Segurança em Redes Industriais: Aplicação da técnica de autenticação HB-MP* em rede Modbus

Frederico Duarte Fagundes

Universidade Federal de Uberlândia
Faculdade de Engenharia Elétrica
Programa de Pós-Graduação em Engenharia Elétrica

Uberlândia
2022

Frederico Duarte Fagundes

Segurança em Redes Industriais: Aplicação da técnica de autenticação HB-MP* em rede Modbus

Tese de doutorado apresentada ao Programa de Pós-graduação da Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Doutor em Engenharia Elétrica.

Área de concentração: Engenharia Elétrica

Orientador: Ernane Antônio Alves Coelho
Coorientador: Márcio José da Cunha

Uberlândia
2022

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

F156 Fagundes, Frederico Duarte, 1988-
2022 Segurança em Redes Industriais [recurso eletrônico] : Aplicação da técnica de autenticação HB-MP* em rede Modbus / Frederico Duarte Fagundes. - 2022.

Orientador: Ernane Antônio Alves Coelho .
Coorientador: Márcio José da Cunha.
Tese (Doutorado) - Universidade Federal de Uberlândia, Pós-graduação em Engenharia Elétrica.
Modo de acesso: Internet.
Disponível em: http://doi.org/10.14393/ufu.te.2022.422
Inclui bibliografia.
Inclui ilustrações.

1. Engenharia elétrica. I. , Ernane Antônio Alves Coelho,1962-, (Orient.). II. Cunha, Márcio José da,1978-, (Coorient.). III. Universidade Federal de Uberlândia. Pós-graduação em Engenharia Elétrica. IV. Título.

CDU: 621.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Engenharia Elétrica
Av. João Naves de Ávila, 2121, Bloco 3N - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4707 - www.posgrad.feelt.ufu.br - copel@ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Engenharia Elétrica				
Defesa de:	Tese de Doutorado, 299, PPGEELT				
Data:	Vinte e nove de julho de dois mil e vinte e dois	Hora de início:	09:30	Hora de encerramento:	12:00
Matrícula do Discente:	11713EEL004				
Nome do Discente:	Frederico Duarte Fagundes				
Título do Trabalho:	Segurança em Redes Industriais: Aplicação da técnica de autenticação HB-MP* em rede Modbus				
Área de concentração:	Sistemas de energia elétrica				
Linha de pesquisa:	Eletrônica de potência				
Projeto de Pesquisa de vinculação:	Estudo sobre aplicações de Internet das Coisas (IoT)				

Reuniu-se por meio de videoconferência a Banca Examinadora designada pelo Colegiado do Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, assim composta: Professores Doutores: Renato Ferreira Fernandes Jr. - FEELT/UFU; Marcelo Barros de Almeida - FEELT/UFU; Andre Luis Dias - IFSP; Afonso Celso Turcato - IFSP, Prof. Márcio José da Cunha - FEELT/UFU, coorientador do candidato, e Ernane Antônio Alves Coelho - FEELT/UFU, orientador do candidato.

Conforme decisão do Colegiado do Programa de Pós-Graduação em Engenharia Elétrica, proferida na 352ª Reunião Ordinária, realizada em primeiro de julho de 2022, em caráter excepcional e em conformidade com a Resolução CONPEP Nº17/2022, foi determinada a participação concomitante do orientador e do coorientador na banca examinadora, sendo estabelecido que o orientador, Professor Ernane Antônio Alves Coelho, exercerá a presidência da sessão de defesa, mas não emitirá parecer acerca da avaliação do candidato.

Iniciando os trabalhos, o presidente da sessão, Dr. Ernane Antônio Alves Coelho, agradeceu a presença de todos e realizou os esclarecimentos quanto ao caráter excepcional de formação da banca examinadora e solicitou que o Prof. Márcio José da Cunha, coorientador do trabalho, fizesse a apresentação do candidato, do tema de tese, e da Comissão Examinadora. Na sequência, foi passada a palavra ao candidato para a apresentação da tese, conforme as normas do Programa.

Finalizada a apresentação do candidato, iniciou-se a fase de arguição, quando cada membro da banca examinadora arguiu o candidato quanto à tese apresentada. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, sem a presença do orientador, atribuiu o resultado final, considerando o candidato:

Aprovado.

Esta defesa faz parte dos requisitos necessários à obtenção do título de Doutor.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **ANDRE LUIS DIAS**, **Usuário Externo**, em 29/07/2022, às 14:14, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Afonso Celso Turcato**, **Usuário Externo**, em 29/07/2022, às 15:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Ernane Antonio Alves Coelho**, **Professor(a) do Magistério Superior**, em 29/07/2022, às 17:31, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Renato Ferreira Fernandes Junior**, **Professor(a) do Magistério Superior**, em 30/07/2022, às 18:59, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Marcio José da Cunha**, **Professor(a) do Magistério Superior**, em 01/08/2022, às 09:41, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Marcelo Barros de Almeida**, **Professor(a) do Magistério Superior**, em 01/08/2022, às 09:44, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3798330** e o código CRC **5F3E3D2A**.

Referência: Processo nº 23117.034815/2022-02 SEI nº 3798330

Agradecimentos

Os meus agradecimentos à todos aqueles que contribuíram para que esta tese se concretizasse.

Primeiramente agradeço à minha esposa Angelica, pelo amor, dedicação, ajuda e, principalmente, compreensão pelas ausências. Agradeço também aos meus pais, Geraldo e Edna, pelo carinho e incentivo ao longo de toda minha trajetória acadêmica, além da educação e exemplo com os quais fui criado.

Quero agradecer ao Prof. Dr. Márcio José da Cunha, por ter me convidado a pesquisar sobre esse tema. Também ao Prof. Dr. Márcio e ao Prof. Dr. Ernane Antônio Alves Coelho, agradeço por terem acreditado na minha capacidade e pela orientação no desenvolvimento da tese.

Deixo minha gratidão aos demais professores da Universidade Federal de Uberlândia, por terem contribuído com minha formação e compartilhado seus conhecimentos. Também agradeço aos técnicos administrativos, principalmente a secretaria do Programa de Pós-graduação da Faculdade de Engenharia Elétrica, pela dedicação, presteza e atenção com que sempre nos atenderam e atendem.

Agradeço aos companheiros de estudo, dentre os quais cito Luis Paulo Fagundes, Renato Nolli, Leandro Mattioli, Jefferson Beethoven e Artur Rios, com conversas e parcerias que sempre contribuíram para meu desenvolvimento acadêmico e para descontração nos momentos que mais precisei.

Por fim, mas não menos importante, agradeço aos meus amigos de trabalho no CEFET-MG, Campus Araxá. À Dra. Jacqueline Borges, agradeço as dicas e auxílio na construção do texto da tese. À todos professores e técnicos com os quais dividi desafios de gestão, agradeço a confiança e compreensão.

Resumo

Sistemas de automação industrial cada vez mais integrados e usando padrões e protocolos comuns às redes de computadores, fazem com que as redes industriais estejam cada vez mais sujeitas a ataques e invasões. O Modbus é um exemplo de protocolo industrial amplamente utilizado que, como outros, não possui ferramentas nativas para comunicação segura. A segurança para Modbus geralmente é obtida pelo uso de técnicas de segurança presentes nas camadas inferiores da pilha de protocolos. Esta tese apresenta as vulnerabilidades associadas ao Modbus, com foco no Modbus TCP, e oferece uma proposta de proteção através da autenticação dos Mestres e Escravos, na camada de aplicação. A técnica foi implementada por meio do protocolo HB-MP*, e focou na proteção contra a invasão de Mestres ou Escravos. A principal contribuição desta tese está no fornecimento de comunicações Modbus de maior segurança no nível da camada de aplicação, sendo possível como implementação única de segurança na rede, ou aumentando a segurança em conjunto com outras técnicas presentes em camadas inferiores. Para validar a proposta, foi configurada uma rede Modbus TCP com nós Mestre e Escravos realizando comunicação Modbus com autenticação HB-MP*, sujeita à diferentes cenários de invasão. Os resultados mostraram que foi possível detectar os nós invasores e evitar alguns ataques. A implementação apresentou um pequeno custo associado a atrasos de processamento e transmissão.

Palavras-chave: Segurança de redes. Sistemas de automação industrial. Autenticação. Modbus TCP.

Abstract

The growth in the integration and use of standards and protocols common to computer networks has seen industrial networks increasingly subject to attacks and invasions. Modbus is an example of a widely used industrial protocol, and like others, does not possess native tools for secure communication. Security for Modbus is achieved generally by use of security techniques on the lower layers of the communication stack. This study highlights vulnerabilities associated with Modbus, with its focus on Modbus TCP, while offering a protection proposal through use of Master and Slave authentication at the application layer level. The technique was implemented with the HB-MP* protocol, which focused on protection against invading Masters or Slaves. The main contribution of this paper is found through providing Modbus communications greater security at the application layer level, as a single security implementation or increasing security in conjunction with other lower layer security techniques. A Modbus TCP network was set up with Master and Slave nodes performing Modbus communication with HB-MP* authentication, subject to different intrusion scenarios. Results showed it was possible to detect invading nodes and prevent some attacks, whereas suffering only a small cost associated with processing and transmission delays.

Keywords: Network security. Networked control systems. Authentication. Modbus TCP.

Lista de ilustrações

Lista de tabelas

Lista de quadros

Lista de siglas

ADU Application Data Unit

ARP Address Resolution Protocol

ASCII American Standard Code for Information Interchange

CRC Cyclic Redundancy Check

CSMA/CA Carrier-Sense Multiple Access with Collision Avoidance

CSMA/CD Carrier-Sense Multiple Access with Collision Detection

FCS Frame Check Sequence

Gbit/s Gigabits por segundo

HMAC Hash-based Message Authentication Code

IEC International Electrotechnical Commission

IEEE Institute of Electrical and Electronics Engineers

IP Internet Protocol

LLC Link Layer Control

LRC Longitudinal Redundancy Check

MAC Media Access Control

OSCORE Object Security for COnstrained Restful Environments

PVLAN Private Virtual Local Area Network

RFID Radio-Frequency IDentification

RTE Real Time Ethernet

RTT Round Trip Time

RTU Remote Terminal Unit

SCTP Stream Control Transmission Protocol

SSI Self-Sovereign Identity

SSL Secure Sockets Layer

TCP Transport Control Protocol

TLS Transport Layer Security

UDP User Datagram Protocol

Sumário

Capítulo 1

Introdução

Um sistema de automação industrial é formado por dispositivos com funções de medição, atuação, monitoramento e controle. Tais sistemas contribuem para a otimização dos processos de fabricação e produção, o que exige a troca de dados de forma confiável entre os dispositivos. Os dados trafegados incluem, mas não se limitam a: variáveis de processo vindas dos transmissores e sensores, comandos enviados aos atuadores e mensagens de diagnóstico. Essa troca de dados será por vezes referida nesse trabalho pelo termo "comunicação", e uma das formas de realizá-la é através de tecnologias de comunicação industrial padronizadas e com ampla difusão no mercado, chamadas de redes industriais. As redes industriais podem ser definidas como o agrupamento de dispositivos (referidos em uma rede pelo termo "nós"), enviando e recebendo dados através de um conjunto de regras pré-definidas (protocolos e padrões), com o objetivo de monitorar, controlar e otimizar processos industriais (ALVES, 2010; ALBUQUERQUE; ALEXANDRIA, 2009; SEN, 2014; GALLOWAY; HANCKE, 2013).

As redes industriais geralmente se apresentam como soluções completas, definindo as funções dos dispositivos na rede, os protocolos utilizados e as topologias possíveis (ALBUQUERQUE; ALEXANDRIA, 2009; SEN, 2014). Entre essas tecnologias de comunicação industrial, aquelas com maior parcela de mercado em 2021 são Profinet, EtherNet/IP, EtherCAT, Profibus DP e Modbus TCP (NIDEBORN, 2021).

A maior parte das redes industriais foi desenvolvida a partir da década de 1970, com protocolos e padrões fechados ao desenvolvimento independente. Além disso, as redes industriais à época eram isoladas fisicamente das redes de computadores, internas ou externas à empresa. Uma importante característica que difere as redes industriais das redes de computadores é que as redes industriais possuem conexão direta com equipamentos físicos, como transmissores e atuadores e, consequentemente, com o processo industrial. Com isso, as redes industriais priorizam confiabilidade e eficiência na comunicação, visto que falhas podem causar acidentes com consequências pessoais e materiais (GALLOWAY; HANCKE, 2013; CHEMINOD; DURANTE; VALENZANO, 2013).

A partir da década de 1990, começou a aparecer a tendência de integração das re-

des industriais com as redes de computadores, inicialmente apenas com a rede interna da empresa e, posteriormente, com a internet. Para facilitar e incentivar essa integração, começaram a surgir opções de redes industriais que usam padrões abertos e comuns às redes de computadores, como o Ethernet, o Protocolo da Camada de Transporte e o Protocolo de Internet (*Transport Control Protocol* (TCP) e *Internet Protocol* (IP), respectivamente). Com isso, as preocupações com a segurança de rede aumentaram, e passaram a figurar entre as prioridades na escolha da tecnologia de comunicação (VOLKOVA et al., 2019; GALLOWAY; HANCKE, 2013; CHEMINOD; DURANTE; VALENZANO, 2013; FOVINO et al., 2009).

Nessa época, surge uma nova categoria de redes industriais, as redes Ethernet industriais. O padrão Ethernet define funções de modulação e acesso ao meio, sendo a forma de comunicação mais utilizada em redes de computadores atualmente. A integração das redes industriais com as redes de computadores e redes comerciais torna-se mais simples devido ao compartilhamento do padrão, possibilitando que dispositivos de campo sejam acessados por sistemas de planejamento e gerenciamento. Além disso, o padrão Ethernet traz outras vantagens, como facilidade de instalação, confiabilidade, eficiência e ampla difusão de mercado (BERGE, 2002; NIDEBORN, 2021; GALLOWAY; HANCKE, 2013).

As redes Ethernet industriais têm apresentado grande potencial de crescimento. Segundo a HMS (NIDEBORN, 2021), 2018 foi o primeiro ano em que as aplicações de Ethernet industrial ultrapassaram as redes industriais tradicionais (chamadas de redes de campo ou *fieldbuses*). Esse crescimento se manteve, enquanto a empresa HMS observou uma redução percentual dos nós em comunicação com *fieldbuses*.

Em uma rede industrial integrada, bases de dados, sistemas de Controle Supervisório e Aquisição de Dados e até Controladores Lógicos Programáveis estão conectados à rede de computadores local da empresa e, muitas vezes, também à internet. Com isso, essas redes tornam-se mais vulneráveis a ataques e invasões, internos e externos, comuns às redes de computadores. Além disso, ataques em rede específicos para aplicações industriais foram desenvolvidos, como, por exemplo, o notório caso do vírus Stuxnet, desenvolvido para explorar uma vulnerabilidade de um modelo específico de CLP. Desde 2010, técnicas de segurança passaram a ser um requisito para redes industriais tão importante quanto a confiabilidade e a eficiência, buscando prevenir falhas intencionais e invasões causadas por agentes maliciosos (VOLKOVA et al., 2019; ZHAO et al., 2020; GALLOWAY; HANCKE, 2013; CHEN, 2010; KUSHNER, 2013).

O cenário atual das redes industriais é ilustrado na Figura 1. O cenário sem integração seria apenas a parte inferior da Figura, abaixo da linha tracejada. O cenário com integração parcial está representado pela conexão com a parte superior esquerda da Figura e, com integração total pela conexão com a internet, na parte superior direita.

Na maioria das vezes, técnicas de segurança baseadas em redes de computadores não podem ser diretamente aplicadas em redes industriais. Um dos principais motivos é que,

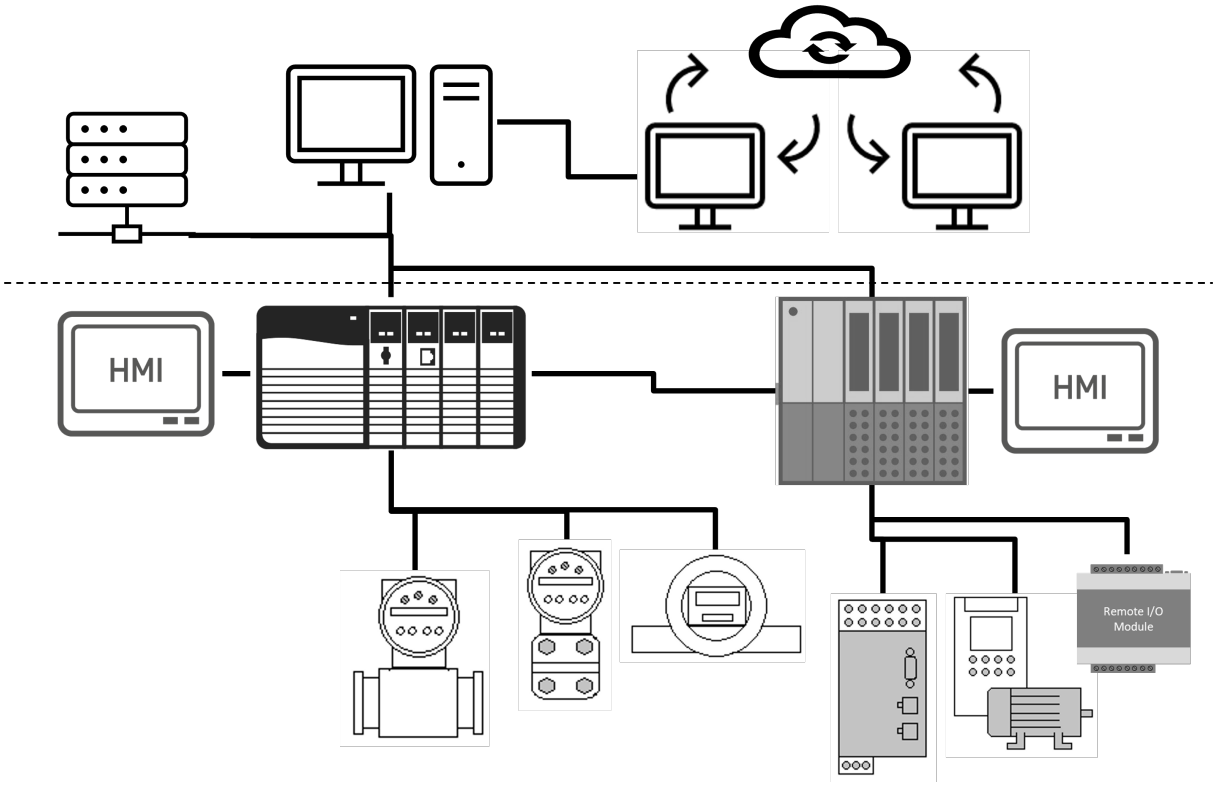

Figura 1 – Exemplo de rede industrial com ligação com a internet e acesso remoto
Fonte: Elaborada pelo autor

com a conexão direta com o processo físico, ataques e invasões em redes industriais podem causar não somente o vazamento de dados, que é uma das maiores preocupações em redes de computadores, mas também podem causar danos pessoais e materiais, um prejuízo consideravelmente mais grave (CHEMINOD et al., 2018).

Para a análise de segurança em uma rede, seja de computadores ou industrial, são considerados três aspectos principais: a disponibilidade, a integridade e a confidencialidade. Em sistemas de automação industrial, a disponibilidade e a integridade podem ser melhoradas fisicamente, através de redundância de equipamentos e infraestrutura, e redução do ruído, com a melhoria de cabeamentos e isolação física dos cabos de rede. Proteções contra negação de serviço (ataque mais conhecido pelo termo em inglês *Denial of Service*) e contra modificações nos dados trafegados, também contribuem para a disponibilidade e a integridade. Já a confidencialidade diz respeito à prevenção de acesso não autorizado, que pode ser feita com autenticação e criptografia (CHEMINOD et al., 2018; VOLKOVA et al., 2019; GALLOWAY; HANCKE, 2013).

Para suprir esses três aspectos de segurança, além das formas supracitadas, sistemas de automação industrial em rede podem usar proteções externas e internas. Um exemplo de proteção é o sistema de detecção de intrusão (ou *Intrusion Detection System* (IDS)), cujas funções incluem o monitoramento e a análise do tráfego na rede, o levantamento de

vulnerabilidades e a identificação de ataques e situações incomuns. Para proteção externa de uma rede, é comum o uso de credenciais para identificação no acesso e/ou *firewalls*. O *firewall* pode ser classificado como um IDS que, além da detecção, busca prevenir a intrusão através de um conjunto de regras, que vão desde o bloqueio de determinados endereços de rede até o reconhecimento de padrões por inteligência artificial (VOLKOVA et al., 2019; GALLOWAY; HANCKE, 2013; BHUYAN; BHATTACHARYYA; KALITA, 2014). São exemplos de *firewalls* industriais os desenvolvidos por Fovino et al. (2012) e Cheminod et al. (2018).

A proteção interna é feita para prevenir falhas locais causadas com ou sem intenção. Em casos intencionais, agentes maliciosos com acesso físico aos equipamentos de rede podem fazer uso de seu acesso privilegiado para atacar o sistema, por motivações que vão desde problemas pessoais com a empresa até ganho financeiro (VOLKOVA et al., 2019).

São inegáveis as vantagens que a integração atual das redes industriais traz para uma determinada empresa, como maior transparência e facilidade de acesso aos dados. Entretanto, os riscos se fazem presentes, tanto de ataques cibernéticos comuns em redes de computadores, quanto de ataques específicos para redes industriais. Com o uso de padrões e protocolos amplamente utilizados, esses ataques não requerem equipamento industrial específico e podem capturar, comprometer ou alterar os dados trafegados.

O Modbus TCP é um exemplo de rede industrial que mostra bem essa vulnerabilidade, pois usa o padrão Ethernet para as camadas física e de enlace, o TCP para a camada de transporte e o IP para a camada de rede, todos amplamente utilizados em redes de computadores. Além de ser uma das redes com maior parcela de mercado, o Modbus TCP não possui ferramentas nativas de segurança. Conforme assinala Fachkha (2019), ataques vindos da internet, cujo objetivo final são redes Modbus TCP, são comuns e devem ser levados em consideração.

