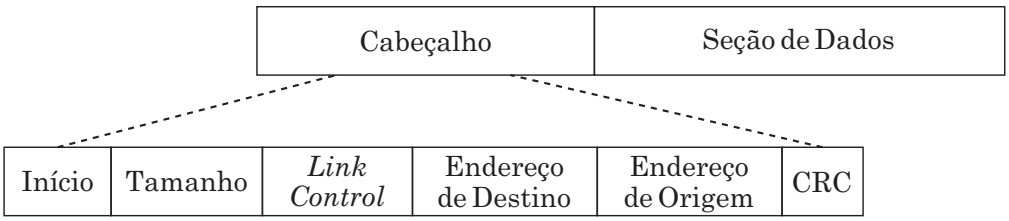

Figura 9: Quadro da camada de enlace.

Fonte: Adaptado de (CLARKE et al., 2004).

O elemento *link control* contém informações que controlam o fluxo de dados, proporcionam sequenciamento e estebelecem a função do *frame* (IGURE *et al.*, 2006). Esses dados auxiliam no reconhecimento do dispositivo que iniciou a comunicação, fornecem o *status* do *link* e determinam se o dispositivo refere-se ao mestre ou *oustation* da sessão. Além disso, o campo *link control* apresenta um código de função de quatro *bits*, que determina o intuito da informação (EAST *et al.*, 2009).

O elemento *link control* também possui dois *flags* para sincronização da comunicação e controle de fluxo. O endereço de destino é composto por 16 *bits* e especifica o endereço do dispositivo receptor desejado, além de disponibilizar também um envio de *broadcast* (CLARKE *et al.*, 2004). O campo referente ao endereço de origem determina o endereço do dispositivo remetente da mensagem, esse elemento também é constituído por 16 *bits* (JIN *et al.*, 2011). Além disso, é incluso 16 *bits* para o CRC, responsável por verificar a integridade da transmissão (EAST *et al.*, 2009).

2.5 IEC 61850

O avanço tecnológico de equipamentos usados em sistemas de energia tornou disponível que diversos sinais analógicos e digitais estejam presentes em um único dispositivo (MACKIEWICZ, 2006). Além disso, esse avanço possibilitou atingir taxas de transmissão elevadas, diferentemente do que possuíam os formatos antigos (BAIGENT *et al.*, 2004). Sendo assim, as maiores preocupações em sistemas como esse tornaram-se outras, como a arquitetura e configuração dos componentes, por exemplo. No entanto, isso requer um formato padronizado capaz de descrever todos os elementos de um sistema inteligente como esse. Para atender essa necessidade originou-se o padrão IEC 61850, voltado a sistemas e redes de comunicação em subestações.

O padrão IEC 61850 não refere-se apenas a um protocolo de comunicação. A norma IEC 61850 é mais que um protocolo de comunicação (HUANG, 2018). O padrão IEC 61850 possibilita um formato sistêmico de especificar o dispositivo e o sistema, para posteriormente executar os serviços de comunicação pertinentes à estrutura (HUANG, 2018).

Os principais protocolos utilizados na indústria de energia suportam o padrão IEC 61850, tais como os protocolos Modbus, Profibus, Profinet, SEP2.0, TCP/IP e DNP3. Além disso, o padrão IEC 61850 também é direcionado a padronizar e integrar os sistemas de controle e supervisão de sistemas de energia (MACKIEWICZ, 2006).

Os protocolos de comunicação herdados (DNP3 ou Modbus), tem como base índices ou registradores, respectivamente (HUANG, 2018). Todos os pontos relacionados a informações recebem um número de índice composto pelo tipo de dado a ser enviado (MACKIEWICZ, 2006). Por exemplo, a corrente de carga da fase A pode ser definida como o índice de entrada analógica 1 na lista de pontos de dados, que será posteriormente entregue a sistemas SCADA (HUANG, 2018). A estrutura SCADA incontestavelmente deve utilizar a mesma lista de pontos de dados de comunicação do sistema adotado (BAIGENT *et al.*, 2004). Além disso, se alguma alteração ou atualização for realizada na lista de dados, será necessária uma modificação nos parâmetros do sistema SCADA para atender a nova lista (HUANG, 2018).

A norma IEC 61850 busca padronizar a comunicação de dispositivos inteligentes voltados a sistemas de energia. O padrão é voltado a atender diversos protocolos de comunicação e pode ser aplicado em redes TCP/IP. Além disso, atua com mapeamento de padrões MMS (*Manufacturing Message Specification*), XML (*eXtensible Markup Language*), SV (*Sampled Variables*) e GOOSE (*Generic Object Oriented Substation Event*) (BAIGENT *et al.*, 2004).

As principais vantagens do uso do padrão IEC 61850 refere-se a interoperabilidade, facilidade de configuração, uso de infraestrutura padrão (Ethernet, por exemplo). O uso desse padrão em sistemas de energia têm se tornado essencial, porque diversos dispositivos diferentes, oriundos de fabricantes distintos, utilizando protocolos específicos são posicionados em sistemas de geração de energia.

O padrão IEC 61850 com seu modelo abstrato de dados, possibilita um perfil para descrever os dispositivos presentes em um sistema de energia com uma estrutura idêntica, que relacionam as mesma funções dentro do sistema (MACKIEWICZ, 2006).

Embora o modelo abstrato seja algo essencial para obter um nível considerável de interoperabilidade, esses modelos necessitam de operação baseada em protocolos práticos e comumente aplicados em setores de energia (BAIGENT *et al.*, 2004). Como é o caso do protocolo DNP3.0.

A norma IEC 61850 disponibiliza um modelo completo de como os equipamentos do sistema de energia precisam estruturar os dados de modo consistente sobre qualquer tipo e fabricante de produtos (BAIGENT *et al.*, 2004). Isso reduz consideravelmente o esforço empenhado na configuração de sistemas, pois isso permite com que os equipamentos possam se configurar (MACKIEWICZ, 2006). Por exemplo, se no sistema for requerido o uso de uma entrada analógica para um transformador de corrente, esse sistema pode identificar o módulo e o incluir como uma unidade de medição sem necessitar da interação do usuário, somente por atender as recomendações da norma (BAIGENT *et al.*, 2004). Portanto, devido ao fato de possuir esse padrão bem conhecido, se não houver medidas de segurança nos sistemas compostos pelo padrão IEC 61850, um invasor poderá identificar os dados trafegando na rede facilmente, bem como conseguirá alterar os dados pertencentes a essa estrutura.

A norma IEC 61850 possui 10 partes, o presente trabalho vai atentar somente a detalhes compostos na parte 7 e 9 da IEC, que referem-se a estrutura básica de comunicação para subestações e equipamentos alimentadores e mapa de serviços de comunicação específicos, respectivamente. Com isso, alguns conceitos são indispensáveis, como nós lógicos e classe comum de dados.

2.5.1 Nós lógicos

Em um sistema de geração de energia são realizadas diversas trocas de informações. Essas informações são formadas por um agrupamento de dados e aplicações, e são trocados pelas funções estabelecidas nos elementos do sistema. A menor parte dessa função que faz a troca de informações é denominada nó lógico. Além disso, um conjunto de nós lógicos formam os denominados LD's (*Logical Devices*), que estão inclusos nos dispositivos presentes em um sistema de geração de energia (BAIGENT *et al.*, 2004).

A norma IEC 61850-7-4 estabelece 92 nós lógicos distintos, definidos por quatro letras. Alguns exemplos de nós lógicos definidos pela IEC são `XSWI` (chave seccionadora), `XCBR` (disjuntor), `XFUS` (fusível) `TCTR` (transformador de corrente) e `YPTR` (transformador

de potência). Os nós lógicos são divididos em grupos. O que estabelece a categoria de cada nó é sua inicial, conforme demonstra a Tabela 1.

Tabela 1: Categorias de nós lógicos.

Inicial	Grupo	Nó Lógico
A	Controle automático	4
C	Controle	5
G	Funções genéricas	3
I	Interface e armazenamento	4
L	Nós lógicos do sistema	3
M	Medição	8
P	Funções de proteção	28
R	LN relativos a proteção	10
S	Sensor e monitoramento	8
T	Transformadores	2
X	Equipamentos de manobra	2
Y	Transformadores de potência	4
Z	Equipamentos adicionais	15

Para atingir os requisitos básicos de comunicação em um sistema de energia é necessário o reconhecimento de todas as funções. Essa categorização respeita a abordagem LN (*Logical Node*) e PICOM (*Piece of Information for COMmunication*). Além disso, compõe-se em três partes:

- Descrição da função, bem como a divisão em LN.
- Descrição do nó lógico, bem como o PICOM comutado.
- Descrição do PICOM, bem como seus atributos.

Cada nó lógico destinatário necessita possuir a competência de reconhecimento sobre a verificação se os dados disponibilizados são íntegros, válidos e de qualidade apropriada. Quando ocorrer o recebimento de dados perdidos ou corrompidos, o nó lógico receptor não deverá atuar no formato comum, porém poderá ser degradado. Dessa forma, esses formatos possíveis devem ser pré-estabelecidos. Porém, quando ocorrer a degradação do LN, o comportamento deve ser determinado de forma única, pois excede o escopo do padrão IEC 61850. Além disso, é possível definir qualquer LN estabelecido na recomendação, bem como elaborar dados, atributos e funções distintas do SAS (Sistema de Automação de Subestações) (MACKIEWICZ, 2006). No entanto, toda mudança deve cumprir as recomendações de interoperabilidade.

2.5.2 Classe comum de dados

Todo componente de um LN, segue um perfil e são estabelecidos em conformidade com a categorização da *Common Data Class* (CDC), determinada na norma IEC 61850-7-3. Entretanto, a IEC também possibilita o desenvolvimento de novos dados, desde que sejam cumpridas as recomendações presentes no padrão.

O CDC foi elaborado utilizando blocos comuns de modo a formar objetos de dados. Dessa forma, é possível mapear serviços e objetos para outros protocolos.

As classes comuns de dados da IEC 61850 são determinadas conforme a especificação da IEC 61850-7-3 (DNP, 2018). Os conceitos utilizados no perfil estabelecido segundo a recomendação podem ser verificados na Tabela 2.

Tabela 2: Conceitos das colunas da IEC 61850.

Nome	Informação	Descrição
ACT	Estado	Proteção ativa
ACD	Estado	Proteção direcional ativa
SEC	Estado	Contador de violações
SPS	Estado	Ponto único
DPS	Estado	Ponto duplo
INS	Estado	Inteiro
BCR	Estado	Contador binário
SPC	Controle	Ponto único controlável
DPC	Controle	Ponto duplo controlável
INC	Controle	Inteiro controlável
BSC	Controle	Informação binária de controle de posição de passos
ISC	Controle	Informação analógica de controle de posição de passos
APC	Controle	*Setpoint* analógico controlável
SPG	Configuração	Parâmetros de um ponto de medição
ING	Configuração	Parâmetros de valores inteiros
ASG	Configuração	Configuração analógica
CURVE	Configuração	Configuração de curva
DPL	Supervisão	Identificador do dispositivo
LPL	Supervisão	Identificador do nó lógico
CSD	Supervisão	Descrição do formato de curva
MV	Medição	Valor medido
CMV	Medição	Valor complexo

SAV	Medição	Valor análogo
WYE	Medição	Valor fase-terra de sistema trifásico conectado em estrela
DEL	Medição	Valor fase-terra de sistema trifásico conectado em triângulo
SEQ	Medição	Sequência de fases
HMV	Medição	Valor de harmônica
HWYE	Medição	Valor de harmônica de sistema trifásico conectado em estrela
HDEL	Medição	Valor de harmônica de sistema trifásico conectado em triângulo

2.5.3 Modelo de dados

No modelo de dados IEC 61850, cada dado possui um nome legível específico. Essas denominações possuem uma estrutura hierárquica, similar a um sistema de arquivos. Essa estrutura é apresentada na Figura 10.

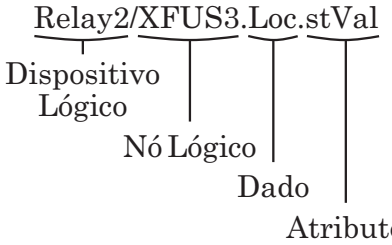

Figura 10: Exemplo de nome - IEC 61850.

Fonte: O Autor (2021).

O padrão IEC 61850 possui um vasto conjunto de objetos de dados, tendo cerca de 1400 definições distintas. A Tabela 3 apresenta alguns dos dados referenciados na norma. Além disso, quando necessário ainda é possível definir um atributo para o modelo em questão, conforme a necessidade de cada ponto.

Tabela 3: Definições de objetos de dados do padrão IEC 61850.

Objeto de Dados	CDC	Descrição
TotVA	MV	Potência aparente total em um circuito trifásico
TotW	MV	Potência ativa total em um circuito trifásico
TotVAr	MV	Potência reativa total em um circuito trifásico
A	WYE	Corrente de linha
PPV	DEL	Tensão fase-fase (VL1, VL2, ...)
TotVAh	BCR	Energia aparente
TotWh	BCR	Energia ativa
Ang	MV	Ângulo entre tensão em corrente
Pos	DPC	Interruptor (2 *bits*=intermediário, aberto, fechado e falha)
Loc	SPS	Operação local

Para definir um dado IEC, seu nome deve ser composto pelo dispositivo lógico referente, nó lógico, dado e os atributos definidos pelo CDC. Por exemplo, como demonstrado na Figura 10, o dispositivo lógico é o segundo relé do sistema, o nó lógico refere-se a um fusível, o dado designa que trata-se de uma operação local e o atributo refere-se ao valor do *status*.

2.6 CRIPTOGRAFIA

O elevado nível de ataques a estruturas *Smart Grids* SCADA utilizando protocolos abertos no últimos anos incentivou a busca por tecnologias que ampliassem a segurança desses sistemas. Sendo assim, foram realizadas diversas linhas de estudos, uma dessas basicamente seguindo dois conceitos principais: o posicionamento de uma camada de segurança diretamente no protocolo e a inserção de criptografia sobre o dado transmitido.

O propósito da criptografia é proporcionar sigilo a uma comunicação (VAUDENAY, 2006). Há indícios de que a criptografia surgiu há milhares de anos, por exemplo, a primeira referência a uma ferramenta de criptografia encontrada é de cerca de 700 a.C. (RISTIC, 2013). No entanto, o formato atual da criptografia surgiu por volta do século XX, focada no uso militar (RISTIC, 2013).

Para a criptografia ser efetiva é necessário um algoritmo de criptografia e uma chave forte (THOMAS, 2000). Um algoritmo de criptografia significativo normalmente gera um texto cifrado aleatoriamente, que inviabiliza a análise do invasor para revelação de informações do sistema (RISTIC, 2013). Um exemplo fraco de algoritmo é a *cifra de substituição*, formato que refere-se a um modo de substituição de letras compostas numa mensagem, por exemplo. Nesse caso, o invasor com acesso a mensagem é capaz substituir o texto cifrado até encontrar uma frequência semelhante as letras no inglês, encontrando um padrão e assim, desvendando o conteúdo da mensagem (OPPLIGER, 2016). Em uma cifra aprimorada, a única opção para o invasor decifrar a mensagem é testando todos os padrões de chaves para decriptação possíveis, esse formato é denominado pesquisa exaustiva de chaves (PAAR; PELZL, 2009). Portanto, a segurança do texto cifrado é exclusivamente associada a força da chave. Com isso, se a chave possui um tamanho considerável e a quebra da criptografia demanda de uma iteração de um número extremamente significativo de chaves possíveis, pode-se declarar que a cifra é computacionalmente segura (RISTIC, 2013).

A criptografia tornou-se uma tecnologia interessante para proteção de sistemas SCADA, porque quando desenvolvida corretamente, pode atender aos requisitos básicos de segurança: confidencialidade, autenticidade e integridade do dado transmitido (THOMAS, 2000). A criptografia pode ser dividida em dois ramos principais: criptografia simétrica e criptografia assimétrica (PAAR; PELZL, 2009). Além disso, existem também os protocolos criptográficos, como o TLS, em que esse trabalho é baseado. Na seção seguinte será abordado especificamente sobre esse formato. A seguir será apresentada uma breve introdução aos formatos de criptografia simétrica e assimétrica.

2.6.1 Criptografia simétrica

A criptografia simétrica refere-se a um método de ofuscação que viabiliza a transmissão segura de informações por meio do uso de canais de comunicação inseguros (RISTIC, 2013). Além disso, a comunicação entre as duas extremidades nesse formato basicamente funciona por meio do compartilhamento de uma chave secreta entre o receptor e transmissor (PAAR; PELZL, 2009). O nome técnico de criptografia simétrica relaciona-se ao fato de ambas as extremidades conhecerem a mesma informação secreta (THOMAS, 2000).

O formato de criptografia simétrica é descrito na Figura 11. Inicialmente, para estabelecer a comunicação segura é necessário que o Usuário 1 e 2 autorizem o algoritmo de criptografia e uma chave secreta. Posteriormente, no momento que o Usuário 1 desejar enviar uma mensagem ao Usuário 2, o primeiro utilizará a chave secreta para encriptar a informação. Em seguida, o Usuário 2 utiliza a mesma chave para fazer a decriptação da mensagem para que assim possa receber a informação integralmente. Caso um Usuário 3 obtenha ingresso a conexão, o mesmo poderá ter acesso ao pacote enviado. Porém não conseguirá decriptar a mensagem devido ao fato de não possuir a chave secreta.

Os algoritmos, ou cifras de criptografia, baseados na técnica de criptografia simétrica normalmente referem-se a transformações matemáticas. Essas transformações são realizadas nas informações pretendidas a serem criptografadas, juntamente com a chave secreta (THOMAS, 2000).

A eficácia de uma cifra refere-se diretamente ao tamanho da chave secreta (OPPLIGER, 2016). Por exemplo, em um algoritmo composto com uma chave de 2 *bits*, somente seria possível a utilização de quatro chaves. Portanto, caso um invasor obtivesse acesso ao dado criptografado, o mesmo poderia tentar as quatro chaves possíveis

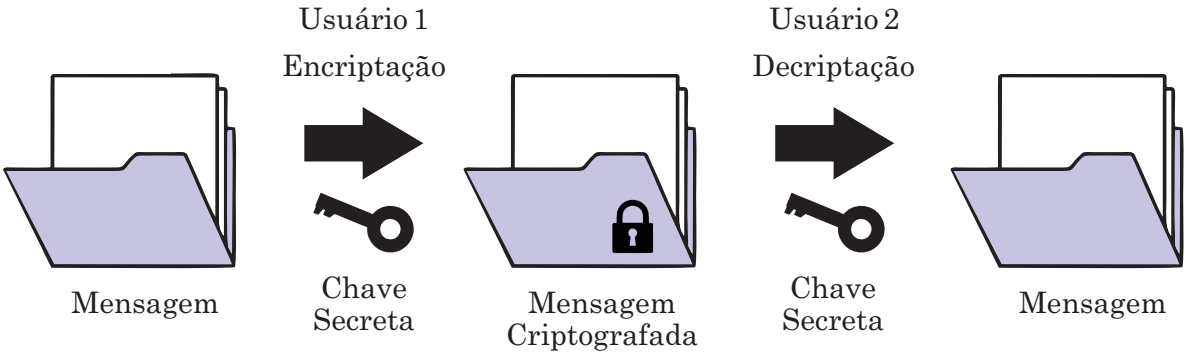

Figura 11: Formato de criptografia simétrica.

Fonte: O Autor (2021).

e assim ter acesso integralmente ao dado transmitido (THOMAS, 2000). Dessa forma, para proporcionar uma segurança significativa ao sistema submetido a criptografia, é fundamental que as chaves secretas contenham um tamanho considerável.

Os algoritmos de criptografia são categorizados de acordo com o formato de processamento dos dados de entrada, podendo ser classificados como: algoritmos de fluxo ou de bloco (RISTIC, 2013). Os algoritmos de fluxo atuam processando as informações de entrada um *byte* de cada vez e são capazes de admitir qualquer tamanho de entrada para criptografia (THOMAS, 2000).

Os algoritmos de bloco, no entanto, atuam em informações de entrada de tamanho fixo, normalmente compostas por 8 *bytes* (OPPLIGER, 2016). Dessa forma, esse formato permite a utilização de um menor recurso de computação, além de possuir uma menor vulnerabilidade a ataques (THOMAS, 2000). No entanto, o uso desse formato é desfavorável, devido ao fato dos dados de entrada raramente possuírem tamanhos semelhantes ao bloco da cifra.

Para criptografar um algoritmo de bloco é necessário que as informações sejam divididas em blocos, e caso o último bloco não esteja no tamanho certo é preciso adicionar um *padding* (RISTIC, 2013). Além disso, as cifras de bloco necessitam de um vetor de inicialização para agir nos dados de *padding*, de modo a iniciar o processo de criptografia (THOMAS, 2000). Esse vetor organiza o algoritmo de acordo com os dados irrelevantes, proporcionando para que a cifra atue com força total antes que a mensagem original seja exibida (OPPLIGER, 2016).

A criptografia simétrica é voltada a sistemas com quantidades consideráveis de dados e velocidades altas (PAAR; PELZL, 2009). Porém, não é muito efetivo quando o

número de partes envolvidas na conexão aumenta (RISTIC, 2013). Nesse caso, o formato de criptografia assimétrica é recomendado.

2.6.2 Criptografia assimétrica

Em um sistema de criptografia assimétrica são utilizadas chaves públicas e privadas (PAAR; PELZL, 2009). Assim, cada extremidade têm sua chave privada, que é exclusiva dessa parte, e também uma chave pública, que é compartilhada abertamente com todas as extremidades do sistema. Cada uma das extremidades utiliza chaves separadas, uma para criptografar o dado e outra para descriptografar (OPPLIGER, 2016).

A criptografia assimétrica é fundamentada em problemas matemáticos fáceis de gerar, porém complexos para a resolução (THOMAS, 2000). Para exemplificar, supondo um problema hipotético, encontre a resolução do produto entre 140 e 36, o resultado é 5040, simples de ser obtido. Porém, tendo somente o número 5040, é difícil encontrar os dois números inteiros originais que multiplicados geraram o 5040. Portanto, essa é uma analogia básica ao formato de criptografia assimétrica.

A Figura 12 demonstra como funciona o formato de criptografia assimétrico. No momento que o Usuário 2 deseja uma comunicação segura com o Usuário 1 são geradas duas chaves, a secreta que fica somente com o Usuário 2 e a pública, distribuída abertamente. O Usuário 1 identifica a chave pública e a utiliza para criptografar os dados de envio. Em seguida, o Usuário 1 envia a mensagem criptografada pela chave pública do destinatário. Esse então decifra a mensagem utilizando sua chave privada. Como somente o Usuário 2 possui sua chave privada, apenas o próprio é capaz de decriptar o dado corretamente.

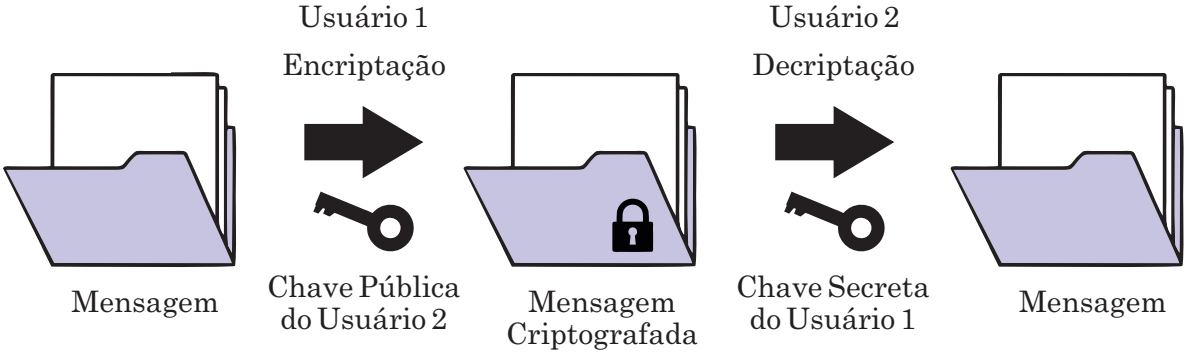

Figura 12: Formato de criptografia assimétrica.

