
Pós-Graduação em Ciência da Computação

Ubiratan Alves do Carmo

# Ilha de Segurança para Redes de Automação em Sistemas Elétricos


Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

RECIFE

2017

Ubiratan Alves do Carmo

# Ilha de Segurança para Redes de Automação em Sistemas Elétricos

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Orientador: Dra. Judith Kelner

RECIFE
2017

Catalogação na fonte
Bibliotecária Elaine Cristina de Freitas CRB 4-1790

C287i Carmo, Ubiratan Alves do
Ilha de Segurança para Redes de Automação em Sistemas Elétricos / Ubiratan Alves do Carmo . – 2017.
153 f.: fig., tab.

Orientadora: Judith Kelner
Tese (Doutorado) – Universidade Federal de Pernambuco. Cin. Ciência da Computação. Recife, 2017.
Inclui referências e apêndices

1. Redes de Computadores 2. Segurança Cibernética. 3. Redes de Automação. I. Kelner, Judith. (orientadora). II. Título

004.65 CDD (22. ed.) UFPE-MEI 2018-16

**Ubiratan Alves do Carmo**

**Ilha de Segurança para Redes de Automação de Sistemas Elétricos**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutora em Ciência da Computação

Aprovado em: 14/03/2017.

_____

**Orientador: Profa. Dra. Judith Kelner**

**BANCA EXAMINADORA**

_____

Prof. Dr. José Augusto Suruagy Monteiro
Centro de Informática / UFPE

_____

Prof. Dr. Benemar Alencar de Souza
Departamento de Engenharia Elétrica / UFCG

_____

Prof. Dr. Eduardo Luzeiro Feitosa
Instituto de Computação / UFAM

_____

Profa. Dra. Rossana Maria de Castro Andrade
Departamento de Computação / UFC

_____

Prof. Dr. Danilo Ricardo Barbosa de Araújo
Departamento de Estatística e Informática / UFRPE

*Dedico esse trabalho a toda a minha família, amigos e professores que me deram o apoio necessário para chegar aqui.*

# Agradecimentos

Primeiramente, agradeço a Deus pela saúde a mim concedida e pela força que tenho recebido para superar os obstáculos em minha caminhada para atingir o objetivo de conclusão desta atividade acadêmica.

Agradeço à minha família e aos meus filhos, principalmente, pelo apoio e a compreensão de minhas ausências durante o período em que dediquei meu tempo a esta pesquisa.

Agradeço à professora Dra. Judith Kelner que, com sua simplicidade e carisma, pôde me guiar durante todo meu percurso nesta jornada, sempre com a compreensão de minhas limitações e fragilidades, de forma a poder alcançar os objetivos traçados.

Agradeço também ao Dr. Rafael Aschoff pelo seu companheirismo e incentivo dado durante a coorientação a este trabalho.

A todos da equipe do laboratório GPRT, os quais não apenas se dedicaram a me ajudar nos problemas encontrados, mas também se tornaram amigos e companheiros de trabalho. Eu os quero ter sempre presentes ao longo de minha vida profissional.

Ao Dr. Djamel Sadok pela oportunidade dos trabalhos paralelos realizados e pelo enriquecimento do meu conhecimento através de discussões diversas na área de ciência da computação. Foi uma grande satisfação ter convivido durante este período e saber que podia contar com uma fonte de conhecimento que sempre contribuiu para que minhas atividades acadêmicas fossem desempenhadas com sucesso.

Um agradecimento especial a equipe do projeto SIRCAM, que tanto se dispôs a colaborar na solução dos problemas encontrados ao decorrer das atividades, para que o êxito desta Tese fosse atingido.

À Equipe de design do GRVM, que tanto me apoiou com todo suporte possível para o sucesso das minhas apresentações e documentação audiovisual deste projeto.

A todos os professores membros da pós-graduação do curso de ciência da computação do Centro de Informática que contribuíram para minha formação com a disponibilidade de informação e conhecimento.

Agradeço à Companhia Hidro Elétrica do São Francisco pelo suporte dado para as minhas atividades nesta pesquisa e a disponibilização do projeto SIRCAM para realizar os experimentos necessários para o sucesso deste trabalho.

A todos os meus colegas da UFPE e da CHESF, que acreditaram no sucesso desta jornada, sempre confiantes na minha dedicação e capacidade de trabalho.

A todo o staff do GPRT e GRVM que, indiretamente ajudou, bastante no desenvolvi-

mento desta pesquisa, tornando os dias de trabalho no laboratório mais descontraídos e fornecendo todo o suporte necessário para que as atividades fossem realizadas.

*"Eu acredito, que às vezes são as pessoas que ninguém espera nada que fazem as coisas que ninguém consegue imaginar."*

*(Alan Turing)*

# Resumo

Até a década de 1970, o controle e a automação dos sistemas críticos eram baseados em dispositivos eletromecânicos conectados com fios, normalmente estáticos e sem nenhuma flexibilidade. Com a introdução dos dispositivos eletrônicos e das redes de computadores, estes dispositivos adotaram novas tecnologias, aliadas à filosofia de software e hardware proprietários. A desregulamentação do setor elétrico associada à pressão da concorrência do mercado, obrigaram as empresas produtoras e transmissoras de energia elétrica a buscarem projetos de automação cada vez mais competitivos. Esse contexto obrigou os fabricantes a abandonar os sistemas proprietários e adotar padrões abertos, economicamente mais atraentes. Consequentemente, as redes de automação e controle começaram a operar com sistemas operacionais conhecidos, baseados em tecnologias abertas e redes de comunicação Ethernet. No entanto, isso tornou essas redes mais vulneráveis a ataques cibernéticos. Além disso, o crescimento dos grupos terroristas no mundo aumentou a preocupação dos governantes com a vulnerabilidade da infraestrutura crítica. À medida que as interrupções do fornecimento de energia elétrica afetam todos os setores da infraestrutura crítica, os sistemas de automação e controle deste setor tornaram-se um alvo potencial de um possível ataque cibernético. Esta Tese apresenta um modelo conceitual, referenciado, como ilha de segurança da informação para ser aplicado às redes de automação dos sistemas de produção e transmissão de energia elétrica. Este modelo é fundamentado no conceito de controle de acesso baseado em políticas (RBAC), infraestrutura de chave pública (ICP) e hardware seguro. O controle de acesso definido no modelo engloba os computadores que pertencem à rede dos sistemas de automação, bem como os usuários que desejam acessar essa rede. A ideia por trás da ilha de segurança foi inspirada nos castelos medievais, para criar uma área (perímetro) ou domínio controlado que pode ser acessado através de uma única porta. Outro aspecto que contribuiu para realizar este agrupamento lógico é a heterogeneidade das tecnologias existentes nas redes de automação. O perímetro pode ser definido como uma parte ou por toda rede de automação de uma subestação, usinas e centro de controle de sistema. Para validar o modelo da ilha de segurança foram idealizados casos de uso. Os cenários de validação foram implementados para contemplar testes reais e também simulam a execução da especificação formal do modelo. O sistema real usado foi a rede de automação de um centro de controle de sistema elétrico de potência (SEP), pertencente a Companhia Hidro Elétrica do São Francisco (CHESF).

**Palavras-chaves**: Segurança cibernética. redes de automação. produção e transmissão de energia.

# Abstract

Until the 1970s, control and automation of critical systems were based on wired electromechanical devices, and were usually static and offered no flexibility. Next, advances of electronic devices and computer networks introduced new technologies, coupled with proprietary software and hardware philosophy. The deregulation of the electricity sector, together with the market competitive pressure, forced companies producing and transmitting electric energy to seek increasingly competitive automation projects. This context enforced manufacturers to abandon proprietary systems and adopt open standards that are economically more attractive. Consequently, the automation and control networks started operating with known operating systems, based on open technologies and Ethernet communication networks. Nonetheless, this made these networks more vulnerable to cyberattacks. Furthermore, the growth of the terrorist groups in the world raised the governments concern with the vulnerability of the critical infrastructure. As power outages affect all sectors of critical infrastructure, the automation and control systems in this industry have become a potential target for a potential cyberattack. This thesis presents a conceptual model, referred to as, the secure information island. It applies to automation networks of the systems of production and transmission of electric energy. This model is based on the concept of policy-based access control (ORBAC), public key infrastructure (PKI), and secure hardware. The access control defined in the model is enforced for the computers of the automation network as well as users. The idea behind secure information island was inspired from medieval castles, in order to create a controlled area or domain that can be accessed through a single door. Another aspect that contributed to this logical grouping is the heterogeneity of existing technologies in the automation networks. This perimeter can be defined as a part or the entire automation network of a substation, power plants and system control center. To validate this model, use cases were idealized. The validation scenarios implement real tests and simulate the execution of the model´s formal specification. The real system was the automation network of a power system control center (SEP), belonging to Companhia Hidro Elétrica do São Francisco (CHESF).

**Key-words**: Cyber security. automation networks. production and transmission of energy.

# Lista de ilustrações

# Lista de tabelas

# Siglas

**AAA** Authentication, Authorization, Accounting.

**AC** Autoridade Certificadora.

**AC** Attribute Certificate.

**ACL** Access Control List.

**ADT** Abstract Data Type.

**AES** Advanced Encryptation Standard.

**APDU** Application Protocol Data Unit.

**ASIC** Application Specific Integrated Circuits.

**CBC** Cipher Block Chaining.

**CEPEL** Centro de Pesquisa da Eletrobrás.

**CHESF** Companhia Hidro Elétrica do São Francisco.

**DAC** Discretionary Access Control.

**DES** Data Encryption Standard.

**DoS** Denial of Service.

**DSAS** Digital Substation Automation System.

**EAP** Extensible Authentication Protocol.

**EAPOL** Extensible Authentication Protocol Over LAN.

**ECC** Elliptic Curve Cryptography.

**EMS** Sistemas de Gerenciamento de Energia.

**ETS** Executable Test Suite.

**FPGA** Field Programmable Gate Array.

**GOOSE** Generic Object Oriented Substation Event.

**GUI** Graphical User Interface.

**HBCI** Home Banking Computer Interface.

**HMAC** Hash-Based Message Authentication Code.

**HTTP** HyperText Transfer Protocol.

**ICC** Integrated Circuit Cards.

**ICP** Infraestrutura de Chave Pública.

**IDS** Intrusion Detection System.

**IED** Intelligent Electronic Device.

**IETF** Internet Engineering Task Force.

**IFD** Interface Devices.

**IHM** Interface Homem Máquina.

**IKE** Internet Key Exchange.

**IMU** Integrated Merging Unit.

**IP** Internet Protocol.

**ITU-T** International Telecommunication Union.

**JCE** JavaCard Runtime Environment.

**KISA** Korea Internet & Security Agency.

**MAC** Mandatory Access Control.

**MAC** Message Authentication Code.

**MCT** Main Test Component.

**MSC** Message Sequence Charts.

**NAS** Network Access Server.

**NTP** Network Time Protocol.

**OSI** Open System Interconnection.

**PAE** Port Access Entity.

**PC/SC** Personal Computer/Smart Card.

**PDP** Policy Decision Point.

**PEP** Policy Enforcement Point.

**PIN** Personal Identification Number.

**PKCS** Public-Key Cryptography Standards.

**PMT** Policy Management Tool.

**PR** Policy Repository.

**PTC** Parallel Test Components.

**PTP** Precision Time Protocol.

**RADIUS** Remote Access Dial In User Service.

**RBAC** Role-Based Access Control.

**RFC** Request for Comments.

**SAD** Servidores de Aquisição de Dados.

**SAGE** Sistema Aberto de Gerenciamento de Energia.

**SAS** Servidores de Aplicações SAGE.

**SCADA** Supervisory Control And Data Acquisition.

**SDL** Specification and Description Language.

**SDN** Software-Defined Networking.

**SEP** Sistema Elétrico de Potência.

**SGBD** Sistema de Gerenciamento de Banco de Dados.

**SGDP** Serviço de Gerenciamento e Decisão de Políticas.

**SIHM** Servidores de Interface Homem Maquina.

**SIM** Subscriber Identity Module.

**SIRCAM** Segurança da Informação de Redes de Controle e Automação.

**SMV** Sampled Measured Values.

**SSC** Smart Substation Controller.

**SUT** System Under Test.

**TLS** Transport Layer Security.

**TPM** Trusted Platform Module.

**TSI** Test System Interface.

**TTCN-3** Testing and Test Control Notation.

**UTR** Unidade Terminal Remota.

**VLAN** Virtual Local Area Network.

**VM** Virtual Machine.

**VPN** Virtual Private Network.

**VTN** Virtual Tenant Network.

# Sumário

# 1

# INTRODUÇÃO

Na década de 1970, os sistemas de automação eram construídos com relés eletromecânicos e a transferência de informações entre estes era arquitetada através de fios e cabos. As modificações desses sistemas eram difíceis e o custo era elevado. Com o desenvolvimento tecnológico, os sistemas adotaram a tecnologia digital para implementar os dispositivos de proteção e automação de subestações. Neste contexto, ao invés do uso de cabos e fios metálicos para transmitir informações, a comunicação passou a ser através de redes de computadores (MACKAY, 2004).

Nesse cenário, os sistemas de automação possuíam sistemas operacionais e protocolos de comunicações proprietários, dos quais pouquíssimos especialistas tinham conhecimento. Em geral, o domínio dessas soluções proprietárias estava restrito aos fabricantes de equipamentos. Não existia uma preocupação com o conceito de segurança cibernética como nos dias atuais e a segurança da informação era realizada basicamente por isolamento e desconhecimento dos sistemas existentes. Decorrente disso, a única preocupação das empresas com os sistemas de automação e controle era garantir a sua funcionalidade e controlar o acesso físico das suas instalações, (STOUFFER; FALCO; SCARFONE, 2011).

Com a desregulamentação do setor de energia elétrica e com o aumento da competitividade entre as empresas, houve a necessidade de procurar soluções cada vez mais econômicas para a construção dos sistemas de automação e controle do Sistema Elétrico de Potência (SEP). Por outro lado, a evolução das tecnologias Ethernet, o surgimento de redes sem fio, a criação de sistemas operacionais abertos e o crescimento da Internet proporcionaram a adoção dessas tecnologias nos projetos dos novos sistemas de automação, de forma a atingir os objetivos econômicos desejados pelas empresas. Se, por um lado, essas tecnologias permitiram o barateamento dos custos de desenvolvimento dos sistemas de automação, por outro lado, decorrente do uso de protocolos abertos e de sistemas operacionais conhecidos pelo grande público não pertencente ao setor elétrico, houve o aumento da vulnerabilidade das redes de automação (FEITOSA et al., 2008).

Outro fator impactante é a convergência digital que direciona a interligação das redes de automação com as redes de dados tradicionais (corporativas). Dessa forma, uma falha

de segurança na corporação pode afetar a rede industrial. No cenário atual os dispositivos de rede e protocolos de comunicação dos sistemas de automação e controle existentes, em sua grande maioria, não foram projetados levando em consideração os requisitos de segurança cibernética. Geralmente, os dispositivos que possuem tal característica são construídos utilizando mecanismos simples de segurança, como *login* e senha.

No cenário da informática corporativa, já existem diversas tecnologias e soluções para a segurança das informações que operam satisfatoriamente para o setor. No entanto, estas soluções não podem ser aplicadas diretamente às redes de automação, devido às exigências dos sistemas de tempo real e das características dos dispositivos que compõem esses sistemas. Normalmente, as instituições financeiras realizam bilhões de operações monetárias dentro de um padrão considerado aceitável de confiabilidade. Para estes sistemas, a aplicação de retardos nas suas operações decorrente da aplicação de níveis de controle de segurança é considerada aceitável pelos usuários.

Em um ambiente de sistema de automação, as operações precisam ser realizadas em tempo real, portanto, esses retardos gerado pelo acréscimo de camada adicional de segurança não são aceitáveis. As soluções de segurança, que estão sendo pesquisadas atualmente para estes ambientes, têm como finalidade agregar mecanismo de segurança atendendo aos requisitos operacionais dos sistemas de automação e tempo real. Por fim, a falta de comunicação entre as equipes de segurança e as equipes de controle da produção também contribui para tornar as redes industriais vulneráveis a ataques e anomalias (WEI et al., 2011).

Além das questões tecnológicas citadas, questões políticas, religiosas e econômicas motivam o aumento das atividades de indivíduos mal-intencionados. Essas atividades vão desde um adolescente buscando afirmação até um terrorista apoiado por grupos ou nação inimiga, tornando um ataque cibernético às infraestruturas críticas formadas pelos setores de gás, óleo, água e energia de um país, uma possibilidade cada vez mais plausível. É nesse contexto que as universidades, os institutos de pesquisas, as empresas utilitárias de energia e os fabricantes buscam soluções que permitam resolver os problemas de segurança cibernética existentes nas redes de automação. Estas pesquisas visam solucionar problemas, levando em consideração o atendimento dos requisitos dos sistemas de automação com o foco em tempo real e nas limitações do poder computacional dos dispositivos que compõem essas redes.

As pesquisas atuais têm sido direcionadas para as seguintes abordagens: (a) controle de acesso de mensagens, através da utilização de *Message Authentication Code (MAC)*; (b) análise do tráfego de rede com a utilização do *IDS* (HONG; LIU; GOVINDARASU, 2014); (c) mecanismo de controle de acesso e utilização de hardwares seguros, como cartões inteligentes e computação confiável (VAIDYA; MAKRAKIS; MOUFTAH, 2013; HONG; LIU; GOVINDARASU, 2014; BURMESTER, 2013).

Se, por um lado, soluções baseadas no controle de mensagens são, em geral, mais efici-

entes no que se refere ao controle de segurança, por outro, o processamento de mecanismos de segurança aplicado a essas mensagens se torna um fator limitante quando considerado o baixo poder computacional dos dispositivos de automação. Na prática, os dispositivos de automação ainda não têm capacidade de processamento para executar os esquemas de controle de mensagens propostos nas pesquisas. Além disso, é uma solução que requer a substituição de todos os dispositivos que compõem a rede de automação a ser controlada, ou seja, um custo elevado que não atrai as indústrias da área.

A abordagem de controle de acesso tem a vantagem de preservar todos os dispositivos de automação existentes, sem a necessidade de realizar a sua atualização ou substituição. Outra vantagem é que os mecanismos de controle de acesso somente introduzem retardo na inicialização dos sistemas. Neste caso, o atraso é aceitável, considerando que o sistema ainda não se encontra em operação. Esta situação difere de outras abordagens que introduzem retardos durante a operação dos dispositivos. Esses retardos podem inviabilizar os requisitos exigidos pela operação em tempo real.

Para possibilitar a modernização e a integração de sistemas de automação, ao mesmo tempo em que são resguardadas características de isolamento, como o agrupamento lógico ou administrativo de alguns equipamentos, é proposto um modelo que define o conceito de ilhas de automação. No escopo desta Tese, ilhas de automação definem uma divisão física ou lógica entre os dispositivos de automação e suas redes de comunicações com objetivos específicos. Adotando a estratégia de controle de acesso como base, esta Tese especifica e valida um modelo de ilha de automação segura (ilha de segurança), para sistemas de automação associados aos sistemas elétricos de potência.

Uma ilha de segurança pode ser definida como uma ilha de automação cujo acesso é controlado e realizado através de um único ponto. Nesta pesquisa o controle de acesso é baseado em políticas organizacionais e hardware seguro. A estruturação dos sistemas críticos em ilhas de automação, no entanto, não garante, por si só, eliminar as possíveis ameaças às brechas nos sistemas, apesar de facilitar a adoção de políticas de segurança.

A abordagem em ilha de segurança visa preservar todo o investimento ocorrido nos dispositivos legados, considerando que não haverá a necessidade de realizar atualização de *firmwares* ou substituição de dispositivos. Além disso, a solução apresentada tem um custo menor e um tempo de implantação reduzido. Experimentos executados, tanto em ambientes reais quanto em ambientes simulados, comprovam a usabilidade e eficácia desta Tese.

## 1.1 Hipótese

Nesse trabalho, é explorada a **hipótese** de que é possível estabelecer regiões seguras e confiáveis compostas pelos diversos ativos que compõem o sistema de proteção e automação de um sistema elétrico de potência, sem a necessidade de substituição dos dispositivos

eletrônicos inteligentes ou o comprometimento da eficácia do sistema de automação, por perda de desempenho durante a sua operação.

## 1.2 Objetivos e Contribuições

Fundamentado na hipótese, o objetivo principal desta Tese compreende a definição de um modelo conceitual de segurança baseado na autenticação do sujeito na autorização de acesso ao objeto, focado em papéis e elementos que ofereçam garantia dos parâmetros de segurança para sistemas de automação associados ao SEP. Além do objetivo principal, os seguintes objetivos específicos podem ser elencados:

1. Implementação da especificação do modelo utilizando a linguagem formal de especificação e descrição.

2. Definição de critérios de segurança baseados em papéis para os usuários de redes de automação de sistemas elétricos.

As principais contribuições resultantes da pesquisa realizada no escopo desta Tese são:

1. Definição de um modelo de segurança para redes de automação de sistemas elétricos de potência baseado em controle de acesso e em definições de papéis dos usuários;

2. Definição de uma especificação para o modelo em linguagem formal;

3. Definição de um modelo de aplicação de política executada no computador do cliente;

4. Identificação dos papéis dos usuários das redes de automação dos sistemas elétricos de potência.

Além disso, o seguinte artigo: *IEC 61850 Traffic Analysis in Electrical Automation*, correlato ao assunto 'analise de tráfego', publicado na conferência IEEE SmartGridComm, Miami, Florida USA em 2015.

## 1.3 Metodologia de Pesquisa

Os trabalhos conduzidos para elaboração desta pesquisa seguiram uma metodologia que envolveu: (1) a realização de um levantamento do estado da arte para avaliar as diversas abordagens acerca da segurança cibernética para redes de automação; (2) uma análise das arquiteturas e protocolos dos sistemas de automação através do levantamento de requisitos destes sistemas; além de (3) uma investigação dos controles de segurança nos dispositivos de automação atuais que serviram de base para proposição de novos modelos de controle; (4) a elaboração de um modelo conceitual de uma arquitetura de segurança

para as redes de um sistema de automação e controle denominado nesta Tese de ilha de segurança; (5) finalmente, a validação do modelo através das implementações de cenários de testes reais e simulação da execução da especificação.

## 1.4 Estrutura da Tese

Esta Tese está estruturada em seis capítulos, abordando os conceitos fundamentais de segurança cibernética, os sistemas de automação dos sistemas elétricos de potência, a arquitetura da ilha segura de automação, os resultados obtidos e as conclusões desta pesquisa.

**Capítulo 2:** Apresenta os fundamentos conceituais de segurança da informação, de forma que o leitor possa ter um melhor entendimento da concepção do modelo de segurança proposto nesta Tese. Este ainda discorre sobre as redes de automação de sistemas elétricos, detalha os conceitos, a topologia de um sistema elétrico de potência, a arquitetura de ilhas de automação de subestação e apresenta um sumário dos principais protocolos para esses sistemas. Este capítulo tem como objetivo apresentar fundamentos da área de automação de um sistema elétrico de potência.

**Capítulo 3:** Apresenta o estado da arte em segurança cibernética em redes de automação de sistemas elétricos. Neste capítulo, são descritas as diferentes abordagens das pesquisas realizadas atualmente na área de segurança cibernética para sistemas de automação de sistemas elétricos de potência.

**Capítulo 4:** Expressa o modelo conceitual de arquitetura de segurança cibernética para os sistemas de automação do SEP. Neste capítulo é apresentado o modelo de segurança baseado no controle de acesso para os sistemas de automação de sistemas elétricos de potência.

**Capítulo 5:** Demonstra o emprego do modelo conceitual aplicado aos sistemas de automação de sistemas elétricos. Neste capítulo são evidenciadas as soluções de hardware e software que implementam o modelo conceitual em um cenário de sistemas de automação de sistemas elétricos. Além disso, trata os problemas de desenvolvimento e integração do modelo, através da execução da sua especificação em um ambiente simulado, e aborda os resultados encontrados e as dificuldades que ocorreram durante a implementação do modelo.

**Conclusões:** Apresenta uma avaliação geral do desenvolvimento e da implantação realizada nos cenários real e simulado e denota recomendações para trabalhos futuros.

# 2

# FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica dos assuntos abordados nesta Tese, objetivando facilitar o entendimento das propostas e soluções apresentadas. Inicialmente, serão abordados os conceitos relativos às redes de automação utilizadas no sistema elétrico de potência. Em seguida, será descrita a arquitetura das redes de automação e os principais protocolos utilizados na comunicação dos dispositivos que compõem essas redes.

Também será apresentada uma introdução à segurança cibernética, incluindo os conceitos e definições básicas de segurança e as técnicas de controle de acesso no contexto de segurança da informação. Por fim, o capítulo apresenta o conceito de redes definidas por softwares e a utilização do controle de fluxo de pacotes, como fator de decisão do mecanismo de segurança em um cenário de controle de acesso.

## 2.1 Introdução

As redes de automação de um sistema de potência são categorizadas em dois tipos: (a) redes de automação de subestações/usinas e (b) redes de centro de controle.

O sistema elétrico de potência é formado por: (1) uma fonte de geração de energia, que pode ser hidráulica, térmica, eólica, fotovoltaica ou nuclear; (2) uma rede de transmissão de energia; e (3) um centro consumidor de carga (MOMOH, 2008).

Devido aos métodos de produção de energia elétrica, as fontes de geração são usualmente construídas distantes dos centros de carga. Neste caso, é necessário construir linhas de transmissão para levar a energia produzida pelas fontes de geração até o centro de consumo.

Ao longo das linhas de transmissão, existem instalações denominadas de subestações, que têm a finalidade de ajustar a tensão para um valor adequado ao funcionamento de cada parte do sistema. Esta malha, formada de fonte de geração, transmissão, centro consumidor e instalações necessárias à sua operação, é denominada de sistema elétrico de potência. Os equipamentos que compõem esta malha são as linhas de transmissão, os

transformadores e os mecanismos de chaveamento. Eles são denominados de equipamentos primários.

A Figura 1 apresenta um extrato da representação de um sistema elétrico de potência, na forma de um diagrama unifilar.[1]

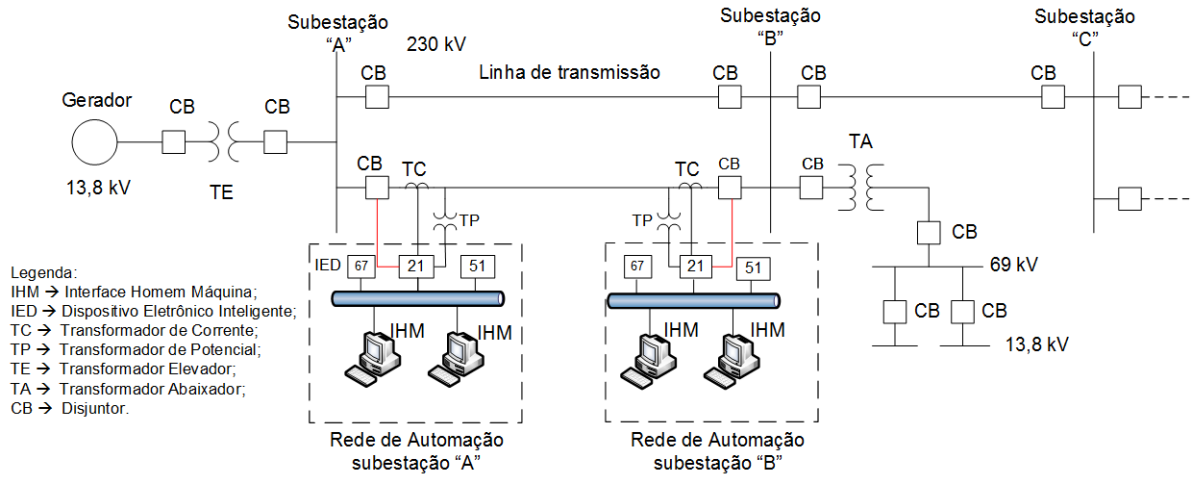


Figura 1 – Exemplo de Diagrama Unifilar de um Sistema Elétrico de Potência (SEP) (Fonte: Autor).

Para facilitar a construção e reduzir custos de implementação, as unidades geradoras de energia elétrica são construídas com uma tensão de geração na ordem de 13,8kV. Este valor é o resultado de uma relação ótima de projeto entre a tensão de geração, o isolamento das partes ativas e a dimensão do gerador. Devido à necessidade do transporte de grandes blocos de potência, a tensão de geração é elevada para um valor maior, usualmente de 230kV ou 500kV, denominada de tensão de transmissão (LEÃO, 2009).

Ao elevar-se a tensão de geração se reduz a corrente nas linhas de transmissão, o que permite empregar cabos condutores de menor secção e, portanto, de menor peso. Com cabos mais leves, ocorre a redução do esforço nas estruturas destas linhas. Por fim, a quantidade de material utilizado na construção da linha é menor, tornando-a mais econômica.

Próximo aos centros de carga, a tensão de transmissão é reduzida para um valor de 69kV, denominada tensão de subtransmissão e, em seguida, é diminuída para 13,8kV, valor adequado para realizar a distribuição de energia nos centros de consumo. Ao redor dos edifícios e das residências, há uma nova atenuação para um valor conveniente ao consumo, ou seja, 110 volts ou 220 volts monofásicos para residências e 380 volts trifásicos nas indústrias. Nas subestações, existe uma rede de dispositivos que tem a função de proteger e controlar os equipamentos primários. Esses dispositivos são denominados de equipamentos secundários.

[1] Diagrama unifilar é uma representação gráfica de uma instalação elétrica na qual o conjunto de condutores de um circuito é denotado por uma única linha.

## 2.2 Arquitetura de Sistemas de Automação de Subestações

A estrutura padrão dos sistemas de automação de subestações é composta de diversos *Intelligent Electronic Devices (IEDs)* que têm como função controlar, monitorar e proteger os equipamentos primários do SEP contra curtos-circuitos e surtos de tensão inerentes à sua operação. Eles são alocados de forma distribuída e agrupados conforme a função do equipamento primário que está sendo controlado e protegido. A comunicação desses dispositivos é realizada por redes Ethernet e utiliza, nas camadas superiores, protocolos de comunicação específicos [2] da área de automação do setor elétrico. A Figura 2 apresenta a estrutura típica de um *Digital Substation Automation System (DSAS)*.

A topologia das redes de automação segue uma estrutura hierárquica que inicia no nível do processo denominado de nível 0. Logo após, segue-se o nível de *bay*, (nível 1), chegando-se ao topo da cadeia hierárquica interna. Este novo ponto é chamado de nível de subestação ou nível 2.

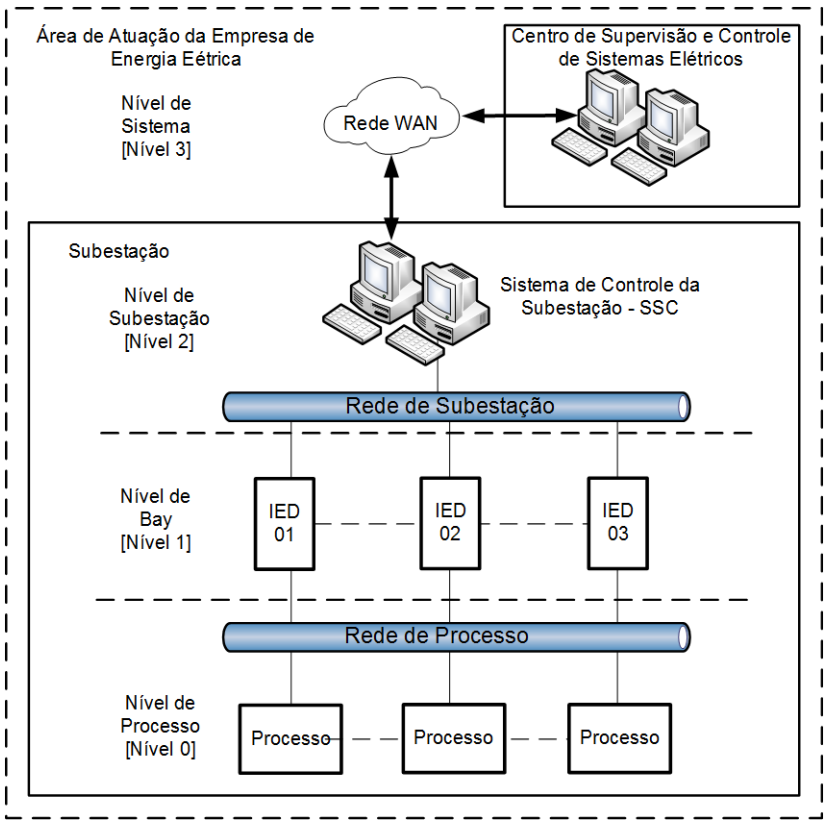


Figura 2 – Estrutura Genérica do SAS, Níveis de Função e Interface (Fonte: Autor).

O nível 0 é composto por sensores que têm as funções de obter informações e de realizar atuação nos equipamentos primários do processo. Neste nível, os sensores recebem

[2] As normas IEC 61850 e DNP3, realiza a descrição dos principais, protocolos de comunicação das redes de automação.

informações dos estados desses equipamentos e de valores analógicos de corrente e tensão do sistema elétrico de potência. Esses valores são coletadas por equipamentos denominados de transformadores de instrumentos que fornecem grandezas elétricas proporcionais aos valores de corrente e de tensão do sistema primário. Essas informações juntamente com os eventos associados ao equipamentos primário são entregues aos IEDs de um *bay* específico da subestação. Esses dados possibilitam a execução das funções de controle e proteção desse *bay*, ou para complementar a cadeia de lógica dos outros *bays* da subestação.

No Brasil, até o presente momento, não existe nenhuma subestação com o barramento de processo implementado, a não ser raras iniciativas de instalação de protótipos envolvendo, normalmente, um único *bay* da subestação com o intuito de demonstração.

O nível de *bay* é composto de dispositivos eletrônicos inteligentes, os quais possuem função de controle e proteção do equipamento primário associado a cada vão da subestação.

A topologia da subestação é formada por um conjunto de vãos[3] interligados por barramentos.

Os IEDs utilizam dados associados aos *bays* que estão sendo controlados e também podem exercer comandos sobre os atuadores dos dispositivos de chaveamento associados. Além da troca de informação com o processo, estas funções também realizam o diálogo com funções pertencentes a outros *bays*, de forma a realizar operações de intertravamento ou resolver dependências das diversas lógicas de controle. Esta comunicação é conhecida como horizontal. Além desta, os IEDs realizam a comunicação com o sistema computacional ao nível de subestação (nível 2).

No topo da cadeia hierárquica da subestação, localiza-se o nível 2, que é composto por um sistema computacional responsável pelo monitoramento e o controle desta, através de um *framework* de software denominado de *Supervisory Control And Data Acquisition (SCADA)*. O hardware que faz parte do sistema computacional é denominado de controlador da subestação. A Interface Homem Máquina (IHM) deste sistema computacional disponibiliza para o operador as funções de supervisão e controle da subestação. Neste ponto, além da realização da comunicação com os dispositivos dos níveis inferiores, é realizada a comunicação com os centros de controle remoto, permitindo, dessa forma, a operação sistêmica do SEP. As interfaces que suportam estas funções são denominadas de telecontrole e telemonitoramento.

Na extremidade, encontra-se o nível três. Esse nível está posicionado em um centro de controle que normalmente está fisicamente localizado distante das subestações. Estes centros de controle têm a função de monitorar e controlar o sistema elétrico de potência de forma sistêmica. Estes gerenciam o SEP de uma ou mais regiões de um país.

A topologia física típica de uma subestação é composta por uma área que abriga os

[3] Um vão ou *bay* de subestação é uma parte formada por equipamentos primários e um conjunto de mecanismos de chaveamento necessário para executar uma função do sistema elétrico de potência.

equipamentos primários denominada de pátio de manobra ou pátio da subestação. No interior do pátio de manobra existem cabanas de relés que são distribuídas por *bay* ou setor, e que abrigam os IEDs ao nível de vão (nível um). Outra edificação que faz parte do conjunto arquitetônico da subestação é o edifício de comando, neste último está localizado o sistema computacional da subestação (nível dois).

A infraestrutura de rede de automação de uma subestação digital utiliza como meio físico fibras ópticas que estão distribuídas fisicamente no pátio, nas cabanas de relés e na sala de comando. No pátio e nas cabanas de relés, esta infraestrutura usa uma topologia em anel e na sala de comando utiliza a topologia em estrela. A Figura 3 detalha uma arquitetura de uma rede de automação de subestação contendo apenas as redes ao nível de *bay* e ao nível de subestação. Conforme observado nesta figura, os IEDs na cabana de relés estão interligados por anéis individuais para cada *bay*.

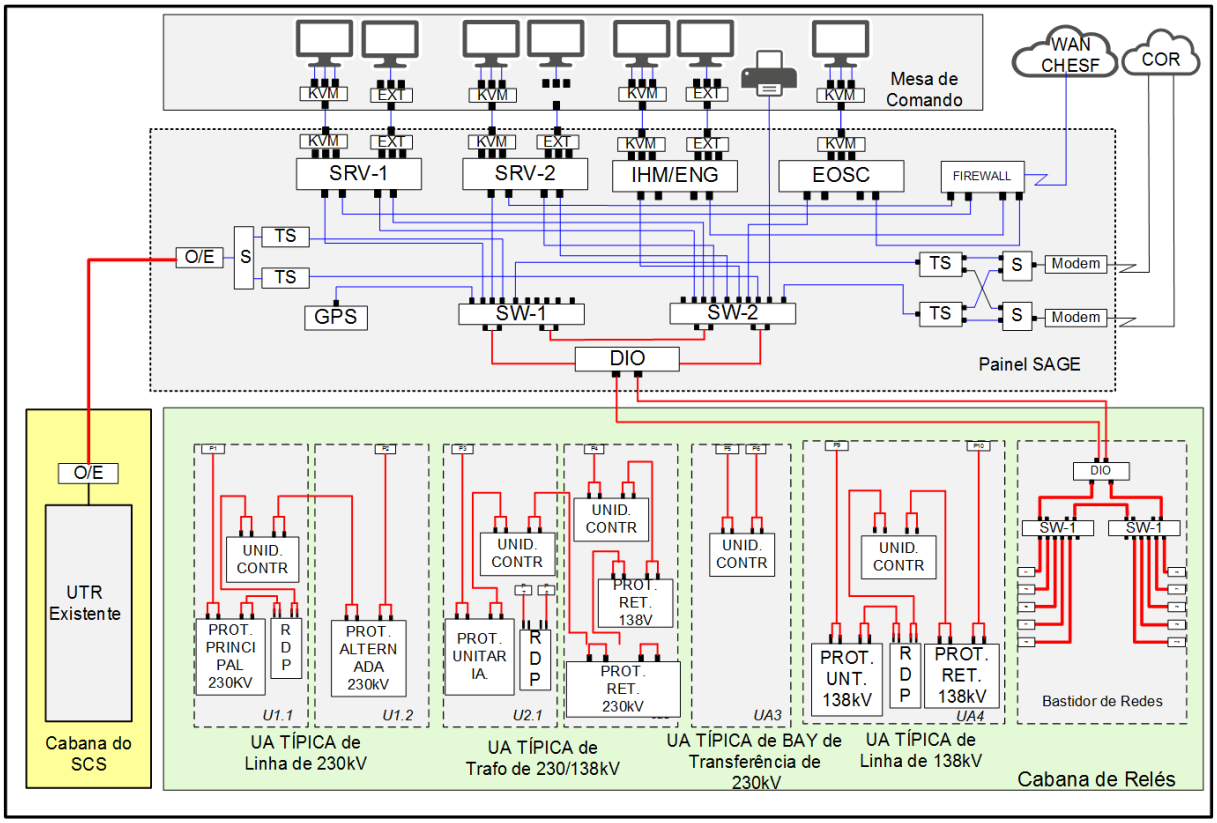


Figura 3 – Arquitetura Típica de um SAS (Fonte: CHESF).

Estes anéis são fechados por *switches* em um bastidor de rede localizado na cabana de relés de cada setor. Estes *switches* estão conectados via distribuidor óptico aos *switches* da rede de nível dois da subestação localizada na sala de comando, formando um anel ótico adicional. A rede de nível dois abriga os servidores do sistema SCADA, as IHM do operador e os terminais *servers* que realizam a comunicação com os centros de controle remoto ou com outros dispositivos legados que utilizam a comunicação serial como meio de

transmissão. Pode-se citar como exemplo deste tipo de dispositivo uma Unidade Terminal Remota (UTR).

## 2.3 Arquitetura de Redes de Automação dos Centros de Controle de Sistemas Elétricos

A arquitetura básica de um centro de controle é composta por hardwares, softwares e serviços necessários à operação de um SEP. Esses serviços são hospedados em servidores específicos como: servidores SCADA, servidores Sistemas de Gerenciamento de Energia (EMS) e servidores terminais de comunicação. Pode-se citar como exemplo de aplicações EMS, os estimadores de estado, o controle automático de geração, o despacho ótimo de potência e as aplicações para treinamento dos operadores.

No Brasil, o sistema SCADA mais utilizado pelas empresas concessionárias de energia elétrica (utilitárias) é um sistema desenvolvido pelo Centro de Pesquisa da Eletrobrás (CEPEL) em conjunto com as empresas utilitárias do sistema Eletrobrás, denominado de Sistema Aberto de Gerenciamento de Energia (SAGE).

A implementação de uma arquitetura típica de um centro de controle com as funções SCADA e com o EMS de uma empresa concessionária de energia é composta por servidores de aplicação, servidores de comunicação, e as IHMs que realizam a função de interfaces homem/máquina. Os servidores são divididos em Servidores de Aquisição de Dados (SAD), Servidores de Aplicações SAGE (SAS) e Servidores de Interface Homem Maquina (SIHM). Esta arquitetura apresenta uma topologia de rede em redundância na qual cada servidor é composto por um servidor principal e um servidor secundário.

Os servidores de aplicação SAGE se conectam entre si por meio de uma rede denominada de difusão confiável para troca de informações em tempo real. Essa aplicação se comunica com os demais servidores através de uma outra rede chamada de SAGE. Estas são instaladas fisicamente no mesmo *switch*, com separação lógica através de *Virtual Local Area Network (VLAN)*. As VLANs são nomeadas de tempo real e de difusão. Os *switches* também são duplicados, o que garante a redundância total das redes de difusão e da rede SAGE.

Uma terceira rede conhecida por rede operacional oferece suporte à comunicação com as IHMs e com outros componentes que realizam funções adicionais em um centro de controle. Esses componentes são os servidores de sincronização de tempo, os servidores de terminais de comunicações, os sistemas de impressão e os servidores de gerenciamento de *vídeo wall*. A Figura 4 apresenta a arquitetura típica de um centro de controle baseado no sistema SCADA/ SAGE.

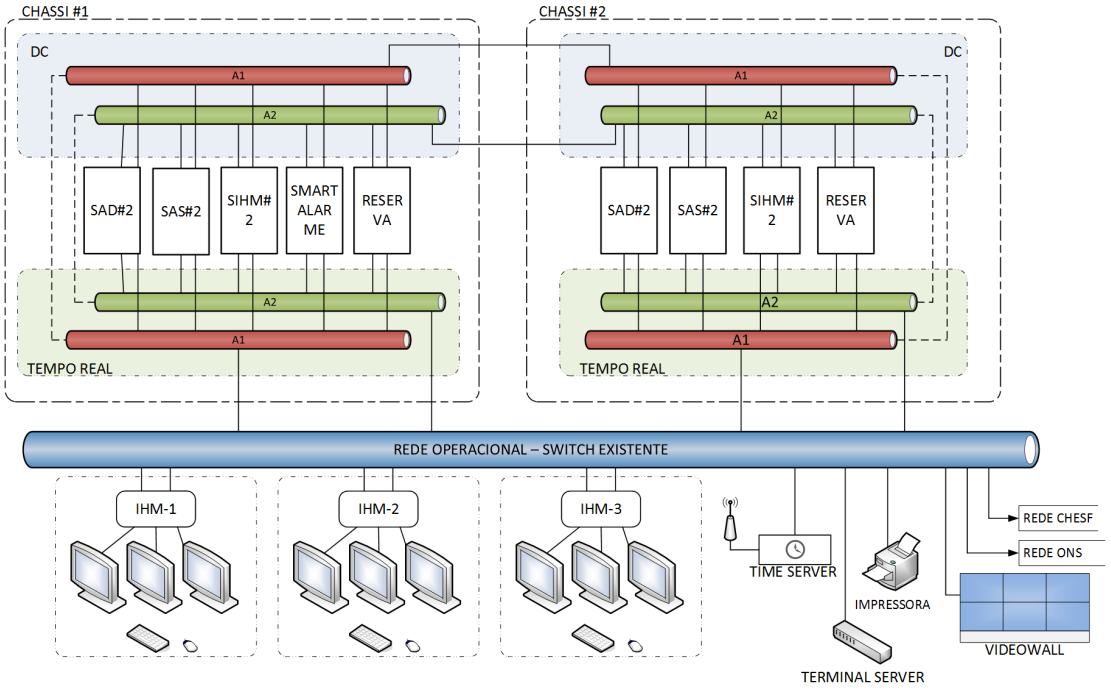


Figura 4 – Arquitetura Típica de um Centro de Controle Baseado no SAGE (Fonte: CHESF).

## 2.4 Segurança Cibernética

Segurança cibernética é a tecnologia que tem como objetivo identificar as vulnerabilidades dos sistemas computacionais e garantir a proteção das informações e dos sistemas contra ameaças cibernéticas. As principais ameaças identificadas nesta tecnologia são: terrorismo cibernético, guerras cibernéticas e espionagens cibernéticas (STALLINGS, 2006).

Outro conceito importante no âmbito da segurança da informação é o da vulnerabilidade. Esta é definida como uma fraqueza que permite que um atacante reduza a garantia da obtenção da informação de um sistema computacional. Ela é resultado da interseção de três elementos: suscetibilidade a falhas do sistema, acesso do atacante a esta falha e a sua consequente exploração (IGURE; WILLIAMS, 2008).

A recomendação X.800 do *ITU-T* define uma arquitetura de segurança baseada no modelo de segurança do *Open System Interconnection (OSI)*. Esta arquitetura classifica um ambiente nas seguintes abordagens:

- Ataque à segurança é qualquer ação que comprometa a informação de uma organização. Estes são classificados em ativos ou passivos. O artigo (DOULIGERIS; MITROKOTSA, 2004; MIRKOVIC, 2003; SAI, 2003) descreve os principais tipos de ataques que podem ser utilizados em uma rede de automação;

- Mecanismos de segurança é um processo ou um dispositivo incorporado, que é projetado para detectar, impedir ou permitir a recuperação de um ataque à segurança;

- Serviço de segurança é uma atividade de processamento ou de comunicação que aumenta a segurança dos sistemas computacionais e de transferências de informação de uma organização.