Para suprir as deficiências de segurança da rede Modbus TCP e do próprio protocolo Modbus, propostas de segurança já foram levantadas, entre elas o Secure Modbus (FOVINO et al., 2009), o uBus (DUDAK et al., 2019), o Modbus com SSI (LORENZO; BENITO; ARRIZABALAGA, 2021), e, principalmente, o Modbus Security Protocol (MODBUS ORG., 2018). Essas soluções são apresentadas na seção 3.2. A maioria das propostas de segurança para Modbus são aplicadas nas camadas inferiores da pilha de protocolos, sendo que grande parte é aplicada na camada de transporte. Contudo, algumas vantagens na aplicação de técnicas de segurança na camada de aplicação foram mostradas por Narayanaswamy e Kumar (2019). Segundo os autores, a segurança na camada de aplicação pode ser aplicada de forma suplementar, melhorando a proteção geral da rede, ou de forma única, visto que nem todas aplicações em rede usam todas camadas de protocolos do modelo OSI.

Diante dessas considerações, o objetivo dessa tese é apresentar uma proposta de proteção visando tornar as redes Modbus TCP mais seguras, na camada de aplicação, com

a utilização de uma técnica de autenticação leve. Com a autenticação das entidades da camada de aplicação (Mestre e Escravo em redes Modbus), é possível mitigar vulnerabilidades em todas outras camadas de rede. Além disso, outras técnicas de segurança podem ser aplicadas nas camadas inferiores, incrementando a segurança da rede.

Esta tese está assim dividida: as definições necessárias para o entendimento das Redes Industriais estão apresentadas no Capítulo 2. As vulnerabilidades do Modbus TCP e os trabalhos relacionados são apresentadas no Capítulo 3. A técnica de segurança selecionada é apresentada no Capítulo 4. A metodologia e o desenvolvimento da aplicação são detalhados no Capítulo 5, com os resultados obtidos mostrados no Capítulo 6. O Capítulo 7 mostra a comparação da proposta desta tese com os trabalhos relacionados. Finalmente, o Capítulo 8 mostra a análise e as conclusões do trabalho.

Capítulo 2

Redes Industriais

A comunicação em ambiente industrial consiste na transmissão de um dado, seja ele um comando, uma variável ou uma notificação, entre dois ou mais pontos de interesse no processo de manufatura discreta e/ou contínua. Historicamente, essa comunicação iniciou-se com controles manuais e hidráulicos, passando para malhas de controle eletrônico e, posteriormente, para controladores digitais microprocessados com comunicação analógica e digital. Os controladores ainda são muito utilizados em conjunto com uma tecnologia que surgiu depois, que são as redes de campo (conhecidas pelo termo em inglês *Fieldbus Networks*). Essas redes utilizam a comunicação digital para trafegar dados, o que possibilita maior quantidade de informações trocadas entre os dispositivos, além da formação de redes eficientes, rápidas, dinâmicas e diagnosticáveis (SEN, 2014; GALLOWAY; HANCKE, 2013).

Com demandas de integração de informação de campo, as redes se desenvolveram para incorporar todos os níveis em uma indústria através da adoção do padrão Ethernet. O Ethernet atende aos requisitos de confiabilidade e eficiência da comunicação industrial, além de facilitar a integração, por também ser utilizado nas redes de computadores nos níveis de gerenciamento e planejamento da empresa. Com isso, novas tecnologias de comunicação industrial foram desenvolvidas, bem como houve a adaptação de tecnologias antigas para trabalhar com Ethernet. Atualmente as redes Ethernet industriais conseguem atender, inclusive, os requisitos de determinismo de alguns sistemas de automação. Com o crescimento do uso do padrão Ethernet, as redes industriais passaram a se dividir em categorias, sendo as mais conhecidas as redes de campo e as redes Ethernet industriais (GALLOWAY; HANCKE, 2013). Também há a categoria das redes sem fio, que embora seja menos utilizada, apresenta a maior taxa de crescimento no mercado (NIDEBORN, 2021).

Os dispositivos em comunicação em uma rede industrial podem ser chamados de nós. Os nós transmitem conjuntos de bits por meio de sinais elétricos em cabos coaxiais e pares trançados; pulsos luminosos em fibras óticas; e pulsos eletromagnéticos propagando no ar. Todos os requisitos para o bom funcionamento da rede precisam seguir um ou mais

protocolos padronizados, importantes para a interoperabilidade do sistema de automação industrial (SEN, 2014; BERGE, 2002).

Os protocolos são conjuntos de regras, como a codificação e a decodificação dos grupos de bits, para que tanto o transmissor quanto o receptor possam gerar e interpretar os dados de forma adequada. Uma rede pode utilizar diversos protocolos, divididos em camadas com funções definidas, atuando em conjunto para que a comunicação seja eficiente e atenda aos requisitos da rede. O arranjo em camadas permite que protocolos sejam desenvolvidos e aplicados de forma paralela e independente. Na comunicação entre computadores, por exemplo, existem diferentes protocolos para a camada de aplicação, que separam aplicações de e-mail, sistemas de pesquisa online e acesso remoto de arquivos em servidores. Esses protocolos de aplicação são utilizados para fins diferentes (daí o termo "aplicação"), mas compartilham protocolos nas outras camadas (BOYER, 2004; SEN, 2014; KUROSE; ROSS, 2014).

Segundo Kurose e Ross (2014), as camadas das redes de computadores, além da camada de aplicação, são:

- Transporte: a camada de transporte define as regras para estabelecer a comunicação lógica entre dois nós, trazendo confiabilidade e integridade na comunicação. São exemplos de protocolos da camada de transporte o TCP e o *User Datagram Protocol* (UDP), sendo o primeiro orientado à conexão fim a fim e o segundo, mais rápido e menos confiável, um protocolo que permite a troca de dados entre dispositivos sem que seja estabelecida uma conexão lógica entre eles.

- Rede: a camada de rede é responsável pelo roteamento de dados na rede, com protocolos necessários para que os dados cheguem do transmissor ao receptor. O endereço de rede é utilizado para esse fim e o protocolo mais conhecido da camada de rede é o IP.

- Enlace: a camada de enlace (mais conhecida pelo termo em inglês *link*) define regras de acesso ao meio e controle de fluxo de dados. Por ter suas funções definidas no padrão Ethernet, a camada de enlace será detalhada na Seção 2.1.2.

- Física: a camada física inclui protocolos de modulação dos bits e definições de transmissão, sendo a responsável por transformar os bits em sinais. Também é uma camada com funções definidas pelo padrão Ethernet, e por isso será detalhada na Seção 2.1.1.

A forma como as camadas supracitadas estão organizadas e como os dados são passados de uma camada para outra estão mostrados na Figura 2. As camadas se organizam de forma que os protocolos de uma camada passem os dados para o protocolo da camada imediatamente inferior, começando pela camada de aplicação até chegar na camada física. Na camada física, os dados são transformados em sinais a serem propagados até o próximo

ponto da rede. No destino, os sinais recebidos passam pelo caminho inverso, iniciado na camada física, que passa os dados para o protocolo da camada de enlace, até que os dados cheguem à camada de aplicação e sejam tratados adequadamente, conforme a aplicação de rede.

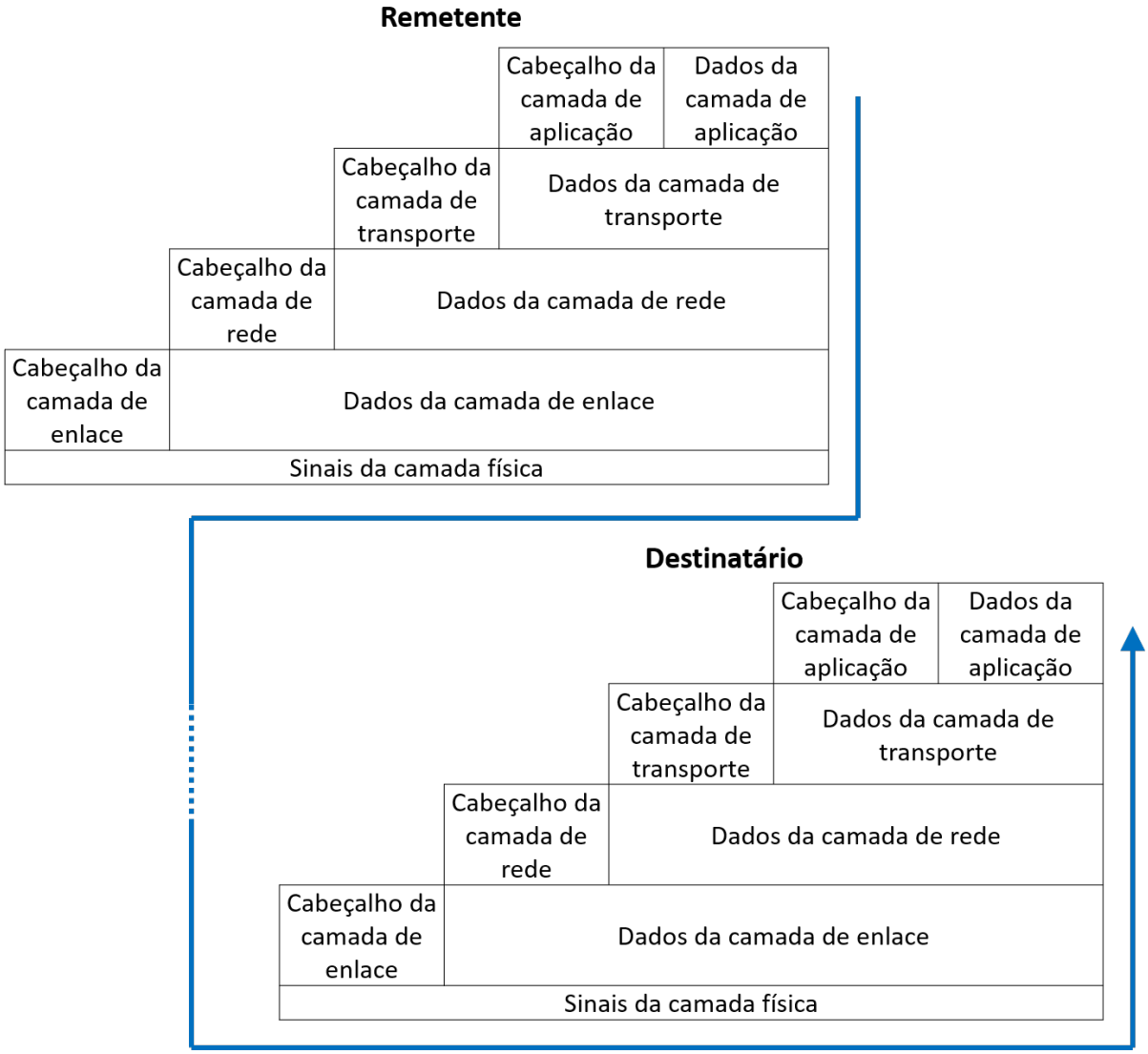

Figura 2 – Comunicação e organização em camadas
Fonte: Elaborada pelo autor

Nas redes, o arranjo físico dos dispositivos e como se conectam definem a topologia. Podem-se citar as topologias de rede barramento, anel, estrela e árvore. Na topologia em barramento, os nós da rede são conectados entre si por um único cabeamento, ou por um cabeamento e suas ramificações, ou ainda com os dispositivos em encadeamento (essa variação é mais conhecida pelo termo em inglês *daisy chain*). Devido ao cabeamento compartilhado, os sinais trafegados são comuns para todos os nós, portanto, é necessário tanto o endereçamento para identificar os nós transmissores e receptores, quanto o con-

trole de acesso ao meio físico [1]. Essa topologia tem um ponto de falha crítico que é o próprio barramento, por isso é comum o uso de cabeamento redundante (ALBUQUERQUE; ALEXANDRIA, 2009; MORAES; CASTRUCCI, 2001). Em redes Ethernet, o uso de concentradores (mais conhecidos pelo termo em inglês *hubs*) é considerado uma forma de barramento lógico, pois esse dispositivo de rede difunde todos os pacotes de dados que nele chegam (REYNDERS; MACKAY; WRIGHT, 2004).

A topologia em anel se assemelha a uma topologia em encadeamento, com a característica que o último nó da rede se conecta ao primeiro, formando um círculo. Os dados são transmitidos de um dispositivo a outro em apenas um sentido ou, para incrementar a confiabilidade através da redundância, em ambos os sentidos (SEN, 2014).

Na topologia em estrela, um nó assume a função de receptor e distribuidor de todos os dados. Todos os nós se comunicam exclusivamente com ele, e o dado é redirecionado por ele, quando necessário. Uma das vantagens dessa topologia é a interoperabilidade, pois nós que seguem protocolos diferentes de comunicação podem entrar na rede, desde que o nó central possua os requisitos de comunicação com aquele protocolo. A desvantagem da topologia é que falhas no nó central prejudicam toda a rede (ALBUQUERQUE; ALEXANDRIA, 2009). Um exemplo da topologia em estrela são as redes tradicionais de comunicação industrial, com controladores centrais e cartões de entradas e saídas analógicas e digitais.

Em redes Ethernet, a topologia em estrela se apresenta com o uso de comutadores (mais conhecidos pelo termo em inglês *switches*). Diferentemente do concentrador, o comutador interpreta os dados recebidos e os direciona somente ao destino, sendo considerado um nó central e, dessa forma, a rede se caracteriza como estrela (SEN, 2014; KUROSE; ROSS, 2014).

A Figura 3 mostra uma rede dividida em três sub-redes. A Sub-rede 1 é um exemplo de topologia em estrela, na qual o Comutador 1 exerce o papel de nó central. O Comutador 1 também faz parte da Sub-rede 2, sendo o primeiro nó de uma rede em barramento (encadeamento). O Comutador 2 tem uma porta de comunicação como parte da Sub-rede 1 e outras duas portas são parte da Sub-rede 3, com topologia em anel.

A topologia em árvore é uma topologia mista, onde vários barramentos são conectados (como galhos de uma árvore), sendo que geralmente um deles é um barramento central (associado a um tronco). Essa topologia apresenta boa confiabilidade e é comum em sistemas de automação industrial que foram expandidos de sua configuração original (ALBUQUERQUE; ALEXANDRIA, 2009; MORAES; CASTRUCCI, 2001).

A próxima seção detalha o padrão Ethernet, com ênfase na sua aplicação industrial. Já a seção 2.2 detalha o protocolo Modbus, destacando sua aplicação com Ethernet, chamada Modbus TCP, objeto de estudo desta tese.

[1] É importante ressaltar que o endereçamento e controle de acesso também podem ser necessários em outras topologias.

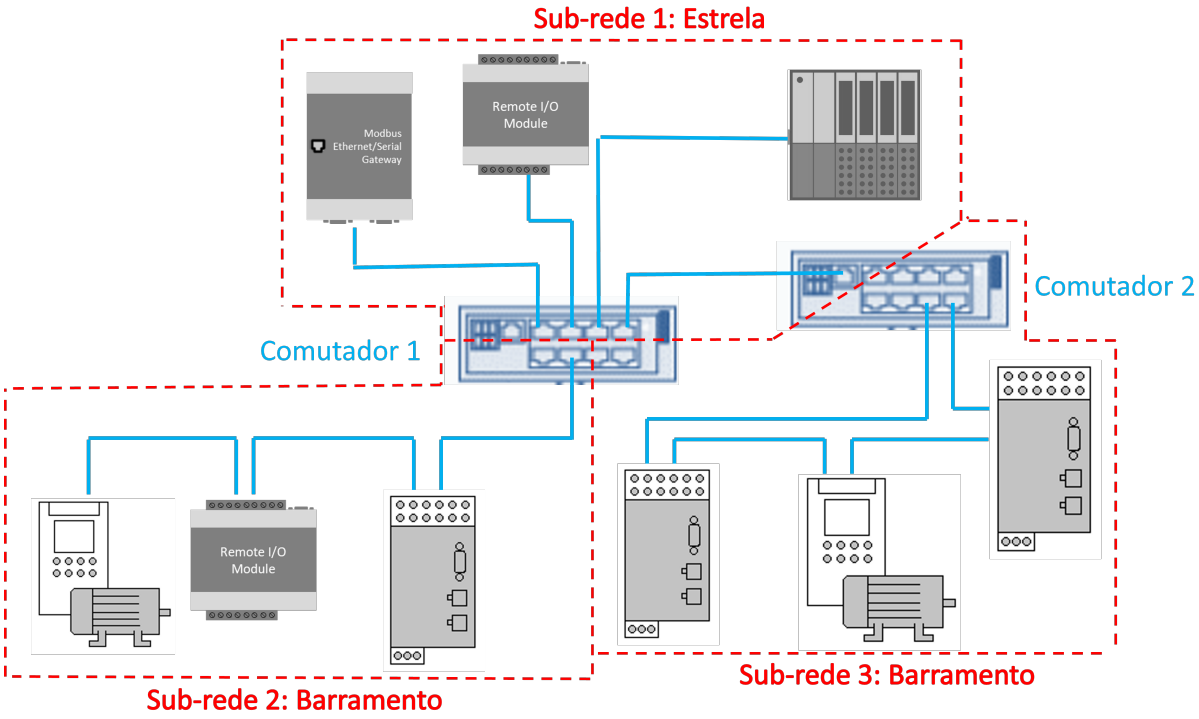

Figura 3 – Exemplo de uma rede com múltiplas topologias
Fonte: Elaborada pelo autor

2.1 Ethernet

O padrão Ethernet é chamado de pilha de protocolos por definir funções tanto da camada física quanto da de enlace. Essas funções estão especificadas na coleção de normas IEEE 802.3. O padrão está em constante desenvolvimento e as versões atuais permitem comunicação tanto por meio compartilhado (comunicação *half-duplex*) quanto por condutores exclusivos para transmissão e recepção (comunicação *full-duplex*). Além disso, a velocidade de transmissão pode chegar às centenas de Gigabits por segundo (Gbit/s) (REYNDERS; MACKAY; WRIGHT, 2004; IEEE, 2018). As duas camadas do padrão serão detalhadas nas próximas subseções.

2.1.1 Meios de transmissão no padrão Ethernet

Os meios de transmissão utilizados atualmente no padrão Ethernet são cabos coaxiais, de fibra ótica ou de pares trançados. O cabo de pares trançados é um cabo que pode conter um ou mais pares de condutores isolados e trançados. É bem comum em redes locais com base no padrão Ethernet o uso de cabos com quatro pares. Cada par tem uma função específica, conforme as necessidades de comunicação da rede. Os condutores utilizados possuem seção transversal entre 0,13 mm^2 e 0,33 mm^2, o que garante maleabilidade e reduz o custo do cabeamento, mas impede que o cabeamento seja utilizado com desempenho

satisfatório em distâncias maiores que 100 metros. Os cabos são agrupados em categorias de acordo com sua imunidade a ruído, seção transversal dos condutores e presença de malha de blindagem (IEEE, 2018; TIA, 2018).

O cabo coaxial é composto de um condutor envolvido em um material que deve isolar eletricamente o condutor de uma malha, que atua como blindagem e como condutor de referência. Além da blindagem, há uma isolação externa que reveste todo o cabo. Devido ao material dielétrico que compõe o isolamento interno, e à seção transversal do condutor, que pode ser de até 4 mm^2, esse tipo de cabo possui imunidade a ruído maior que o cabo de pares trançados e perdas de sinal bem menores. Por isso, são mais indicados que os pares trançados para a comunicação a distâncias maiores que 100 metros (IEEE, 2018).

A fibra ótica permite maiores distâncias e velocidades que ambos os cabeamentos baseados em condutores elétricos. Ela utiliza sinais de luz guiados através de um meio (chamado de núcleo) com baixa refração, projetado para não interferir ou absorver o sinal de luz, apenas refleti-lo. A energia não sai nem entra no núcleo, o que traz imunidade a ruídos eletromagnéticos. Esses dois fatores permitem que os cabos de fibra propaguem sinais por quilômetros sem perda. Além disso, a velocidade de propagação dos sinais é bem maior, permitindo a transmissão de sinais a até 100 Gbit/s (REYNDERS; MACKAY; WRIGHT, 2004; IEEE, 2018).

A comunicação sem fio é também bastante utilizada em conjunto com os meios supracitados. O conjunto de normas IEEE 802.11 define as especificações de redes sem fio locais, com alcance de até algumas centenas de metros, mas usualmente limitadas a dezenas de metros. A tecnologia é conhecida pelo termo WiFi e tem, em comum com as redes Ethernet 802.3, o protocolo *Media Access Control* (MAC) na camada de enlace. Os sinais correspondem a pulsos eletromagnéticos sem guia físico, que são transmitidos e recebidos por antenas. Por utilizarem um meio compartilhado (a atmosfera), colisões podem ser mais comuns que nas formas de transmissão cabeadas. Para lidar com isso, na camada de enlace é utilizada a técnica *Carrier-Sense Multiple Access with Collision Avoidance* (CSMA/CA), que difere da CSMA *with Collision Detection* (CSMA/CD) por tentar prever colisões, em vez de apenas detectá-las e reduzir o tempo de recuperação. A comunicação entre os padrões IEEE 802.3 e IEEE 802.11 é feita por meio de modems (moduladores e demoduladores de sinais), que, em aplicações residenciais comuns, recebem um sinal de fibra ótica externo, fornecido por um provedor de rede, e distribuem o sinal localmente por pares trançados e WiFi (KUROSE; ROSS, 2014; SEN, 2014). A Figura 4 ilustra esses meios de transmissão.

Esses meios de transmissão definem a camada física do padrão Ethernet. A camada física, no nó remetente, recebe o pacote de dados da camada de enlace para convertê-los em sinais. No nó destinatário, a camada física é a primeira a receber os sinais e é responsável por passá-los como pacote de dados para a camada de enlace (ver Figura 2).