Fonte: O Autor (2021).

Além dos formatos citados, há também a possibilidade de implementar protocolos criptográficos. Esse formato opera em aplicações de algoritmos de criptografia. Protocolos comuns em comunicações seguras na internet utilizam os blocos de construção simétrico e assimétrico (RISTIC, 2013). O TLS é um exemplo de protocolo criptográfico amplamente utilizado em aplicações baseadas em TCP/IP (PAAR; PELZL, 2009).

2.7 AUTENTICAÇÃO

A autenticação refere-se ao método de confirmação da identidade do usuário ou entidade da conexão (BREWER, 2005). Além disso, esse formato de segurança é amplamente aplicado para que um invasor não seja capaz de se passar por um usuário legítimo da conexão. Desse modo, em um sistema SCADA, a autenticação torna-se uma ferramenta de proteção importante, pois se um usuário malicioso conseguir o acesso a rede do sistema, o mesmo é capaz de se passar por uma das extremidades e obter dados essenciais, bem como repassar informações errôneas sobre a estrutura. Sendo assim, com o formato de autenticação requerido, o usuário malicioso ainda pode ter acesso a conexão. Porém, os dados obtidos pelo invasor tornam-se incompreensíveis, dificultando assim a sua capacidade de realizar alterações nos dados ou mesmo obter informações críticas sobre a estrutura.

O método de autenticação age estabelecendo uma barreira no acesso ao meio. Esse bloqueio é estabelecido de várias formas, como pelo uso de certificados digitais no sistema, por exemplo. A metodologia para efetuar a autenticação são dividas em dois conjuntos, a autenticação de mensagem e autenticação de usuário. Além disso, existem diversos métodos para implementar a autenticação em uma conexão, como a inclusão de protocolos de autenticação (HAHN *et al.*, 2005), como o TLS, por exemplo (VIEGA; MESSIER, 2003).

2.7.1 Autenticação de mensagem

A autenticação da mensagem baseia-se na verificação da informação enviada, se realmente foi enviado pelo remetente exposto. Além disso, é voltada a análise do conteúdo da mensagem, para definir se o mesmo foi modificado. Sendo assim, se o dado foi alterado, também é responsável pela verificação da alteração, se foi ocasionada acidentalmente ou propositalmente.

O foco da autenticação da mensagem está na identidade e na origem, devido ao fato de que essas características estão diretamente associadas aos controles de acesso e segurança de uma rede (BREWER, 2005). A autenticação trata inteiramente da aceitação da requisição de um usuário da conexão, se o mesmo possui um nível suficiente de autenticidade para acesso a conexão, sistemas ou aplicações (BREWER, 2005), como é o caso da estrutura SCADA, por exemplo. As principais técnicas usadas na autenticação de mensagem referem-se a autenticação MAC e assinatura digital.

- A autenticação MAC refere-se a uma função unidirecional que capta a mensagem e uma chave secreta composta em aplicações voltadas a assegurar a integridade dos dados, não necessitando da privacidade e que só é possível reconhecer o dado de saída com a posse da chave.
- A assinatura digital refere-se ao formato de autenticação que representa uma assinatura real. Essa assinatura pode ser produzida pelo remetente da mensagem, porém qualquer usuário é capaz de identificar quem foi o emissor.

2.7.2 Autenticação de usuário

A verificação de um usuário estabelece se a identidade digital requerida é destinada a alguém que realmente possui os dados de identidade precisos (BREWER, 2005). Se a identidade for confiável suficientemente, um nome de usuário é atribuído ao requerente. Esse nome define a identidade digital do usuário. Além disso, é realizada uma correlação entre a identidade e a rede em questão para admitir o acesso do usuário a sistemas, aplicativos e dados presentes na conexão.

Segundo (BURNETT, 2006), é de responsabilidade do usuário apresentar algo que o identifique. Essa representação é definida em três categorias:

- **Algo próprio:** refere-se a alguma característica própria física ou comportamental, que geralmente nunca muda. Isso pode ser relacionado a biometria, impressão digital ou retina (íris), entre outros (BURNETT, 2006).
- **Algo que possui:** relaciona-se a qualquer dispositivo que somente possa estar em um lugar por vez, que seja capaz de identificar o usuário. Esse dispositivo pode ser um cartão magnético, uma chave USB, um cartão inteligente, entre outros (BURNETT, 2006).

- **Algo conhecido:** pode ser relacionado a uma senha capaz de ser gerada a qualquer momento para o sistema de autenticação. A senha é um elemento crucial para um sistema de segurança e não pode ser negligenciada. No entanto, para ampliar o formato de segurança é possível implementar outros fatores adicionais a senha, como por exemplo, somente possuir a senha não é o ideal caso você não possua o cartão relativo a senha em questão.

2.7.3 Certificados digitais

Certificados digitais compostos por pares de chaves públicas e privadas relacionam-se a outro modo de efetuar a autenticação por um nome de usuário (VIEGA; MESSIER, 2003). Desse modo, um terceiro envolvido é responsável pela emissão dos certificados digitais para ambas as extremidades (usuários ou dispositivos) submetidas a transação, resultando em uma verificação da identidade dos canais de conexão (BREWER, 2005).

Um uso regular para os certificados digitais é a autenticação de um endereço da *web*. Desse modo, o servidor da Web possuirá um certificado digital composto por uma chave pública (HAHN *et al.*, 2005). O dispositivo pertencente ao usuário final possuirá um certificado digital que é requerido apenas no momento que a senha correta é inserida, sendo assim o serviço da Web assegura a identidade do usuário (BREWER, 2005).

Para utilizar o certificado digital como um procedimento para a autenticação de modo a verificar se um usuário final característico está presente na outra extremidade, um certificado é carregado diretamente no dispositivo do usuário. Sendo assim, o certificado é iniciado para uso na comunicação entre o usuário e um servidor Web *host* composto por um PIN ou senha, o que permite confirmar que ao menos há uma pessoa no dispositivo que reconhece a senha. Desse modo, a expectativa é de que o usuário refira-se a pessoa em questão (BREWER, 2005).

2.8 PROTOCOLO TLS

Tecnologias responsáveis por ampliar a segurança de sistemas industriais tem se tornado importantes na atualidade. Um fator relevante para isso foi o aumento de sistemas conectados a Internet e consequentemente, a crescente de ataques difundidos contra essas arquiteturas.

Toda estrutura industrial possui dados privados, bem como informações que não podem ser modificadas, pois com a alteração de somente um dado é possível causar

danos significativos em toda a arquitetura. Portanto, aplicar uma segurança adicional ao sistema é algo relevante, ainda mais em modelos que não possuem suporte a segurança, como é o caso de estruturas industriais baseadas no protocolo DNP3.

O protocolo DNP3 foi amplamente difundido em setores de energia, principalmente em sistemas SCADA aplicados a Redes Elétricas Inteligentes. Durante o desenvolvimento do padrão, no entanto, não foi pensando na autenticação de identidade, criptografia dos dados e controle de acesso (MAJDALAWIEH *et al.*, 2007). O que viabilizou diversos ataques focados a estrutura *Smart Grids* SCADA.

Os ataques podem causar vários danos em um sistema SCADA que utiliza o protocolo DNP3. As alterações vindas de invasores podem gerar intermitências na rede, roubo de eletricidade, perda da taxa de resposta do sistema, supressão do suprimento de energia, mau funcionamento dos dispositivos, blecaute do sistema, entre outros.

A preocupação com a segurança dos dados em sistemas compostos com o protocolo DNP3 proporcionou o estudo de diversos formatos capazes de atender essa ausência de proteção. Os modelos desenvolvidos para suprir essa demanda são focados basicamente na criptografia de cada extremidade do meio de comunicação ou também voltados ao aperfeiçoamento de segurança posicionados diretamente no protocolo utilizado (MAJDALAWIEH *et al.*, 2007). Com isso, diversos métodos foram desenvolvidos para suprir essa necessidade, como o DNP3-SA (*Distributed Network Protocol Version 3 Secure Authentication*), formato que dedica-se a autenticação somente, sem o adicional de criptografia ou outras medidas de segurança (GILCHRIST, 2008), o DNPSec (*Distributed Network Protocol Version 3 Security Framework*) (MAJDALAWIEH *et al.*, 2007), que amplia a segurança adicionando criptografia ao protocolo (FARUK, 2008) e também o denominado DNP3 sob TLS, que acrescenta autenticação e criptografia ao protocolo DNP3, formato no qual o presente trabalho é baseado.

O *Transport Layer Security* (TLS) refere-se a um protocolo da camada de transporte, que fundamenta seu princípio na comunicação segura de estruturas baseadas em TCP/IP (OPPLIGER, 2016). O protocolo foi lançado oficialmente em 1999, baseado na versão 3 do SSL (*Secure Sockets Layer*), o que pode ser analisado claramente, pois as semelhanças entre ambos eram consideráveis (RISTIC, 2013).

A Versão 1.2 do TLS lançada em 2008, acrescentou o suporte à criptografia autenticada, o que tornou o protocolo flexível (RISTIC, 2013). A criptografia é a base da comunicação segura (PAAR; PELZL, 2009). Com isso, o TLS foi amplamente utilizado

em diversos protocolos de aplicação, como o HTTP (HTTPS), SMTP, e até mesmo em algumas aplicações de VPN (OPPLIGER, 2016).

O propósito definido durante o desenvolvimento do protocolo era especificamente prover comunicação segura por meio de infraestruturas inseguras (RISTIC, 2013). Portanto, isso significa que quando implementado corretamente, o protocolo TLS é capaz de fornecer um canal de comunicação para um serviço arbitrário na Internet, transmitindo dados seguros com o servidor desejado (RISTIC, 2013).

O protocolo TLS utiliza as formas assimétrica e simétrica de criptografia. No TLS, a forma assimétrica é direcionada para assegurar a autenticidade, e a simétrica para a confidencialidade e integridade dos dados.

O protocolo TLS é um protocolo criptográfico no formato cliente-servidor que é empilhado sobre um protocolo confiável da camada de transporte, como o TCP, no caso da pilha de protocolos TCP/IP, por exemplo (OPPLIGER, 2016). O protocolo TLS pode ser posicionado na camada apresentação do modelo OSI (RISTIC, 2013) ou como uma camada intermediária, posicionado entre a camada de aplicação e transporte do modelo Internet (OPPLIGER, 2016).

A Figura 13 apresenta o modelo Internet, bem como o formato do protocolo TLS. A camada inferior do TLS, engloba o protocolo `TLS Record`, que é utilizado para o encapsulamento das informações do protocolo da camada superior (RISTIC, 2013). A camada superior é formada sobre quatro protocolos:

- `TLS Handshake`: o principal protocolo do TLS, proporciona que os pares comunicantes autentiquem-se uns com os outros, negociando um conjunto de criptografia e um modelo de compactação usado para a comunicação (OPPLIGER, 2016).
- `TLS Change Cipher Spec`: proporciona aos pares comunicantes sinalizar uma mudança na estratégia de cifra e no modo como as informações são protegidas criptograficamente (OPPLIGER, 2016). O `TLS Handshake` é aplicado para negociar parâmetros de segurança e o `Change Cipher Spec` é voltado para posicionar esses parâmetros no lugar e efetivá-los.
- `TLS Alert`: possibilita aos pares comunicantes sinalizar possíveis problemas e gerar mensagens de alerta respectivas.
- Dados de aplicação: esse campo é utilizado para prover a transmissão segura dos dados de aplicação. Esse elemento atua capturando dados da camada superior e os

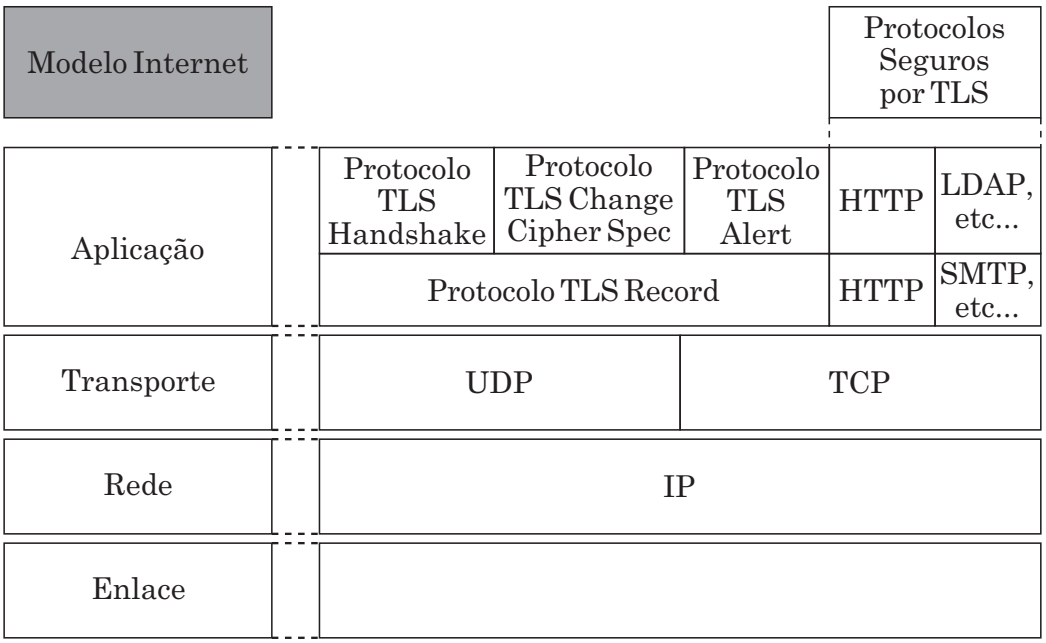

Figura 13: Modelo da Internet, sub-camadas e sub-protocolos TLS.

Fonte: Adaptado de (OPPLIGER, 2016).

fortalece com o `TLS Record`, para assim possuir proteção criptográfica e transmissão segura (RISTIC, 2013).

O protocolo TLS provê comunicação segura e autenticada para diversos serviços na Internet. Esses serviços são baseados em diversos protocolos, como o IMAP (*Internet Message Access Protocol*), POP3 (*Post Office Protocol Version 3*), FTP (*File Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*), HTTP e LDAP (*Lightweight Directory Access Protocol*), conforme demonstrados na Figura 14. Um exemplo apresentado na Figura 14, refere-se a adição de TLS ao protocolo HTTP sob TCP provendo assim, uma comunicação segura, representada pelo protocolo HTTPS. Além disso, é definido uma porta TCP padronizada para o protocolo HTTPS, a porta 443.

2.8.1 Protocolo TLS *Record*

O protocolo `TLS Record` é responsável por transportar e criptografar as informações de camadas inferiores transmitidas em uma conexão (RISTIC, 2013). Além disso, é destinado a encapsular todas as mensagens da camada superior (OPPLIGER, 2016).

O protocolo TLS possui um cabeçalho curto composto por 5 *bytes* de comprimento, que contém informações sobre o tipo de conteúdo, versão do protocolo e tamanho, que precedem a seção de dados do protocolo (RISTIC, 2013). A Figura 15 demonstra como é formado o protocolo `TLS Record`.

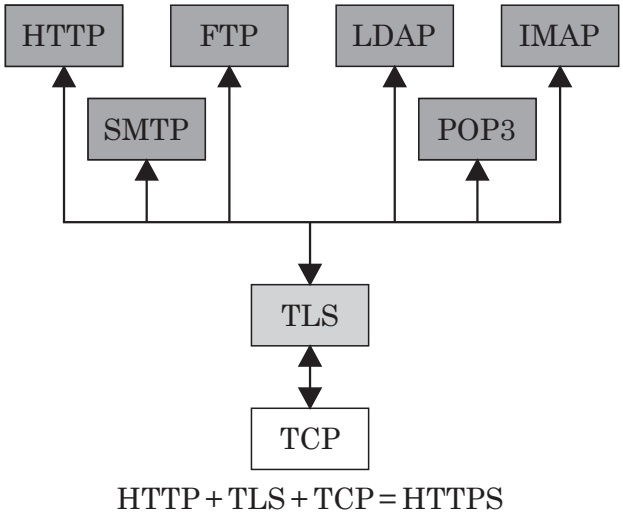

Figura 14: Protocolos de uso frequente protegidos pelo protocolo TLS.

Fonte: O Autor (2021).

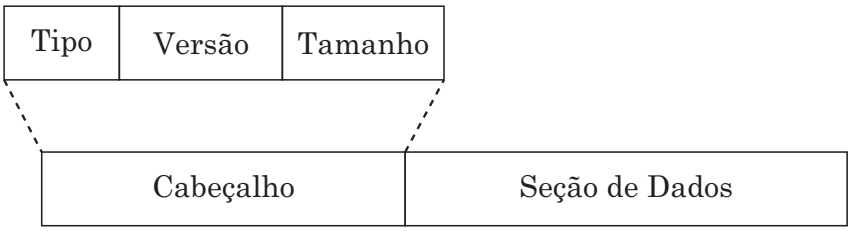

Figura 15: Formato do campo TLS Record.

Fonte: Adaptado de (OPPLIGER, 2016).

O protocolo `TLS Record` é uma abstração útil que é responsável por diversos dados relevantes, como o transporte de mensagens, criptografia e validação de integridade, compressão e extensibilidade (RISTIC, 2013; OPPLIGER, 2016). A criptografia é opcional, e o elemento de compressão não é mais utilizado desde 2012, por prover insegurança (RISTIC, 2013). Abaixo será discutido brevemente sobre esses dados.

- Transporte de Mensagens: o `TLS Record` atua transportando *buffers* de dados opacos, que são submetidos a esse campo por outras camadas do protocolo. Se caso o *buffer* tiver um comprimento maior que o limite (16.384 *bytes*), o TLS Record faz a fragmentação em partes menores. Além disso, o TLS Record também converte *buffers* pequenos em um único *buffer*, que não exceda o limite de comprimento (THOMAS, 2000).

- Criptografia e validação de integridade: quando é realizada uma nova conexão, os dados são trocados sem segurança. Nesse caso, é utilizado o conjunto de cifras `TLS_NULL_WITH_NULL_NULL` (OPPLIGER, 2016). Isso se faz necessário para indu-

zir a primeira negociação. Entretanto, no momento que o *handshake* é concluído, a camada `TLS Record` começa a prover criptografia e a validação de integridade (OPPLIGER, 2016).

- Extensibilidade: o protocolo `TLS Record` têm a função do transporte e da criptografia das informações, mas delega outras ações aos subprotocolos. Isso faz com que o protocolo TLS torne-se extensível, porque permite a inclusão de novos subprotocolos facilmente. Além disso, com a criptografia aplicada, todos os subprotocolos possuem proteção devido a definição dos parâmetros de conexão (RISTIC, 2013).

Uma visão geral sobre o processamento do protocolo `TLS Record` é descrito na Figura 16. Na primeira etapa, a fragmentação, o protocolo fragmenta as informações da camada superior em blocos de até 214 *bytes* e cada bloco é compactado em uma estrutura `TLSPlaintext` (OPPLIGER, 2016). Nesse caso, os limites da mensagem do cliente não são mantidos, o que pode acarretar em várias mensagens de mesmo tipo a uma estrutura `TLSPlaintext` (THOMAS, 2000). Na segunda etapa, a compressão, o TLS compacta o `TLSPlaintext` conforme o método de compactação declarado no estado de sessão TLS, esse método é estipulado nulo, definindo a compressão um modo opcional (OPPLIGER, 2016).

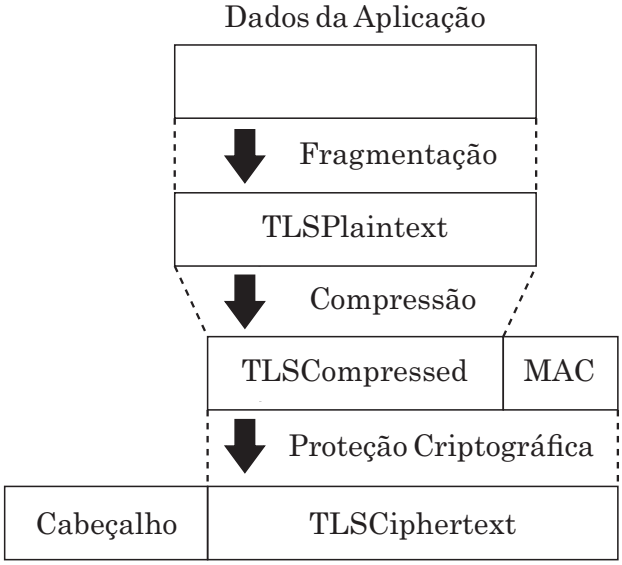

Figura 16: Visão geral do processamento do TLS Record.

Fonte: Adaptado de (OPPLIGER, 2016).

Na etapa de proteção criptográfica, o protocolo `TLS Record` opera a segurança de uma estrutura conforme a `CipherSpec` estabelecida no estado de sessão TLS (OPPLIGER, 2016). O termo `CipherSpec` refere-se a um par de algoritmos utilizados para efetuar

a segurança de informações de forma criptografada (THOMAS, 2000). Essa etapa consiste na autenticação de mensagens e um algoritmo capaz de criptografar as informações (RISTIC, 2013). A `CipherSpec` é integrada a um algoritmo de troca de chaves, formando assim as suítes criptográficas (OPPLIGER, 2016). Para o TLS são definidos trinta e uma suítes criptográficas possíveis, demonstrados na Tabela 4, a tabela apresenta as suítes criptográficas, cifras, trocas de chave e *hash* utilizados nos trinta e um modelos. Por exemplo, `TSL_DH_RSA_WITH_3DES_CBC_SHA` refere-se a suíte criptográfica que utiliza a assinatura de troca de chave de *Diffie-Hellmann* composta pelo algoritmo `RSA`, que será responsável por assegurar a autenticidade das extremidades, indica também que as informações serão criptografadas utilizando a cifra `3DES` em modo `CBC` e `SHA` informa que a construção `HMAC` terá como base a função de dispersão criptográfica, ou *hash function* `SHA-1` (THOMAS, 2000).

Tabela 4: Suítes criptográficas TLS.