### 2.4.1 Ataques à Segurança

O aumento da conectividade dos usuários e a quantidade de novos serviços oferecidos pela Internet proporcionam aos atacantes *(crackers)* a capacidade de criar ou explorar novas técnicas, possibilitando uma gama de ataques, ameaçando a integridade das infra-estruturas de rede e violando a privacidade dos usuários (SHIREY, 2000). A recomendação X.800 da ITU-T e a RFC 2828 *Internet Security Glossary* da *Internet Engineering Task Force (IETF)* classificam os ataques em duas naturezas:

- Ataques passivos: este tipo de ataque tem por natureza observar ou monitorar as transmissões da informação a fim de obter conhecimento do sistema da vítima que servirá de base para o planejamento de um ataque futuro. Este tipo de agressão é ainda subclassificado em liberação do conteúdo da mensagem, isto é, a obtenção de informações importantes ou confidenciais do sistema da vítima e de análise de tráfego. Neste último caso, o ataque é mais sutil, pois o oponente obtém informações do conteúdo e do comportamento das mensagens válidas que poderão ser utilizadas para realização de um ataque ativo;

- Ataques ativos: têm como característica o envolvimento de algum tipo de modificação ou de criação de fluxo de dados falsos que são divididos ainda em quatro subcategorias: disfarce, repetição, modificação de mensagens e negação de serviço.

### 2.4.2 Serviços de Segurança

São serviços fornecidos por uma camada de protocolos de comunicações de sistemas abertos que garante a segurança adequada dos sistemas computacionais ou da transferência da informação entre eles. Eles são divididos nas seguintes categorias (STALLINGS, 2006):

- **Serviço de Autenticação** - estabelece ou avalia a identidade de algo ou alguém como verdadeira, através da confirmação da procedência de um objeto ou pessoa;

- **Serviço de controle de acesso** - provê o impedimento de uso não autorizado de um recurso, ou seja, garante o controle de quem pode ter acesso a um determinado recurso e sob que condições o acesso pode ocorrer e quais os direitos de uso desse recurso;

- **Serviço de confidencialidade** - é a proteção dos dados transmitidos contra ataques passivos. Além da garantia da proteção contra a observação dos conteúdos

das mensagens, este serviço oferece garantia contra a observação da informação, da origem, do destino, da frequência de transmissão e do tamanho das mensagens;

- **Serviço de integridade** - garante que as informações sejam recebidas conforme foram enviadas, sem nenhum tipo de duplicação, inserção, modificação, reordenação ou repetição;
- **Serviço de irretratabilidade** - impede que o receptor ou transmissor negue autoria de uma mensagem enviada. Dessa maneira, quando uma mensagem é enviada, o emissor pode provar que ela foi recebida. Da mesma forma, o receptor pode provar que uma determinada mensagem foi enviada pelo emissor;
- **Serviço de disponibilidade** - concede a garantia de que um sistema ou que um recurso estará sempre acessível e utilizável conforme a especificação de desempenho do sistema.

### 2.4.3 Controle de Acesso

Em sistemas computacionais, o controle de acesso define que objeto está habilitado para executar uma operação. O objeto pode ser um processo, um usuário, um computador ou uma entidade computacional ativa. As operações envolvidas nesse processo podem ser: escrita, leitura, execução, busca e eliminação. A finalidade do controle de acesso é garantir a confidencialidade e integridade da informação e, em uma extensão menor, a disponibilidade (THION, 2008).

O controle de acesso fornece apenas as permissões necessárias para o usuário obter acesso a um objeto. Quando o usuário está realizando uma escrita autorizada, o controle de acesso está garantindo a integridade, isto é, está evitando uma escrita indevida realizada por pessoas não credenciadas. O mesmo fato se dá quando é realizada uma leitura autorizada, pois o controle de acesso está garantindo a confidencialidade dos dados. Neste caso, evita que pessoas não autorizadas acessem o conteúdo das informações.

O serviço de controle de acesso é composto dos processos de autenticação, autorização e auditoria. Nesse contexto, o controle de acesso pode permitir ou negar a utilização de um objeto por uma entidade ativa, um indivíduo ou um processo.

A identificação e autenticação é um processo que se estabelece em duas etapas. Na etapa de identificação, o usuário informa quem ele é, através de um nome de usuário ou de uma característica própria incontestável, como, por exemplo, a biometria. Na etapa de autenticação, a identidade do usuário é verificada através de uma credencial fornecida por ele. Esta credencial pode ser um nome e senha, uma informação biométrica ou através de uma chave armazenada em um hardware seguro garantida por um terceiro confiável.

A autorização é o processo que define quais os direitos e permissões que o usuário tem sobre o sistema. A auditoria é o processo que trata da coleta de informações relacionadas com a utilização dos recursos de um sistema computacional pelo usuário.

As estratégias de controle de acesso podem ser categorizadas nas técnicas discricionária, mandatória e baseada em papéis.

O *Discretionary Access Control (DAC)* é uma técnica cuja política de acesso é determinada pelo proprietário do recurso. Nesta técnica, eles decidem qual usuário tem permissão de acesso e quais os privilégios que ele possui. Muitos dos atuais sistemas operacionais, como Unix e Windows, baseiam a sua segurança no conceito de controle de acesso arbitrário e fornecem apenas duas categorias de usuários:

a) Administradores: que têm acesso total a todos os recursos dos sistemas;

b) Usuários normais: que têm acesso a aplicações e arquivos necessários às suas demandas.

O principal problema do controle de acesso arbitrário é que não existe aplicação de políticas de segurança em todo o sistema e as medidas de proteção dependem das permissões de cada usuário. Qualquer programa executado herda as permissões do usuário responsável. Nesse caso, o programa ativo pode modificar todo e qualquer arquivo que o usuário tenha permissão de acesso. Dessa forma, um vírus pode infectar este e se espalhar facilmente pela máquina do usuário e atingir toda a rede industrial.

Uma alternativa para esse problema é o *Mandatory Access Control (MAC)*. Com o MAC, a política de acesso é determinada pelo sistema e não pelo proprietário do recurso. Neste tipo de controle de acesso, todos os sujeitos e objetos têm um rótulo de sensibilidade. Este define o nível de confiança de ambos. Neste caso, o sujeito deve ter um nível de confiança maior ou igual ao do objeto.

Sistemas usando MAC são implementados utilizando *Access Control List (ACL)*. A ACL é uma lista que define os direitos e as permissões que são dados a um sujeito sobre um determinado objeto. O MAC fornece mecanismos para um administrador central aplicar as políticas de acesso aos serviços que serão executados pelo sistema operacional. Ele fornece segurança individual e isolada, uma vez que cada usuário precisa ter privilégios de acesso específico para cada recurso. O MAC suporta uma grande variedade de categorias de usuários e restringe o dano causado por um software malicioso.

O *Role-Based Access Control (RBAC)* define os direitos e as permissões de um usuário a um recurso baseado no papel que este exerce na organização. Esta técnica tem a vantagem de que os direitos e permissões são dados por grupos de usuários. Neste caso, em uma situação em que o usuário mude de responsabilidade, a simples troca de papel do usuário no sistema, habilita os direitos e as permissões necessárias à execução da sua nova atribuição.

A diferença de filosofia entre DAC e MAC acontece essencialmente nas políticas de acesso. Enquanto DAC parte da premissa de que todos os recursos do sistema são acessíveis e a segurança é adicionada através de restrições de acesso a alguns componentes, o MAC parte da premissa de que nenhum recurso é acessível e todas as operações têm que, explicitamente, serem permitidas. Sendo assim, o controle de acesso obrigatório MAC se apresenta, hoje em dia, como o melhor mecanismo para proteção de sistemas críticos das ameaças internas e externas às suas redes de automação.

### 2.4.4 Gerenciamento de Políticas

Um sistema baseado em políticas é gerenciado através de um conjunto de regras que determinam a ação a ser tomada, de acordo com um conjunto de parâmetros de entrada (VERMA, 2000).

No contexto de gerenciamento de redes, alterações, tanto na topologia da rede quanto na configuração dos seus componentes, como roteadores e servidores, podem aumentar ainda mais a complexidade do gerenciamento (RAMOS, 2014). Quando esta necessidade de controle agrega também o gerenciamento de recursos locais (diretórios e programas), a complexidade é potencializada.

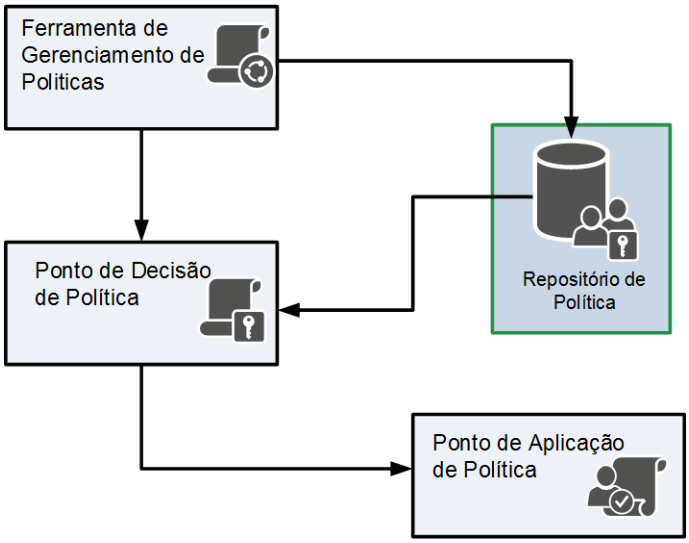


Figura 5 – Modelo Genérico de Gerenciamento de Políticas (Fonte: (VERMA, 2002)).

O modelo proposto, vide Figura 5, utiliza o arcabouço desenvolvido pelo IETF que define um padrão para gerenciamento de políticas compreendido pelo ponto de decisão de políticas, do acrônimo em inglês *Policy Decision Point (PDP)*, pelo Ponto de execução de políticas, do inglês *Policy Enforcement Point (PEP)*, pelo repositório de políticas, do inglês *Policy Repository (PR)* e pela ferramenta de gerenciamento de políticas, do inglês *Policy Management Tool (PMT)*.

O PMT permite a edição, tradução e validação das políticas definidas que são armazenadas no repositório e posteriormente postas em prática. Este módulo realiza a tradução

de conceitos abstratos para a sintaxe do modelo de informações do repositório e a sua validação consiste na checagem das sintáticas e semânticas básicas. O PMT também suporta a especificação do modelo de informação da rede, edição e gerenciamento de políticas através de interface gráfica amigável ao administrador da rede (VERMA, 2002).

O PR é o local onde as políticas definidas para um determinado domínio, são armazenadas, ou seja, é a base de informações sobre as políticas. Exemplos deste nível são servidores de diretórios e de banco de dados.

O PDP é responsável por efetuar o processamento lógico das regras definidas para o sistema, juntamente com outros dados relevantes para a administração da rede. Ele adquire, distribui e opcionalmente traduz regras em mecanismos de políticas do objeto. O PDP toma as decisões consultando o repositório e posteriormente as transforma em novas configurações que são enviadas para os PEPs.

O PEP é responsável por gerenciar cada equipamento de acordo com as instruções recebidas pelo PDP. Ele possui também a responsabilidade de traduzir e configurar as políticas nos equipamentos finais da rede.

Para que a troca de informações seja efetuada com sucesso entre os vários componentes da arquitetura, existe a necessidade da utilização de protocolos normalizados. Assim, estes protocolos permitem uma comunicação sem restrições entre os produtos de vários fabricantes, assegurando o caráter global da solução.

#### *2.4.4.1 Gerenciamento de Políticas Baseado em Papéis*

O modelo de acesso especificado para a ilha de segurança é fundamentado no RBAC, que é um modelo de controle de acesso não discricionário, que permite e promove a administração central de uma política de segurança específica da organização (FERRAIOLO; KUHN, 2009).

No modelo RBAC, as regras definidas para um sistema não se aplicam diretamente a um determinado usuário e sim aos papéis aos quais estão associados. Se o usuário possui um papel, então herdará suas permissões e proibições, tendo os mesmos privilégios e as mesmas proibições dos usuários que compõem o mesmo papel.

Na ilha de segurança, as políticas foram definidas em três categorias: políticas gerais, de grupo e individuais:

- Políticas de grupo: a cada função no ambiente protegido será associado um papel (*Role*), e cada papel deve ter políticas que permitam que suas funções na organização sejam desempenhadas, observando-se com cuidado a sanidade do sistema. As Políticas de Grupo adicionam mais regras aos sujeitos participantes, ou seja, além das permissões e proibições concedidas pelas políticas gerais, cada grupo tem suas autorizações específicas. Caso uma política de grupo entre em conflito com uma

política geral, a primeira política, por ser mais específica, precede em prioridade em relação à segunda;

- Políticas individuais: estas políticas são específicas para um usuário, que também está submetido às políticas gerais e às políticas de grupo correspondentes ao papel que pertencem. Essas políticas são prioritárias sobre qualquer outra, nos casos de conflito, e, por este motivo, (por criarem uma vulnerabilidade prioritária sobre qualquer outra regra), devem ter seu uso bastante limitado. Uma opção para criar uma barreira contra o uso impróprio de políticas individuais é usar o recurso de contextos temporais, oferecidos pelo OrBAC. Em outras palavras, as políticas individuais são automaticamente invalidadas após o fim do seu tempo de vida definido. Esta estratégia evita que usuários possuam excesso de privilégios associados a políticas individuais. Neste caso, eles têm suas ações limitadas a um determinado período de tempo, e que uma permissão dada desta maneira precise de constante revalidação por parte do responsável. Assim, espera-se evitar possíveis "brechas" operacionais na segurança do sistema.

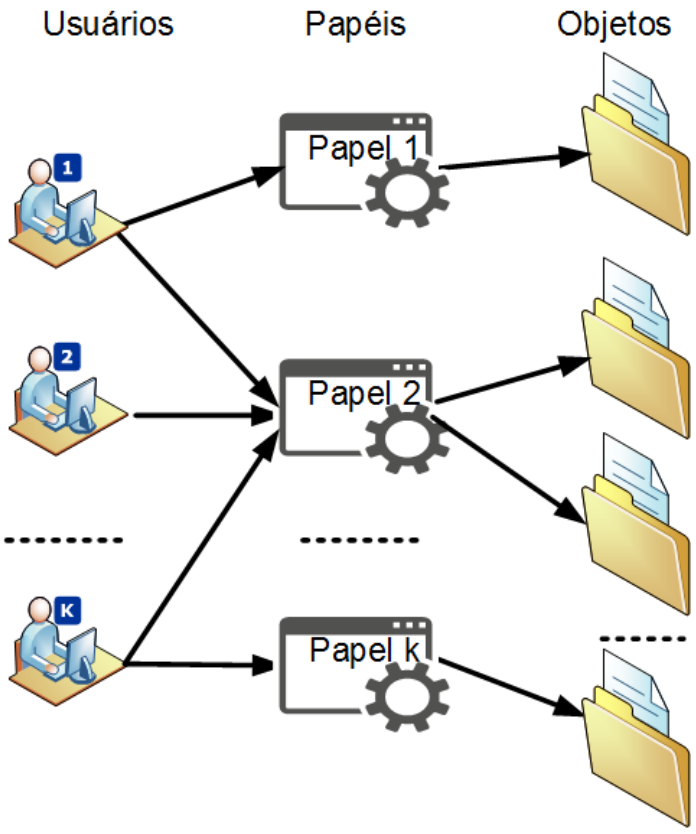


Figura 6 – Representação do Modelo de Gerenciamento Baseado em Papéis (Fonte: Proj. SIRCAM).

O conceito de hierarquia de papéis também é introduzido nesse modelo. Nele, é possível que papéis herdem permissões de outros na hierarquia. Restrições também são aplicáveis,

pois, é possível definir que qualquer usuário poderá exercer dois ou mais papéis. A Figura 6 exemplifica esse comportamento.

Em um ambiente de automação, pode-se identificar funções e papéis que dependem dos objetivos das empresas. Por exemplo, para o setor elétrico, identificam-se os papéis de operador de sistema elétrico, supervisor de operação, administrador de banco de dados SCADA, administrador de segurança, mantenedor e integrador de sistemas. A Tabela 1 sumariza as funções e papéis dos principais usuários de um sistema de automação do setor elétrico. Essas funções e papéis podem sofrer mudanças, dependendo da estrutura organizacional das empresas.

Tabela 1 – Mapeamento das Funções da Ilha de Automação em Papéis OrBAC (Fonte: Autor).

| Função | Descrição | Papel |
|---|---|---|
| Operador de Sistema Elétrico | Sua função é operar o sistema elétrico de potência, sua única interação com a rede de automação é através da interface gráfica do sistema SCADA | Operador |
| Supervisor de Operação | Possui a função de supervisionar ações tomadas pelos operadores e pode realizar pequenas tarefas, além da interação com o sistema SCADA, como, por exemplo, reinicializar o sistema. | Supervisor |
| Administrador de Base de Dados | Responsável pela gerência do banco de dados do sistema SCADA | AdminBaseDados |
| Administrador de Segurança | Designado para estabelecer e manter as regras de segurança do ambiente protegido | AdminSegurança |
| Mantenedor | Encarregado por manter a funcionalidade e disponibilidade da rede de automação | Mantenedor |
| Integrador de Sistemas | Responsável pela ampliação e melhorias do sistema de automação e sua integração com outros sistemas e dispositivos | Integrador |

De fato, para controle de acesso RBAC, é mais importante saber quais as responsabilidades do usuário em relação à sua posição na organização, ao invés de identificar regras individuais para cada usuário do sistema. Percebe-se que esta abordagem é mais facilmente escalável em relação a modelos de políticas diretamente associadas aos usuários. No entanto, o modelo RBAC não permite a especificação de permissões atribuídas a contextos, um recurso que permite um mapeamento mais realista, uma vez que pode definir regras específicas a determinados usuários e em certas condições. Em outras palavras, se uma dada permissão for concedida a um determinado papel, todos os usuários que exercem este papel terão a permissão concedida.

### 2.4.5 Autenticação Através de Cartões Inteligentes

A norma ISO 7816 define as características de cartões inteligentes (*smartcards*) em vários aspectos como: características físicas (tamanho e posição de contato), características elétricas, os protocolos de transmissão entre o cartão e o leitor de cartão e um conjunto de instrução de 109 comandos. Entretanto, essa norma não define a interface entre o leitor de cartões e o computador do usuário. Para atender esta lacuna, existem várias iniciativas de fabricantes que definem APIs para realizar esta comunicação. Estas APIs estabelecem características comuns e complementares de forma que o critério da escolha do uso depende dos requisitos do projeto. Na prática, as principais bibliotecas em uso são baseadas nas especificações OpenSC, Personal Computer/Smart Card (PC/SC) e JavaCard.

O OpenSC é um projeto de código aberto sob os termos de licenciamento LGPL (FSF, 1999), que tem como objetivo fornecer um conjunto de ferramentas e bibliotecas para serem usadas com *smart cards*, tendo como foco os que podem ser criptografados (JAKUGE, 2016). A sua versão mais recente é a 0.13.0 que implementa os padrões *Public-Key Cryptography Standards (PKCS)#11* (OPENSC, 2009) e *PKCS#15* (LABORATORIES, 2000) definindo o uso de *smart cards* em aplicações.

JavaCard é um conjunto de biblioteca utilizada pelos cartões com suporte a linguagem Java. Na sua arquitetura de software contém o ambiente *JavaCard Runtime Environment (JCE)* que tem capacidade de executar bytes de códigos Java. Na arquitetura de hardware, o JCE inclui uma *Virtual Machine (VM)* que está localizada na memória ROM do cartão e uma biblioteca de classe básica do Java. Essa máquina em conjunto com a biblioteca Java executam o papel de sistema operacional, controlando os recursos de memória e I/O do cartão. Para o mundo externo, o JavaCard se comporta como um cartão comum em conformidade com a norma ISO 7816. Neste caso, implementa-se a comunicação através de *Application Protocol Data Unit (APDU)*, utilizando os protocolos T0 e T1 definidos nesta norma.

O OpenSC se tornou compatível com uma série de aplicações, que vão desde navegadores como o Mozilla Firefox, passando por aplicações como gerenciadores de e-mail como, por exemplo, o Mozilla Thunderbird, as aplicações em assinatura digital como empregada pelo Adobe Reader e os projetos usados pelo setor bancário, como o *Home Banking Computer Interface (HBCI)* (ZKA, 2003).

A arquitetura da biblioteca OpenSC é disposta sob a forma de camadas e módulos que operam entre si para fornecer às aplicações a possibilidade de manipular dados e realizar operações com *smart cards*. O OpenSC apresenta API para comunicação em conformidade com os padrões PKCS#11 e PKCS#15.

O PC/SC *(Personal Computer/Smart Card)* é uma especificação desenvolvida com a finalidade de descrever os requisitos mínimos dos *Integrated Circuit Cards (ICC), Interface Devices (IFD)* [4] e Computadores Pessoais. Esta especificação possibilita a existência de in-

[4] *Interface Device* (IFD) são os dispositivos que realiza a leitura do cartão

teroperabilidade entre dispositivos fornecidos por diversos fabricantes (PCSC, 2005). Este padrão constitui um esforço conjunto da Apple, Axato, Gemalto, Infineon, Microsoft, Philips e Toshiba para estabelecer critérios que pudessem ser usados por toda a indústria. O padrão definido pelo PC/SC está presente nos principais sistemas operacionais do mercado. Sua função em cenários que utilizam *smart cards* ou *tokens* é servir como meio de gerenciamento do transporte para os dados das operações com tais dispositivos. A arquitetura básica é mostrada na Figura 7.

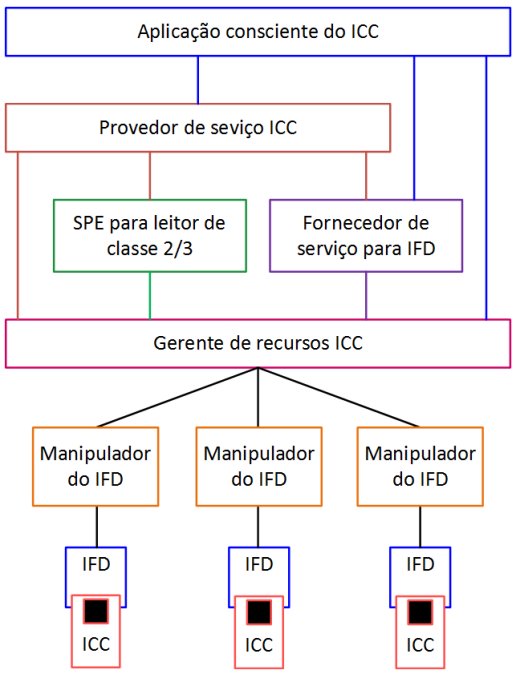


Figura 7 – Arquitetura Básica do PC/SC (Fonte: (PCSC, 2005) ).

No primeiro nível, têm-se os ICC *Aware Applications* que correspondem às aplicações que estão rodando no sistema operacional e que desejam usar os recursos fornecidos por um ou mais ICCs.

No segundo nível, existem os provedores de serviços que correspondem a um encapsulamento das funcionalidades fornecidas pelo ICC ou IFD específico, permitindo a existência de interfaces de alto nível que estão disponíveis para serem usadas pelas aplicações do cartão. Esta camada não é necessariamente obrigatória, podendo a aplicação se comunicar diretamente com o ICC *Resource Manager* da camada inferior. Nela, estão os módulos responsáveis por fornecer recursos específicos de um dado sistema operacional de um ICC ou de um IFD.

No terceiro nível, tem-se o ICC *Resource Manager (ICCRM)*, que é responsável pelo controle de acesso aos ICC's, assumindo o papel de ponto chave na arquitetura. Ademais, o ICCRM é responsável por descobrir os IFDs que estão instalados no sistema e tornar esta informação acessível a outras aplicações, além de gerenciar o acesso concorrente aos ICCs, mantendo a consistência do estado da operação e das aplicações em uso.

No quarto nível, há o IFD *Handle* (IFDH), que é responsável por mapear as capacidades nativas de um IFD para a interface do IFD *Handle*, (ZKA, 2003; PCSCWGP, 2007). O IFD *Handle* se comporta como um software de baixo nível que suporta um canal específico de comunicação. Como exemplo de IFDH, existe a biblioteca **libccid**, que implementa uma interface para comunicação com dispositivos que estejam ligados a uma porta serial ou USB do computador. Além disso, é usado para conectar o IFD ao PC e prover acesso a uma funcionalidade específica do IFD (PCSCWGP, 2005). Assim, por meio desse mapeamento existente, o IFDH é capaz de fornecer chamadas de comunicação de baixo nível para os IFDs.

No quinto e último nível, existem os IFDs e os ICCs, que constituem a parte física da arquitetura. O IFD tem como característica prover o meio de comunicação com o *smart card*, conforme as especificações ISO 7816 (ISO, 2006), que definem como tais dispositivos devem se comunicar e realizar suas operações.

## 2.5 Segurança em Redes de Automação de Sistemas Elétricos

Atualmente, grandes empresas - particularmente aquelas cujas atividades envolvem infraestrutura crítica como geradoras de energia, telecomunicações, água e esgoto, transporte coletivo, óleo e gás - enfrentam desafios na área de segurança dos sistemas de automação. De um lado, existe a ameaça real do terrorismo cibernético e de falhas de segurança involuntárias e/ou maliciosas. Por outro lado, existe a pressão para aperfeiçoar o desempenho financeiro das empresas, através da integração simplificada de suas operações, visando otimizar os recursos.

Políticas como a convergência de redes, uso de tecnologias com sistema operacionais conhecidos e protocolos abertos aumentam os problemas de segurança cibernética dos sistemas de automação. Portanto, existe a necessidade de ações que venham mitigar esses problemas.

Na prática, no caso do setor elétrico, existem empresas utilitárias que mantêm uma rede para operação do processo e outra para atender à gestão de forma isolada. A questão a ser considerada é operar, como no passado, em que as redes e os sistemas ficavam totalmente isolados e desconectados da Internet ou compartilhar a infraestrutura existente, diminuindo a ineficiência e colaborando para um melhor desempenho do processo produtivo.

O primeiro passo para prover a segurança é compreender quais são as vulnerabilidades do sistema e analisar como elas estão presentes nos processos corporativos e de automação. Por exemplo, um serviço web usando *HyperText Transfer Protocol (HTTP)* que controla algum processo produtivo em uma rede industrial pode ser considerado como um potencial ponto de falhas e vulnerabilidades. Dessa forma, é preciso entender as vulnerabilidades

relacionadas ao protocolo HTTP, bem como o software utilizado no servidor. Além disso, se esse serviço está disponível para acesso através das redes administrativas, compostas por centenas de *hosts* que possuem acesso à Internet, existem muitas evidências de que as vulnerabilidades possam ser detectadas e exploradas por um ou vários atacantes.

Segundo (DERYNCK, 2004), as redes industriais utilizam os mesmos equipamentos e tecnologias de comunicação das redes de dados convencionais utilizadas na gestão. Entretanto, o seu foco de atuação é voltado para informações de controle do processo produtivo (e não para o tráfego de informações como e-mail, páginas web, etc), existem certas peculiaridades que as diferem das redes de gestão, como por exemplo, equipamentos específicos e restrições operacionais em função da operação em tempo real. Desta forma, qualquer solução de segurança deve necessariamente mesclar a experiência do especialista em segurança cibernética com a experiência do especialista da área de sistema de controle de produção.

Atualmente, os mecanismos de controle de segurança cibernética dos sistemas de automação de subestações são implementados com base na utilização de senhas e nome de usuário para controlar o acesso aos IEDs, os sistemas computacionais de controle local da subestação e os sistemas de automação de centros de controle remoto (NEUMANN, 2007).

O controle destas senhas é realizado por aplicativos que as geram e as armazenam em um repositório. A implantação das senhas nos dispositivos é realizada manualmente durante o processo de manutenção periódica. Outro mecanismo de segurança é a separação lógica das redes, através de segmentação por VLANs, conforme a funcionalidade de cada segmento de rede. Esta separação garante a privacidade dos dados de forma que os acessos serão realizados apenas pelos dispositivos que compartilham a mesma VLAN (BROWN, 2005).

No nível de subestação, um *firewall* é utilizado para fornecer proteção à conexão do DSAS com os ambientes corporativos das empresas. Esta conexão tem como objetivo permitir o acesso dos equipamentos das equipes de manutenção durante processos de manutenção remota dos dispositivos do sistema de automação da subestação. Outro propósito é a conexão dos dispositivos de automação a centros de controle que estejam localizados externamente à empresa concessionária de energia. Usualmente, esta conexão é realizada por *Virtual Private Network (VPN)* para garantir a privacidade dos dados.

## 2.6 Ações de Segurança em Redes de Automação

No que se refere às ações que podem ser tomadas para garantir segurança em redes de automação, especialistas recomendam a realização das seguintes atividades (DERYNCK, 2004):

(1) Monitoramento: realizado de forma automática e contínua; (2) Detecção: reconhecimento de atividades diferentes dos padrões da rede; (3) Notificação: realizada em tempo

real e com capacidade de geração de alertas à entidade competente; (4) Proteção: neutralização e quarentena dos possíveis ataques; (5) Recuperação: executada de forma segura, em tempo de execução e após ataques bem-sucedidos.

Baseado nos passos descritos, algumas soluções para segurança podem ser adotadas, tais como:

1. Aumentar a quantidade de sistemas de monitoramento em tempo real da rede, como por exemplo, *Intrusion Detection System (IDS)* e *Intrusion Prevention System (IPS)*;

2. Instalação de *gatekeeper*, *firewall* ou outros dispositivos e ferramentas para isolar a rede de outros sistemas, como por exemplo, isolar a rede de subestação da rede corporativa;

3. Utilizar dispositivos com capacidade de realizar controle automático de mensagem MAC, mantendo os requisitos de tempo de uma operação em tempo real;

4. Desenvolver soluções especialistas em sistemas operacionais de rede projetados para fornecer proteção e integridade da rede, como por exemplo, serviços de gerenciamento e controle de políticas;

5. Realizar o controle de acesso a computadores e programas de forma a garantir o direito de permissão a quem realmente tem este direito.

## 2.7 Linguagem de Especificação e Descrição

O processo de desenvolvimento de software para funções críticas tem evoluído de forma expressiva nas últimas décadas. Na engenharia de software atual não é mais aceitável a utilização de procedimentos textuais que produzem ambiguidades durante a fase de especificação. Métodos formais são progressivamente mais aceitos, dentro de uma gama cada vez maior de segmentos industriais, como a melhor maneira de satisfazer essas necessidades. Além disso, à medida que o mercado industrial cresce, equipamentos de diferentes fabricantes devem ser capazes de se comunicar com os outros.

O processo de modelagem envolve a concepção dos elementos importantes para o sistema, e o relacionamento entre eles e a sua representação usando linguagens específicas e bem definidas. Um modelo adequado de sistema deverá apresentar determinadas características consideradas importantes, tais como (ITU-T, 1992):

1. Completude: todos os serviços e restrições requeridos pelo usuário devem ser definidas;

2. Consistência: os requisitos não devem ter definições contraditórias;

3. Não ter ambiguidade: não deve permitir diferentes interpretações para um mesmo requisito. Esta definição deve ser clara e precisa;

4. Ser verificável: existir facilidade de aplicação de técnicas de verificação.

Para desenvolver sistemas dentro dos padrões de qualidade e segurança exigidos, faz-se necessário que uma linguagem de especificação atenda aos critérios estabelecidos durante o processo de modelagem. A linguagem de especificação e descrição, do inglês *Specification Decription Language (SDL)*, é um método formal definida pelo ITU-T para especificar sistemas complexos e de aplicações em tempo real. A robustez do SDL é baseada na capacidade de descrever a estrutura, o comportamento e os dados de um sistema. Esta linguagem possui o seguinte conjunto de características (ITU-T, 1992):

a) **Padrão:** padronizada pelo ITU-T através das normas Z.100 e Z.105;

b) **Formal:** é uma linguagem que garante precisão e consistência no projeto, o que é fundamental para aplicações de missões críticas;

c) **Baseada em gráfico e símbolo:** O que proporciona clareza e facilidade de uso durante a implementação e documentação do projeto;

d) **Orientada a objeto:** A *Specification and Description Language (SDL)* suporta encapsulamento, polimorfismo e vínculo dinâmico. Entre estes aspectos, estende o conceito de classe orientada a objeto, inserindo objetos ativos tais como: sistemas, blocos e máquina de estados.

e) **Elevada capacidade de teste:** Possui um alto grau de verificação e de teste, resultado da aplicação do formalismo para as funções de paralelismo, de interfaces, de comunicação e de temporização.

O modelo teórico básico da SDL consiste de um conjunto de máquina de estado estendida que rodam de forma paralela. Essas máquinas são independentes entre si e se comunicam através de sinais discretos. Um sistema descrito em SDL é formado pelo seguintes componentes:

1. Estrutura: sistema, blocos, processos e procedimentos;

2. Comunicação: sinais, canais ou rotas de sinais. Opcionalmente, os sinais podem conter parâmetros;

3. Comportamento: são descritos pelos processos;

4. Dados: conjunto de dados abstratos, do inglês *Abstract Data Type (ADT)*;

5. Herança: descreve relações e especialização.

Na arquitetura da SDL, o sistema é formado por entidades denominadas de blocos, processos e procedimentos. Inicialmente, o sistema é decomposto em blocos funcionais e cada um deste pode ser fragmentado em sub-blocos, assim por diante, até que as funcionalidade dos últimos blocos sejam suficientemente simples. Os blocos são divididos em processos que se comunicam com outros processos através de mensagens que são chamadas de sinais. O processo pode ser considerado como uma tarefa que tem uma fila de mensagens aguardando receber mensagens de outras tarefas. Os procedimentos materializam o comportamento do processo. A Figura 8 mostra o detalhamento da decomposição de um sistema em uma especificação SDL.

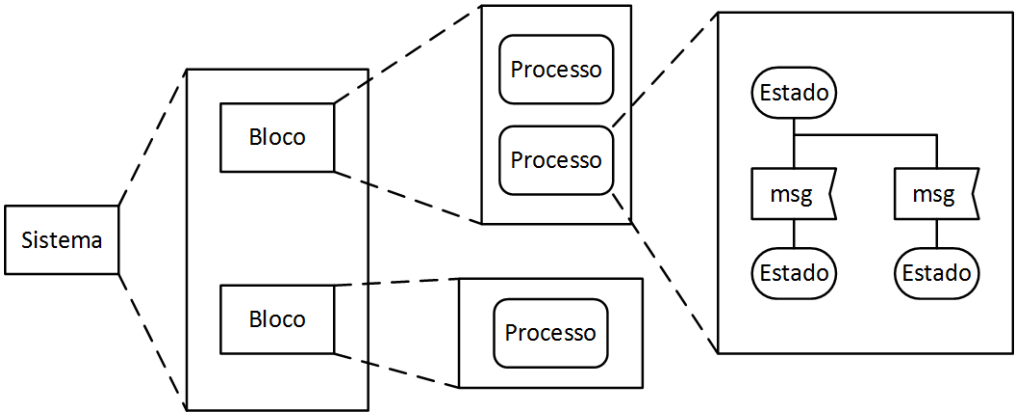


Figura 8 – Decomposição de um Sistema Especificado em SDL (Fonte: (ITU-T, 1992)).

A SDL possui dois mecanismos de comunicação: o primeiro é formado por sinais assíncronos e o segundo por chamadas de procedimento remoto síncronos. Esses mecanismos podem possuir parâmetros para intercâmbio de informações entre os processos SDL e o ambiente. Esta linguagem define interface entre os blocos e processos por intermédio de uma arquitetura que combina canais e rotas de sinais. Além disso, a SDL especifica temporizadores que são utilizados para descrever aspectos importantes de sistemas de tempo real distribuídos. Prioridades de processo e de sinais não estão definidos. Os sinais são enviados para uma determinada instância de processo, um de cada vez. Para descrever uma comunicação de propagação de sinais, o usuário deve implementar funções adicionais no pacote da SDL para possibilitar a utilização desse tipo de comunicação.

A hierarquia de sistemas e blocos é somente uma descrição estática da arquitetura do sistema. Os comportamentos dinâmicos destes são descritos pelos processos. O sistema pode ter uma ou mais instâncias de processo. Estes podem ser criados ou eliminados estaticamente durante a elaboração da especificação ou dinamicamente durante o tempo de execução. Cada instância é reconhecida por um identificador específico denominado de (PId). Isso possibilita enviar sinais de forma individual para cada processo. Eles são representados graficamente por uma máquina de estado finito, que descreve seu estado e comportamento.

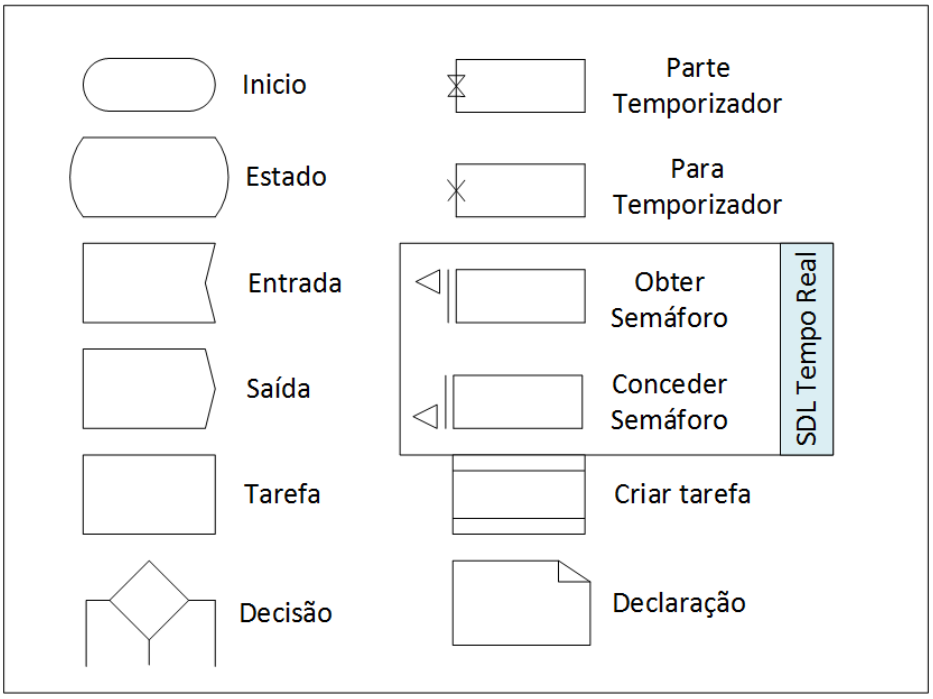


Figura 9 – Sumário dos Componentes de um Processo SDL (Fonte: ITU-T).

Os procedimentos de um processo que indicam o estado do processo, os eventos (mensagens), as decisões, as manipulações de temporizadores e de semáforos são representados por símbolos específicos. A Figura 9 apresenta um sumários desses símbolos. A SDL aceita os formatos de tipo de dados abstratos ADT e ASN.1. A integração do ASN.1 permite o compartilhamento de dados entre diversos tipos de linguagens e o reuso de estruturas de dados já existentes.

## 2.8 Notação de Teste e de Controle de Teste

A notação de teste e de controle de teste, *TTCN-3*, é uma linguagem de programação, que se destina especificar coleções abstratas de teste, *Abstract Test Suites* (ATS). Para realizar um caso de teste em uma ATS, é necessária a utilização de uma ferramenta para transformar o código abstrato em um conjunto de testes executáveis, do acrônimo em inglês, *Executable Test Suite (ETS)*.

A linguagem TTCN-3 tem como principal elemento o núcleo da linguagem. Este é independente do ambiente, do objeto sob teste e do tipo de teste que está sendo realizado (ETSI, 2016). O núcleo da linguagem suporta os diferentes tipos de teste denominados de módulo de teste, teste de integração e teste de conformidade.

O objeto sob teste pode ser uma biblioteca de funções escrita em uma determinada linguagem de programação. Para realizar os testes, é necessário implementar uma interface entre a ferramenta de teste e o sistema sob teste. A linguagem TTCN-3 não suporta testes em tempo real. Neste caso, os eventos não possuem tarja de tempo e não é possível a leitura

dos tempos absoluto e do sistema.

O TTCN-3 núcleo define uma entidade de software denominada de módulo. Este pode ser analisado, compilado e interpretado e tem a capacidade de conter um único ou vários casos de teste. Neste caso, um módulo pode ser usado como biblioteca de caso de teste para outros módulos. A linguagem não padroniza a forma como os módulos se relacionam entre si e nem como eles são armazenados nos arquivos.

Cada módulo é dividido em duas partes: a primeira é a de definição, que contém as definições de alto nível como: tipos e modelo de dados, constantes, portas, componentes, funções e os caso de teste. É possível importar as definições de outro módulo de forma a permitir a sua visibilidade pelo módulo importador. A segunda é a de controle, considerada como a função principal do módulo e a sua finalidade é realizar a chamada dos casos de testes estabelecidos na parte de definição. Além de conter a lógica de escalonamento da execução dos casos de teste, tem o poder de aplicar condicionantes a esta lógica de forma a controlar o escopo do teste. Ela também pode utilizar tipos específicos que são declarados na parte de definição nas variáveis locais do seu ambiente. A Figura 10 apresenta um exemplo da descrição de um módulo.

```
module MyModule
    {

        // Definition part
        import from otherModule all;
        type integer MyPosInt (0...Infinity);
        testcase tc\_myFirstCase() runs on MyComponent system MyTsi
        {
        ...
        }
        // Control part
        control
        {
        control
        }
           execute(tc\_myFirstCase(), 10.0);  //Maximum execution time 10.0 seconds
           execute(tc\_mySecondCase());  //No maximum execution time
        }

 }
```

Figura 10 – Exemplo de Descrição de um Módulo (Fonte: (ETSI, 2016) ).

O TTCN-3 núcleo possui um elemento de linguagem denominado de *testcase*. Este é responsável pela execução da função principal do teste (GRABOWSKI et al., 2003). Um caso de teste consiste de um *testcase* e de qualquer outra funcionalidade executada em paralelo

durante o teste. O *testcase* é sempre executado dentro de uma entidade denominada de componente, podendo chamar funções normais e do tipo *altsteps*, que estende seu comportamento. O resultado da execução de um *testcase* é o veredito. Este informa se o resultado do teste obteve êxito ou não.

Um caso de teste pode ser do tipo de avaliação de mensagem ou de procedimentos (ETSI, 2016). O primeiro caso consiste em enviar mensagens para o *System Under Test (SUT)*, receber as respostas e verificar se estas foram recebidas no tempo esperado, se a ordem está correta e os valores adequados. O teste baseado em procedimentos consiste em verificar o comportamento do sistema através da chamada de funções para receber valores de retorno e de exceções, passar valores e para forçar exceções para o SUT.

O TTCN-3 em sua arquitetura de teste define uma entidade específica, denominada de componente. Este contém os recursos necessários para realizar o teste, tais como: as portas de comunicação, as variáveis e os temporizadores. As portas são definidas pelo usuário e, por meio delas, é realizada toda interação com os outros componentes do SUT, utilizando-se de mensagens e procedimentos operacionais para realizar a troca de informações. Os componentes em si não especificam nenhum tipo de comportamento, apenas proporcionam o ambiente necessário para isso. Uma funcionalidade inicializada em um componente, que pode ser um *testcase* ou uma função, utiliza os recursos deste para executar o seu propósito.

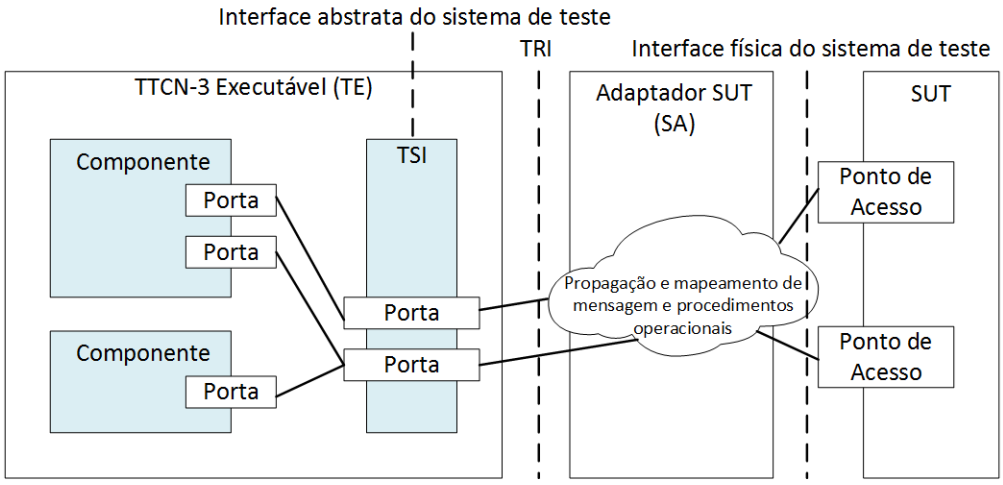


Figura 11 – Interfaces de um Sistema de Testes TTCN-3 (Fonte: (ETSI, 2016)).

Quando um caso de teste é especificado, é criado um componente denominado de componente principal de teste do inglês, *Main Test Component (MCT)*. Este é inciado e executado automaticamente. O caso de teste é concluído ao final da sua execução. O MCT pode criar e iniciar outros componentes denominados de componente de teste paralelos (*Parallel Test Components (PTC)*). Além destes, existe durante a fase de teste o componente chamado de interface do sistema de teste, (*Test System Interface (TSI)*). De forma

diferente dos outros, esta interface não pode inicializar nenhum tipo de funcionalidade e não tem variáveis internas e nem temporizadores.

Ela atua como uma interface abstrata entre o caso de teste e o SUT. As portas da TSI são visíveis às portas do SUT. Portanto, estas realizam o roteamento de qualquer mensagem ou procedimentos entre o caso de teste e o sistema sob teste. A Figura 11 detalha a arquitetura de teste e todas as interfaces entre o caso de teste e o sistema sob teste.

## 2.9 Considerações finais do capítulo

Neste capítulo foi apresentada sucintamente uma introdução aos sistemas de automação abordando suas arquiteturas, possíveis problemas e as ações de segurança que possam garantir o perfeito funcionamento desses sistemas. Também foram descritos alguns conceitos de segurança com ênfase nos tipos de ataques e mecanismos fundamentados em controle de acesso baseado em políticas. As linguagens de especificação e descrição SDL e a de notação e controle de teste TTCN-3 foram introduzidas. Esses tópicos tiveram como objetivo apresentar uma visão geral dos conhecimentos utilizados nessa Tese e facilitar o entendimento dos capítulos seguintes.