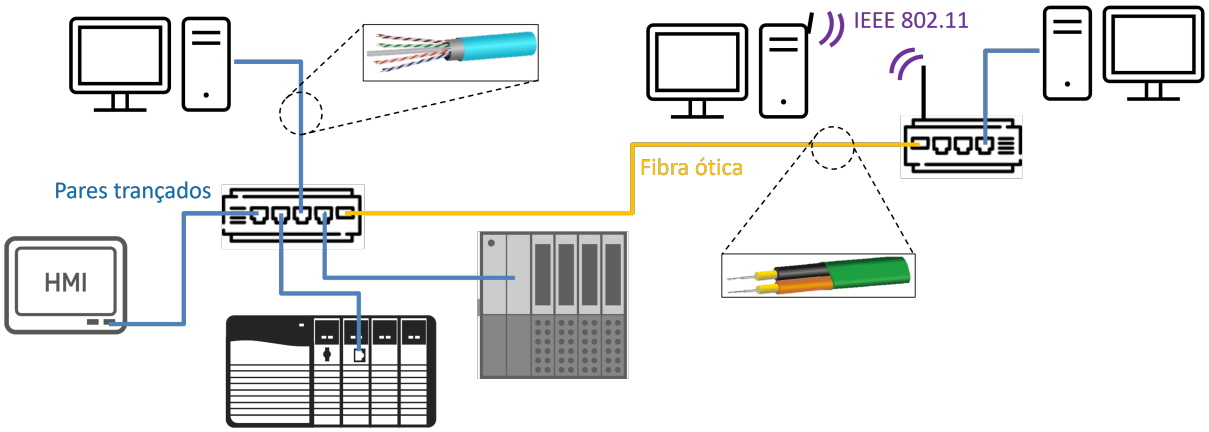

Figura 4 – Exemplo dos meios de transmissão em uma rede Ethernet
Fonte: Elaborada pelo autor

2.1.2 Camada de enlace no padrão Ethernet

As funções da camada de enlace em redes Ethernet são o controle de acesso ao meio, através do protocolo MAC, e o controle de enlace, através do *Link Layer Control* (LLC). O protocolo de controle de acesso ao meio é responsável por encapsular os dados de forma adequada ao meio físico, adicionar bits de verificação de erros por meio da técnica de verificação de pacote de dados *Frame Check Sequence* (FCS) e encaminhar os dados para a camada física. Nessa última etapa, também é função do protocolo aplicar técnicas de controle de acesso ao meio. O protocolo de controle de enlace LLC tem funções de controle de vazão de dados, retransmissões automáticas e multiplexação. As duas primeiras são pouco utilizadas, mas a multiplexação é importante para definir qual protocolo está sendo utilizado nas camadas superiores à de enlace (diferencia dados Modbus de outros dados TCP/IP, por exemplo) (KUROSE; ROSS, 2014; REYNDERS; MACKAY; WRIGHT, 2004).

Para estabelecer os enlaces Ethernet entre dispositivos com meio de transmissão par trançado, podem ser utilizados Concentradores e Comutadores. Todos dispositivos em redes IEEE 802 são identificados de acordo com o endereço MAC[2], que é único para cada interface de rede. O Comutador direciona os pacotes de dados da camada de enlace[3] recebidos através de uma tabela que relaciona os endereços MAC com suas portas físicas. Sempre que o Comutador recebe um pacote de dados por uma porta, esta é associada ao endereço MAC remetente. Caso um pacote de dados seja posteriormente direcionado a esse MAC, o Comutador verifica na tabela a porta relacionada e encaminha o pacote

[2] O termo técnico correto é endereço da camada de enlace, mas sua terminologia mais comum é "endereço MAC", por isso será referido assim nesta tese.

[3] Os pacotes de dados das camadas possuem nomes específicos para diferenciação. Na camada de enlace, os termos utilizados são, em português, “quadro”, e em inglês, “*frame*”. Nesta tese, essa diferenciação não foi feita, optou-se pelo uso do termo geral “pacote de dados”, seguido da identificação da camada, quando necessário.

de dados para ela. Se o Comutador não possuir uma porta associada ao endereço MAC destinatário, o pacote de dados é difundido para todas as portas (KUROSE; ROSS, 2014).

No nó remetente, a camada de enlace recebe o pacote de dados da camada superior e adiciona seu cabeçalho, com os dados de controle e endereçamento supracitados. No nó destinatário, o pacote de dados é recebido da camada física e interpretado pelos protocolos MAC e LLC. A porção de dados é então extraída e encaminhada para a camada superior, conforme o cabeçalho de enlace (ver Figura 2).

Na próxima subseção, são detalhados alguns aspectos do padrão Ethernet, quando utilizado em aplicações industriais.

2.1.3 Ethernet industrial

As chamadas redes Ethernet industriais, também conhecidas pela sigla em inglês RTE (*Real Time Ethernet*), são as que usam as especificações Ethernet em conjunto com outros protocolos nas camadas superiores: rede, transporte e aplicação, visando atender especificações da indústria (REYNDERS; MACKAY; WRIGHT, 2004). É possível que diferentes aplicações de rede compartilhem a mesma infraestrutura simultaneamente, mas apesar de usarem o mesmo tipo de cabeamento e controle de acesso ao meio, devido às diferenças nos protocolos das camadas superiores, as redes só são interoperáveis com uso de conversores de protocolo (*gateways*) específicos ou caso permitam encapsulamento dos pacotes de dados.

Uma importante diferença das aplicações industriais de Ethernet se dá na camada física. Devido a características nocivas dos ambientes industriais, como maior interferência eletromagnética, calor e umidade excessivos e presença de óleos e gases, o cabeamento Ethernet deve ser especificado com cuidado. Existem opções de cabos que oferecem maior imunidade a ruído através da presença da malha de blindagem por par e por cabo, cabos com isolação externa de material antitérmico, cabos com isolação reforçada, entre outros. Além disso, existem opções de conectores reforçados e mais confiáveis, para garantir a conexão física diante de dilatações, vibrações, esforços mecânicos e contato com fluidos (REYNDERS; MACKAY; WRIGHT, 2004; BELDEN, 2022). A Figura 5 ilustra a diferença entre conectores RJ45 tradicionais, à direita, e conectores M12 reforçados, à esquerda.

2.2 Modbus

Modbus é um protocolo industrial da camada de aplicação e, por especificar funções apenas de aplicação, pode e deve ser aplicado com outros protocolos das camadas inferiores. Os demais protocolos devem atender aos objetivos da instalação de rede, conforme características e restrições da instalação. É comum que o termo Modbus seja acompanhado de outro termo, que geralmente identifica quais os demais protocolos utilizados. O

Figura 5 – Conectores Lumberg M12 e RJ45
Fonte: Adaptado de Belden (2022)

Modbus é comumente utilizado com as tecnologias de comunicação serial RS-232, RS-422 e RS-485, que definem especificações físicas e de modulação de sinal. Nessas aplicações seriais, o Modbus pode ser identificado como Modbus RTU e Modbus ASCII, que são duas formas para a codificação dos dados da camada de aplicação. Em aplicações Modbus RTU, caracteres hexadecimais são utilizados para representar cada dado de uma mensagem Modbus. Cada caractere usa 4 bits. Já em Modbus ASCII, a codificação ASCII é utilizada, logo cada caractere é representado por 7 bits, mais o bit de paridade. O Modbus ASCII aumenta consideravelmente a carga de dados trafegados e, por isso, é menos utilizado (SEN, 2014; REYNDERS; MACKAY; WRIGHT, 2004).

A forma de aplicação do Modbus com maior crescimento atualmente é o Modbus TCP (NIDEBORN, 2021), que consiste no protocolo Modbus aplicado com os protocolos TCP e IP para as camadas de transporte e rede, respectivamente, e o padrão Ethernet nas camadas de enlace e física. O Modbus TCP será mais detalhado na subseção 2.2.1. O Modbus também possui variações na própria camada de aplicação, que são o Modbus Plus e o Modbus II, mas ambas têm pouco uso, tanto em instalações industriais antigas quanto em novas (SEN, 2014; REYNDERS; MACKAY; WRIGHT, 2004).

Os dispositivos que se comunicam pelo protocolo Modbus vão desde atuadores, sensores e unidades de controle remotas, até sistemas de supervisão, controladores e centros de carga. O Modbus é um protocolo Mestre/Escravo, no qual um dispositivo Escravo disponibiliza dados e realiza tarefas como resultado de requisições de um dispositivo hierarquicamente superior, o Mestre. As mensagens no protocolo Modbus originadas do Mestre são Requisições Modbus; enquanto as originadas do Escravo, conforme Requisição recebida, são Respostas Modbus. As ações possíveis para os Escravos são: executar o

comando solicitado; enviar os dados requeridos; e informar que a Requisição não pôde ser executada. A comunicação sempre é iniciada pelo Mestre e se dá por transmissão individual para um Escravo, que sempre gera uma resposta; ou por transmissão para todos Escravos por difusão. A Requisição em difusão é obrigatoriamente uma escrita e não gera uma Resposta (SEN, 2014; REYNDERS; MACKAY; WRIGHT, 2004).

O pacote de dados Modbus RTU é chamado de ADU (*Application Data Unit*) e é dividido em quatro campos, mostrados na Tabela 1. O primeiro campo, **Endereço**, é responsável por identificar o dispositivo Escravo. Os escravos podem ser identificados de 1 a 247, sendo o endereço 0 (zero) usado para mensagens em difusão. O segundo campo, **Código da função**, identifica qual função está sendo requisitada pelo Mestre. As funções são identificadas de 1 a 255, sendo as mais comuns na comunicação de dados cíclicos (REYNDERS; MACKAY; WRIGHT, 2004):

- 01: leitura de bobinas (saídas digitais);
- 02: leitura de entradas digitais;
- 03: leitura de registradores de entrada (entradas analógicas);
- 05: escrita em bobina;
- 06: escrita em registrador de saída (saída analógica);
- 15: escrita em múltiplas bobinas;
- 16: escrita em múltiplos registradores.

O campo **Dados** varia de acordo com a função. São exemplos de dados o endereço de uma saída e os dados a serem escritos na mesma; os endereços das entradas solicitadas em uma requisição de leitura; e parâmetros em um comando de configuração. Com o último campo do pacote de dados, **Verificação de erros**, é possível que os dispositivos verifiquem a presença de erros de transmissão através de técnicas como a Verificação Redundante Cíclica e a Verificação Redundante Longitudinal (*Cyclic Redundancy Check* (CRC) e *Longitudinal Redundancy Check* (LRC), respectivamente) (SEN, 2014; REYNDERS; MACKAY; WRIGHT, 2004).

Tabela 1 – Pacote de dados Modbus RTU

Endereço	**Código da função**	**Dados**	**Verificação de erros**
1 byte	1 byte	n bytes	2 bytes

Fonte: Elaborada pelo autor

2.2.1 Modbus TCP

Em redes Modbus TCP, o protocolo Modbus é aplicado com o padrão Ethernet e a pilha de protocolos TCP/IP. O TCP tem a função de estabelecer a conexão lógica entre os processos e controlar a vazão de dados. Já o IP define o roteamento dos dados na rede, com base no endereçamento de rede. O pacote de dados padrão TCP/IP tem um cabeçalho específico e sua porção de dados é preenchida pelos dados de aplicação do protocolo Modbus, com exceção do campo de verificação de erros (ver Tabela 1). Esse campo é suprimido, visto que o protocolo TCP já inclui verificação de erros (SEN, 2014; KUROSE; ROSS, 2014). A Figura 6 mostra a pilha de camadas usadas pelo Modbus e o pacote de dados Modbus incluído no pacote de dados TCP, que faz parte do pacote de dados trafegado pelo padrão Ethernet. Para simplificação, o pacote de dados final pode ser chamado de pacote de dados Modbus TCP.

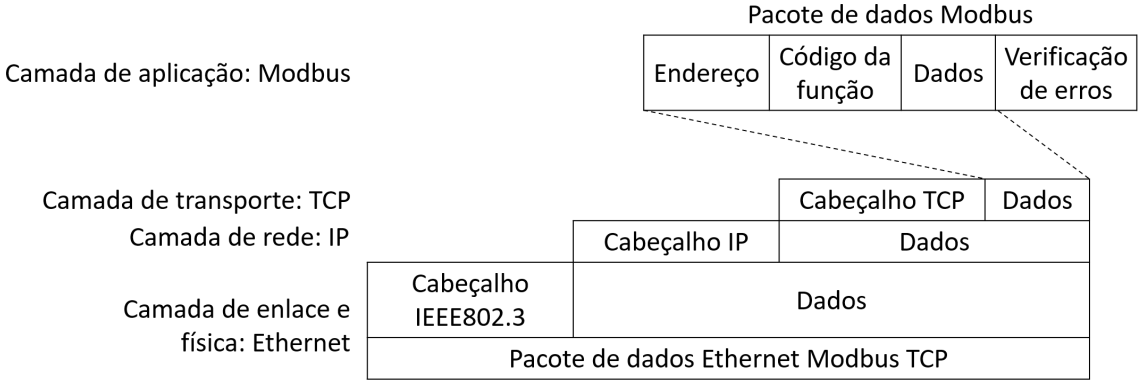

Figura 6 – Pilha de camadas Modbus TCP e composição do pacote de dados Modbus TCP

Fonte: Adaptado de Sen (2014)

Para exemplificar o arranjo em camadas do pacote de dados Modbus TCP, é mostrado na Figura 7 uma captura em uma rede real de uma Requisição de escrita em registrador. Na Figura os dados são representados por caracteres hexadecimais. Cada caractere hexadecimal representa um *nibble* (conjunto de quatro bits) e dois caracteres representam um byte. As diferentes cores representam os cabeçalhos das diferentes camadas, sendo vermelho, a camada de enlace; verde, a camada de rede; e azul, a camada de transporte. A cor marrom representa os dados Modbus da camada de aplicação.

Os dois primeiros campos destacados na Figura 7 são os endereços MAC do remetente e do destinatário. O próximo campo, *Ethertype*, identifica o tipo de pacote de dados da camada superior (rede). No caso, os bytes $08\,00_{hex}$ identificam que, para a camada de rede, está sendo utilizado o protocolo IP. O próximo campo destacado já se refere à camada de rede, no qual o primeiro *nibble* identifica que o protocolo IP utilizado é a versão 4. O segundo *nibble* da camada de rede identifica que o cabeçalho da camada de rede possui, ao

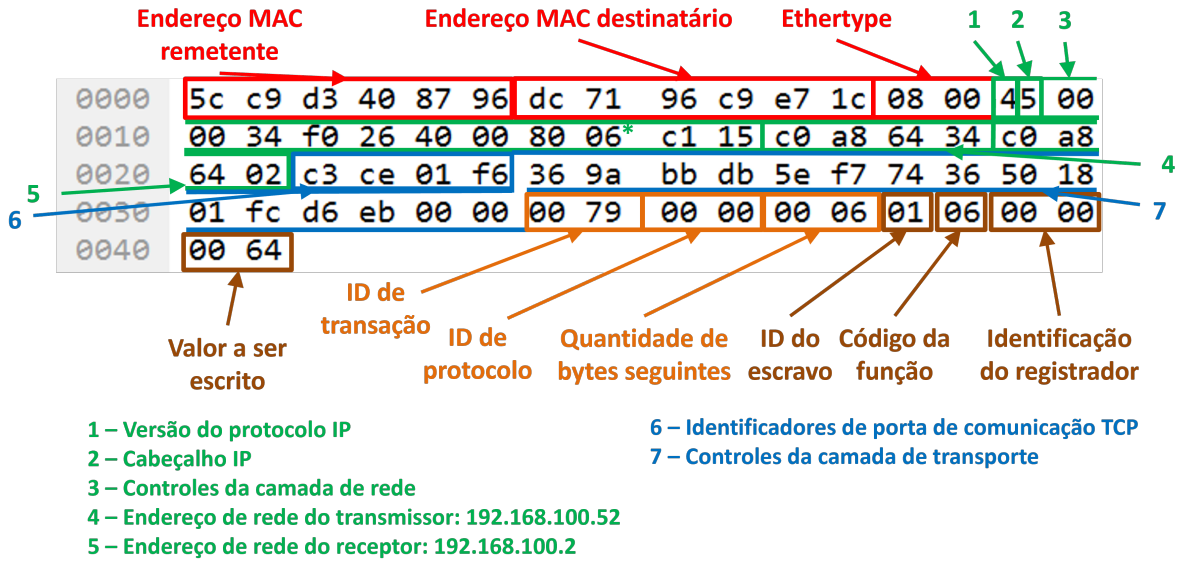

Figura 7 – Pacote de dados Modbus TCP capturado
Fonte: Elaborada pelo autor

todo, 20 bytes. O destaque de número 3 compreende vários bytes de controle necessários à camada de rede, incluindo o identificador para qual protocolo está sendo usado na camada de transporte (*nibble* 6, destacado com um asterisco). Os destaques 4 e 5 são os endereços de rede do remetente (Cliente) e do destinatário (Servidor), respectivamente.

Os campos destacados em azul representam dados da camada de transporte, sendo os quatro primeiros bytes (destaque 6) identificadores da porta de comunicação TCP. O destaque 7 compreende bytes de controle necessários à camada de transporte. Após esses, em marrom claro, são destacados o identificador de transação (usado para diferenciar cada Requisição, sendo copiado na respectiva Resposta), o identificador do protocolo (os bytes $00\,00_{hex}$) identificam o protocolo Modbus), e o contador de bytes no restante do pacote (nesse caso, 6 bytes).

Já nos dados Modbus, foram destacados o identificador do escravo e o identificador da função de escrita em registrador de saída (06). O registrador a ser escrito possui endereço $00\,00_{hex}$ e o valor a ser escrito é $00\,64_{hex}$ (equivale a 100 no sistema de numeração decimal).

Em aplicações Modbus TCP os termos Cliente e Servidor são comumente usados de forma análoga aos termos Mestre e Escravo. Entretanto, os termos Cliente e Servidor se referem aos protocolos da camada de transporte e de rede, enquanto os termos Mestre e Escravo são específicos para o protocolo da camada de aplicação Modbus, diferenciando o gerador das Requisições do executante. Com isso, o endereçamento de um nó em Modbus TCP é tanto pelo endereço de rede do Cliente ou do Servidor (comumente chamado de endereço IP) quanto pelo endereço do Mestre ou do Escravo (que também pode ser chamado de identificador de Unidade).

Nessa tese, por se tratar de uma proposta para a camada de aplicação, os termos Mes-

tre e Escravo serão utilizados tanto para identificar os programas que executam as funções da camada de aplicação Modbus, como para identificar dispositivos físicos hospedeiros. Os termos Cliente e Servidor também serão utilizados, mas apenas quando a diferença for clara entre as funções de cada camada. Uma situação que exemplifica essa diferença entre as formas de endereçamento e os termos é ilustrada na Figura 8, na qual o endereço IP do dispositivo conversor entre redes Modbus TCP e Modbus serial (RTU ou ASCII) é compartilhado por mais de um endereço da camada de aplicação (MOXA, 2021).

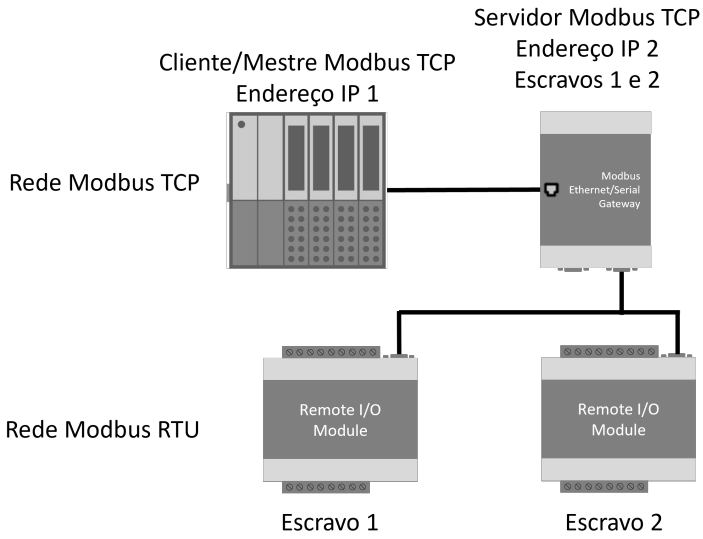

Figura 8 – Exemplo da diferença entre endereço de rede e identificador de escravo em uma rede Modbus TCP

Fonte: Elaborada pelo autor

O protocolo Modbus não possui em sua concepção original ferramentas de segurança. Vale lembrar que, na época de seu desenvolvimento, na década de 1970, pela AEG-Modicon, pouco se pesquisava sobre segurança em rede industrial. Isso é verdade tanto para o protocolo Modbus quanto para vários outros protocolos e padrões industriais, mesmo os mais recentes (CHEMINOD; DURANTE; VALENZANO, 2013; VOLKOVA et al., 2019). Considerando que as questões de segurança passaram a ter alta relevância para sistemas de automação industrial, o próximo Capítulo (3) traz os detalhes envolvidos na segurança de redes, com foco nas redes Modbus TCP.

Capítulo 3

Segurança em redes Modbus TCP

Sistemas de automação industrial dependem da transmissão de dados. As redes industriais, desenvolvidas a partir da década de 1970, possuíam uma característica de isolamento físico, ou seja, estavam confinadas ao espaço da empresa. Além disso, por estarem conectadas diretamente a um equipamento físico e serem usadas para monitorar ações e condições físicas, suas prioridades são diferentes das redes comerciais de computadores, sendo elas confiabilidade e eficiência. Entretanto, o isolamento das redes de campo deixou de ser verdade com a integração dos sistemas atuais. Diante disso, a segurança também passou a ser uma preocupação nos sistemas de automação industrial. Falhas em redes podem ocorrer acidentalmente, e.g., devido a ruídos na comunicação e falhas de envio, ou intencionalmente, ocasionados por um atacante malicioso. As técnicas de segurança são para prevenir essas falhas intencionais ou invasões causadas por agentes maliciosos (VOLKOVA et al., 2019; GALLOWAY; HANCKE, 2013; CHEMINOD et al., 2018).

Na indústria, as restrições para técnicas de segurança são mais rígidas que nas redes comerciais, pois não há somente a preocupação com a segurança dos dados. Isso porque falhas causadas por invasores podem trazer danos pessoais e materiais, devido à conexão direta com o processo. Os primeiros casos de vulnerabilidades de segurança industrial foram identificados por via indireta, através de partes da rede industrial conectadas à internet, como historiadores, sistemas de supervisão e controle e bancos de dados. Ataques a esses dispositivos podem causar danos à rede de campo de forma indireta, por invasão aos dados confidenciais e negação de serviço, por exemplo. Entretanto, agentes maliciosos específicos para redes e equipamentos industriais começaram a surgir recentemente. Um caso de 2010 chamou bastante atenção, quando um software malicioso, o Stuxnet, foi descoberto. O Stuxnet foi desenvolvido especificamente para explorar uma vulnerabilidade de um modelo de Controlador Lógico Programável. Sua presença altera os parâmetros enviados do campo para o controlador, mantendo os dados originais para os sistemas de supervisão, passando, dessa forma, despercebido (VOLKOVA et al., 2019; GALLOWAY; HANCKE, 2013; CHEN, 2010; KUSHNER, 2013).