Suíte Criptográficas	Troca de Chave	Cifra	Hash
TLS_NULL_WITH_NULL_NULL	NULL	NULL	NULL
TLS_RSA_WITH_NULL_MD5	RSA	NULL	MD5
TLS_RSA_WITH_NULL_SHA	RSA	NULL	SHA
TLS_RSA_EXPORT_WITH_RC4_40_MD5	RSA_EXPORT	RC4_40	MD5
TLS_RSA_WITH_RC4_128_MD5	RSA	RC4_128	MD5
TLS_RSA_WITH_RC4_128_SHA	RSA	RC4_128	SHA
TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5	RSA_EXPORT	RC2_CBC_40	MD5
TLS_RSA_WITH_IDEA_CBC_SHA	RSA	IDEA_CBC	SHA
TLS_RSA_EXPORT_WITH_DES40_CBC_SHA	RSA_EXPORT	DES40_CBC	SHA
TLS_RSA_WITH_DES_CBC_SHA	RSA	DES_CBC	SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA	RSA	3DES_EDE_CBC	SHA
TLS_DH_DSS_EXPORT_WITH_DES40_CBC_SHA	DH_DSS_EXPORT	DES40_CBC	SHA
TLS_DH_DSS_WITH_DES_CBC_SHA	DH_DSS	DES_CBC	SHA
TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA	DH_DSS	3DES_EDE_CBC	SHA
TLS_DH_RSA_EXPORT_WITH_DES40_CBC_SHA	DH_RSA_EXPORT	DES40_CBC	SHA
TLS_DH_RSA_WITH_DES_CBC_SHA	DH_RSA	DES_CBC	SHA
TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA	DH_RSA	3DES_EDE_CBC	SHA
TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	DHE_DSS_EXPORT	DES40_CBC	SHA
TLS_DHE_DSS_WITH_DES_CBC_SHA	DHE_DSS	DES_CBC	SHA
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA	DHE_DSS	3DES_EDE_CBC	SHA
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	DHE_RSA_EXPORT	DES40_CBC	SHA
TLS_DHE_RSA_WITH_DES_CBC_SHA	DHE_RSA	DES_CBC	SHA
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	DHE_RSA	3DES_EDE_CBC	SHA
TLS_DH_anon_EXPORT_WITH_RC4_40_MD5	DH_anon_EXPORT	RC4_40	MD5
TLS_DH_anon_WITH_RC4_128_MD5	DH_anon	RC4_128	MD5
TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA	DH_anon	DES40_CBC	SHA
TLS_DH_anon_WITH_DES_CBC_SHA	DH_anon	DES_CBC	SHA
TLS_DH_anon_WITH_3DES_EDE_CBC_SHA	DH_anon	3DES_EDE_CBC	SHA
TLS_FORTEZZA_KEA_WITH_NULL_SHA	FORTEZZA_KEA	NULL	SHA
TLS_FORTEZZA_KEA_WITH_CBC_SHA	FORTEZZA_KEA	FORTEZZA_CBC	SHA
TLS_FORTEZZA_KEA_WITH_RC4_128_SHA	FORTEZZA_KEA	RC4_128	SHA

A proteção criptográfica do protocolo TLS Record é composta por autenticação e criptografia dos dados (RISTIC, 2013). Essa proteção é realizada em três opções, na primeira é autenticada a mensagem e em seguida criptografada e enviado o texto cifrado composto pelo MAC ao destinatário (THOMAS, 2000). Na segunda opção, a mensagem

é criptografada e em seguida é realizada a autenticação do texto cifrado e posteriormente é realizado o envio do texto cifrado juntamente com o MAC ao destinatário (OPPLIGER, 2016). O terceiro formato refere-se inicialmente na criptografia da mensagem, em seguida autenticação da mensagem e por último o envio do texto cifrado junto ao MAC para o destinatário (OPPLIGER, 2016).

2.8.2 Protocolo TLS *Handshake*

O protocolo `TLS Handshake` é o responsável pela negociação das sessões TLS (THOMAS, 2000). Toda conexão TLS depende exclusivamente de um *handshake* para se concretizar (RISTIC, 2013). Esse protocolo depende exclusivamente da camada `TLS Record` para realizar o encapsulamento de suas mensagens (THOMAS, 2000).

Durante o processo de *handshake*, o cliente e o servidor exercerão quatro funções principais: trocar recursos e concordar com os parâmetros de conexão pretendidos; validar o(s) certificados(s) exposto(s) ou autenticar de acordo com o meio determinado; autorizar a chave compartilhada; e verificar se as mensagens de *handshake* não foram alteradas por usuários desconhecidos (RISTIC, 2013). A segunda e terceira função, geralmente são exercidas em uma única etapa, denominada troca de chaves (OPPLIGER, 2016).

O protocolo `TLS Handshake` porporciona ao cliente e o servidor uma autenticação entre ambos e assim uma negociação de itens, como por exemplo suítes criptográficas e modelos de compressão (THOMAS, 2000). O funcionamento do protocolo `TLS Handshake` é demonstrado na Figura 17. A mensagem `ClientHello` enviada pelo cliente é gerada para iniciar o processo de *handshake*, enviando os recursos do cliente ao servidor (RISTIC, 2013). Essa mensagem é composta pela versão mais atual suportada pelo cliente (geralmente é a versão 3.0), um número aleatório gerado pelo cliente de 32 *bytes*, o identificador de uma sessão anterior, uma lista composta por quais suítes criptográficas são suportadas pelo cliente e uma lista de modelos de compressão e extensões suportadas pelo cliente (OPPLIGER, 2016).

O servidor retorna a mensagem `ServerHello`, que define a seleção dos parâmetros de conexão pelo servidor (RISTIC, 2013). Esse dado possui uma estrutura semelhante a mensagem `ClientHello`, a diferença é que possui somente a suíte criptográfica e o método de compressão selecionados e não uma lista como na mensagem comparada (OPPLIGER, 2016). Se o servidor efetuar a autenticação, o que é comum, ele pode enviar

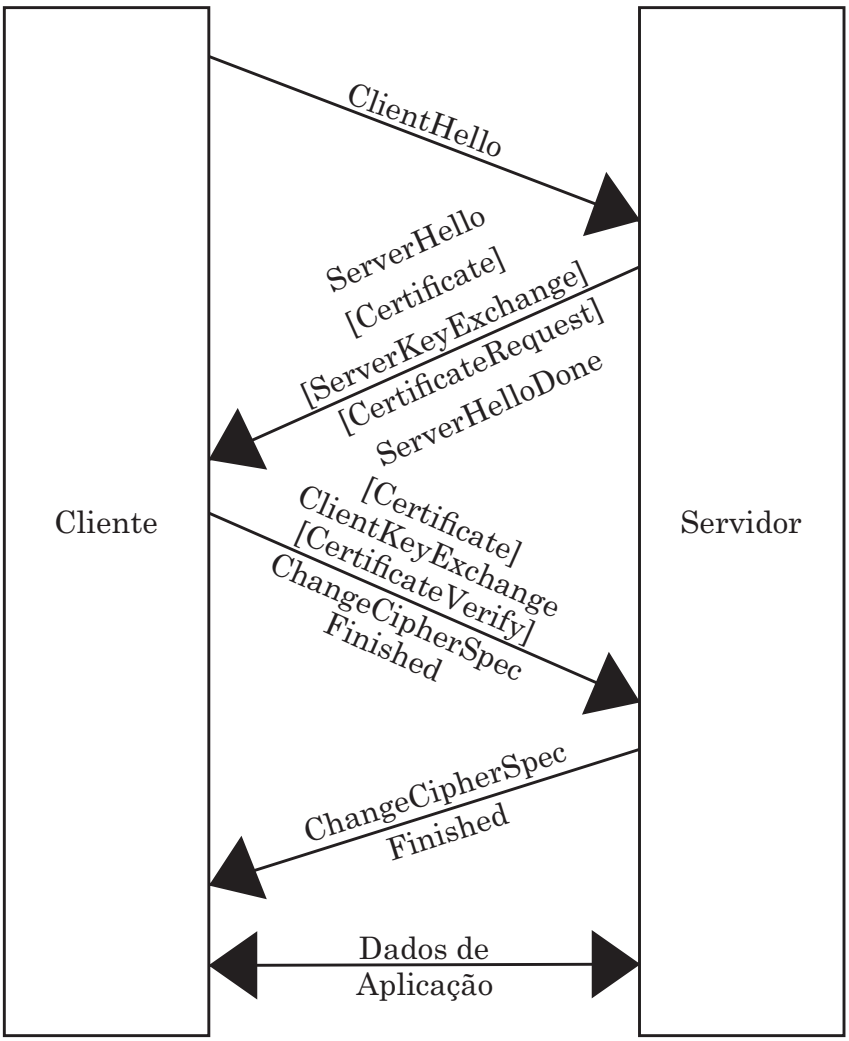

Figura 17: Processo de Handshake do TLS.

Fonte: Adaptado de (OPPLIGER, 2016).

uma mensagem composta pelos seus certificados, denominada `Certificate` (THOMAS, 2000). A mensagem `ServerKeyExchange` é enviada quando o modelo de troca de chaves, definido pela suíte criptográfica, requisitar dados a mais do que os presentes nos certificados, que são alterados de acordo com modelo de troca de chaves aplicado (OPPLIGER, 2016). A mensagem `CertificateRequest` é enviada dependendo da troca de chaves selecionada. Assim, o servidor envia a informação para gerar o *master secret* da conexão (RISTIC, 2013). Além disso essa mensagem é composta por uma lista de certificados reconhecidos, responsáveis pela certificação e algoritmos suportados. Após isso, o servidor envia a mensagem `ServerHelloDone` finalizando a troca de informações iniciais (THOMAS, 2000).

Após essa troca de mensagens iniciais, o cliente envia novamente algumas informações ao servidor. A primeira mensagem enviada é `Certificate`, em que é composta

por informações adicionais para poder efetuar a geração do *master secret* da conexão (RISTIC, 2013). O cliente também troca o modo de criptografia e envia a informação ao servidor. O conteúdo dessa mensagem depende do algoritmo de troca de chaves utilizado (OPPLIGER, 2016). Se o cliente enviou um certificado ao servidor, é necessário também o envio da mensagem `CertificateVerify`, dado que é composto pela chave privada correspondente a chave pública do certificado (THOMAS, 2000). Na sequência, o cliente envia a mensagem `ChangeCipherSpec`, composta pelo MAC dos dados de *handshake* enviados e recebidos (RISTIC, 2013), que também é responsável por indicar que as informações recebidas são capazes de gerar os parâmetros e chaves, sinalizando também que o formato seguro do canal está disponível (THOMAS, 2000). É importante ressaltar que a mensagem `ChangeCipherSpec` faz parte do protocolo TLS `ChangeCipherSpec` (OPPLIGER, 2016). Dessa forma, essa mensagem é transmitida por um quadro único do protocolo `TLS Record` e assim, os dados enviados posteriormente a esse estarão submetidos a novos parâmetros negociados (RISTIC, 2013).

O cliente envia então a mensagem `Finished` para o servidor, sendo a primeira mensagem criptografada sob a nova `CipherSpec` (THOMAS, 2000). Essa mensagem é composta pelo valor *hash* de todos os dados transmitidos pelo protocolo `TLS Handshake` até então e também pelo *master secret* da conexão (OPPLIGER, 2016).

O quarto conjunto de mensagens é composto por duas mensagens enviadas do servidor ao cliente. O servidor envia a informação `ChangeCipherSpec`, mudando o formato de criptografia e assim informando ao cliente (RISTIC, 2013). Na sequência o servidor finaliza as mensagens enviando `Finished`, criptografada pela nova `CipherSpec` agora exercida por ambas as extremidades (THOMAS, 2000). A partir desse momento, cliente e servidor podem se comunicar com segurança. Sendo assim, o cliente pode enviar uma mensagem `ClientHello` ao servidor, solicitando uma nova sessão e também o servidor pode enviar uma mensagem `HelloRequest`, requerendo uma renegociação dos parâmetros da sessão (OPPLIGER, 2016).

2.8.3 Protocolo TLS *Alert*

O protocolo `TLS Alert` é responsável por proporcionar aos pares comunicantes trocas de mensagens de alerta (OPPLIGER, 2016). Essas mensagens podem indicar situações atípicas no sistema e o encerramento da conexão (THOMAS, 2000). Cada mensagem de alerta é composta por um nível de alerta e uma descrição (OPPLIGER, 2016). O nível de alerta possui o tamanho de 1 *byte*, em que o valor 1 corresponde a

informação aviso e 2 a fatal (RISTIC, 2013). Se caso houver um nível de alerta não especificado em uma mensagem de erro, o remetente da mensagem pode estabelecer se o dado é ou não fatal (OPPLIGER, 2016). Todas as mensagens de alerta fatal devem ser tratadas de forma especial. Pois após o recebimento, ambas as extremidades encerram a conexão e descartam qualquer dado relacionado (THOMAS, 2000).

A Figura 18 demonstra uma mensagem `TLS Alert`. Essa mensagem é composta de um cabeçalho de 5 *bytes* do `TLS Record`, em que referencia o que é a mensagem, nesse caso tipo 21, que refere-se ao protocolo `TLS Alert` (RISTIC, 2013). O restante do cabeçalho do `TLS Record` é composto por um campo que determina a versão e um campo de tamanho. O comprimento é 2 *bytes*, em que referencia o tamanho do campo nível de alerta e descrição (OPPLIGER, 2016).

Tipo	Versão		Tamanho
21	3	0	0
	Nível	Descrição	
2	1/2		

Figura 18: Mensagem de alerta TLS.

Fonte: Adaptado de (OPPLIGER, 2016).

Um dos setores em que o TLS é extensamente aperfeiçoado em comparação ao SSL refere-se nos procedimentos de notificação de alertas, bem como as proteções reais e possíveis (RISTIC, 2013). O protocolo TLS estabelece aproximadamente o dobro de descrições de alerta em comparativo ao protocolo SSL (THOMAS, 2000).

2.9 TRABALHOS RELACIONADOS

Nessa seção serão apresentados conceitos importantes para o entendimento dos problemas abordados e das soluções propostas neste trabalho.

A maioria dos protocolos aplicados a sistemas de geração distribuída atualmente, não possuem segurança em sua composição, ou ainda, possuem, porém apresentam vulnerabilidades a ataques cibernéticos. Com isso, medidas para aprimorar a segurança dos dados presentes nesses sistemas tornou-se algo essencial. Desse modo, diversos estudos foram elaborados com intenção de prover segurança a essas estruturas. Como, por exemplo, o denominado, `Module-OT` (*Module for Operational Technology*) (HUPP *et al.*, 2020), que refere-se a um módulo de segurança baseado em *hardware* voltado a sistemas de distribuição de energia. Esse módulo atua integrado a um sistema de comunicação na

camada de transporte do modelo OSI, implementando criptografia, autenticação, autorização, gerenciamento de certificados e controle de acesso do usuário. Esse formato de segurança atua basicamente como uma alternativa ao *gateway* proposto neste trabalho.

Outro trabalho relevante sobre inclusão de segurança com base em criptografia refere-se ao `SEDEA` (*State Estimation-Based Dynamic Encryption and Authentication*) (LIU *et al.*, 2017), baseado em estimativas de estado para proteger a comunicação entre o centro de controle e RTU's (Remote Terminal Units) de *Smart Grids*. Esse sistema atua selecionando as medições dos sistemas de potência conectados a rede para gerar chaves criptográficas. Nesse modelo, a chave criptográfica varia conforme são realizadas alterações no sistema de energia. Desse modo, o centro de controle estima novas chaves de todas as RTU's de maneira dinâmica e síncrona. Além disso, a chave criptográfica pode ser calculada em qualquer RTU do sistema. O trabalho atua protegendo o centro de controles e as RTU's provenientes de sistemas de energia em funcionamento. Esse formato de segurança pode trazer ao *gateway* proposto uma segurança adicional, ou seja, o SEDEA pode atuar como um formato de segurança complementar ao desenvolvido nesse trabalho.

2.10 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Nesse capítulo de revisão da literatura foi discutido sobre como é categorizada uma estrutura SCADA, em seguida foi comentado sobre alguns dos principais formatos de ataques utilizados para invadir essas estruturas. Além disso, foram abordados os princípios fundamentais dos protocolos DNP3 e TLS, bem como criptografia e autenticação, que referem-se a formatos amplamente difundidos à ampliação da segurança de conexões. Verificou-se que a aplicação de formatos de segurança como esses são capazes de proporcionar a proteção desejada em sistemas compostos pelo protocolo DNP3. Ainda, foi discutido que a autenticação e criptografia, que está presente no protocolo TLS é capaz de limitar o funcionamento de alguns dos principais formatos de ataque voltados a estruturas *Smart Grids* SCADA. No próximo capítulo será apresentado em etapas o desenvolvimento do sistema SCADA utilizado no presente trabalho. Essas etapas referem-se a implementação da comunicação DNP3, a troca de dados, o mapeamento de dados para atender o perfil IEC61850, bem como a adição da comunicação segura com o protocolo TLS.

3 DESENVOLVIMENTO DA SOLUÇÃO PROPOSTA

Após o surgimento de ataques a sistemas de geração de energia, como o ocorrido Ucrânia em 2015, por exemplo, a preocupação com segurança em sistemas de geração distribuída têm aumentado consideravelmente. Sendo assim, novas soluções baseadas em protocolos como o SEP2.0 surgiram para ampliar a segurança desses sistemas. No entanto, diversas infraestruturas de geração, como a fotovoltaica, foram instaladas até que essas soluções fossem disponibilizadas comercialmente. Em muitos casos as características utilizadas no momento de elaboração desses sistemas perduram ainda até os dias atuais. Dessa forma, é de suma importância fornecer segurança a esses sistemas legados, pois são os que mais possuem vulnerabilidades. Portanto, o presente trabalho consiste no desenvolvimento de um *gateway* de comunicação com a finalidade de prover segurança por meio da aplicação de autenticação e criptografia com o uso do protocolo TLS na troca de dados realizada entre um sistema SCADA e gerenciadores de energia baseado no protocolo DNP3.

Inversores fotovoltaicos com suporte ao protocolo DNP3 são consideralvemente raros. No entanto, a proposta faz parte de um projeto maior, que associa também o desenvolvimento de um gerenciador de energia para fazer a agregação dos dados de vários inversores e convertê-los em um único inversor lógico para repassar a concessionária de energia, além de fornecer o suporte ao DNP3/TLS.

A Figura 19 apresenta um diagrama em blocos do sistema, com o propósito de auxiliar no entendimento geral da proposta desenvolvida nessa dissertação. Dessa forma, foi utilizada a estrutura hierárquica DNP3 para atender a demanda requerida, de um sistema de supervisão posicionado na rede de comunicação da concessionária de energia, atuando como o mestre do sistema. Na presente proposta efetua-se a troca de dados com o gerenciador de energia, posicionado em uma rede externa (cliente), com um *gateway* entre as duas extremidades, para efetuar o interfaceamento dos dados de forma segura, com o uso do protocolo TLS.

Os sistemas de supervisão disponíveis atualmente não possuem suporte ao protocolo TLS, o que prejudica a segurança dessas estruturas. Por isso, o trabalho desen-

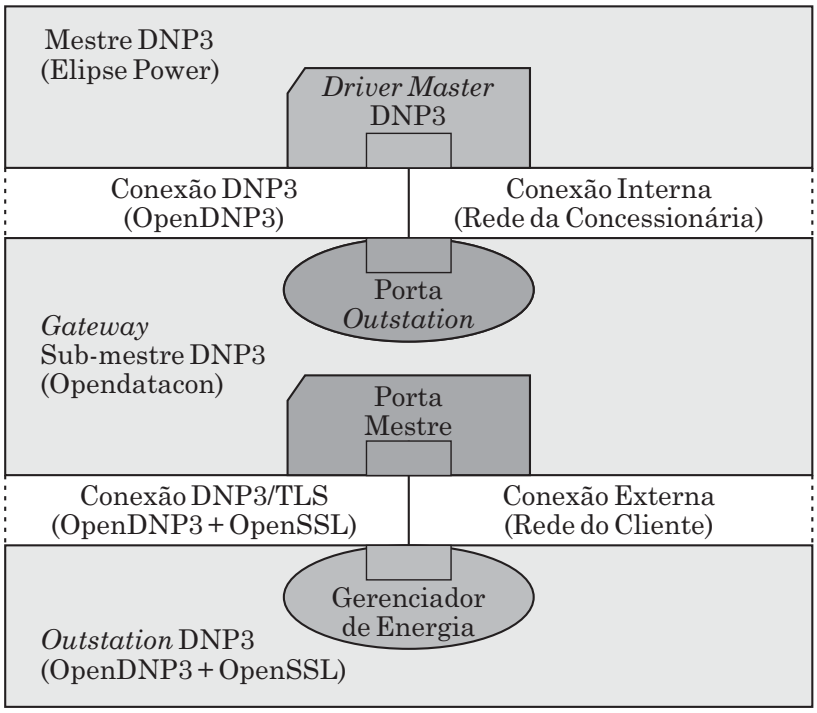

Figura 19: Diagrama do sistema desenvolvido no trabalho.

Fonte: O Autor (2021).

volvido visa incluir essa segurança ausente em arquiteturas pré-existentes, proporcionando uma conexão sem suporte ao TLS somente a uma rede local, dificultando o acesso por usuários mal intencionados a estruturas críticas, como em sistemas de geração distribuída. Isso será possível com o uso do *gateway* desenvolvido, posicionado diretamente na rede da concessionária de energia. O *gateway* fará a interface dos dados entre a concessionária e a estrutura SCADA. O *gateway* desenvolvido, ainda, proporciona uma segurança adicional, pois induz com que as extremidades do sistema conectem-se diretamente a ele. Normalmente, o sistema de geração distribuída seria um servidor da conexão. No entanto, utilizando a arquitetura desenvolvida, o sistema de geração distribuída será um cliente que se conecta ao *gateway*. Da mesma forma, a estrutura SCADA também será um cliente que se conectará ao *gateway*.

O *gateway* desenvolvido é voltado a extremidade da conexão disposta na concessionária de energia, o que proporciona uma diminuição na superfície de ataque. Sendo assim, a única porta de conexão DNP3 com a Internet será a disponibilizada no *gateway*. Dessa forma, as preocupações com segurança passam a ser concentradas na rede da concessionária e não mais em seus clientes.

Como a estrutura é voltada a um sistema de geração distribuída de energia, é necessário que o sistema seja desenvolvido com base em padrões e protocolos exclusivos

para essas arquiteturas. Tal sistema foi desenvolvido durante a execução do projeto de pesquisa e desenvolvimento "PD 2866-0468/2017 - Gerador e inversor inteligente para conexão de sistemas fotovoltaicos em geração distribuída de energia" firmado entre a UTFPR e a ANEEL/COPEL. Portanto, irá seguir perfis utilizados pelas companhias de energia brasileiras. Desse modo, foi estipulado o uso de conexões compostas pelo protocolo DNP3, atendendo aos requisitos do padrão IEC 61850. Para que isso ocorra, inicialmente foi necessário adequar a estrutura SCADA desenvolvida para reconhecimento de dados IEEE 1815/2012. Além disso, por se tratar de uma estrutura crítica, voltada a sistemas de energia, foi necessário adequar os dados trocados no sistema ao perfil recomendado pela IEC 61850.

O sistema de supervisão designado para o projeto, foi o Elipse Power Studio, devido ao fato de possuir suporte à diversos protocolos de comunicação, inclusive a padrões caraterísticos voltados para geração distribuída, como o DNP3, por exemplo. O Elipse Power Studio também cumpre os requisitos da norma IEC 61850, bem como é uma plataforma amplamente utilizada por concessionárias de energia no território nacional devido a quantidade considerável de aplicações suportadas. Além disso, é o sistema padronizado em subestações da COPEL, empresa apoiadora do projeto de P&D em que o presente trabalho foi desenvolvido. No entanto, o módulo de comunicação DNP3 do *software* Elipse Power Studio não apresenta nativamente suporte a criptografia nem ao protocolo TLS, assim como ferramentas semelhantes disponíveis, o que impossibilita uma conexão segura do sistema a um *outstation* externo. Portanto, a elaboração de um *gateway* para fazer a interface com os dados externos é algo essencial para proteger as informações presentes no sistema SCADA de invasões ou usuários mal intecionados.