# 3

# REVISÃO DA LITERATURA CIENTÍFICA

Neste capítulo, a literatura relacionada com o tema da Tese é revisada, os trabalhos mais relevantes diretamente relacionados foram selecionados e comentados. Os estudos citados nesta Tese indicam que as análises atuais no contexto de segurança em redes de automação seguem as abordagens de autenticação de mensagem, análise de tráfego e controle de acesso. Uma taxonomia de métodos de defesa em redes de automação foi introduzida como uma contribuição desta revisão.

## 3.1 Principais estudos de segurança cibernéticas das redes de automação

O problema de segurança cibernética está sendo pesquisado e estudado em três linhas de abordagens diferentes: autenticação de mensagem, controle de acesso e análise de tráfego. A primeira abordagem é subdivida em código de autenticação de mensagem, assinatura digital, e aplicação de criptografia em toda mensagem. A segunda abordagem é subdividida em metodologia de controle de acesso discricionário, mandatório e baseada em papéis. A terceira abordagem estuda a aplicação de sistemas de detecção e prevenção de intrusão. Soluções agregando mais de uma tecnologia podem ser empregadas. A Figura 12 mostra a taxonomia das abordagens utilizadas nos estudos de segurança das redes de automação dos sistemas elétricos.

A metodologia de autenticação de mensagem é uma abordagem que garante a autenticidade e a integridade destas durante uma sessão de comunicação entre os diversos IEDs nas redes de automação. A premissa desta metodologia explora o fato de que, na comunicação em tempo real, não existe uma obrigatoriedade de confidencialidade (MOREIRA et al., 2016) . Neste caso, a autenticação das mensagens utiliza métodos que aplicam funções criptográficas. O processamento destas funções tem um custo computacional menor do

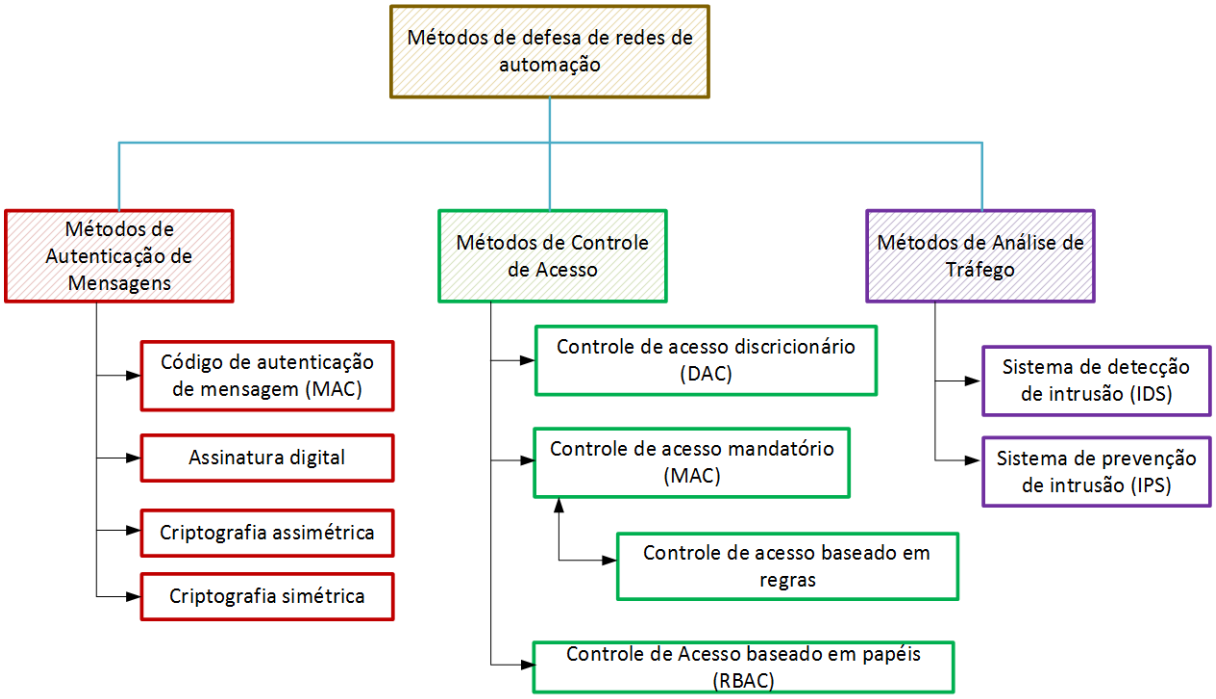


Figura 12 – Taxonomia de Métodos de Defesa em Redes de Automação (Fonte:Autor).

que o processamento de criptografia das mensagens, por exemplo, utilizar uma função para calcular um código MAC é muito mais ágil do que criptografar toda a mensagem.

A metodologia de controle de acesso visa garantir a autenticidade, a integridade e de forma indireta a confidencialidade das informações. O poder de acesso ao sujeito ou ao objeto detentor destes direitos permite que a escrita e a leitura dos dados sejam realizadas apenas por quem tem permissão de uso. A pesquisa realizada nessa Tese utiliza o conceito de controle de acesso baseado em papéis usando hardware seguro como metodologia de apoio ao processo de autenticação.

Metodologia de análises de tráfego realiza o monitoramento do tráfego da rede, e executa uma avaliação do seu comportamento com a finalidade de identificar padrão de tráfego que são produzidos por agentes maliciosos. Conforme os resultados desta análise tomam-se as medidas necessárias para a proteção da rede. Uma restrição desta metodologia é que a ação é tomada após a invasão. Neste caso, é difícil prever a extensão dos danos causados pelo código malicioso no período do início da invasão até a ação de bloqueio.

## 3.2 Estudos Baseados na Abordagem de Autenticação de Mensagens

Nesta seção, é apresentado o estado da arte de segurança cibernética utilizando a metodologia de autenticação de mensagens nos sistemas de automação. A busca de dife-

rentes métodos de autenticação de mensagem é pesquisada considerando-se a premissa de que o custo computacional das funções criptográficas é menor que o custo de encriptar a mensagem por inteiro. Mecanismos como a assinatura digital, código MAC associados a diferentes usos de distribuição de chaves são proposições apresentadas.

### 3.2.1 Estudos de Sugwon Hong, Dae-Yong Shin e Myongho Lee

O estudo de (HONG; SHIN; LEE, 2009) apresenta uma proposta de protocolo de segurança para garantir a autenticação e a integridade das mensagens de *SMV* e de mensagens *GOOSE* especificadas pela norma IEC 61850. O objetivo principal do artigo é avaliar se o desempenho de um protocolo de segurança apresentado está em conformidade com os critérios impostos por esta norma. Os autores propuseram o uso de MAC para resolver os problemas apontados.

Foram usados dois métodos para o cálculo do MAC. O primeiro, empregando cifra de bloco. Nesse método, a mensagem é encriptada utilizando o algoritmo *Advanced Encryptation Standard (AES)*, no modo de *Cipher Block Chaining (CBC)*. Nesse caso, todo texto cifrado é descartado, exceto o último bloco, que é o código MAC calculado. O segundo método foi o de *Hash-Based Message Authentication Code (HMAC)*.

O estudo considera o controlador de subestação como um dispositivo confiável e que mantém comunicação com todos os IEDs da rede de automação. Portanto, estabelece que o controlador é a entidade responsável pela distribuição de chaves. Neste contexto, foi especificada a utilização de duas chaves: uma chave de sessão ($K_{AB}$) para a comunicação do controlador com os demais IEDs da rede e outra ($K_E$) para realizar a encriptação das mensagens durante a distribuição da chave ($K_{AB}$). Além da definição das mesmas, o artigo apresenta um protocolo para a sua distribuição. As chaves iniciais são implantadas durante a inicialização do sistema.

Os experimentos foram conduzidos em duas plataformas: uma baseada em PC e outra composta por dispositivos embarcados. A plataforma de PCs é composta por dispositivos com as seguintes especificações: 4,3GHz de relógio, 1GB RAM, HDD 80GB e sistema operacional Windows XP SP2. A plataforma de sistemas embarcado é composta de dispositivos Xscale PXA270, rodando o sistema operacional Linux 2.6.12, com as seguintes características: 520MHz de relógio, 128MB de SDRAM e memória Flash de 64MB.

A arquitetura da plataforma PC é formada por uma rede de automação com quatro máquinas que têm como função simular um dispositivo eletrônico inteligente (IED), uma *Integrated Merging Unit (IMU)*, um dispositivo de atuação e um sistema de controle de subestação (nível II).

Os testes realizados usaram o algoritmo de criptografia *AES* e outro algoritmo denominado SEED, produzidos pela *Korea Internet & Security Agency (KISA)*. Além disso, utilizou-se o método de compressão de mensagem (HMAC). Nesse estudo, foram realiza-

dos vários experimentos empregando diversas combinações dos métodos de criptografia e de funções *hash* criptográficas. A Tabela 2 apresenta os resultados encontrados.

O estudo comprova que a utilização de criptografia das mensagens em simultaneidade com o cálculo HMAC acarreta um aumento significativo do tempo de processamento. O melhor desempenho analisado foi aplicando o método MD5, com um tempo de 2,82750 us, seguido pelo HMAC-MD5 com o tempo de 4,68950 us. Observa-se que, nesses casos, as mensagens não são encriptadas e esses métodos garantem apenas a integridade. No sistema usando o Xscale, notou-se que houve perda de pacotes e incapacidade de processamento para arquitetura com um número de *merging units (MU)* superior a dez unidades.

A Figura 13 apresenta esta perda. Percebe-se que a plataforma de PC, além de atingir o requisito de tempo de transmissão para mensagens menor que 3ms (IEC, 2003), conseguiu processar todos os pacotes recebidos.

Tabela 2 – Tempo de Encriptação para a Plataforma PC (Fonte: (HONG; SHIN; LEE, 2009)).

| Criptografia | MAC | Tempo (us) |
|---|---|---|
| AES256 | - | 5,38400 |
| AES256 | SEED | 11,79450 |
| AES256 | MD5 | 7,23000 |
| AES256 | HMAC-MD5 | 9,15900 |
| SEED | - | 5,36000 |
| SEED | AES256 | 10,13950 |
| SEED | MD5 | 8,54600 |
| SEED | HMAC-MD5 | 11,66550 |
| | MD5 | 2,82750 |
| | HMAC-MD5 | 4,68950 |

### 3.2.2 Estudos de Frank Hohlbaum, Markus Braendle e Fernando Alvarez

O estudo de (HOHLBAUM; BRAENDLE; ALVAREZ, 2010) tem como objetivo realizar uma avaliação do desempenho da rede, mediante a implantação dos mecanismos de segurança recomendados pela norma *Power Systems management and associated information exchange - Data and Communication Security"* IEC 62351-X. A pesquisa visa também verificar os retardos de tempo decorridos durante a transmissão de mensagens GOOSE e SMV.

Os autores definiram quatro cenários de teste para realizar a análise das comunicações entre os dispositivos de uma rede de automação, empregando o protocolo IEC 61850. O primeiro utiliza uma plataforma PC para simulação de dispositivos de uma rede de

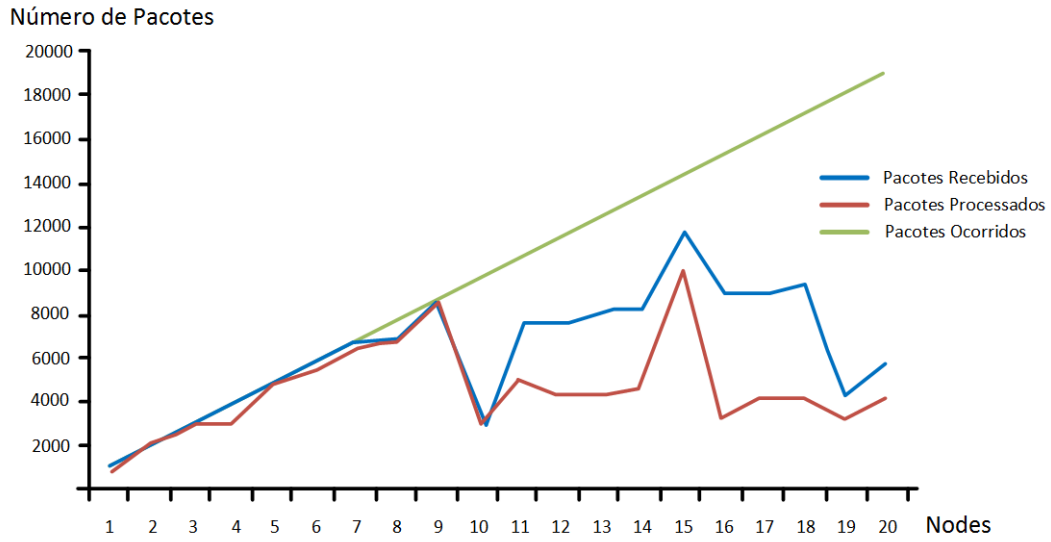


Figura 13 – Perda de Pacotes em Função do Número de MU (Fonte: (HONG; SHIN; LEE, 2009)).

automação. O segundo cenário faz a avaliação sobre uma plataforma formada por dispositivos *Field Programmable Gate Array (FPGA)* para simular os IEDs. No terceiro, os mecanismos de segurança são implementados em dispositivos constituídos por *Application Specific Integrated Circuits (ASIC)* e o quarto abrange a utilização de RSA[1] *crypto chips* nos processos de encriptação da autenticação das mensagens dos IEDs.

A norma IEC 62351 especifica para as mensagens GOOSE e SMV que o mecanismo de segurança é operacionalizado utilizando assinatura digital. Esta assinatura é realizada executando a encriptação com o algoritmo de criptografia pública RSA, a partir do resultado de uma função *hash* criptográfica da mensagem SHA256. Para a plataforma PC, as soluções apresentadas para os cálculos, usando a mesma metodologia, indicaram um custo computacional de 1,5 ms que representa a metade do requisito do tempo mínimo de transmissão de uma mensagem GOOSE. A Tabela 3 exibe os resultados do custo de tempo para a operação da assinatura da mensagem, empregando chaves de 1024 bits e 512 bits para duas configurações de PCs diferentes.

Na plataforma utilizando FPGA, o custo de tempo das operações criptográficas para o cálculo da assinatura da função hash criptográfica com o RSA revelou valores da ordem de 2ms a 4ms, para 512 e 1024, bits respectivamente. Neste caso, apesar do resultado apresentado ter um valor mínimo de 2ms, os autores do artigo consideram que o tempo de resposta nessa plataforma não é adequado devido ao período restante ser insuficiente para o processamento das demais funções do IED. A Tabela 4 apresenta os valores de tempo de mensagens GOOSE utilizando mecanismos de segurança implementados em FPGA.

[1] Sobrenome dos autores de um algoritmo de criptografia de chave pública desenvolvido no (MIT): Ronald Rivest, Adi Shamir e Leonard Adleman,

Tabela 3 – Custo de Tempo na Operação de Assinatura Digital usando RSA (Fonte: (HOHLBAUM; BRAENDLE; ALVAREZ, 2010)).

| Tamanho da Chave | Pentium M 1,7 GHz (1GB RAM) | Intel Core 2 Duo @ 2,2 GHz(2 GB RAM) |
|---|---|---|
| 1024 | 6,8 ms | 4,0 ms |
| 512 | 3,9 ms | 1,5 ms |

Tabela 4 – Tempo de GOOSE Utilizando Mecanismos de Segurança em FPGA (Fonte: (HOHLBAUM; BRAENDLE; ALVAREZ, 2010)).

| Relógio do FPGA | GOOSE Enviado | GOOSE Recebido | Total do Processamento |
|---|---|---|---|
| 100 MHz | 3,748 ms | 0,155 ms | 3,903 ms |
| 200 MHz | 1,917 ms | 0,129 ms | 2,036 ms |

Na plataforma ASIC, os melhores resultados encontrados, considerando uma chave de 1024 bits e o algoritmo de assinatura RSA, foram 0,16ms, 0,34ms e 0,98ms, dependendo do tipo de prioridade de mensagens definidas pela norma IEC 61850. Esta solução atende aos requisitos das mensagens GOOSE, mas não atende aos valores de mensagens amostradas. Por exemplo, no caso de classe de mensagens de medição de desempenho M3, a norma IEC 61850 define uma taxa de amostragem de 12.000 Hz. Essa taxa exige um tempo de transmissão mínimo de 83us, muito inferior aos valores encontrados.

A plataforma utilizando RSA *crypto chips* faz uso dos parâmetros de criptografia com o comprimento de chave de 1024 bits e o algoritmo RSA, para realizar a assinatura das mensagens. Neste caso, verificou-se um desempenho de 23,8us, que é suficiente para atender aos requisitos das mensagens de classe de desempenho M3, conforme requisitos da norma IEC 61850 parte 5, (IEC, 2003). Os autores consideram que a inclusão desse tipo de chip no IED causará a modificação total em seu projeto ficando inviável para uma solução em curto prazo.

### 3.2.3 Estudos de Rainer Falk, e Steffen Fries

O estudo de (FALK; FRIES, 2013) propõe um sistema de segurança para rede de automação baseada em redes *multicast*. A arquitetura apresentada visa garantir a autenticidade da origem da mensagem e a integridade dela durante o percurso de transmissão. O artigo considera que o modelo de comunicações publicação/assinatura especificado pela norma IEC 61850 é um caso particular de redes *multicast*. Neste caso, o sistema de segurança foca na proteção das mensagens GOOSE e de SMV, utilizando o conceito de assinatura

digital, especificado na mesma norma.

O custo computacional do uso da criptografia assimétrica é considerado elevado para os processadores dos IEDs. Como solução, o estudo adotou o uso de criptografia simétrica para cálculo do código de autenticação das mensagens. Isso implica em outro problema, que é a distribuição de chaves de modo seguro na rede *multicast*. Neste caso, a chave é compartilhada por um grupo de usuários. Este fato ocasiona a impossibilidade de garantir a autenticidade do emitente. Para resolver esta questão, o artigo propõe a utilização do protocolo *Timed Efficient Stream Loss-tolerant Authentication (TESLA)*.

O protocolo TESLA emprega a criptografia simétrica para geração de um código MAC, usando uma cadeia de chaves, que é gerada pela entidade emissora. O envio das mensagens é realizado em intervalos de tempo específicos. A distribuição de uma chave associada a uma determinada mensagem ocorre em um período posterior ao intervalo de transmissão. As chaves geradas, após a janela de tempo de transmissão das mensagens, são consideradas inválidas. A Figura 14 apresenta um diagrama que mostra a assimetria no processo de gerenciamento de chaves escalonadas no tempo.


Figura 14 – Conceito de Distribuição de Chaves Pelo Protocolo TESLA (Fonte: (FALK; FRIES, 2013))

O retardo entre o emissor e o receptor proporciona uma assimetria entre o tempo de transmissão de mensagens e o de distribuição da chave. Isto é crítico para mensagens de sistemas em tempo real, principalmente para aquelas com classe de desempenho 1A (IEC, 2003). Para manter o sincronismo entre o transmissor e o receptor, faz-se necessário manter os relógios em fase no tempo. Este procedimento pode ser local, através de um sistema global de posicionamento (GPS) ou por intermédio de um servidor de tempo utilizando o protocolo *Network Time Protocol (NTP)* ou *Precision Time Protocol (PTP)*.

No processo de criação dos elementos da cadeia de chaves $K_0, K_1, ..., K_N$, o emissor envia inicialmente $K_{init}$ para o receptor de forma segura, usando assinatura digital. Esta cadeia é gerada pela função $K_{(i-1)} = H(K_i)$. Neste contexto, uma chave $k_i$ pode ser verificada aplicando-se uma função hash criptográfica à chave $K_i$ e comparando o valor *hash* com a chave $k_{i-1}$. Neste processo, inicialmente se calcula a chave $K_{init}$ por uma função pseudoaleatória e, em seguida, esta é distribuída para os assinantes no tempo $t_{(n-1)}$. Cada assinante (IED) deve possuir a chave pública Kpu do emissor.

Para resolver o problema da criptografia de chave pública durante a distribuição da chave $k_{init}$ do protocolo TESLA, os autores propõem uma versão modificada, denominada uTESLA, que utiliza uma estação base com conexão autenticada com os IEDs, usando criptografia simétrica. No caso específico de redes de automação apresentadas no

artigo, este papel é assumido pelo controlador da subestação. As demais características do uTESLA são similares ao protocolo TESLA.

Os autores consideram que o TESLA fornece uma solução para autenticação com atraso, permitindo que um IED execute uma ação dedicada em tempo real e realize a verificação de segurança associada posteriormente. É óbvio que há um período de incerteza entre a recepção e a verificação de uma mensagem, o que as torna inadequadas para o tráfego de controle que requer um tempo de reação muito curto (por exemplo, desligamentos de emergência em caso de sobrecarga) para ações que podem não ser reversíveis.

Assim, há basicamente uma negociação se a reação imediata a um comando for mais importante do que a autenticação do remetente. Também é possível suportar diferentes esquemas de autenticação *multicast* dentro de uma solução de segurança. Um exemplo é usar as metodologias descritas somente para mensagens críticas, isto é, com reação imediata da ordem de comando, enquanto para as outras mensagens, o ideal é usar a abordagem convencional para realizar a verificação da mensagem antes de executar a operação do seu conteúdo, independente do retardo apresentado.

### 3.2.4 Estudos de Wang Fangfang, Wang Huazhong, Chen Dongqing e Pen Yong

O estudo de (FANGFANG et al., 2013) propõe um esquema de criptografia misto, combinando o algoritmo de criptografia simétrica, do acrônimo em inglês, *Data Encryption Standard (DES)* e o algoritmo de criptografia assimétrica RSA. A ideia é explorar as vantagens que cada algoritmo apresenta individualmente e mitigar os problemas gerados por cada um deles, isoladamente. Além disso, o esquema tem a finalidade de melhorar a segurança das mensagens nos sistemas de automação de subestação, atendendo aos requisitos de tempo impostos pela norma IEC 61850. O conceito básico dessa solução é garantir a integridade, a confidencialidade e a autenticidade das informações.

A utilização de um único esquema de criptografia ocasiona problemas em que o custo de tempo das operações criptográficas, associado ao tempo de processamento e transmissão de mensagens, não atende ao requisito de tempo real das aplicações de uma subestação. No esquema proposto, o emissor gera uma chave compartilhada, e em seguida, a mensagem é encriptada por esta. Na etapa seguinte, a chave compartilhada é encriptada pela chave pública do receptor, que é posteriormente destruída. A mensagem citada é transmitida em conjunto com a chave compartilhada cifrada.

No lado do receptor, o processo de recuperação da mensagem realiza, na ordem inversa, os mesmos passos executado no lado transmissor. Isto é, primeiro efetua a decriptação da chave compartilhada, utilizando a chave privada do receptor e, em seguida, realiza-se a decriptação da mensagem usando a chave compartilhada. Nesse esquema, foi escolhido o algoritmo DES devido à sua característica de manter o comprimento da mensagem

invariável durante o processo de encriptação.

Por outro lado, observa-se a vulnerabilidade no DES em função do comprimento da chave. A escolha desta com um comprimento pequeno pode facilitar a sua descoberta, sem um grande esforço computacional. Assim sendo, a solução adotada foi a inclusão do algoritmo RSA para encriptar a chave utilizada pelo algoritmo DES.

O artigo analisa o desempenho do esquema empregando o simulador de redes Opnet. Nesta avaliação, foi considerado um cenário com uma subestação do tipo D2-1 (subestação de distribuição de porte médio), composta pelos seguintes *bays*: dois transformadores, seis de linhas de transmissão e um de transferência. Os dispositivos de automação de cada *bay* têm a seguinte distribuição: o transformador é composto de duas *merging units (MU)*, uma unidade de controle principal e duas unidades de proteção de transformadores. O de linha é formado por uma *merging unit (MU)*, duas unidades de proteção de linha e uma unidade de proteção diferencial.

O de barra é composto de uma *merging unit* e uma unidade de proteção de barra. Todos os *bays* são equipados com dispositivos de manobra que têm a capacidade de executar chaveamentos automáticos. A Figura 15 detalha o ambiente de teste de uma subestação do tipo D2-1. Para a análise do desempenho, os autores usaram uma topologia de rede

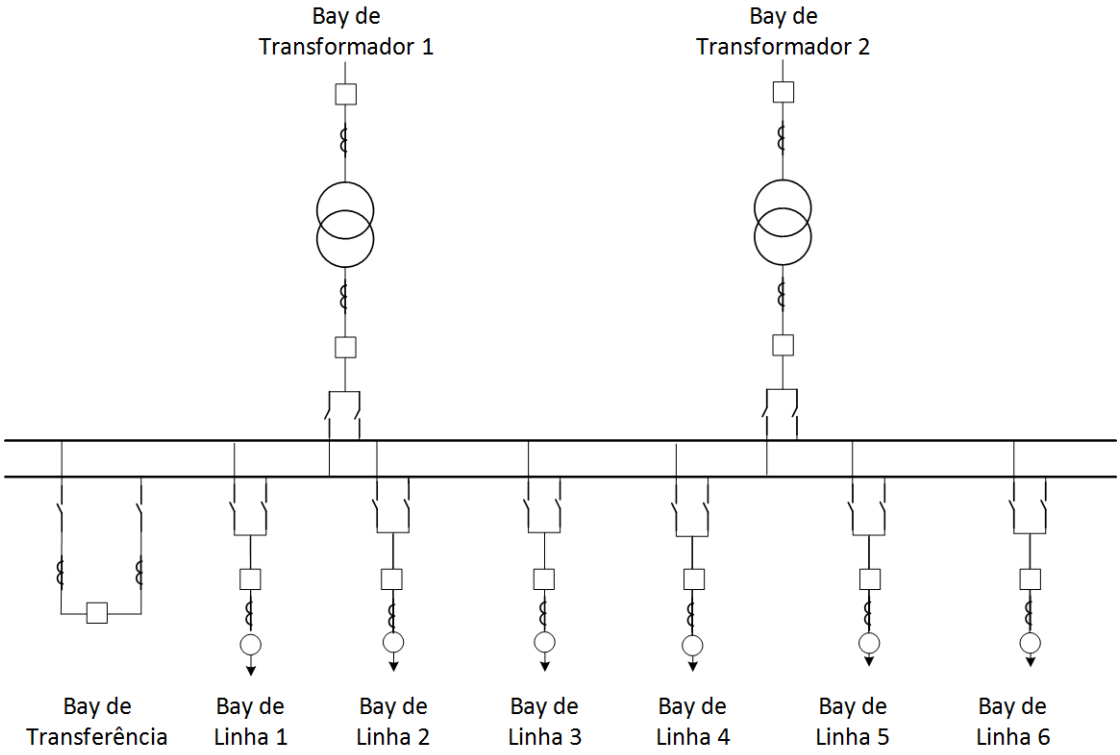


Figura 15 – Cenário de Teste de uma Subestação do Tipo D2-1 (Fonte: (FANGFANG et al., 2013))

em estrela e outra topologia em anel, analisando o comportamento destas nas velocidades de 10Mbps e 100Mbps, durante a simulação de uma perturbação no sistema elétrico. Os

parâmetros de configuração da MU estão de acordo com a classe de desempenho P3 da norma IEC 61850, que prevê um comprimento da mensagem de 113 bytes e em uma frequência de amostragem de 2400Hz.

Os equipamentos de manobras inteligentes enviam mensagens de comprimento de 130 bytes, contendo a localização do defeito do sistema elétrico e o estado de disjuntor. Os autores analisaram o atraso das mensagens GOOSE para uma falta na subestação, considerando o esquema de segurança proposto. Os resultados apresentam um aumento considerável do retardo para a topologia em anel. Esta situação é decorrente do fato de que na topologia em anel, a mensagem é repetida por cada nó até chegar ao seu destino final, enquanto na topologia em estrela, a mensagem é transmitida diretamente ao seu destino.

A simulação considerou o percurso fim a fim das mensagens, logo, o tempo total é composto pelos seguintes segmentos por ordem de ocorrência: $T_{d1}$ , $T_{r1}$, $T_{dly}$, $T_{d2}$ e $T_{r2}$. A Equação 3.1 calcula $T_{d1}$ e $T_{d2}$, que são os retardos decorrentes da encriptação e decriptação do DES. Esses cálculos consideram a velocidade de cifragem do DES de 288,44 Mbits/s no FPGA utilizado no estudo. Os retardos $T_{r1}$ e $T_{r2}$, calculados pela Equação 3.2, são o tempo gasto na encriptação e decriptação do RSA, considerando a velocidade de cifragem do RSA no FPGA de $13,19 Mb/s$. A Equação 3.3 realiza o cálculo do tempo total de transmissão da mensagem incluindo os tempos de encriptação e decriptação. A Tabela 5 sumariza esses resultados.

$$T_{d1} = T_{d2} = \frac{B_{ptext}}{V_c} \tag{3.1}$$

$$T_{r1} = T_{r2} = \frac{B_{Dkey}}{V_d} \tag{3.2}$$

$$T = \sum_{i=0}^{n} T_i = T_{d1} + T_{r1} + T_{dy} + T_{d2} + T_{r2} \tag{3.3}$$

Os autores apresentam as seguintes vantagens do esquema criptográfico empregando os algoritmos de criptografia híbrida:

a) Não se faz necessário um esquema adcional de gerenciamento de distribuição de chaves. Estas são distribuídas em conjunto com a mensagem;

b) O esquema apresenta uma segurança dupla decorrente do fato de que as mensagens GOOSE são encriptadas pela chave do DES e, sendo esta encriptada pelo algoritmo de chaves assimétricas do RSA;

c) O algoritmo RSA limita o comprimento do texto cifrado ao tamanho do texto plano. Isso tem como consequência a diminuição dos tempos de encriptação e decriptação no esquema proposto.

Tabela 5 – Resultado das Simulações para o Cenário de Teste (Adaptada: (FANGFANG et al., 2013))

| Retardo | Velocidade | Estrela | Anel |
|---|---|---|---|
| Rede | 10 Mbps | 1418 us | 1690 us |
| | 100 Mbps | 91 us | 435 us |
| DES | | 3,4 us | 3,4 us |
| RSA | | 73,56 us | 73,56 us |
| Total | 10 Mbps | 1,844 ms | 1,572 ms |
| | 100 Mbps | 0,245 ms | 0,589 ms |

Por fim, os autores consideram que o uso dos algoritmos de criptografia DES e RSA de forma híbrida é uma solução viável para alcançar o requisito de 4 milissegundos para mensagens GOOSE e SMV imposto pela norma IEC 61850 em sistemas de automação de subestação.

## 3.3 Estudos Baseado na Abordagem de Análise de Tráfego

Nesta seção, é retratado o estado da arte da abordagem da análise e de tráfego apresentando diferentes metodologias de sistemas de detecção e prevenção de intrusão. Além disso, são abordados os problemas apresentados na aprendizagem dos IDS e as questões de falso positivo e falso negativo.

### 3.3.1 Estudos de Junho Hong, Chen-Ching Liu e Manimaran Govindarasu

O estudo de (HONG; LIU; GOVINDARASU, 2014) propõe um IDS baseado em rede para proteger as comunicações dos sistemas de automação das subestações. O IDS tem a capacidade de detectar as anomalias do tráfego de redes que apresentam comportamentos anormais em um ambiente de tempo real. O IDS proposto tem como foco principal detectar irregularidades através da análise do tráfego das mensagens. No caso do protocolo IEC 61850, as mensagens GOOSE e SMV utilizam o tipo de comunicação publicador/assinante que tem características *multicast*.

O estudo propõe a utilização de algoritmos de detecção baseados em especificações lógicas que oferecem suporte à detecção de intrusão fundamentada em redes. Este tipo de algoritmo identifica o desvio de um perfil do comportamento normal de operação.

A Figura 16 mostra a arquitetura desse tipo de IDS que utiliza a abordagem de lista branca *(whitelist)*. Com a finalidade de treinar esses equipamentos, é necessária uma verificação para checar se os dados do sistema correspondem ao seu comportamento no

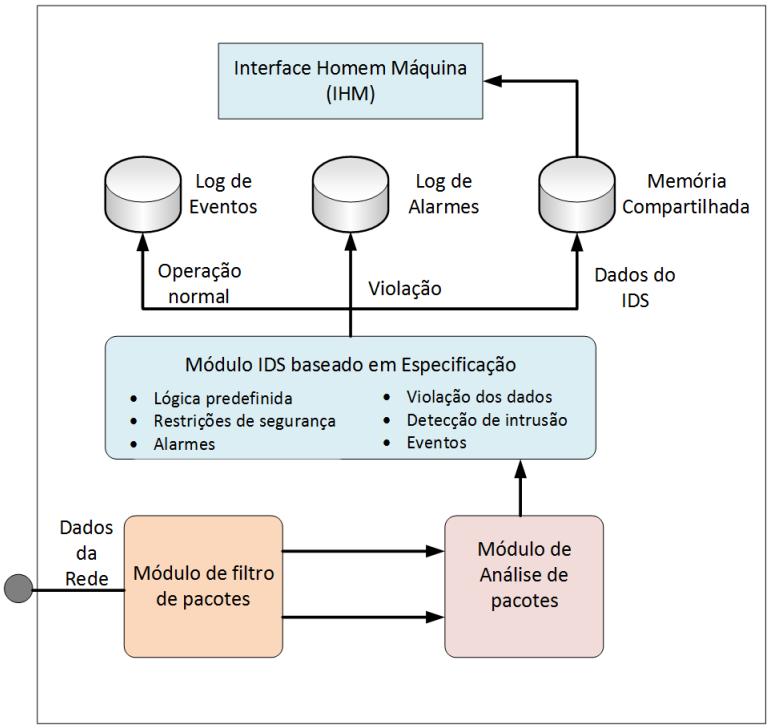


Figura 16 – Módulo de IDS Baseado em Especificação (Fonte: (HONG; LIU; GOVINDARASU, 2014)).

estado de operação normal ou correto. Para isso, o IDS realiza um monitoramento em tempo real das atividades do tráfego da rede de automação para verificar se há alguma violação aos limites que foram previamente estabelecidos.

Os pesquisadores consideram que a detecção baseada em especificação tem suas limitações, visto que o IDS necessita de um projetista de sistema de segurança que possua um elevado grau de conhecimento do comportamento da rede de automação, para definir a lógica de especificação.

A arquitetura do IDS segue o seguinte fluxo: um módulo de filtragem de pacotes que os recebe a partir da rede do sistema de automação da subestação. Em seguida, o módulo de filtro que só permite a passagem de mensagens do tipo GOOSE e SMV rejeitando as demais. Neste caso, a carga de processamento é reduzida, tendo como consequência uma melhoria no desempenho do sistema. O módulo analisador de pacotes irá extrair os pacotes GOOSE e SMV das três camadas inferiores da rede. Após isso, o módulo principal do IDS realiza uma detecção de intrusão através da análise comparativa do comportamento dos pacotes com as regras e lógicas anteriormente definidas. Após esta interpretação, o IDS calcula o índice de vulnerabilidade $V_n$ que é descrito pela Equação 3.4.

$$V_n = \begin{cases} 1, se\ \beta^G = verdade \\ 1, se\ \beta^{SV} = verdade \\ 0,\ de\ outra\ forma \end{cases} \tag{3.4}$$

Na Equação 3.4, $\beta^G$ e $\beta^{SV}$ são os indicadores de intrusão calculados pelo IDS. Qualquer violação detectada pelo IDS para as mensagens GOOSE e/ou SMV muda os valores $\beta^G$ e $\beta^{SV}$ modificando de estado para o nível lógico um, indicando que houve uma detecção de anormalidade. Quando $V_n$ está no nível lógico zero, ele assinala que não há evidência de invasão na rede. O pseudo código descrito na Tabela 6 detalha as regras de detecção de intrusão que são tomadas como base para a verificação da existência ou não de uma anormalidade no tráfego das redes de automação.

Tabela 6 – Exemplo de Regras de Segurança Pré-Definidas para o IDS

| Tipo | Regras IDS | Indicador de Violação |
|---|---|---|
| GOOSE | $if \alpha^G_{data,i} \neq \alpha^G_{data,i+1} then$ | |
| | $if(\alpha^G_{st,i+1} > \alpha^G_{st,i}) \vee (\alpha^G_{sq,i+1} == 0)$ | |
| | $\pi^G_{data} = 1$; return (A) | $\pi^G_{data}$ |
| | $else \pi^{SV}_{data} = 0$; | |
| | $endif$ | |
| SMV | $if C^{SV}_{pkt,1sec} > \alpha^{SV}_{th}$ | |
| | $\pi^{SV}_{SP} = 1$; return | $\pi^{SV}_{data}$ |
| | $else \pi^{SV}_{SP} = 0$; | |
| | $endif$ | |

A norma IEC 61850-5 estabelece que o estado da mensagem GOOSE $\alpha^G_{st,i}$ será incrementado e, simultaneamente, o número de sequência $\alpha^G_{sq,i}$ será setado para zero, quando houver a mudança de estado do dado. Neste caso, $i$ é o $i$-ésimo número de sequência que corresponde ao estado das regras apresentadas na Tabela 6. Se o número de sequência de uma mensagem GOOSE for diferente de zero após a variação de estado do dado, significa que houve uma anomalia. Esta pode ter sido provocada através de uma modificação ou de uma injeção de pacotes na rede da subestação. Neste caso, o indicador $\pi_data^G$ mudará de estado zero para um. A especificação IEC 61850 classifica as mensagens SMV por classe de desempenho P1, P2 e P3 para proteção e M1,M2 e M3 para medição, conforme a norma IEC 61850 parte5, (IEC, 2003) . Dessa forma, também define o número de bits do conversor analógico/digital para estas classes. Fundamentado nestes critérios, o IDS analisa se em uma mensagem SMV o número total de pacotes por segundo $C^{SV}_{pkt,sec}$ é maior que um determinado limiar $\alpha^S_{th}V$. Este fato caracteriza uma anormalidade. Esta pode ser causada por um ataque de negação de serviço, do acrônimo em inglês, *Denial of Service (DoS)* à rede da subestação. O indicador de violação $\pi^S_{th}V$ mudará de estado zero para um, sinalizando que houve uma anomalia. O valor deste é calculado pela Equação 3.5.

$$\alpha^{SV}_{th} = (\sum_{j}^{m} \alpha^{SV}_{sr,j} \times f_j) \times (1 + \delta^{SV}_{me}) \tag{3.5}$$

O parâmetro $m$ é o número de *merging unit (MU)* do sistema, e $\alpha_{sr,j}^{SV}$ é a resolução da amplitude do conversor analógico digital em bits, $f_j$ é a frequência da j-ésima *merging units* e $\delta_{me}^{SV}$ é a margem de erro que usualmente na prática é de 20%. Como validação da proposta, o estudo utiliza um ambiente de teste que explora vários tipos de ataques cibernéticos. As consequências desses ataques sobre o comportamento do sistema na ausência do IDS são apresentadas na Tabela 7.

Tabela 7 – Consequência dos Comportamentos Maliciosos de GOOSE e SMV sem o IDS (Fonte: (HONG; LIU; GOVINDARASU, 2014)).

| | Intrusão | Resultados |
|---|---|---|
| GOOSE | Ataques de repetição | Abre o disjuntor |
| | Modificação do tempo de transferência | Alarme no disjuntor |
| | Modificação de GOOSE de Controle | Abre o disjuntor |
| | Ataque de Negação de Serviço | Perde disponibilidade de IED |
| | Geração de GOOSE de Controle | Abre o disjuntor |
| SMV | Modificação dos Valores Medidos | Abre o disjuntor |
| | Modificando *dataset* do SMV | Alarme no IED |
| | Ataque de Negação de Serviço | Perde disponibilidade de IED |
| | Geração de dados de SMV | Abre o disjuntor |

O estudo realiza uma simulação na qual o estado inicial é a condição normal de operação seguido de ataques usando os métodos de repetição de GOOSE, inserção de SMV e de GOOSE de controle, em diferentes locais da rede de automação, na topologia do cenário de teste. Nesse, o IDS percebe as intrusões e gera os respectivos alarmes da detecção das invasões.

O artigo apresenta os resultados do número de falso negativos (FNR) e de falso positivos (FPR) para os pacotes maliciosos, considerando um intervalo de 1, 10, 20 e 30 ms para a chegada destes durante o ataque. Estes resultados estão listados na Tabela 8.

Os autores verificam um aumento desses valores para o caso de diminuição dos intervalos de tempo entre pacotes. Isso é devido ao fato de que o IDS pode perder pacotes, em razão desse intervalo de tempo ser menor de que o tempo necessário para processamento dos pacotes. Para realizar avaliação de detecção de ataques simultâneos em múltiplas subestações, o artigo utilizou um cenário considerando o padrão 39 barras do IEEE. A simulação inicia com as subestações em condições normais de operação, seguindo por ataque que é realizado utilizando os intervalos de tempo 5, 10 e 15 ms entre pacotes

Tabela 8 – Erros de Falso Negativo e de Falso Positivo Apresentados Pelo IDS Sob Ataques Cibernéticos (Fonte: (HONG; LIU; GOVINDARASU, 2014)).

| Atraso (ms) | FNR | FPR |
|---|---|---|
| 1 | 7,72x10-4 | 4,66x10-4 |
| 10 | 5,91x10-4 | 3,64x10-4 |
| 20 | 3,05x10-4 | 2,0x10-4 |
| 30 | 2,29x10-4 | 1,61x10-4 |

e modificando os métodos de ataques entre repetição e inserção. As soluções dos testes encontrados no estudo são apresentadas na Tabela 9.

Tabela 9 – Resultado de Ataque Simultâneo a um Sistema de 39 Barras do IEEE (Fonte: (HONG; LIU; GOVINDARASU, 2014)).

| Tempo (Segundos) | Subestações Alvos | Método de ataque | Resultado de operação do IDS |
|---|---|---|---|
| 5 | 2,25,30,37 | Repetição de pacotes GOOSE | Detecção/Alarme |
| 10 | 24,28 | Geração de SMV (Valor elevado de corrente) | Detecção/Alarme |
| 15 | 29,38 | Geração de controle via GOOSE | Detecção/Alarme |

A pesquisa concluiu que a metodologia de monitoramento de detecção de intrusão simultânea é capaz de identificar ataques em uma ou várias subestações. Esses mecanismos foram validados em ambientes de teste de intrusão realístico, onde o sistema apresentou resultados satisfatórios para os ataques de reprodução, modificação, homem no meio, geração de pacotes e negação de serviço. Os autores também consideram que o número de falso positivos e falso negativos está dentro do padrão esperado. Por outro lado, observa-se que o algoritmo de detecção de intrusão fundamentado em rede precisa ser atualizado periodicamente, uma vez que não é capaz de detectar ataques desconhecidos que não estejam definidos na sua base de informações.

## 3.4 Estudos Baseados na Abordagem do Controle de Acesso

Nesta seção, é apresentada a metodologia de segurança cibernética em redes de automação de sistemas elétricos.

### 3.4.1 Estudos de Binod Vaidya, Dimitrios Makrakis e Hussein Mouftah

O estudo apresentado por (VAIDYA; MAKRAKIS; MOUFTAH, 2013) é uma solução de segurança baseada em uma arquitetura que utiliza um esquema de autenticação e autorização multifatorial e multinível baseado em *Elliptic Curve Cryptography (ECC)* e certificados intrínsecos. O esquema faz uso de protocolo de conhecimento nulo assistido por servidores (SAV) e explora o uso de certificados de atributos (AC) para os serviços autorizados.

Os autores evidenciam que, nas redes SCADA atuais e na maioria das arquiteturas dos DSAS em operação, os IEDs utilizam um modelo de segurança com base em níveis de permissão protegidos por usuário e senha que controlam o acesso aos seus diferentes níveis de funcionalidade, leitura de dados e de alteração de configurações.

O artigo apresenta um panorama no qual um servidor de acesso remoto autentica um usuário de forma a permitir um estabelecimento de comunicação com os IEDs. A arquitetura de segurança proposta pelo artigo é formada por um *Smart Substation Controller (SSC)*, usado como o ponto central do esquema de controle de acesso aos IEDs. O controlador de subestação, além de realizar a função de comando e supervisão, deverá proporcionar serviços de autenticação e autorização para o usuário que deseja obter acesso ao sistema de automação. Embora a norma IEC 62351 especifique explicitamente o RSA como uma solução para proteger mensagens nos DSASs, o conceito de ECC tem atraído cada vez mais atenção nas redes SCADA, uma vez que tem vantagens sobre o RSA em termos de requisitos de comprimentos de chave e tempo de processamento.

Na abordagem proposta pelo artigo, o ECC usa criptografia assimétrica. O mecanismo de autenticação é composto pelas seguintes fases: inicialização, registro, autenticação e autorização.

Na fase de "inicialização e registro", uma autoridade confiável (TA) possui um par de chaves formado por uma chave privada (xTA) e uma pública (XTA). Esta autoridade distribui a sua chave pública para os componentes do DSAS. O usuário (UA) tem o papel do sujeito que precisa provar sua identidade, o IED (UTR) que está localizado na subestação tem a função de verificador e o SSC tem a função de *gateway*. Ainda nessa fase, o usuário tem que obter com a autoridade certificadora de confiança TA uma senha específica do IED que deseja acessar. A autoridade está localizada em um centro de controle e a senha funciona como um segredo compartilhado. Durante a fase de registro, o usuário cria uma senha que fornece ao TA para que ele gere, de modo seguro, um *token*, contendo um certificado implícito ($\tau A$) e parâmetros de assinatura que possibilitem o UA calcular sua chave privada (xA) e sua chave pública (XA).

Na fase de autenticação o artigo apresenta duas possibilidades. A primeira opção denominada de esquema "A", é um arranjo no qual o usuário acessa o IED utilizando o SSC como *gateway*. Este é responsável pela primeira camada de verificação da autenticação e pelas comunicações com os IEDs. O usuário envia ao SSC uma mensagem de solicitação

de autenticação, que é formada por um testemunho (W) e por seu certificado ($\tau A$).

O SSC realiza a checagem do certificado do usuário através da chave pública (XTA) da autoridade certificadora e caso seja um usuário válido, o SSC repassa a mensagem para o IED. Este responde com o desafio "c". O usuário (UA) responde ao desafio incluindo a senha do IED na sua resposta. O dispositivo realiza a verificação da resposta e caso esta seja válida, o usuário obtém o acesso solicitado. A Figura 17 detalha o fluxo de informação dos esquemas de autenticação A e B.