Esse caso, que foi apenas o primeiro ataque detectado especificamente para um equipa-

mento industrial, alavancou as pesquisas de segurança em redes industriais. Abordagens importadas diretamente do campo da tecnologia da informação são, muitas vezes, inviáveis para as especificidades da comunicação industrial. Atualmente, há a preocupação de tornar a rede industrial segura em todos os níveis, protegendo o acesso remoto, os sistemas de gerenciamento da planta e de supervisão, as interfaces homem-máquina, os controladores e os dispositivos de campo. A proteção externa de uma rede industrial pode ser feita por controle de acesso através de credenciais (considerado um nível baixo de segurança), pelo estabelecimento de conexões virtuais seguras, e por *firewalls*, que são programas ou equipamentos que monitoram o tráfego da rede (essas duas últimas opções são consideradas mais adequadas pelo nível de segurança que proveem). As funções dos *firewalls* vão desde bloqueadores de acesso para endereços pré-determinados até sistemas de verificação e classificação do conteúdo dos pacotes de dados (FOVINO et al., 2012; GALLOWAY; HANCKE, 2013; CHEMINOD et al., 2018).

As ferramentas supracitadas são exemplos de proteções contra ataques remotos à rede da empresa. Além desses, ataques podem ter sua origem na rede interna. Ataques internos geralmente são realizados por um agente com acesso autorizado aos equipamentos de rede. Esse invasor interno, mesmo se possuir baixo conhecimento cibernético, pode ter conhecimento privilegiado sobre o processo, podendo explorar melhor as falhas. As motivações de um invasor interno podem variar desde problemas pessoais com a empresa até interesses financeiros na "traição" da confiança da empresa (VOLKOVA et al., 2019; GALLOWAY; HANCKE, 2013).

Outro detalhe importante sobre segurança de redes industriais, tratado nos Capítulos anteriores, é o crescente uso de infraestrutura de rede comum às redes locais de computadores. Esse fator pode facilitar a invasão local, pois ataques bem conhecidos em redes comerciais podem ser executados com equipamentos de uso comum, como um notebook. Isso vale para redes industriais que usam o padrão Ethernet, como EtherNet/IP, Profinet, EtherCAT e Modbus TCP (GALLOWAY; HANCKE, 2013; VALENZANO, 2014). O Modbus TCP é uma rede aberta com ampla documentação disponível, amplo espaço de mercado, e que carece de ferramentas de segurança nativas. Por ser o objetivo desse trabalho, na próxima Seção serão detalhadas as vulnerabilidades dessa rede industrial.

3.1 Vulnerabilidades de redes Ethernet e do Modbus TCP

A primeira etapa de diversos ataques em redes locais Ethernet diz respeito à obtenção dos dados dos nós. Em redes Ethernet geralmente são utilizados comutadores ou concentradores para conectar fisicamente os nós. Para que um dispositivo obtenha os dados dos demais em uma rede com um concentrador, basta que esse dispositivo se conecte a ele. Conforme explicado no Capítulo 2, os pacotes de dados são difundidos pelo concentrador

para todos os dispositivos conectados, portanto, um invasor que se conecte terá acesso a todo o tráfego. Com esse acesso, o invasor pode descobrir endereços de rede, endereços MAC e dados característicos dos protocolos utilizados. Quando a rede Ethernet utiliza um comutador, um invasor não tem acesso direto a esses dados, entretanto, é possível descobri-los. O recurso de descoberta de rede é simples e pode ser usado para esse fim. A descoberta de rede pode ser bloqueada no dispositivo, tornando-o inacessível para outros nós na rede, mas esse não é o caso em grande parte das redes locais industriais por ser um recurso constantemente utilizado na verificação de nós ativos e inativos, além de ser necessário para que um determinado nó atue como servidor em uma conexão TCP (KUROSE; ROSS, 2014; VOLKOVA et al., 2019).

Em termos de dados da camada de enlace em redes baseadas em comutador, um invasor pode fazer com que os pacotes de dados sejam destinados a ele no lugar do destino correto. Esse ataque se chama Falsificação do Protocolo de Resolução de Endereços (ou *Address Resolution Protocol* (ARP) *spoofing*). Tomando como exemplo uma rede Modbus TCP, a Falsificação do ARP pode ser feita com o envio de um pacote de dados com endereço MAC remetente adulterado (o do nó hospedeiro do Escravo legítimo, por exemplo). O comutador então atualizará sua tabela de resolução de endereços (ver Seção 2.1.2), que associará o endereço MAC legítimo à porta conectada ao invasor. Com isso, utilizando o exemplo citado, quando o Mestre enviar requisições novas para aquele endereço MAC, quem as receberá será o invasor. Essa interceptação é possível em redes com comutadores, mas não em redes com concentradores, pois estes não possuem a tabela de resolução de endereços. Esse ataque pode ser considerado negação de serviço, por impedir que um nó acesse aquele serviço solicitado, visto que o invasor será o receptor dos pacotes de dados. O mais comum é que a Falsificação do ARP seja uma etapa para executar o ataque Homem-no-Meio (*Man-in-the-Middle*), permitindo que o invasor altere pacotes de dados e os redirecione. A Falsificação do ARP é um ataque que independe dos protocolos das camadas superiores à de enlace, por explorar uma vulnerabilidade do comutador (VOLKOVA et al., 2019; ÅKERBERG; BJÖRKMAN, 2009; HUITSING et al., 2008).

Para evitar a Falsificação do ARP em redes baseadas em comutadores, pode-se fazer uso de uma Rede Local Virtual Privada (*Private Virtual Local Area Network* (PVLAN)) na rede local Ethernet. Alguns modelos de comutadores permitem a criação dessas redes. Nas PVLAN, cada porta física do comutador é configurada como `Promíscua`, `Isolada` ou `Comum`. A porta `Promíscua` pode se comunicar com todas outras, sendo que somente uma porta `Promíscua` é permitida para cada PVLAN. As portas `Isoladas` só são acessíveis para a porta `Promíscua`, e todos seus pacotes de dados são encaminhados para ela (não há comunicação direta entre portas `Isoladas`). As portas do tipo `Comum` são abertas, acessadas tanto pelas outras portas `Comuns` quanto pela porta `Promíscua` (BHAIJI, 2008).

Por se tratar apenas da troca do comutador utilizado, é possível utilizar a PVLAN em quaisquer redes industriais baseadas em Ethernet, incluindo o Modbus TCP. Conectando

o Mestre Modbus a uma porta `Promíscua` e os Escravos às portas `Isoladas`, o invasor não terá acesso aos endereços de rede e MAC dos dispositivos e nada mudará para a rede Modbus TCP, pois os Escravos já possuem comunicação exclusiva com o Mestre. Esta solução não compromete a eficiência da rede nem altera os protocolos utilizados. O uso de PVLAN possui algumas limitações, como um invasor com acesso físico ao dispositivo de rede, que pode efetuar a troca das portas do comutador, prejudicando todo o arranjo da PVLAN. Além disso, o isolamento de portas no comutador apresenta as mesmas desvantagens do bloqueio do recurso de descoberta de rede e, portanto, pode ser indesejável em alguns casos (VOLKOVA et al., 2019).

O ataque ao comutador também pode ser prevenido com o recurso *port security*, que se trata de uma forma de proteger as portas físicas do comutador, através da associação fixa de um endereço MAC a uma porta. Mudanças na tabela ARP só podem ser feitas com acesso especial e, dessa forma, um invasor não consegue realizar a Falsificação do ARP (VOLKOVA et al., 2019).

Além do redirecionamento de pacotes de dados, uma rede Modbus TCP pode ser atacada explorando vulnerabilidades na camada de transporte. O TCP é um protocolo orientado à conexão, no qual um servidor aguarda que um cliente solicite a conexão e, através dela, ambos trocam dados. Os Escravos em uma rede Modbus TCP assumem o papel de servidores e ficam em estado de escuta constante, aguardando conexão com o Mestre, com papel de cliente que, após a conexão, envia suas requisições e aguarda as respostas. A conexão TCP pode ser finalizada quando o Mestre encerrar as requisições ao Escravo, por tempo, ou através de uma mensagem denominada TCP RST. Essa mensagem é usada para finalizar imediatamente uma conexão devido a algum erro detectado na comunicação. Entretanto, um atacante pode usar desse recurso para enviar ele próprio a mensagem TCP RST e terminar uma conexão entre Mestre e Escravo Modbus. A mensagem TCP RST deve ter os dados que identificam a conexão, que são os endereços de rede remetente e destinatário, as portas remetente e destinatário, e o número de sequência, além do comando RST. Se o invasor possuir esses valores, ele pode forçar o término da conexão, impedindo o Mestre de enviar requisições até que estabeleça uma nova conexão, causando Negação de Serviço (DU, 2019; KUROSE; ROSS, 2014; HUITSING et al., 2008).

Outro ataque visando à camada de transporte é o Sequestro TCP (TCP *hijacking*), no qual um invasor envia seus pacotes de dados entre os pacotes de dados legítimos em uma conexão já estabelecida. Esse ataque é possível se o invasor possuir os dados que identificam a conexão TCP (através da escuta ou da interceptação) e enviar pacotes de dados ao Cliente ou ao Servidor (Mestre e Escravo na rede Modbus TCP) com esses campos corretos. O nó que receber as mensagens acreditará que aquele pacote de dados faz parte da conexão e irá trata-lo como legítimo. O número de sequência é extremamente importante para o sucesso desse ataque, por ser variável e crescente. Se o número de sequência na invasão for muito maior que o valor correto ou menor, o ataque não surtirá

efeito. No primeiro caso, o número de sequência maior que o esperado fará com que o pacote só seja interpretado quando "chegar sua vez". No segundo caso, o pacote será descartado por ser considerado um reenvio. Entretanto, se o algoritmo do atacante gerar números de sequência iguais ou pouco maiores que os números de sequência que seriam gerados pelo nó legítimo, há a chance de que os pacotes de dados legítimos é que serão descartados, por serem considerados redundantes, a depender de qual pacote de dados chegará primeiro ao destino (DU, 2019; ALOTAIBI et al., 2017; HUITSING et al., 2008).

O sequestro da conexão TCP pode ser usado em redes Modbus TCP de duas formas, a primeira caracterizada pelo invasor que assume o papel de Escravo, enviando respostas falsas às requisições direcionadas a um Escravo legítimo. A segunda forma é o invasor assumir papel de Mestre, enviando Requisições extras ao Escravo que já estava recebendo requisições legítimas.

Para prevenir ataques baseados em vulnerabilidades do TCP, Hayes e El-Khatib (2013) propuseram o uso de *Stream Control Transmission Protocol* (SCTP) na camada de transporte, combinado com *Hash-based Message Authentication Code* (HMAC). Essa abordagem troca o protocolo da camada de tranporte por uma versão mais segura, fazendo com que a rede não seja mais uma rede Modbus "TCP". Em outras palavras, todos os dispositivos na rede devem ter compatibilidade com o SCTP e o HMAC. Uma abordagem semelhante foi introduzida pelo documento oficial que descreve o protocolo Modbus TCP Security - ou apenas Modbus Security (MODBUS ORG., 2018). Este protocolo, bem como outras proteções desenvolvidas especificamente para redes Modbus, serão discutidas na seção 3.2.

Além das vulnerabilidades da camada de enlace e de transporte, o Modbus TCP também é vulnerável na camada de aplicação, onde está sujeito a comandos não autorizados, devido à falta de autenticação do Mestre e do Escravo. Um dispositivo invasor, sem necessidade de Falsificação do ARP, Negação de Serviço ou Sequestro TCP, pode facilmente assumir o papel de Mestre na rede. Isso por que os Escravos estão em modo de escuta constante e não existem ferramentas nativas ao Modbus TCP que rejeitem um pedido de nova conexão. Com isso, o invasor pode simplesmente estabelecer uma nova conexão com um Escravo. O invasor ainda precisará dos endereços de rede e MAC do alvo para escrever valores nas suas bobinas e registradores, bem como para requisitar dados das entradas. Isso pode ser feito individualmente com o identificador de unidade, ou utilizando mensagens em difusão com o identificador de unidade zero (ver Seção 2.2). As requisições do Mestre falso serão aceitas pelos Escravos e acatadas, podendo causar acidentes, danos materiais, perda de eficiência de controle de processo e, em casos menos invasivos, leitura não autorizada de variáveis (MODBUS ORG., 2012; HUITSING et al., 2008).

Ainda explorando vulnerabilidades específicas do Modbus TCP, há uma forma de realizar a Negação de Serviço no nível de aplicação. Ela pode ser feita pela inundação da rede com uma grande quantidade de requisições Modbus por um invasor em um pequeno

intervalo de tempo, tornando impossível que o Escravo trate todas e, consequentemente, não responda às requisições legítimas do Mestre. Essa sobrecarga também é factível nas outras camadas, como a inundação de requisições de conexão TCP (SYN *flooding*) ou a inundação de pacotes de dados para o dispositivo de rede (MAC *flooding*) (MODBUS ORG., 2012; HUITSING et al., 2008). Além disso, Homem-no-Meio e Ataque de Repetição (*Replay Attack*) são exemplos de ataques que podem explorar vulnerabilidades em camadas inferiores, combinados com vulnerabilidades do próprio Modbus (FOVINO et al., 2009). A Figura 9 exemplifica uma rede Modbus TCP e as vulnerabilidades descritas.

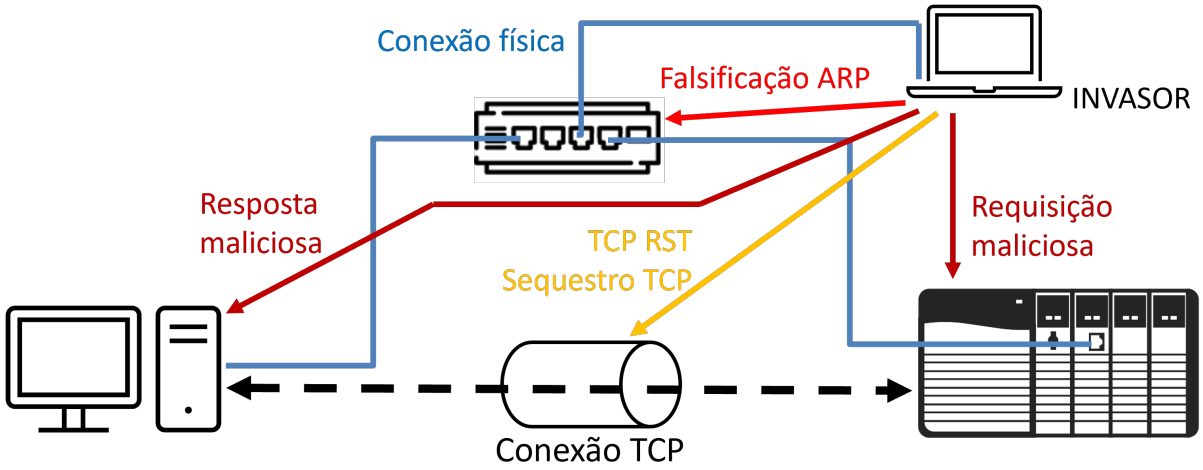

Figura 9 – Exemplo de rede Modbus TCP e os principais alvos de um invasor local
Fonte: Elaborada pelo autor

Considerando essas vulnerabilidades da camada de aplicação, tanto em redes Modbus TCP quanto nas demais, alguns estudos indicam que a implementação de técnicas de segurança na camada de aplicação não é apenas um complemento, mas uma necessidade. Aplicações como Internet das Coisas se beneficiam de protocolos como o *Object Security for COnstrained Restful Environments* (OSCORE), que pode ser usado para atingir o nível de segurança desejado (PéREZ et al., 2020). Outros estudos mostraram que pode ser eficiente implementar técnicas de segurança de outras camadas no nível de aplicação, como criptografia e troca segura de chaves, conforme oferecido pelos protocolos *Transport Layer Security* (TLS) e *Secure Sockets Layer* (SSL) (NARAYANASWAMY; KUMAR, 2019; ALIZAI et al., 2019).

3.2 Estudos relacionados

Nesta seção são discutidas soluções de segurança aplicáveis a quaisquer redes industriais, incluindo Modbus TCP. Também são discutidas soluções desenvolvidas especificamente para aplicações Modbus.

Ao se considerarem técnicas de segurança para redes industriais, fatores importantes devem ser levados em conta. O primeiro deles é a longa vida útil dos equipamentos industriais, algo que pode fazer com que técnicas de segurança eficientes no período de implantação se tornem obsoletas enquanto o equipamento ainda está operante. O segundo fator é o tamanho da porção de dados responsável pela técnica, que influenciará tanto no armazenamento necessário nos dispositivos quanto no acréscimo no tamanho do pacote de dados trafegado. O terceiro fator refere-se à complexidade de implementação de alguns algoritmos, pois a capacidade de processamento de equipamentos industriais pode ser limitada (GALLOWAY; HANCKE, 2013).

Os dois últimos fatores influenciam no quarto fator, que é o acréscimo de tempo. Pacotes de dados maiores levam mais tempo para serem transmitidos, além de levarem mais tempo para serem processados, principalmente se o processamento for complexo, como em alguns algoritmos de criptografia e autenticação. Conforme já dito, a técnica implementada em um dispositivo deve ser mantida pela vida útil do mesmo, que pode ser de vários anos. Portanto, para atingir a segurança em uma rede industrial, é importante considerar uma ou várias técnicas de segurança que assegurem equilíbrio entre complexidade, armazenamento e vida útil (GALLOWAY; HANCKE, 2013).

Atendendo alguns desses critérios, Fovino et al. (2012) propõem um *firewall* na rede interna da empresa, monitorando todo o tráfego. A função tradicional do *firewall*, de identificar possíveis pacotes invasores através de características preestabelecidas, é revista, constatando-se que essa identificação não cobre certas situações danosas e específicas de um sistema de automação industrial. O estudo cita o exemplo de um sistema de controle de temperatura com uma válvula de saída e uma de entrada de vapor. Nesse sistema, se um invasor envia um comando para abrir totalmente a válvula de entrada e fechar totalmente a válvula de saída, do ponto de vista de um algoritmo de identificação de pacotes de dados invasores, os comandos são válidos, mas podem causar danos sérios ao processo. Para prevenir esse tipo de situação, a proposta dos autores é que o *firewall* conheça as situações indesejadas do sistema, chamadas de estado crítico. Com isso, é possível determinar se o estado crítico do processo está se aproximando ou foi atingido. Para atingir esse nível de monitoria, é necessário um modelo virtual do sistema, que deve contar com a representação de seus estados críticos, além de um monitor de estados que, junto à representação dos estados, detecta a aproximação ou o próprio estado crítico.

Com o modelo e o monitoramento dos estados, o *firewall* proposto bloqueia no nível de rede pacotes que possam levar um sistema de automação industrial ao estado crítico e avisa quando o sistema está chegando próximo ao nível crítico. Para mensurar o impacto, os autores instalaram o *firewall* entre o sistema de supervisão e o controlador do processo, o que acrescentou um atraso de comunicação máximo de 2 ms. O *firewall* requisita os dados do controlador para monitorar o sistema (realizando a leitura de até 2000 bobinas em menos de 1 ms) e o tempo para cálculo de 2000 regras de estado crítico leva cerca de

60 ms. Os autores avaliam a eficiência da proposta com a medição dos falsos positivos (pacotes legítimos bloqueados), que foi de 0,12%, e falsos negativos (pacotes invasores não bloqueados), que foi de 0,08%. Uma das principais justificativas para os falsos positivos ou negativos foi o fato de um determinado estado crítico não ter sido incluído nas regras. A proposta independe do protocolo de rede utilizado e pode ser aplicado em conjunto com outras técnicas de segurança, até porque que não previne ataques à infraestrutura da rede ou Negações de Serviço. A grande contribuição é que o *firewall* proposto pode proteger contra qualquer ataque, desde que esse cause uma situação definida como estado crítico. Apesar disso, a técnica demanda profundo conhecimento sobre o sistema monitorado e um árduo trabalho de programação do *firewall* (FOVINO et al., 2012).

Com uma abordagem similar, o trabalho de Sestito et al. (2018) apresenta um detector de anomalias, que são situações que fazem com que a rede saia do que foi definido como funcionamento normal. As anomalias podem ser interrupções de funcionamento de rede, aumento súbito de demanda de tráfego, falhas de diagnóstico e ataques à rede. Um classificador é aplicado aos pacotes trafegados na rede, diferenciando pacotes de tráfego normal de pacotes anormais, que podem ser causados pelos motivos supracitados. Por ser uma abordagem que depende do monitoramento da rede e independe de ações dos dispositivos em comunicação, pode ser usado em conjunto com outras técnicas, o que pode contribuir para a segurança do sistema de automação industrial. Além disso, é mais completa que a solução de Fovino et al. (2012), pois pode detectar situações como a interrupção da conexão e a inundação da rede, visto que são situações de tráfego anormal, e utiliza classificadores automáticos para a modelagem do tráfego da rede.

Aplicações de segurança específicas do Modbus foram desenvolvidas ao longo dos anos. O *Secure Modbus*, apresentado em 2009, propõe um encapsulamento do pacote de dados Modbus TCP pelo uso da função *Secure Hashing*, proporcionando garantia de integridade aos pacotes de dados. No *Secure Modbus*, o serviço de assinaturas criptográficas RSA garante que não haja acesso não autorizado (confidencialidade). A abordagem também usa carimbos de tempo (*timestamps*) para prevenir Ataques de Repetição (FOVINO et al., 2009).

Outra solução para as vulnerabilidades de redes Modbus foi proposta por Hayes e El-Khatib (2013), consistindo no uso do SCTP na camada de transporte, aliado à técnica de autenticação HMAC. Os dados Modbus são encapsulados em um cabeçalho de 36 bytes (SCTP e HMAC), que inclui técnicas de segurança contra comandos e leituras não autorizadas, interceptação e Negação de Serviço. Para comparação, o cabeçalho TCP possui 20 bytes, logo, a proposta acrescenta 16 bytes ao pacote de dados.