A maioria dos ataques a sistemas *Smart Grids* SCADA ocorre pelo acesso dessas estruturas à Internet, disponibilizando assim dados cruciais em uma rede pública. No entanto, o simples fato de não permitir o acesso à rede mundial de computadores nessas estruturas talvez não seja a melhor opção, pois em uma parcela considerável de sistemas o uso da Internet é indispensável, como é o caso de plantas dispersas geograficamente, por exemplo. Entretanto, assegurar a troca de informações de plantas da estrutura é algo relevante. Pois, o simples fato de um usuário externo possuir acesso a informações cruciais da estrutura pode ocasionar prejuízos e problemas para o sistema. Essa menção refere-se a um ataque de confidenciabilidade, em que informações privadas são disponibilizadas a usuários externos a estrutura. Além disso, caso o usuário externo mal intencionado consiga efetuar um ataque como esse, dependendo das informações adquiridas é possí-

vel com que o atacante execute falsos comandos e monitoramentos no sistema, o que pode ocasionar problemas consideráveis. Para evitar ataques como esses, algumas opções relevantes baseiam-se na inclusão de autenticação e criptografia no sistema. Portanto, o presente trabalho busca minimizar esses ataques, adicionando essa segurança com base na aplicação do protocolo TLS a conexão DNP3 existente de estruturas SCADA de geração distribuída.

O sistema de segurança proposto, além de prover autenticação e criptografia, fornecerá uma diminuição na superfície de ataque. Isso será possível devido ao fato da arquitetura desenvolvida designar a concessionária de energia como o servidor da rede. O que viabiliza uma redução de possíveis ataques DDoS, por exemplo. Pois, nesse formato, esse tipo de ataque terá que ser realizado na rede da concessionária. O diferencial presente nessa metodologia se dá por meio de que a concessionária possui uma quantidade maior de recursos se comparados a rede do cliente, como equipes de TI monitorando a rede, sistemas comerciais de contra-medidas, entre outros. Outro fator relevante responsável por diminuir a superfície de ataque, refere-se também que com o uso do sistema proposto no trabalho, somente a conexão entre a porta mestre do *gateway* e o *outstation* (cliente), disponibilizará um IP público. Entretanto, justamente nesse setor será aplicado o uso de criptografia e autenticação de dados com o uso do protocolo TLS, fazendo com que os dados trocados só possam ser identificados por usuários autorizados e que possuam as chaves e certificados criptográficos correspondentes.

Devido a proteção que o protocolo TLS pode proporcionar a estruturas SCADA voltadas a *Smart Grids* baseadas no protocolo DNP3, foi definida a utilização dessa topologia para aplicação em uma estrutura SCADA de geração distribuída, visando aprimorar a segurança da transmissão de dados entre clientes e concessionárias de energia.

O trabalho proposto foi desenvolvido seguindo os padrões do protocolo DNP3, no modo hierárquico. Portanto, a estrutura que compõe o sistema é composta por mestre, sub-mestre e *outstation* DNP3.

3.1 SISTEMA DE SUPERVISÃO - MESTRE

O mestre, em um sistema SCADA é responsável por processar informações e efetuar o controle do processo. Na plataforma SCADA utilizada para implementar o mestre foi aplicado o uso do *software* Elipse Power Studio, conforme comentado anteriormente.

O sistema de supervisão é a parte de operação da estrutura, responsável por monitorar e supervisionar as variáveis, bem como efetuar comandos a dispositivos de controle associados a arquitetura. Portanto, para proporcionar uma estrutura SCADA completa é indispensável a associação de um sistema de supervisão à arquitetura concebida.

Para implementar a comunicação utilizando o protocolo DNP3 no Elipse, foi necessário o uso do *driver* DNP3.0 *Master*, disponibilizado pelo fabricante. Entretanto, esse *driver* de comunicação não possui suporte ao protocolo TLS. Sendo assim, foi necessário limitar a comunicação do mestre à rede interna da concessionária, de modo com que a segurança do sistema não fosse comprometida.

O funcionamento correto da aplicação desenvolvida para o trabalho fundamenta-se na comunicação, telas e *scripts*. As telas são a parte principal para o operador do sistema, pois são esses elementos que fornecem todas as informações necessárias para o monitoramento de dados e tomadas de decisão. No entanto, na maioria das vezes, para os comandos ou a exibição de informações serem executados de maneira pretendida é necessário implementar *scripts* ou comandos internos, implementados por meio do uso da ferramenta de desenvolvimento do fabricante.

O sistema de supervisão foi associado a um sistema de geração distribuída completo, contendo dados reais de medição de uma micro-rede. Sendo assim, a interface do sistema de supervisão foi implementada de modo a atender todos os parâmetros necessários para desempenhar as funções de controle e operação da estrutura. A Figura 20 apresenta a tela de informações presentes na micro-rede associada, uma das principais telas desenvolvidas no trabalho.

A aplicação desenvolvida foi destinada a monitorar informações de dados como: potência ativa de linha e das fases (A, B e C); potência reativa de linha e das fases (A, B e C); tensão de linha e das fases (A, B e C); referência de potência ativa e reativa; frequência nominal e da rede; fator de potência; *status* do modo de potência limitada; *status* do modo VAr constante; e *status* de geração. Além disso, foi necessário implementar os comandos: habilitar e desabilitar geração; valor limite para potência ativa; habilitar e desabilitar modo de potência limitada; valor para referência de potência ativa; habilitar e desabilitar modo VAr constante. Esses dados são recebidos no centro de controle da concessionária de energia. Além disso, é também nesse local em que é feito o monitoramento e a tomada de decisão sobre os comandos de operação do sistema.

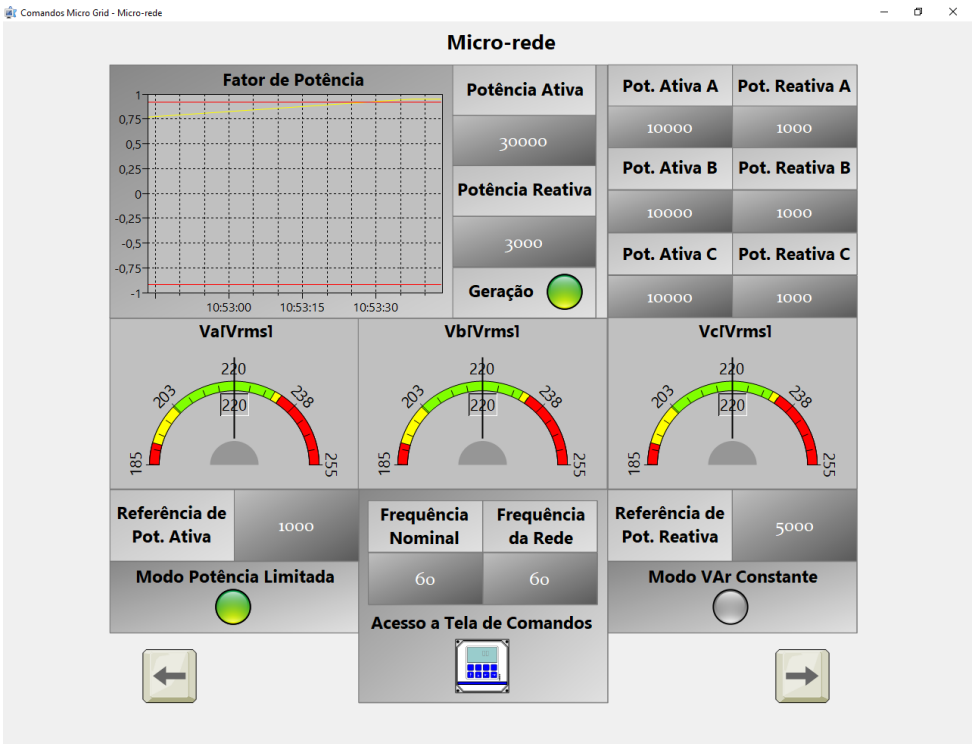

Figura 20: Tela de monitoramento de dados da micro-rede associada a estrutura.

Fonte: O Autor (2021).

Para reconhecimento das informações citadas, foi necessário implementar TAG's no *software* Elipse, compostos pelos dados DNP3 referentes a: entradas analógicas do grupo 30 e variação 01; entradas digitais de grupo 01 e variação 02; saídas analógicas baseadas no grupo 41 e variação 02; e os comandos CROB (*Control Relay Output Block*), utilizando o grupo 12 e variação 01. Além disso, os índices dos dados utilizados no sistema foram associados a informação relacionada a cada dado exclusivo, seguindo o perfil de pontos recomendado pelo padrão IEC 61850.

Para a completa implementação da aplicação desenvolvida, foi necessário adquirir uma licença de uso ilimitada do *software* Elipse Power Studio. Essa licença foi obtida por meio do projeto de pesquisa e desenvolvimento associado ao trabalho.

3.2 *GATEWAY* - SUB-MESTRE

Como sub-mestre do sistema foi implementado um *gateway*, utilizando como base o projeto `opendatacon` (STEPHENS, 2014). O *gateway* proposto neste trabalho irá realizar o encapsulamento dos dados inseguros do *driver* DNP3 implementado no *software* de supervisão em um pacote de comunicação DNP3/TLS. Essa comunicação com o sistema

de supervisão será realizada em uma conexão interna composta pelo protocolo DNP3, sem suporte ao TLS, realizado pela porta *outstation* do *gateway*.

O projeto `opendatacon` refere-se a um concentrador de dados capaz de efetuar a interface de dados externos a outro canal interno ou externo. Além disso, o `opendatacon` possui suporte aos sistemas operacionais Windows e Linux. O projeto `opendatacon` remete-se a uma ferramenta flexível e altamente configurável para proporcionar uma alta diversidade de soluções.

O *gateway* é uma máquina que possui duas interfaces de rede. No sistema proposto, uma dessas interfaces é direcionada à rede pública e a outra em uma rede interna da concessionária. Executar o *gateway* em um ambiente Windows é necessário, pois é fundamental que o *gateway* seja implementado no mesmo dispositivo em que o sistema de supervisão será executado. Essa exigência deve-se ao fato do *software* Elipse Power Studio somente possuir suporte para o sistema operacional Microsoft Windows. Logo, a conexão DNP3 sem suporte ao protocolo TLS com o supervisório será limitada a uma conexão interna (conexão na interface de *loopback*), o que proporciona uma redução na fragilidade do sistema.

O requisito mínimo de um concentrador de dados é associar diversos canais de transmissão de dados a um único destino. O `opendatacon` atende a esse requisito, inclusive é capaz de fazer transformações nos dados transmitidos entre os canais. No entanto, na aplicação desenvolvida, isso não será necessário.

O `opendatacon` utiliza a biblioteca `OpenDNP3` como uma das opções de entrada e saída de dados. Entretanto, a ferramenta possui suporte a vários outros formatos, como o JSON ou mesmo o Modbus, por exemplo.

O sistema de supervisão é sempre mestre do sistema e se conecta a um *outstation* DNP3 no *gateway*. Dessa forma, os gerenciadores de energia ou sistemas de geração distribuída associados a estrutura, geralmente são *outstations* e, devido a isso, o *gateway* deve implementar um mestre para essas *outstations*. Sendo assim, para utilizar os protocolos DNP3 e TLS na porta mestre do *gateway* na API (*Application Programming Interface*) implementada, foi realizado o uso das bibliotecas `OpenDNP3` e `OpenSSL`. Além disso, foi necessário a implementação de suporte ao TLS no *gateway*, pois o projeto `opendatacon` não possue em sua composição suporte a esse protocolo de segurança.

Para a porta *oustation* do *gateway*, também foi necessário o uso da biblioteca `OpenDNP3`. Além disso, foi necessário configurar a porta *outstation* para atender o mestre do sistema (Elipse).

A biblioteca `OpenDNP3` foi selecionada por oferecer suporte a comunicações RS-232/485, TCP/IP e TLS, além de tratar-se de um código aberto, licenciado sobre Apache 2.0. Logo, a biblioteca pode ser utilizada tanto na conexão local sem suporte a TLS, quanto na comunicação externa utilizando o protocolo TLS sobre o DNP3.

O projeto `opendatacon` é um programa completo, não refere-se a uma biblioteca. Portanto, a definição de conexões e transformações de dados são determinadas em arquivos de configuração. Com base nessas configurações, o `opendatacon` decide quais portas serão elaboradas, e faz chamadas as bibliotecas que implementam os protocolos referentes as portas cedidas.

Os componentes básicos para a configuração do `opendatacon` são demonstrados nas Figuras 21, 22, 23 e 24. No formato demonstrado na Figura 21, as portas 1 e 2 são ligadas entre si por uma única conexão. No entanto, na Figura 22, as portas 1 e 2 são conectadas uma pela outra, bem como as portas 1 e 3. Portanto, nessa configuração cada um possui uma conexão própria.

Figura 21: Configuração para duas portas em uma única conexão.

Fonte: (STEPHENS, 2014).

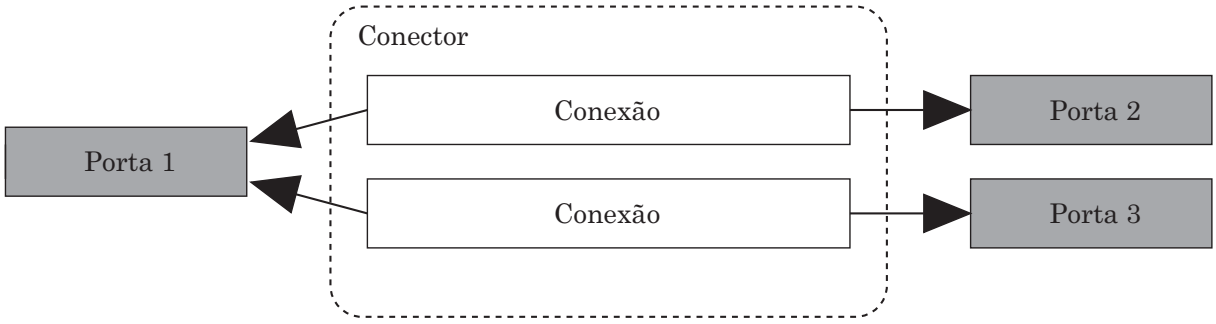

Figura 22: Configuração de conexão exclusiva com a porta de destino para três ou mais portas.

Fonte: (STEPHENS, 2014).

Na configuração presente na Figura 23, as portas 1 e 2 também são ligadas entre si. Porém, nesse formato há a presença de uma transformação unidirecional, que afeta as informações enviadas da porta 1 para a porta 2.

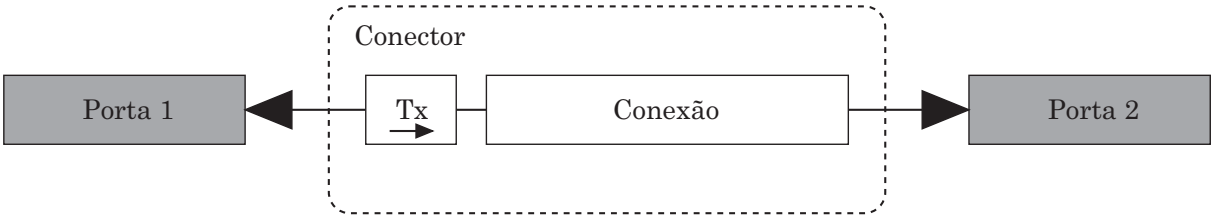

Figura 23: Configuração para duas portas em uma única conexão com transformação unidirecional da porta 1 para a porta 2.

Fonte: (STEPHENS, 2014).

Para uma configuração com mais portas, possui a opção apresentada na Figura 24, em que as portas 1 e 2 são conectadas, tal como as portas 1 e 3. No entanto, nesse formato a opção de transformação unidirecional dos dados enviados da porta 1 a porta 2 e 3. Além disso, há outra transformação direcional que influencia a transmissão dos dados enviados da porta 2 para porta 1, para adequar a necessidade do formato pretendido.

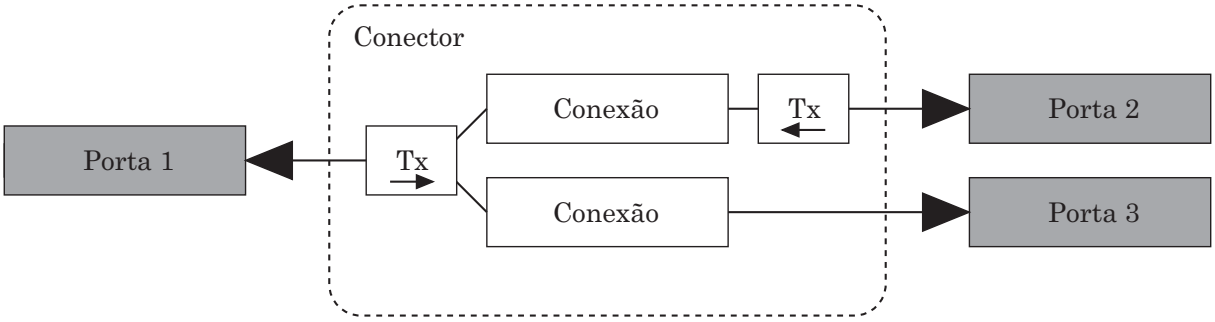

Figura 24: Configuração de conexão exclusiva com a porta de destino para três ou mais portas com transformações unidirecionais para portas selecionadas.

Fonte: (STEPHENS, 2014).

As portas são responsáveis por realizar a interface entre o projeto `opendatacon` e canais externos. É de responsabilidade do arquivo `Port` fazer as conversões recebidas ou enviadas pela estrutura de dados interna a um protocolo externo. O projeto `opendatacon` possui suporte a porta cliente JSON (*JavaScript Object Notation*), portas mestre e *outstation* DNP3 e Modbus, porta de simulação e nula.

Os conectores são responsáveis por delinear logicamente a junção de conexões e transformações. Além disso, os conectores são utilizados para proporcionar uma flexibilidade e permutação superiores à medida da necessidade do projeto. Por exemplo, um perfil semelhante de conexões/transformações pode ser duplicado de modo simples ape-

nas com o uso de uma configuração de conector comum e realizando assim, a substituição somente dos principais atributos.

As conexões no projeto `opendatacon` são responsáveis basicamente por abrir um caminho de dados bidirecional entre duas portas. Além disso, faz com que as portas sejam associadas entre si e também o roteamento de dados entre elas.

As transformações, apresentadas nas Figuras 23 e 24, fazem a alteração do filtro e dos dados diretamente na entrada do conector de modo unidirecional com destino para a porta de envio dos dados. As transformações podem ser implementadas para adicionar registros de data e hora a dados que não possuem o carimbo de data/hora ou banda morta, ou até mesmo restringir um fluxo de dados.

Os componentes básicos do projeto `opendatacon` descritos acima constituem a base da API de leitura. As portas e transformações podem ser elaboradas de modo a desenvolver a interface e ou quebrar protocolos arbitrários, bem como fazer a manipulação arbitrária de dados.

Os componentes incluídos no projeto `opendatacon` utilizam o formato de dados JSON para efetuar sua configuração. O JSON foi selecionado para configuração porque refere-se a um formato leve de troca de dados, bem como é simples para compreensão e desenvolvimento.

De modo geral, a configuração de um conector, porta, transformação ou conexão está incluso em um objeto JSON. Um objeto JSON é um conjunto de pares de valores-chave, em que os valores podem ser objetos aninhados ou matrizes. Desse modo, esses objetos são estabelecidos em um objeto JSON geral. Sendo assim, para efetuar uma configuração da topologia DNP3 pretendida, é preciso estabelecer o formato no arquivo principal de configuração do projeto `opendatacon`.

O papel do *gateway* no trabalho é basicamente ampliar a segurança do sistema de supervisão. Sendo assim, com o uso dessa ferramenta, a comunicação sem suporte a criptografia e autenticação será limitada exclusivamente a uma conexão interna.

A elaboração do trabalho, baseia-se na topologia hierárquica DNP3 utilizando uma *oustation* para o sistema de geração distribuída que comunica DNP3 seguro (composto pelo protocolo TLS) com o sistema operacional Windows (sub-mestre), que a partir do *gateway* desenvolvido irá efetuar a transmissão de informações para o sistema SCADA (mestre) sem suporte a segurança.

Como comentado anteriormente, o projeto `opendatacon` possui suporte a biblioteca `OpenDNP3` e por isso, a estrutura de dados interna baseia-se no formato dessa biblioteca. Dessa forma, os conceitos baseiam-se no encapsulamento de dados por eventos. Esses eventos possuem três partes: a carga útil, responsável por transmitir o formato de dado (CROB, analógico ou binário), um *timestamp* e um índice. Com isso, as portas são capazes de enviar e receber eventos, e utiliza-se o índice e o tipo como base para mapear os dados enviados ou recebidos.

Foi utilizado na implementação da API, o componente `DNP3PORT`, de modo a configurar as portas ao protocolo DNP3. Além disso, o formato definido para associação, inclui uma porta mestre DNP3 para efetuar a conexão local e uma porta *outstation* DNP3 para interface com dados externos.

Inicialmente o projeto não possuía uma especificação de portas DNP3 composta por TLS. Essa configuração foi adicionada e consequentemente uma porta composta por DNP3 com TLS elaborada, de modo a efetuar a comunicação com dados externos seguros.

No projeto `opendatacon`, as conexões são definidas em arquivos de configuração. As configurações utilizadas no sistema proposto são apresentadas no Apêndice A. Além disso, foram adicionadas opções para suporte a verificação do pacote de chaves criptográficas utilizadas na comunicação TLS ao arquivo `DNP3PortConf.h`, disponibilizado pelo repositório do projeto. Portanto, nesse arquivo foram inclusas as variáveis para suporte ao TLS, bem como os certificados de autorização, a cadeia de certificados e a chave privada, definidos para uso no trabalho.

Durante a implementação da API, verificou-se que o projeto `opendatacon` não possuía suporte a saídas analógicas quando era aplicado o uso do protocolo DNP3. Como esse tipo de dado é amplamente utilizado em sistemas *Smart Grids* SCADA, foi necessário adicionar o suporte a saídas analógicas ao projeto. Desse modo, para suporte aos índices de saídas analógicas, foi adicionado um código, que efetua a requisição dos pontos a partir do índice pretendido, uma faixa para diversas requisições, ou ainda valores definidos para início e fim.

A parte fundamental do *gateway* desenvolvido é que a porta DNP3 não segura somente escuta conexões locais. Logo, desde que o ambiente onde a API esteja associada seja seguro, a conexão permanece segura.

3.3 *OUTSTATION* - CLIENTE

O *oustation* é relacionado a coleta de dados e execução dos comandos repassados pelo mestre da operação. O *oustation* do sistema proposto é basicamente uma estrutura de geração distribuída, conforme apresentado na Figura 25. Essa estrutura utiliza-se de sistemas embarcados, baseados em linux, que compõem um gerenciador de energia conectado a vários inversores.