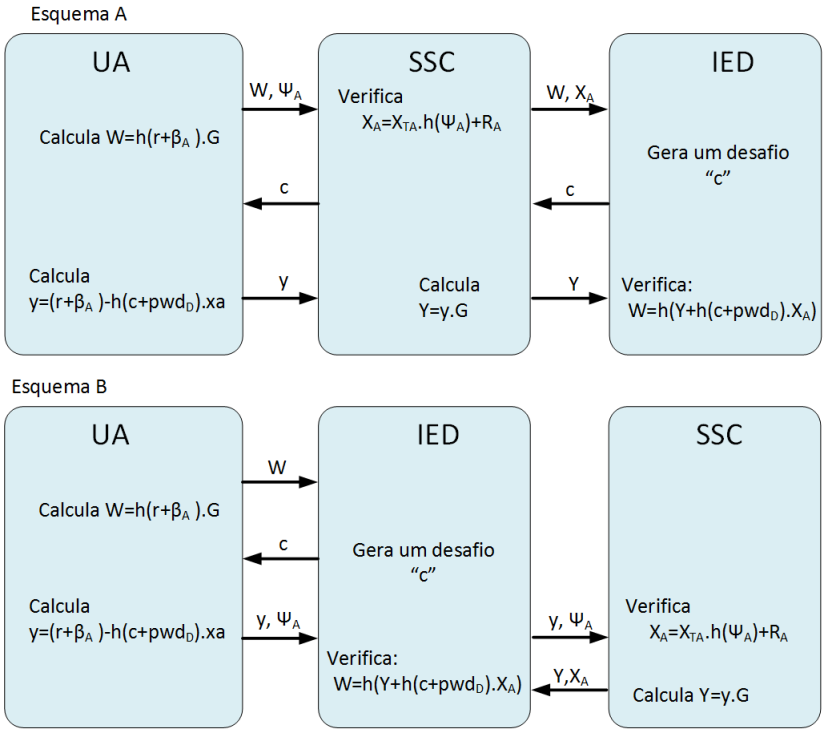


Figura 17 – Fluxo de Mensagens Para os Esquemas de Autenticação A e B (Fonte: (VAIDYA; MAKRAKIS; MOUFTAH, 2013)).

No esquema de autenticação "B", o usuário acessa diretamente o IED enviando o testemunho. Após receber o testemunho, o IED lança um desafio para o usuário, que o responde e envia em conjunto com o certificado $\sigma_a$. O IED remete o certificado para o SSC. Este faz a verificação e, caso a autenticação seja positiva, retorna o resultado para o IED. O esquema de autenticação B é usado, caso o UA necessite acessar diretamente um IED através de uma rede IP ou um modem *dial-up*.

No processo de autorização, o artigo utiliza o conceito de certificados de atributos (ACs), que é similar ao certificado digital (PKCs), mas diferentemente do PKCs, armazena características do usuário ao invés de uma chave pública. Dependendo da necessidade, uma infraestrutura de certificados de atributos pode ter diferentes modelos de autorização em um ambiente de infraestrutura de gerenciamento de privilégio (PMI). O artigo propõe um tipo de delegação que é composto de quatro componentes: Fonte de Autoridade (SOA), Autoridade de Atributo (AA), Titular de Privilégio (PH) e Verificador de Privilégio (PV).

No processo de autorização, o AA é a entidade que assina o *Attribute Certificate (AC)* enquanto o SOA é a raiz de confiança do PMI, que é capaz de delegar o seu poder aos AAs.

Após a autenticação do usuário e do dispositivo por um dos esquemas A ou B, o SSC fornece um certificado que descreve suas permissões de forma individual. Este envia uma mensagem assinada digitalmente para o IED contendo o seu certificado implícito e o de atributo. O IED faz a validação com a ajuda do SSC, em seguida, checa a autenticidade das mensagens e do AC usando as chaves públicas do usuário e do SSC. Com base nas informações colhidas no AC, o IED responde ao usuário. Os ACs são projetados para terem um tempo de vida curto e terem atributos específicos de cada usuário, o que facilita e flexibiliza a escalabilidade de uma PMI. A Figura 18 apresenta o modelo de delegação da PMI.

Os autores realizaram uma análise abstrata na qual foi definido o custo computacional para cada operação do processo de autenticação. Neste caso, $t_H$ é o tempo para execução de uma função *hash* criptográfica, $t_M$ corresponde ao tempo de uma multiplicação modular, $t_A$ é o tempo para execução da aritmética modular $t_{ECM}$, e $t_{ECA}$ é o período para multiplicação e adição de ponto ECM, respectivamente.

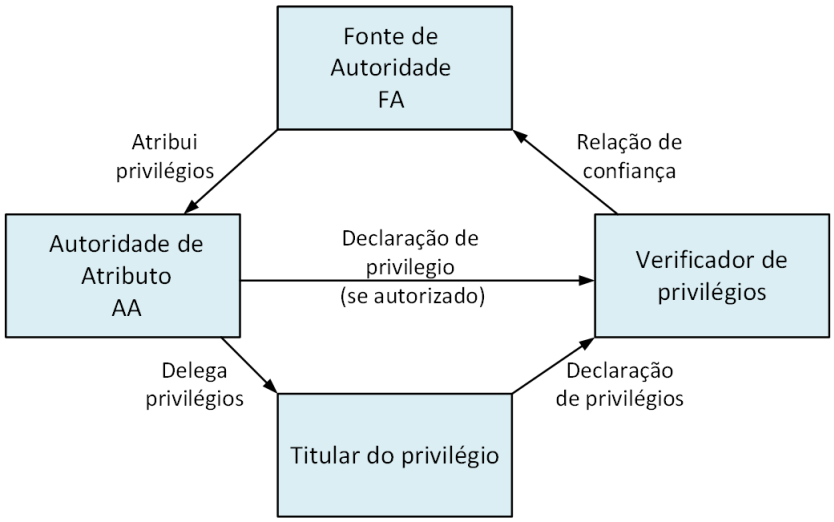


Figura 18 – Modelo de Delegação para a PMI (Fonte: (VAIDYA; MAKRAKIS; MOUFTAH, 2013)).

O custo computacional total na fase de autenticação é $6t_H+4t_A+1t_M+4t_{ECM}+2t_{ECA}$, que os autores consideram de pequena significância. O artigo conclui que o esquema proposto é eficiente e robusto e oferece uma melhor segurança que os esquemas existentes nos sistemas de automação de subestação. Apesar de esses esquemas apresentarem robustez, a sua aceitação necessita de modificações dos projetos dos IEDs existentes, ocasionando custos adicionais.

### 3.4.2 Estudos de Shailendra Fuloria e Ross Anderson

Os estudos de (FULORIA; ANDERSON, 2011) analisam as principais ameaças a uma rede de automação, apresentam uma proposta de uma arquitetura de segurança baseada em controle de acesso, sugere opções de política de segurança e propõem um protocolos para o gerenciamento de chaves criptográficas.

No estudo, os autores identificam que os sistemas de automação das subestações do SEP estão submetidos à vulnerabilidade, que pode ser exploradas por intruso, pessoal interno insatisfeito com a organização, cadeia de suprimentos e pela rede de comunicação.

Os autores afirmam que, atualmente, a maioria dos IEDs não suporta autenticação no processo de comunicação, e que um atacante com acesso a eles pode ler as informações dos sensores e realizar operações indevidas nos atuadores. É possível que este panorama continue por muitos anos devido à grande quantidade e diversidade dos IEDs atualmente em operação.

O artigo propõe a definição de um perímetro lógico, seguro, nas áreas críticas das subestações. As seguintes premissas devem ser observadas para definição desse perímetro ou domínio das redes de automação de subestação:

- A comunicação da subestação deve ser concentrada através de um único canal de comunicação e que, através dele, seja realizada a autenticação para um ou mais dispositivos de controle de borda.

- A autenticação e autorização devem ser realizadas ao nível da rede de subestação, tanto para os dispositivos individuais como para os computadores do centro de controle e servidores do sistema SCADA.

- O acesso externo a um IED, individualmente, pode tornar as tarefas de manutenção e diagnóstico convenientes para equipes de manutenção, mas a acumulação deste tipo de acesso ao longo do tempo destrói a possibilidade de uma arquitetura protegida.

- Se existe uma necessidade de conectar um equipamento ou dispositivo que faz parte da lógica da subestação, mas não faz parte da rede física dela, esse dispositivo deve ser inserido na rede lógica, através de um canal de comunicação seguro.

O artigo considera que a gestão das chaves criptográficas, ao longo do ciclo de vida da subestação, é uma das questões mais difíceis em um contexto de segurança de redes de automação. Esta tarefa pode ser realizada usando o conceito de criptografias simétricas (chave secreta) ou assimétricas (chave pública). As subestações, normalmente, têm uma topologia de rede em estrela com um número limitado de dispositivos, portanto, é provável que sistemas de gerenciamento de chave simétrica sejam economicamente mais viáveis. Nessa perspectiva, cada IED compartilha uma chave com o controlador da subestação ou o servidor SCADA. Nesse contexto, uma forma de distribuir essas chaves é instalar

uma chave de partida, como, por exemplo, uma chave AES de 128 bits no IED, durante o processo de fabricação. Esta deverá ser impressa na embalagem do IED, de forma que o instalador tenha conhecimento desta quando da instalação do IED.

Para instalar este novo dispositivo na rede da subestação, o responsável pela instalação, após realizar a conexão física do dispositivo, informa a chave de partida para o controlador. Este procedimento inicia um protocolo de associação (pedido de conexão) que estabelece a comunicação entre o controlador da subestação e o IED.

Inicialmente, o IED Y enviará um pedido de associação ao controlador da subestação C, encriptado com a chave de partida $k_m$, Equação 3.6. Esta mensagem é composta pela identificação do IED Y um número aleatório de desafio N, de forma que o IED possa verificar se a resposta do controlador da subestação não é um *replay*. Este irá decriptar a solicitação e, em seguida, retornar uma mensagem que contém o desafio aleatório, composto dos seguintes parâmetros: número de série do dispositivo, uma única chave de dispositivo KY e a chave de seção $K_n$, criptografados com a chave de partida $K_m$, Equação 3.7 . O IED irá confirmar o recebimento, retornando o desafio aleatório N criptografado sob a chave de sessão KN, Equação 3.8. Em notação formal, o protocolo de associação pode ser escrito pelas equações:

$$Y \rightarrow C : \{Y, N\}K_m \tag{3.6}$$

$$C \rightarrow Y : \{N, Y, KY, Kn\}k_m \tag{3.7}$$

$$Y \rightarrow C : \{N\}_{KN} \tag{3.8}$$

Uma vez estabelecida a associação, o controlador irá gravar uma entrada para Y em seu banco de dados, contendo as chaves Km e KY. O IED irá, então, usar a chave de seção KN para autenticar a comunicação com o controlador, assim como as mensagens de multicast (GOOSE e SMV) dos outros IEDs pertencentes à rede de automação da instalação. Um esquema de renovação de chave periódico é considerado durante o tempo de vida da subestação. Neste processo, a chave KY é utilizada.

No caso de criptografia assimétrica, a chave de partida $K_m$ impressa na embalagem do dispositivo é o *hash* criptográfico de um certificado digital X.509, emitido pela empresa utilitária de energia ou pelo fabricante do dispositivo. O protocolo de chave pública padrão TLS é usado para estabelecer um canal seguro para comunicação entre o IED recém-instalado e o controlador da subestação. Uma vez que esta sessão tenha sido estabelecida, o controlador envia a chave de sessão KN para o IED.

Nos casos em que os certificados de clientes são instalados nos IEDs, no início da operação, cada dispositivo deve ser equipado com uma chave de gerenciamento de facilidade. O dispositivo Y é provisionado com a sua chave pública KY e a correspondente

chave privada $KY^{-1}$, um certificado $(cert)_u(KY)$ assinado com a chave da empresa utilitária de energia e uma chave de partida $K_m$. Esta é o *hash* criptográfico do certificado $hash(cert_u(KY))$ que é impresso na embalagem do dispositivo. O certificado raiz do utilitário também é instalado no equipamento, de modo que ele possa ser usado para verificar outros, assinados pela chave pública do utilitário.
Ao adicionar um novo IED na rede da subestação, utiliza-se um protocolo de associação para realizar a troca de certificados, caso seja necessário.

O estudo considera que as empresas utilitárias de energia elétrica devem manter a segurança da subestação com o objetivo de controlar o direito básico de acesso. Portanto, deve ser definido um modelo de ameaça, possuir política de segurança bem documentada e os mecanismos adequados para se fazer cumprir essa política. A análise realizada pelos autores mostra que diminuir o perímetro eletrônico da rede da subestação ainda continua sendo o maior impacto sobre a garantia da segurança. Políticas de correções regulares e devidamente atualizadas devem ser priorizadas nas subestações. No caso de rádio e dispositivos remotos que fazem parte da rede da subestação, as concessionárias devem utilizar exclusivamente criptografia nos enlaces de comunicações.

### 3.4.3 Estudos de Iman Ben Abdelkrim, Amine Baina e Mostafa Bellafkih

Os estudos de (ABDELKRIM; BAINA; BELLAFKIH, 2016) consideram que os serviços que constituem a infraestrutura crítica de uma nação, formam uma cadeia interdependente de informações. Para o perfeito funcionamento dessa infraestrutura, existe a necessidade da troca de informações de maneira segura, entre aquelas que se relacionam de forma colaborativa para compartilhar aplicações e recursos.

Infraestrutura crítica, do acrônimo em inglês, *Critical Infrastructures* (CIs) é um termo utilizado para distinguir serviços de natureza específica, de modo que, se forem comprometidos ou destruídos, possam potencialmente causar pertubações maciças em sistemas ou serviços dependentes. Comprometer tais serviços pode ter como consequências elevadas perdas financeiras, ameaças às instalações e à segurança pública. As CIs são controladas por um rede de informações e comunicações denominada de infraestrutura de informações crítica, do acrônimo em inglês, *Critical Information Infrastructure* (CII). Essa infraestrutura usa sistemas vulneráveis e, localizados em um meio hostil, sendo potenciais alvos de ataques de elementos mal-intencionados.

O estudo sugere a aplicação de controle de acesso utilizando o conceito de coalizão dinâmica. Para isso, apresenta o modelo de controle de acesso baseado em coalizão, do acrônimo em inglês, *Coalition-Based Access Control* (CBAC). Nesse modelo, uma coalizão é criada para acessar recursos compartilhados entre sistemas autônomos. As organizações acordam políticas comuns ao nível da coalizão, e essa políticas são traduzidas ao nível de implementação. Esse modelo é composto de três camadas: coalizão, papel e objeto-usuário. O protocolo de tradução de políticas usa os segmentos: coalizão, papéis e objeto-usuário.

Um segundo modelo destacado pelo estudo foi o controle de acesso baseado em papéis distribuídos, do acrônimo em inglês, *distributed Role Based Access Control* (dRBAC). A proposta desse modelo é resolver o controle de acesso em ambientes de coalizão, especificando um mecanismo de controle de acesso, utilizando múltiplos domínios administrativos. Nesse modelo, a autorização para acessar recursos seguros é concedida, verificando se o sujeito solicitante recebeu um papel que o recurso seguro requer para acesso. A arquitetura do modelo é formada pelos blocos:

1. Credenciais: referem-se à relação entre o sujeito e o papel a ele concedido;

2. Cadeias de credenciais: uma credencial pode ser repassada para outro usuário. Neste caso, um usuário U que tem um papel R garantido, pode delegar este papel a outro usuário;

3. Entidades: o dRBAC não faz distinção entre os recursos protegidos e usuários que desejam acessa-los, todos os atores do modelo são denominados de entidades;

4. Papel: todo usuário que deseja acessar um recurso protegido tem que provar que possui o papel necessário para fazê-lo;

5. Delegação: papéis são garantidos, quando submetidos a um processo de delegação.

O autores abordaram um terceiro modelo que realiza negociações de acesso para coalizões dinâmicas. Esse modelo foi desenvolvido para definir uma negociação automática de um estado comum de acesso. Neste caso, existe um acordo comum de compartilhamento de recurso entre os domínios. O acordo é administrado de forma conjunta ou de forma individual pelos domínios participantes. O modelo especifica uma transição de estado que negocia um estado de acesso comum. Esta negociação é realizada por meio de uma linguagem de negociação (NL) que expressa as restrições de negociação de acesso ao domínio.

A política de negociação dos recursos da transição do estado do modelo especifica como proceder a negociação, para permitir aos domínios da coalizão compartilhar os recursos, satisfazendo as restrições da negociação. Isso garante que não haja violação de acesso de qualquer membro do domínio.

As negociações são caracterizadas em três classes:

1. Sem restrições: todos os objetivos dos domínios são iguais e são de conhecimento de todos os domínios;

2. Restrições globais: alguns dos objetivos de alguns domínios podem não coincidir com os de outros domínios, mas todos os domínios têm conhecimento dos objetivos uns dos outros;

3. Restrições locais: neste caso, alguns dos objetivos de alguns domínios podem não coincidir com os objetivos dos outros domínios;

Os autores apresentam um estudo de caso para o modelo, em um cenário de formação de coalizão dinâmica aplicado a um sistema elétrico de potência (SEP). Um SEP é formado por três funções primárias: geração, transmissão e distribuição. Cada função pode ser associada a um domínio de informação, que pode ser gerenciado pela mesma organização ou uma organização diferente.

A infraestrutura de informação é composta por um centro de controle da transmissão que recebe informações das subestação da transmissão, um centro de controle da distribuição que recebe informações das subestações da distribuição, e repassa as informações de interesse da transmissão, para o seu centro de controle.

No caso da necessidade de realizar um corte de carga automático para restabelecer as condições operativas do SEP, o acesso às informações deve seguir as negociações entre os domínios ou organizações ao nível de coalização. Nessa situação, o controle de acesso pode seguir ser qualquer uma das três metodologias apresentadas no artigo.

### 3.4.4 Estudos de Domencio Rotoni e Salvatore Piccione

O paradigma de internet das coisas, do acrônimo em inglês, *Internet of Things* (IoT), está cada vez mais próximo de se tornar uma realidade, que revolucionará o comportamento da humanidade. Este ambiente apresenta novos desafios de segurança que exigem novas soluções ou uma revisão substancial das existentes. Os estudos de (ROTONDI; PICCIONE, 2012) fornecem uma descrição do sistema de controle de acesso baseado em capacidade, do acrônimo em inglês, *Capability Based Access Control (CapBAC)*, como solução de segurança para o ambiente de internet das coisas.

Para resolver as questões de segurança neste cenário, enfrenta-se um elevado grau de desafios, que abrange, desde a heterogeneidade dos sistemas conectados, às tecnologias de comunicações utilizadas e às restrições de recursos de processamento, armazenamento e comunicações dos dispositivos. Além disso, este ambiente tem potencial para um número ilimitado de elementos interagindo entre si, com padrões de interação diversificados, tais como: dispositivos, aplicativos e humanos.

Neste ambiente, a utilização de controle de acesso baseado em papéis ou baseado em atributos, apresentam problemas de usabilidade relevantes. Os papéis não podem ser definidos de forma consistente, cada usuário é quase autônomo e o seu significado e utilidade é difícil de ser capturado. Além disso, os dispositivos desse contexto não têm suporte computacional suficiente para implantar os sistemas de RBAC e ABAC que necessitam de recursos para disponibilizar funcionalidades, tais como: repositórios de regras de acesso, pontos de decisão de políticas, mecanismos de autenticação e de gerenciamento de identidade.

Em um sistema de controle de acesso tradicional, o provedor de serviços verifica se o usuário tem, de forma direta ou indireta (por exemplo, através de uma função de propriedade do usuário), o direito de realizar as operações solicitadas em um pedido de recurso. Por outro lado, em um sistema baseado em capacidade, o usuário deve fornecer sua capacidade de autorização, e demonstrar que possui as exigências necessárias solicitadas pelo provedor de serviços.

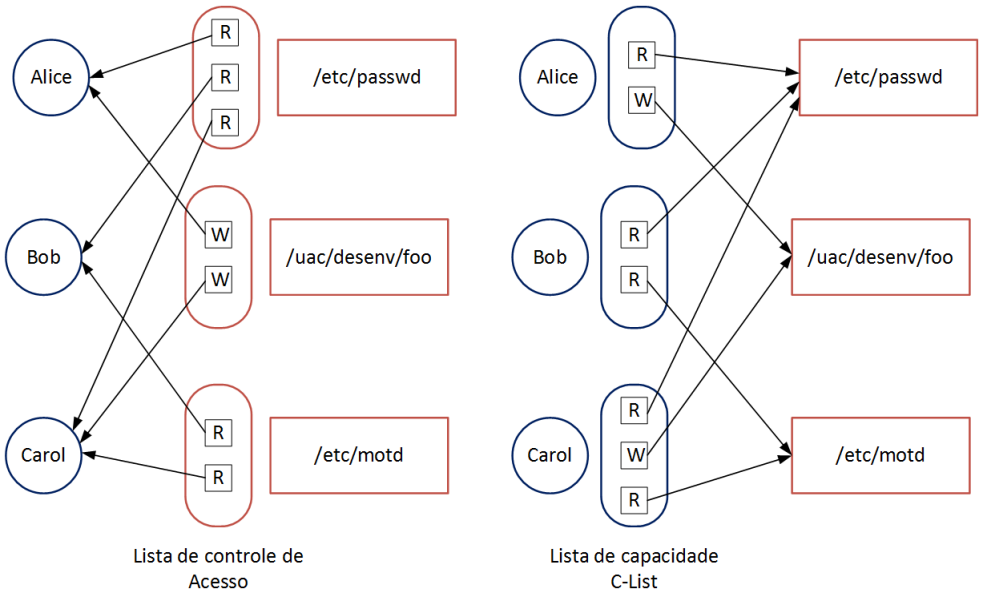


Figura 19 – Modelo de Controle de Acesso Baseado em Lista de Acesso Versus Modelo Baseado em Listas de Capacidades (Fonte: (ROTONDI; PICCIONE, 2012)).

A Figura 19 apresenta as diferenças do modelo de um sistema tradicional de controle de acesso baseado em lista de acesso e um sistema de controle de acesso baseado em lista de capacidade. No modelo baseado em lista de acesso, o sujeito não tem conhecimento dos recursos que tem direito e nem dos limites destes recursos. Nesta situação, torna a delegação dos direitos ao recurso uma tarefa extremamente árdua. No caso do modelo baseado em capacidade o sujeito tem um perfeito conhecimento dos recursos, e quais são os seus direitos sobre eles. Neste caso, tem pleno conhecimento dos limites dos recursos, o que torna a tarefa de delegação fácil de ser implementada.

## 3.5 Considerações Finais do Capítulo

Nesta seção, apresentamos as considerações finais do capítulo, apresentando as ponderações analisadas nesta Tese sobre os três tipos de abordagem estudadas na revisão bibliográfica.

### 3.5.1 Autenticação de Mensagens

A metodologia de autenticação de mensagens apresenta uma excelente abordagem teórica de segurança. Na prática, existe como contraponto o fato de que as soluções apresentadas como satisfatórias são baseadas em esquemas que propõem modificações de arquitetura de hardware dos IEDs (FPGA, ASIC e *crypto chips*). Isso significa que a aplicação de segurança estudada tem como consequência a implementação de novos projetos de IEDs. O emprego dessas soluções apresenta todo o ciclo de desenvolvimento de um produto.

Por outro lado, as empresas utilitárias de energia têm atualmente um legado de IEDs cujos mecanismos de segurança não estão atendendo ao requisitos exigidos no cenário atual. Esses mecanismos são baseados na identificação do usuário e na senha de acesso. Eles são facilmente quebrados através de um ataque de dicionário. A inclusão de novos IEDs como estratégia de segurança baseada na metodologia de autenticação de mensagem implica na substituição do parque existente.

No setor elétrico, o tempo de vida dos sistemas de automação na tecnologia atual é da ordem de 10 a 15 anos. Os critérios de substituição são definidos por obsolescência, por desempenho ou por mudanças na regulamentação do setor. A adoção dessa estratégia implica em submeter as redes de automação a um período de vulnerabilidade nesta ordem de grandeza.

O impacto da substituição dos IEDs não é somente no custo de aquisição e instalação. Nesse procedimento, existe a necessidade de parar os serviços que estes dispositivos oferecem. Isto pode ocasionar o surgimento de fragilidades no controle e proteção do sistema elétrico de potência. Além disso, nessa etapa são necessários testes de comissionamento, que poderão ocasionar obrigatoriedade de desligamento do sistema elétrico, proporcionando um custo elevado de implantação e possíveis paralisações do fornecimento de energia elétrica.

### 3.5.2 Análise de Tráfego

A abordagem de segurança cibernética focada na análise de tráfego apresenta problemas considerados relevantes para a sua adoção como solução única e definitiva em redes de automação. Sua eficácia depende dos métodos de detecção de intrusão considerados. No caso da técnica baseada em redes apresentadas nos estudos, evidenciam-se os seguintes pontos negativos: perda de pacotes em redes saturadas, dificuldade de entendimento de protocolos de aplicação específica para as redes de automação, ausência de capacidade de monitorar cabeçalhos de protocolos que estão embutidos em *payloads* cifrados e dificuldades de sua aplicação em redes segmentadas. Portanto, essa técnica pode funcionar como um esquema complementar para outras abordagens que estão sendo discutidas nesta pesquisa. Sua utilização poderá ser concebida como uma ferramenta adicional dentro de um

projeto de segurança cibernética.

### 3.5.3 Controle de Acesso

A metodologia, utilizando o mecanismos de segurança de controle de acesso, apesar de não oferecer controles que garantam a confidencialidade plena, demonstra que o resultado final para o contexto de redes de automação é tão eficiente quanto o de outras metodologias, tendo em vista que o conceito de confidencialidade não é tão exigido como em uma rede de gestão corporativa. O mecanismo de controle de acesso assegura que a obtenção de um determinado recurso seja realizado apenas por um sujeito ou por objeto autorizado. Para o caso do fundamento de confidencialidade, somente o usuário com direitos de acesso a um recurso pode obter ou visualizar a informação deste. Para o fundamento de integridade, somente o usuário autorizado poderá escrever ou modificar a informação no recurso.

Os mecanismos de controle apresentados nos estudos analisados nesta seção propõem que as funções do serviço de autenticação sejam realizadas pelo controlador da subestação, acrescidas de novas funcionalidades do protocolo de segurança aos IEDs, de forma a capacitá-los para execução do esquema de segurança propostos.

A utilização do controlador da subestação como servidor de autenticação pode proporcionar os seguintes problemas: comprometimento das funções de controle e aquisição de dados pela perda de desempenho motivado pela execução das funções de segurança; declínio do controlador de subestação decorrente de um ataque cibernético ocasionado pela invasão do serviço de autenticação; necessidade de substituição dos IEDs devido à participação destes na cadeia de autenticação e autorização.

O modelo de controle de acesso adotado nesta Tese foi motivado por apresentar, as seguintes vantagens em relação às outras metodologias apresentadas neste capítulo; (1) não utilizar componentes da rede de automação como fornecedor de serviços para o modelo; (2) não depender de desenvolvimento ou de atualizações de projetos de IEDs para atender aos mecanismos de segurança exigidos; (3) não necessitar de substituição dos IEDs existentes; (4) apresentar um tempo de implantação reduzido. O atendimento destas premissas permite obter uma solução de segurança, na qual a implantação é mais rápida e econômica, preservando, dessa forma, todo o investimento realizado no sistema de automação. A Tabela 10 exibe os principais resultados obtidos pelos estudos de segurança das redes de automação apresentados neste capítulo.

Tabela 10 – Resumo dos Mecanismos de Segurança e Desempenho (Fonte: Autor)

| Mecanismos de segurança | Algoritmos e funções criptográficas | Serviço de segurança | Máximo Desempenho | Substituição de IED |
|---|---|---|---|---|
| Código de autenticação de mensagens (MAC) | AES256 SEED | Integridade autenticidade | MD5 = 2,82 us | Sim |
| | MD5 HMAC MD5 | | HMAC-MD5 = 4,68us | |
| Assinatura digital | SHA256 RSA | Integridade Autenticidade | PC de 2,2 GHz= 3,9 ms | Sim |
| | | | FPGA de 200 Mz = 2,0 ms | |
| | | | ASIC = 0,16 ms | |
| | | | Crypto chip = 23,8 us | |
| Assinatura digital | DES, RSA | Integridade Autenticidade | Rede estrela 100 Mbps = 0,245 ms | Sim |
| | | | Rede Anel 100 Mbps = 0,589 ms | |
| Autenticação Multifator | | Integridade | 6tH+4tA+1tM+ 4tECM+2tECA | Sim |
| | | Autenticidade | | |
| Detecção de intrusão (IDS) | IDS Baseado em redes | | Intervalo entre pacotes de 1ms | Não |
| | | | Relação de falso Negativo = 7,72 x10-4 | % |
| | | | Relação de falso Positivo = 4,66x10-4 | % |

# 4

# MODELO CONCEITUAL

Este capítulo apresenta o modelo conceitual de ilha de segurança aplicado ao contexto de redes industriais, especificado e projetado para minimizar as brechas de segurança existentes nos sistemas legados, sem a necessidade de substituir os dispositivos e, limitando o impacto de implantação na cultura atual da indústria. Nas seções seguintes são detalhados os principais módulos e conceitos que compõem esse modelo. Finalizando o capítulo é apresentada a especificação do modelo em linguagem de descrição e configuração (SDL) juntamente com a descrição de todos os blocos funcionais que estruturam essa especificação.

## 4.1 Introdução

Até recentemente, uma rede industrial era caracterizada pela presença de equipamentos distintos que se comunicavam por *intranets* isoladas através de protocolos proprietários e específicos para cada solução. Esse ambiente restrito, indiretamente, favorecia a segurança dos dados e informações por obscuridade, já que um possível invasor teria que compreender diferentes arquiteturas de sistemas e vários protocolos de conhecimento restrito aos fabricantes. No caso de funcionários demitidos, o mecanismo de segurança adotado era a restrição do acesso físico às instalações da rede de automação.

Com o crescimento da Internet, evolução das tecnologias de acesso às redes e a convergência digital, muitas redes industriais migraram gradativamente da infraestrutura fechada para soluções abertas, que favorecem a integração e padronização dos sistemas e dispositivos. Apesar de oferecer maior flexibilidade, essas soluções expõem maiores riscos à segurança dos sistemas industriais.

A interligação das redes industriais com as redes de dados corporativos, por exemplo, aumenta a fragilidade do sistema como um todo, uma vez que uma falha de segurança na rede corporativa pode afetar a rede de automação industrial. Esse fato merece atenção especial, uma vez que a grande maioria dos dispositivos de automação e os protocolos existentes não foram projetados para atender aos requisitos de segurança. Por fim, a

inexistência ou a precariedade das equipes de segurança na área de produção, bem como maus hábitos de utilização contribuem para aumentar os problemas de segurança desses sistemas.

A alternativa mais segura, portanto, compreenderia uma reestruturação completa dos sistemas de automação e rotinas de segurança. Por outro lado, os custos elevados relacionados à troca de equipamentos e mudança abrupta de comportamento dos usuários muitas vezes é inviável, principalmente por representar um grande impacto nas atividades da organização.

Em outras palavras, além das limitações financeiras, alterar o funcionamento de um sistema de automação industrial legado para um contexto totalmente protegido (de acordo com as recomendações das normas atuais), pode gerar grande insatisfação nos envolvidos por causa da necessidade de mudança de comportamento brusco. Além de poder comprometer a atuação da empresa, cujos setores não estariam habituados com as novas práticas estabelecidas. É imperativo que, para evitar esses contratempos, seja mantido o nível de satisfação dos usuários e o desempenho do sistema industrial no mesmo patamar da etapa anterior à implantação do sistema de segurança.

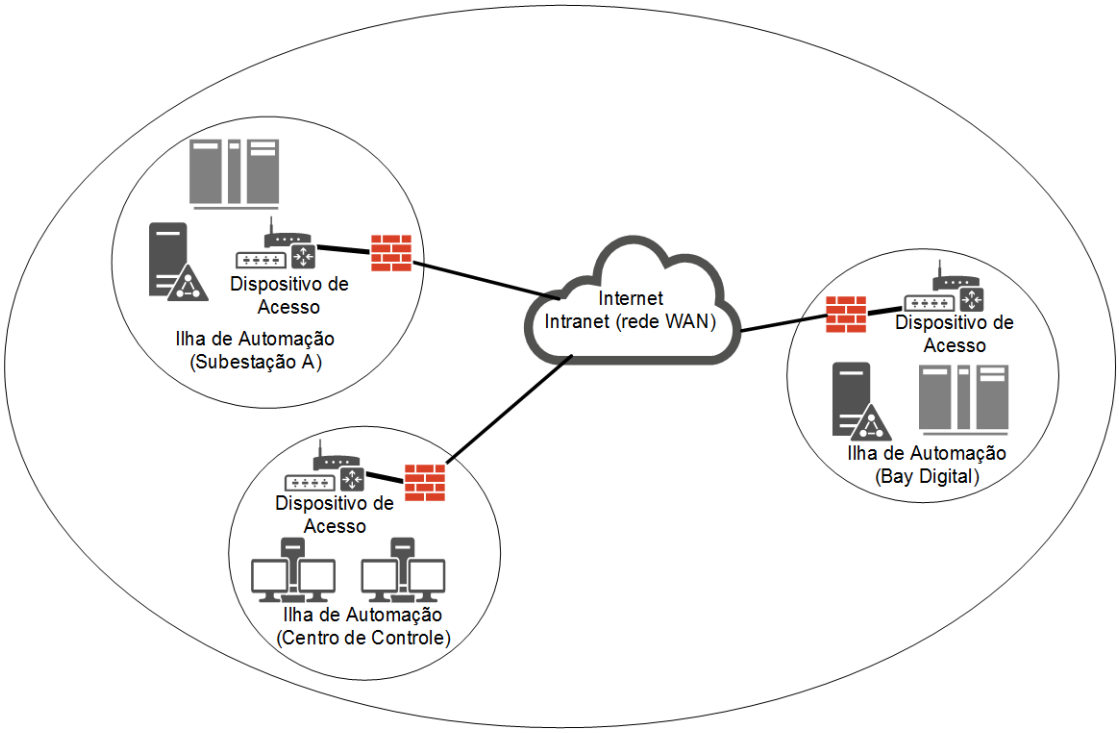


Figura 20 – Ilhas de Automação em um Ambiente do Setor de Energia Elétrica (Fonte: Autor).

Nesse contexto, é possível argumentar que organizações que trabalham com sistemas críticos podem se beneficiar de uma solução que aumente a segurança dos sistemas sem quase impactar suas atividades. A Figura 20 apresenta um exemplo de sistema crítico (setor de energia elétrica) onde existe uma integração, não só entre diferentes centros de

operação e subestações, agrupados em ilhas de Automação, como com a rede corporativa (WAN) e a Internet. O modelo de ilha de segurança proposto nesta Tese utiliza a estruturação de sistemas críticos em ilhas de Automação e aplica estratégias de controle de acesso para prover a segurança desses sistemas.

## 4.2 Arquitetura do Modelo Conceitual de Segurança

Como visto no Capítulo 3, diferentes tecnologias e métodos podem ser utilizadas na garantia de segurança dos sistemas de automação. Além disso, essas redes possuem requisitos específicos de segurança, como autenticidade e integridade das mensagens, enquanto a confidencialidade geralmente não é requisito fundamental.

Para o caso específico de sistemas elétricos de potência, em adição aos requisitos gerais, algumas operações possuem um requisito de tempo bastante restrito. Por exemplo, a leitura e atuação por parte dos IEDs devem acontecer em menos de 4ms. Essa restrição adicional impõe desafio extra sobre a escolha de um modelo de segurança apropriado, dado que tradicionais funções de criptografia de mensagens podem acrescentar atraso adicional não aceitável ao sistema.

Dessa forma, a arquitetura foi especificada tendo por base o controle de admissão e direitos de acesso baseados em políticas. O modelo busca atender aos requisitos de autenticidade e integridade da informação e, em uma extensão menor, a confidencialidade e a disponibilidade (ver Seção 2.4.3), ao mesmo tempo que produz mínimo impacto na infraestrutura das redes de automação existentes.

A abordagem de controle de acesso insere diretrizes de segurança no ambiente das redes de automação dos sistemas elétricos, sem afetar de maneira abrupta o processo da operação.

O conceito de ilha de segurança apresenta a estratégia para garantir a segurança de um perímetro lógico, com um acesso ao ambiente externo através de uma única porta. O ambiente externo mesmo localizado internamente à organização é considerado hostil. No interior da ilha, os serviços de segurança que executam a função de controle de acesso, compartilham toda infraestrutura com os serviços relacionados às funções de automação (ver Figura 21).

A arquitetura proposta é formada por hardwares e softwares que realizam as funções de autenticação, autorização e auditoria inerentes a um ambiente de controle de acesso. Ela é composta dos seguintes blocos funcionais: (a) Serviço de Acesso à Rede; (b) Serviço de Gerenciamento de Políticas; (c) Serviço de Gerenciamento de Usuário; e (d) Cliente do Serviço de Segurança. A Figura 21 mostra a composição desses serviços internamente à ilha de segurança e os detalhes das interações entre eles.

O **Serviço de Acesso à Rede** é responsável pela autenticação do usuário, pela execução do PEP de rede após a fase de autorização, realizada pelo serviço de gerenciamento

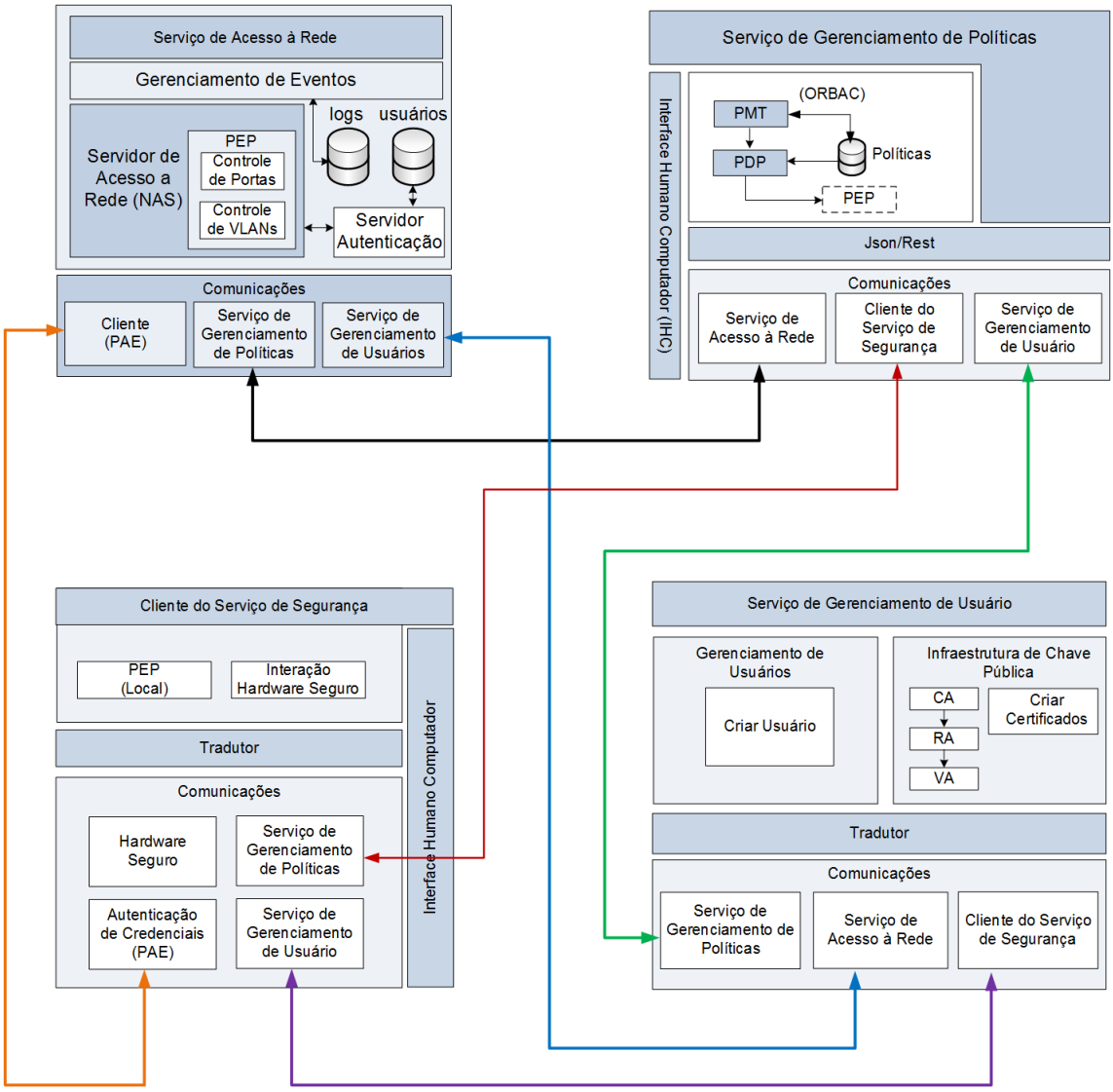


Figura 21 – Modelo Conceitual da Ilha de Segurança (Fonte: Autor).

de política, e pelo gerenciamento dos registros de eventos da ilha de automação. O **Serviço de Gerenciamento de Usuário** tem a função de gerenciar a criação e remoção de usuários e armazenar as informações dos usuários necessárias ao processo de autenticação destes. Além disso, este serviço realiza todas as funções necessárias ao gerenciamento de certificados conforme o modelo de infraestrutura de chave pública X.509.

O **Serviço de Gerenciamento de Políticas** executas as funções de PDP e PMT, gerenciando as políticas do modelo.

Por fim, o **Cliente do Serviço de Segurança** é responsável pelo processo *Port Access Entity (PAE)*. Em outras palavras, o processo executa a função de suplicante no contexto de autenticação. Outra função executada por este serviço é o PEP local.

As próximas seções descrevem em maior detalhe a funcionalidade dos principais blocos funcionais que compõe a ilha de Segurança.

## 4.3 Componentes do Modelo da Ilha de Segurança

### 4.3.1 Usuários da Ilha de Segurança

Os usuários, no contexto do modelo, são categorizados como fornecedores ou consumidores de serviço. Outra classificação de usuário é com relação à sua localização na ilha de segurança, este pode estar localizado internamente ou externamente à mesma. Normalmente, os usuários externos são consumidores de serviço. Cada usuário receberá um tratamento específico, dependendo da sua categoria.

Para os consumidores de serviço, as credenciais são repassadas pelo cliente Serviço de Gerenciamento e Decisão de Políticas (SGDP) para o autenticador, através de um evento gerado por um dispositivo confiável, no qual estão armazenados os dados pessoais e as chaves de segurança dos usuários. No caso de fornecedores de serviço, o processo de obter as credenciais é realizado automaticamente através de uma consulta do cliente SGDP a uma base de credenciais segura.

Com relação à localização do usuário, o modelo apresenta diferenças no processo de autenticação. Para os usuários internos, cujo acesso é realizado diretamente no dispositivo autenticador, o processo de autenticação permite a liberação ou não da porta na qual eles estão conectados. O acesso externo é realizado através do *switch* de borda da rede corporativa que atende a edificação onde está localizada a ilha de automação. Este *switch* está conectado fisicamente a uma porta de entrada do dispositivo de autenticação. Portanto, se esta for liberada para um usuário autenticado, ela permanecerá liberada para os demais usuários, independentemente do resultado da autenticação.

O modelo da ilha de segurança utiliza uma VPN que será usada como porta de entrada para os usuários externos. Esta VPN será responsável pela avaliação dos usuários, empregando as mesmas credenciais que eles usam para um acesso interno à ilha de segurança. Para acesso externo ao domínio da empresa, o usuário se autenticará inicialmente na VPN da rede corporativa e, em seguida, se autenticará na VPN da ilha de segurança.

### 4.3.2 Serviços Protegidos

Serviços protegidos representam qualquer aplicação ou sistema que não faz parte do esquema de segurança da ilha. Serviços protegidos são intrinsecamente ligados ao processo produtivo da corporação. Como exemplo destes, existem os sistemas SCADA, utilizados em centros de controle de sistema, nos controladores de subestações e nos sistemas de automação localizados no nível de processo que estão situados no pátio e nas cabanas de relés de uma subestação ou usina. É importante ressaltar que os serviços protegidos são autenticados na rede de automação quando inicializados.

Por envolver, possivelmente, o uso de serviços críticos e não divulgados, o modelo de segurança trata qualquer serviço que possa ser integrado à sua estrutura de forma

transparente. Sabe-se que aquele servidor possui um papel bem definido em uma dada fatia da rede, mas a forma como este serviço funciona permanece invisível aos demais componentes da arquitetura da ilha. A única dependência exigida destes servidores é possuir as bibliotecas de comunicação e autenticação necessárias aos serviços de segurança da ilha.

### 4.3.3 Serviço de Acesso à Rede

No contexto de segurança da informação, o controle de acesso é dividido em três fases denominadas autenticação, autorização e auditoria, do acrônimo em inglês *Authentication, Authorization, Accounting (AAA)*. É importante salientar que, para existir algum tipo de controle de acesso, é necessário que a gestão de recursos, sistemas e usuários estejam presentes. Neste ambiente, a autenticação é responsável pela identificação do usuário, a autorização é responsável por atribuir os direitos de acesso aos recursos solicitados por eles e a auditoria realiza o registro das atividades dos usuários na rede. A Figura 22 detalha o modelo funcional do serviço de acesso à rede.

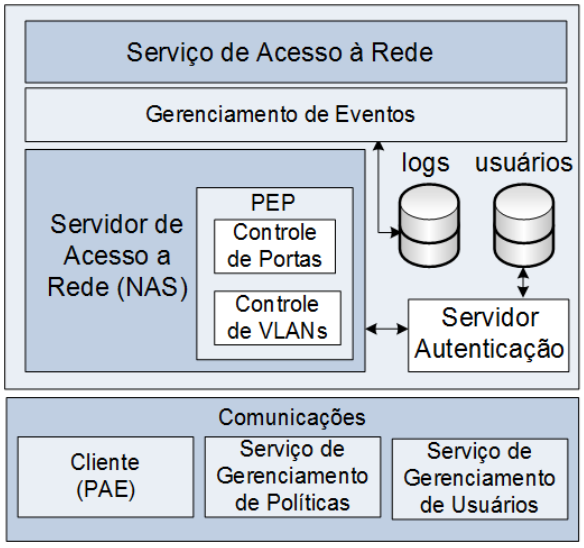


Figura 22 – Modelo do Serviço de Acesso à Rede (Fonte: Autor).

A autenticação corresponde à primeira fase do processo, sendo encarregada de identificar os usuários credenciados ou não credenciados. Os credenciados são aqueles reconhecidos pela rotina de checagem de identidade e os não credenciados são os que não foram reconhecidos ou os que não conseguem comprovar a sua identidade. Portanto, uma rotina de autenticação deve responder de maneira segura, ou computacionalmente segura, se o usuário que deseja acessar um serviço do ambiente da rede de automação é credenciado ou não.