Também em 2013, foi publicada uma proposta de segurança baseada na modelagem de mensagens Modbus periódicas entre uma Interface Humano-Máquina e um Controlador Lógico Programável. A proposta foi traçada da seguinte forma: se as mensagens Modbus seguissem um padrão cíclico, as invasões poderiam ser caracterizadas por mensagens fora

desse padrão (GOLDENBERG; WOOL, 2013). A proposta se assemelha às propostas de Fovino et al. (2012) e Sestito et al. (2018), entretanto foi projetada especificamente para a camada de aplicação, detectando mensagens Modbus fora do padrão.

O protocolo uBus é uma modificação do Modbus original, projetado para barramentos EIA/TIA 485, e foi desenvolvido com base no Modbus visando mitigar suas limitações de segurança. O uBus possui implementações extras sobre o Modbus Mestre-Escravo tradicional, incluindo criptografia de mensagem e troca de chave pública (DUDAK et al., 2019).

A segurança Modbus também foi aplicada com base em SSI (*Self-Sovereign Identity*), proporcionando resultados promissores na autenticação e autorização de acesso para sistemas de automação industrial de forma descentralizada. A segurança descentralizada traz grandes benefícios para aplicações de Internet das Coisas Industrial (LORENZO; BENITO; ARRIZABALAGA, 2021).

Convém citar que existem publicações que visam normatizar os procedimentos e técnicas de segurança em sistemas de automação industrial. A norma IEC 62443 traz procedimentos e requisitos de segurança para esses sistemas, com preocupações como procedimentos operacionais e treinamento de pessoal, bem como técnicas para levantamento de riscos. Já a norma IEC 62351 abrange a questão de segurança de forma mais tecnológica, apontando ferramentas de segurança para situações específicas (IEC, 2021; SCHLEGEL; OBERMEIER; SCHNEIDER, 2017).

Em 2018, a organização Modbus apresentou o *Modbus Security*, que especifica o uso do protocolo TLS para a camada de transporte. O TLS é recomendado pela IEC 62351 e permite várias técnicas de segurança da camada de transporte. Incluídos no TLS, estão a troca segura de chaves para autenticação e não repúdio (*non-repudiation*) [1]; a criptografia para confidencialidade; e as funções *hash* para integridade. As entidades em uma rede *Modbus Security* devem concordar com um conjunto de regras utilizadas, possibilitando um ambiente seguro com menos ou mais custos de processamento e transmissão, dependendo dos requisitos de segurança da aplicação (MODBUS ORG., 2018; RESCORLA, 2018; SCHLEGEL; OBERMEIER; SCHNEIDER, 2017).

As propostas de segurança para o Modbus anteriores ou contemporâneas ao *Modbus Security* contribuíram para identificar vulnerabilidades que a especificação oficial aborda. Mas considerando que o *Modbus Security* pode ser muito complexo para algumas aplicações (ver Capítulo 7) e tentando trazer uma alternativa com menor impacto na comunicação e no processamento, o próximo Capítulo inicia a apresentação de nossa proposta de segurança.

[1] O não repúdio é prevenir que um atacante negue uma ação legítima (FOVINO et al., 2009). Em uma rede Modbus padrão, por exemplo, um Mestre invasor pode enviar uma requisição posterior para sobrescrever uma determinada requisição legítima

Capítulo 4

Protocolo de Autenticação HP-MP*

Em algumas aplicações de automação industrial, pode ser necessária uma combinação de mais de uma técnica para cumprir todos os requisitos de segurança da rede. Porém, para redes industriais, a complexidade de cada técnica é relevante, pois técnicas mais complexas exigem mais processamento e memória. O aumento no tamanho do pacote de dados também é relevante, pois aumenta o tempo necessário para troca de dados (GALLOWAY; HANCKE, 2013). Considerando essas características, e que o objetivo final de muitos ataques à rede é no nível da aplicação, esta tese propõe uma forma de tornar o próprio protocolo da camada de aplicação, no caso o Modbus, mais seguro. Com essa abordagem, a segurança geral do sistema de automação industrial pode ser incrementada com outras técnicas em outras camadas, ou o sistema pode aplicar a proposta como única solução de segurança, caracterizando uma alternativa com menor impacto para a rede.

A pesquisa por uma técnica que atendesse às limitações e expectativas citadas nos levou à seleção do protocolo de autenticação HB-MP* [1]. O protocolo HB-MP* faz parte da família de protocolos HB, composta por algoritmos de autenticação de dispositivos, em sua maioria aplicados em Identificação por Radiofrequência (*Radio-Frequency IDentification* (RFID)). Esses protocolos proveem segurança com operações simples, visto que os dispositivos RFID possuem arquitetura simples e hardware limitado.

O primeiro protocolo HB foi apresentado em 2001, funcionando com uma técnica de autenticação por chave compartilhada. Nessa versão, o dispositivo autenticador, de posse da chave, envia um desafio binário gerado aleatoriamente para o dispositivo a ser autenticado. O dispositivo desafiado deve responder com a operação $z = a \cdot x \oplus v$, na qual z é a resposta ao desafio, a é o desafio aleatório recebido, x é a chave compartilhada e v é um ruído. O dispositivo autenticador realiza a mesma operação, sem o ruído, e verifica se a resposta recebida coincide com a resposta calculada. O ruído inclui respostas propositalmente incorretas para dificultar a descoberta da chave secreta por um dispositivo

[1] HB são as iniciais dos desenvolvedores do primeiro protocolo HB, Hopper e Blum. MP são as iniciais de Munilla e Peinado, desenvolvedores do HB-MP. O asterisco (*) indica que o HB-MP* é versão mais recente do HB-MP.

invasor que, com acesso aos dados trafegados, pode tentar descobri-la através dos desafios e respostas capturados. Entretanto, já que há a presença de algumas respostas incorretas, mesmo em um dispositivo que possui a chave correta, a autenticação é efetivada não apenas com um desafio e uma resposta, mas após uma certa quantidade de ciclos, caso a maioria das respostas seja correta. A quantidade de bits de x, z, a e v deve ser igual, devido às operações envolvidas. É uma técnica que envolve operações simples e no nível binário, com poucos bits armazenados na memória e poucos bits trafegados para autenticação (HOPPER; BLUM, 2001).

Devido à simplicidade do protocolo HB original, capturando uma quantidade suficiente de dados, o invasor ainda conseguia identificar a chave secreta. Diante disso, esse protocolo foi sendo aprimorado ao longo dos anos, mas sempre com a preocupação de manter as operações simples e a quantidade de bits baixa (as primeiras versões poderiam ser implementadas com apenas um byte para memória e mensagens). Dentre as versões do protocolo HB, se destaca a HB+, de Juels, Weis et al. (2005), que inclui uma segunda chave de autenticação compartilhada pelos nós. Isso aumenta a confiabilidade da técnica, pois pode dobrar o trabalho necessário para uma invasão. Posteriormente, surgiu o HB-MP, apresentado por Munilla e Peinado (2007). O HB-MP continua utilizando duas chaves e adiciona a operação de rotação dos bits, dificultando ainda mais o trabalho do invasor.

Até o desenvolvimento de nossa proposta de trabalho a versão HB-MP* era a mais recente da família de protocolos HB. Comparado aos protocolos anteriores, o HB-MP* tem algumas vantagens. Nessa versão, os autores mantiveram as operações do protocolo HB-MP, porém desenvolveram um algoritmo que utiliza apenas uma chave, economizando armazenamento nos dispositivos. Na apresentação de sua proposta, Aseeri et al. (2016) testaram e consideraram o protocolo mais seguro do que seus predecessores. Um ciclo de desafio e resposta do protocolo HB-MP* é mostrado no Quadro 1.

Conforme o Quadro 1 e o trabalho de Aseeri et al. (2016), ambos nós possuem uma chave X. A chave X deve ser igual a um número primo e também deve ser representada por um número de bits primo [2]. A cada iteração o nó 1, o autenticador, gera um desafio aleatório A. Os bits A' (gerados pela operação OU Exclusivo entre A e X) são enviados para o nó 2, que será autenticado. O nó 1 consegue calcular A, visto que também possui a chave X. Ambos os nós calculam $\hat{A}$ e S (que correspondem aos valores de A e X rotacionados, respectivamente).

O nó 2 gera Z através da operação OU Exclusivo entre três vetores: o primeiro vetor é o resultado do produto escalar entre A e $\hat{A}$, o segundo vetor é o resultado do produto escalar entre X e S, e o terceiro vetor é o ruído aleatório V. O ruído aleatório continua com a mesma função do protocolo HB original, de incluir respostas incorretas intencionalmente.

[2] Os números primos são usados em técnicas de segurança por não serem fatoráveis, diminuindo ainda mais as chances de um invasor conseguir descobrir o número de forma forçada

Quadro 1: Um ciclo de autenticação pelo protocolo HB-MP* e suas notações

Etapa	**Nó 1**	**Nó 2**
0	Chave X (k bits, k número primo)	Chave X
1	Desafio aleatório A (k bits)	
2	Envia A' (A' = A $\oplus$ X)	
3		Recebe A'
4	Calcula $\hat{A}$	A = A' $\oplus$ X
5	Calcula S	Calcula $\hat{A}$
6		Calcula S
7		Gera ruído aleatório V com maior probabilidade de ser 0
8		Z = A · $\hat{A}$ $\oplus$ X · S $\oplus$ V
9		Vetor aleatório B (B · S = Z)
10		Envia B' = B $\oplus$ X
11	Recebe B' e calcula B = B' $\oplus$ X	
12	Verifica se A · $\hat{A}$ $\oplus$ X · S = B · S	

$\oplus$ = Operação OU Exclusivo
· = Produto escalar de dois vetores
$\hat{A}$ = Rotação à esquerda de A n posições, sendo n o decimal equivalente aos 4 bits mais significativos de A
S = Rotação à esquerda de X q posições, sendo q o decimal equivalente aos 4 bits mais significativos de ($\hat{A}$ $\oplus$ X)

Fonte: Adaptado de Aseeri et al. (2016)

O nó 2, de posse de Z, gera um vetor aleatório B, que, embora aleatório, deve satisfazer a condição de resultar em Z quando aplicado em um operação de produto escalar com S. O vetor B é a resposta HB-MP* ao desafio do nó 1, mas antes de ser enviada, é modificada para B' pela operação OU Exclusivo entre B e X. Com o valor de B', o nó 1 consegue verificar a autenticidade do nó 2. Há de se destacar que, devido ao ruído V, são previstos falsos negativos em um único ciclo, portanto a autenticação deve ser considerada após uma determinada quantidade de ciclos.

Por apresentar características de pouco custo de processamento e de comunicação, a aplicação do protocolo de autenticação em redes industriais é viável e vantajosa. Sob essa perspectiva, esta tese se propôs a aplicar o HB-MP* a uma rede Modbus TCP para evitar gravações e leituras não autorizadas por mestres invasores e injeção de respostas Modbus por escravos invasores. O protocolo HB-MP* foi incluído como parte do campo de dados Modbus, adicionando segurança à camada de aplicação. O descritivo do algoritmo será retomado especificamente para nossa aplicação na seção 5.1, pois algumas adaptações à versão original foram feitas. Nossa proposta de segurança será chamada de HB Modbus quando comparada com outras soluções nas seções subsequentes.

Capítulo 5

Metodologia

Este capítulo descreve os métodos utilizados para a programação do HB Modbus, proposto nesta tese. O HB Modbus é baseado no protocolo HB-MP*, que foi selecionado por suas baixas exigências de memória e processamento, e pequeno acréscimo de dados trafegados. Também é apresentada a metodologia de validação da proposta. Os recursos da validação utilizados foram:

- PC 1: um computador pessoal com sistema operacional Windows 10, processador Intel Core i5 e memória RAM de 8 GB. O PC 1 é o hospedeiro (*host*) do Mestre HB Modbus (ou somente Mestre HB) e dos contêineres.
- PC 2: um computador pessoal com sistema operacional Windows 8.1, processador Intel Core i5 e memória RAM de 6 GB. O PC 2 é o hospedeiro do simulador de Escravos ModbusPal.
- BBB 1 e BBB 2: dois sistemas embarcados BeagleBoneBlack com sistema operacional Debian, processador AM335x ARM Cortex-A8 e memória RAM de 512 MB. Cada BBB atuou como Escravo HB Modbus (ou somente Escravo HB).
- Periféricos para os BBBs: 4 LEDs de 5 mm, 4 resistores 0.25 W (180 ohms) e fios condutores de uso geral.
- Dispositivo de rede: um dispositivo de rede provedor de acesso à internet, modelo ONU UNEE, com funções de roteamento e comutação de pacotes, utilizado como substituto ao comutador.
- Cabos Ethernet categoria 5e: utilizados para conectar os BBBs ao dispositivo de rede.
- Plataforma IDLE: instalada no PC 1 e no PC 2 para programação na linguagem Python.
- Wireshark: instalado no PC 1 para análise de rede.

- ModbusPal: um simulador de Escravos Modbus instalado no PC 2.
- Plataforma Docker Desktop: um construtor e gerenciador de contêineres instalado no PC 1.
- Contêineres Python3: pacotes de software independentes contendo tudo o que é necessário para executar um Escravo HB simulado. Os contêineres foram usados devido às limitações de hardware para implementar mais Escravos físicos HB, bem como para que os programas fossem executados em um ambiente confiável.

Os recursos de hardware, que são os PCs 1 e 2, os BBB 1 e 2, os periféricos para os BBBs, o dispositivo de rede e os cabos Ethernet, são mostrados na Figura 10.

Figura 10 – Recursos de hardware utilizados na tese
Fonte: Catálogo pessoal

5.1 Programação do HB-MP*

O desenvolvimento de nossa proposta começou com a programação do protocolo HB-MP*. Todas variáveis no protocolo devem possuir o mesmo número de bits (k). Para

definir o valor de k, considerou-se que há duas operações de rotação descritas no protocolo, sendo que a maior rotação possível é de 15 posições. Para garantir que a rotação fosse menor que a quantidade de bits, considerou-se usar 17 bits, por ser o menor número primo maior que as 15 posições de rotação. Entretanto, os pacotes de dados Ethernet são divididos em bytes, logo 17 bits ocupam a mesma quantidade de bytes que 24 bits. O número 23, que também é primo, foi então selecionado, considerando que a maior quantidade de bits poderia incrementar a eficiência da técnica. Dentre as opções de números primos representados por 23 bits (com o bit mais significativo igual a 1), o número 5969621 ($5B16D5_{hex}$) foi escolhido arbitrariamente. Em testes posteriores, foram usados 31 bits e uma nova chave foi selecionada, sendo que os motivos dessa mudança e seus resultados são mostrados na seção 6.1. O protocolo HB-MP* foi desenvolvido para executar a autenticação conforme os passos descritos no Capítulo anterior e resumidos no fluxograma da Figura 11. Os passos 1 e 5 são executados pelo nó autenticador, já os passos 2 e 4 são executados pelo nó autenticado, e o passo 3 é executado por ambos.

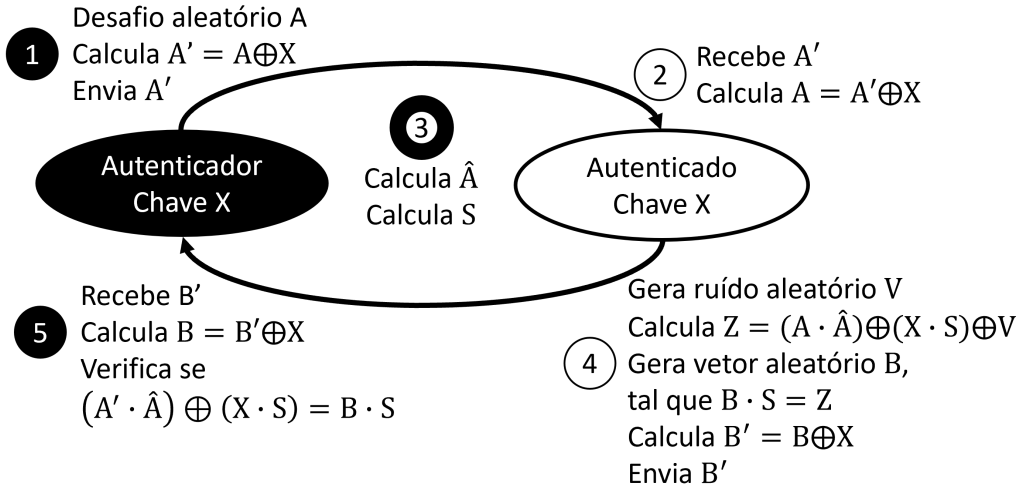

Figura 11 – Processo de autenticação com o protocolo HB-MP*
Fonte: Elaborada pelo autor

O processo de autenticação foi incluído como parte do protocolo da camada de aplicação Modbus. Com isso, após a conexão TCP, um Mestre envia uma requisição Modbus e um desafio modificado (A') ao Escravo conectado. O Escravo deve executar a requisição e enviar de volta uma resposta Modbus, bem como a resposta modificada (B') ao desafio HB-MP*. Com (B') o Mestre é capaz de verificar a autenticidade do Escravo.

Na concepção original, o HB-MP* executava autenticação apenas unidirecional (ASE-ERI et al., 2016). Na presente proposta, o Escravo também deve autenticar o Mestre e, para isso, foi feita uma modificação no protocolo. Os pacotes de dados Modbus passa-

ram a incluir tanto um desafio quanto uma resposta HB-MP*, permitindo a autenticação bidirecional. Considerando a característica aleatória da resposta (vetor B), e que os desafios A também são aleatórios, ficou definido que cada resposta HB-MP* também será considerada como um desafio. Como ficará mais claro na próxima seção, ao receber uma requisição Modbus com o desafio HB-MP*, o Escravo deverá enviar uma resposta Modbus com uma resposta HB-MP* ao desafio do Mestre. Essa resposta HB-MP* é considerada também um desafio, por parte do Escravo, para que o Mestre responda na próxima requisição. Com isso, há redução nos custos computacionais para gerar desafios, e o aumento do pacote de dados continua sendo de 3 ou mais bytes, não 6 ou mais. Pelo fato da resposta modificada B' e o próximo desafio modificado A' serem iguais, exceto na primeira requisição Modbus, eles são chamados simplesmente de dados HB-MP*. O pacote de dados Modbus TCP resultante com os dados HB-MP* é mostrado na Tabela 2.

Tabela 2 – Pacote de dados Modbus TCP com autenticação por HB-MP*

Cabeçalho Ethernet	**Cabeçalho IP**	**Cabeçalho TCP**	**Endereço (ID) do escravo**	**Código da função**	**Dados Modbus**	**Dados HB-MP***
14 bytes	20 bytes	26 bytes	1 byte	1 byte	0-249 bytes	3 ou mais bytes

Fonte: Elaborada pelo autor

A programação do protocolo HB-MP* e das funções Modbus foi feita na plataforma IDLE, com a linguagem Python em sua versão 3.7. Foram desenvolvidos dois programas: Mestre HB e Escravo HB. O trabalho focou em demonstrar a aplicabilidade e as contribuições da autenticação no Modbus TCP, e não no desenvolvimento de uma aplicação Modbus com todas funções, por isso os programas foram limitados ao Código de função 06: escrita em registrador, bem comum mesmo em dispositivos Modbus mais simples. A inclusão de outras funções não impactaria o processo de autenticação, como é mostrado no capítulo 6. Os programas são descritos na próxima seção.

5.2 Mestre HB e Escravo HB

Os programas Mestre HB e Escravo HB, desenvolvidos para realizar a comunicação por Modbus TCP com autenticação por HB-MP*, têm suas funções e interações (linhas tracejadas) ilustradas no fluxograma da Figura 12. O código-fonte está disponível no repositório GitLab pelo link: <https://gitlab.com/fredericofagundes.cefetmg/hb-modbus.git>.

Tanto o Mestre HB quanto o Escravo HB começam com a variável n igual a zero, a qual conta as interações de autenticação entre os dois. O Mestre HB possui um valor de n para cada Escravo na rede. Após a conexão TCP, o Mestre HB verifica se esta é a primeira comunicação a ocorrer com um determinado Escravo HB. Nesse caso, ele gera um desafio aleatório A e armazena A' como Dados HB-MP*. Caso contrário, o Mestre HB já haveria sido desafiado pelo Escravo HB e deveria gerar uma resposta HB-MP* igual a

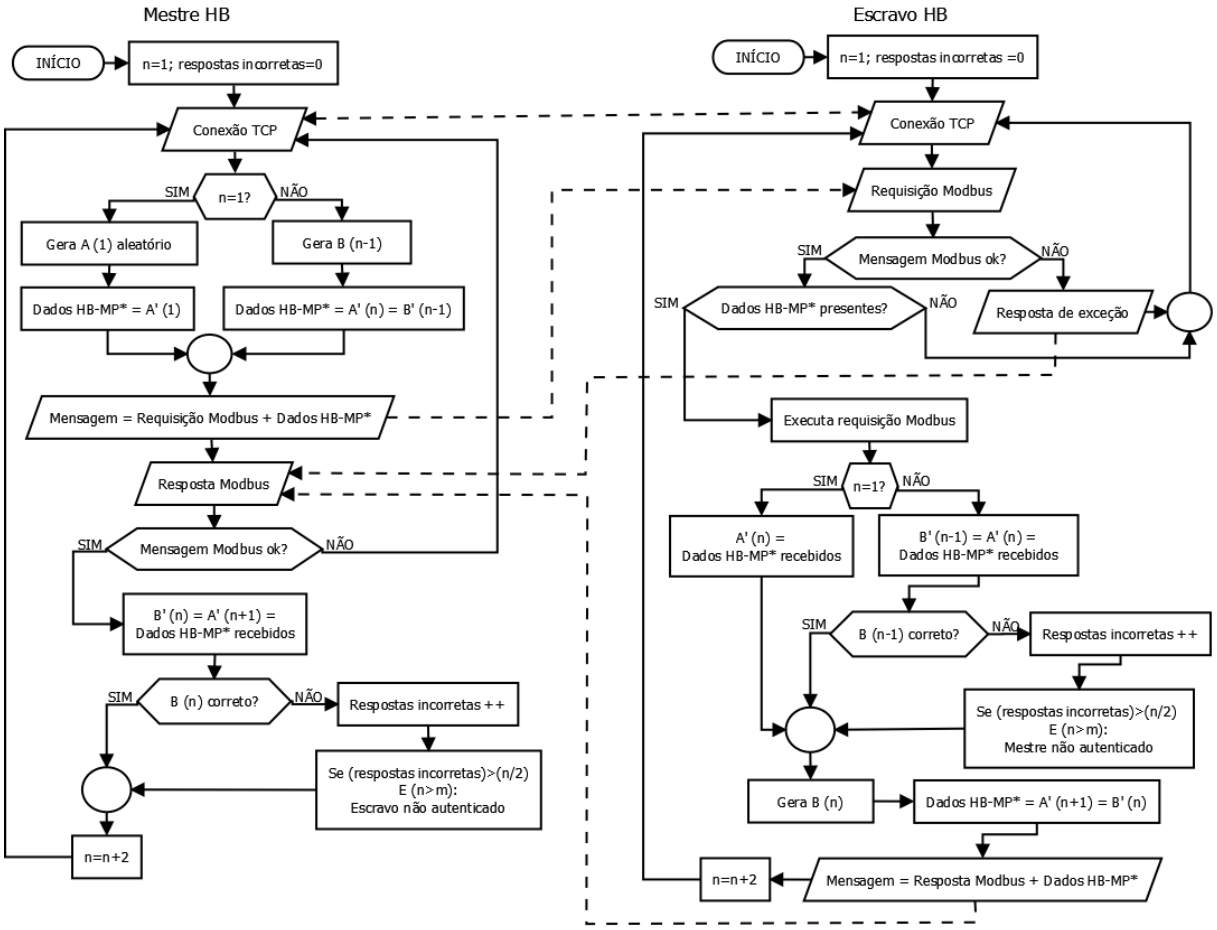

Figura 12 – Fluxograma geral dos programas HB Modbus e mensagens trocadas
Fonte: Elaborada pelo autor

B. B' (B modificado) é armazenado como Dados HB-MP*, usados tanto para autenticar o Escravo ($A'(n)$) quanto para que o Mestre seja autenticado pelo Escravo ($B'(n-1)$). Essas primeiras etapas são destacadas do fluxograma principal e detalhadas na Figura 13.