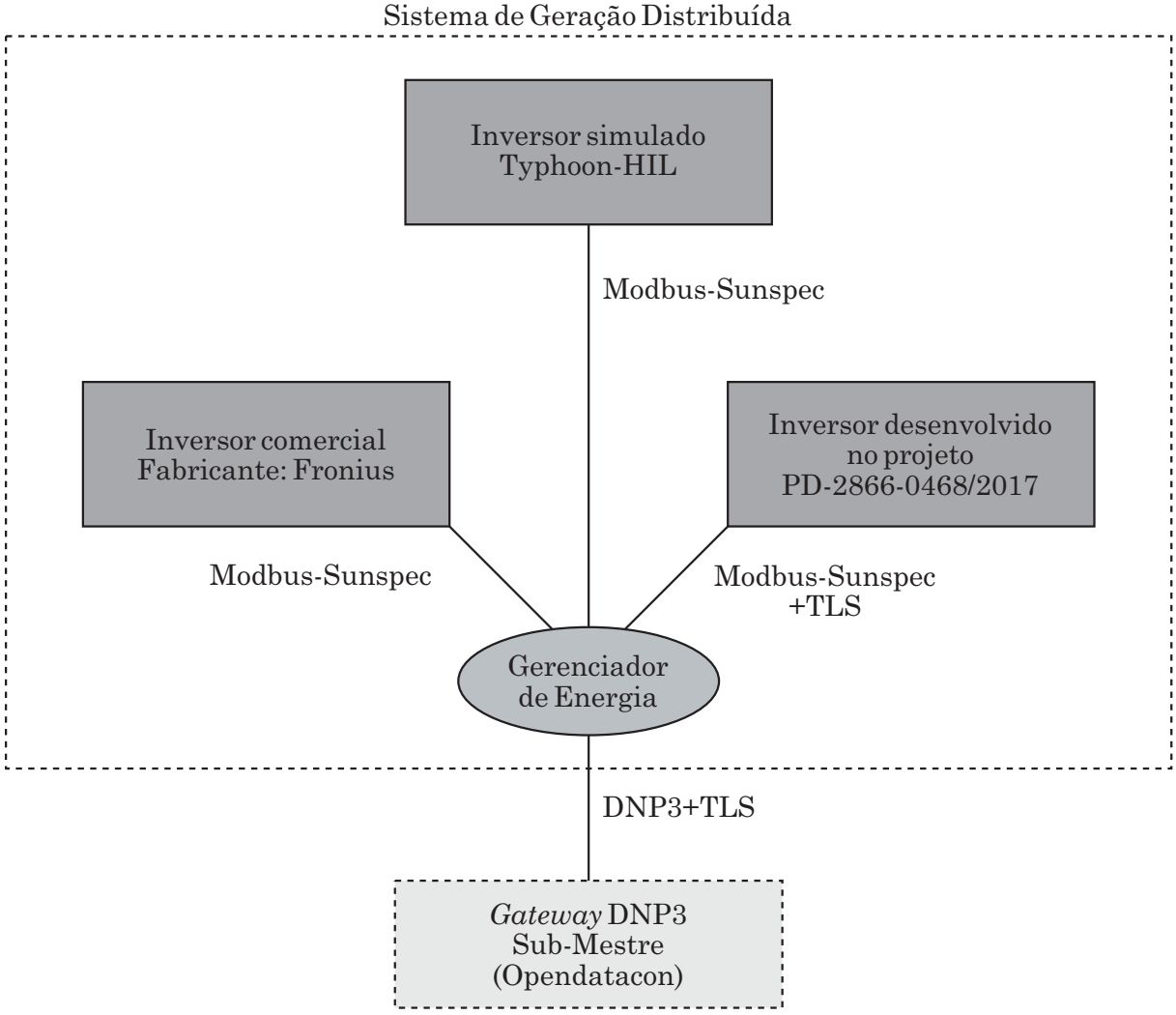

Figura 25: Configuração do cliente da estrutura.

Fonte: O Autor (2021).

O gerenciador de energia é responsável por executar comandos gerados pelo sistema de supervisão e enviar dados de medição do sistema de geração distribuída para a concessionária de energia. Além disso, o gerenciador agrega os dados dos inversores conectados ao sistema e os apresenta em um modelo de dados definidos pelo perfil do padrão IEC 61850.

Geralmente, as *outstations* são posicionadas em um local externo ao centro de controle do sistema, mais precisamente em clientes da concessionária de energia. Portanto, o sistema foi desenvolvido voltado a estruturas semelhantes. Logo, a conexão DNP3 segura, composta pelo protocolo TLS nesse local é extremamente relevante, para proteger a estrutura de eventuais ataques cibernéticos.

Para o *oustation* do sistema foram adaptados os perfis recomendados da IEC 61850 aos formatos de dados do DNP3, com base na biblioteca `OpenDNP3`. Além disso, foi acrescentado suporte aos formatos de dados DNP3, como entradas analógicas, entradas binárias, saídas analógicas e comandos CROB.

Os dados de entradas analógicas e binárias são obtidos diretamente do local em que o *outstation* está posicionado. No entanto, os comandos como CROB e saídas analógicas, são repassados pelo *gateway* da estrutura, que recebe a informação designada pelo mestre do sistema. Desse modo, como os dados são transmitidos e recebidos por usuários externos é indispensável que essa comunicação seja segura.

Para que a conexão DNP3 pudesse ter suporte ao TLS, também foi necessário incluir a configuração do TLS no *outstation* DNP3 desenvolvido baseado na biblioteca `OpenDNP3`. O uso do protocolo TLS será realizado por meio da aplicação da biblioteca `OpenSSL`. Logo, o *outstation* implementado, atua passando os certificados e chaves para a chamada da `OpenSSL`, que abre um *socket* TCP e efetua o *handshake* da conexão.

Para atender os requisitos da comunicação DNP3 sob TLS, foi implementado na aplicação desenvolvida, inicialmente, o suporte a verificação de chaves criptográficas utilizadas na comunicação TLS. Sendo assim, foram inclusas também as variáveis para suporte ao protocolo seguro, certificados de autorização, cadeia de certificados e a chave privada. É interessante ressaltar que os arquivos precisam ser compatíveis com os implementados na comunicação segura do *gateway*, para efetuar a comunicação corretamente.

3.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Nesse capítulo foi apresentada a solução proposta, que compreende a implementação das etapas necessárias para o desenvolvimento da comunicação e operação do sistema de geração distribuída. Inicialmente, foi apresentado o mestre da conexão, o sistema de supervisão. Sendo assim, comentou-se brevemente sobre o desenvolvimento desta etapa, apresentando todas as ferramentas utilizadas, bem como as alterações e implementações efetuadas para atender a necessidade do sistema. Em seguida, foi apresentado o

desenvolvimento da etapa que compreende o *gateway*, implementado para ser o sub-mestre do sistema. Nessa etapa foram apresentadas também as ferramentas utilizadas, bem como as implementações necessárias para que o *gateway* seja capaz de fazer a interface de dados externos seguros a comunicação interna sem suporte ao protocolo TLS. Posteriormente, foi discutido sobre a etapa de desenvolvimento do cliente da conexão, setor responsável por fazer a coleta dos dados do sistema e executar ações pré-determinadas pela concessionária de energia. Além disso, nessa seção também foi apresentada a implementação do suporte a comunicação segura.

O sistema desenvolvido encontra-se em operação atualmente e livre de erros. No entanto, para efetuar a validação do sistema foi necessário efetuar testes de segurança na estrutura desenvolvida. Dessa forma, foi realizado a análise da solução proposta que será apresentada no próximo Capítulo.

4 RESULTADOS E DISCUSSÕES

No capítulo anterior foram apresentadas as etapas relacionadas ao desenvolvimento da estrutura SCADA voltada a sistemas de geração de energia distribuída, bem como foi apresentada uma proposta para ampliar a segurança nessas estruturas. Além disso, foram apresentados os tópicos necessários e o desenvolvimento das etapas para obter esse sistema de segurança capaz de prover proteção a estruturas *Smart Grids* SCADA. Sendo assim, após o desenvolvimento de todas as etapas que correspondem a arquitetura SCADA e o sistema de segurança, as estruturas obtidas foram associada ao sistema de geração distribuída elaborado durante o projeto de P&D 2866-0468/2017.

No setor destinado a concessionária de energia, foram realizadas duas conexões, uma interna e outra externa, como apresentado na Figura 26. A conexão interna é responsável por interfacear os dados com o mestre (Elipse), setor sem suporte ao protocolo TLS. Essa conexão ficou restrita a uma conexão local, direta com a porta *outstation* presente no *gateway*. O interfaceamento de dados externos é destinado a porta mestre do *gateway* desenvolvido. Sendo assim, esse canal é o único ponto que permite acesso a rede pública. Porém, a conexão nesse ponto é realizada com autenticação e criptografia, o que busca diminuir a incidência de ataques e devido a característica do sistema desenvolvido, também reduzir consideravelmente a superfície de ataque. Para demonstrar o funcionamento do sistema e efetuar a análise dos resultados é fundamental executar ataques a estrutura desenvolvida, pois tentar invadir a estrutura talvez seja a melhor forma de demonstrar sua eficácia.

Ataques a estrutura SCADA são realizados por diversos métodos distintos, como o MITM, *Denial of Service*, *replay*, FDI, entre outros. É indicado que ataques de negação de serviço sejam investigados pela equipe de TI da concessionária de energia. Além disso, com a diminuição da superfície de ataque proporcionada pelo sistema proposto é possível reduzir o tempo aplicado nessa busca. Entretanto, devido a quantidade considerável de incidentes envolvendo o formato MITM à sistemas SCADA, esse método foi escolhido para realizar os testes de ataque na estrutura proposta. É importante destacar que o sistema desenvolvido também possui resultados satisfatórios na proteção de ataques FDI e *replay*, visto que o ataque MITM envia uma injeção de dados falsos quando

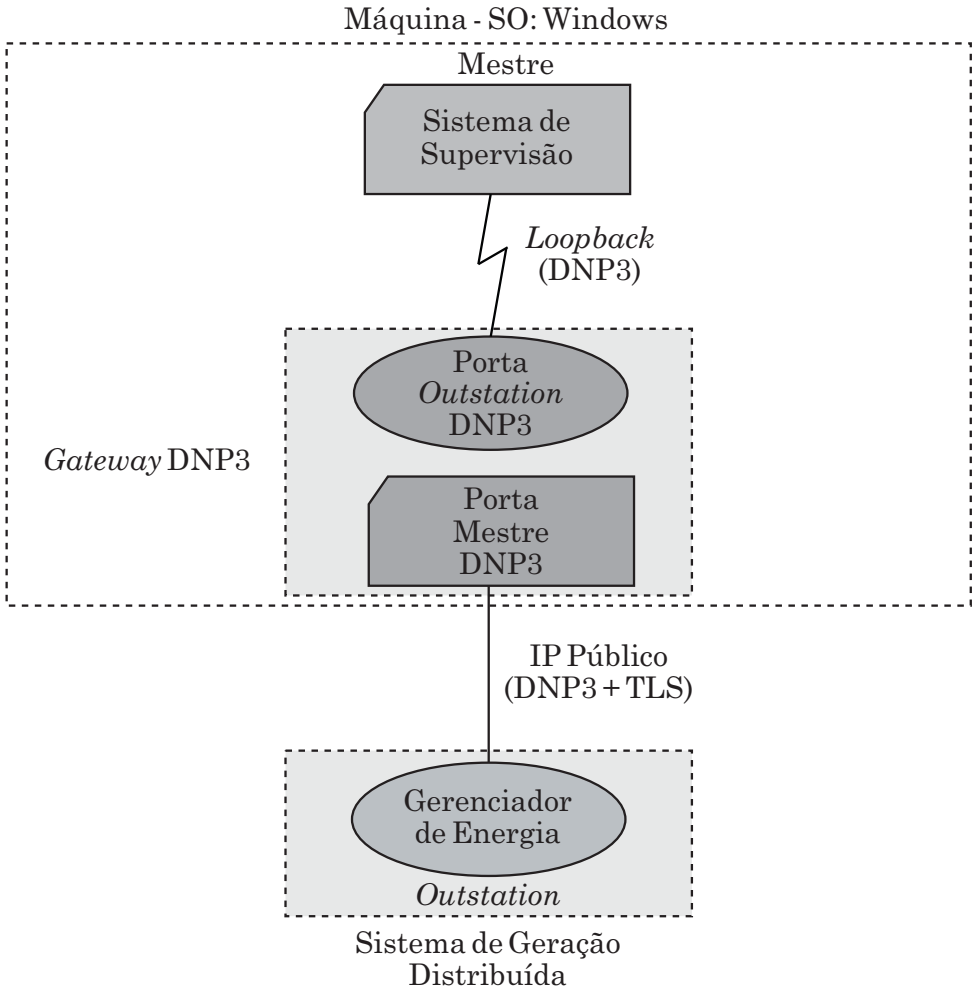

Figura 26: Conexões do gateway na estrutura proposta.

Fonte: O Autor (2021).

um dado diferente do real é enviado. Além disso, quando diversas requisições da mesma informação é enviada durante um ataque MITM é desenvolvido um ataque de repetição.

Para efetuar os ataques com o intuito de analisar e reconhecer o funcionamento do sistema desenvolvido foi recorrido ao auxílio de um projeto, o Ettercap, que baseia-se em uma ferramenta de ataque MITM para testes de sistemas de segurança. A ferramenta foi selecionada por se tratar de um projeto de código aberto e livre para desenvolvedores no setor de segurança de sistemas.

4.1 FERRAMENTA DE ANÁLISE DE DADOS

A fim de obter a análise do sistema de segurança desenvolvido foi empregue o uso de uma ferramenta com base em um projeto de código aberto, voltado a testes de segurança. Esse projeto é o denominado Ettercap. O Ettercap (ORNAGHI; VALLERI,

2014), têm como base o desenvolvimento de uma ferramenta para utilização em testes de segurança de sistemas. Essa ferramenta permite ao usuário interceptar o tráfego em um segmento de rede e escutar comunicações de diversos protocolos. Além disso, caso o protocolo em questão não seja nativo da ferramenta é possível incluir *plug-ins* que podem ser desenvolvidos com base na necessidade de cada projeto.

A ferramenta proporciona quatro modos de operação: filtragem de pacotes de IP's determinados, filtragem com base em endereços MAC e envenenamento ARP. Além disso, apresenta uma interface para monitoramento de conexões e filtragem de conteúdo, além de fornecer suporte a recursos para análises de rede e *host*. É uma ferramenta multi-plataforma, possuí suporte para sistemas operacionais como o Debian/Ubuntu, Fedora, Gentoo, Pentoo, Mac OSX, Freebsd, entre outros.

Essa ferramenta proporciona efetuar um ataque MITM a usuários selecionados presentes na mesma rede. Porém, a ferramenta só proporciona no modo *Man-In-The-Middle* a interceptação dos dados presentes na rede. Sendo assim, a alteração e transmissão de dados falsos são efetuados por meio de *plug-ins* desenvolvidos em linguagem `C` acrescentados a estrutura.

4.2 DESENVOLVIMENTO DO ATAQUE PROPOSTO

A estrutura de segurança desenvolvida busca proteger um sistema de geração distribuída não-seguro com a inclusão de autenticação e criptografia. Dessa forma, a análise do sistema de segurança será fundamentada em dois ataques distintos, um capaz de efetuar a leitura de um dado DNP3 na rede e outro capaz de efetuar alterações no dado DNP3 transmitido.

O primeiro ataque, baseia-se na realização do ataque MITM, somente para leitura de dados DNP3 presentes na rede. Esse ataque afeta diretamente a confidencialidade. A perda de confidencialidade ocorre quando informações relevantes sobre a configuração do dispositivo ou a topologia de rede são obtidas por um invasor. Geralmente, esse é o primeiro passo para um ataque mais sério, onde o reconhecimento é realizado para identificar pontos fracos no sistema e condições para facilitar a entrada.

O segundo formato é destinado a afetar a integridade. Os ataques de integridade são responsáveis por inserir dados incorretos ou reconfigurar os dados pré-estabelecidos nos sistemas. A perda de integridade ocorre quando o centro de controle não possui informações precisas sobre o estado da estrutura. Por exemplo, se o invasor con-

seguiu acesso a rede, ele pode enviar um comando para desarmar um disjuntor e impedir que um alarme referente a mudança de estado chegue ao operador. Tais ataques podem ocasionar incidentes graves, porque seus efeitos podem perdurar um tempo considerável. Pois, com o passar do tempo sem monitoramento ou controle de um determinado dado é possível com que esse valor modificado chegue a afetar o sistema drasticamente. Outros ataques perigosos são os que resultam na perda de controle da estrutura. Um invasor que é capaz de conseguir o controle total de um mestre de um sistema SCADA pode causar danos consideráveis, até mesmo inviabilizar completamente o controle administrativo da estrutura.

Para viabilizar o ataque de confidenciabilidade, foi necessário desenvolver um *plug-in* capaz de reconhecer e estruturar o dado DNP3, que será lido e alterado na estrutura. O *plug-in* desenvolvido encontra-se no repositório disponível em (FIGUEIREDO, 2021). Esse *plug-in* foi desenvolvido com base no código aberto disponibilizado em (PESTANA, 2019), que fundamenta-se em testes de modificação de pacotes DNP3 e IEC 60870-5-104 para fins de pesquisa. Sendo assim, para atender a demanda necessária foi alterada a estrutura do dado para atender uma saída analógica de 32 *bits*. O *plug-in* atua escutando a rede e analisando a presença de pacotes DNP3, quando há algum dado semelhante ao pré-estabelecido, o mesmo faz a análise completa do pacote e repassa as informações para o atacante. Sendo assim, foi definida a leitura do valor de saída de um dado *Analog Output* de 32 *bits*, de objeto 41 e variação 01, de código de função `Direct Operate` e índice 33. Esse dado foi utilizado somente para testes do ataque, não é um dado utilizado no sistema proposto. Porém, seguindo os perfis do padrão IEC 61850, esse dado poderia ser um valor definido para limite de potência ativa, ou um valor de referência para potência reativa, utilizados na aplicação SCADA desenvolvida no trabalho, por exemplo.

A análise de somente um tipo de dado pelo *plug-in* foi definido devido ao fato do protocolo DNP3 possuir vários pacotes de dados distintos, ao todo são 81 pacotes. Dessa forma, para modificar a estrutura do dado a ser analisado pelo atacante é necessário efetuar algumas alterações, como a dimensão do *bit* do dado em questão e o código de função, por exemplo. Recomenda-se o uso de uma ferramenta de análise de pacotes, como o *Wireshark* por exemplo, para reconhecer e estabelecer a estrutura desejada. No entanto, para efetuar a modificação dos dados DNP3 trafegados na rede requer algumas configurações extras.

Para implementar o ataque de integridade, inicialmente foi necessário adicionar suporte a estrutura de um dado DNP3 analógico de 32 *bits* no *plug-in*. Em seguida,

definir o valor desejado para repasse ao *outstation* do sistema, nesse caso, 66 e copiar esse valor para o campo de valor do dado DNP3 estabelecido na estrutura. Posteriormente, também foi necessário efetuar a mudança do valor de saída para *little-endian*. Além disso, um algoritmo `CRC` (*Cyclic Redundancy Check*), foi implementado. Pois, com a alteração de *bytes* do dado, é gerado um novo pacote, distinto do recebido na conexão e consequentemente o CRC correspondente ao pacote é alterado. Dessa forma, é necessário desenvolver um algoritmo que seja capaz de calcular o novo CRC para o novo pacote. Esse algoritmo foi elaborado com base no `CRC` utilizado na biblioteca `OpenDNP3`. O *plug-in* para efetuar o ataque de integridade também encontra-se disponível em (FIGUEIREDO, 2021).

Para o reconhecimento e análise dos dados DNP3 na rede foi utilizada a ferramenta de monitoramento *Wireshark*. Essa ferramenta permite analisar todas as transmissões de dados presentes na rede do usuário monitorado. Dessa forma, foi aplicado o uso da ferramenta em todos os usuários, o *outstation*, *sub-mestre* e atacante. Além disso, foi utilizado também como agente de análise as próprias API's desenvolvidas para a estrutura. Essas API's possuem uma interface simplificada, em que é possível analisar a troca e recebimento de dados DNP3 transmitidos entre as extremidades. Além disso, no próprio *plug-in* desenvolvido para a ferramenta `Ettercap` foi elaborada uma interface simplificada capaz de apresentar o dado DNP3 real obtido na comunicação, bem como o alterado pelo *plug-in*.

4.3 ANÁLISE DOS ATAQUES APLICADOS A ESTRUTURA DESENVOLVIDA

O sistema atacante foi desenvolvido para ler e efetuar a alteração de dados presentes em um sistema de geração distribuída. No entanto, para efetuar o ataque é necessário ter acesso a rede de comunicação pretendida, que pode ser uma rede conhecida pelo invasor ou o mesmo obtendo o reconhecimento por outras formas, como e-mails de *spear-phishing*, como no caso BlackEnergy3 e Dragonfly/HAVEX, acesso a sites duplicados como também aconteceu no caso Dragonfly/HAVEX, por dispositivos USB contendo um *malware*, como no caso Stuxnet, entre outras formas possíveis. Sendo assim, o atacante desenvolvido foi implementado no setor que possui acesso à rede pública do sistema, uma rede que nativamente seria vulnerável a ataques por disponibilizar IP's públicos na Internet. Nessa rede ocorre a troca de informações entre a porta mestre do *gateway* desenvolvido e a *outstation* do sistema. Inicialmente, para validar o ataque desenvolvido, foi simulada essa conexão DNP3 sem suporte ao TLS, que teoricamente permite ao atacante efetuar ataques de confidenciabilidade e integridade aos dados presentes na estrutura.

Em seguida, foram efetuados os ataques durante o uso da comunicação composta pelos protocolos DNP3 e TLS, para análise e validação da efetividade da segurança do sistema após a inclusão de autenticação e criptografia.

4.3.1 Ataque de confidenciabilidade

Inicialmente, foi realizada uma conexão composta pelo protocolo DNP3 da porta mestre do *gateway* de endereço IP `10.0.2.1` e MAC `52:54:00:12:35:01` ao *outstation* do sistema, de IP `10.0.2.2` e MAC `92:3f:44:6a:02:88`. Entretanto, nessa mesma rede foi conectado o usuário malicioso, de endereço MAC `08:00:27:6d:ef:ec` e IP `10.0.2.3`.

No usuário malicioso, com o uso da ferramenta `Ettercap` foi selecionado o *host* com endereço de IP `10.0.2.1` e endereço MAC `52:54:00:12:35:01`, bem como o *host* de IP `10.0.2.2` e MAC `92:3f:44:6a:02:88` como alvos do ataque. Em seguida, foi realizado o ataque MITM de envenenamento ARP, disponibilizado pela própria ferramenta. Esse ataque modifica a troca de informações entre *gateway* e *outstation*, que passa a ser realizada pelo intermédio do atacante, conforme apresentado na Figura 27. Dessa forma, isso possibilita ao atacante a alteração dos MAC's conhecidos pelos outros usuários, de modo que para o *outstation* todos os dados provenientes do endereço `10.0.2.1` agora está direcionado ao MAC do usuário malicioso (`08:00:27:6d:ef:ec`), conforme demonstrado na Figura 28. Isso também ocorre na porta mestre do *gateway* que reconhece o usuário malicioso de MAC `08:00:27:6d:ef:ec` como o usuário correspondente ao IP `10.0.2.2`, conforme apresentado na Figura 29.

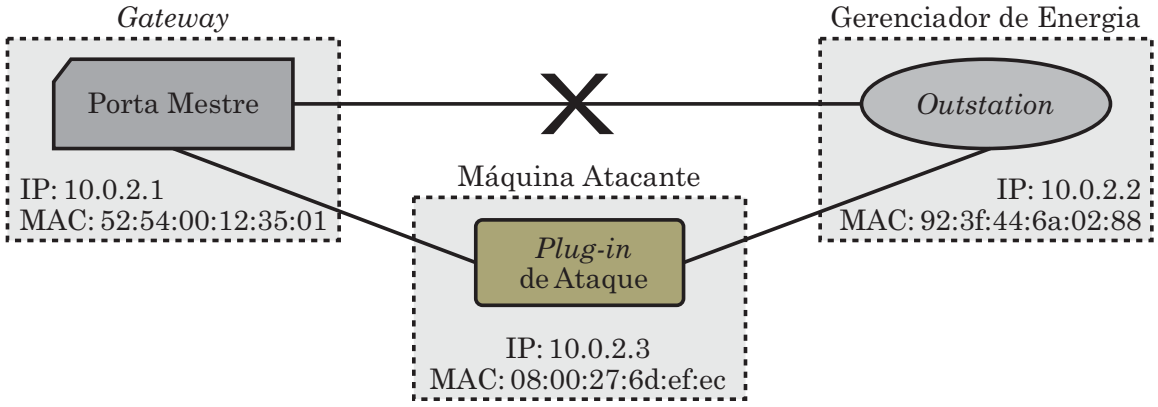

Figura 27: Máquina atacante interferindo na conexão do gateway com o outstation.