A autorização é responsável pela segunda fase do processo. Nesta etapa, será realizada a comunicação com o serviço de gerenciamento de políticas para obter deste os resultados

da função de decisão de políticas. Uma vez recebidos, através da função PEP, será aplicada a política no servidor de acesso à rede, do acrônimo em inglês, *Network Access Server (NAS)*.

No último estágio do processo de controle de acesso, é executada a auditoria, através do registro dos eventos das atividades do usuário e dos serviços que irão permitir a realização de análise forense ou métricas estatísticas. Estes eventos deverão possuir uma estampa de tempo e são armazenados em repositórios específicos.

Além dessas funções, o serviço de acesso à rede executa, através de um hardware exclusivo, o papel de servidor de acesso à rede, sendo realizadas as funções de PEP para as etapas de autenticação e autorização. Na primeira, a função de controle de porta, permite, ou não, o acesso do usuário a uma determinada porta, dependendo do estado de credenciamento do mesmo. A segunda, realiza o controle PEP atribuindo VLAN aos usuários autorizados.

Outra atribuição desse serviço é a realização da comunicação necessária ao processo de autenticação do cliente dos serviços de segurança (suplicante). Esta comunicação é realizada por meio da função entidade de acesso à porta, e de um protocolo específico de autenticação, que pode ser aberto ou proprietário.

### 4.3.4 Serviço de Gerenciamento de Políticas

O modelo do serviço de gerenciamento de políticas é composto por um repositório de políticas, por módulos que são responsáveis pela execução das funções de PMT, PDP e de comunicações, além de uma interface de interação humano computador. O primeiro executa as funções necessárias ao gerenciamento como criação, edição e eliminação de políticas.

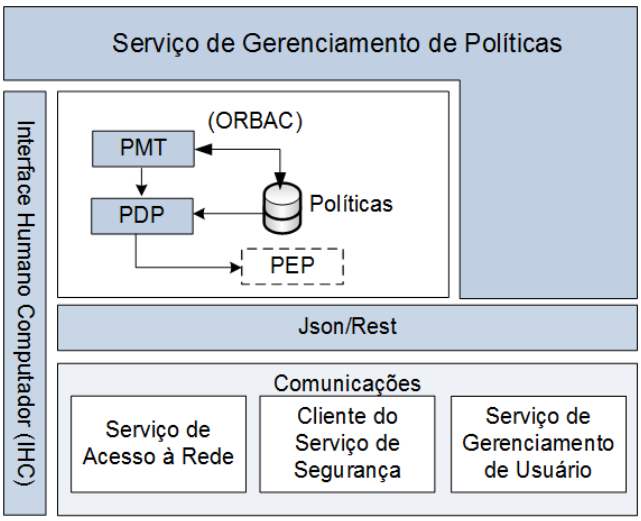


Figura 23 – Modelo do Serviço de Gerenciamento de Políticas (Fonte: Autor).

As interações necessárias com a administração de segurança são realizadas através de uma interface de interação humano computador. A função PDP executa as decisões de acordo com as regras definidas para cada usuário. Após esse estágio, é realizada a comunicação com o serviço de acesso à rede para o envio dos comandos necessários a fim de realizar as funções de PEP. A Figura 23 apresenta o diagrama do serviço de gerenciamento de políticas.

#### *4.3.4.1 Interação dos Serviço de Autenticação e Autorização com as Bases de Dados de Usuários e Políticas*

Iniciando o processo de autenticação, o usuário digita o seu número de identificação pessoal, do acrônimo em inglês, *Personal Identification Number (PIN)*. Considerando que este esteja correto, o processo de autenticação continua em dois passos na estação de trabalho do usuário.

No primeiro momento, conforme descrito anteriormente, as credenciais do usuário são utilizadas para verificação de qual sub-rede poderá ser acessada. Para tal, o servidor *RADIUS* consulta no banco de dados qual a configuração que deve ser realizada no dispositivo de acesso para o usuário.

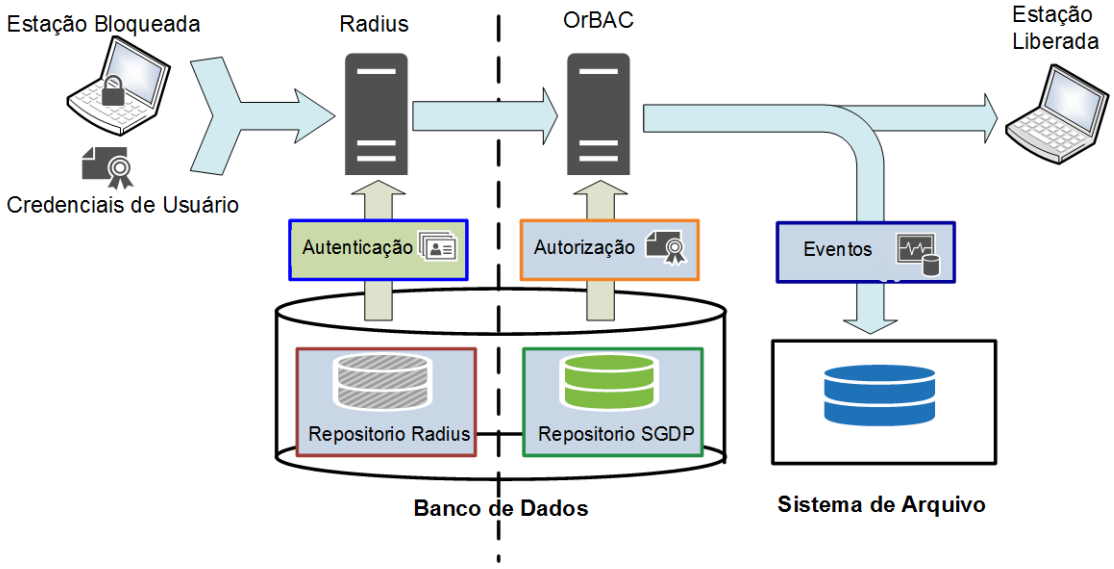


Figura 24 – Interação dos Módulos de Autenticação e Autorização com o Banco de Dados (Fonte: Autor).

No passo seguinte, o SGDP verifica as permissões do usuário, identificando quais são os recursos locais ou de rede que o usuário no processo de autorização tem permissão de uso. O módulo de verificação de políticas, então, utiliza a identificação do usuário para recuperar as permissões e os impedimentos associados ao seu papel. Tanto as regras quanto as credenciais a serem consultadas estão armazenadas em banco de dados e as alterações

de configuração nas regras de comportamento realizadas na ferramenta de gerenciamento de políticas são refletidas nas tabelas do banco de dados.

A interação é executada com a base de dados de políticas. Nesta, são armazenadas as regras das políticas e os recursos de serviços, de redes e de sistemas operacionais. Baseado nestas informações, o SGDP definirá as permissões dos usuários na fase de autorização. Após a etapa de gravação, o usuário que possui as credenciais no sistema pode fazer uso dos recursos da rede. A Figura 24 detalha o esquema do módulo de políticas OrBAC, coletando as regras que definem o grau de autorização do usuário e o registro do evento de autenticação (ou falha).

### 4.3.5 Serviço de Gerenciamento de Usuários

Este serviço é responsável pelas ações de gerenciamento de usuários e de certificados, pelas interações com o hardware seguro durante o processo de autenticação e criação de certificados. Na fase de cadastramento do usuário, o módulo de gerência de usuários é responsável pela criação, edição e eliminação dos usuários e dos atributos a eles relacionados. Essas informações são armazenadas na base de dados de usuários e serão utilizadas no processo de credenciamento destes na rede. A Figura 25 apresenta o modelo do serviço de gerenciamento do usuário.

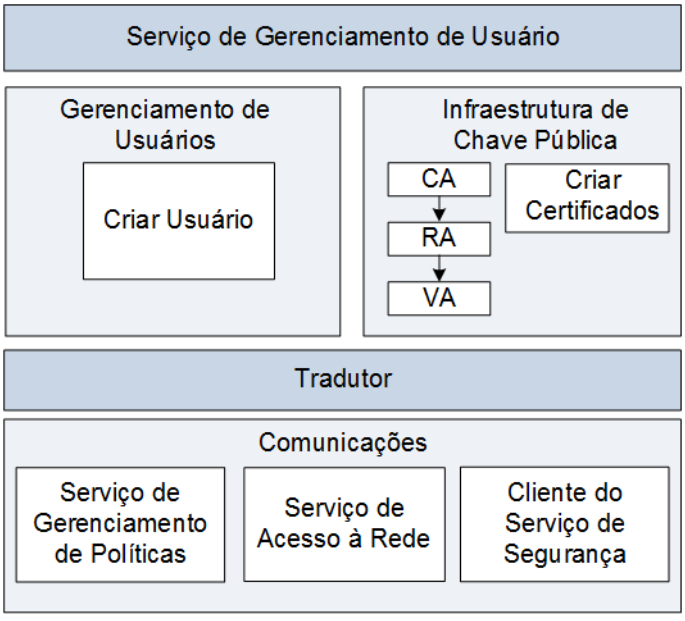


Figura 25 – Modelo do Serviço de Gerenciamento de Usuários (Fonte: Autor).

Na fase de cadastramento, o módulo de gerenciamento de certificado gera um certificado para o usuário, e as respectivas chaves públicas e privadas. O certificado é assinado digitalmente pela autoridade certificadora raiz e armazenado no hardware seguro que identifica o usuário. Para isso, o serviço de gerenciamento de usuários utiliza uma Infraestrutura de Chave Pública (ICP) contendo uma autoridade autocertificada. Caso a

empresa tenha uma ICP implantada ou contratada, esta poderá ser utilizada na solução de controle de acesso das redes de automação.

O serviço de gerenciamento de usuários realiza comunicação do serviço de acesso à rede para fins de auditoria, com o cliente de serviço de segurança durante a fase de armazenamento dos certificados no hardware seguro e com o serviço de gerenciamento de política para enviar parâmetros para a base de políticas.

#### *4.3.5.1 Fluxo de interação das Bases de Dados da Ilha de Segurança*

Para demonstrar a interação do serviço de gerente de usuários com os bancos de dados, um caso de uso de criação de usuário é descrito a seguir. Vale ressaltar que, apesar dos diversos módulos e tecnologias estarem envolvidos no processo, apenas o conteúdo relacionado a bancos de dados e persistência será destacado.

A sequência inicial de interações de um usuário com o sistema é através da criação deste e sua posterior autenticação na estação de trabalho. Os dados do usuário são inseridos através da interface do serviço de gerenciamento de usuários, de forma que possibilite a criação e assinatura do certificado pelo servidor da ICP e, posteriormente, realize a gravação no dispositivo seguro (*smartcard*).

Após a conclusão desse processo, os dados pessoais do novo usuário e os dados do seu certificado são dispostos no banco de dados de usuários, sendo armazenados os dados gerais do usuário como: nome, email, telefone e o seu certificado. Essas informações serão utilizadas pelos serviços de acesso à rede e de gerenciamento de políticas, posteriormente nas fases de autenticação e autorização. O registro das ações executadas é armazenado em forma de arquivo de texto. A Figura 26 apresenta o detalhamento das ações durante o cadastro de um usuário ilustrando as interações dos módulos com as bases de dados do modelo da ilha de segurança.

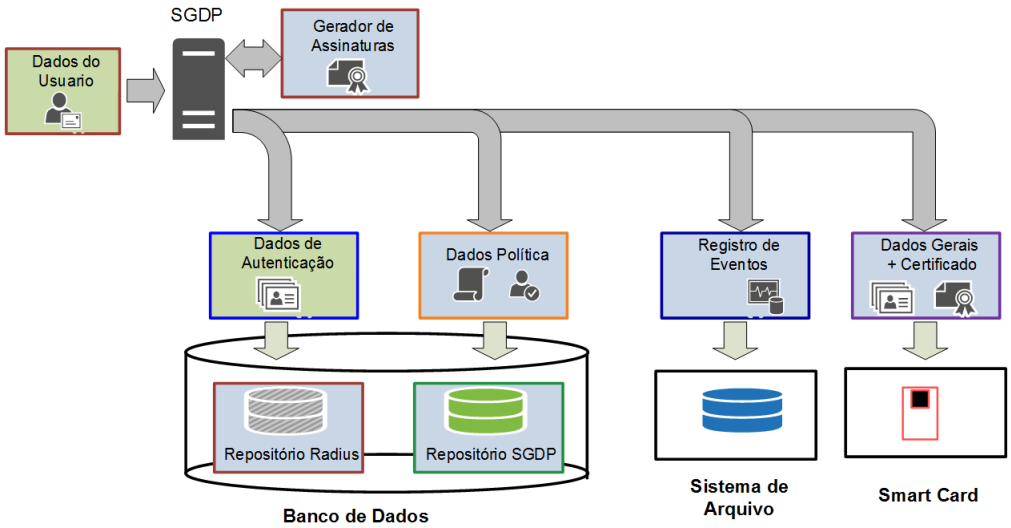


Figura 26 – Interação dos Módulos de Criação do Usuário com o BD (Fonte: Autor).

#### *4.3.5.2 Infraestrutura de Chave Pública do Modelo da Ilha de Segurança*

A infraestrutura de chave pública ICP, também conhecida pela terminologia em inglês *Public Key Infrastructure* (PKI), é um sistema para a criação, armazenamento e distribuição de certificados digitais usado para verificar a identidade de uma determinada entidade. A ICP cria certificados digitais que mapeiam chaves públicas para entidades de forma segura. Outra funcionalidade da ICP é armazenar esses certificados em um repositório central e revogá-los, caso seja necessário (CHANDRA; MESSIER; VIEGA, 2002; VACCA, 2004).

Uma Infraestrutura de Chaves Públicas é composta por: uma Autoridade Certificadora Raiz (AC - Raiz), uma Autoridade Certificadora (AC), uma Autoridade de Registro (AR) e uma Autoridade Certificadora de Tempo (ACT).

A primeira autoridade é a entidade mais importante da cadeia de certificação, sendo responsável por executar as Políticas de Certificados aprovadas pela organização que a define. Compete à AC-Raiz emitir, expedir, distribuir e revogar os certificados das autoridades certificadoras de nível imediatamente inferior ao seu. Esta entidade também está encarregada de emitir a lista de certificados revogados, além de fiscalizar e auditar as ACs e ARs abaixo de sua cadeia.

A Autoridade Certificadora é subordinada à hierarquia da AC-Raiz e é responsável por emitir, distribuir, renovar, revogar e gerenciar certificados digitais. Esta entidade é responsável por verificar se o titular do certificado possui a chave privada que corresponde à chave pública que faz parte do certificado. Também cabe à AC emitir a lista de certificados revogados e manter registros de suas operações, sempre obedecendo às práticas definidas pela AC-Raiz.

As Autoridades Registradoras são responsáveis pela comunicação entre o usuário e a Autoridade Certificadora. Vinculada a uma AC, a AR valida e encaminha solicitações de emissão ou revogação de certificados digitais e identificação de forma presencial de seus solicitantes.

Uma Autoridade Certificadora de Tempo tem como atribuição legal a geração, conservação e disseminação da hora legal vigente e da localidade em questão. Esta tecnologia permite levar aos documentos digitais a hora legal de forma segura, autêntica e auditável, trazendo não só a prova temporal, como também a garantia do conteúdo.

### 4.3.6 Cliente do Serviço de Segurança

O serviço de cliente de autenticação tem como objetivo principal realizar a função de autenticação de credenciais em conjunto com o serviço de acesso à rede, com o propósito de obter as credenciais necessárias para o acesso à rede. Além disso, realiza as interações com o serviço de políticas para conseguir as autorizações sobre os recursos dos serviços de automação disponíveis na rede.

Outra atribuição desse serviço é realizar a comunicação com os hardwares seguros durante as fases de autenticação e de criação de certificados para realizar o armazenamento desse no dispositivo seguro.

Ademais, esse serviço realiza a função de PEP para as decisões tomadas pelo serviço de gerenciamento de políticas ao nível local. A Figura 27 apresenta a estrutura do cliente do serviço de segurança.

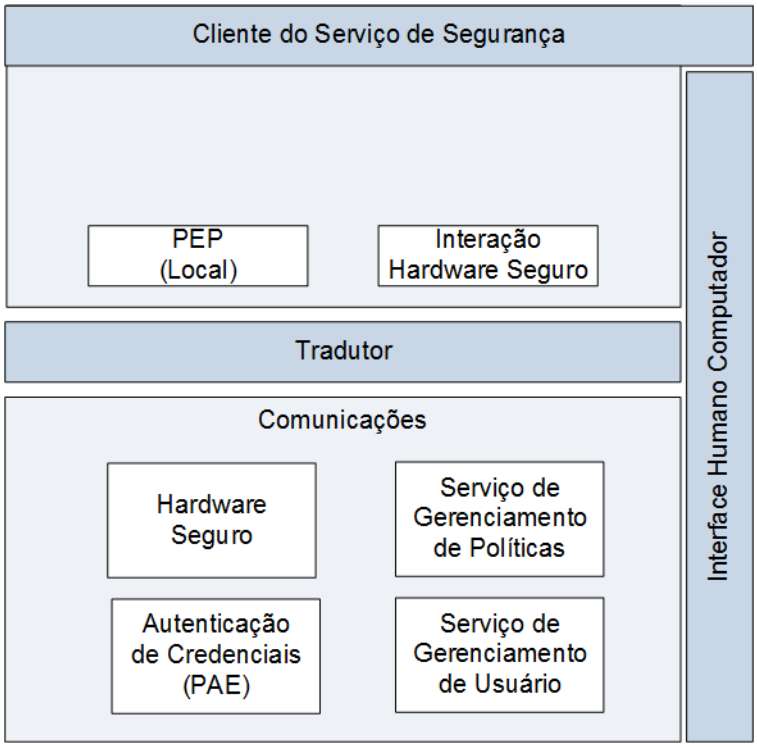


Figura 27 – Modelo do Cliente do Serviço de Segurança (Fonte: Autor).

### 4.3.7 Autenticação Usando Dispositivos Seguros (hardwares)

A autenticação individual revelou ser um item essencial para prover maior proteção na comunicação entre usuários, serviços e recursos. Autenticação, conceitualmente falando, é o ato de estabelecer ou confirmar algo, ou alguém, como autêntico. Quando realizada de forma adequada, a autenticação consegue comprovar o elemento legítimo, seja usuário ou serviço, dentro da rede.

Os mecanismos de autenticação de usuários podem ser identificados em três categorias (também chamadas de fatores), classificados de acordo com o modo de obtenção da credencial utilizada: baseados no conhecimento (o que se sabe), baseados em propriedade (o que se possui) e baseados em características (o que se é) (YOUNG; KIRSTEIN; IBBETSON, 1996).

Os *smart cards* possuem diversas aplicações e foram empregados em diversos serviços nas últimas duas décadas (RANKL; EFFING, 2010). A primeira delas e também a principal

é a utilização de tal tecnologia para identificação perante os serviços que exijam algum nível de autenticação. Por fim, uma das principais atribuições de tais cartões é proteger os dados neles contidos, de forma a evitar que um usuário não autorizado possa acessá-lo.

### 4.3.8 Persistência da Informação e Suporte para Segurança da Ilha de Segurança

À medida que sistemas ganham complexidade e afetam vários ramos de uma organização, as informações que eles manipulam precisam ser mais elaboradas para serem perpetuadas de forma consistente e eficiente por todos os módulos. Os sistemas de gerenciamento de bancos de dados, por causa de sua natureza robusta e geralmente distribuída, são a melhor abordagem de persistência para sistemas extensos e que exijam acesso concorrente.

Dentro dos Sistema de Gerenciamento de Banco de Dados (SGBD), diferentes técnicas priorizam os requisitos de persistência tais como: a robustez, o acesso paralelo, o acesso distribuído, a proteção dos dados e o desempenho de consulta. No contexto desta Tese, as explorações foram baseadas nas abordagens que estão em conformidade com os requisitos do sistema de segurança proposto. Além destes, considera-se como requisito não-funcional a utilização de banco de dados aberto, uma vez que isso facilita sua adaptação às necessidades do modelo de segurança e traz como efeito colateral positivo o suporte da comunidade de desenvolvimento.

Dos módulos propostos pelo modelo de segurança da ilha de automação, a persistência da informação precisa ser observada na interação com o sistema de autenticação, com o sistema de políticas e com o sistema de registro de eventos.

O SGBD deve ser capaz de separar as visões das bases de dados, para evitar que ocorra acesso indevido às informações, de forma que a parte crítica dos dados não seja comprometida pela parte mais vulnerável. Tendo em vista a natureza de segurança proposta no modelo da ilha e sabendo-se que informações como senhas de acesso e chaves serão dados recorrentes no sistema, é necessário que se tenha um banco de dados encriptado, para que as entidades com acesso físico ao servidor de banco de dados não corrompam ou acessem os dados armazenados sem possuir autorização para tal.

O SGBD deve oferecer portabilidade, isto é, possuir a facilidade na qual este possa ser conectado aos diversos servidores envolvidos no processo. Alguns bancos de dados têm um paradigma distribuído, o que pode aumentar o desempenho do banco em consultas e em escalabilidade, mas abrem mão da portabilidade ao satisfazer estes requisitos. Para o modelo de segurança proposto, no escopo desta Tese, faz-se necessário que o SGBD possua portabilidade, tenha capacidade de acesso protegido, ser contido num arcabouço de robustez e permitir sincronização com outros bancos, tendo em vista o seu uso em sistemas redundantes.

#### *4.3.8.1 Repositório de Certificado em Hardwares Dedicados*

Para garantir a legitimidade dos serviços que são fornecidos e, consequentemente, aumentar a confiabilidade da ilha de segurança, é preciso usar uma abordagem de autenticação desses serviços, visto que os *smart cards*, utilizando leitores manuais, necessitam da intervenção do usuário no processo de autenticação. No caso de servidores, é utilizado um hardware que armazena fisicamente os cartões de autenticação destes, eliminando esta intervenção. Portanto, da mesma forma que existe uma raiz de confiança para os usuários, é necessário que exista dentro da ICP uma raiz específica para os serviços, de forma que autentique e registre os certificados.

Cada serviço possui seu certificado, que é utilizado durante o processo de autenticação. Para tal processo acontecer, é preciso armazenar os certificados de todos os serviços em seus respectivos *smart cards* para que, assim que eles forem inicializados, efetuem a autenticação e, posteriormente, estejam acessíveis para os usuários. Uma forma de organizar os *smart cards* é realizar um agrupamento de dispositivos para armazenamento de *Subscriber Identity Module (SIM)* ou *Cryptographical Cards*. Este arranjo é denominado grid de *smart cards* ou servidor de *smart cards*. Tais arranjos são capazes de lidar com uma grande quantidade de ICCs simultaneamente e de forma paralela, assim como um grande número de solicitações.

O repositório do certificado é capaz de gerenciar e armazenar os ICCs de forma automatizada. Esse gerenciamento ocorre de maneira análoga aos leitores de *smart cards*, controlando o acesso, gerenciando a transferência e recepção de dados. Toda essa comunicação é possível graças a um protocolo de comunicação específico com os *smart cards*. O protocolo oferece suporte a recursos de gerenciamento como: alocação, ativação, desligamento e checagem de estado para os cartões que estão no *grid*. Diferentemente dos leitores de *smart cards* convencionais, o repositório de certificado não está conectado à aplicação que o utiliza por meios convencionais como uma porta USB, mas por meio de uma conexão TCP em rede. Tal característica permite que o dispositivo possa ser usado como servidor de *smart cards* para diversas máquinas. Como principais características desses dispositivos, têm-se:

- Acesso paralelo e concorrente a *smart cards* presentes no *grid*;
- Possibilidade de checagem de estado de operação, *lock* e *unlock* de *smart cards*, além da possibilidade de realizar operações de *reset*;
- Entrada e saída de dados de acordo com as especificações da ISO 7816.

Com a finalidade de armazenar as senhas privadas e certificado de cada serviço que compõe a ilha de segurança, uma matriz de *SimCard* é incorporada à rede de automação. As informações necessárias à autenticação de cada serviço são armazenadas em um *Sim-*

*Card* específico da matriz. A Figura 28 detalha o processo de autenticação destes serviços e a forma como o repositório *SimCard* interage com os demais componentes da ilha.

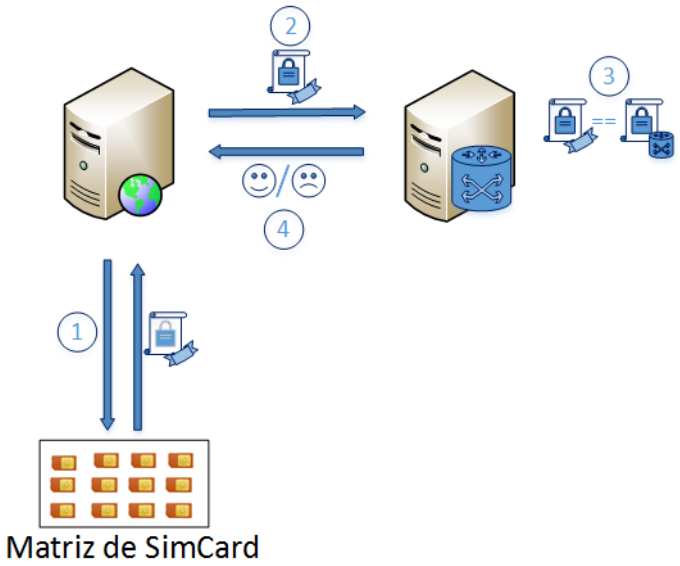


Figura 28 – Processo de Autenticação de Serviços Usando Matriz de SimCard.

a) No passo 1, ao iniciar a máquina onde se encontra, o serviço faz uma requisição do seu certificado através do suplicante, utilizando a interface de comunicação com o repositório de certificado.

b) De posse do certificado, no passo 2, o serviço dá início ao processo de autenticação, utilizando o protocolo de autenticação, e o certificado é enviado para averiguação.

c) No passo 3, o servidor verifica se o certificado está assinado pela autoridade certificadora para cada serviço. Isso garante que o certificado é legítimo e válido.

d) Em seguida, no passo 4, o servidor retorna o resultado da autenticação com um *SUCCESS* ou *FAILURE*. Caso a autenticação seja bem-sucedida, o serviço será direcionado para as VLANs estabelecidas pela política de segurança, as quais são disponibilizadas para os clientes desses serviços quando eles se autenticarem.

Dessa forma, o uso da matriz de *SimCard* permite que um serviço se autentique na rede utilizando *SimCards* para persistir suas credenciais.

## 4.4 Especificação do Modelo da Ilha de Segurança

O modelo foi especificado utilizando a linguagem de especificação e descrição SDL. A escolha desta linguagem é devida às suas características de poder descrever formalmente a estrutura, o comportamento, as comunicações e a descrição dos dados de um modelo de

sistemas reativos. Uma especificação SDL é um modelo formal que define as propriedades de um sistema existente ou a ser construído.

### 4.4.1 Sistema Ilha de Segurança

A especificação do modelo da ilha de segurança em linguagem SDL define um sistema composto pelos serviços de acesso à rede, de gerenciamento de usuário, de gerenciamento de políticas e cliente de segurança. Esses serviços serão especificados utilizando-se os blocos. Estes são denominados de agentes. Cada bloco é formado por processos que contêm os procedimentos que representam uma funcionalidade da ilha de segurança. Os blocos se relacionam através de canais de comunicação. A Figura 29 mostra o detalhamento dos blocos e canais do sistema ilha de segurança.

O sistema da ilha de segurança; é formado pelos blocos *AcessoRede*, que especifica o serviço de acesso à rede; o *GerentePoliticas*, que especifica o serviço de gerenciamento de políticas; o *GerenteUsuario*, que especifica o serviço de gerenciamento de usuários e o *ClientSeg*, que especifica o serviço cliente de segurança.

Além desses blocos, foi configurado um módulo de interface gráfica de usuário (GUI) localizado no ambiente externo ao sistema, embora não esteja representado no modelo de ilha de segurança, sua funcionalidade é apresentada em tempo de simulação. Esta GUI emula o leitor de cartões e a interação de entrada e saída do usuário com o sistema. A comunicação interna ao sistema da ilha de segurança é realizada por meio de canais que interligam os diversos serviços de segurança que estão contidos neste ambiente. Esta rede é formada pelos seguintes canais:

1. Acesso à Rede Cliente de Segurança (ARCS): é o canal que liga o serviço de acesso à rede, bloco (*AcessoRede*), ao serviço cliente de segurança, bloco (*ClientSeg*);

2. Acesso à Rede Gerência de Políticas (ARGP): é o canal que liga os serviços de acesso à rede, bloco (*AcessoRede*), ao serviço gerência de política, bloco *(*GerentePoliticas);

3. Acesso à Rede Gerente de Usuários (ARGU): é o canal que liga os serviços de acesso à rede, bloco (*AcessoRede*), ao serviço gerência de usuário, bloco *(*GerenteUsuarios);

4. Gerente Políticas Cliente de Segurança (GPCS): é o canal que liga o serviço de gerência de políticas, bloco *(*GerentePoliticas), ao serviço cliente de segurança, bloco *(*ClienteSeg);

5. Gerente Políticas Gerente Usuário (GPGU): interliga o serviço de políticas, bloco *(*GerentePoliticas), ao serviço de gerência de usuário, bloco *(*GerenteUsuario);

6. Gerente Usuários Cliente de Segurança (GUCS): conecta o serviço de gerência de usuário, bloco *(*GerenteUsuario), ao serviço de acesso à rede, bloco (*AcessoRede*);

7. Hardware Seguro Cliente Segurança (HSCS): conecta o hardware seguro, no ambiente externo, ao serviço de acesso à rede, bloco (*AcessoRede*);

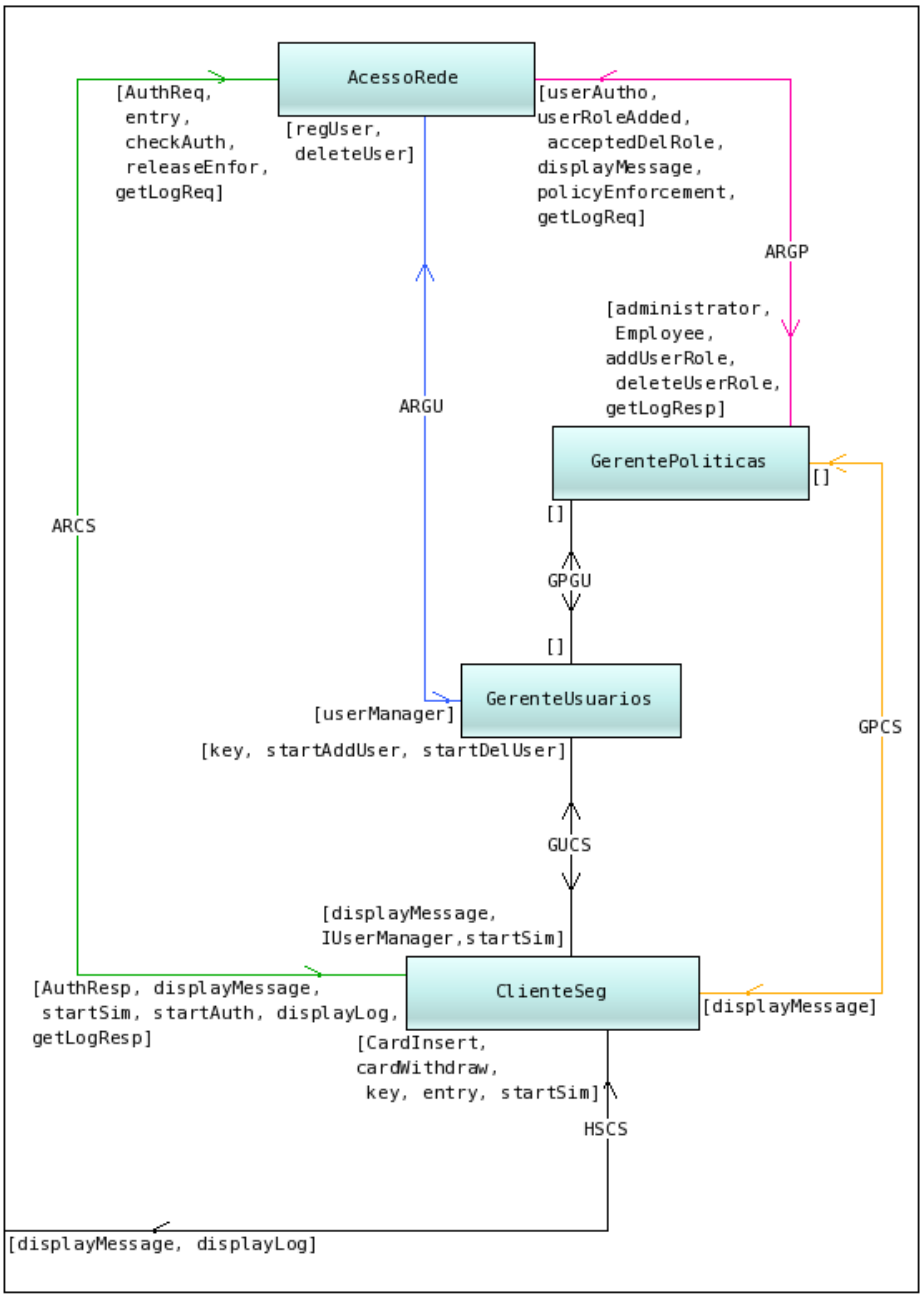


Figura 29 – Especificação do Sistema Ilha de Segurança em SDL (Fonte: Autor).

Um conjunto de mensagens foi definido para fornecer os estímulos necessário para as operações da ilha de segurança. A Tabela 11 mostra a lista de mensagens utilizada no modelo e a ação realizada.

Tabela 11 – Desempenho do Cenário Real do Centro de Controle (Fonte: Autor).

| Mensagem | Ação |
|---|---|
| acceptedDelRole(INTEGER) | Eliminação de papel aceita |
| addUserRole(userRoleInfoType) | Solicita a adição de papel do usuário na base de dados |
| administrator(INTEGER) | Informa que usuário corrente é administrador |
| AuthReq(userInfoType) | Solicita autenticação de um determinado usuário |
| AuthResp | Resposta à solicitação de autenticação |
| CardInsert(INTEGER) | Notifica ao sistema a inserção de um cartão no leitor |
| cardWithdraw | Informa ao sistema a retirada de um cartão |
| checkAuth(INTEGER) | Checa se o usuário já está autenticado e autorizado |
| delEnfor(INTEGER) | Elimina a aplicação de política de um determinado usuário |
| deleteUser(userInfoType) | Solicita a eliminação do usuário da base de dados |
| deleteUserRole(INTEGER) | Solicita a eliminação de papel do usuário da base de papéis |
| displayLog(logInfoType) | Solicita registro de mensagem no log |
| displayMessage(CHARSTRING) | Mostra no display da GUI as mensagens trafegadas |
| Employee(INTEGER) | Informa que usuário corrente é um empregado |
| entry(CHARSTRING) | Envia uma *string* que é digitada no campo *entry* da GUI; |
| getLogReq(CHARSTRING) | Solicita incluir uma mensagem no log |
| getLogResp | Resposta da realização do registro do evento |
| startSim | Informa ao sistema para ir para o estado inicial |
| IUserManager | Entrada no estado de gerenciamento do usuário |
| key(CHARACTER) | Envia o código da tecla digitada no GUI |
| policyEnforcement(UserSectionInfoType) | Ordem de aplicação de políticas |
| regUser(userInfoType) | Solicita o registro do usuário na base de dados de usuários |
| releaseEnfor(INTEGER) | Solicita a liberação da aplicação da política para um determinado usuário |
| sGetUserRoleInfo(INTEGER) | Solicita informações sobre o papel do usuário |
| sGetUserRoleResp(userRoleInfoType) | Resposta à solicitação sobre o papel do usuário |
| startAddUser(INTEGER) | Solicita o registro de um usuário no sistema |
| startAuth(INTEGER) | Solicitação de início de autenticação |
| startDelUser(INTEGER) | Solicita a eliminação do registro do usuário |
| userAutho(userRoleInfoType) | Usuário autorizado |
| userManager | Solicita ao serviço GerenteUsuarios para entrar no estado de gerenciamento de usuário; |
| userRoleAdded | Papel do usuário estabelecido |

O bloco *ClientSeg* é responsável por enviar toda a comunicação da ilha de segurança para o ambiente externo. Esta troca de informações é realizada através do canal HSCS, no qual trafegam as seguintes mensagens: *CardInsert*, *Entry*, *key*, *cardWithdraw*, *startSim*, *displayMessage* e *displayLog*. Além da comunicação com o ambiente externo, o cliente de segurança realiza comunicação com os serviços internos da ilha de segurança. No caso do serviço de acesso à rede, a comunicação é realizada pelo canal ARCS utilizando as seguintes mensagens: *AuthReq, entry, checkAuth, releaseEnfor, getLogReq, AuthResp, displayMessage, startSim, startAuth, displayLog* e *getLogResp*. Outra comunicação desse bloco é realizada com o bloco GerentePoliticas utilizando o canal GPCS no qual trafega a mensagem *displayMenssage*. Também realiza comunicação com o bloco GerenteUsua-

rios, utilizando o canal GUCS, e usa a mensagens *key, startAddUser, startDelUser, displayMessage, startSim e IUserManager*. Os detalhes do bloco ClientSeg são descritos na Seção 4.4.2

O bloco GerenteUsuario é o encarregado pela criação e remoção de usuários e de geração de certificado. Para tanto, o bloco GerenteUsuario necessita realizar comunicação com o bloco AcessoRede através do canal ARGU utilizando as mensagens: *userManager, regUser e deleteUser*. Além deste, realiza comunicação com o bloco ClienteSeg e emprega o seguinte conjunto de mensagens: *regUser, deleteUser e userManager*. Outra comunicação deste bloco é com o bloco GerentePoliticas e utiliza o canal GPGU. Os detalhes do bloco GerenteUsuario são descritos na Seção 4.4.4.

O bloco GerentePoliticas realiza o gerenciamento das políticas e se comunica com o bloco de AcessoRede através do canal ARGP utilizando as mensagens: *userAutho, userRoleAdded, acceptedDelRole, displayMessage, policyEnforcement, getLogReq, Administrador, Employee, addUserRole, deleteUserRole e getLogResp*. Também realiza a comunicação com o bloco de ClienteSeg utilizando o canal GPCS no qual trafega a mensagem *displayMenssage* e com o bloco Gerenteusuarios por meio do canal GPGU. Os detalhes do bloco GerentePoliticas são descritos na Seção 4.4.3.

O bloco AcessoRede realiza as funções de autenticação, autorização e auditoria. Este bloco se comunica com o bloco ClienteSeg, utilizando ARCS, por meio das mensagens: *AuthReq, entry, checkAuth, releaseEnfor, getLogReq, AuthResp, displayMessage, startSim startAuth, displayLog* e *getLogResp*, com o canal ARGU através das mensagens:*userManager, regUser e deleteUser* e com o bloco GerentePoliticas operando no canal ARGP com as mensagens: *userAutho, userRoleAdded, acceptedDelRole, displayMessage, policyEnforcement, getLogReq, Administrador, Employee, addUserRole, deleteUserRole e getLogResp*. Os detalhes do bloco ClientSeg são descritos na Seção 4.4.5.

### 4.4.2 Bloco Cliente de Segurança

O bloco ClientSeg realiza o papel de suplicante e também é responsável pela interação do usuário com a ilha de automação. Este bloco é composto pelos processos entidade de acesso à porta (PAE), processo interface de hardware seguro (InteHwSeg) que resolve a comunicação com o ambiente externo à ilha de segurança e pelo processo de aplicação de políticas local (PEP). O processo InteHwSeg, ao receber a mensagem *CardInsert*, informando o ID do cartão inserido vindo do ambiente externo, gera uma mensagem *checkAuth* para o processo PAE com a finalidade de verificar junto ao bloco AcessoRede se o cartão já está autenticado. Caso positivo, retorna uma *displayMessage* informando que usuário já está autenticado. Nessa situação, uma mensagem displayLog também é gerada. Caso o usuário não esteja logado, o controle é direcionado para o processo PAE com a finalidade de iniciar a tarefa de autenticação do usuário.

Outra tarefa realizada pelo processo InteHwSeq é quando do recebimento da mensagem *cardWithdraw*. Neste contexto é enviada um mensagem *displayMessage* solicitando o ID do cartão retirado e, ao receber a resposta, envia uma mensagem *releaseEnfor* para o bloco de AcessoRede para excluir a sessão do usuário. Por fim, este processo quando receber uma mensagem IUserManager redireciona a interface gráfica do usuário para o bloco GerenteUsuarios. Outra função deste processo é a retransmissão das mensagens *displayMessage* e *entry* para os outros serviços da ilha de segurança.

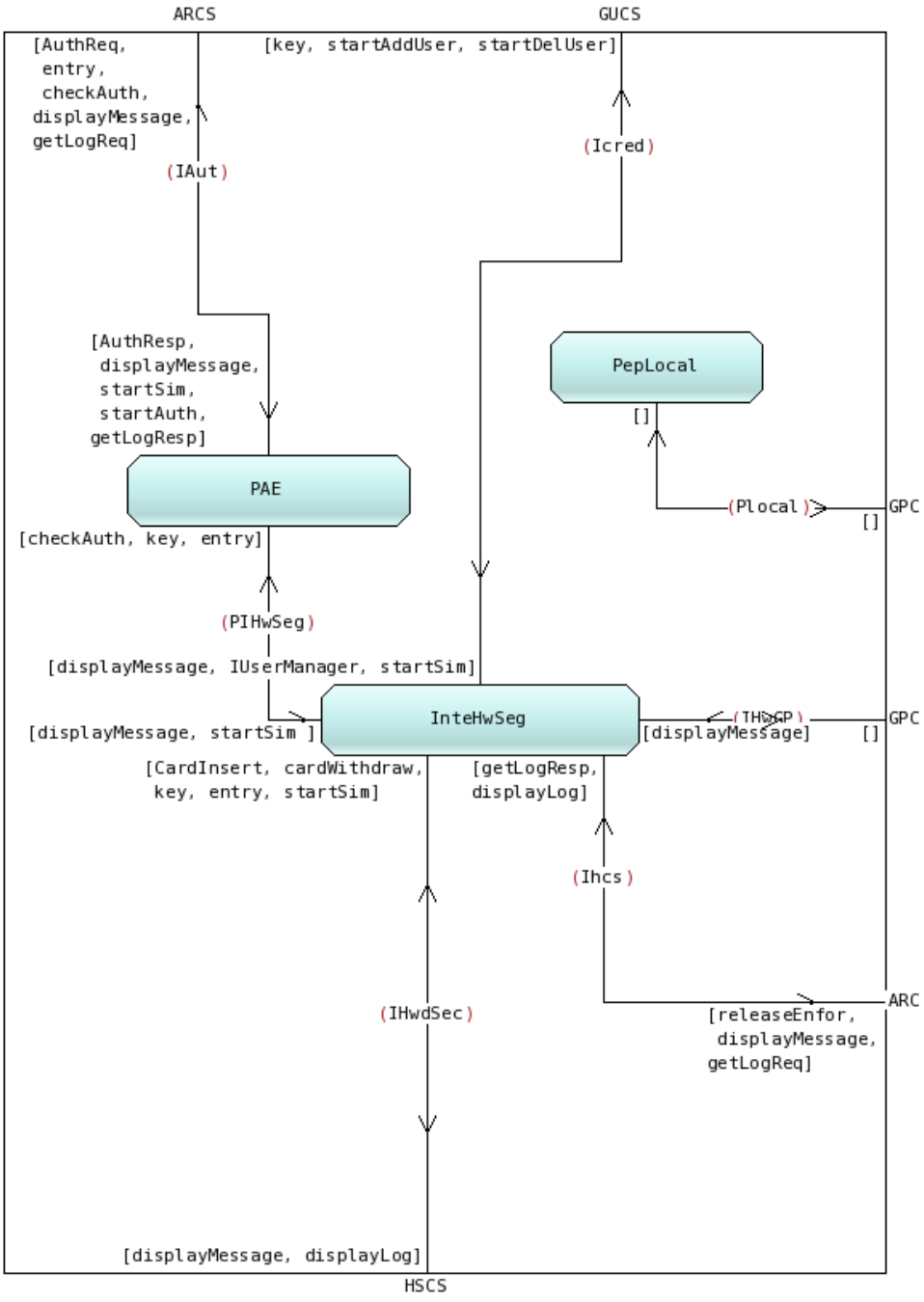


Figura 30 – Detalhamento dos Processos do Bloco Cliente de Segurança (Fonte: Autor).

O processo PAE, após o recebimento da mensagem *startAuth* do bloco AcessoRede confirmado que o usuário não está logado, inicia a tarefa de entrada da identificação do usuário, através de uma mensagem *displayMessage*, solicitando a digitação do PIN. Após a conclusão desta etapa, o PAE emite a mensagem *AuthReq* para o bloco AcessoRede solicitando a autenticação do usuário. Esta mensagem contém como parâmetro a identificação do cartão e número PIN do usuário. O processo *PepLocal* realiza a aplicação de política ao nível local, como exemplo do bloqueio de tela da estação do usuário. A Figura 30 mostra o detalhe do bloco de cliente de segurança e o detalhe dos fluxo de sinais e das mensagens que são trocadas entre os processos que compõem esse bloco.

### 4.4.3 Bloco Gerente de Políticas

O bloco gerente de políticas é formado pelo processo de decisão de políticas PDP e pelo processo de gerenciamento de políticas pPMT. O primeiro é responsável pela tomada das decisões das políticas aplicadas e o segundo é encarregado pela criação, eliminação e armazenamento destas.

O processo PDP estando no estado inicial deste processo e ao receber a mensagem *administrator* que encontra-se em processo de autenticação, emite a mensagem *addUserRole* para o processo PMT, para armazenar o papel dele. Após o recebimento da mensagem *userRoleAdded*, o PDP transmite a mensagem *userAutho* para o módulo de AcessoRede, indicando que o usuário está autorizado no papel de administrador.

No caso de receber a mensagem *Employee*, o PDP emite a mensagem *sGetUserRoleInfo* para o PMT para buscar o papel do usuário. Uma vez encontrado o papel do usuário, o PDP recebe a mensagem *sGetUserRoleResp* e inicia a busca das políticas associadas a este papel. Caso a busca tenha sucesso, o PDP emite a mensagem *policyEnforcement* para o bloco AcessoRede para criar uma sessão do usuário. Caso contrário, emite a mensagem *displayMessage e displayLog* informando o insucesso da operação.