A mensagem do Mestre HB é formada pela requisição Modbus e pelos dados HB-MP*, sendo que a única diferença é como o Mestre HB gera os dados HB-MP*. Após receber essa mensagem, o Escravo HB examina o endereço e o código de função. O Escravo HB também verifica a presença dos dados HB-MP* na mensagem, visto que ele foi programado para não executar a requisição se não houver dados de autenticação. Se todos os campos estiverem em conformidade, o Escravo HB executa a requisição Modbus. Essa etapa de envio e verificação inicial da mensagem pelo Escravo HB é destacada no fluxograma detalhado da Figura 14.

Posteriormente à execução da requisição, o Escravo HB verifica se essa é a primeira requisição do Mestre HB (através da variável n). Caso seja, o Escravo HB ainda não enviou um desafio ao Mestre HB; com isso, os dados HB-MP* recebidos correspondem somente a A' vindo do Mestre HB. Caso não seja a primeira requisição, o Escravo HB já terá enviado um desafio ao Mestre HB e os dados HB-MP* correspondem à $A'(n)$ (desafio

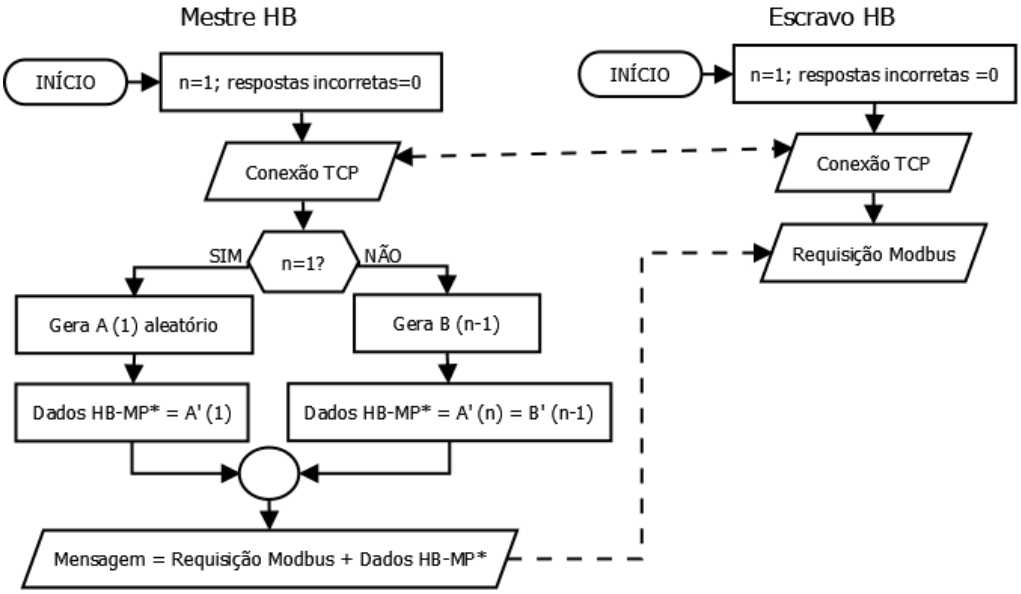

Figura 13 – Fluxograma detalhado das etapas de conexão TCP e geração dos dados HB-MP* pelo Mestre HB

Fonte: Elaborada pelo autor

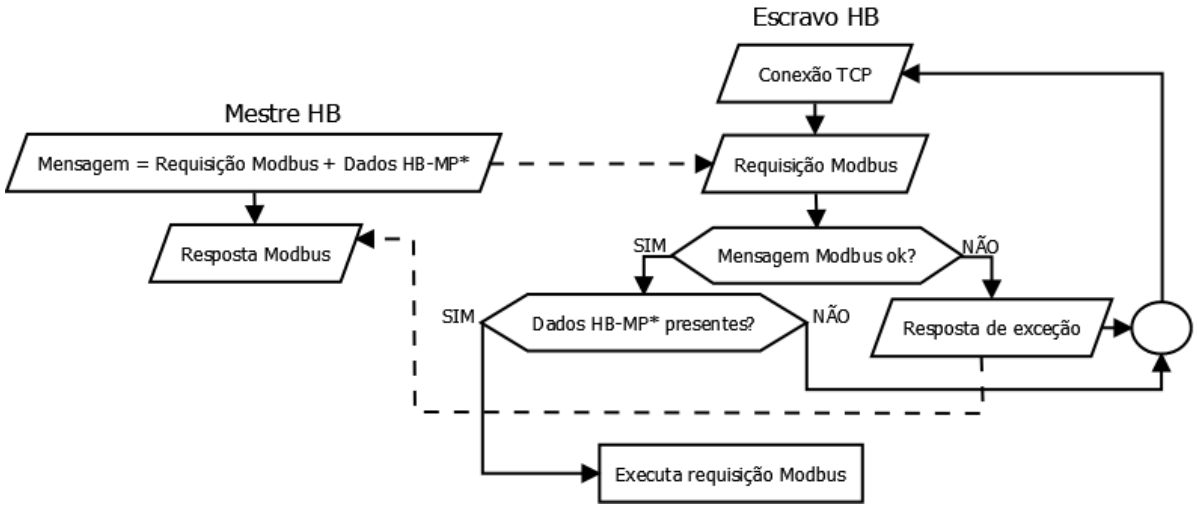

Figura 14 – Fluxograma detalhado da etapa de verificação inicial da mensagem pelo Escravo HB

Fonte: Elaborada pelo autor

modificado do Mestre HB) e à $B'(n-1)$ (resposta modificada do Mestre HB ao desafio anterior do Escravo HB). Nesse segundo caso, o Escravo HB verifica se o Mestre HB respondeu ao desafio corretamente. Em caso negativo, o Escravo HB verifica se já existe uma maioria de respostas incorretas e, se houver, indica a não autenticidade do Mestre HB. Essa indicação é feita apenas após uma quantidade arbitrária de ciclos m. Essas etapas são ilustradas no fluxograma detalhado da Figura 15.

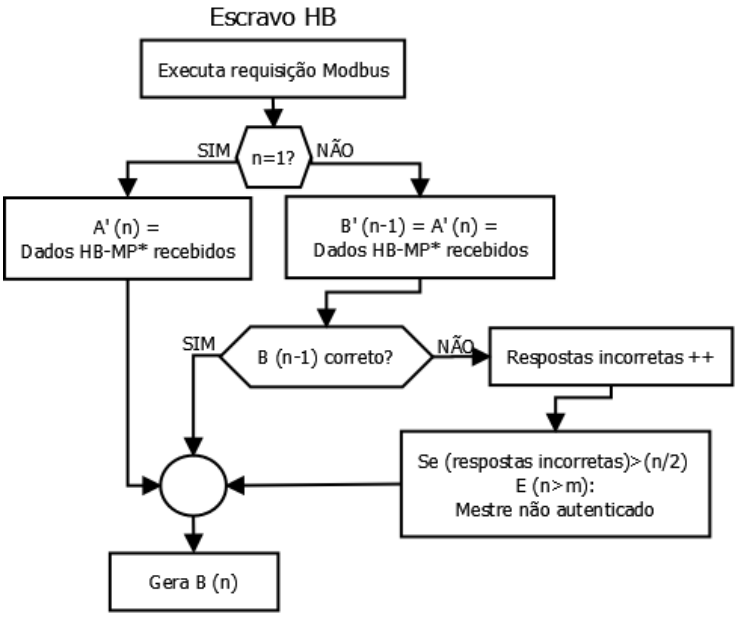

Figura 15 – Fluxograma detalhado das etapas de decisão executadas pelo Escravo HB
Fonte: Elaborada pelo autor

Após as decisões sobre como tratar os dados HB-MP* do Mestre HB, o Escravo HB gera seus próprios dados HB-MP*, correspondentes à resposta $B'(n)$ ao desafio atual do Mestre HB, e ao desafio $A'(n+1)$ para ser respondido no próximo ciclo pelo Mestre HB. O Escravo HB envia uma mensagem com a resposta Modbus e os dados HB-MP*. Antes de retornar ao estado no qual aguarda uma nova conexão TCP, o Escravo HB incrementa em duas unidades a variável n, contando o número total de ciclos de autenticação entre o Mestre HB e o Escravo HB, não apenas o número de autenticações executadas pelo próprio Escravo HB. As etapas finais realizadas pelo Escravo HB são mostradas no fluxograma detalhado da Figura 16.

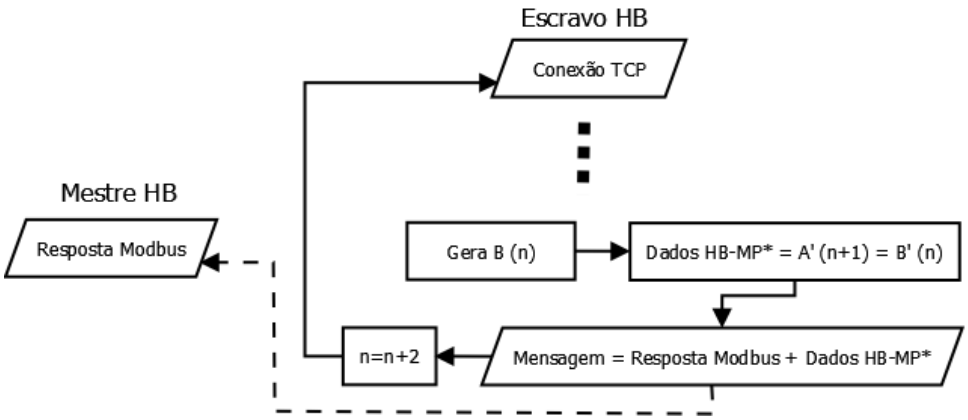

Figura 16 – Fluxograma detalhado das etapas finais de resposta do Escravo HB
Fonte: Elaborada pelo autor

Na sequência, o Mestre HB recebe a mensagem do Escravo HB. Se for uma mensagem indicando erro, o Mestre HB retorna ao estado de conexão. Se a mensagem estiver conforme, os dados HB-MP* são armazenados como $A'(n+1)$ (desafio a ser respondido no próximo ciclo) e como $B'(n)$ (resposta ao desafio enviado). O Mestre HB verifica $B(n)$ e, se for uma resposta incorreta, o Mestre HB deve também verificar se já existe maioria de respostas incorretas para, nesse caso, indicar a não autenticidade do Escravo HB. Como no programa Escravo HB, a indicação é feita apenas após uma quantidade arbitrária de ciclos m. Após a verificação da resposta, a variável n é acrescida de 2 unidades e o Mestre HB retorna ao estado de conexão TCP. Essas etapas realizadas pelo Mestre HB são mostradas no fluxograma detalhado da Figura 17.

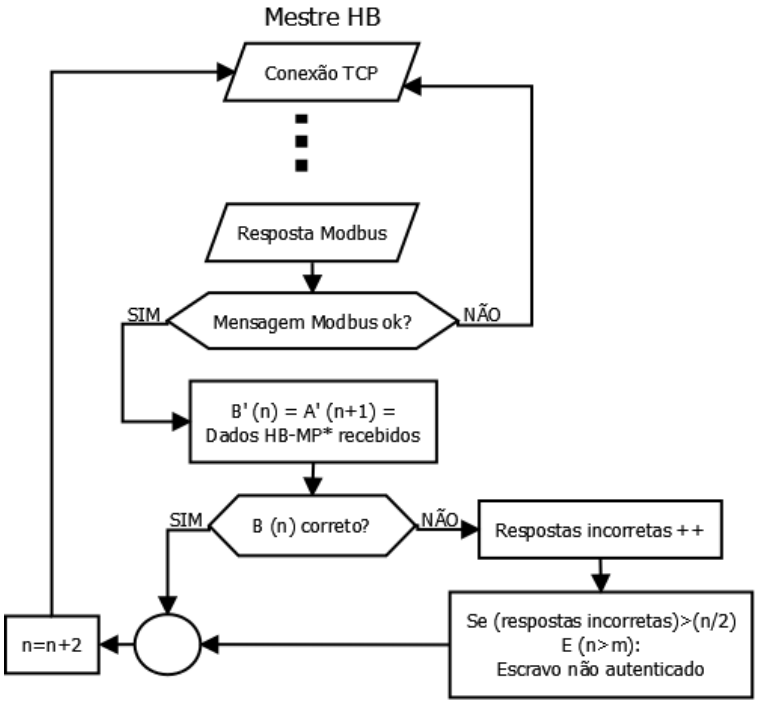

Figura 17 – Fluxograma detalhado das etapas realizadas pelo Mestre HB acerca da mensagem recebida

Fonte: Elaborada pelo autor

A variável n não é incluída em nenhuma das mensagens, mas tem o mesmo valor no Escravo HB e no Mestre HB. Desafios com n ímpar são aqueles enviados pelo Mestre HB, dessa forma o Escravo HB envia respostas com n ímpar. Desafios com n par são enviados pelo Escravo HB, e o Mestre HB envia respostas com n par. O Mestre HB mantém uma variável n para cada Escravo na rede.

Duas versões modificadas do programa Mestre HB foram implementadas para simular os invasores na rede Modbus TCP. A primeira versão não possui as funções do protocolo HB-MP*, realizando apenas a conexão TCP, bem como o envio e o recebimento das mensagens Modbus. Essa versão corresponde a um mestre padrão Modbus TCP. A segunda

versão modificada pode ser descrita pelo mesmo fluxograma da Figura 12, com a diferença da geração dos dados HB-MP* (detalhada na Figura 11 e no fluxograma da Figura 13). Essa versão gera dados HB-MP* aleatoriamente, sem uso da chave ou dos dados enviados pelo Escravo HB. A segunda versão modificada foi feita para simular um Mestre invasor com conhecimento sobre o processo de autenticação, mas sem o conhecimento da chave secreta X.

5.3 Testes

Para validação da proposta do HB Modbus foram definidos alguns testes, divididos em testes de funcionalidade, testes de proteção e testes de desempenho. Os testes de funcionalidade visaram validar a comunicação efetiva dos programas Mestre HB e Escravo HB. O primeiro teste de funcionalidade foi feito com a análise dos pacotes de dados trocados entre os programas Mestre HB e Escravo HB, utilizando o programa Wireshark para "escuta" da rede.

O segundo teste de funcionalidade foi feito para detectar convergência dos dados HB-MP*, uma possibilidade devido ao aproveitamento de respostas HB-MP* como desafios HB-MP*. O teste foi realizado com a análise de 400 pacotes de dados, quantidade definida arbitrariamente. A análise foi realizada pela captura dos pacotes de dados através de interface presente no próprio programa, importação dos dados para uma planilha, e classificação dos dados na planilha com relação à frequência em que aparecem.

O terceiro teste de funcionalidade foi feito entre o Mestre HB e o programa ModbusPal, visando identificar a necessidade de um dispositivo conversor intermediário entre o HB Modbus e o Modbus TCP tradicional. O teste foi feito com a própria interface dos programas Mestre HB e ModbusPal e com o Wireshark.

Os testes de proteção foram realizados através de cenários de ataque à rede, visando apontar as situações nas quais o HB Modbus é uma proteção efetiva e as situações para as quais a técnica é vulnerável. Foram testadas situações com o ModbusPal simulando um Escravo invasor e com modificações manuais feitas diretamente no *prompt* de comando do sistema operacional hospedeiro, a fim de realizar o sequestro TCP e o redirecionamento.

O programa Mestre HB foi modificado, conforme descrito na seção anterior, para realização de testes de proteção em situação de Mestre invasor. Nesse teste, a eficiência da proteção foi mensurada pela quantidade de vezes em que o Mestre invasor respondia corretamente à um desafio HB-MP*. A própria interface do programa forneceu essas informações.

Os testes de desempenho foram realizados visando mensurar o efeito da inclusão da técnica de autenticação HB-MP*, comparando os tempos necessários para a comunicação com autenticação e sem autenticação. Os resultados desses testes são apresentados no próximo Capítulo.

Capítulo 6

Resultados

Nesse capítulo, são discutidos os resultados obtidos a partir dos testes realizados, de acordo com o que foi descrito na Metodologia. Os primeiros testes verificaram se os programas desenvolvidos conseguiriam estabelecer uma comunicação Modbus TCP válida com autenticação. O programa Wireshark foi usado para capturar e analisar os pacotes de dados transmitidos. O Wireshark identificou todos os protocolos envolvidos em uma comunicação Modbus TCP usual e os dados HB-MP* diretamente após os dados Modbus. A função de escrita em registrador, normalmente com 2 bytes de dados Modbus, aparece com 5 bytes no Wireshark. A análise de pacote de dados mostrou o uso correto de MAC, IP e TCP como parte de uma transmissão cíclica de dados típica em uma rede Modbus TCP. Essa captura é mostrada na Figura 18.

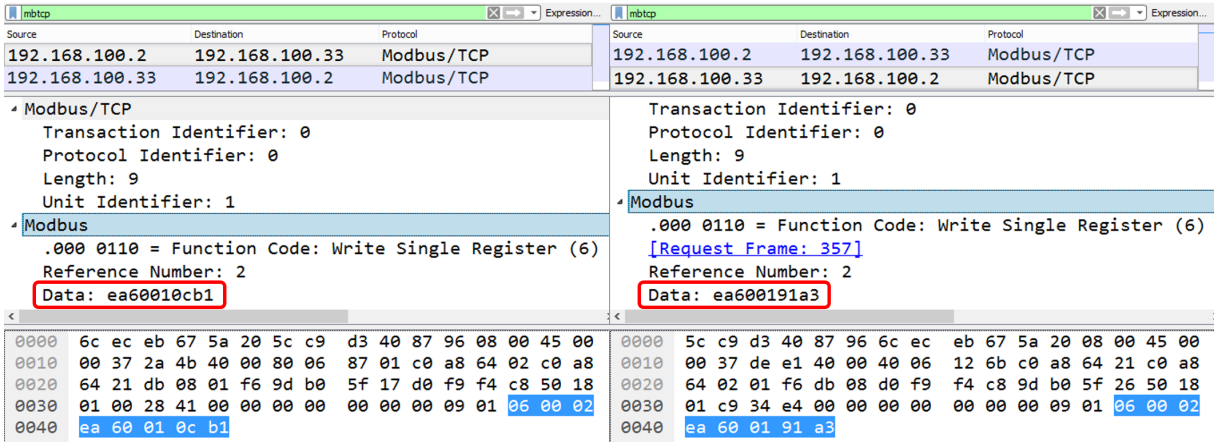

Figura 18 – Captura dos pacotes de dados de Requisição Modbus com desafio HB-MP* e Resposta Modbus com resposta HB-MP*
Fonte: Elaborado pelo autor

Como os dados HB-MP* são considerados tanto como resposta atual quanto como próximo desafio, algumas preocupações precisaram ser resolvidas. A primeira preocupação diz respeito a se os dados HB-MP* convergem para um valor constante após alguns ciclos, possivelmente invalidando a autenticação. A segunda preocupação se refere a se sequências

específicas de bits aparecem regularmente e se tornam previsíveis, prejudicando a proposta de segurança do algoritmo.

Para resolver essas preocupações, 400 pacotes de dados entre um Mestre HB e um Escravo HB foram analisados, dentro dos quais nenhuma convergência foi identificada. Os dados de um ciclo de autenticação para o próximo não se repetiram, nem entre ciclos do Mestre HB ou do Escravo HB, nem entre um ciclo do Mestre HB e um ciclo do Escravo HB. Pesquisando as amostras de dados HB-MP*, descobriu-se que uma combinação de 3 bytes repetiu 8 vezes entre 400 amostras, representando 2%. Outras combinações também apareceram mais de uma vez, e essa frequência (quantas vezes uma combinação apareceu nessas 400 amostras) é mostrada na Fig. 19. No entanto, essas combinações foram repetidas fora de ordem e nunca em sequência e, com isso, nenhum padrão de repetição foi identificado.