Fonte: O Autor (2021).

```
bob@bob:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 92:3f:44:6a:02:88 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.2/24 brd 10.0.2.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::903f:44ff:fe6a:288/64 scope link
       valid_lft forever preferred_lft forever
bob@bob:~$ ip nei
10.0.2.3 dev enp0s3 lladdr 08:00:27:6d:ef:ec STALE
10.0.2.1 dev enp0s3 lladdr 08:00:27:6d:ef:ec REACHABLE
```

Figura 28: IP's e MAC's conhecidos pelo outstation do sistema.

Fonte: O Autor (2021).

```
alice@alice:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 52:54:00:12:35:01 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.1/24 brd 10.0.2.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fe12:3501/64 scope link
       valid_lft forever preferred_lft forever
alice@alice:~$ ip nei
10.0.2.2 dev enp0s3 lladdr 08:00:27:6d:ef:ec REACHABLE
10.0.2.3 dev enp0s3 lladdr 08:00:27:6d:ef:ec STALE
```

Figura 29: IP's e MAC's conhecidos pelo gateway.

Fonte: O Autor (2021).

Para efetuar a leitura de dados, é necessário a transmissão de algum dado na rede. Dessa forma, foi enviado um pacote, contendo as informações: `Direct Operate`, objeto 12, variação 01 e índice 15 e valor 01. Porém, como o ataque de confidenciabilidade foi efetivado, o dado não vai direto ao destinatário (*outstation*). Sendo assim, conforme apresentado na Figura 30, que demonstra análise da rede feita no sistema que compõe o usuário malicioso, o dado foi enviado para o IP `10.0.2.2` corretamente, como desejado. Porém, após a execução do ataque, conforme demonstrado anteriormente na Figura 29, o sistema que compreende o *outstation* da conexão reconhece que o IP `10.0.2.2` é destinado ao MAC `08:00:27:6d:ef:ec`. Portanto, o dado é enviado primeiro ao intermediário, o usuário malicioso, que obtém as informações confidenciais da rede em questão.

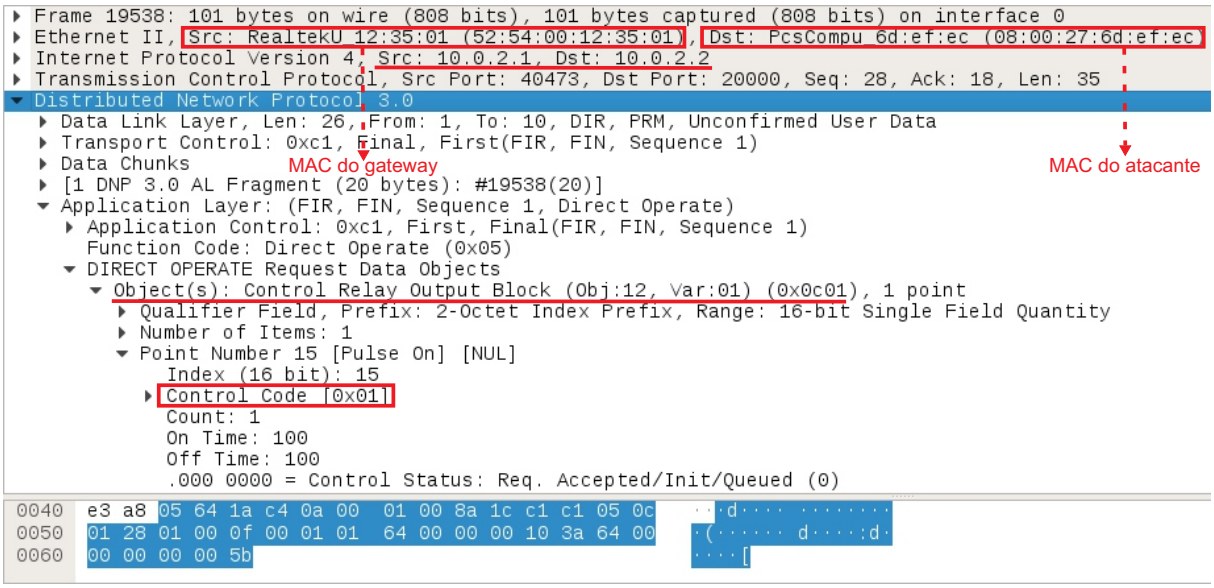

Figura 30: Monitoramento do pacote enviado pela porta mestre do gateway.

Fonte: O Autor (2021).

Como o ataque busca somente acesso a informações privadas, o dado é enviado ao destinatário de forma correta, o que pode ser analisado por meio do monitoramento de rede feito no *outstation* do sistema. Sendo assim, é possível verificar que o remetente da informação refere-se ao IP `10.0.2.1` normalmente. Porém, o MAC da fonte remete-se ao usuário malicioso, como pode ser analisado na Figura 31.

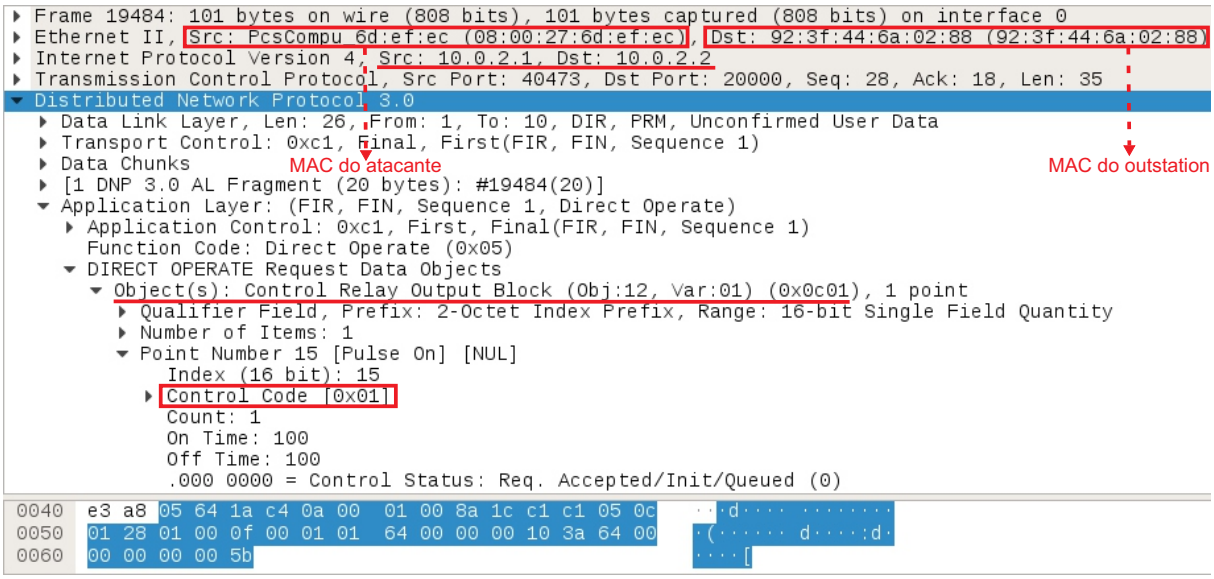

Figura 31: Monitoramento do pacote recebido pelo outstation da conexão.

Fonte: O Autor (2021).

O ataque de confidenciabilidade visa buscar informações privadas e exclusivas de estruturas críticas. O que pode fornecer informações sigilosas sobre um formato de

sistema único e exclusivo, por exemplo. Sendo assim, essas informações podem até mesmo proporcionar a usuários indevidos a possibilidade de efetuar cópias desses sistemas. No entanto, os principais focos da execução de ataques de confidenciabilidade são identificar pontos fracos e/ou uma porta de entrada para a execução de ataques críticos, como um ataque de integridade, por exemplo.

4.3.2 Ataque de integridade

Com a realização do ataque de confidenciabilidade, foi possível analisar a estrutura do dado transmitido e modificar o *plug-in*, de modo que seja capaz de alterar o dado transmitido, efetuando assim um ataque de integridade. Desse modo, a etapa seguinte foi ativar o *plug-in* desenvolvido para assim que haja um comando `Direct Operate`, dessa vez com objeto 41, variação 01 e índice 33, ocorra a mudança do valor da saída sempre para 66. Sendo assim, foi enviado um pacote pelo mestre da estrutura ao *outstation* de objeto 41, variação 01, índice 33 com a função `Direct Operate` e valor de saída de 99, para testar a eficácia da ferramenta de ataque. Esse pacote pode ser analisado na Figura 32, que demonstra o pacote enviado pela interface da API baseada na biblioteca `OpenDNP3`.

```
ms(1604693648575) WARN    master - Timeout waiting for response
Summary: FAILURE_RESPONSE_TIMEOUT
Header: 0 Index: 33 State: INIT Status: UNDEFINEDms(1604693648575) INFO    master - Begining task: Command Task
ms(1604693648575) --AL->  master - C3 05 29 01 28 01 00 21 00 63 00 00 00 00
ms(1604693648575) --AL->  master - FIR: 1 FIN: 1 CON: 0 UNS: 0 SEQ: 3 FUNC: DIRECT_OPERATE
ms(1604693648575) --AL->  master - 041,001 Analog Output - 32-bit With Flag, 16-bit count and prefix [1]
```

Figura 32: Interface da API demonstrando o pacote DNP3 enviado.

Fonte: O Autor (2021).

Com o tráfego de dados DNP3 na rede e o *plug-in* habilitado, a ferramenta `Ettercap` escuta a conexão e caso haja a presença de um pacote com as características de uma saída analógica de índice 33 a ferramenta demonstra em sua própria interface o dado que compõe o pacote. Nesse caso, a interface do `Ettercap` foi capaz de escutar a presença de um pacote na rede e assim apresentou os seguintes dados sobre o pacote, que é o pacote original enviado pelo mestre do sistema, conforme apresenta a Figura 33.

No momento em que o *plug-in* detecta a presença do pacote desejado na rede é realizada a alteração pré-estabelecida, que nesse caso é modificar o valor de saída analógica para 66. Dessa forma, é possível visualizar na interface do `Ettercap` o pacote enviado ao endereço de destino `10.0.2.2`, conforme apresentado na Figura 34. Na Figura 34 é possível analisar que o resultado desejado pelo ataque foi alcançado, pois o campo *Value*

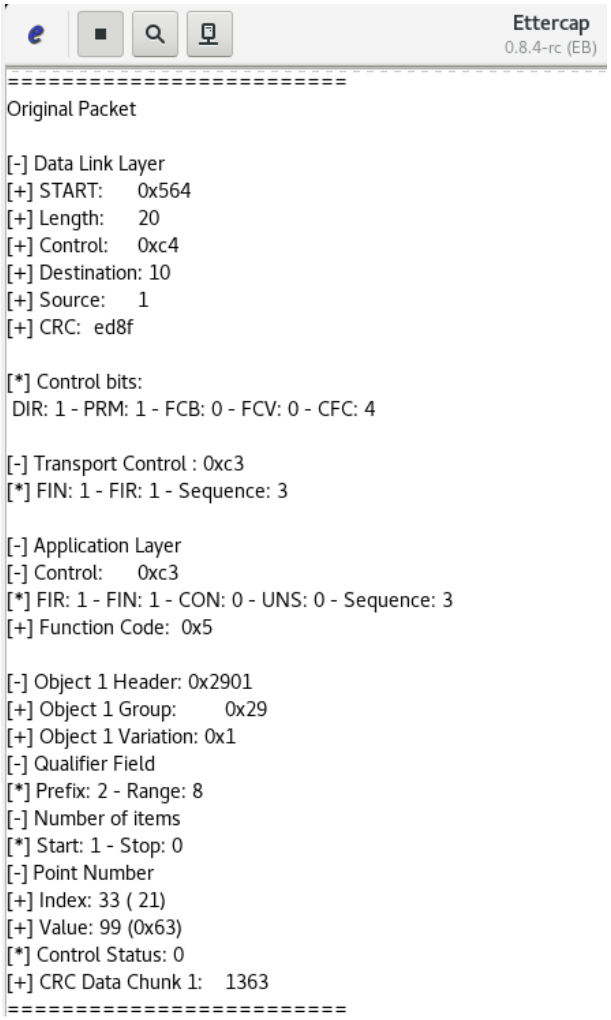

Figura 33: Pacote original captado pelo usuário malicioso.

Fonte: O Autor (2021).

foi alterado de 99 para 66, bem como o CRC *Data Chunk* 1, que alterou seu valor para atender a verificação do novo pacote.

O valor alterado "66" é um valor simbólico, utilizado somente para demonstrar o funcionamento do ataque. Porém, um ataque como esse pode ser considerado crucial a um equipamento presente em um sistema de geração distribuída. Pois, supondo que esse ataque foi responsável por modificar um limite de operação do inversor, como um limite de tensão, por exemplo. Sendo assim, supondo também que nativamente espera-se que o inversor não exceda o limite definido como 70. Portanto, caso o valor 70 seja ultrapassado, um dispositivo auxiliar, programado na estrutura faria a proteção do componente, auxiliando nesse processo de sobretensão para que não ocorra nenhum dano ao equipamento. No entanto, com a atuação do ataque de integridade, o equipamento não

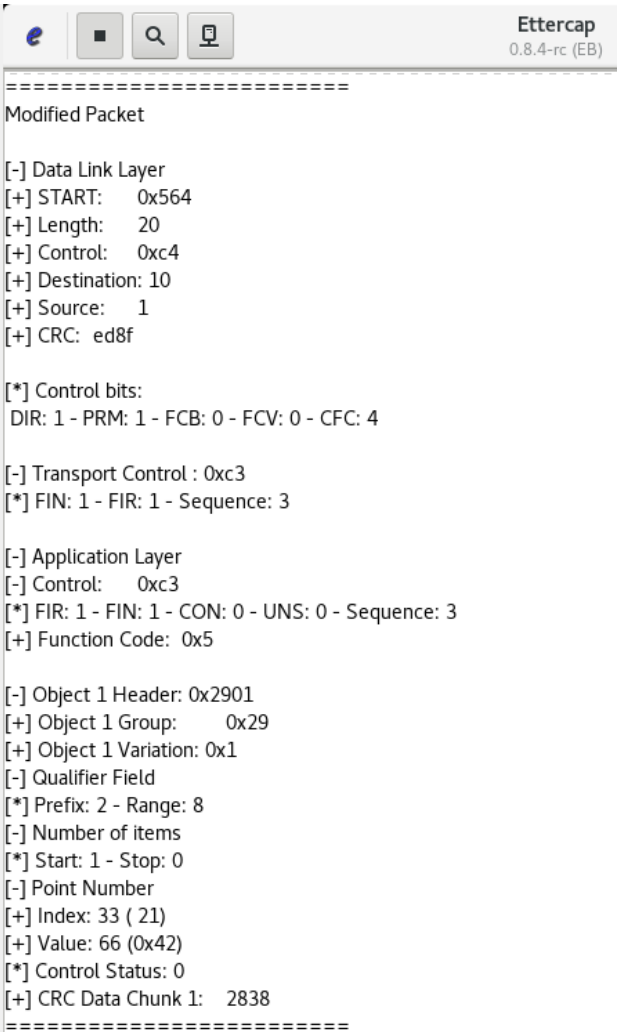

Figura 34: Pacote alterado pelo usuário malicioso.

Fonte: O Autor (2021).

iria reconhecer essa sobretensão. Sendo assim, o dispositivo auxiliar não iria atuar, o que poderia hipoteticamente ocasionar um dano substancial ao componente em questão.

Outra situação hipotética que poderia ocorrer seria durante uma manutenção na rede, por exemplo. Sendo assim, o operador determina que o sistema seja desconectado, enviando um comando para desabilitar a geração. Com um ataque de integridade atuando na rede, o mesmo poderia falsificar esse comando de desconexão e enviar ao *outstation* que o valor referente a esse comando não foi alterado, ou que ainda encontra-se habilitado. O atacante poderia alterar ainda mais dados, com por exemplo, retornar ao sistema de operação que a estrutura foi desabilitada após a requisição e zerar todos os dados de monitoramento da estrutura. Logo, isso poderia ocasionar problemas futuros no momento em que fosse ocorrer a manutenção. Além disso, outros diversos problemas poderiam ocorrer, como na operação do sistema e na medição.

Um exemplo de irregularidade na operação pode ser relacionado a um comando não efetivado. Por exemplo, a concessionária poderia enviar um comando de desconexão dos inversores, mas caso obtivesse um ataque de integridade ativo, esse comando seria ignorado, mantendo assim os inversores gerando energia.

Um ataque de integridade sobre informações de medição, poderia ser relacionado a uma adulteração nos dados sobre a geração. Nesse caso o cliente poderia até mesmo fornecer um dado de geração superior a que o sistema atingiu para a concessionária, resultando assim em fraude.

Pôde-se analisar que a ferramenta `Ettercap` atuou conforme desejado. Porém, para comprovar a eficácia do *plug-in* desenvolvido foi necessário a utilização do analisador de pacotes *Wireshark* em cada uma das extremidades da conexão, para assegurar que o ataque foi efetivado. Sendo assim, na Figura 35 é apresentado o dado de saída analógica de 32 *bits* sendo enviado ao IP `10.0.2.2`, com destino ao *outstation* da estrutura. Porém, nesse caso, com a atuação do ataque MITM na rede, o MAC de destino refere-se ao usuário malicioso e não ao sub-mestre da conexão.

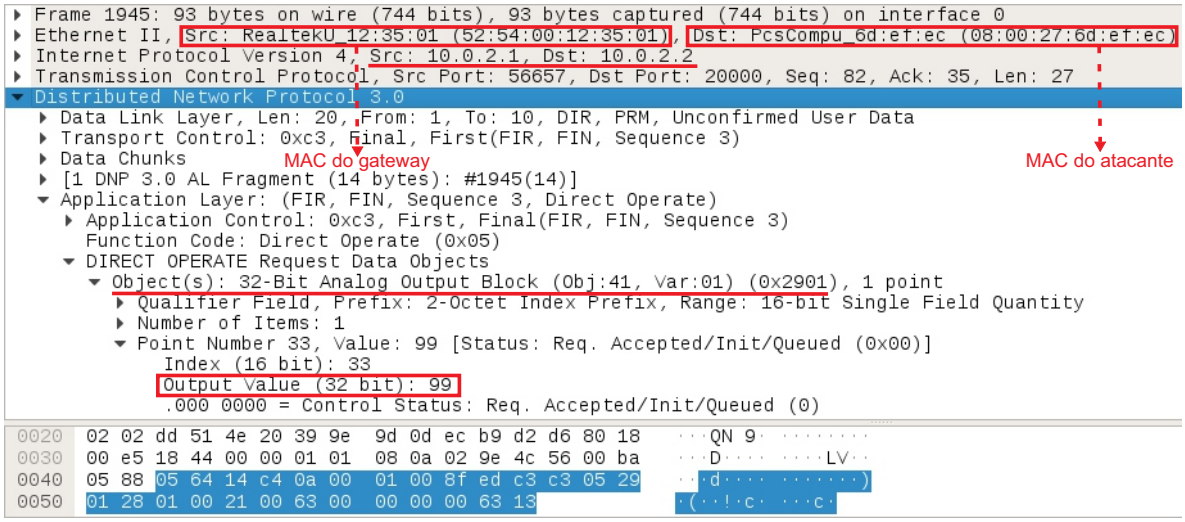

Figura 35: Pacote original enviado pelo gateway do sistema.

Fonte: O Autor (2021).

Com a aplicação do *Wireshark*, no *outstation* do sistema obteve-se os dados apresentados na Figura 36. É possível analisar que o ataque foi bem sucedido. Pois, o usuário remetente refere-se ao endereço de IP `10.0.2.1`, endereço do *gateway* do sistema. Porém, nesse caso o MAC do remetente não é o endereço MAC real do *gateway* e sim o da máquina em que o `Ettercap` está habilitado (`08:00:27:6d:ef:ec`). Além disso, o valor de saída chegado ao *outstation* da conexão foi 66, conforme previamente estabelecido no

plug-in. Esse valor foi alterado de maneira correta, pois o valor emitido originalmente era 99. Sendo assim, o ataque foi concluído de modo satisfatório, alterando somente o valor de saída referente ao dado de saída analógica, conforme desejado.

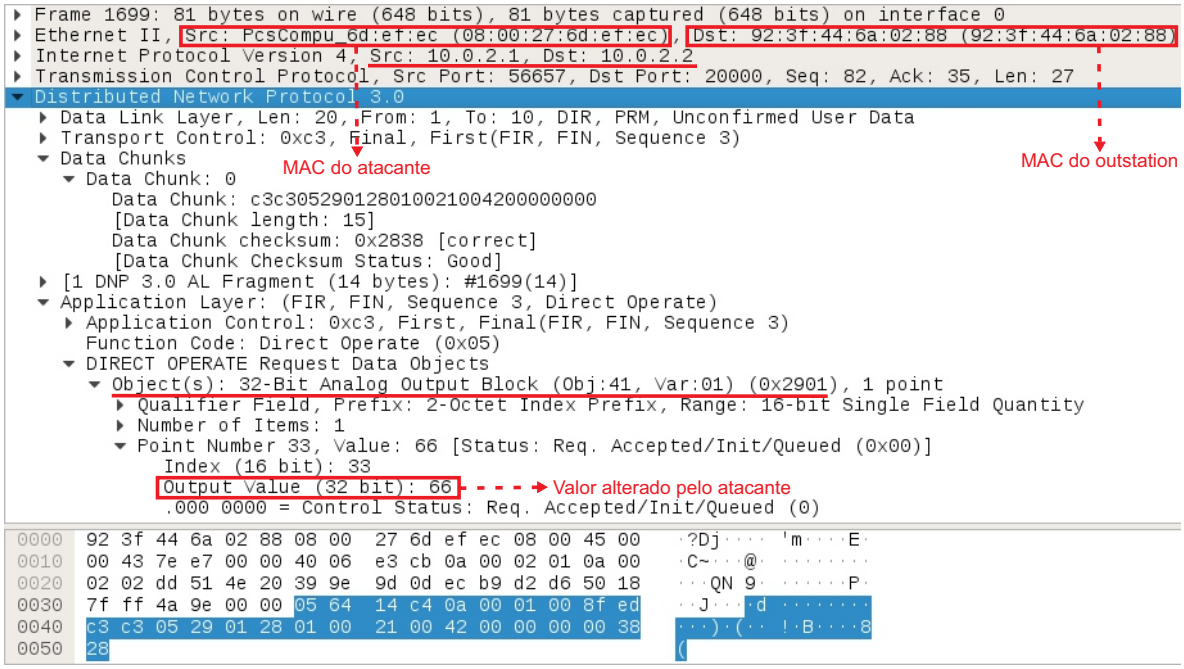

Figura 36: Pacote modificado recebido pelo outstation do sistema.

Fonte: O Autor (2021).

Com uma ferramenta de Autenticação, como o DNP3-SA seria possível identificar o ataque de integridade. No entanto, a estrutura ainda ficaria suscetível a ataques de confidenciabilidade, visto que os dados trafegados na rede não possuíam criptografia.

4.3.3 Análise dos ataques a estrutura DNP3/TLS

Como pôde-se evidenciar, o ataque desenvolvido funciona em uma estrutura DNP3 tradicional. Entretanto, para efetuar o teste da estrutura proposta no trabalho, inicialmente, foi modificada novamente a conexão entre a porta mestre do *gateway* de endereço IP `10.0.2.1` e MAC `52:54:00:12:35:01` e o *outstation* do sistema, de IP `10.0.2.2` e MAC `92:3f:44:6a:02:88`, agora com a aplicação de criptografia e autenticação com o uso do protocolo TLS adicionado ao sistema DNP3. Em seguida, foi conectado o usuário malicioso de endereço IP `10.0.2.3` e MAC `08:00:27:6d:ef:ec` na mesma rede. É interessante ressaltar que as máquinas utilizadas nessa análise são as mesmas usadas no testes anteriores sem o uso do protocolo TLS.