No processo pPMT, ao receber a mensagem *addUserRole*, o pPMT armazena o papel do usuário em uma variável que emula a base de dados de políticas. Os parâmetros desta mensagem é um tipo *array* específico, cuja estrutura é formada pela identificação do cartão e do papel do usuário. Na situação em que a mensagem recebida for *deleteUserRole*, o pPMT realiza uma busca na base de dados de papéis para exclusão do papel solicitado. Caso haja sucesso, o pPMT envia a mensagem *acceptedDelRole* para o bloco AcessoRede para exclusão das aplicações de políticas associadas ao papel excluído. Se não houver sucesso na operação de exclusão, são emitidas mensagens *displayMessage e displayLog*, informando e registrando o insucesso. A Figura 31 apresenta o bloco gerente de políticas e os detalhe dos fluxos de sinais e das mensagens que compõem este bloco.

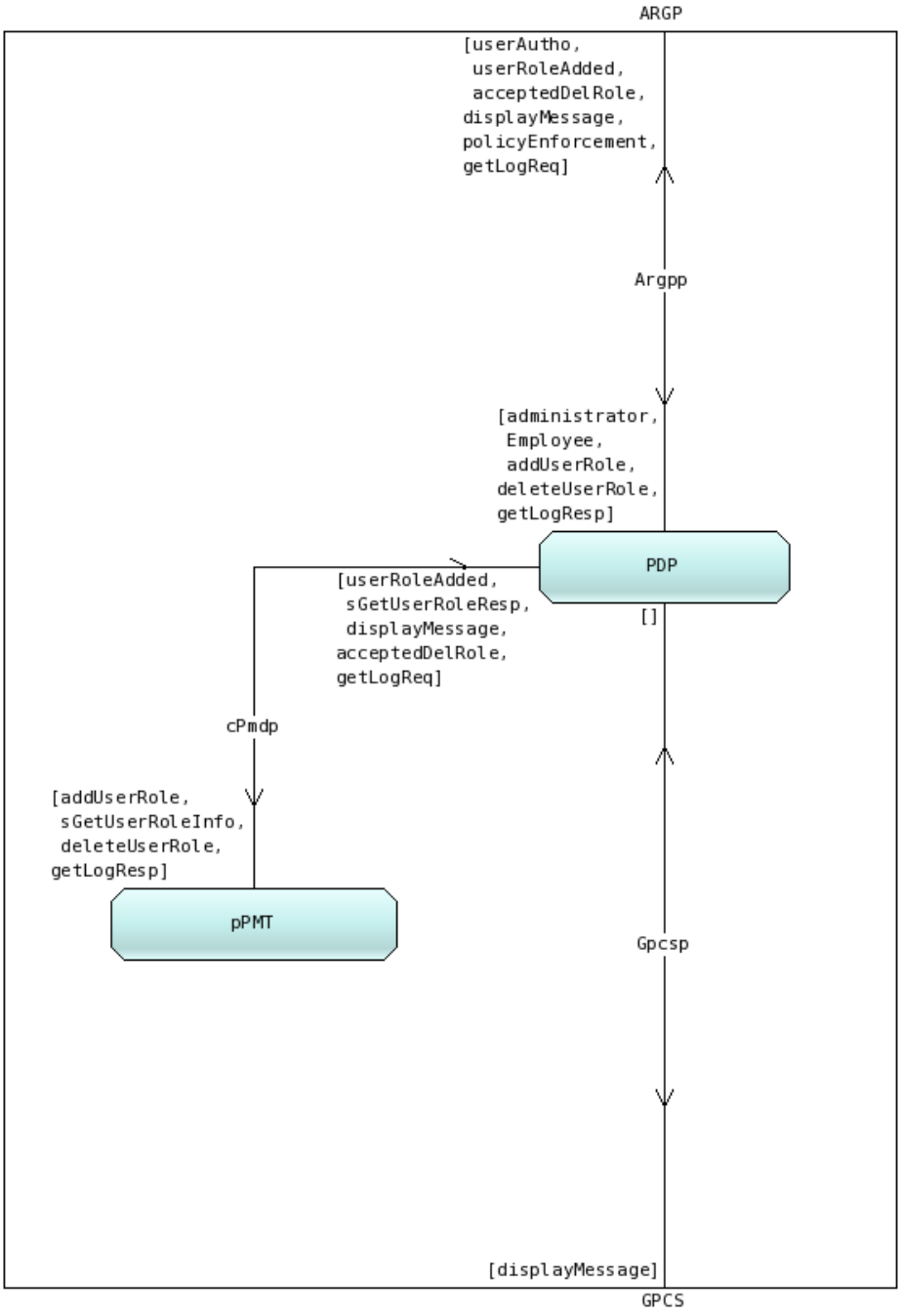


Figura 31 – Detalhamento dos Processos e Fluxo de Sinais do Bloco Gerente de Políticas (Fonte: Autor).

### 4.4.4 Bloco Serviço de Gerência de Usuários

A composição deste bloco é formada pelo processo *UserManager*, responsável pela seleção das ações de criação e eliminação de usuários. Ao receber a mensagem *userManager* o processo UserManager entra no modo de gerenciamento de usuários, oferecendo a opção para criação ou exclusão de usuário. Caso a opção seja a criação de usuário, o processo

solicita entrar com o novo cartão. Após a entrada de um cartão detectada pela mensagem *startAddUser*, o processo UserManager envia a mensagem solicitando a inclusão de um código de identificação do usuário. Após o recebimento do código, o processo UserManager emite a mensagem *regUser* com os parâmetros de identificação do cartão e do usuário para o bloco AcessoRede para registro de usuário na base de dados de usuários localizada nesse bloco.

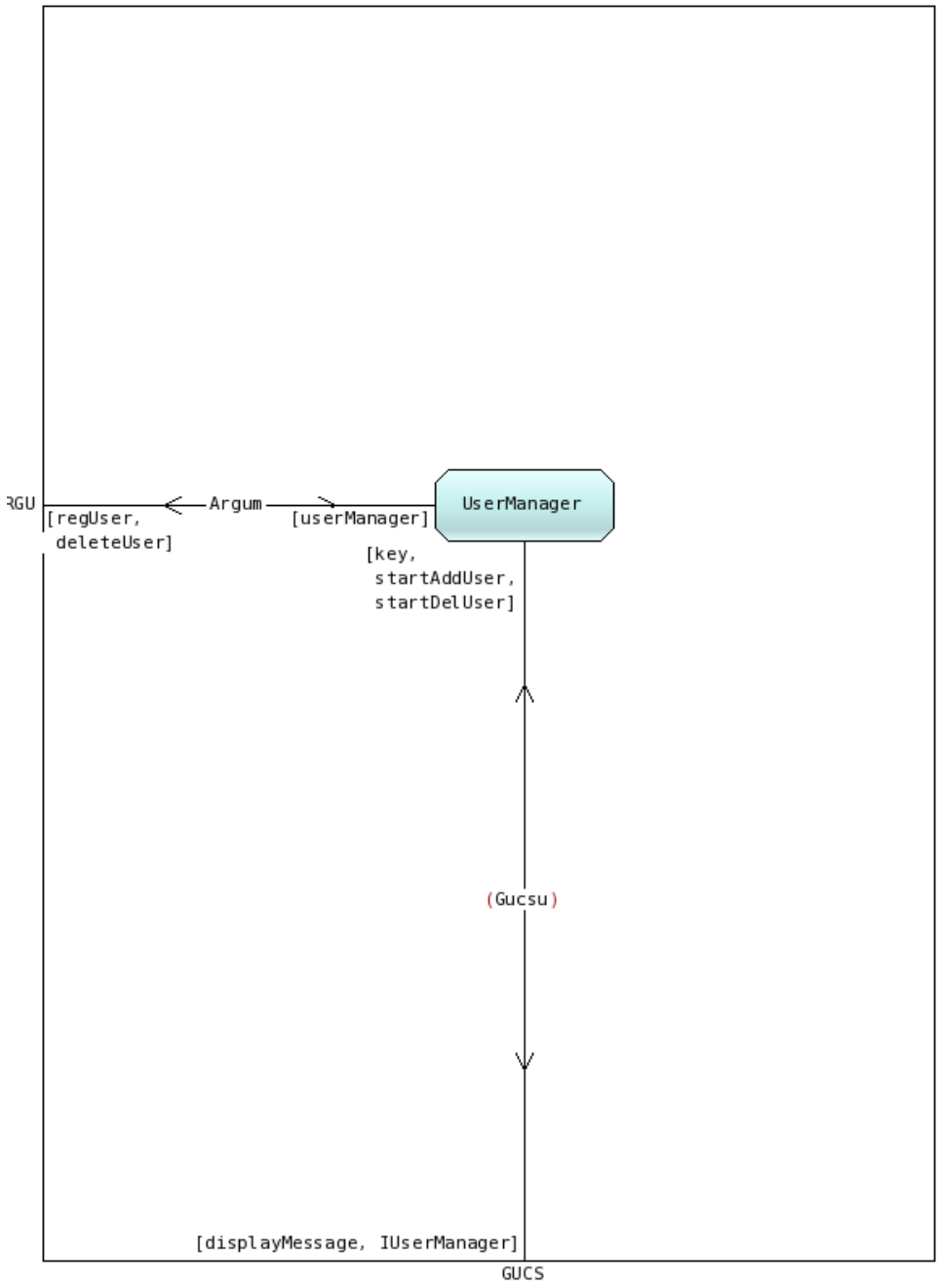


Figura 32 – Detalhamento dos Processos e do Fluxo de Sinais do Bloco Gerente de Usuário (Fonte: Autor).

Quando a opção for eliminação do usuário, o processo UserManager, após receber a mensagem *startDelUser*, solicita o código do usuário que será deletado. Posterior ao atendimento desta solicitação, o processo UserManager envia a mensagem *deleteUser* para o bloco AcessoRede para exclusão do usuário da base de dados. A Figura 32 apresenta o bloco gerenciamento de usuário, mostrando o fluxo de sinais e as mensagens associadas àquele processo.

### 4.4.5 Bloco Serviço de Acesso à Rede

O bloco de acesso à rede é constituído pelos processos ServAuth que especifica a função autenticação do usuário e a emulação da base de dados de usuários. Além deste, existe o processo PepeRede, que é responsável pelo gerenciamento das ações de aplicação de políticas ao nível de rede. Estas ações são emuladas na especificação do modelo, considerando que na prática elas são aplicadas a um *switch*. O terceiro e último processo é o ServAud, que especifica a função de auditoria das ações do usuário na rede.

O processo ServAuth, após o recebimento da mensagem *AuthReq*, verifica qual o tipo de usuário que está solicitando autenticação. No caso do usuário ter o papel de administrador, o processo ServAuth emite a mensagem *Administrator* para o bloco GerentePoliticas para associar o usuário a esse papel. Após o sucesso desta operação, o ServAuth recebe a mensagem *userAutho*, indicando que o papel de administrador foi associado. Nesta situação, o ServAuth emite a mensagem *UserManager* para o bloco GerenciaUsuario para entrar no ambiente de gerenciamento de usuário. No caso do usuário possuir outro papel na organização, o processo ServAuth verifica se o usuário está registrado. Se esta verificação for positiva, a mensagem *Employee* é enviada para o bloco GerentePoliticas para associar o usuário ao seu papel.

Outra atividade do processo ServAuth é realizada com o recebimento da mensagem *regUser*. Nesta circunstância, o ServAuth solicita o papel do usuário, via interface gráfica do usuário. No passo seguinte, o ServAuth recebe a mensagem *entry* contendo o papel do usuário e neste caso, envia a mensagem *addUserRole* ao bloco GerentePoliticas para incluir o papel do usuário na base de papéis. A mensagem *userRoleAdded* confirma que a operação foi realizada com sucesso. Neste caso, o ServAuth envia as mensagens *displayMessage e displayLog* para informar e registrar o estado da operação.

Na fase de eliminação de usuário, o ServAuth recebe a mensagem *deleteUser* e este realiza a busca na base de dados de usuários para verificar se está registrado. No caso da busca ser positiva, o ServAuth exclui o usuário da base e envia a mensagem *deleteUserRole* para o bloco GerentePoliticas para desassociação do usuário ao papel concedido para ele. A mensagem *acceptedDelRole* confirma o sucesso da operação. No passo seguinte, o ServAuth envia a mensagem *delEnfor* para o processo PepRede para exclusão da sessão corrente do usuário.

Ao receber a mensagem *policyEnforcement*, o processo PepRede realiza o registro de uma variável que emula a sessão do usuário. Essa é do tipo *array* e sua estrutura é composta pelo identificação do cartão, do papel do usuário e da ação a ser tomada (política). Após a criação da sessão, o PepRede envia mensagens *displayMessage e displayLog* para informar e registrar o sucesso da operação. Outro procedimento realizado por este processo é após o recebimento da mensagem *checkAuth*.

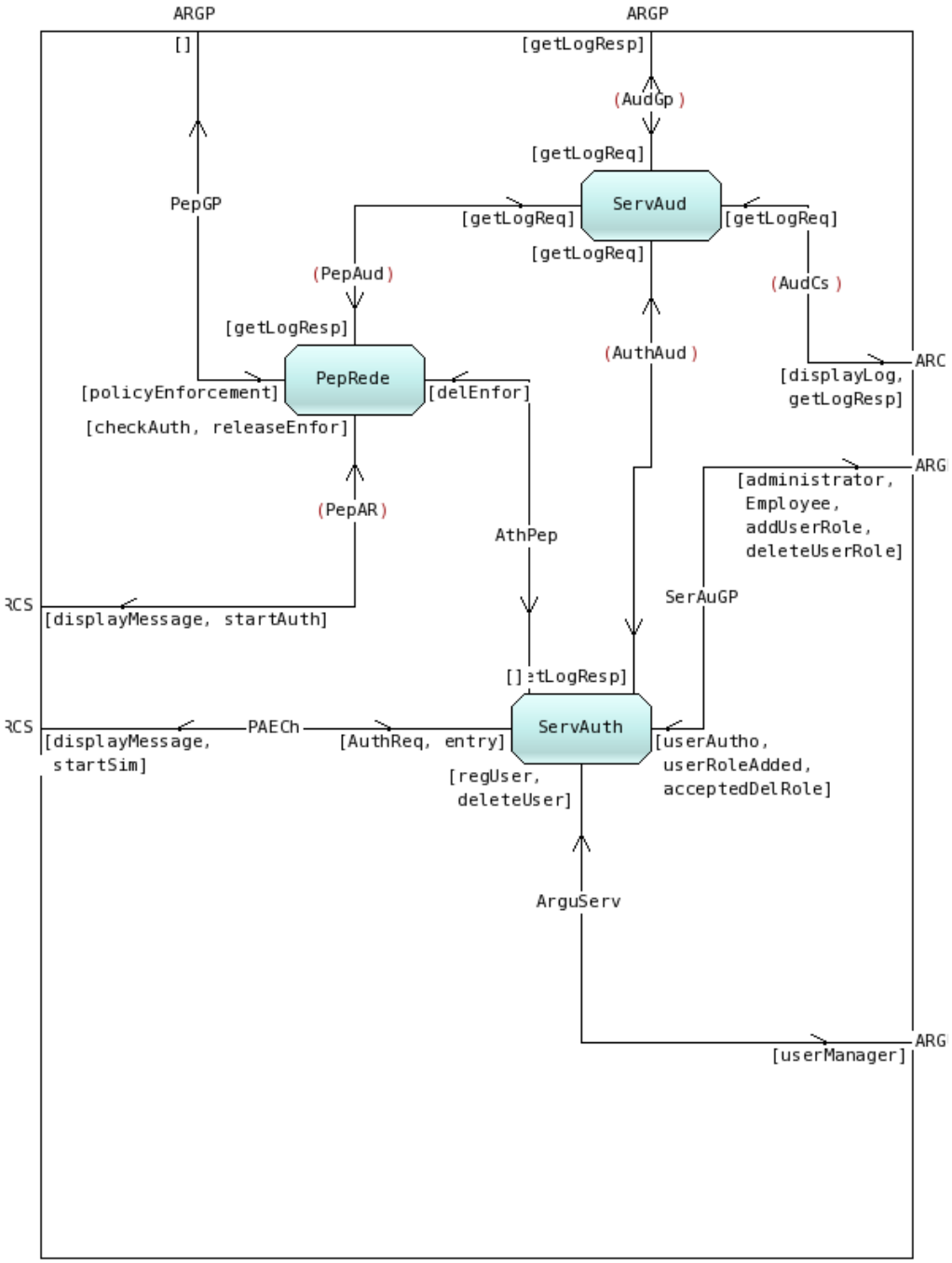


Figura 33 – Detalhamento dos Processos e do Fluxo de Sinais do Bloco de Acesso à Rede (Fonte: Autor).

Neste caso, o PepRede realiza uma busca na base de sessão, para verificar a existência de sessão em andamento para o cartão pesquisado. No caso de o resultado ser positivo, o PepRede emite mensagem *displayMessage e displayLog*, para informar e registrar que o usuário já está logado. Se o resultado for negativo o PepRede emite a mensagem *startAuth* para dar continuidade com o processo de autenticação do usuário.

Na situação em que o PepRede recebe a mensagem *releaseEnfor* decorrente da retirada do cartão do leitor de cartão, o PepRede realiza uma busca na base de sessões para verificar se existe uma sessão corrente para o cartão. No caso de a busca ser positiva, o PepRede exclui a sessão e emite mensagens informando e registrando o estado da operação. No caso de a busca ser negativa, emite mensagem informando que usuário não está logado.

No cenário de eliminação de usuário, o processo PepRede recebe uma mensagem *delEnfor* e também realiza uma busca na base de sessão para verificar a existência de sessão corrente para o cartão. Uma vez localizada a sessão para o cartão, o PepRede exclui a sessão e emite mensagens informando e registrando o sucesso da operação. Caso não exista uma sessão corrente para o cartão, o PepRede envia mensagem informando e registrando que usuário não está logado. A Figura 33 mostra o detalhes dos processos do bloco AcessoRede e os fluxos de sinais que interligam esses processos.

## 4.5 Considerações Finais do Capítulo

Este capítulo introduziu o modelo de ilha de segurança para as redes de automação de sistemas elétricos, utilizando o conceito de controle de acesso baseado em RBAC. O modelo foi especificado na linguagem formal de especificação e descrição SDL. Este modelo teve como premissa prover uma solução de segurança utilizando o conceito de controle de acesso para redes de automação de sistemas elétricos. O modelo resolve o problema de que os IEDs existentes ainda não têm a capacidade de executar mecanismos de segurança criptográficos para atender os requisitos de tempo exigidos pelas as operações em tempo real. Outra vantagem desta solução, é que a sua implantação não exige a substituição do parque IED existente. Nesse caso, eliminam-se os transtornos nos sistemas elétricos e nos consumidores ocasionados por esta substituição.

No Capítulo 5, serão apresentados os resultados dos testes de validação através de simulação da execução da especificação do modelo, emulando a interação do usuário com o modelo. Um segundo método de validação apresentado é a utilização de casos de testes empregando a linguagem de testes TTCN-3. Além desses testes, um terceiro método de validação do modelo foi realizado por meio da implementação em um ambiente real e em laboratório. Este ambiente é uma rede de automação de um centro de controle de sistemas elétricos.

Esse modelo poderá ser aplicado a qualquer sistema de automação da infraestrutura crítica. Neste caso, os requisitos de tempo real imposto ao modelo será de acordo com as

exigências do processo primário que o sistema de automação está controlando.

# 5

# IMPLEMENTAÇÃO DO MODELO

Neste capítulo, será apresentado o estudo de caso do modelo de segurança proposto nesta Tese. Inicialmente, o modelo foi validado pela verificação da sua funcionalidade através da execução de sua especificação em linguagem SDL. Essa funcionalidade simulou ações necessárias para o gerenciamento e autenticação de um usuário na ilha de automação. Nessa etapa, as ações são executadas via uma interface gráfica que emula o leitor de cartões e os dispositivos de entrada e saída do usuário. A segunda etapa de verificação foi realizada utilizando um caso de teste conforme estabelecido na linguagem TTCN-3. Como etapa final, o modelo da ilha de segurança foi validado em um ambiente emulando um cenário real de uma rede de automação aplicada a um centro de controle de uma empresa de energia elétrica.

## 5.1 Validação da Especificação SDL do Modelo da Ilha de Segurança

Para realizar o projeto da especificação em linguagem SDL, foi utilizada a ferramenta PragmaDev studio 5.12 . Essa ferramenta tem capacidade de realizar funções de especificação, desenvolvimento, rastreamento e testes no contexto da linguagem SDL. No ambiente de teste, a ferramenta pode realizar simulações do modelo através da execução de sua especificação ou de um módulo de teste em linguagem TTCN-3.

A validação da especificação do modelo foi realizada em duas fases: a primeira, através de teste de simulação verificando o comportamento das principais funções da ilha de automação. Esses testes foram realizados manualmente através de interação de usuários com o sistema, no papel de administrador e de empregado, verificando-se o comportamento no modelo através da interface gráfica ao usuário (GUI), definida na ferramenta de simulação. Uma segunda fase de verificação foi realizada utilizando a linguagem TTCN-3. O caso de teste contém a simulação da execução das principais funções que operacionalizam uma ilha de segurança. O conjunto de testes utilizados nessas fases foi estabelecido conforme a relação apresentada a seguir:

1. Autenticação do administrador;
2. Criação de usuário;
3. Autenticação de um usuário registrado;
4. Tentativa de autenticação de um usuário não registrado;
5. Tentativa de autenticação de um usuário já autenticado;
6. Retirada de cartão do leitor de cartão de um usuário não autenticado;
7. Tentativa de eliminação de um usuário não registrado;
8. Retirada de cartão do leitor de cartão de um usuário já autenticado;
9. Eliminação de um usuário registrado.

### 5.1.1 Testes Utilizando Simulação e Interação do Usuário com a Ilha de Segurança

Neste contexto, foi utilizada uma ferramenta de simulação de linguagem SDL. Essa ferramenta, além da edição e compilação da linguagem SDL, tem a capacidade de implementar uma interface gráfica configurada, conforme a necessidade do usuário. Neste caso, essa interface foi utilizada para emular o leitor de cartões e os dispositivos de entrada e saída utilizado pelo usuário.

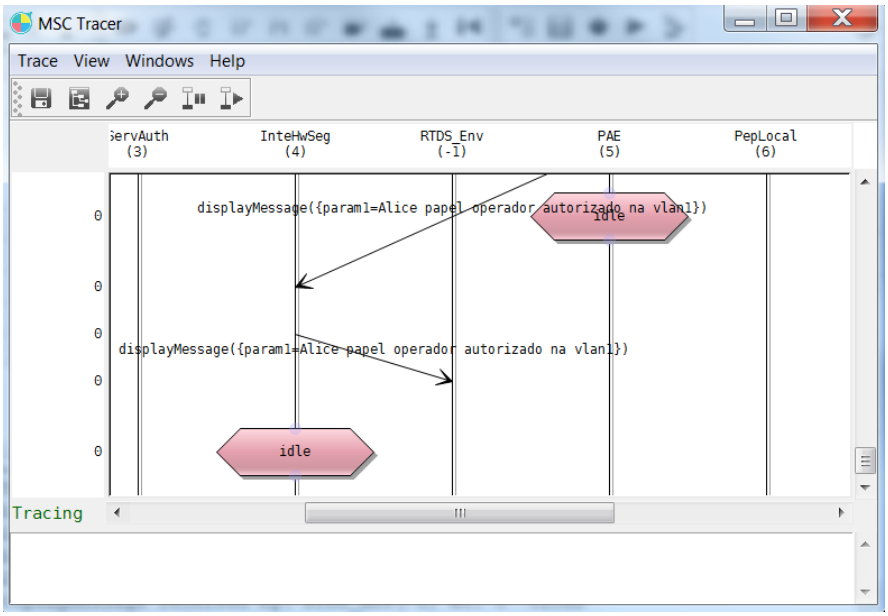


Figura 34 – Gráfico de Sequência de Mensagens para o Teste de Autenticação de Usuário (Fonte: Autor).

Neste cenário, os testes do conjunto de teste apresentaram resultados satisfatórios. A Figura 34 ilustra a imagem do gráfico do rastreamento da sequência de mensagens, do acrônimo em inglês *Message Sequence Charts (MSC)*, para o teste de autenticação de usuário.

### 5.1.2 Testes Utilizando Simulação e Linguagem de Teste TTCN3

A validação do modelo da ilha de segurança foi realizada utilizando uma ferramenta de simulação de linguagem SDL que tem a capacidade de realizar testes da especificação do modelo em SDL utilizando a linguagem TTCN-3. A especificação do modelo da ilha de segurança é considerada o sistema sob teste, e o seu módulo de teste é composto pelos seguintes arquivos:

- TTCN_Declarations.ttcn3 contém as declarações do tipo de variáveis necessárias ao módulo de teste;
- TTCN_Templates.ttcn3 contém as declarações das mensagens utilizadas no teste;
- TTCN_TesteAndControl.ttcn3 apresenta a sequência de entradas a que o teste será submetido;
- TTCN_CControler esse arquivo funciona como o módulo principal do teste e contém a sequência de execução deste.

O código a seguir mostra um extrato do arquivo de declaração da linguagem TTCN-3 utilizado nos testes de validação da ilha de segurança. O Apêndice A apresenta o conjunto de arquivos de configuração para o módulo de teste usado na validação da ilha de segurança.

**Arquivo TTCN_ Declarations.ttcn3**

```
module TTCN_Declarations {
//  New types declarations
  type record TTCN_LogInfoType {
    integer logTime,
    charstring logMessage
  }
// Records declaration
  type enumerated cardWithdraw { e_cardWithdraw }
  type record displayMessage {
    charstring param1
    }
  type record key {
```

```
    charstring param1
    }
  type enumerated inicio { e_inicio }
  type record entry {
    charstring param1
    }
  type record CardInsert {
    integer param1
    }
  type record displayLog {
    TTCN_LogInfoType param1
    }
// Ports declaration
  type port port_HSCS message
    {
    out CardInsert;
    out cardWithdraw;
    out startSim;
    out entry;
    out key;
    in displayMessage;
    in displayLog;
    }
// TSI and MTC component declaration
  type component runsOn_ilha {
    port port_HSCS HSCS;
  }
  type component ilha {
    port port_HSCS HSCS;
  }
}
```

Os resultados dos testes da especificação SDL podem ser observado tanto no ambiente de simulação quanto no ambiente do caso de teste, conforme apresentado na Figura 35.

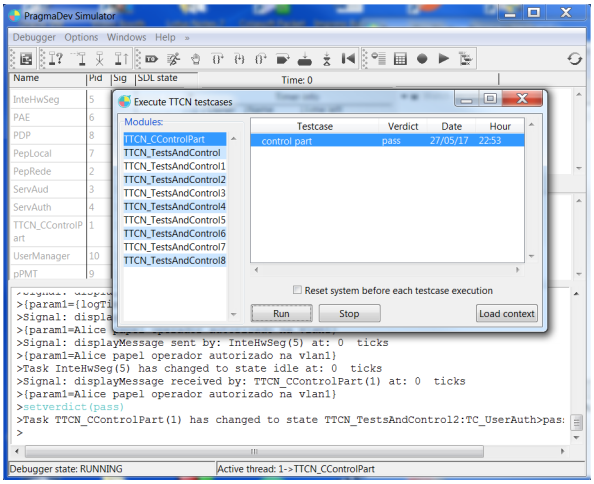


Figura 35 – Resultado do Teste do Modelo Ilha de Segurança Utilizando as Ferramentas de Simulação e o Caso de Testes (Fonte: Autor).

Outra forma de observar os resultados dos testes é através do diagrama de sequência de mensagens gerado pela ferramenta de rastreamento. A Figura 36 mostra um extrato dos resultados do caso de teste de validação da ilha de segurança utilizando o diagrama de sequência para o teste de autenticação de usuário.

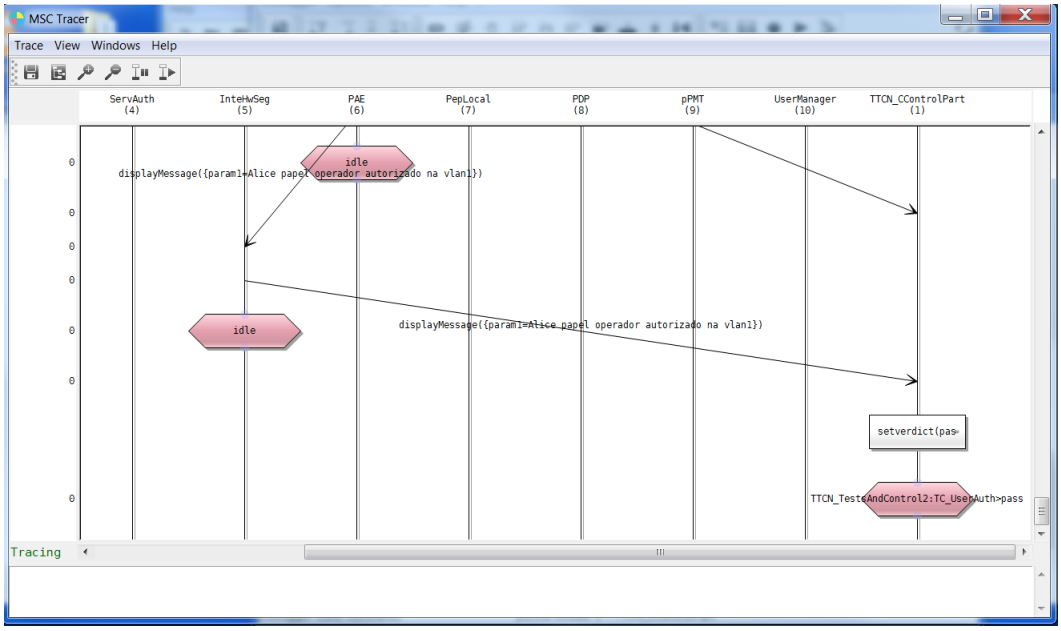


Figura 36 – Resultado da Execução do Módulo de Teste do Modelo Ilha de Segurança Utilizando TTCN-3 (Fonte: Autor).

## 5.2 Validação do Modelo em um Cenário Real

A rede de automação dos centros de controle de sistema é responsável pelo suporte computacional e de comunicações para as funções de controle e supervisão necessárias à

operação do SEP. No caso de uma eventual invasão desta rede, possibilitará ao atacante meios de obter o controle de todo o sistema elétrico de potência. Com essa preocupação, a CHESF e o Grupo de Pesquisa em Redes e Telecomunicação do Centro de Informática da Universidade Federal de Pernambuco desenvolveram um projeto de pesquisa e desenvolvimento, denominado de Segurança da Informação de Redes de Controle e Automação (SIRCAM), para prover uma solução de segurança cibernética para as redes de automação da CHESF. O modelo proposto nesta Tese serviu de base para a implementação do estudo de caso do sistema de segurança proposto pelo projeto SIRCAM.

Os usuários deste cenário são os servidores fornecedores de serviços das redes de automação, as equipes de operadores de sistemas elétricos, as equipes de manutenção de sistemas de automação, de manutenção de base de dados, de integradores dos sistemas de automação e a equipe de segurança cibernética. No caso específico da aplicação do conceito de ilha de segurança para as redes de automação de um centro de controle de sistemas elétricos, os principais serviços que serão protegidos são:

1. Controle, supervisão e aquisição de dados (SCADA);
2. Gerenciamento de energia (EMS);
3. Treinamento de operadores;
4. Base de dados histórica;
5. Configuração de base de dados;
6. Comunicação com as unidades terminais remotas e sistemas digitais;
7. Comunicação com o Operador Nacional do Sistema Elétrico (ONS).

### 5.2.1 Arquitetura de Teste para Validação do Modelo em um Cenário Real

A arquitetura deste ambiente é formadas por duas máquinas físicas que hospedam VMs, que dão suporte a cada serviço de segurança de forma individual. O sistema oferece um arranjo em redundância de forma que, na falha de qualquer serviço de segurança, o sistema ainda continua em operação. Cada máquina física hospeda uma instância dos seguintes serviços:

1. Serviços de SGDP;
2. Banco de dados *postgres*;
3. Servidor Remote Access Dial In User Service (RADIUS);
4. Servidor de acesso remoto *strongswan*.

A ICP está implementada em uma máquina exclusiva, que é protegida por um *Trusted Platform Module (TPM)* que tem como finalidade garantir a inviolabilidade desse ambiente. Para atender ao requisito de redundância do sistema, as bases de dados dos serviço de banco de dados que dão suporte à ilha de segurança são sincronizadas, utilizando os recursos do próprio SGBD. A Figura 37 detalha a arquitetura de teste construída para o cenário real da implementação da ilha de segurança.

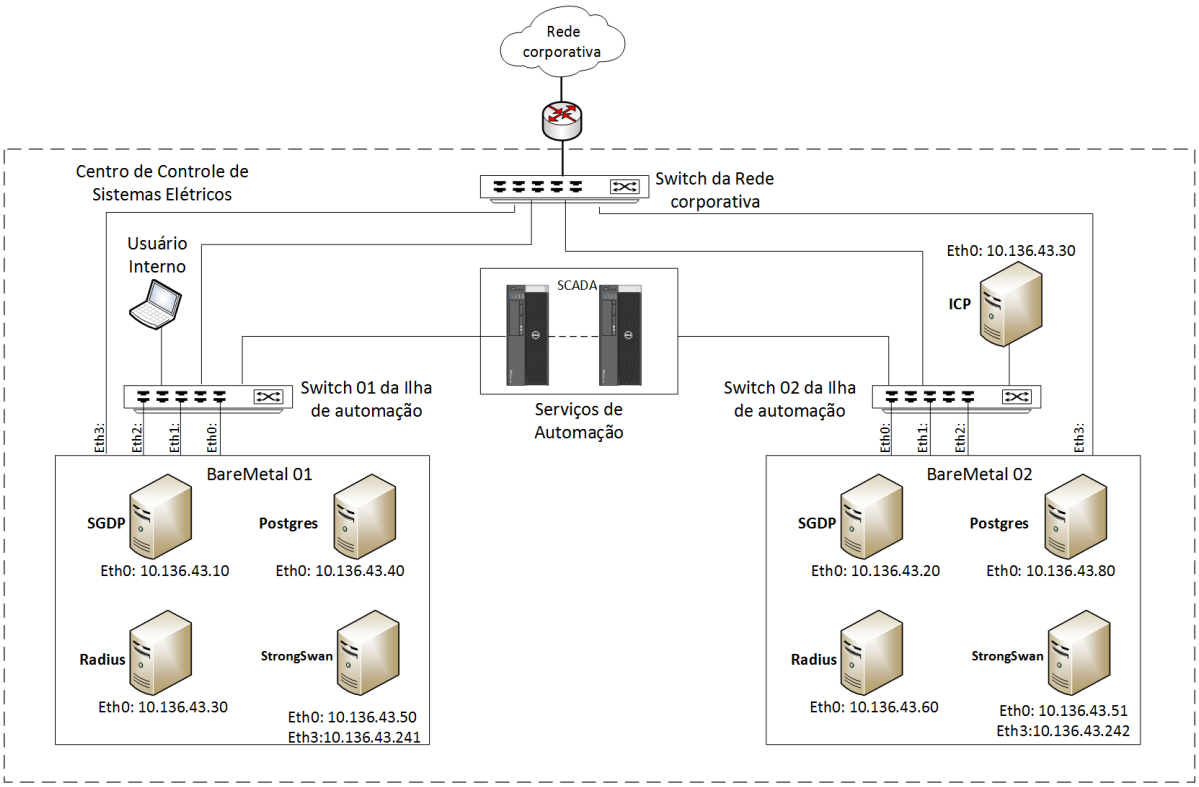


Figura 37 – Cenário Real da Ilha de Segurança para um Centro de Controle de Sistemas Elétricos (Fonte: Autor).

#### *5.2.1.1 Servidor de Acesso de Redes*

O servidor de acesso de redes (NAS) foi arquitetado utilizando o *switch Enterasys*, modelo A2H124-24P. A escolha deste equipamento foi devido ao fato de ele possuir os requisitos dos protocolos de segurança necessários para atender às funções de autenticador no modelo proposto, em conformidade com as especificações *Request for Comments (RFC)* do *RADIUS* e das necessidades de comunicações exigidas nos centros de controle.

Este *switch* possui as seguintes características: (a) permite a utilização de processo de autenticação através da utilização de servidores *RADIUS*, em conformidade com a RFC 2865; (b) usa protocolos *Extensible Authentication Protocol (EAP)* sobre *RADIUS* e autenticador EAP com protocolo *Extensible Authentication Protocol Over LAN (EAPOL)*; (c) suporta o protocolo IEEE 802.1x (IEEE, 2006), com capacidade de implementar redes

locais virtuais VLANs em atendimento ao padrão Ethernet IEEE 802.1Q (IEEE, 2014); (d) e possui atribuição dinâmica de VLAN (auto vlan) via *RADIUS*.

O nível de controle de rede foi implementado utilizando o conceito de VLANs, conforme o padrão 802.1Q, visto que este permite a separação lógica das sub-redes com o controle de entrada, de saída e de aplicação de filtro de quadros nas portas do *switch*. A configuração do controle de acesso à VLAN pode ser realizado estaticamente no *switch* ou de forma dinâmica pelo servidor de autenticação. É importante salientar que não basta conseguir a autenticação no *switch*, é preciso que exista autorização para ingresso às VLANs. Nesta implementação, a alocação final das VLANs é realizada pelo servidor SGDP.

O número de VLANs implementado depende da arquitetura do sistema de automação e das políticas definidas no projeto de segurança da corporação. Para o ambiente de controle, foram criadas duas VLANs de serviços, uma configurada dinamicamente, que é a de usuários autenticados, e outra alocada estaticamente, para permitir o sincronismo dos bancos de dados.

O *switch* foi configurado para suportar o protocolo EAPOL de forma a realizar a comunicação com o Cliente SGDP (suplicante). Este cliente foi implementado com JRadius -1.1.5 e realiza as interações necessárias para o usuário obter o acesso à rede. A Figura 38 ilustra a sequência de negociação dos protocolos EAPOL e *RADIUS*. A função de suplicante (PAE) está embutida na implementação do cliente SGDP. Esta é responsável pela comunicação com o autenticador. Para realizar a troca de informações com o servidor de autenticação RADIUS, o *switch* foi configurado para suportar EAP sobre RADIUS.

#### *5.2.1.2 Servidor de Autenticação*

A autenticação na ilha de segurança é fundamentada no controle de acesso baseado em políticas que foram identificadas, através de levantamento de requisitos com os usuários. Neste processo, estabeleceu-se que o controle de permissões se limitaria a três itens: às aplicações do sistema de automação, aos diretórios e aos serviços da rede. Portanto, foram elaborados níveis de controle de acesso para a rede e para os sistemas operacionais.

O módulo de autenticação foi construído utilizando o *FreeRADIUS14 server* 2.2.5 como servidor de autenticação. Este foi configurado para responder à autenticação via protocolo 802.1X, empregando o protocolo EAP e usando o protocolo *Transport Layer Security (TLS)* como método de autenticação. A escolha do textitFreeRadius ocorreu por ser um dos servidores RADIUS *opensource*, bastante conhecidos e utilizados por diversas empresas (FREERADIUS, 2016), além de estar disponível na maioria das distribuições Linux.

Para armazenar as políticas de autorização da rede e realizar a persistência de algumas configurações do servidor *FreeRadius*, foi instalado e configurado o SGBD *Postgres* 9.3. O motivo principal que influenciou na escolha do *Postgres* foi a existência de um módulo de integração com o servidor de autenticação RADIUS.

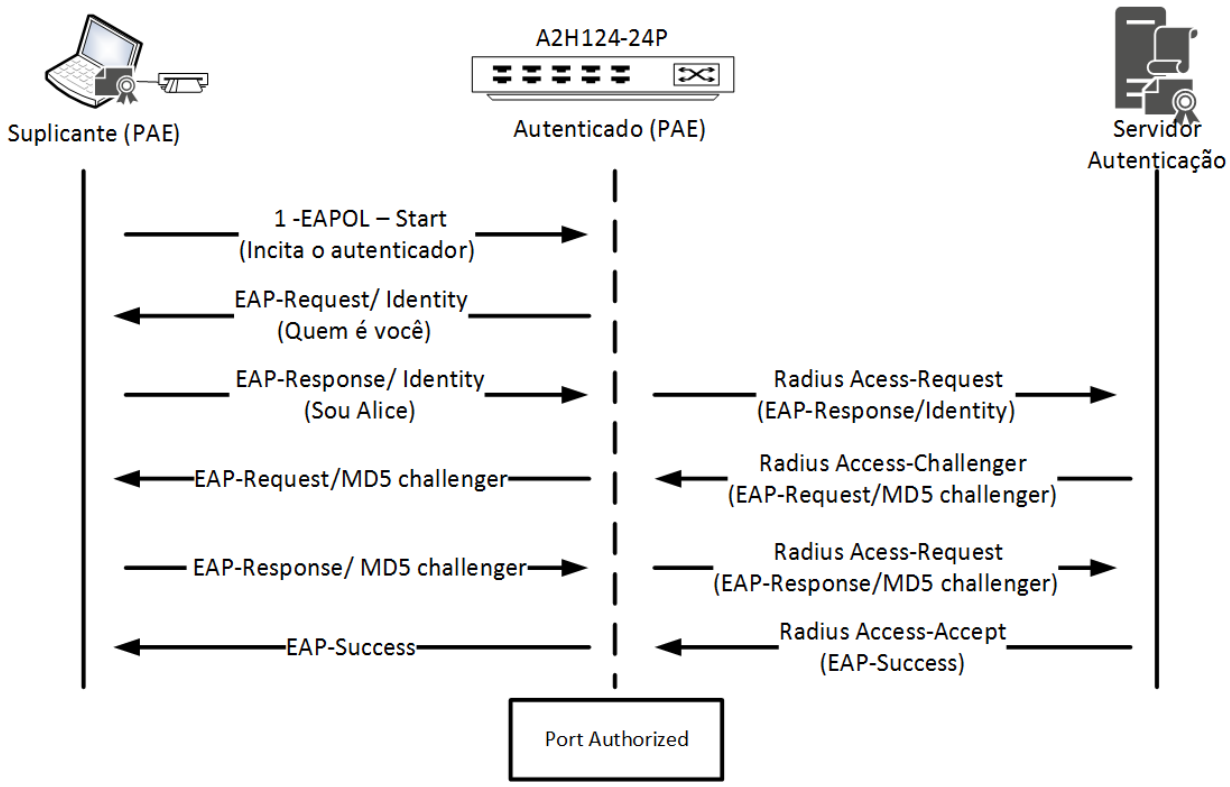


Figura 38 – Diagrama de Sequência da Comunicação entre o Suplicante e o Servidor de Autenticação (Fonte: Autor).

Para a conexão entre o servidor de autenticação e o servidor SGDP, foi utilizado o JRadius 1.1.5, o qual é constituído por um módulo e uma biblioteca no servidor *FreeRadius*. A Figura 39 mostra o detalhe do fluxo de informação entre o SGDP e o servidor RADIUS.

#### *5.2.1.3 Servidor de Acesso Externo*

O servidor de acesso externo foi implementado utilizando o *strongSwan*. Essa é uma solução de Ipsec para Linux. O foco dessa aplicação é o mecanismo de autenticação que usa certificados de chave pública, conforme o padrão X.509, e pode opcionalmente realizar o armazenamento seguro da chave privada em cartão inteligente de acordo com a interface PKCS#11.

Este recurso utiliza o protocolo *Internet Key Exchange (IKE)*, nas versões um e dois, de forma a estabelecer uma associação de segurança (SA) entre dois pares. Este proporciona autenticação forte de ambos os pares e fornece uma chave de sessão criptográfica exclusiva para a sessão em execução. Este processo é denominado de IKE_SA. O IKE realiza o procedimento de autenticação em duas fases.

Na primeira fase, o objetivo é autenticar os pares IPSec e configurar um canal seguro entre eles para habilitar a troca de IKE. Esta fase consiste basicamente na troca de

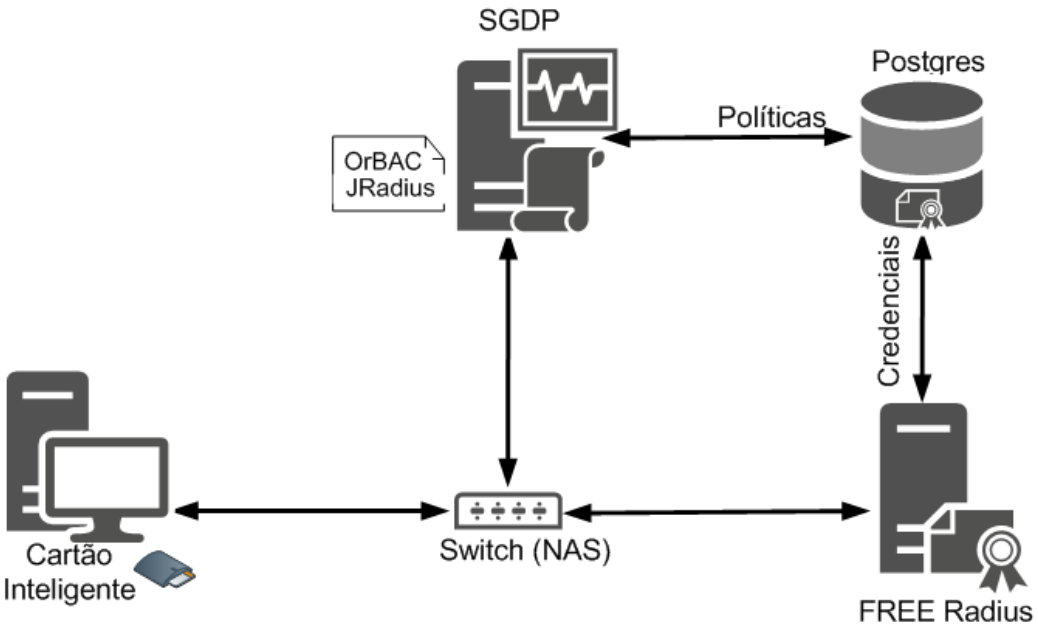


Figura 39 – Arquitetura de Autenticação Radius (Fonte: Proj. Sircam).

mensagem em texto claro, devido ao fato de que os parâmetros de criptografia e as chaves compartilhadas ainda não foram negociados. Estas são estabelecidas usando o acordo de chave de Diffie-Hellman (MEL; BAKER, 2001).

Nesta etapa, as seguintes funções são realizadas: autenticação mútua entre os pares IPSec, definição de um conjunto de políticas IKE SA para proteger a troca de IKE, configuração de um túnel seguro e execução da negociação dos parâmetros IKE que serão utilizados na fase seguinte. Ao término desta, foi configurado um canal seguro entre os pares. Após a geração da primeira chave compartilhada, o protocolo realiza de forma segura as atividades de criação de uma chave de encriptação, de uma chave de autenticação e um número secreto adicional.