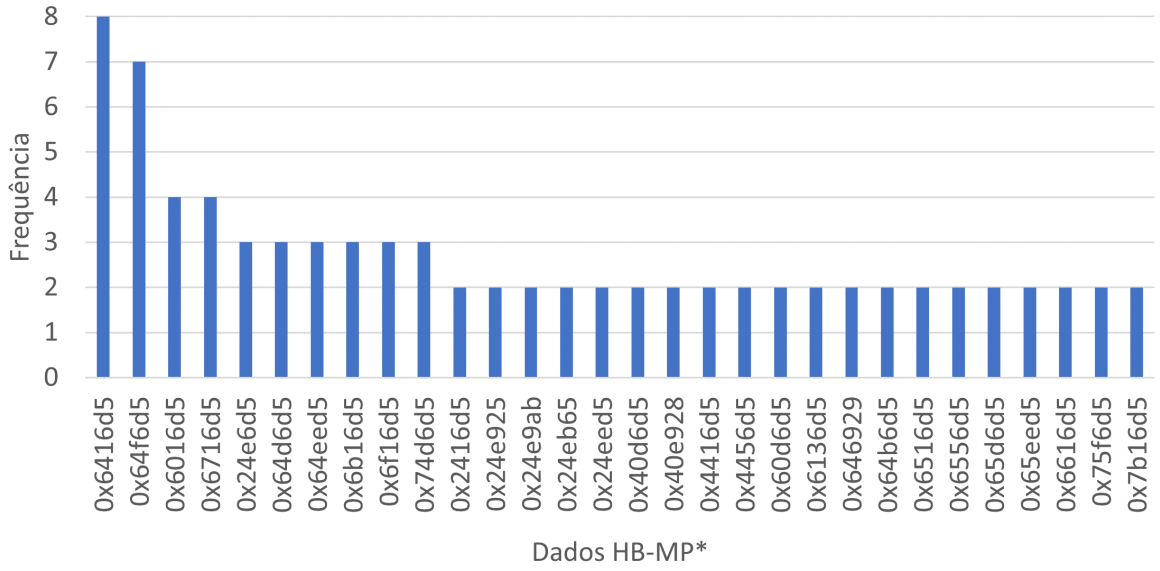

Figura 19 – Dados HB-MP* que mais apareceram em 400 amostras
Fonte: Elaborado pelo autor

Também foram realizados testes para verificar se o Mestre HB poderia se comunicar com Escravos Modbus TCP padrão. Considerando as limitações de hardware para dispositivos físicos Modbus TCP, como controladores e módulos remotos de entradas e saídas, o Escravo Modbus padrão usado foi o programa ModbusPal [1]. Esse teste de comunicação é mostrado na Figura 20, com o traço verde indicando uma comunicação entre dois dispositivos HB Modbus, e o traço vermelho indicando a comunicação entre um Mestre HB e um Escravo Modbus padrão. O teste foi gravado e pode ser visualizado em vídeo, disponível no repositório YouTube pelo link: <https://youtu.be/7tYokIqejZo>.

[1] O ModbusPal é um programa livre de código aberto disponível em: <http://modbuspal.sourceforge.net>

Através da utilização das interfaces do Wireshark e do ModbusPal, verificou-se que as requisições HB Modbus endereçadas ao ModbusPal foram todas aceitas e executadas, independentemente dos dados HB-MP* da requisição, mostrando possível compatibilidade retroativa com Escravos Modbus TCP padrão. O ModbusPal ignorou os dados do HB-MP* e executou a solicitação do Modbus de acordo. Isso ocorreu devido ao fato de que em uma mensagem padrão do Modbus, o código de função também atua como um limitador de campo de dados. No HB Modbus, como a função utilizada foi escrever em um único registrador, e cada registrador possui até 2 bytes, os bytes de dados considerados para o comando foram apenas os dois primeiros. O ModbusPal foi programado para lidar com bytes de dados de acordo com o código de função, portanto, os bytes excedentes foram ignorados. Esse teste confirma que outros códigos de função não interfeririam no processo de autenticação. As respostas geradas pelo ModbusPal eram de natureza padrão, sem dados HB-MP*. O Mestre HB, ao não receber nenhum dado HB-MP* daquele Escravo, indicou sua não autenticidade. Apesar do teste mostrar a compatibilidade, é importante ressaltar que em uma rede mista, os dispositivos Modbus TCP padrão não estariam protegidos dos cenários de ataque citados na próxima seção.

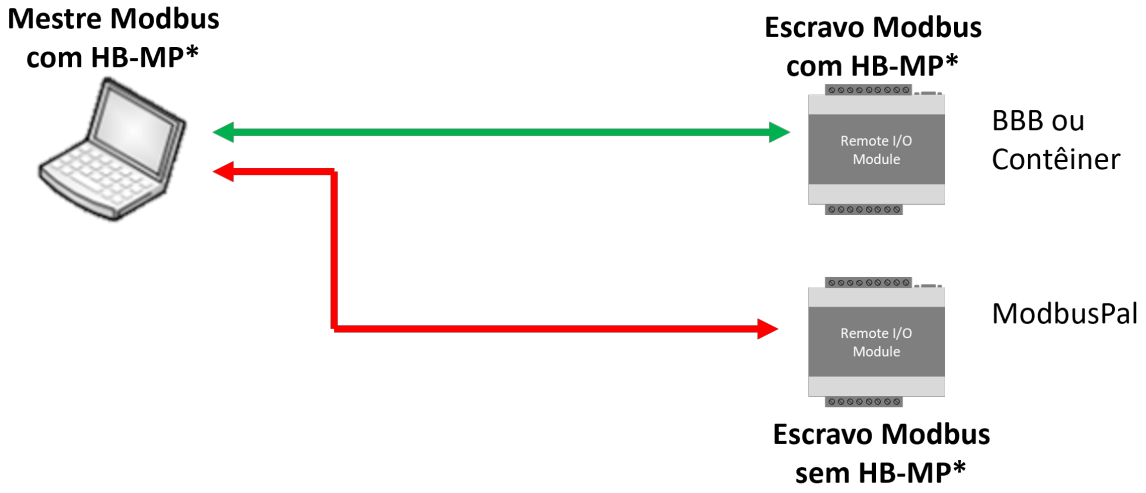

Figura 20 – Diagrama da comunicação HB Modbus com um Escravo HB e um Escravo Modbus padrão

Fonte: Elaborado pelo autor

6.1 Cenários de ataque à rede

Finalizados os testes de funcionalidade, o HB Modbus foi testado quanto à seu impacto efetivo na segurança da rede através de cenários de ataque. A primeira situação testada foi a de um Escravo invasor tentando se comunicar com um Mestre HB. No Modbus TCP, o Mestre sempre toma a iniciativa de conexão, então o Escravo invasor teria que invadir

uma conexão já estabelecida (Sequestro TCP), redirecionar mensagens de requisição para si mesmo (Falsificação do ARP) ou capturar e reutilizar respostas válidas (Ataque de Repetição).

Em termos de sequestro de conexão TCP, nós simulamos a invasão de uma comunicação HB Modbus ocorrendo entre o Mestre HB no PC 1 e o Escravo HB na BBB. O invasor foi uma cópia do programa Escravo HB, no PC 2, modificado para não enviar dados HB-MP*. O PC 2 teve seu endereço de rede reconfigurado (mesmo endereço de rede da BBB), com isso foi possível simular o Sequestro TCP. O invasor enviou sua resposta maliciosa com os identificadores da conexão TCP idênticos aos da resposta legítima do Escravo HB e, no teste realizado, não ocorreu a situação de descarte da resposta legítima, que aconteceria se até o número de sequência fosse idêntico. O que ocorreu foi o Mestre HB recebendo duas respostas, que passaram pela camada de transporte e chegaram à camada de aplicação. A resposta maliciosa, estando sem os dados HB-MP*, foi suficiente para o Mestre HB detectar a invasão.

Ao lidar com o redirecionamento da requisição, esse cenário de ataque foi simulado com o Escravo ModbusPal no PC 2, como um Escravo invasor, redirecionando os pacotes de dados de requisição para ele [2]. De forma similar ao Sequestro TCP, as respostas maliciosas não possuíam dados HB-MP*, portanto a invasão foi detectada pelo Mestre HB. A principal diferença entre o Sequestro TCP e o redirecionamento é que ocorre a negação de serviço no redirecionamento, pois o Escravo HB legítimo torna-se inacessível. Os dois ataques descritos, envolvendo um Escravo invasor, são ilustrados na Figura 21. O Ataque de Repetição executado por um Escravo invasor será discutido posteriormente nesta seção.

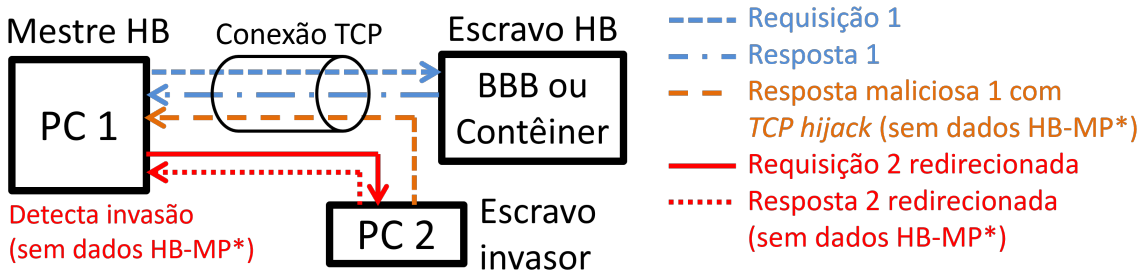

Figura 21 – Escravo invasor enviando respostas ao Mestre HB
Fonte: Elaborado pelo autor

Além de um Escravo invasor, é possível um invasor assumir o papel de Mestre invasor, enviando solicitações Modbus padrão para um Escravo HB. Essa invasão pode ser realizada por sequestro da conexão legítima TCP ou simplesmente com uma nova conexão com um Escravo HB (o HB Modbus não abrange a camada de transporte). Em ambas

[2] O teste foi realizado com redirecionamento manual, não com falsificação ARP.

as situações, ao nível da aplicação, o Escravo HB recebe uma requisição Modbus sem dados HB-MP*, mas não a executa, protegendo assim o sistema de automação industrial de requisições não autenticadas. Nos testes realizados, o Escravo HB não enviou nenhuma resposta e retornou para a espera de nova conexão. Esse cenário é ilustrado pela "requisição maliciosa 1" na Figura 22.

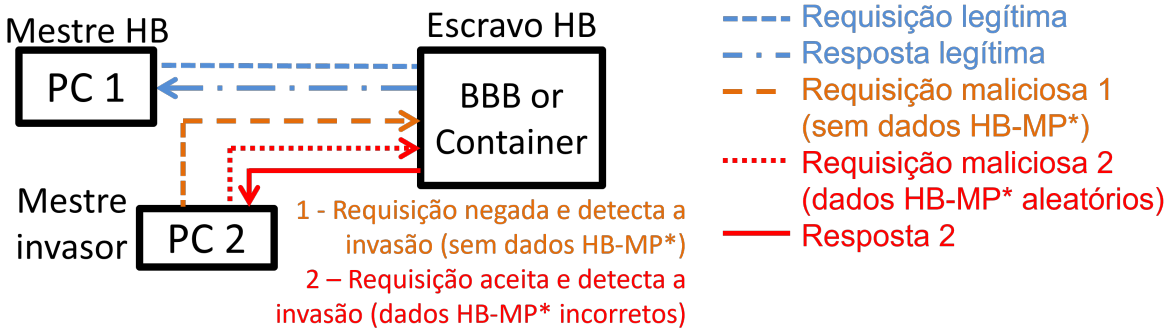

Figura 22 – Mestre invasor enviando requisições ao Escravo HB
Fonte: Elaborado pelo autor

Se um Mestre invasor escuta maliciosamente a rede, ele pode notar que o campo de dados contém mais bytes do que o esperado. O invasor pode inferir que uma técnica de segurança está em vigor e tentar forçar uma autenticação usando três métodos: gerar bytes aleatórios para forçar a combinação correta; tentar identificar padrões repetitivos nos bytes; ou reutilizar os bytes extras que foram identificados.

A autenticação forçada com bytes aleatórios foi testada com a segunda versão modificada do Mestre, e é ilustrada pela "requisição maliciosa 2" na Figura 22. Gerando 3 bytes HB-MP* aleatórios, o Mestre invasor foi incorretamente autenticado 105 vezes em 999 desafios pelo Escravo HB, representando 10,51% de falsos positivos [3]. Este alto valor de falsos positivos pode ser explicado pela condição de autenticação: $B \cdot S = Z$, a qual pode ser satisfeita por mais de uma combinação de bytes B. Mesmo assim, o protocolo HB-MP* se baseia na maioria das respostas erradas e, em nenhum momento, foram identificadas respostas aleatórias corretas em sequência. Ou seja, o Mestre invasor foi detectado pela técnica após duas ou mais interações com o Escravo HB. Nessa situação de ataque, o Mestre invasor recebe resposta às requisições Modbus, mas não tem informações de quando sua resposta HB-MP* foi correta ou incorreta.

Considerando o alto valor de falsos positivos, decidiu-se verificar se um aumento no número de bytes diminuiria essa proporção. No lugar de 3 bytes foram usado 4 bytes para a autenticação em um teste subsequente. O número primo selecionado arbitrariamente para a chave foi 1793161687 ($6AE179D7_{hex}$), necessitando de 31 bits (também primo). O

[3] O resultado é considerado falso positivo quando o dispositivo não é autêntico e, ainda assim, a técnica o identifica como autêntico. De forma similar, um falso negativo é caracterizado por um dispositivo autêntico sendo identificado como não autêntico

mesmo teste de autenticação forçada com bytes aleatórios foi realizado. Com o aumento de bits, a proporção de falsos positivos diminuiu para 6,31%, mostrando melhora na técnica ao utilizar mais bits para a chave e para o processo de autenticação.

Outro meio de forçar a autenticação é quando um invasor tenta identificar padrões no processo de autenticação, afim de repeti-los e incrementar suas chances. Durante os testes, uma certa combinação repetiu 8 vezes em 400 amostras, conforme ilustrado pela Figura 19. Outras combinações também se repetiram, e a maior preocupação foi com o byte `d5`, que apareceu em 135 amostras. Isso pode representar uma chance maior de um invasor forçar a autenticação, o que foi mitigado pelo uso de mais bits.

A última situação de ataque testada foi o Ataque de Repetição realizado por um Mestre invasor e por um Escravo invasor. Os invasores capturaram e reutilizaram mensagens legítimas, mas os testes mostraram que capturar e reenviar dados HB-MP* não foi útil. Para cada mensagem, o desafio HB-MP* muda, então a resposta esperada também mudará. Tanto o Mestre invasor quanto o Escravo invasor são detectáveis no Ataque de Repetição.

O ataque Homem-no-Meio não foi testado devido a limitações de hardware, pois os testes foram realizados em período de pandemia do coronavirus Sars-CoV-2, o que impossibilitou a realização em laboratórios adequados. Entretanto, o ataque Homem-no-Meio caracteriza uma limitação do HB Modbus, considerando que o ataque pode capturar e reutilizar todos os dados contidos no pacote de dados, inclusive os dados HB-MP* corretos. Essa limitação é discutida no capítulo 7.

O quadro 2 resume os ataques descritos nessa seção, concentrando no efeito na camada de aplicação Modbus. É possível destacar que alguns ataques que usam dados HB-MP* aleatórios ou reutilizados têm sucesso parcial, pois a requisição ou a resposta é aceita pelo alvo. Entretanto, mesmo que uma ou mais requisições e repostas sejam aceitas, o invasor é detectado e é possível a tomada de alguma decisão.

6.2 Resultados de desempenho

Sistemas de automação industrial priorizam a eficiência na comunicação, portanto, a inclusão de técnicas de segurança deve ter impacto mínimo no processamento e no tempo de transmissão. Para analisar esse impacto para o HB Modbus, foram realizados testes de comunicação com nós diversos, conforme mostrado na Figura 23. Os testes mostraram que o Mestre HB foi capaz de enviar requisições e obter respostas de dois Escravos HB em dispositivos embarcados BBB (Escravos 1 e 2), um Escravo Modbus padrão ModbusPal (Escravo 3), e quatro Escravos HB funcionando em contêineres virtuais. Os Escravos HB em contêineres possuem o mesmo programa que os BBBs, sendo que a única diferença é que estes possuem registradores simulados, não saídas reais.

O primeiro teste de desempenho visou medir o Tempo de Ida e Volta (*Round Trip Time*

Quadro 2: Resumo dos ataques direcionados à camada de aplicação Modbus e os efeitos do HB Modbus

Objetivo	**Método de ataque**	**Modbus TCP**	**HB Modbus**
Resposta maliciosa	Sequestro TCP (sem dados HB-MP*)	Respostas duplicadas	Respostas duplicadas, Invasão detectada
Resposta maliciosa	Redirecionamento da requisição (sem dados HB-MP*)	Escravo legítimo inacessível, resposta maliciosa aceita	Escravo legítimo inacessível, Invasão detectada
Requisição maliciosa	Requisição Modbus padrão	Requisição maliciosa aceita	Requisição maliciosa negada, Invasão detectada
Requisição maliciosa	Requisição maliciosa com autenticação forçada	Requisição maliciosa aceita	Requisição maliciosa aceita, Invasão detectada
Requisição / resposta maliciosa	Homem-no-Meio	Ataque bem sucedido	Ataque bem sucedido com reutilização dos dados HB-MP*
Requisição / resposta maliciosa	Ataque de Repetição com dados HB-MP*	Ataque bem sucedido	Requisição / resposta maliciosa aceita, Invasão detectada

Fonte: Elaborado pelo autor

(RTT)) entre o envio de uma requisição e o recebimento de uma resposta de um dos BBB e do ModbusPal. Este intervalo de tempo corresponde ao atraso de transmissão pelo Mestre HB, acrescido do atraso de propagação, do atraso de processamento pelo Escravo HB ou pelo ModbusPal e do atraso de transmissão pelo Escravo HB ou pelo ModbusPal. Por meio dessa medida, foi estimada a influência do processo de autenticação.

O teste resultou em RTT médio em 200 ciclos com autenticação de 2,23 ms e RTT médio sem autenticação de 1,56 ms, com o mesmo número de ciclos. A diferença entre os RTTs médios fornece uma estimativa do impacto da autenticação sobre os atrasos de processamento e transmissão dos dados HB-MP*, que foi aproximadamente 0,67 ms (representando um aumento de 42%). O aumento proporcional foi considerável, mas o aumento do tempo em si, não. A literatura considera que RTTs abaixo de 10 ms são comuns em redes industriais aplicadas em controle de processo, manufatura discreta, automação predial e distribuição de utilidades. Uma exceção que requer RTTs mais baixos é o controle de movimento, mas para essa aplicação mesmo o Modbus TCP padrão não é adequado (GALLOWAY; HANCKE, 2013; REYNDERS; MACKAY; WRIGHT, 2004; PROFIBUS INTERNATIONAL, 2014).

O tempo total do ciclo também foi medido para a rede mostrada na Figura 23. Todos os 8 Escravos receberam uma requisição e responderam uma vez em cada ciclo. Ao medir 30 tempos totais de ciclo, a variação encontrada foi entre 748 ms e 1204 ms, com média de 865 ms. Esta não é uma soma direta dos RTTs médios (2,23 ms e 1,56 ms), pois o tempo

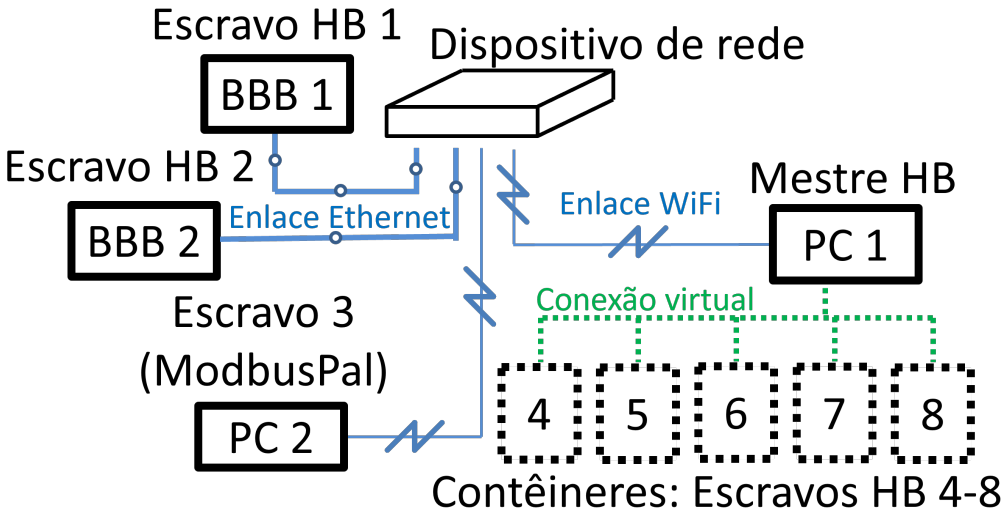

Figura 23 – Rede HB Modbus estabelecida
Fonte: Elaborado pelo autor

total do ciclo inclui todas as etapas mostradas na Figura 12 para cada Escravo. O ciclo completo também mostrou que o Mestre HB pode se comunicar com vários Escravos com arquiteturas distintas (Escravos HB embarcados e em contêineres e um Escravo Modbus padrão). Os tempos medidos estão ilustrados na Figura 24, na qual as requisições e respostas ilustradas contêm os dados HB-MP*, exceto para a resposta do ModbusPal.