No usuário malicioso, foi aplicado também o uso da ferramenta `Ettercap` e selecionado o *host* de endereço IP 10.0.2.1 e MAC `52:54:00:12:35:01`, bem como o IP `10.0.2.2` e MAC `92:3f:44:6a:02:88` como alvos. Em seguida, foi realizado o ataque *Man-In-The-Middle* de envenenamento ARP. Esse ataque modificou os MAC's conhecidos pelos usuários da rede, para que todos os usuários da conexão repassem informações primeiramente ao usuário malicioso de MAC `08:00:27:6d:ef:ec`.

Com as conexões estabelecidas e o ataque MITM efetuado, foi ativado o *plug-in* desenvolvido para efetuar os ataques de confidenciabilidade e integridade. Em seguida, foi enviado ao *outstation* da conexão um pacote contendo as informações de objeto 41, variação 01, índice 33 com a função `Direct Operate` e valor de saída de 99. No entanto, nesse cenário o *plug-in* não foi capaz de detectar a presença do dado, nem sequer efetuar a alteração. Isso pode ser analisado na interface disponibilizada pela ferramenta `Ettercap`, conforme apresentado na Figura 37.

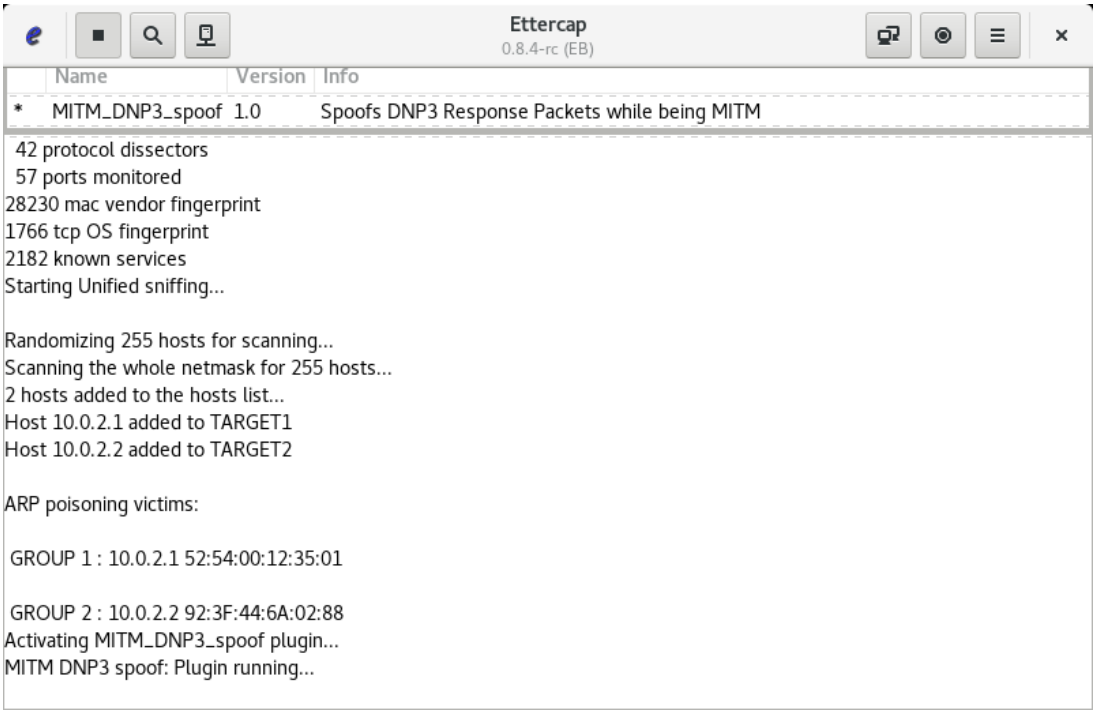

Figura 37: Interface de monitoramento do Ettercap sobre pacote DNP3/TLS.

Fonte: O Autor (2021).

O dado não consegue ser captado, muito menos alterado pelo *plug-in* devido ao fato de que o usuário malicioso não possui autenticação, nem mesmo acesso as chaves criptográficas inseridas no sistema. O que pode ser analisado nas extremidades da conexão, com o auxílio da ferramenta *Wireshark*, agora efetuando somente a filtragem dos

pacotes SSL presentes na conexão, apresentados na Figura 38, com a análise no *gateway* e na Figura 39, analisando o *outstation* do sistema.

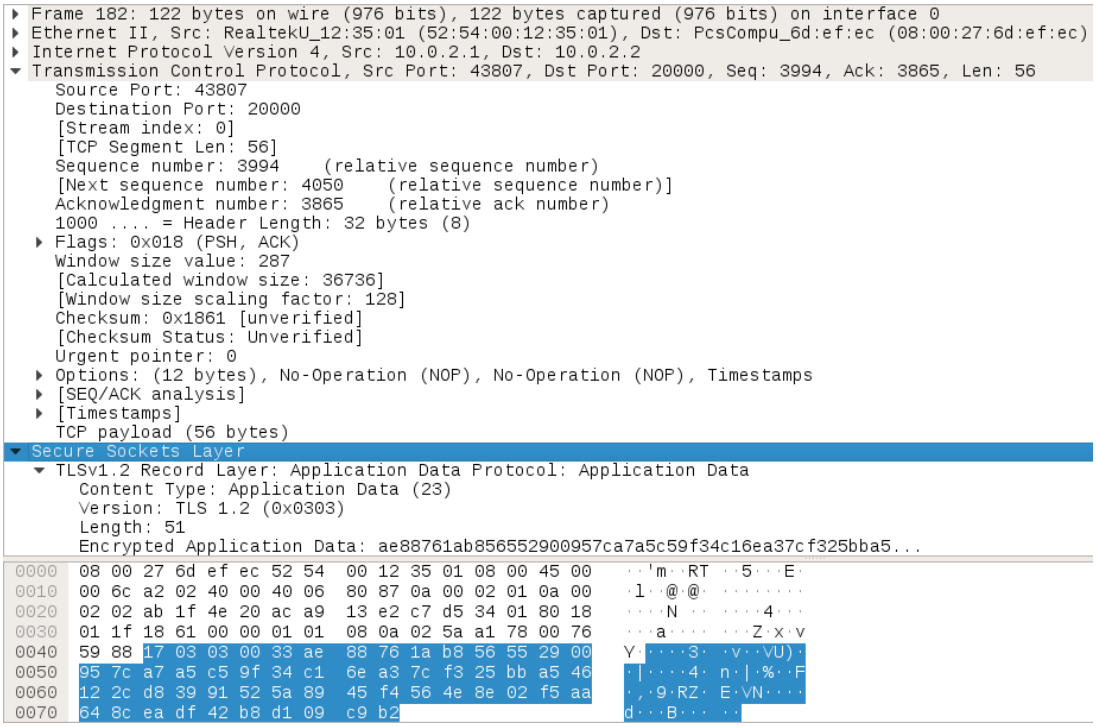

Figura 38: Interface de monitoramento de pacotes TLS no gateway do sistema.

Fonte: O Autor (2021).

Com essas análises é possível verificar que MITM foi realizado, os dados passam pelo intermediário (usuário malicioso). Porém, a ferramenta não é capaz de efetuar a alteração, devido ao fato de que o *plug-in* foi desenvolvido para modificar pacotes DNP3, e agora o pacote transmitido também possui as características do protocolo TLS. Portanto, mesmo o usuário malicioso conseguindo acesso ao pacote, não é capaz de descriptografar sem possuir as chaves e certificados correspondentes.

É importante ressaltar que a utilização do protocolo TLS no sistema não impede que os pacotes sejam retransmitidos por um usuário malicioso. Além disso, esse invasor poderia cortar a comunicação da concessionária com os usuários da rede. No entanto, com o TLS aplicado sobre o sistema não seria possível ler as informações e/ou modificar os dados presentes na estrutura. Nesse cenário a concessionária de energia iria detectar uma perda de comunicação com os usuários e poderia tomar uma contra-medida para identificar rapidamente o ataque.

Somente a aplicação do TLS ao sistema não é capaz de resolver todo o problema de segurança. Por isso, a equipe de TI da concessionária deve monitorar o sistema

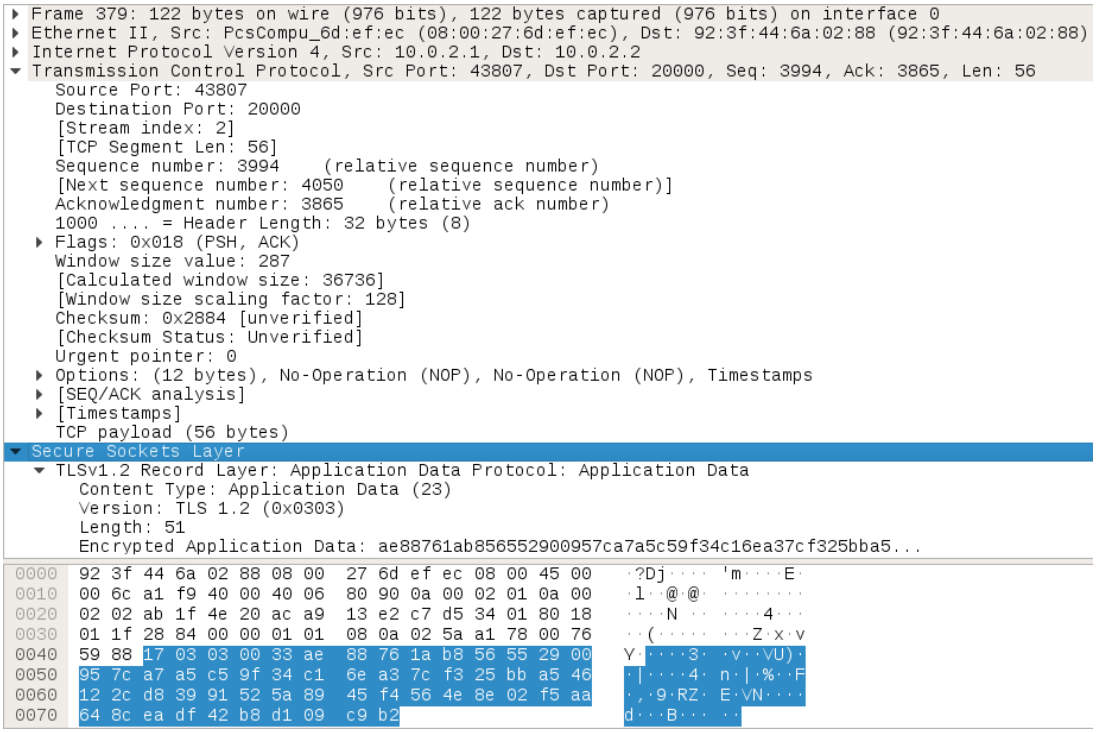

Figura 39: Interface de monitoramento de pacotes TLS no outstation do sistema.

Fonte: O Autor (2021).

continuamente e verificar caso haja algum indício de um possível usuário não autorizado na rede. Além disso, essa verificação pode ser feita com um tempo reduzido, devido a redução da superfície de ataque proporcionada pelo uso do sistema desenvolvido.

4.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Nesse capítulo foram apresentados os formatos de ataques aplicados ao sistema de geração distribuída para análise dos resultados atingidos no trabalho. Sendo assim, inicialmente, foi realizado uma breve introdução apresentando o tipo de ataque utilizado, o MITM de envenenamento ARP e o porquê de sua aplicação. Além disso, com o uso desse formato foram aplicados dois ataques, o de confidenciabilidade, que age na leitura de dados presentes em sistemas exclusivos, bem como o ataque de integridade, voltado a falsificar um dado real do sistema. Sendo assim, foi demonstrada a forma desses ataques a uma estrutura de geração distribuída composta pelo protocolo DNP3. Também foi discutido sobre a captação de informações restritas de sistemas exclusivos e sobre a modificação dos dados reais de sistemas. Além disso, foram apresentados também alguns problemas hipotéticos que podem ocorrer quando há a possibilidade de modificar os dados presentes em um sistema *Smart Grid*. No entanto, os danos que ataques de integridade podem

causar são imensuráveis. Pois, basicamente depende do formato do dado modificado e da categoria de atuação da estrutura invadida.

Pôde-se evidenciar também que sistemas que utilizam o protocolo DNP3 possuem uma significativa escassez de segurança. Pois, somente com o uso de uma ferramenta de monitoramento de pacotes é possível estruturar um dado DNP3. Além disso, com essas informações é possível desenvolver uma ferramenta capaz de monitorar e modificar dados cruciais da estrutura sem que os usuários notem a invasão. No entanto, pôde-se verificar também que no momento que o sistema é submetido a uma conexão DNP3/TLS os ataques são inviabilizados. Por isso, é possível garantir que ferramentas que proporcionam autenticação e criptografia a estruturas críticas são consideradas essenciais.

5 CONCLUSÃO

Nesse trabalho foi apresentado uma ferramenta capaz de desempenhar a função de ampliação na segurança de uma estrutura SCADA de geração distribuída baseada no protocolo DNP3. Essa segurança foi aprimorada com base na adição de criptografia e autenticação com o uso do protocolo TLS, bem como pelo fato de realizar a inversão da conexão, fazendo com que a concessionária de energia, o mestre do sistema, seja o servidor da conexão. A adição de criptografia e autenticação com base no protocolo TLS a uma comunicação DNP3 é considerada satisfatória, pois trata-se de uma solução eficaz e validada pelos resultados obtidos no trabalho.

A adição de formatos capazes de prover segurança a ferramentas utilizando protocolos abertos, como é o caso da aplicação do TLS em comunicações DNP3 é algo indispensável. Além disso, com a utilização de autenticação e criptografia, os dados presentes no sistema serão restritos a usuários autenticados com os certificados do TLS. Dessa forma, os ataques serão reduzidos a essa estrutura. Portanto, certamente o trabalho desenvolvido é de extrema relevância para os setores de geração distribuída, visto que ocorreram diversos ataques ao longo da história a estruturas críticas semelhantes à utilizada no trabalho.

Com a utilização do *gateway* desenvolvido em sistemas de energia será possível maximizar a proteção sobre ameaças externas a estrutura. Pois, irá dificultar a invasão e alteração de dados críticos por um usuário indesejado. Além disso, a parte fundamental do *gateway* elaborado refere-se ao fato de que a porta DNP3 não segura disponibilizada para atender a necessidade do sistema de supervisão somente escuta conexões locais. Sendo assim, somente os clientes autorizados, com as chaves e certificados reais da estrutura serão capazes de se conectar ao *gateway*, o que possibilita uma redução de usuários clandestinos no sistema. Dessa forma, somente o *gateway* da concessionária necessita de um IP público e concede a porta para conexão DNP3 a Internet. Então, desde que a ferramenta `opendatacon` composta pelas alterações desenvolvidas no presente trabalho esteja conectada a um ambiente seguro, a transmissão de dados para o sistema de supervisão será segura.

É importante salientar que a arquitetura implementada encontra-se em operação atualmente, livre de erros e executando a comunicação desejada desde as requisições do sistema de supervisão mestre até a estação remota composta pelo sistema de geração distribuída, conforme pretendido. Além disso, foram realizados testes de segurança que foram capazes de validar a proteção disponibilizada a estrutura pela ferramenta desenvolvida. Esses testes simularam ataques tradicionais conhecidos na história, baseados no formato *Man-In-The-Middle*, que foram desenvolvidos para a validação do presente trabalho. Pôde-se notar que o resultado obtido foi satisfatório, pois a ferramenta atacante desenvolvida foi capaz de interceptar e modificar pacotes DNP3 durante os ataques de confidenciabilidade e integridade em um sistema SCADA de geração distribuída não-seguro. Entretanto, quando nesse mesmo sistema foi aplicado o uso da estrutura de segurança proposta no presente trabalho, os ataques foram inviabilizados.

A ferramenta de ataque desenvolvida foi capaz de efetuar um ataque de integridade corretamente em sistemas não seguros. Um ataque como esse poderia, por exemplo, ocasionar tomada de decisões errôneas devido a retransmissão de um dado falso na rede, podendo ocasionar desde de um cálculo falso de tarifação, até mesmo uma falha crítica de algum dispositivo crucial da estrutura. Pois, um dado modificado é capaz de demonstrar que um inversor está operando normalmente, quando na realidade pode estar atuando sobre falhas, sobrecargas, entre outras possibilidades. No entanto, a ferramenta demonstrou-se incapaz de agir contra o sistema de segurança proposto, o que comprova a necessidade de implementação do protocolo TLS a estruturas compostas por protocolos abertos, como o DNP3.

É relevante salientar que a aplicação do sistema desenvolvido não é capaz de solucionar todos os problemas de segurança. No entanto, com a redução da superfície de ataque e o uso do TLS em estruturas de geração distribuída, usuários malicioso poderiam somente cortar comunicações, o que ainda poderia ocasionar um transtorno indesejado. Entretanto, caso isso ocorra, com o monitoramento constante do sistema por uma equipe de TI é possível reduzir o tempo para identificação de usuários não autorizados na estrutura devido a característica presente no sistema de segurança proposto.

Pôde-se observar então que o sistema de segurança desenvolvido atua corretamente. Pois, o suporte a segurança advindo do sistema proposto é capaz de ampliar a proteção de uma estrutura de geração distribuída. Além disso, o *gateway* desenvolvido foi capaz de prover a segurança necessária a ferramentas sem suporte a atualizações, ou que possuam limitações. Pois, no presente trabalho foi incluso o *gateway* para atender a

um sistema de supervisão que não possui suporte ao protocolo TLS. Sendo assim, com a aplicação desse formato foi possível fornecer a rede pública somente dados protegidos pela autenticação e criptografia oriundos desse *software*. Dessa forma, diversas ferramentas ou plantas existentes que não possuam proteção a usuários externos podem ser asseguradas com o uso do *gateway*, fornecendo a esses dispositivos e/ou estruturas a segurança necessária.

5.1 SUGESTÃO PARA TRABALHOS FUTUROS

Com base no estudo desenvolvido durante esse trabalho, foi possível encontrar algumas questões que são de extrema relevância e poderiam ser abordadas em trabalhos futuros, como por exemplo:

- Integrar o sistema proposto à um sistema de detecção de intrusão;
- Integrar o sistema proposto à um sistema de prevenção de intrusão.

A inclusão de sistemas de detecção e prevenção de intrusão ao *gateway* DNP3 seguro seria de suma importância, pois esses sistemas são capazes de amplificar a segurança obtida no trabalho. Por exemplo, o sistema de prevenção de intrusão poderia ser útil para interferir que outros usuários acessem a rede. Sendo assim, a adição de um sistema como esse a estrutura desenvolvida possibilitaria identificar e bloquear todo o tráfego proveniente de usuários considerados maliciosos pelo sistema. Além disso, a inclusão de um sistema de detecção de intrusão a estrutura proporcionaria detectar eventos incomuns na rede com uma eficiência superior.

5.2 TRABALHOS PUBLICADOS

FERST, MATHEUS K. ; DE FIGUEIREDO, HUGO F. M. ; DENARDIN, GUSTAVO ; LOPES, JULIANO . Implementation of Secure Communication With Modbus and Transport Layer Security protocols. **In: 13th IEEE International Conference on Industry Applications (INDUSCON)**, São Paulo, Brazil, 2018. p. 155. doi: <10.1109/induscon.2018.8627306>

DE FIGUEIREDO, HUGO F. M.; FERST, MATHEUS K.; DENARDIN, GUSTAVO W. An Overview About Detection of Cyber-Attacks on Power SCADA Systems. **In: 5th IEEE**

Southern Power Electronics Conference (IEEE SPEC 19) joint with 15th Brazilian Power Electronics Conference (COBEP 19), Santos, Brazil, 2019, pp. 1-6. doi: <10.1109/cobep/spec44138.2019.9065353>

FERST, MATHEUS K. ; DE FIGUEIREDO, HUGO F. M. ; DENARDIN, GUSTAVO W. . Connection Time in Modbus/TLS for Secure Communications on Photovoltaic Systems. **In: 5th IEEE Southern Power Electronics Conference (IEEE SPEC 19) joint with 15th Brazilian Power Electronics Conference (COBEP 19)**, Santos, Brazil, 2019, p. 1. doi: <10.1109/cobep/spec44138.2019.9065406>

REFERÊNCIAS

Abdallah, A.; Shen, X. S. Efficient prevention technique for false data injection attack in smart grid. In: **2016 IEEE International Conference on Communications (ICC)**. [S.l.: s.n.], 2016. p. 1–6.

ALMALAWI, Abdulmohsen; FAHAD, Adil; TARI, Zahir; ALAMRI, Abdullah; ALGHAMDI, Rayed; ZOMAYA, Albert Y. An Efficient Data-Driven Clustering Technique to Detect Attacks in SCADA Systems. **IEEE Transactions on Information Forensics and Security**, IEEE, v. 11, n. 5, p. 893–906, 2016.

ASSANTE, Michael J; LEE, Robert M. The industrial control system cyber kill chain. **SANS Institute InfoSec Reading Room**, v. 1, 2015.

BAIGENT, Drew; ADAMIAK, Mark; MACKIEWICZ, Ralph; SISCO, GMGM. Iec 61850 communication networks and systems in substations: An overview for users. **SISCO Systems**, 2004.

BARBOSA, Rafael Ramos Regis; SADRE, Ramin; PRAS, Aiko. A first look into scada network traffic. In: IEEE. **2012 IEEE Network Operations and Management Symposium**. [S.l.], 2012. p. 518–521.

BOYER, Stuart A. **SCADA: supervisory control and data acquisition**. [S.l.]: International Society of Automation, 2009.

BREWER, Dennis C. **Security Controls For Sarbanes – Oxley Section 404-I**. [S.l.]: John Wiley & Sons, 2005.

BURNETT, Mark. **Perfect password: Selection, protection, authentication**. [S.l.]: Elsevier, 2006.

CAMPBELL, Richard J. **Cybersecurity Issues for the bulk power system**. [S.l.]: Congressional Research Service, 2015.

CASE, Defense Use. Analysis of the cyber attack on the ukrainian power grid. **Electricity Information Sharing and Analysis Center (E-ISAC)**, 2016.

CHALAMASETTY, Goutham K; MEMBER, Student. Secure SCADA Communication Network for Detecting and Preventing Cyber-Attacks on Power Systems. **2016 Clemson University Power Systems Conference (PSC)**, IEEE, p. 1–7, 2016.

CHEN, Bo; HO, Daniel WC; HU, Guoqiang; YU, Li. Secure fusion estimation for bandwidth constrained cyber-physical systems under replay attacks. **IEEE transactions on cybernetics**, IEEE, v. 48, n. 6, p. 1862–1876, 2017.

CHEN, Thomas; ABU-NIMEH, Saeed. Lessons from stuxnet. **Computer**, v. 44, n. 4, p. 91–93, 2011.

CLARKE, Gordon; REYNDERS, Deon; WRIGHT, Edwin. **Practical modern SCADA protocols: DNP3, 60870.5 and related systems**. [S.l.]: Newnes, 2004.

CONTI, Mauro; MEMBER, Senior; DRAGONI, Nicola; LESYK, Viktor. A Survey of Man In The Middle Attacks. **IEEE Communications Surveys & Tutorials**, IEEE, v. 18, n. 3, p. 2027–2051, 2016.