Na segunda fase, o IKE negocia IPSec SAs para configurar um túnel IPSec. Além disso, são executadas as seguintes funções: negocia os parâmetros de proteção IPSec AS para uma determinada IKE AS, estabelece associação de segurança para o IPSec, renegocia periodicamente IPSec SAs para garantir a segurança da conexão e executa opcionalmente o algoritmo de acordo de chave Diffie-Helmman.

Na implementação do modelo, o *strongSwan* foi instalado em uma máquina virtual que possui quatro interfaces de rede. Desse modo, duas são conectadas ao servidor de acesso que receberá o tráfego interno da rede de automação e as outras duas são conectadas aos *switches* de borda da rede corporativa.

#### *5.2.1.4 Serviço de Gerenciamento e Decisão de Políticas*

É o componente central do processo de autorização, sendo responsável por avaliar as requisições do cliente SGDP de acordo com as políticas aplicáveis e informar para o

servidor de autenticação *Radius* uma decisão de autorização ou não, de forma que este possa ordenar a aplicação desta decisão ao autenticador NAS.

O SGDP agrega algumas funções do modelo de segurança da ilha de automação. Sua função principal é realizar a etapa de autorização no processo de controle de acesso. Após a conclusão do processo de autenticação e considerando que o usuário já foi autenticado, o SGDP inicia a fase de checar e estabelecer as permissões do usuário aos serviços, aos recursos de rede e aos recursos do sistema operacional.

No processo de determinar as permissões de um usuário, o SGDP consulta a base de políticas de forma a obter as informações necessárias sobre os direitos de acesso aos recursos desejados pelo usuário. A decisão sobre as permissões do usuário é realizada pelo módulo *PDP*, conforme o módulo de gerenciamento baseado em políticas apresentado na Figura 40.

Existem vários modelos organizacionais de gerência de políticas de controle de acesso. Dois fatores precisam ser avaliados antes da escolha desses: a estrutura da corporação e a granularidade dos recursos que serão geridos pelas políticas. O SGDP realiza o gerenciamento permitindo a criação, atualização e eliminação das políticas definidas pela organização para a segurança da ilha de automação. O módulo prevê uma *Graphical User Interface (GUI)* para o SGDP, que possui recursos avançados necessários para o gerenciamento de políticas, permitindo a definição de regras, o controle de usuários e dos recursos da rede, o acesso aos registros de eventos e a simulação das consequências das políticas na rede.

Além disso, o SGDP realiza o controle de certificados para os usuários através da comunicação do SGDP como servidor de ICP e com o leitor de cartões inteligentes. A comunicação com este leitor é realizada através do cliente SGDP. Após a etapa de autorização, os usuários estão efetivamente no ambiente da ilha de automação e terão acesso a todos os recursos a eles destinados. Em seguida, o SGDP inicia o controle dos registros de eventos, a coleta de informações sobre as ações executadas por um usuário, a data de acesso, o período de uso e os demais dados necessários para o processo de auditoria.

O SGDP é uma ferramenta de acompanhamento de rede e, para tal, além de se comunicar com o servidor de banco de dados para persistência, comunica-se também com o servidor de controle de acesso e com o ICP.

A camada de comunicação provê uma interface de troca de mensagens de forma aberta com outros componentes da rede, através da transferência de pacotes sobre TCP/IP em texto claro ou de forma protegida, usando um túnel SSL. Esta camada também possui uma interface direta com o servidor de autenticação (*RADIUS*), para fins de monitoramento do sistema. O tradutor faz as devidas conversões de pacotes de mensagem em estruturas compreendidas pelo SGDP e vice-versa. O módulo repositório de credenciais faz a comunicação com o servidor de banco de dados, executando o papel de ponte para a persistência de dados de usuários, recursos e políticas.

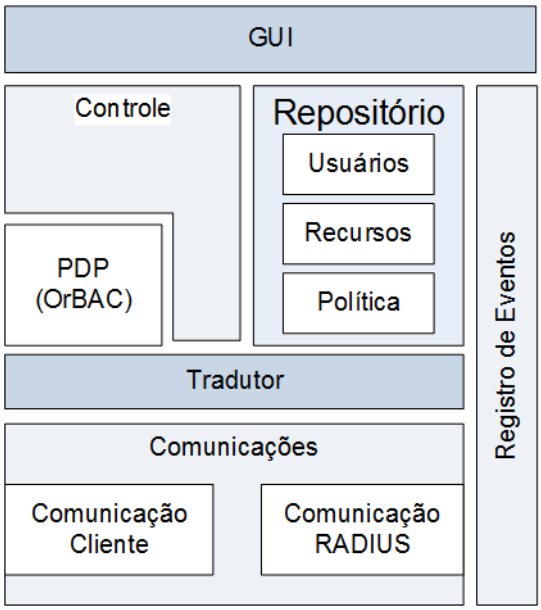


Figura 40 – Arquitetura do Serviço de Políticas e Controle de Acesso (Fonte: Proj. Sircam).

O submódulo de controle centraliza a troca de mensagens entre os módulos e facilita o registro dos eventos.

#### *5.2.1.5 Cliente do Serviço de Políticas e Controle de Acesso*

No processo de autenticação e autorização, o cliente do serviço de gerenciamento e decisão de políticas executa três funções:

- Realiza a comunicação com um repositório de credencial ou com um dispositivo confiável para acessar o certificado armazenado;
- Executa o processo PAE de forma a processar a função de suplicante definida no padrão de autenticação 802.1X;
- Exerce as interações necessárias do processo de cliente do SGDP e realiza a função PEP local aplicando as políticas determinadas pelo SGDP e recebidas do servidor de autenticação.

Para realizar essas interações, a arquitetura do cliente é composta por módulos de softwares com a função de interface gráfica, controle, tradução e de comunicações. A interface gráfica, inicialmente, interage com o usuário apresentando bloqueios de tela, de forma que a única operação possível seja a inserção de credenciais. A liberação do bloqueio e o acesso aos serviços da rede devem ser realizados apenas após a autenticação do usuário.

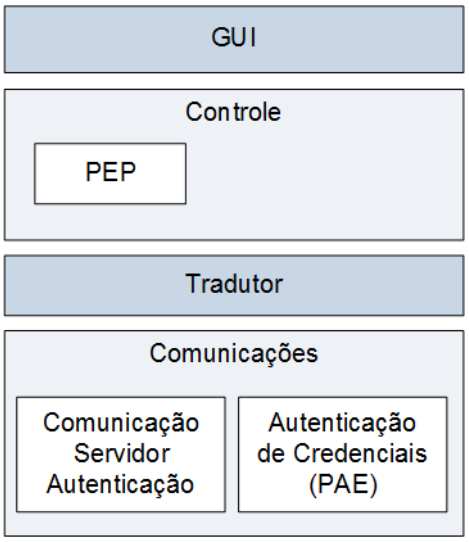


Figura 41 – Arquitetura do Cliente SGDP (Fonte: Proj. Sircam).

A Figura 41 detalha os principais blocos funcionais da arquitetura do cliente SGDP. O módulo de controle é responsável pelo processamento das funções de execução do PEP e executa as decisões tomadas pelo SGDP baseado nas informações do processo de autorização.

O módulo tradutor realiza as traduções necessárias entre o módulo de comunicações e os demais módulos das camadas superiores do cliente SGDP. O módulo de comunicações, inicialmente, executa uma solicitação de autenticação ao servidor de acesso à rede através do protocolo EAPOL. Este último envia um pedido de admissão para o servidor de controle de acesso à rede que utiliza as características de associação do protocolo EAP, com a finalidade de autenticar e autorizar os dispositivos conectados a uma determinada porta. O acesso é negado quando o processo de autenticação falhar. Neste contexto, a porta é considerada um ponto único de conexão com a infraestrutura da rede.

Após o usuário ser autenticado, é iniciada a fase do processo de autorização. Nela, o cliente SGDP envia uma requisição ao servidor SGDP para determinar quais os recursos de serviços, de redes e de sistema operacional que o usuário autenticado tem direito. Como visto anteriormente, este último consulta a base de políticas de forma a extrair as permissões do usuário em nível de sistema operacional local e tomar as decisões no PDP. Uma vez que a decisão foi estabelecida, o SGDP envia o resultado para o servidor de autenticação de forma que este realize o controle de portas ou para o cliente do SGDP para execução dela no nível local.

O cliente SGDP é um aplicativo implementado em Java. Esse cliente pode ser representado por uma máquina de estados. A Figura 42 ilustra a máquina de estado do cliente SGDP.

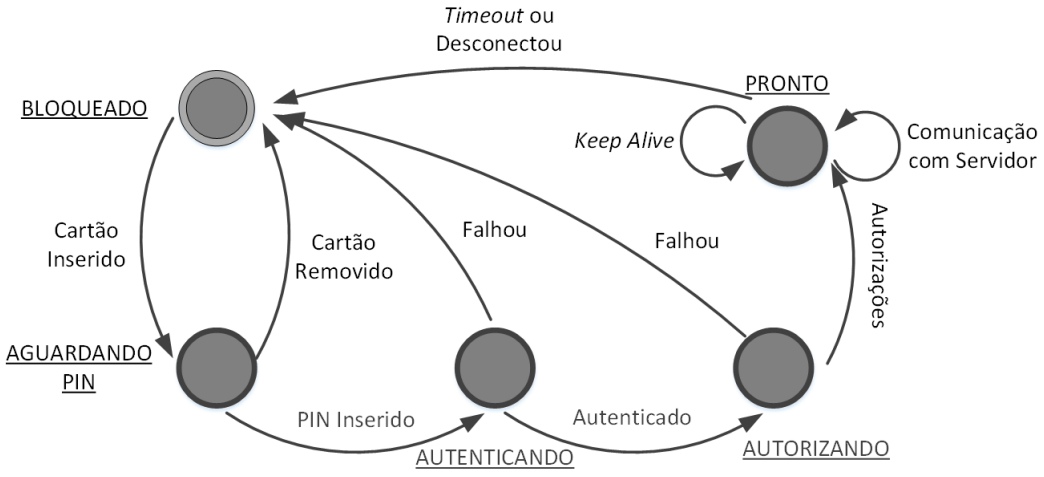


Figura 42 – Máquina de Estado do Cliente SGDP (Fonte: Proj. SIRCAM).

No estado inicial, a função principal do cliente SGDP, denominada de ciclo de vida do SGDP, ativa o módulo de visualização que realiza o bloqueio da interface gráfica, teclado e mouse e aguarda pelos eventos de inserção do cartão de autenticação e entrada da senha pelo usuário. Uma vez recebidos esses eventos, a função principal da aplicação aciona o PAE, implementado com *wpa_suplicant*, de forma que ele execute a sequência de associação com o NAS, utilizando o protocolo EAPOL.

Após a execução desta fase, o módulo JOrBAC do cliente do SGDP efetua a comunicação com o módulo de OrBAC do serviço SGDP para que este efetue o processo de autorização do usuário. Após a definição do processo de autorização, o SGDP retorna as decisões para o cliente SGDP que realiza a aplicação das políticas.

#### *5.2.1.6 Uso de Cartões Inteligentes na Implementação do Modelo*

O processo de interação com o cartão inteligente utilizado nesta Tese é realizado pelos módulos Cliente SGDP e pelo servidor de gerenciamento e decisão de política - SGDP. A troca de informação é realizada através do conjunto de bibliotecas OpenSC, SC/PC e JavaCard. A figura 43 ilustra o processo de interação entre a aplicação SGDP e o cartão inteligente. O módulo de aplicação neste contexto é o servidor SGDP e o cliente SGDP. O cliente SGDP utiliza a biblioteca OpenSC que permite a interação com o PC/SC. Esta é realizada através de uma biblioteca cliente PCSC-Lite, desenvolvida pelo grupo M.U.S.C.L.E.17.[1] O relacionamento do módulo SGDP com o PC/SC é realizado pela biblioteca cliente pcsclite.so.1 do pacote javax.smartcardio. O PC/SC executa a interface com o cartão, através da biblioteca ICCD.

[1] M.U.S.C.L.E é a sigla do movimento dos usuários de *Smart Card* em ambiente Linux (do inglês *Movement for the Use of Smart Card in a Linux Environment*).

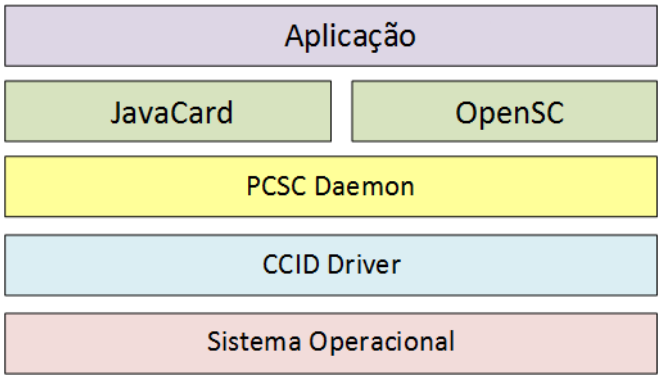


Figura 43 – Modelo de Interação Entre o SGDP e o Smart Card (Fonte: Proj. Sircam).

#### *5.2.1.7 Estrutura de Hardware e Geração de Certificados*

Para realizar a função de geração de chaves e assinatura de certificados, é utilizado um hardware autocontido e dedicado cujas ações são protegidas por um módulo TPM. Ele hospeda a função de autoridade certificadora (CA). Para realizar as funções de uma CA, como assinar e revogar certificados, foi utilizado o OpenSSL com a *libengine-tpm-openssl* e a *engine* do módulo TPM, permitindo, assim, utilizar o par de chaves do módulo. A comunicação do servidor SGDP com o hardware de geração de certificados é realizada utilizando-se o SSH.

O TPM foi desenvolvido como parte de um conceito na área de segurança de sistemas conhecido como *Trusted Computing* (TCG, 2005). O TPM tem a característica de ser embutido em uma plataforma e garantir, entre outras funcionalidades, a integridade e a autenticidade desta. A segurança baseada em um componente em hardware visa remover as vulnerabilidades originadas pelo processo realizado apenas por meio de software e disponibilizar um modelo de segurança mais robusto e confiável.

#### *5.2.1.8 Geração de Certificados*

Para a geração de certificados, foi necessário criar uma infraestrutura de chave pública própria para gerenciar os certificados criados e armazenados em cada cartão. A plataforma PC atua dentro da infraestrutura ICP, como uma Autoridade Certificadora (AC) e contém um módulo TPM que guarda a chave privada da AC e que garante que esta esteja íntegra e inviolável. A Figura 44 detalha o esquema de geração de certificados.

O processo de geração de certificados é dividido em cinco partes. No passo 1, o administrador insere um *smart card* novo e sem dados registrados para ser preenchido com as informações do usuário. Ainda no leitor de cartões, o passo 2 segue com a execução de comandos do *opensc-tools + pkcs11* para gerar a chave pública e privada e armazená-la no cartão. Em seguida, no passo 3, de posse do ID da chave privada do cartão, é gerado um arquivo '.csr' que será enviado para o servidor de ICP que está instalado na plataforma PC, para que o certificado seja emitido.

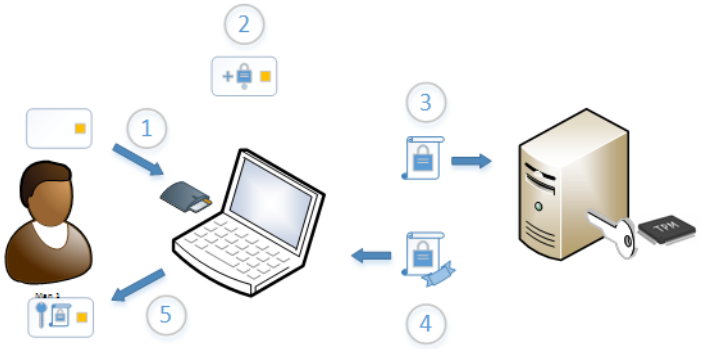


Figura 44 – Fluxo do Processo de Geração de Certificados (Fonte: Proj. SIRCAM).

Por fim, o CA retorna o certificado assinado, passo 4, que é salvo em conjunto com a chave privada do usuário no *smart card*, passo 5. Após esse processo, o usuário estará com um *smart card* pronto e apto para ser autenticado na rede da ilha de segurança.

#### *5.2.1.9 Persistência da Fase de Autenticação*

Para cada serviço de AAA, utiliza-se um serviço de persistência, normalmente instanciado por um banco de dados. No modelo de segurança proposto, as funções de autenticação são realizadas pelo servidor *RADIUS*, que usa uma instância de banco de dados para registrar as credenciais do usuário. Isso é, um repositório para armazenar os certificados válidos.

Embora este servidor tenha a capacidade de executar as funções de autorização e auditoria, essas funções são realizadas pelo módulo SGDP. Neste caso, os requisitos mínimos de persistência para este servidor são inferiores ao exigido quando do uso de sua funcionalidade plena (AAA). Além disso, a base de informações do *RADIUS* deve possuir acesso controlado para que ações paralelas não ocasionem a perda de funcionalidades do sistema.

As tabelas utilizada pelo *RADIUS* nesta fase foram: *radcheck, radreplay, nas, radusergroup, readgroupcheck* e *radgroupreply*. Em *radcheck*, os dados dos usuários finais da rede são verificados no momento de sua autenticação, e em caso de verificação bem-sucedida, os dados que devem ser informados aos clientes *switches* estarão armazenados em *radreply*. A tabela *radusergroup* associa os usuários individuais a grupos de acesso, mais fáceis de manipular e diretamente relacionáveis aos papéis de usuário. A lista de grupos gerenciados pelo *RADIUS* fica registrada na tabela *radgroupcheck*, e, de forma análoga ao que acontece com os usuários individuais, a tabela *radgroupreply* contém as informações de configuração dos clientes *switches* referentes aos grupos.

#### *5.2.1.10 Persistência Durante a Fase de Autorização*

O SGDP tem como principal recurso a facilidade na administração do ambiente protegido, através do gerenciamento baseado em políticas. Dentre as funcionalidades do SGDP, a centralização de administração da rede, através de políticas é uma das funcionalidades principais do processo de autorização. Para realizar a função de autorização adequadamente, é necessária a persistência de informações sobre os componentes envolvidos no processo. Para oferecer uma interface de administração adequada e completa, o SGDP precisa conter dados relevantes das redes, dos recursos destas e dos usuários que desejam obter acesso a esses recursos. Portanto, o SGDP necessita das seguintes informações:

- As sub-redes do sistema alocadas em VLANs, cada uma sendo uma entidade abstrata, contendo um identificador e os recursos que devem fazer parte da sub-rede;
- Recursos da rede são uma abstração para serviços, computadores pessoais, interfaces de uso associadas a uma sub-rede;
- Papéis de usuários, em que cada papel é uma abstração de uma série de atividades que podem ou não ser realizadas por um determinado grupo;
- Uma relação de usuários representando a entidade final que utiliza o sistema. Cada usuário é associado a um papel;
- Políticas, que são as regras que representam permissões ou proibições de papéis aos recursos. Este recurso deve dar suporte à resolução de conflitos por prioridade;
- Tipos de acesso que são um refinamento da forma como os usuários podem acessar os recursos.

A etapa de autorização do modelo da ilha de segurança é uma tarefa dividida entre o servidor de autorização SGDP e o servidor de autenticação *RADIUS*. O SGDP realiza o processo de decisão baseado na política do usuário e envia comandos para o servidor *RADIUS* que executa a aplicação da política em conjunto com o servidor de autenticação, política de rede, ou com o cliente SGDP, que executa a política ao nível local.

#### *5.2.1.11 Persistência na Fase de Registros e Auditoria*

Para obter uma melhor granularidade e um gerenciamento mais específico, o modelo da ilha de automação define que a função de auditoria das atividades dos usuários seja realizada pelo SGDP, em detrimento do uso dos recursos de auditoria do *RADIUS*.

O modelo da ilha de segurança utiliza um sistema de autenticação individual não-repudiável e um mecanismo de registros de eventos vinculados a esta autenticação. Por ser um registro de ações consultado apenas na ocorrência de eventos raros, a principal

exigência para persistência do registro de eventos é escrita rápida e capacidade de concatenação de campos associados aos diversos eventos. Como exemplo de estrutura dos registros, os *logs* podem conter as seguintes informações:

- Data e hora do evento;
- ID do usuário que executou a ação;
- Ação executada;
- Os recursos utilizados na ação;
- Informações adicionais, se aplicáveis.

## 5.3 Testes e Resultados do Cenário Real

Os testes realizados neste cenário visam avaliar o comportamento de um sistema baseado no modelo proposto considerando a resistência a invasão e o desempenho da solução. Esses testes ratificam os resultados obtidos nos testes de funcionalidades, utilizando a linguagem TTCN apresentada no Capítulo 2 e sessão 5.1.2, quando a especificação do modelo foi executada. Os testes de invasão da ilha de segurança têm como objetivo quebrar os mecanismos de segurança e realizar análises das respostas da implementação do modelo diante da tentativa de um acesso não autorizado. Os testes submetem a implementação proposta a ataques semelhantes, aos realizados por indivíduos ou organizações mal intencionadas. Os testes de desempenho buscam verificar o comportamento da implementação do modelo, diante dos aspectos de escalabilidade e de carregamento da solução de segurança apresentada.

### 5.3.1 Resultados dos Testes de Invasão

Os testes de invasão realizados neste cenário focam principalmente em tentativas de acessar de forma indevida uma rede de automação protegida por instância de implementação do modelo proposto. Os seguintes testes foram aplicados para este cenário:

1. Testes considerando o caso de invasão física da sala do servidor por um terceiro ou funcionário mal-intencionado, obtendo acesso a uma porta do *switch* configurada como autenticação forçada;
2. Testes considerando o caso de invasão física da sala do servidor por um terceiro ou funcionário mal-intencionado e tendo acesso a uma porta não gerenciável do *switch* e que esteja rodando um serviço;

3. Teste de acesso à máquina local da rede de automação. Ocorre quando um usuário não autenticado realiza uma tentativa de acesso a uma IHM ou uma máquina da rede de automação (VPNs necessárias);

4. Teste de acesso a um serviço não autorizado. Ocorre quando um funcionário, já autenticado no sistema, tenta usar um serviço e não possui permissão de acesso.

5. Testes de tentativas de acesso de um usuário não autorizado e externo à ilha de automação.

O modelo de ataque utilizado nesta Tese é o modelo de ataque diamante. Este modelo descreve que um adversário ataca uma vítima dependendo da superioridade de sua capacidade sobre algumas infraestruturas em relação à da vítima (CALTAGIRONE; PENDERGAST; BETZ, 2013). Este modelo consiste de quatro elementos básicos: adversário, infraestrutura, capacidade e vítima. Além desses elementos, o modelo possui meta-características como estampa de tempo, fases, resultados, direção, metodologia e recursos. A escolha deste tipo de modelo foi devido à facilidade de uso e da falta de necessidade de extenso detalhamento das fases como no caso do modelo gráfico de ataque.

#### *5.3.1.1 Teste de Invasão Física com Acesso a uma Porta Não Gerenciável do switch da Ilha de Automação*

O objetivo do teste é obter acesso a uma porta não gerenciável do *switch* da ilha de automação.

Modelo de ataque:

1. Vitima: O *switch* da rede de automação;

2. Capacidade: Ferramenta de varredura de porta e de ataque de dicionário;

3. Infraestrutura: IP do atacante desconhecido e não pode ser rastreado;

4. Adversário: Uso de derivação de rede em rede privada;

A lista a seguir mostra as principais metas características do ataque. Fases:

1. Realização de varredura de porta;

2. Ataque de dicionário.

Metodologia:

1. Escuta de pacotes para descobrir o provável endereço da sub-rede;

2. Auto assinar endereços IP até que seja descoberto um endereço válido na sub-rede;

3. Escanear todos os IPs da sub-rede para coletar informações de vulnerabilidades.

Resultados obtidos:

Foram capturados pacotes que expuseram o endereço Internet Protocol (IP) 10.136.43.20, revelando que provavelmente a sub-rede 10.136.43.255 seria válida. Assumiu-se que a máscara poderia ser 255.255.0.0 ou 255.255.255.0

Ao designar a configuração da interface eth0 "inet 10.136.43.111 netmask 255.255.0.0 ", foi possível executar o *ping* no *switch* "10.136.50.1". Em seguida, foi realizada uma varredura com o software nmap de toda a sub-rede e foram encontrados diversos endereços conectados a ela. Dentre estes, foi descoberto que o *switch* possui um servidor web que possibilita a sua configuração. A Figura 45 apresenta um extrato do relatório da varredura de porta no cenário real.

```
Nmap scan report for 10.136.50.1
Host is up (0.018s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE   VERSION
22/tcp  open  ssh       FreSSH 0.8 (protocol 2.0)
23/tcp  open  telnet    Enterasys C2H124-48 switch telnetd
80/tcp  open  http      Embedded Web Server
443/tcp open  ssl/https Embedded Web Server
2 services unrecognized despite returning data. If you know the
service/version, please submit the following fingerprints at
https://nmap.org/cgi-bin/submit.cgi?new-service :
```

Figura 45 – Extrato do Relatório de Varredura de Porta dos Dispositivos no Cenário Real (Fonte: Autor).

Uma vez conhecidas as portas do *switch*, iniciou-se um ataque de dicionário baseado em lista de palavras para quebrar os dados de autenticação do usuário no *switch*. Observa-se que o serviço SSH estava disponível na porta 22 e foi detectado o uso da versão *FreSSH 0.8 (protocol 2.0)*, que possui vulnerabilidades conhecidas. Nesse caso, o ataque de dicionário baseado em lista de palavras foi bem-sucedido. Não houve a necessidade de explorar as vulnerabilidades já conhecidas da versão do *FreSSH*. A Figura 46 apresenta um extrato do ataque de dicionário.

```
"
[*] Attempting to login to http://10.136.50.1:80/
[...]
[+] HTTP - Success: 'admin:chesfadmin'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
"
```

Figura 46 – Extrato do Relatório do Ataque de Dicionário no *switch* (Fonte: Autor).

Observam-se para estes testes que as falhas encontradas foram decorrentes das decisões na fase de implementação e configurações dos sistemas e não devido à concepção do

modelo da ilha de segurança. Para solucionar essas falhas no *switch*, pode-se desabilitar as interfaces web ou mudar as credenciais de forma a impossibilitar ataques de dicionário. Em produção, as credenciais do SSH também devem ser alteradas junto à sua porta de acesso. Em decorrência do *switch* permitir configuração via porta serial,o SSH deve ser desabilitado.

#### *5.3.1.2 Teste de Invasão Física com Acesso a uma Porta com Autenticação Forçada*

O objetivo do teste é obter acesso a uma porta do *switch* da ilha de automação configurada como "Autenticação forçada".

Modelo de ataque:

1. Vítima: O *switch* da rede de automação;
2. Capacidade: Ferramenta de varredura de porta e de ataques de dicionário;
3. Infraestrutura: IP do atacante desconhecido, mas pode ser rastreado;
4. Adversário: Uso de derivação de rede em rede privada;

Metas características do ataque:

1. Fases:
    a) Realização de varredura de porta;
    b) Ataque de dicionário.
2. Metodologia:
    a) Escuta de pacotes para descobrir o provável endereço da sub-rede;
    b) Auto assinar endereços IP até que seja descoberto um endereço válido na sub-rede;
    c) Escanear todos os IPs da sub-rede para coletar informações de vulnerabilidades.

Resultados obtidos:

Observa-se que, sem realizar a autenticação na ilha de segurança, não é possível encontrar o endereço da sub-rede, mesmo executando o escaneamento de toda faixa. Além disso, também não é possível detectar qualquer informação dos equipamentos da rede de automação. Portanto, não é possível detectar qualquer pacote que revele o endereço IP de uma determinada máquina. O único pacote capturado foi do protocolo EAP que revela apenas o fabricante do *switch*, o qual já é conhecido, tendo em vista que o usuário mal-intencionado possui acesso físico ao ambiente da rede de automação. A Figura 47 mostra o pacote EAPOL detectado durante o teste.

```
"
EAP packet captured:
Enterasy_79:e7:a0       Nearest EAP     60      Request,the Identity
"
```

Figura 47 – Resultado do Teste de Invasão Física aos switches (Fonte: Author).

#### *5.3.1.3 Teste de Invasão Física a uma IHM não Autenticada e Fisicamente Conectada a uma Porta de Autenticação Forçada*

O objetivo do teste é realizar uma invasão física a uma máquina com o cliente SGDP instalado, rodando com a tela bloqueada, fisicamente conectado a uma porta de autenticação forçada com o usuário não autenticado. A meta é escapar do bloqueio de tela e tentar obter acesso a recursos da rede. Os resultados obtidos ocorreram com a troca de TTY ( Ctrl + Alt + Fkey), sendo assim, foi possível escalar privilégios para matar o processo *pkill java*, desbloqueando a tela.

Modelo de ataque:

1. Vítima: IHM de um usuário conectado à rede de automação;
2. Capacidade: Ferramenta de varredura de porta e de ataques de dicionário;
3. Infraestrutura: IP da própria IHM podendo ser rastreado;
4. Adversário: Uso de acesso físico a uma IHM bloqueada da rede de automação;

Metas características do ataque:

1. Fases:
   a) Quebra o bloqueio de tela;
   b) Realizar varredura de porta buscando eventuais vulnerabilidades;
   c) Ataque de dicionário.
2. Metodologia:
   a) Utilizar teclas de TTY ( Ctrl + Alt + Fkey) para quebrar o bloqueio de tela;
   b) Matar o processo *pkill java* para interromper o serviço cliente SPCA;
   c) Escanear todos os IPs da sub-rede para coletar informações de vulnerabilidades.

Resultados obtidos:

Como a máquina não está autenticada, obtém-se resultado semelhante ao teste de invasão física com acesso a uma porta com autenticação forçada. Assim sendo, detecta-se apenas a publicação do protocolo EAPOL enviado pelo *switch Enterasys*. Mesmo assim,

obteve-se acesso aos recursos locais da máquina. Para garantir que isso não aconteça, os usuários devem ter o direito de uso aos TTYs explicitamente bloqueados, garantindo a segurança dos recursos locais da máquina e da rede.

#### *5.3.1.4 Teste de Invasão Física a uma IHM Autenticada e Fisicamente Conectada a uma Porta de Autenticação Forçada*

O objetivo do teste é, após uma autenticação bem-sucedida em um papel, obter acesso a serviços não autorizados pertencente a outro papel independente do cartão de autenticação.

Modelo de ataque:

1. Vítima: Serviços da rede de automação;
2. Capacidade: Ferramenta de varredura de porta e de ataques de dicionário;
3. Infraestrutura: IP da própria IHM podendo ser rastreado;
4. Adversário: Uso de acesso físico a uma IHM autenticada em um determinado papel da rede de automação;

Metas características do ataque:

1. Fases:
    a) Quebra o bloqueio de tela;
    b) Realizar varredura de porta buscando eventuais vulnerabilidades;
    c) Ataque de dicionário.
2. Metodologia:
    a) Matar o processo *pkill java* para interromper o serviço cliente SPCA;
    b) Escanear todos os IPs da sub-rede para coletar informações de vulnerabilidades.

Resultados obtidos: O resultado apresentado é que após a máquina ser autenticada, é possível matar o processo java da aplicação sem necessitar da troca de TTY. Com o cliente inativo, se a IHM já estiver autenticada, é viável acessar todos os recursos fornecidos pela rede. Com o cliente inativo, não é possível ser visualizada na IHM no painel de administração do SPCA, portanto, não é possível impedir o acesso à rede. A Figura 48 destaca o exemplo de execução de ping no *switch*. A segurança da aplicação se limita à segurança de cartões e das instalações físicas. Em caso de perda ou roubo, o acesso via cartão deve ser bloqueado o mais rápido possível pelo administrador, embora ainda exista o PIN que limita as possibilidades de invasão. Além disso, a distribuição do software cliente SGDP deve ser restrita e controlada. Todos os casos de sucesso na penetração dependeram de acesso local aos *switches* ou da autenticação prévia do cartão.

```
"
ping 10.136.50.2 -c 100
64 bytes from 10.136.50.2 icmp_seq=1 ttl=64 time=0.739 ms
[...]
--- 10.136.50.2 ping statistics ---
100 packets transmited, 100 packets recived, 0% packet loss, time
        99023ms
rtt min/avg/max/mdev = 0.690/4.098/151.173/19.121 ms
"
```

Figura 48 – Ping no *switch* Após Invasão Física de uma IHM Autenticada (Fonte: Autor).

### 5.3.2 Testes de Desempenho

A análise em cenário real tem o objetivo de verificar o desempenho e a escalabilidade do modelo proposto. No caso de desempenho, o teste calcula os intervalos de tempo gasto em cada fase do processo de autenticação. No caso da escalabilidade, é verificado qual o número de autenticação no qual o módulo de autenticação consegue manter a sua funcionalidade sem à degradação do processo.

#### *5.3.2.1 Teste de Escalabilidade*

A finalidade desse teste é verificar se o desempenho do processo de autenticação é afetado pelo número de repetições de solicitação de acesso ao sistema. A metodologia aplicada foi submeter à implementação do modelo a uma sequência de autenticação de um ou mais usuários, solicitando autenticação a cada 10 segundos.

Para esse teste foi utilizado dois cenários: o primeiro considerando a autenticação de um usuário e o segundo a autenticação de cinco usuários simultaneamente. Esses cenários correspondem um carregamento real de um sistema de automação de um centro de controle de sistemas elétrico de potência. As métricas de tempo obtidas são: leitura do cartão, autenticação EAPOL/RADIUS, aplicação das políticas e o total do processo fim a fim.

No teste de autenticação de um usuário de forma isolada, obteve-se um tempo total de autenticação da ordem de 12 segundos. O maior consumidor desse tempo foi o processo de autenticação EAPOL que levou 6,5 segundos, seguido pelo processo de leitura de cartão com um tempo de 5 segundos. A Figura 49 apresenta os resultados para esse teste.

Para verificar o comportamento da concorrência no processo de autenticação, foi realizado um teste com usuários realizando autenticação simultânea de uma operação a cada dez segundos. Essas autenticações são realizadas de forma assíncrona. Na prática, o grupo de usuários das redes de automação do setor elétrico é formado pelo pessoal de operação de sistemas elétricos e de manutenção dos sistemas de automação.

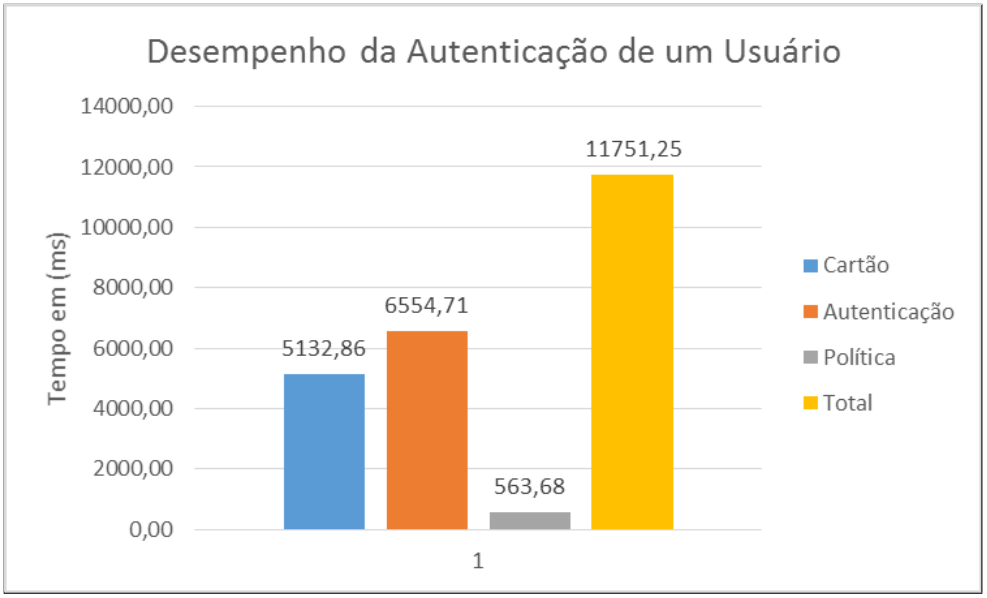


Figura 49 – Resultado do Teste de Desempenho para Autenticação de um Usuário (Fonte: Autor).

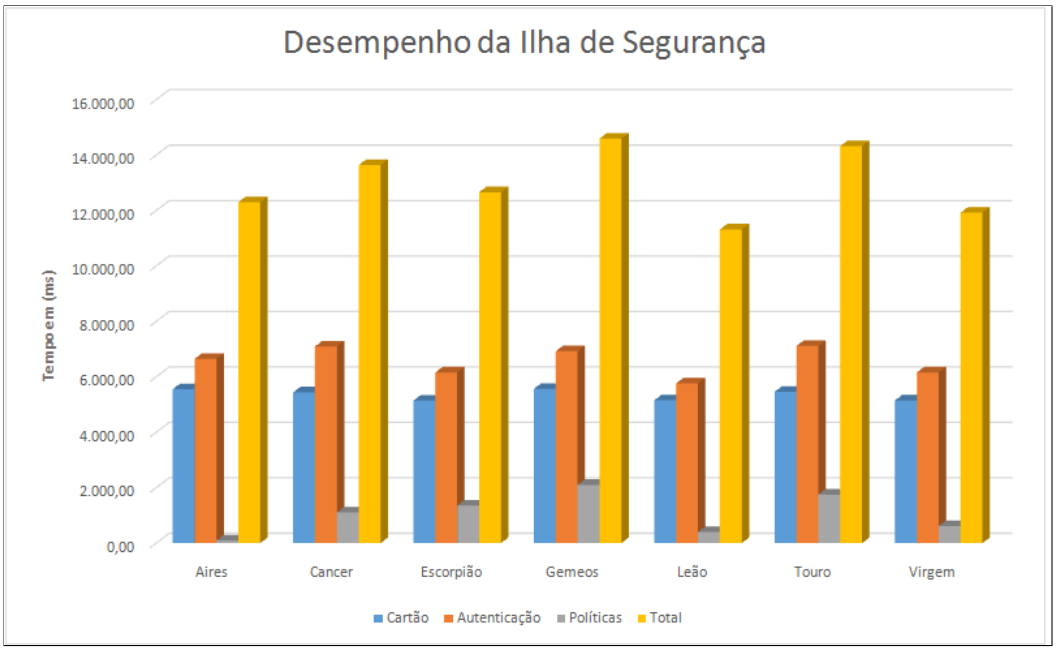


Figura 50 – Resultado do Teste de Desempenho para Autenticação Simultânea de Sete Usuários (Fonte: Autor).

Essas equipes formam um grupo reduzido de pessoas em relação ao número total de empregados que acessam a rede corporativa da organização. O número de sete usuários autenticando, simultaneamente, na frequência estabelecida pelo teste, é suficiente para representar um carregamento real no ambiente de produção. A Figura 50 mostra o desempenho da ilha de segurança para autenticação simultânea de sete usuários.

Tabela 12 – Desempenho do Cenário Real do Centro de Controle (Fonte: Autor).

| | Leitura cartão (ms) | Autenticação (ms) | Aplicação de políticas (ms) | Total (ms) |
|---|---|---|---|---|
| Desvio Padrão | 355,33 | 765,93 | 311,62 | 1.142,01 |
| Nível de confiança | 0,95 | 0,95 | 0,95 | 0,95 |
| Margem de erros | 35,92 | 77,42 | 31,50 | 115,43 |
| Limite Inferior | 5.121,08 | 6.670,08 | 29,50 | 11.879,07 |
| Mediana | 5.157,00 | 6.747,50 | 61,00 | 11.994,50 |
| Limite Superior | 5.192,92 | 6.824,92 | 92,50 | 12.109,93 |

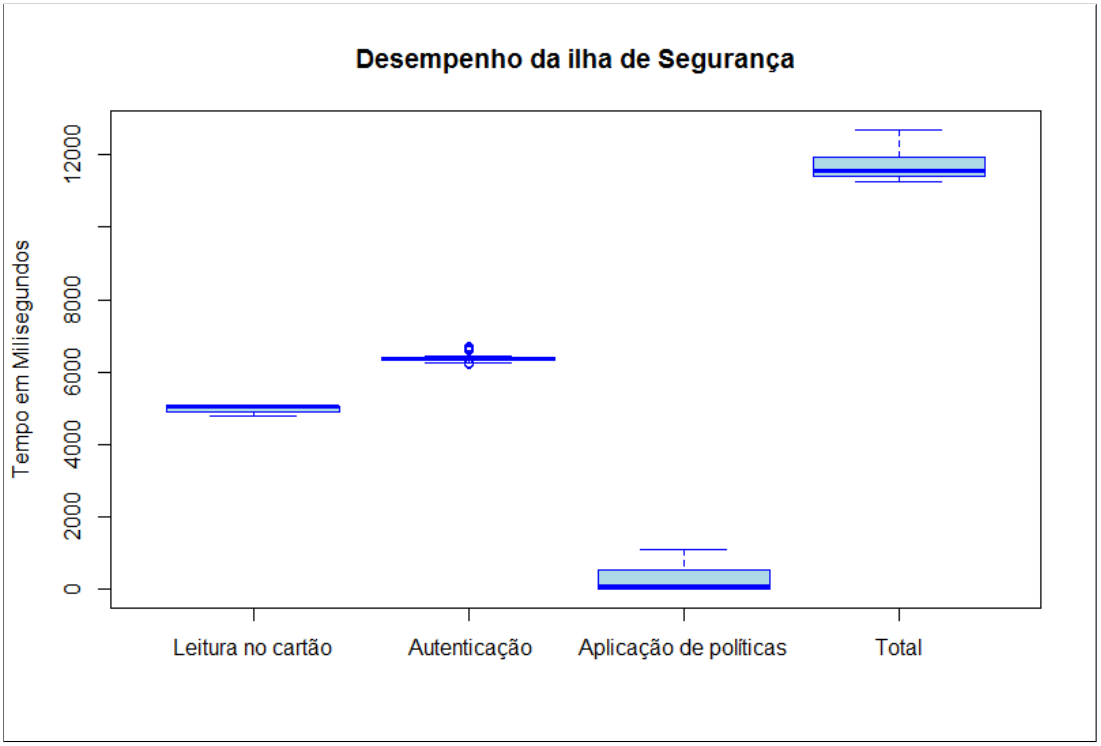


Figura 51 – Resultado do Teste de Desempenho para a Implementação do Modelo (Fonte: Autor).

Os resultados do teste indicam que o tempo de autenticação permaneceu na ordem de 12 segundos e que o comportamento dos tempos intermediários se manteve no mesmo patamar de 6,7 segundo para autenticação e 5,0 segundo para leitura de cartão. O tempo de aplicação de políticas foi de 0,6 segundos. O resumo estatístico dos resultados com um nível de confiança de 95% é apresentado na Tabela 12.

A Figura 51 apresenta os tempos de autenticação da Tabela 12 em forma de *BoxPlot*. Esse tipo de gráfico tem como vantagem apresentar os valores medianos dos resultados acrescidos da informação dos limites de máximo e mínimo e do primeiro e terceiro quartil.

#### *5.3.2.2 Teste de Resposta à Taxa de Autenticação de Usuários*

O objetivo desse teste é realizar a análise do comportamento do modelo com relação à variação da taxa de solicitação de autenticação. A metodologia utilizada foi enviar uma sequência dessas solicitações com o intervalo de tempo cada vez menor entre uma e outra. Os resultados obtidos são apresentados na Figura 52. Pode-se observar que mesmo com a diminuição do intervalo de tempo entre as solicitações de autenticação, os dados obtidos continuam imunes a este tipo de comportamento do processo.

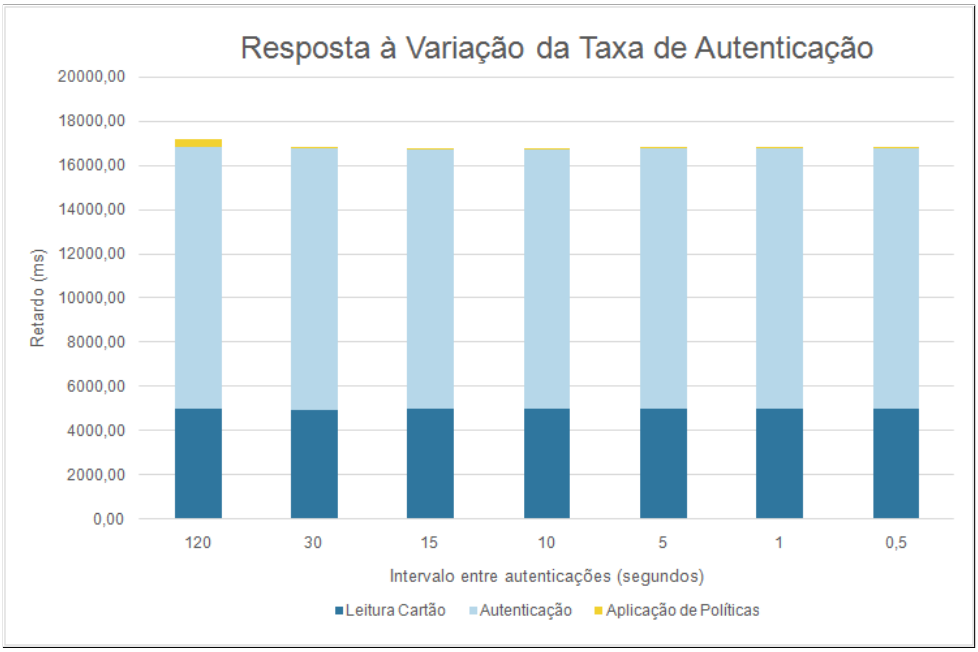


Figura 52 – Resultados dos Testes de Variação da Taxa de Autenticação (Fonte: Autor).

## 5.4 Considerações Finais do Capítulo

Os resultados apresentados mostram que a implementação do modelo em um sistema real é robusta o suficiente para garantir a autenticidade e integridade. Os mecanismos de segurança propostos apresentam um retardo durante o processo de autenticação e autorização. Este retardo é considerado aceitável, tendo em vista que ele somente está presente durante o processo de autenticação e, neste momento, os serviços ainda estão iniciando a sua operação.

Apesar da robustez da proposta, a implementação requer alguns cuidados de configuração como desativar determinados serviços de rede que comprometam a vulnerabilidade da ilha de segurança. Ademais, torna-se uma boa prática verificar se as versões dos softwares têm algum tipo de vulnerabilidade já conhecida e explorada pelo público. Nesse caso, utilizar versões atualizadas.