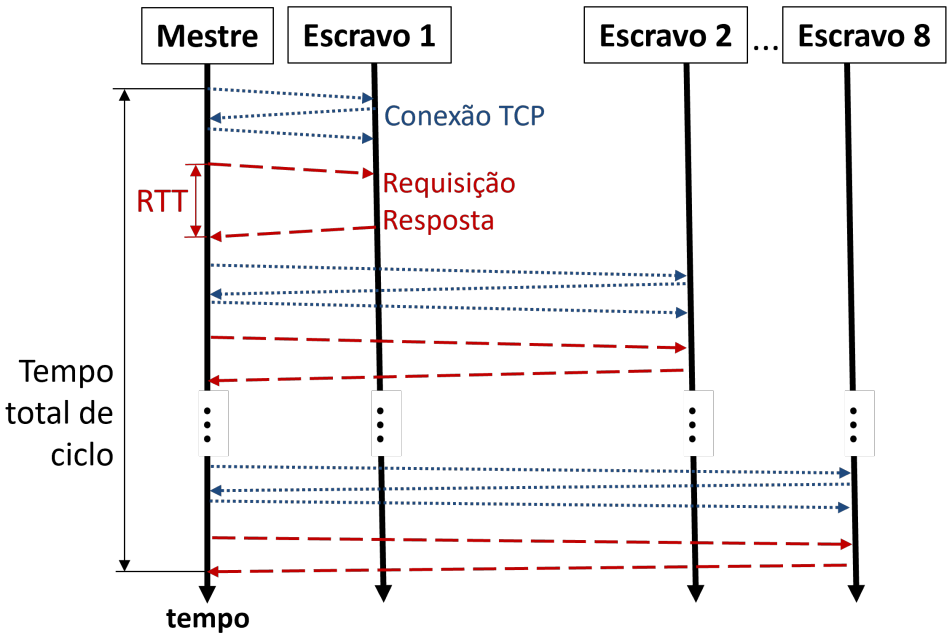

Figura 24 – RTT e tempo total de ciclo no HB Modbus
Fonte: Elaborado pelo autor

Capítulo 7

Limitações e comparação

As características e limitações do HB Modbus foram identificadas através dos testes. Este capítulo irá enumerá-las e comparar o HB Modbus com as técnicas de segurança desenvolvidas especificamente para aplicações Modbus.

A primeira limitação que chama atenção é a alta taxa de falsos positivos, cuja provável causa é também uma das características chave do algoritmo: sua simplicidade. Aumentando o número de bytes houve redução nessa proporção, mas a limitação foi apenas mitigada, não eliminada. Uma segunda limitação identificada diz respeito à vulnerabilidade do HB Modbus a ataques de Negação de Serviço causados por um Mestre invasor, seja com solicitações de conexão TCP exaustivas para o Escravo HB, seja com o comando TCP *Reset*. Ambas vulnerabilidades estão na camada de transporte, que o HB Modbus não cobre.

O HB Modbus também é vulnerável a ataques Homem-no-Meio. Por meio de Falsificação da tabela ARP, um invasor pode interceptar solicitações do Mestre HB e/ou respostas do Escravo HB. A solicitação ou resposta legítima contém a resposta ao desafio HB-MP* anterior, assim o invasor pode alterar o conteúdo do pacote de dados (por exemplo, código de função e dados Modbus) antes de direcioná-lo para o destino legítimo. A proteção contra tal situação não é possível com o HB Modbus.

As soluções de segurança baseadas no comportamento do tráfego, como as propostas por Goldenberg e Wool (2013), Fovino et al. (2012) e Sestito et al. (2018), podem ser categorizadas como *firewalls*, e não como protocolos de segurança. Tais soluções podem ser aplicadas junto ao HB Modbus, pois os *firewalls* não modificam a pilha de protocolos. No entanto, a proposta de Goldenberg e Wool (2013) funcionará apenas se as mensagens Modbus seguirem um padrão cíclico, portanto, pode não ser satisfatória em redes com uma ampla gama de mensagens trocadas entre os nós.

Diferentemente do Secure Modbus proposto por Fovino et al. (2009), o HB Modbus não inclui verificação de integridade, pois essa verificação já é fornecida pela soma de verificação executada pelo TCP e pela verificação redundante cíclica executada pelo padrão Ethernet. A autenticação é fornecida pelo Secure Modbus e pelo HB Modbus, já o não re-

púdio é fornecido apenas parcialmente pelo HB Modbus, visto que alguns ataques são bem sucedidos (embora detectáveis) e podem sobrescrever uma mensagem legítima. Devido à criptografia presente no Secure Modbus, é possível evitar o ataque Homem-no-Meio, descrito como uma limitação do HB Modbus.

O HB Modbus é mais compacto e simples, adicionando 3 bytes ou mais contra 32 bytes do Secure Modbus e realizando apenas operações binárias. Essa simplicidade torna o HB Modbus leve, considerando aspectos como processamento, atrasos na comunicação e percentual de carga de dados de aplicação nos pacotes de dados, mas também o torna menos eficaz, conforme demonstrado pelo índice de falsos positivos e pelas limitações. Os ataques de repetição são evitados em ambas as soluções: no Secure Modbus, por um carimbo de tempo, e no HB Modbus, devido à singularidade de cada combinação de dados HB-MP*.

Tanto o HB Modbus quanto o Secure Modbus são vulneráveis contra um agente malicioso que controle um Mestre ou Escravo genuíno. Outra importante diferença encontrada é que o Mestre HB Modbus foi capaz de se comunicar com um Escravo Modbus TCP padrão, mostrando possível compatibilidade com versões padronizadas do Modbus TCP sem a necessidade de um conversor. Entretanto, essa afirmação carece de testes com outros dispositivos Modbus TCP padrão diferentes do ModbusPal. Além disso, como o HB Modbus é baseado em uma chave comum pré-compartilhada, um invasor que capture a chave e saiba sobre o funcionamento do algoritmo seria capaz de passar pela defesa.

A comparação feita entre o HB Modbus e o Modbus Security apresenta resultados semelhantes à comparação feita com o Secure Modbus. Portanto, o HB Modbus apresenta-se como uma alternativa mais simples, porém menos eficiente, tanto quando comparado com o Modbus Security, como quando comparado com o Secure Modbus. Além disso, o Modbus Security possui ferramentas para proteção contra Negação de Serviço causada por vulnerabilidades da camada de transporte. Além da eficiência, foram feitas algumas comparações de desempenho. Os testes realizados por Ferst et al. (2018) mostraram que o atraso no estabelecimento de uma conexão TLS no Modbus Security foi até mil vezes maior que o atraso no estabelecimento de uma conexão TCP no Modbus TCP padrão. A troca segura de chaves e o aperto de mãos criptografado (TLS *handshake*), além de outras mensagens de segurança, são possíveis motivos para esse atraso (JINGRAN et al., 2020). O HB Modbus não altera a conexão TCP, altera apenas o protocolo da camada de aplicação, portanto, não há aumento do tempo necessário para conexão.

Após estabelecer a conexão, conforme Ferst et al. (2018), o RTT com Modbus Security variou entre 1,3 ms e 2,0 ms, e com Modbus TCP o RTT médio foi de 1,0 ms. Isso representa um aumento de 30% ou mais para pacotes TLS mais simples e até 100% para pacotes mais completos. No HB Modbus, apresentou-se um aumento de cerca de 42%. Outra questão é que, assim como o Secure Modbus, o Modbus Security também requer um conversor para acessar equipamentos Modbus TCP padrão.

As comparações entre o HB Modbus, o Secure Modbus e o Modbus Security são reunidas e resumidas no Quadro 3. As informações contidas na coluna referente ao Secure Modbus têm como fonte o trabalho de Fovino et al. (2009), e as informações na coluna referente ao Modbus Security têm como fonte a publicação oficial MODBUS ORG. (2018) e o trabalho de Ferst et al. (2018).

Quadro 3: Comparativo entre o HB Modbus e os principais estudos relacionados

Técnica / característica	**HB Modbus**	**Secure Modbus**	**Modbus Security**
Acréscimo no pacote de dados	3 bytes ou mais	32 bytes	21 a 73 bytes
Acréscimo no tempo	42%	Não mensurado	30 a 100% no RTT e mais de 1000% na conexão
Verificação de integridade	Provida pelo TCP	Incluída	Provida pelo TCP
Autenticação	Presente	Presente	Presente
Limitações na proteção	Falsificação ARP; TCP RST; Homem-no-Meio; Ataques detectáveis, mas não prevenidos.	Falsificação ARP; TCP RST.	Falsificação ARP; TCP RST.
Diferencial	Compacto; Aplicável com outras técnicas em outras camadas.		Proteção ampla

Fonte: Elaborado pelo autor, com dados de Fovino et al. (2009), MODBUS ORG. (2018) e Ferst et al. (2018)

.

Constata-se que, se as limitações de processamento são uma realidade em um sistema de automação industrial, o HB Modbus pode ser uma solução de segurança satisfatória com baixo custo de processamento e transmissão. Além disso, ele pode ser combinado com outras propostas de segurança, principalmente aquelas que não requerem processamento extra dos nós, como *firewalls* internos e comutadores protegidos. Essas comparações denotam que a proposta pode ser aplicada a sistemas com processamento limitado. Assim, ressaltamos que a proposta HB Modbus não visa substituir ou concorrer com o Modbus Security, mas pode ser uma alternativa para algumas indústrias e laboratórios de automação.

Capítulo 8

Conclusão

A atual integração dos sistemas de automação industrial os torna sujeitos a ataques e intrusões não previstos quando a maioria das redes industriais foi desenvolvida. Com isso, a segurança é uma preocupação crescente, e o Modbus TCP é um exemplo de tecnologia de comunicação amplamente utilizada e especialmente vulnerável. Esta tese apresentou um estudo sobre vulnerabilidades de segurança do Modbus TCP e propôs a implementação de uma técnica de autenticação leve na camada de aplicação.

O protocolo de autenticação HB-MP* foi a técnica selecionada, desenvolvida para exigir baixo poder de processamento, pouca memória e representar baixo acréscimo nos pacotes de dados. Essas características são relevantes quando existem restrições de tempo e limitações de processamento em alguns sistemas de automação industrial.

A implementação foi denominada HB Modbus. Os testes demonstraram alguns ataques à rede, e se foi possível evitar ou detectar a invasão, bem como as limitações associadas a esses testes. O HB Modbus também foi comparado com outras soluções de segurança Modbus. A principal contribuição desta tese é uma proteção leve contra situações indesejadas em uma rede Modbus TCP, tais como requisições falsas e leituras não autorizadas em Escravos Modbus. Embora essas contribuições já sejam providas por outras técnicas, elas são relevantes devido à simplicidade e facilidade de aplicação do HB Modbus. Essas duas características tornam o HB Modbus adequado para sistemas de automação industrial limitados, ou para uso do HB Modbus com outras técnicas de segurança.

Estudos futuros devem incluir a criptografia no HB Modbus, pois a criptografia é uma ferramenta confiável e comprovadamente eficaz para as limitações identificadas no HB Modbus, principalmente para o ataque Homem-no-Meio. A criptografia será um desafio notável, pois uma das principais características do HB Modbus são seus requisitos de processamento leves.

Um resultado não contemplado aqui é que, como a implementação foi realizada diretamente no protocolo Modbus, o processo de autenticação poderia ser incluído em outras redes Modbus, como RTU e ASCII. Estudos futuros poderão ser desenvolvidos para estas

aplicações, mas não se limitando a elas. Além do Modbus TCP, outras redes não incluem técnicas de segurança nativas, portanto, também podem se beneficiar da proposta aqui apresentada.

8.1 Contribuições em Produção Bibliográfica

Em fase de projeto, este trabalho foi apresentada na 2018 13th IEEE International Conference on Industry Applications (INDUSCON), realizada na cidade de São Paulo, em Novembro de 2018. Já concluída, a tese foi submetida ao periódico Journal of Control, Automation and Electrical Systems e aceita para publicação em 28 de Dezembro de 2021, com o título "Industrial Network Security: HB-MP* as an Authentication Technique for Modbus TCP". O artigo está disponível em: <https://doi.org/10.1007/s40313-021-00889-5>.

Referências

ÅKERBERG, J.; BJÖRKMAN, M. Exploring security in Profinet IO. In: IEEE. **33rd Annual IEEE International Computer Software and Applications Conference**. Seattle, EUA, 2009. v. 1, p. 406–412. <https://doi.org/10.1109/COMPSAC.2009.61>.

ALBUQUERQUE, P. U. B. d.; ALEXANDRIA, A. R. d. **Redes industriais: aplicações em sistemas digitais de controle distribuído**. 2. ed. São Paulo: Editora Ensino Profissional, 2009.

ALIZAI, Z. A. et al. Key-based cookie-less session management framework for application layer security. **IEEE Access**, v. 7, p. 128544–128554, 2019. <https://doi.org/10.1109/ACCESS.2019.2940331>.

ALOTAIBI, A. M. et al. Security issues in protocols of TCP/IP model at layers level. **International Journal of Computer Networks and Communications Security**, v. 5, n. 5, p. 96–104, 2017. <http://www.ijcncs.org/published/volume5/issue5/p2_5-5.pdf>.

ALVES, J. L. L. **Instrumentação, Controle e Automação de Processos .** 2. ed. Rio de Janeiro: Grupo Gen-LTC, 2010.

ASEERI, A. et al. Achieving protection against man-in-the-middle attack in HB family protocols implemented in RFID tags. **International Journal of Pervasive Computing and Communications**, Emerald Group Publishing Limited, v. 12, n. 3, p. 375–390, 2016. <https://doi.org/10.1108/IJPCC-03-2016-0015>.

BELDEN. **Belden Online Catalog**. Online: Belden, 2022. <https://catalog.belden.com/>.

BERGE, J. **Fieldbuses for Process Control: Engineering, Operation, and Maintenance**. Durham, EUA: ISA USA, 2002. v. 8.

BHAIJI, Y. **Network security technologies and solutions: CCIE professional development series**. Indianapolis, EUA: Pearson Education, 2008.

BHUYAN, M. H.; BHATTACHARYYA, D. K.; KALITA, J. K. Network anomaly detection: Methods, systems and tools. **IEEE Communications Surveys Tutorials**, v. 16, n. 1, p. 303–336, 2014. <https://doi.org/10.1109/SURV.2013.052213.00046>.

BOYER, S. A. **SCADA: supervisory control and data acquisition**. 3. ed. Durham, EUA: The Instrumentation, Systems and Automation Society, 2004.

CHEMINOD, M. et al. Performance evaluation and modeling of an industrial application-layer firewall. **IEEE Transactions on Industrial Informatics**, v. 14, n. 5, p. 2159–2170, 2018. <https://doi.org/10.1109/TII.2018.2802903>.

CHEMINOD, M.; DURANTE, L.; VALENZANO, A. Review of security issues in industrial networks. **IEEE Transactions on Industrial Informatics**, IEEE, v. 9, n. 1, p. 277–293, 2013. <https://doi.org/10.1109/TII.2012.2198666>.

CHEN, T. Stuxnet, the real start of cyber warfare? [editor's note]. **IEEE Network**, v. 24, n. 6, p. 2–3, 2010. <https://doi.org/10.1109/MNET.2010.5634434>.

DU, W. **Computer Internet Security: A Hands-on Approach**. Syracuse, EUA: Createspace Independent Publishing Platform, 2019.

DUDAK, J. et al. Serial communication protocol with enhanced properties–securing communication layer for smart sensors applications. **IEEE Sensors Journal**, v. 19, n. 1, p. 378–390, 2019. <https://doi.org/10.1109/JSEN.2018.2874898>.

FACHKHA, C. Cyber threat investigation of scada modbus activities. In: **2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)**. [S.l.: s.n.], 2019. p. 1–7. <https://doi.org/10.1109/NTMS.2019.8763817>.

FERST, M. K. et al. Implementation of secure communication with Modbus and transport layer security protocols. In: IEEE. **13th IEEE International Conference on Industry Applications (INDUSCON)**. São Paulo, 2018. p. 155–162. <https://doi.org/10.1109/INDUSCON.2018.8627306>.

FOVINO, I. et al. Design and implementation of a Secure Modbus Protocol. In: **Critical Infrastructure Protection III**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 83–96. ISBN 978-3-642-04798-5. <https://doi.org/10.1007/978-3-642-04798-5_6>.

FOVINO, I. N. et al. Critical state-based filtering system for securing scada network protocols. **IEEE Transactions on Industrial Electronics**, v. 59, n. 10, p. 3943–3950, 2012. <https://doi.org/10.1109/TIE.2011.2181132>.

GALLOWAY, B.; HANCKE, G. P. Introduction to industrial control networks. **IEEE Communications Surveys & Tutorials**, v. 15, n. 2, p. 860–880, 2013. <https://doi.org/10.1109/SURV.2012.071812.00124>.

GOLDENBERG, N.; WOOL, A. Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems. **International Journal of Critical Infrastructure Protection**, v. 6, p. 63–75, 06 2013. <https://doi.org/10.1016/j.ijcip.2013.05.001>.

HAYES, G.; EL-KHATIB, K. Securing Modbus transactions using hash-based message authentication codes and stream transmission control protocol. In: IEEE. **Third International Conference on Communications and Information Technology (ICCIT)**. Beirut, Líbano, 2013. p. 179–184. <https://doi.org/10.1109/ICCITechnology.2013.6579545>.

HOPPER, N. J.; BLUM, M. Secure human identification protocols. In: BOYD, C. (Ed.). **Advances in Cryptology — ASIACRYPT 2001**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. p. 52–66. ISBN 978-3-540-45682-7. <https://doi.org/10.1007/3-540-45682-1_4>.

HUITSING, P. et al. Attack taxonomies for the modbus protocols. **International Journal of Critical Infrastructure Protection**, Elsevier, v. 1, p. 37–44, 2008. <https://doi.org/10.1016/j.ijcip.2008.08.003>.

IEC. **Undestanding IEC 62443**. Online: IEC, 2021. <https://www.iec.ch/blog/understanding-iec-62443>. Acessado em 11 agosto 2022.

IEEE, C. S. **IEEE 802.3-2018: Standard for Ethernet**. Nova Iorque, EUA: IEEE, 2018.

JINGRAN, W. et al. Research and implementation of secure industrial communication protocols. In: **2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS)**. [S.l.: s.n.], 2020. p. 314–317. <https://doi.org/10.1109/ICAIIS49377.2020.9194854>.

JUELS, A.; WEIS, S. A. et al. Authenticating pervasive devices with human protocols. In: SPRINGER. **Annual International Cryptology Conference**. Santa Barbara, EUA, 2005. p. 293–308. <https://doi.org/10.1007/11535218_18>.

KUROSE, J. F.; ROSS, K. **Redes de Computadores: uma abordagem top-down**. 6. ed. São Paulo: Pearson Education do Brasil, 2014.

KUSHNER, D. The real story of stuxnet. **IEEE Spectrum**, v. 50, n. 3, p. 48–53, 2013. <https://doi.org/10.1109/MSPEC.2013.6471059>.

LORENZO, S.; BENITO, J. A.; ARRIZABALAGA, S. Modbus access control system based on SSI over hyperledger fabric blockchain. **Sensors**, v. 21, n. 16, 2021. ISSN 1424-8220. <https://doi.org/10.3390/s21165438>.

MODBUS ORG. **Modbus application protocol specification v1.1b3**. Hopkinton, EUA: Modbus Organization, 2012. <http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf>.

_____. **Modbus/TCP Security: Protocol Specification**. Hopkinton, EUA: Modbus Organization, 2018. <http://modbus.org/docs/MB-TCP-Security-v21_2018-07-24.pdf>.

MORAES, C. C. d.; CASTRUCCI, P. d. L. **Engenharia de automação industrial**. Rio de Janeiro: Grupo Gen-LTC, 2001. 295 p.

MOXA. **MGate MB3280 Series Quick Installation Guide**. Online: Moxa, 2021. <https://cdn-cms.azureedge.net/getmedia/b00c7567-c6b3-484f-93a0-acd78fed5174/moxa-mgate-mb3180-mb3280-mb3480-series-mgate-mb3280-model-qig-v4.2.pdf>.

MUNILLA, J.; PEINADO, A. HB-MP: A further step in the HB-family of lightweight authentication protocols. **Computer Networks**, Elsevier, v. 51, n. 9, p. 2262–2267, 2007. <https://doi.org/10.1016/j.comnet.2007.01.011>.

NARAYANASWAMY, S.; KUMAR, A. Application layer security authentication protocols for the internet of things: A survey. **Advances in Science, Technology and Engineering Systems Journal**, v. 4, n. 1, p. 317–328, 2019. <https://doi.org/10.25046/aj040131>.

NIDEBORN, J. **Industrial network market shares 2021 according to HMS Networks**. Online: HMS, 2021. <https://www.hms-networks.com/news-and-insights/news-from-hms/2021/03/31/continued-growth-for-industrial-networks-despite-pandemic>. Acessado em 05 Dezembro 2021.

PROFIBUS INTERNATIONAL. **Profinet Real-Time Communication**. [S.l.]: PROFIBUS International, 2014. <http://www.profibus.org.pl/index.php?option=com_docman&task=doc_view&gid=28>.

PéREZ, S. et al. Application layer key establishment for end-to-end security in iot. **IEEE Internet of Things Journal**, v. 7, n. 3, p. 2117–2128, 2020. <https://doi.org/10.1109/JIOT.2019.2959428>.

RESCORLA, E. RFC 8446. **The transport layer security (TLS) protocol version 1.3**, Internet Engineering Task Force (IETF), 2018. <https://tools.ietf.org/html/rfc8446>.

REYNDERS, D.; MACKAY, S.; WRIGHT, E. **Practical industrial data communications: best practice techniques**. Oxford, Reino Unido: Elsevier, 2004.

SCHLEGEL, R.; OBERMEIER, S.; SCHNEIDER, J. A security evaluation of IEC 62351. **Journal of Information Security and Applications**, v. 34, p. 197 – 204, 2017. ISSN 2214-2126. Disponível em: <https://doi.org/10.1016/j.jisa.2016.05.007>.

SEN, S. K. **Fieldbus and Networking in Process Automation**. Boca Raton, EUA: CRC Press, 2014.

SESTITO, G. S. et al. A method for anomalies detection in real-time ethernet data traffic applied to profinet. **IEEE Transactions on Industrial Informatics**, v. 14, n. 5, p. 2171–2180, 2018. <https://doi.org/10.1109/TII.2017.2772082>.

TIA. **Commercial building telecommunications cabling standard set: TIA-568 set.** [S.l.]: Telecommunications Industry Association, 2018.

VALENZANO, A. Industrial cybersecurity: Improving security through access control policy models. **IEEE Industrial Electronics Magazine**, v. 8, n. 2, p. 6–17, 2014. <https://doi.org/10.1109/MIE.2014.2311313>.

VOLKOVA, A. et al. Security challenges in control network protocols: A survey. **IEEE Communications Surveys & Tutorials**, IEEE, v. 21, n. 1, p. 619–639, 2019. <https://doi.org/10.1109/COMST.2018.2872114>.

ZHAO, D. et al. A dynamic event-triggered approach to observer-based pid security control subject to deception attacks. **Automatica**, v. 120, p. 109128, 2020. ISSN 0005-1098. <https://doi.org/10.1016/j.automatica.2020.109128>.