CRAIN, Adam. Dnp3 sav5 and tls: Different trust boundaries. 2013. Disponível em: <https://stepfunc.io/blog/sa_vs_tls_trust_boundary/>. Acesso em: 09 de fevereiro de 2021.

CREMERS, Cas; DEHNEL-WILD, Martin; MILNER, Kevin. Secure authentication in the grid: A formal analysis of dnp3 sav5. **Journal of Computer Security**, IOS Press, v. 27, n. 2, p. 203–232, 2019.

DIOVU, RC; AGEE, JT. A cloud-based openflow firewall for mitigation against ddos attacks in smart grid ami networks. In: IEEE. **2017 IEEE PES PowerAfrica**. [S.l.], 2017. p. 28–33.

DIOVU, R C; AGEE, J T. Quantitative Analysis of Firewall Security under DDoS Attacks in Smart Grid AMI Networks. **2017 IEEE 3rd International Conference on Electro-Technology for National Development (NIGERCON)**, p. 1–6, 2017.

DNP, Users Group. Dnp3 profile for communications with distributed energy resources (ders). In: **Technical Bulletin DNP Application Note AN2018-001**. [S.l.: s.n.], 2018. p. 1–251.

DO, Van Long; FILLATRE, Lionel; CÔTE, Université; ANTIPOLIS, Sophia; NIKIFOROV, Igor; TECHNOLOGIE, Université De. Feature Article : Security of SCADA Systems Against Cyber-Physical Attacks. **IEEE Aerospace and Electronic Systems Magazine**, IEEE, v. 32, n. 10, p. 28–45, 2017.

DU, Xiaojiang; SHAYMAN, M; ROZENBLIT, Moshe. Implementation and performance analysis of snmp on a tls/tcp base. In: IEEE. **2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings. Integrated Network Management VII. Integrated Management Strategies for the New Millennium (Cat. No. 01EX470)**. [S.l.], 2001. p. 453–466.

EAST, Samuel; BUTTS, Jonathan; PAPA, Mauricio; SHENOI, Sujeet. A taxonomy of attacks on the dnp3 protocol. In: SPRINGER. **International Conference on Critical Infrastructure Protection**. [S.l.], 2009. p. 67–81.

FALLIERE, Nicolas; MURCHU, Liam O; CHIEN, Eric. W32. stuxnet dossier. **White paper, Symantec Corp., Security Response**, v. 5, n. 6, p. 29, 2011.

FARUK, ABM. **Testing and exploring vulnerabilities of the applications implementing DNP3 protocol**. Dissertação (Mestrado) — Institutt for Telematikk, 2008.

FARWELL, James P; ROHOZINSKI, Rafal. Stuxnet and the future of cyber war. **Survival**, Taylor & Francis, v. 53, n. 1, p. 23–40, 2011.

FIGUEIREDO, Hugo. Dnp3 integrity and confidentialy attack plugin repository. 2021. Disponível em: <https://gitlab.com/hugoffigueiredo/integrity-and-confidentialy-attack-plugin>. Acesso em: 05 de março de 2021.

FIGUEIREDO, Hugo Fernando Magalhães de; PODOLAK, Lucas; SCHULTZ, Lilian Rosana Kremer. Projeto e desenvolvimento de um sistema fotovoltaico autônomo voltado a área rural. **Revista Técnico-Científica**, n. 15, 2018.

FOGLIETTA, Chiara; MASUCCI, Dario; PALAZZO, Cosimo; SANTINI, Riccardo; PANZIERI, Stefano; ROSA, Luis; CRUZ, Tiago; LEV, Leonid. From detecting cyber-attacks to mitigating risk within a hybrid environment. **IEEE Systems Journal**, IEEE, n. 99, p. 1–12, 2018.

GIANI, Annarita; BITAR, Eilyan; GARCIA, Manuel; MCQUEEN, Miles; KHARGONEKAR, Pramod; POOLLA, Kameshwar. Smart grid data integrity attacks: characterizations and countermeasures π. In: IEEE. **2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)**. [S.l.], 2011. p. 232–237.

GILCHRIST, Grant. Secure authentication for dnp3. In: IEEE. **2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century**. [S.l.], 2008. p. 1–3.

GROUP, DNP Users. Dnp3 application note an2018-001 - dnp3 profile for communications with distributed resources. 2018. Disponível em: <https://www.dnp.org/Resources/Document-Library?folderld=1261>. Acesso em: 22 de janeiro de 2020.

GUIMARAES, Pedro Henrique V; MURILLO, Andrés; ANDREONI, Martin; MATTOS, Diogo MF; FERRAZ, Lyno Henrique G; PINTO, Fabio Antonio V; COSTA, Luis Henrique MK; DUARTE, Otto Carlos MB. Comunicaç ao em redes elétricas inteligentes: Eficiência, confiabilidade, segurança e escalabilidade. **Minicursos do XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos SBRC'13**, p. 101–164, 2013.

HAHN, Genebeck; KWON, Taekyoung; SONG, Jooseok. **A Comparative Analysis of Extensible Authentication Protocols**. [S.l.]: Dept. of Software Engineering of Sejong University, 2005.

HILAL, Hamzah; NANGIM, Anas. Network Security Analysis SCADA System Automation on Industrial Process. **2017 International Conference on Broadband Communication, Wireless Sensors and Powering (BCWSP)**, 2017.

HUANG, Wei. Learn iec 61850 configuration in 30 minutes. In: IEEE. **2018 71st Annual Conference for Protective Relay Engineers (CPRE)**. [S.l.], 2018. p. 1–5.

HUG, Gabriela; GIAMPAPA, Joseph Andrew. Vulnerability Assessment of AC State Estimation With Respect to False Data Injection Cyber-Attacks. **IEEE Transactions on Smart Grid**, IEEE, v. 3, n. 3, p. 1362–1370, 2012.

HUNG, Duong Quoc; MITHULANANTHAN, N; BANSAL, RC. Analytical strategies for renewable distributed generation integration considering energy loss minimization. **Applied Energy**, Elsevier, v. 105, p. 75–85, 2013.

HUPP, William; HASANDKA, Adarsh; CARVALHO, Ricardo Siqueira de; SALEEM, Danish. Module-ot: A hardware security module for operational technology. In: IEEE. **2020 IEEE Texas Power and Energy Conference (TPEC)**. [S.l.], 2020. p. 1–6.

IGURE, Vinay M.; LAUGHTER, Sean A.; WILLIAMS, Ronald D. Security issues in scada networks. **Computers & Security**, v. 25, n. 7, p. 498 – 506, 2006. ISSN 0167-4048. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167404806000514>.

IRITA, Takashi; NAMERIKAWA, Toru. Detection of Replay Attack on Smart Grid with Code Signal and Bargaining Game. **2017 American Control Conference (ACC)**, AACC, p. 2112–2117, 2017.

JIN, Dong; NICOL, David M; YAN, Guanhua. An event buffer flooding attack in dnp3 controlled scada systems. In: IEEE. **Proceedings of the 2011 Winter Simulation Conference (WSC)**. [S.l.], 2011. p. 2614–2626.

KALLURI, Rajesh; MAHENDRA, Lagineni. Simulation and Impact Analysis of Denial-of-Service Attacks on Power SCADA. **2016 National Power Systems Conference (NPSC)**, IEEE, n. 1, p. 1–5, 2016.

KANG, BooJoong; MAYNARD, Peter; MCLAUGHLIN, Kieran; SEZER, Sakir; ANDRÉN, Filip; SEITL, Christian; KUPZOG, Friederich; STRASSER, Thomas. Investigating cyber-physical attacks against iec 61850 photovoltaic inverter installations. In: IEEE. **2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)**. [S.l.], 2015. p. 1–8.

KANG, Boojoong; MAYNARD, Peter; MCLAUGHLIN, Kieran; SEZER, Sakir; ANDRÉN, Filip; SEITL, Christian; KUPZOG, Friederich; STRASSER, Thomas. Investigating Cyber-Physical Attacks against IEC 61850 Photovoltaic Inverter Installations. **2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)**, IEEE, p. 1–8, 2015.

KHAN, Rafiullah; MAYNARD, Peter; MCLAUGHLIN, Kieran; LAVERTY, David M; SEZER, Sakir. Threat analysis of blackenergy malware for synchrophasor based real-time control and monitoring in smart grid. **ICS-CSR**, v. 16, p. 1–11, 2016.

KIM, Jinsub; TONG, Lang. On topology attack of a smart grid: Undetectable attacks and countermeasures. **IEEE Journal on Selected Areas in Communications**, IEEE, v. 31, n. 7, p. 1294–1305, 2013.

KNAPP, Eric D.; LANGILL, Joel Thomas. Chapter 7 - hacking industrial control systems. In: KNAPP, Eric D.; LANGILL, Joel Thomas (Ed.). **Industrial Network Security (Second Edition)**. Second edition. Boston: Syngress, 2015. p. 171 – 207. ISBN 978-0-12-420114-9. Disponível em: <http://www.sciencedirect.com/science/article/pii/B9780124201149000071>.

KRUTZ, Ronald L. **Securing SCADA systems**. [S.l.]: John Wiley & Sons, 2006.

LANGER, Lucie; SMITH, Paul; HUTLE, Martin; SCHAEFFER-FILHO, Alberto. Analysing Cyber-physical Attacks to a Smart Grid : A Voltage Control Use Case. **2016 Power Systems Computation Conference (PSCC)**, Power Systems Computation Conference, p. 1–7, 2016.

LANGILL, Joel T. Defending against the dragonfly cyber security attacks. **Retrieved**, v. 11, p. 2015, 2014.

LANGNER, Ralph. Stuxnet: Dissecting a cyberwarfare weapon. **IEEE Security & Privacy**, IEEE, v. 9, n. 3, p. 49–51, 2011.

LANGNER, Ralph. To kill a centrifuge: A technical analysis of what stuxnet's creators tried to achieve. **The Langner Group**, 2013.

LEVILLAIN, Olivier; GOURDIN, Baptiste; DEBAR, Hervé. Tls record protocol: Security analysis and defense-in-depth countermeasures for https. In: **Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security**. [S.l.: s.n.], 2015. p. 225–236.

LIANG, Gaoqi; ZHAO, Junhua; LUO, Fengji; WELLER, Steven R; DONG, Zhao Yang. A review of false data injection attacks against modern power systems. **IEEE Transactions on Smart Grid**, IEEE, v. 8, n. 4, p. 1630–1638, 2016.

LIU, Ting; TIAN, Jue; GUI, Yuhong; LIU, Yang; LIU, Pengfei. Sedea: State estimation-based dynamic encryption and authentication in smart grid. **IEEE Access**, IEEE, v. 5, p. 15682–15693, 2017.

LIU, Yao; NING, Peng; REITER, Michael K. False data injection attacks against state estimation in electric power grids. **ACM Transactions on Information and System Security (TISSEC)**, ACM, v. 14, n. 1, p. 13, 2011.

LOPES, Yona; FRANCO, Ricardo Henrique Frazao; MOLANO, David Acosta; SANTOS, Margareth Apostolo dos; CALHAU, Flávio Galvao; BASTOS, Carlos Alberto Malcher; MARTINS, Joberto SB; FERNANDES, Natalia Castro. Smart grid e iec 61850: Novos desafios em redes e telecomunicaçoes para o sistema elétrico. **XXX Simpósio Brasileiro de Telecomunicações**, 2012.

LU, Kuan-Chu; LIU, I-Hsien; LI, Jung-Shian. Venerability and protection tool surveys of industrial control system. In: IEEE. **2018 Sixth International Symposium on Computing and Networking Workshops (CANDARW)**. [S.l.], 2018. p. 511–515.

LU, Xiang; LU, Zhuo; WANG, Wenye; MA, Jianfeng. On network performance evaluation toward the smart grid: A case study of dnp3 over tcp/ip. In: IEEE. **2011 IEEE Global Telecommunications Conference-GLOBECOM 2011**. [S.l.], 2011. p. 1–6.

MACKIEWICZ, Ralph E. Overview of iec 61850 and benefits. In: IEEE. **2006 IEEE Power Engineering Society General Meeting**. [S.l.], 2006. p. 8–pp.

MAGLARAS, Leandros A; JIANG, Jianmin; CRUZ, Tiago J. Combining ensemble methods and social network metrics for improving accuracy of OCSVM on intrusion detection in SCADA systems. **Journal of Information Security and Applications**, Elsevier Ltd, v. 30, p. 15–26, 2016. ISSN 2214-2126. Disponível em: <http://dx.doi.org/10.1016/j.jisa.2016.04.002>.

MAJDALAWIEH, Munir; PARISI-PRESICCE, Francesco; WIJESEKERA, Duminda. Dnpsec: Distributed network protocol version 3 (dnp3) security framework. In: **Advances in Computer, Information, and Systems Sciences, and Engineering**. [S.l.]: Springer, 2007. p. 227–234.

MATROSOV, Aleksandr; RODIONOV, Eugene; HARLEY, David; MALCHO, Juraj. Stuxnet under the microscope. **ESET LLC (September 2010)**, 2010.

MO, Yilin; CHABUKSWAR, Rohan; MEMBER, Student. Detecting Integrity Attacks on SCADA Systems. **IEEE Transactions on Control Systems Technology**, IEEE, v. 22, n. 4, p. 1396–1407, 2014.

MODBUS. Modbus/tcp security. In: . 2018. Disponível em: <http://modbus.org/docs/MB-TCP-Security-v21_2018-07-24.pdf>. Acesso em: 10 de abril de 2020.

OPPLIGER, Rolf. **SSL and TLS: Theory and Practice**. [S.l.]: Artech House, 2016.

ORNAGHI, Alberto; VALLERI, Marco. Ettercap project repository. 2014. Disponível em: <https://github.com/Ettercap/ettercap>.

PAAR, Christof; PELZL, Jan. **Understanding cryptography: a textbook for students and practitioners**. [S.l.]: Springer Science & Business Media, 2009.

PANDEY, Rajendra Kumar; MISRA, Mohit. Cyber Security Threats - Smart Grid Infrastructure. **2016 National Power Systems Conference (NPSC)**, IEEE, p. 1–6, 2016.

PESTANA, Filipe. Scada ettercap mitm repository. 2019. Disponível em: <https://github.com/filipepestana/SCADA-ettercap-MITM>. Acesso em: 02 de março de 2021.

POUW, Keesje Duarte. **Segurança na arquitetura TCP/IP: de firewalls à canais seguros**. 1999.

PULTAROVA, Tereza. News briefing: Cyber security-ukraine grid hack is wake-up call for network operators. **Engineering & Technology**, IET, v. 11, n. 1, p. 12–13, 2016.

Ramalho, L. A.; Shinoda, A. A.; do Nascimento, V. E.; de Oliveira, R.; Ferreira, E. T. Modeling of state machines in vhdl for encapsulation of dnp3 protocol in p2p zigbee network. In: **2013 IEEE PES Conference on Innovative Smart Grid Technologies (ISGT Latin America)**. [S.l.: s.n.], 2013. p. 1–8.

RESPONSE, Symantec Incident. **Dragonfly: Cyberespionage Attacks Against Energy Suppliers**. [S.l.], 2014.

RISTIC, Ivan. **Bulletproof SSL and TLS: Understanding and Deploying SSL/TLS and PKI to Secure Servers and Web Applications**. [S.l.]: Feisty Duck, 2013.

RIZZETTI, Tiago Antonio; WESSEL, Pedro; RODRIGUES, Alexandre Silva; SILVA, Bolívar Menezes da; MILBRADT, Rafael; CANHA, Luciane Neves. Cyber security and communications network on SCADA systems in the context of Smart Grids. **2015 50th International Universities Power Engineering Conference (UPEC)**, IEEE, p. 1–6, 2015.

ROSBOROUGH, Clifford; GORDON, Colin; WALDRON, Brian. All about eve: Comparing dnp3 secure authentication with standard security technologies for scada communications. 2019.

ROSENBERG, J. Chapter 6 security in embedded systems. In: VEGA, Augusto; BOSE, Pradip; BUYUKTOSUNOGLU, Alper (Ed.). **Rugged Embedded Systems**. Boston: Morgan Kaufmann, 2017. p. 149 – 205. ISBN 978-0-12-802459-1. Disponível em: <http://www.sciencedirect.com/science/article/pii/B9780128024591000063>.

SALEEM, Danish; CARTER, Cedric. **Certification Procedures for Data and Communications Security of Distributed Energy Resources**. [S.l.], 2019.

SGIP. Iec 61850 information model concepts and updates for distributed energy resources (der) use cases and functions. In: SGIP. [S.l.], 2015. p. 1–68.

SRIDHAR, Siddharth; MANIMARAN, G. Data Integrity Attacks and their Impacts on SCADA Control System. **IEEE PES General Meeting**, IEEE, p. 1–6, 2010.

STEPHENS, Neil. Open source data concentrator repository. 2014. Disponível em: <https://github.com/neilstephens/opendatacon>. Acesso em: 10 de abril de 2020.

THOMAS, Stephen. Ssl and tls essentials. **New Yourk**, p. 3, 2000.

TOLEDO, Olga Moraes; FILHO, Delly Oliveira; DINIZ, Antônia Sônia Alves Cardoso. Distributed photovoltaic generation and energy storage systems: A review. **Renewable and Sustainable Energy Reviews**, Elsevier, v. 14, n. 1, p. 506–511, 2010.

VAUDENAY, Serge. **A classical introduction to cryptography: Applications for communications security**. [S.l.]: Springer Science & Business Media, 2006.

VENKATACHARY, Sampath Kumar; PRASAD, Jagdish; SAMIKANNU, Ravi. Economic impacts of cyber security in energy sector: a review. **International Journal of Energy Economics and Policy**, v. 7, n. 5, p. 250–262, 2017.

VIEGA, John; MESSIER, Matt. **Secure Programming Cookbook for C and C++: Recipes for Cryptography, Authentication, Input Validation & More**. [S.l.]: "O'Reilly Media, Inc.", 2003.

WANG, Kun; DU, Miao; MAHARJAN, Sabita; SUN, Yanfei. Strategic honeypot game model for distributed denial of service attacks in the smart grid. **IEEE Transactions on Smart Grid**, IEEE, v. 8, n. 5, p. 2474–2482, 2017.

WANG, Yi; AMIN, Mahmoud M; FU, Jian; MOUSSA, Heba B. A novel data analytical approach for false data injection cyber-physical attack mitigation in smart grids. **IEEE Access**, IEEE, v. 5, p. 26022–26033, 2017.

WEERATHUNGA, Pubudu Eroshan. Securing ieds against cyber threats in critical substation automation and industrial control systems. In: IEEE. **2017 70th Annual Conference for Protective Relay Engineers (CPRE)**. [S.l.], 2017. p. 1–20.

XU, Ruzhi; WANG, Rui; GUAN, Zhitao; WU, Longfei; WU, Jun; DU, Xiaojiang. Achieving efficient detection against false data injection attacks in smart grid. **IEEE Access**, IEEE, v. 5, p. 13787–13798, 2017.

YADAV, Suman Avdhesh; KUMAR, Shipra Ravi; SHARMA, Smita; SINGH, Akanksha. A Review of Possibilities and Solutions of Cyber Attacks in Smart Grids. **2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)**, IEEE, n. Iciccs, p. 60–63, 2016.

YANG, Qingyu; YANG, Jie; YU, Wei; AN, Dou. On False Data-Injection Attacks against Power System State Estimation : Modeling and Countermeasures. **IEEE Transactions on Parallel and Distributed Systems**, IEEE, v. 25, n. 3, p. 717–729, 2014.

Yang, Y.; McLaughlin, K.; Sezer, S.; Littler, T.; Im, E. G.; Pranggono, B.; Wang, H. F. Multiattribute scada-specific intrusion detection system for power networks. **IEEE Transactions on Power Delivery**, v. 29, n. 3, p. 1092–1102, June 2014. ISSN 0885-8977.

ZHAO, Junbo; ZHANG, Gexiang; DONG, Zhao Yang; WONG, Kit Po. Forecasting-aided imperfect false data injection attacks against power system nonlinear state estimation. **IEEE Transactions on Smart Grid**, IEEE, v. 7, n. 1, p. 6–8, 2015.

ZHU, Bonnie; JOSEPH, Anthony; SASTRY, Shankar. A taxonomy of cyber attacks on scada systems. In: IEEE. **2011 International conference on internet of things and 4th international conference on cyber, physical and social computing**. [S.l.], 2011. p. 380–388.

APÊNDICE A - CÓDIGO DE CONFIGURAÇÃO DE PORTAS - GATEWAY

```
{
"LogFileSizekB" : 50000,
"LogName" : "gateway.log",
"NumLogFiles": 1,
"LogLevel": "info",

"Plugins" :
[
{
"Name" : "ConsoleUI-1",
"Type" : "ConsoleUI",
"Library" : "ConsoleUI",
"ConfFilename" : "",
"ConfOverrides" : { }
}
],

"Ports" :
[
{
"Name" : "DNP3Outstation",
"Type" : "DNP3Outstation",
"Library" : "DNP3Port",
"ConfFilename" : "",
"ConfOverrides" :
{
"LOG_LEVEL":"ALL",
"IP" : "127.0.0.1",
"Port" : 20000,
"MasterAddr" : 1,
"OutstationAddr" : 10,
"EnableUnsol": true,
"TCPClientServer":"SERVER",
```

```
"EnableUnsol": true,
"UnsolClass1": true,
"UnsolClass2": true,
"UnsolClass3": true,
"Analogs":[{"Range" : {"Start" : 0, "Stop" : 1009}}],
"Binaries":[{"Range" : {"Start" : 0, "Stop" : 329}}],
"AnalogOutputs":[{"Range" : {"Start" : 0, "Stop" : 670}}],
"BinaryOutputs":[{"Range" : {"Start" : 0, "Stop" : 50}}]
}
},
{
"Name" : "DNP3Master",
"Type" : "DNP3Master",
"Library" : "DNP3Port",
"ConfFilename" : "",
"ConfOverrides" :
{
"LOG_LEVEL":"ALL",
"IP" : "10.0.2.2",
"Port" : 20001,
"TLS":true,
"CACertificate":"ca.pem",
"CertificateChain":"master.pem",
"PrivateKey":"master.key",
"MasterAddr" : 1,
"OutstationAddr" : 10,
"EnableUnsol": true,
"UnsolClass1": true,
"UnsolClass2": true,
"UnsolClass3": true,
"DoUnsolOnStartup" : true,
"StartupIntegrityClass0" : true,
"StartupIntegrityClass1" : true,
"StartupIntegrityClass2" : true,
```

```
"StartupIntegrityClass3" : true,
"IntegrityOnEventOverflowIIN" : true,
"TaskRetryPeriodms" : 30000,
"IntegrityScanRatems" : 0,
"EventClass1ScanRatems" : 0,
"EventClass2ScanRatems" : 0,
"EventClass3ScanRatems" : 0,
"TCPClientServer":"CLIENT",
"Analogs":[{"Range" : {"Start" : 0, "Stop" : 1009}}],
"Binaries":[{"Range" : {"Start" : 0, "Stop" : 329}}],
"AnalogOutputs":[{"Range" : {"Start" : 0, "Stop" : 670}}],
"BinaryOutputs":[{"Range" : {"Start" : 0, "Stop" : 50}}],
"StaticAnalogResponse": "Group30Var1",
"EventAnalogResponse": "Group32Var1",
"StaticBinaryResponse": "Group1Var2",
"EventBinaryResponse": "Group2Var2"
}
}
],

"Connectors" :
[
{
"Name" : "Connector1",
"ConfFilename" : "",
"ConfOverrides" :
{
"Connections" :
[
{
"Name" : "Connection1",
"Port1" : "DNP3Master",
"Port2" : "DNP3Outstation"
}
```

```
]
}
}
]
}
```