Os testes realizados com o módulo de teste TTCN-3 mostraram que a especificação do modelo é funcional para os diversos cenários que emularam as condições de operação do modelo. Isso garante que ele é factível e que pode ser implementado utilizando uma tecnologia corrente. Os testes de desempenho da implementação do modelo garantem que o mesmo é robusto e escalável dentro do escopo de um ambiente de automação.

# 6

# CONCLUSÕES

Este capítulo apresenta as considerações finais sobre o cenário da pesquisa, a solução adotada para a hipótese levantada por esta Tese, as principais dificuldades encontradas e os resultados observados. Por fim, alguns trabalhos futuros são delineados e o desafio de pesquisas em aberto para a utilização de redes definida por softwares (SDN) aplicadas à área de segurança cibernética é levantado.

## 6.1 Considerações

Conforme já apresentado nesta Tese, os sistemas de automação e controle dos sistemas elétricos de potência possuem tecnologias com sistemas operacionais conhecidos realizando sua comunicação baseada em redes Ethernet. Este cenário apresenta um aumento de vulnerabilidade desses sistemas, que são explorados por indivíduos mal-intencionados, que podem agir por conta própria ou patrocinados por alguma instituição nacional e internacional.

Várias soluções vêm sendo pesquisadas pelas empresas do setor, pelos institutos de pesquisas e pelas universidades. Essas pesquisas apontam para a abordagem de controle de mensagens que utilizam, principalmente, técnicas para garantir a autenticidade e integridade das informações dos sistemas de automação.

O baixo poder computacional dos dispositivos de automação contribui para que a adoção de mecanismo de segurança, negligenciando os requisitos impostos pela operação em tempo real de um sistema elétrico de potência. Embora alguns padrões de segurança já estejam em vigor, as implementações dos dispositivos com estes recursos, ainda se encontram em estágio incipiente para serem utilizados pelas empresas.

Outras abordagens estão sendo pesquisadas como, por exemplo, a técnica de segurança cibernética focada na análise de tráfego. Esta metodologia apresenta alguns problemas que afetam a decisão de aceitá-la como uma solução definitiva e única para resolver a questão de segurança das redes de automação. Sua eficácia depende dos métodos de detecção de intrusão a serem considerados. No caso das técnicas de detecção apresentadas no Capí-

tulo 3 baseada em redes, observam-se pontos negativos, tais como: perda de pacotes em redes saturadas, dificuldade de entendimento de protocolos de aplicação específica para as redes de automação, ausência de capacidade de monitorar cabeçalhos de protocolos que estão embutidos em *payloads* cifrados e, finalmente, dificuldade de sua aplicação em redes segmentadas. Portanto, vale salientar que esta técnica funciona como esquema complementar para outras abordagens que foram discutidas nesta Tese. Sua utilização poderá ser concebida como uma ferramenta adicional, dentro de um projeto amplo de segurança cibernética.

Mesmo com a disponibilização dos dispositivos com recursos de segurança no mercado, existe o problema de substituição de todo o parque de dispositivos de automação existentes, o que ocasiona uma perda dos investimentos realizados nestes sistemas. Além disso, existe o custo financeiro e o de tempo no processo de implantação. Outros problemas ocasionados pela substituição dos IEDs, são os transtornos provocados no sistema elétrico de potência, tendo em vista as possíveis necessidades de isolamento deste, durante os testes de implantação.

Esta Tese, explorou a hipótese de que é possível estabelecer regiões seguras e confiáveis compostas pelos diversos ativos que compõem o sistema de proteção e automação de um sistema elétrico de potência, sem a necessidade de substituição dos dispositivos eletrônicos inteligentes ou o comprometimento da eficácia do sistema de automação, por perda de desempenho durante a sua operação.

Nesse contexto, foi apresentado pela Tese, o modelo conceitual de uma arquitetura de segurança fundamentada em controle de acesso baseado em papéis. A arquitetura do modelo é denominada de ilha de segurança. Ela tem como premissa ser uma solução de segurança para as redes de automação sem a necessidade de desenvolvimento ou substituição dos IEDs. Além disso, a sua implantação não ocasiona impacto na operação dos sistemas elétricos.

Através de cenários de teste real, o modelo se mostrou bastante robusto para os testes de invasão realizados, o que garantiu a integridade do sistema de automação. Os retardos de 12 segundos resultante da aplicação dos mecanismos de segurança são considerados aceitáveis, pelo fato de os mesmos serem introduzidos apenas durante a fase de inicialização dos dispositivos e processos do sistemas de automação.

Nos ensaios de desempenho, a solução do modelo apresentou bastante vigor em relação à escalabilidade e desempenho, conforme os resultados obtidos no Capítulo 5. A implementação do modelo foi realizada utilizando elementos clássicos de controle de acesso que pode apresentar alguns problemas, dependendo das decisões de configuração tomadas durante a implementação desta arquitetura. Para tornar a solução do modelo imune a essas condições, sugere-se como trabalho futuro realizar pesquisas da implementação do modelo através de redes definidas por softwares. Ademais, este tipo de tecnologia realiza o controle de fluxo através de parâmetro do cabeçalho do pacote e/ou do quadro. Este

fato permite uma melhor granularidade das decisões tomadas pelo mecanismo de segurança. Outra vantagem é um melhor gerenciamento de regras de forma dinâmica, o que contribui para uma flexibilidade mais aprimorada do comportamento global do sistema de segurança.

## 6.2 Dificuldades Encontradas

A respeito das dificuldades encontradas nesta Tese, foram basicamente na implementação dos cenários real e simulado. No primeiro cenário, conforme já citado, a principal dificuldade foi tratar a questão da autenticação do usuário remoto através do padrão 802.1x, uma vez que a conexão da rede de automação com a rede de gestão é realizada por uma única conexão física entre os *switches*. Uma vez que um usuário esteja autenticado, esta porta estará liberada e todas as autenticações terão acesso livre a esta porta. Este problema foi solucionado aplicando uma interface VPN para controlar este acesso, isso implica em um aumento de retardo no processo de autenticação. Este retardo não foi avaliado em decorrência da não implementação desta solução na construção do modelo.

## 6.3 Trabalho Futuros e Desafios Abertos

Como trabalhos futuros, pode-se ainda analisar no cenário da ilha de segurança implementada através de rede convencionais, as questões do desempenho decorrentes das diferentes técnicas de sincronização dos bancos de dados e dos arranjo de redundância do SGDP.

Além disso, pode-se realizar uma implementação do modelo de ilha de segurança utilizando o conceito de redes *Software-Defined Networking (SDN)*. Nesse cenário, seria importante avaliar o desempenho desta solução nas questões de escalabilidade, desempenho. Outro ponto a ser pesquisado, é a avaliação do controle de acesso mediante a mobilidade do usuário entre diferentes domínios em uma rede *Virtual Tenant Network (VTN)*.

Explorando a capacidade de controle de fluxo de pacotes de uma rede definida por software, podemos estabelecer uma arquitetura de segurança de controle de acesso baseada em papéis e controle de fluxo (SUH et al., 2014).

Esta arquitetura é composta por uma infraestrutura de rede formada pelos seguintes componentes: *switches* SDN, um controlador SDN, um servidor de autenticação, um servidor de banco de dados, uma base de usuário e políticas e um serviço de segurança. Este último, roda como uma aplicação do controlador SDN.

No início da etapa de autenticação, a rede SDN está configurada para permitir o fluxo do protocolo de autenticação (SCHMID; SUOMELA, 2013). Nesse estágio, o serviço de segurança identifica o usuário que está sendo autenticado e se o processo foi realizado com sucesso. Neste caso, o serviço de segurança consulta o papel do usuário na base de

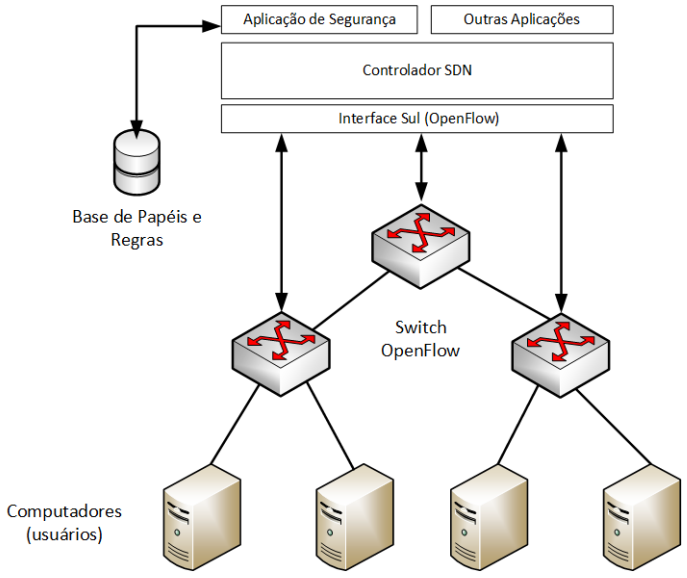


Figura 53 – Arquitetura de Segurança Baseada em Redes SDN (Fonte: Autor).

papéis, e conforme o papel deste, envia uma regra de fluxo para o controlador SDN. Essas regras definem as ações que serão executadas nos pacotes de entrada associado ao tráfego do usuário autenticado. A Figura 53 apresenta os detalhes dessa arquitetura.

As redes virtuais VTN se apresentam como solução para os cenários nos quais o usuário necessita acessar múltiplos domínios (MAMBRETTI; CHEN; YEH, 2016). Nestes, o usuário será credenciado através do servidor de autenticação local. Este servidor deve ter a capacidade de autenticação simultânea nas regiões nas quais o usuário necessita de acesso. Neste caso, a autenticação será realizada em florestas[1] distintas. Em cada domínio, as regras de fluxo inicialmente implantadas permitem a passagem do protocolo de autenticação.

Da mesma forma que em um ambiente de SDN de um único domínio, o serviço de segurança identifica o usuário e verifica se a sua autenticação obteve sucesso. Dessa forma, o serviço de segurança consulta a base de dados de papéis para definir as regras de fluxo. O serviço de segurança envia comandos para o coordenador da SDN determinando as ações a serem realizadas. O coordenador da rede SDN, por sua vez, envia estes comandos, via o gerente de SDN, para os controladores das redes SDN na qual o usuário necessita acessar os recursos. A Figura 54 detalha a arquitetura desse cenário.

[1] A estrutura das redes SDN tem o formato de árvore, um conjunto de árvore corresponde a uma floresta. No caso, especifico de ilha de segurança forma um arquipélago.

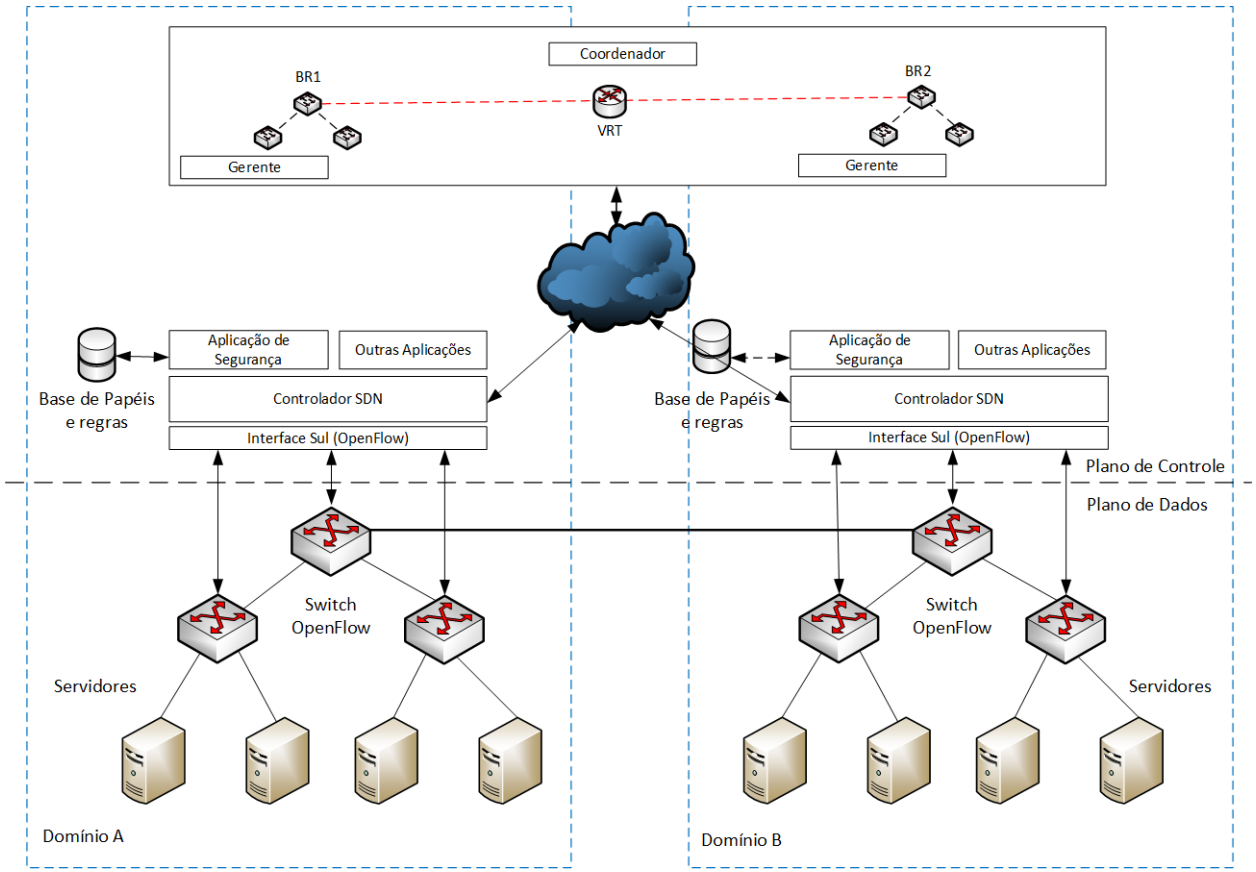


Figura 54 – Arquitetura de Segurança Baseada em Redes VTN (Fonte: Autor).

# REFERÊNCIAS

ABDELKRIM, I. B.; BAINA, A.; BELLAFKIH, M. Automation of access control negotiation in dynamic coalitions for electrical critical infrastructures. In: IEEE. *Electrical and Information Technologies (ICEIT), 2016 International Conference on*. [S.l.], 2016. p. 349–354.

BROWN, T. Security in scada systems: How to handle the growing menace to process automation. *Computing & Control Engineering Journal*, IET, v. 16, n. 3, p. 42–47, 2005.

BURMESTER, M. A trusted computing architecture for critical infrastructure protection. In: IEEE. *Information, Intelligence, Systems and Applications (IISA), 2013 Fourth International Conference on*. [S.l.], 2013. p. 1–6.

CALTAGIRONE, S.; PENDERGAST, A.; BETZ, C. *The diamond model of intrusion analysis*. [S.l.], 2013.

CHANDRA, P.; MESSIER, M.; VIEGA, J. Network security with openssl. *O'Reily, June*, 2002.

DERYNCK, R. Securing critical industrial networks. *Verano white paper*, 2004.

DOULIGERIS, C.; MITROKOTSA, A. Ddos attacks and defense mechanisms: classification and state-of-the-art. *Computer Networks*, v. 44, n. 5, p. 643–666, 2004.

ETSI. *Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language*. Sophia-Antipolis, France, 2016.

FALK, R.; FRIES, S. Efficient multicast authentication in energy automation environments. In: IARIA. *ENERGY 2013; The Third International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies*. [S.l.], 2013. p. 65–71.

FANGFANG, W.; HUAZHONG, W.; DONGQING, C.; YONG, P. Substation communication security research based on hybrid encryption of des and rsa. In: IEEE. *Intelligent Information Hiding and Multimedia Signal Processing, 2013 Ninth International Conference on*. [S.l.], 2013. p. 437–441.

FEITOSA, E.; OLIVEIRA, L.; LINS, B.; JUNIOR, A.; MELO, R.; SADOK, D.; CARMO, U. Security information architecture for automation and control networks. In: *8th Brazilian Symposium of Information Security and Computer Systems, Rio Grande do Sul, SBC, Brazil*. [S.l.: s.n.], 2008. p. 17–30.

FERRAIOLO, D. F.; KUHN, D. R. Role-based access controls. *arXiv preprint arXiv:0903.2171*, 2009.

FREERADIUS. *The FreeRADIUS Project*. 2016. Disponível em: <<http://freeradius.org/>>. Acesso em: 5 fev. 2017.

FSF. *GNU Lesser General Public License, version 2.1*. 1999. Disponível em: <<http://www.gnu.org/licenses/lgpl-2.1.html>>. Acesso em: 2 fev. 2017.

FULORIA, S.; ANDERSON, R. Towards a security architecture for substations. In: IEEE. *Innovative Smart Grid Technologies (ISGT Europe), 2011 2nd IEEE PES International Conference and Exhibition on*. [S.l.], 2011. p. 1–6.

GRABOWSKI, J.; HOGREFE, D.; RÉTHY, G.; SCHIEFERDECKER, I.; WILES, A.; WILLCOCK, C. An introduction to the testing and test control notation (ttcn-3). *Computer Networks*, Elsevier, v. 42, n. 3, p. 375–403, 2003.

HOHLBAUM, F.; BRAENDLE, M.; ALVAREZ, F. Cyber security practical considerations for implementing iec 62351. In: *Proceedings of the PAC World Conference*. [S.l.: s.n.], 2010.

HONG, J.; LIU, C.-C.; GOVINDARASU, M. Detection of cyber intrusions using network-based multicast messages for substation automation. In: IEEE. *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES*. [S.l.], 2014. p. 1–5.

HONG, S.; SHIN, D.-Y.; LEE, M. Evaluating security algorithms in the substation communication architecture. In: IEEE. *Scalable Computing and Communications; Eighth International Conference on Embedded Computing, 2009. SCALCOM-EMBEDDEDCOM '09. International Conference on*. [S.l.], 2009. p. 314–318.

IEC. *Communication networks and systems in substations - Part 5: Communication requirements for functions and device models*. Geneva, Switzerland, 2003.

IEEE. *802.1X - Port Based Network Access Control*. 2006. Disponível em: <<http://www.ieee802.org/1/pages/802.1x.html>>. Acesso em: 5 fev. 2017.

IEEE. *802.1Q - Virtual LANs*. 2014. Disponível em: <<http://www.ieee802.org/1/pages/802.1Q.html>>. Acesso em: 10 de fevereiro de 2017.

IGURE, V. M.; WILLIAMS, R. D. Taxonomies of attacks and vulnerabilities in computer systems. *IEEE Communications Surveys & Tutorials*, IEEE, v. 10, n. 1, 2008.

ISO. *Identification cards — Integrated circuit cards — Part 3: Cards with contacts — Electrical interface and transmission protocols*. Geneva, Switzerland, 2006.

ITU-T. *Specification and description language (SDL) - Recommendation Z100*. [S.l.], 1992.

JAKUGE. *OpenSC – tools and libraries for smart cards*. 2016. <https://github.com/OpenSC/OpenSC/wiki>. Acesso em: 2 fev. 2017.

LABORATORIES, R. *PKCS #15 v1.1: Cryptographic Token Information Syntax*. 2000. <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-15/pkcs-15v1_1.pdf>. Acesso em: 2 fev. 2017.

LEÃO, R. *GTD–Geração, transmissão e distribuição de energia elétrica*. [S.l.], 2009.

MACKAY, S. *Practical industrial data networks: design, installation and troubleshooting*. [S.l.]: Newnes, 2004.

MAMBRETTI, J.; CHEN, J.; YEH, F. Next generation virtual network architecture for multi-tenant distributed clouds: challenges and emerging techniques. In: ACM. *Proceedings of the 4th Workshop on Distributed Cloud Computing*. [S.l.], 2016. p. 1.

MEL, H.; BAKER, D. M. *Cryptography decrypted*. [S.l.]: Addison-Wesley, 2001.

MIRKOVIC, J. *D-WARD: source-end defense against distributed denial-of-service attacks*. Tese (Doutorado) — University of California Los Angeles, 2003.

MOMOH, J. A. *Electric power system applications of optimization*. [S.l.]: CRC Press, 2008.

MOREIRA, N.; MOLINA, E.; LÁZARO, J.; JACOB, E.; ASTARLOA, A. Cyber-security in substation automation systems. *Renewable and Sustainable Energy Reviews*, Elsevier, v. 54, p. 1552–1562, 2016.

NEUMANN, P. Communication in industrial automation—what is going on? *Control Engineering Practice*, Elsevier, v. 15, n. 11, p. 1332–1347, 2007.

OPENSC. *PKCS #11 v2.30: Cryptographic Token Interface Standard*. 2009. <https://github.com/OpenSC/OpenSC/wiki>. Acesso em: 2 fev. 2017.

PCSC. *"Gemalto"; Interoperability Specification for ICCs and Personal Computer Systems*. 2005. Disponível em: <<http://pcscworkgroup.com/Download/Specifications/pcsc1_v2.01.01.pdf>>. Acesso em: 10 fev. 2017.

PCSCWGP. *Interoperability Specification for ICCs and Personal Computer Systems Part 1. Introduction and Architecture Overview*. [S.l.], 2005.

PCSCWGP. *Interoperability Specification for ICCs and Personal Computer Systems Part 3. Requirements for PC-Connected Interface Devices*. [S.l.], 2007.

RAMOS, J. E. A. *Um Estudo Sobre Especificação de Políticas de Redes de Computadores Utilizando Um Modelo Genérico*. 2014. Monografia (Bacharel em Informática), UFPE (Universidade Federal de Pernambuco), Recife, Brazil.

RANKL, W.; EFFING, W. *Smart card handbook*. [S.l.]: John Wiley & Sons, 2010.

ROTONDI, D.; PICCIONE, S. Managing access control for things: A capability based approach. In: ICST (INSTITUTE FOR COMPUTER SCIENCES, SOCIAL-INFORMATICS AND TELECOMMUNICATIONS ENGINEERING). *Proceedings of the 7th International Conference on Body Area Networks*. [S.l.], 2012. p. 263–268.

SAI. *Attacking the DNS Protocol – Security Paper v2*. [S.l.], 2003.

SCHMID, S.; SUOMELA, J. Exploiting locality in distributed sdn control. In: ACM. *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. [S.l.], 2013. p. 121–126.

SHIREY, R. Rfc 2828 - internet security glossary. *The Internet Society*, p. 13, 2000.

STALLINGS, W. *Cryptography and network security: principles and practices*. [S.l.]: Pearson Education India, 2006.

STOUFFER, K.; FALCO, J.; SCARFONE, K. Guide to industrial control systems (ics) security. *NIST special publication*, v. 800, n. 82, p. 16–16, 2011.

SUH, M.; PARK, S. H.; LEE, B.; YANG, S. Building firewall over the software-defined network controller. In: IEEE. *Advanced Communication Technology (ICACT), 2014 16th International Conference on*. [S.l.], 2014. p. 744–748.

TCG. *Trusted Computing Group*. 2005. Disponível em: <<https://trustedcomputinggroup.org/trusted-computing/>>. Acesso em: 5 fev. 2017.

THION, R. Access control models. *Cyber Warfare and Cyber Terrorism*, p. 318–326, 2008.

VACCA, J. R. *Public key infrastructure: building trusted applications and Web services*. [S.l.]: CRC Press, 2004.

VAIDYA, B.; MAKRAKIS, D.; MOUFTAH, H. T. Authentication and authorization mechanisms for substation automation in smart grid network. *Network, IEEE*, IEEE, v. 27, n. 1, p. 5–11, 2013.

VERMA, D. C. *Policy-based networking: architecture and algorithms*. [S.l.]: New Riders Publishing, 2000.

VERMA, D. C. Simplifying network administration using policy-based management. *Network, IEEE*, IEEE, v. 16, n. 2, p. 20–26, 2002.

WEI, D.; LU, Y.; JAFARI, M.; SKARE, P. M.; ROHDE, K. Protecting smart grid automation systems against cyberattacks. *IEEE Transactions on Smart Grid*, IEEE, v. 2, n. 4, p. 782–795, 2011.

YOUNG, A.; KIRSTEIN, P.; IBBETSON, A. *Technologies to support authentication in Higher Education*. 1996. Disponível em: <<http://www.ukoln.ac.uk/services/elib/papers/other/scoping/>>. Acesso em: 16 fev. 2017.

ZKA. *HBCI - HomeBanking Computer Interface*. 2003. Disponível em: <<http://openhbci.sourceforge.net/index.html>>. Acesso em: 2 fev. 2017.

# Testes de validação da especificação SDL da Ilha de Segurança

Nestes apêndice apresentamos os arquivo de configuração utilizados para configurar o módulo de teste TTCN-3 utilizados para validar a funcionalidade da especificação SDL do modelo da ilha de segurança. Na seção seguinte é mostrada uma sequência de mensagens emitidas e recebidas pelo módulo de teste durante os testes de autenticação de um usuário.

## Arquivos de Configuração do Modulo de Teste TTCN-3

1. Arquivo de declarações

```
// Arquivo TTCN_Declarations.ttcn3
module TTCN_Declarations {
//  New types declarations
  type record TTCN_LogInfoType {
    integer logTime,
    charstring logMessage
  }
// Records declaration
  type enumerated cardWithdraw { e_cardWithdraw }
  type record displayMessage {
    charstring param1
    }
  type record key {
    charstring param1
    }
  type enumerated inicio { e_inicio }
```

```
    type record entry {
      charstring param1
      }
    type record CardInsert {
      integer param1
      }
    type record displayLog {
      TTCN_LogInfoType param1
      }
  // Ports declaration
    type port port_HSCS message
      {
      out CardInsert;
      out cardWithdraw;
      out inicio;
      out entry;
      out key;
      in displayMessage;
      in displayLog;
      }
  // TSI and MTC component declaration
    type component runsOn_ilha {
      port port_HSCS HSCS;
    }
    type component ilha {
      port port_HSCS HSCS;
    }
  }
```

2. Templates

```
  // Arquivo TTCN_Templates2.ttcn3
  module TTCN_Templates2 {
    import from TTCN_Declarations all;

    template displayMessage displayMessage_UserAuthMSG12 := {
      param1 := "*"
    };
    template displayMessage displayMessage_UserAuthMSG16 := {
      param1 := "**"
```

```
  };
  template displayMessage displayMessage_UserAuthMSG2 := {
    param1 := "Enter with Card"
  };
  template displayMessage displayMessage_UserAuthMSG30 := {
    param1 := "Alice papel operador autorizado na vlan1"
  };
  template displayMessage displayMessage_UserAuthMSG8 := {
    param1 := "Enter PIN "
  };
  template CardInsert CardInsert_UserAuthMSG3 := {
    param1 := 110
  };
  template displayLog displayLog_UserAuthMSG28 := {
    param1 := {       logTime := 2,
      logMessage := "LOG: Alice papel operador autorizado
      na vlan1" }
  };
  template key key_UserAuthMSG13 := {
    param1 := "2"
  };
  template key key_UserAuthMSG17 := {
    param1 := "3"
  };
  template key key_UserAuthMSG9 := {
    param1 := "1"
  };
  template inicio inicio_UserAuthMSG1 := ?;
}
```

3. Teste e Controle de Teste

```
// Arquivo TTCN_TesteAndControl2.ttcn3
module TTCN_TestsAndControl2
  {
  import from TTCN_Declarations all;
  import from TTCN_Templates2 all;

  altstep RTDS_fail() runs on runsOn_ilha
    {
```

```
    []HSCS.receive
      {
      setverdict(fail, "Fail in default altstep!");
      stop;
      };
    }

  testcase TC_UserAuth() runs on runsOn_ilha system ilha
    {
    activate(RTDS_fail());
    map(self:HSCS, system:HSCS);
    HSCS.send(inicio_UserAuthMSG1);
    HSCS.receive(displayMessage_UserAuthMSG2);
    HSCS.send(CardInsert_UserAuthMSG3);
    HSCS.receive(displayMessage_UserAuthMSG8);
    HSCS.send(key_UserAuthMSG9);
    HSCS.receive(displayMessage_UserAuthMSG12);
    HSCS.send(key_UserAuthMSG13);
    HSCS.receive(displayMessage_UserAuthMSG16);
    HSCS.send(key_UserAuthMSG17);
    HSCS.receive(displayLog_UserAuthMSG28);
    HSCS.receive(displayMessage_UserAuthMSG30);
    setverdict(pass);
    }

 control
    {
    execute(TC_UserAuth());
    }
 }
```

4. Controle

```
// Arquivo TTCN_CControler2.ttcn3
module TTCN_CControlPart {
  import from TTCN_TestsAndControl8 all;
  import from TTCN_TestsAndControl7 all;
  import from TTCN_TestsAndControl6 all;
  import from TTCN_TestsAndControl5 all;
```

```
import from TTCN_TestsAndControl4 all;
import from TTCN_TestsAndControl3 all;
import from TTCN_TestsAndControl2 all;

import from TTCN_TestsAndControl1 all;
import from TTCN_TestsAndControl all;

control
  {
  execute(TC_AdminAccess());
  execute(TC_RegUser());
  execute(TC_UserAuth());
  execute(TC_AuthUserNotReg());
  execute(TC_AuthUserAuth());
  execute(TC_CardWthUserNotReg());
  execute(TC_DelUserNotReg());
  execute(TC_CardWthUseReg());
  execute(TC_DelUserReg());
  }
}
```

# Mensagens Emitidas pelo Módulo de Teste TTCN-3

A listagem a seguir, apresenta a sequência de mensagens emitidas e recebidas pelo módulo de teste TTCN-3, durante a validação da ilha de segurança, para o teste de autenticação de um usuário. Esta sequência finaliza com o veredicto do teste.

```
>Signal: inicio sent by: TTCN_CControlPart(1) at: 0  ticks
>Signal: inicio received by: InteHwSeg(5) at: 0  ticks
>Signal: displayMessage sent by: InteHwSeg(5) at: 0  ticks
>{param1=Enter with Card}
>Task InteHwSeg(5) has changed to state idle at: 0  ticks
>Signal: displayMessage received by: TTCN_CControlPart(1) at:
0  ticks
>{param1=Enter with Card}
>Signal: CardInsert sent by: TTCN_CControlPart(1) at: 0  ticks
>{param1=110}
>Signal: CardInsert received by: InteHwSeg(5) at: 0  ticks
>{param1=110}
```

```
>Timer: cardTimer started by: InteHwSeg(5) at: 0  ticks
>Signal: checkAuth sent by: InteHwSeg(5) at: 0  ticks
>{param1=110}
>Task InteHwSeg(5) has changed to state idle at: 0  ticks
>Signal: checkAuth received by: PAE(6) at: 0  ticks
>{param1=110}
>Signal: checkAuth sent by: PAE(6) at: 0  ticks
>{param1=110}
>Task PAE(6) has changed to state waitCheckAuth at: 0
ticks
>Signal: checkAuth received by: PepRede(2) at: 0  ticks
>{param1=110}
>Signal: startAuth sent by: PepRede(2) at: 0  ticks
>{param1=110}
>Task PepRede(2) has changed to state idle at: 0  ticks
>Signal: startAuth received by: PAE(6) at: 0  ticks
>{param1=110}
>Signal: displayMessage sent by: PAE(6) at: 0  ticks
>{param1=Enter PIN }
>Timer: codTimer started by: PAE(6) at: 0  ticks
>Task PAE(6) has changed to state wait4Code at: 0  ticks
>Signal: displayMessage received by: InteHwSeg(5) at: 0
ticks
>{param1=Enter PIN }
>Signal: displayMessage sent by: InteHwSeg(5) at: 0  ticks
>{param1=Enter PIN }
>Task InteHwSeg(5) has changed to state idle at: 0  ticks
>Signal: displayMessage received by: TTCN_CControlPart(1)
at: 0  ticks
>{param1=Enter PIN }
>Signal: key sent by: TTCN_CControlPart(1) at: 0  ticks
>{param1=1}
>Signal: key received by: InteHwSeg(5) at: 0  ticks
>{param1=1}
>Timer: keyTimer started by: InteHwSeg(5) at: 0  ticks
>Signal: key sent by: InteHwSeg(5) at: 0  ticks
>{param1=1}
>Task InteHwSeg(5) has changed to state idle at: 0  ticks
>Signal: key received by: PAE(6) at: 0  ticks
```

```
>{param1=1}
>Signal: displayMessage sent by: PAE(6) at: 0  ticks
>{param1=*}
>Task PAE(6) has changed to state wait4Code at: 0  ticks
>Signal: displayMessage received by: InteHwSeg(5) at: 0  ticks
>{param1=*}
>Signal: displayMessage sent by: InteHwSeg(5) at: 0  ticks
>{param1=*}
>Task InteHwSeg(5) has changed to state idle at: 0  ticks
>Signal: displayMessage received by: TTCN_CControlPart(1)
at: 0  ticks
>{param1=*}
>Signal: key sent by: TTCN_CControlPart(1) at: 0  ticks
>{param1=2}
>Signal: key received by: InteHwSeg(5) at: 0  ticks
>{param1=2}
>Timer: keyTimer started by: InteHwSeg(5) at: 0  ticks
>Signal: key sent by: InteHwSeg(5) at: 0  ticks
>{param1=2}
>Task InteHwSeg(5) has changed to state idle at: 0  ticks
>Signal: key received by: PAE(6) at: 0  ticks
>{param1=2}
>Signal: displayMessage sent by: PAE(6) at: 0  ticks
>{param1=**}
>Task PAE(6) has changed to state wait4Code at: 0  ticks
>Signal: displayMessage received by: InteHwSeg(5) at: 0  ticks
>{param1=**}
>Signal: displayMessage sent by: InteHwSeg(5) at: 0  ticks
>{param1=**}
>Task InteHwSeg(5) has changed to state idle at: 0  ticks
>Signal: displayMessage received by: TTCN_CControlPart(1)
at: 0  ticks
>{param1=**}
>Signal: key sent by: TTCN_CControlPart(1) at: 0  ticks
>{param1=3}
>Signal: key received by: InteHwSeg(5) at: 0  ticks
>{param1=3}
>Timer: keyTimer started by: InteHwSeg(5) at: 0  ticks
>Signal: key sent by: InteHwSeg(5) at: 0  ticks
```

```
>{param1=3}
>Task InteHwSeg(5) has changed to state idle at: 0  ticks
>Signal: key received by: PAE(6) at: 0  ticks
>{param1=3}
>Signal: AuthReq sent by: PAE(6) at: 0  ticks
>{param1={certID=110,pin=123}}
>Task PAE(6) has changed to state idle at: 0  ticks
>Signal: AuthReq received by: ServAuth(4) at: 0  ticks
>{param1={certID=110,pin=123}}
>Signal: Employee sent by: ServAuth(4) at: 0  ticks
>{param1=110}
>Task ServAuth(4) has changed to state idle at: 0  ticks
>Signal: Employee received by: PDP(8) at: 0  ticks
>{param1=110}
>Signal: sGetUserRoleInfo sent by: PDP(8) at: 0  ticks
>{param1=110}
>Task PDP(8) has changed to state waitUserRoleInfo at:
0  ticks
>Signal: sGetUserRoleInfo received by: pPMT(9) at: 0  ticks
>{param1=110}
>Signal: sGetUserRoleResp sent by: pPMT(9) at: 0  ticks
>{param1={certID=110,role=operador}}
>Task pPMT(9) has changed to state idle at: 0  ticks
>Signal: sGetUserRoleResp received by: PDP(8) at: 0  ticks
>{param1={certID=110,role=operador}}
>Signal: policyEnforcement sent by: PDP(8) at: 0  ticks
>{param1={certID=110,role=operador,Action=vlan1}}
>Task PDP(8) has changed to state idle at: 0  ticks
>Signal: policyEnforcement received by: PepRede(2) at:
0  ticks
>{param1={certID=110,role=operador,Action=vlan1}}
>Signal: getLogReq sent by: PepRede(2) at: 0  ticks
>{param1=LOG: Alice papel operador autorizado na vlan1}
>Task PepRede(2) has changed to state waitLogReq1 at:
0  ticks
>Signal: getLogReq received by: ServAud(3) at: 0  ticks
>{param1=LOG: Alice papel operador autorizado na vlan1}
>Signal: getLogResp sent by: ServAud(3) at: 0  ticks
>Signal: displayLog sent by: ServAud(3) at: 0  ticks
```

```
>{param1={logTime=2,logMessage=LOG: Alice papel operador
autorizado na vlan1}}
>Task ServAud(3) has changed to state idle at: 0  ticks
>Signal: getLogResp received by: PepRede(2) at: 0  ticks
>Signal: displayMessage sent by: PepRede(2) at: 0  ticks
>{param1=Alice papel operador autorizado na vlan1}
>Task PepRede(2) has changed to state idle at: 0  ticks
>Signal: displayLog received by: InteHwSeg(5) at: 0  ticks
>{param1={logTime=2,logMessage=LOG: Alice papel operador
autorizado na vlan1}}
>Signal: displayLog sent by: InteHwSeg(5) at: 0  ticks
>{param1={logTime=2,logMessage=LOG: Alice papel operador
autorizado na vlan1}}
>Task InteHwSeg(5) has changed to state idle at: 0  ticks
>Signal: displayMessage received by: PAE(6) at: 0  ticks
>{param1=Alice papel operador autorizado na vlan1}
>Timer: tDisplay started by: PAE(6) at: 0  ticks
>Signal: displayMessage sent by: PAE(6) at: 0  ticks
>{param1=Alice papel operador autorizado na vlan1}
>Task PAE(6) has changed to state idle at: 0  ticks
>Signal: displayLog received by: TTCN_CControlPart(1) at:
0  ticks
>{param1={logTime=2,logMessage=LOG: Alice papel operador
autorizado na vlan1}}
>Signal: displayMessage received by: InteHwSeg(5) at:
0  ticks
>{param1=Alice papel operador autorizado na vlan1}
>Signal: displayMessage sent by: InteHwSeg(5) at: 0  ticks
>{param1=Alice papel operador autorizado na vlan1}
>Task InteHwSeg(5) has changed to state idle at: 0  ticks
>Signal: displayMessage received by: TTCN_CControlPart(1)
at: 0  ticks
>{param1=Alice papel operador autorizado na vlan1}
>setverdict(pass)
```

# B

# SDL Aplicado na Ilha de Segurança

Neste anexo será apresentado, como exemplo, o detalhamento dos processos do Bloco AcessoRede. O bloco de acesso à rede é constituído pelos processos (1) ServAuth que especifica a função autenticação do usuário e a emulação da base de dados de usuário, (2) PepeRede que é responsável pelo gerenciamento das ações de aplicação de políticas ao nível de rede e (3) ServAud que especifica a função de auditoria das ações do usuário na rede.

## Processo Aplicação Política de Rede - PepRede

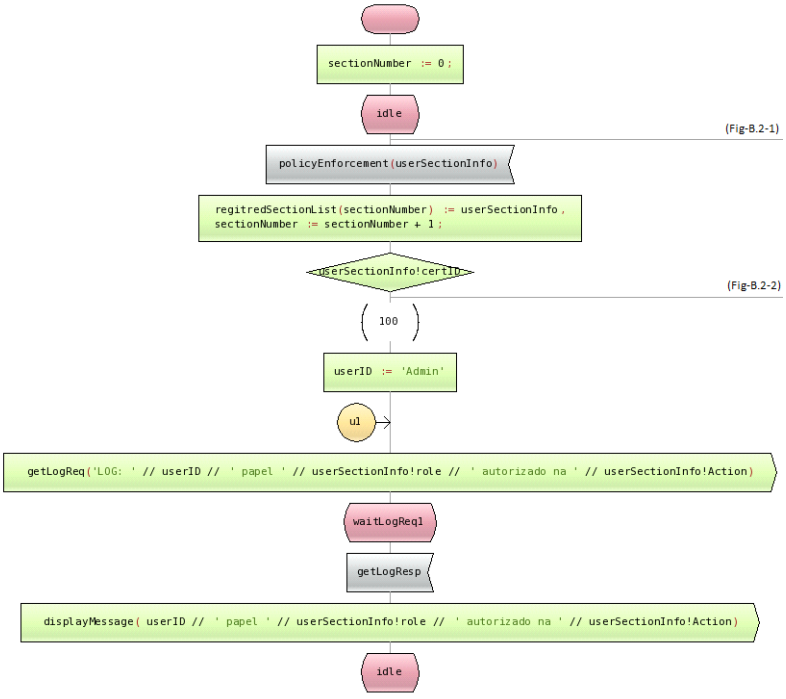


Figura 55 – Procedimento Aplicação de Políticas de Rede (Fonte: Autor).

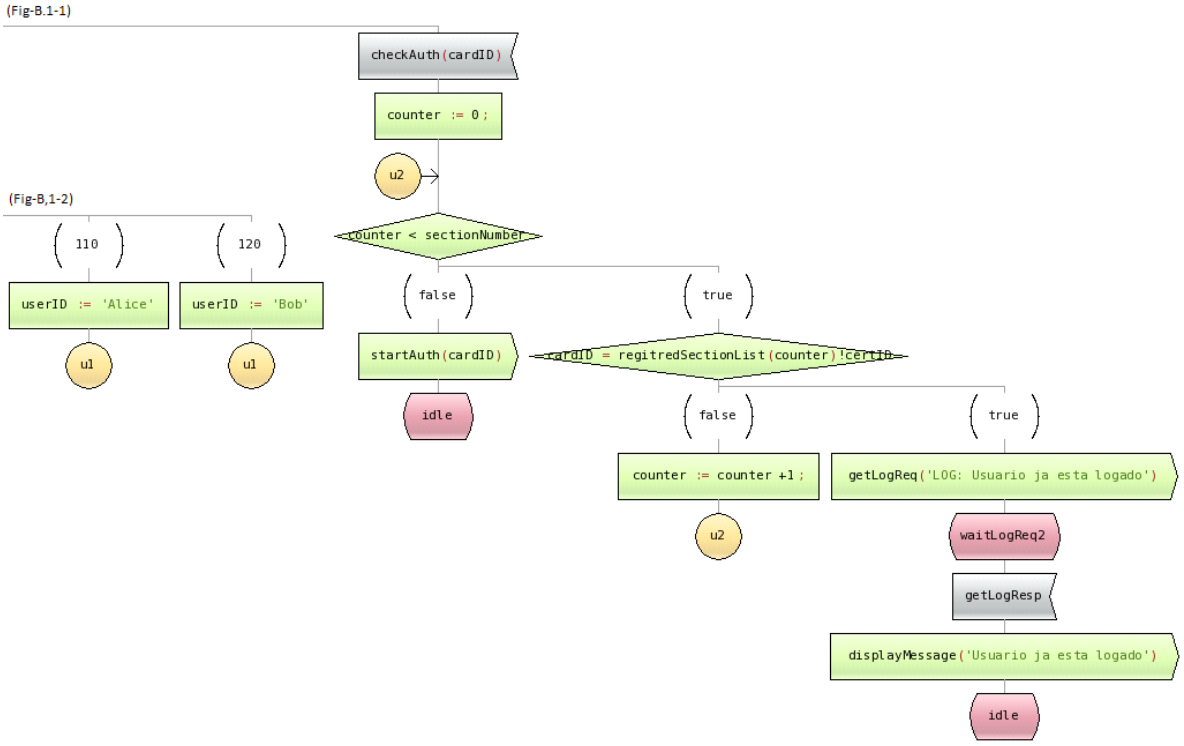


Figura 56 – Procedimento Verificação do Estado de Autenticação (Fonte: Autor).

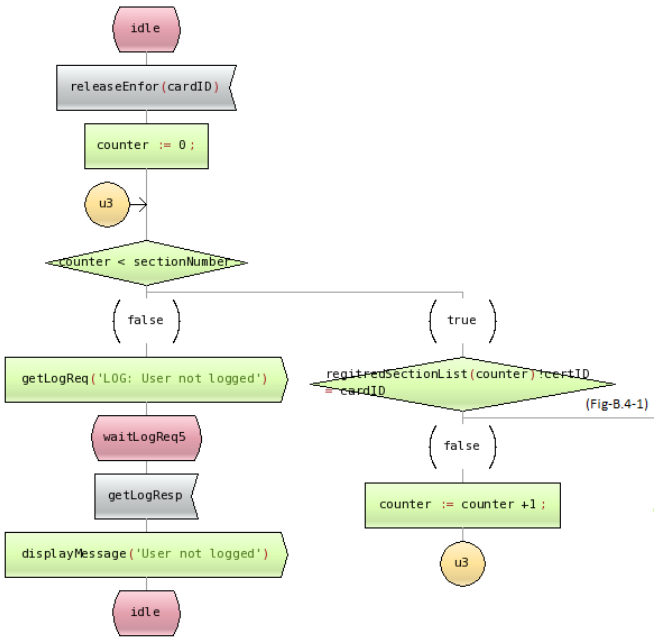


Figura 57 – Procedimento Liberação de Aplicação de Política (a) (Fonte: Autor).

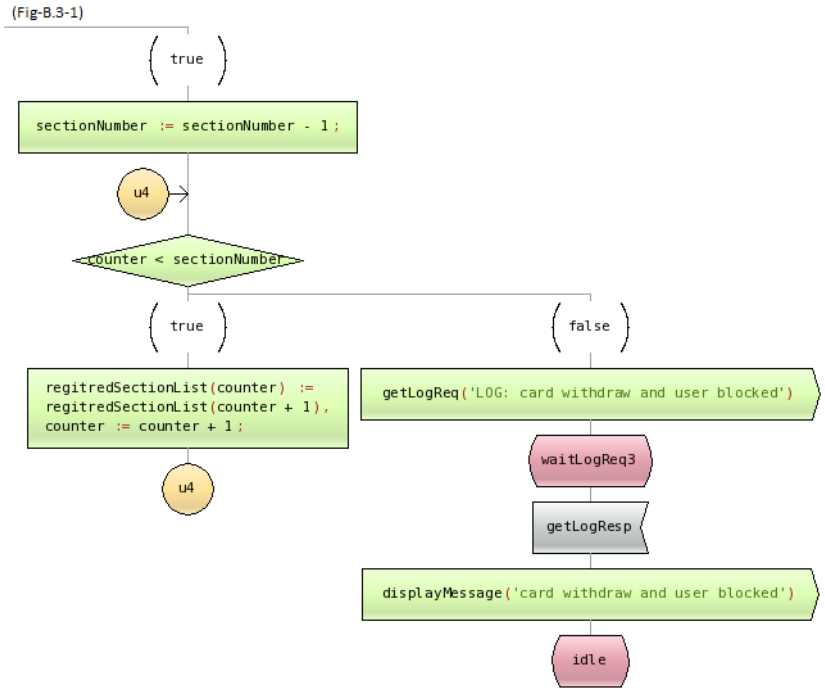


Figura 58 – Procedimento Libera de Aplicação de Política (b) (Fonte: Autor).

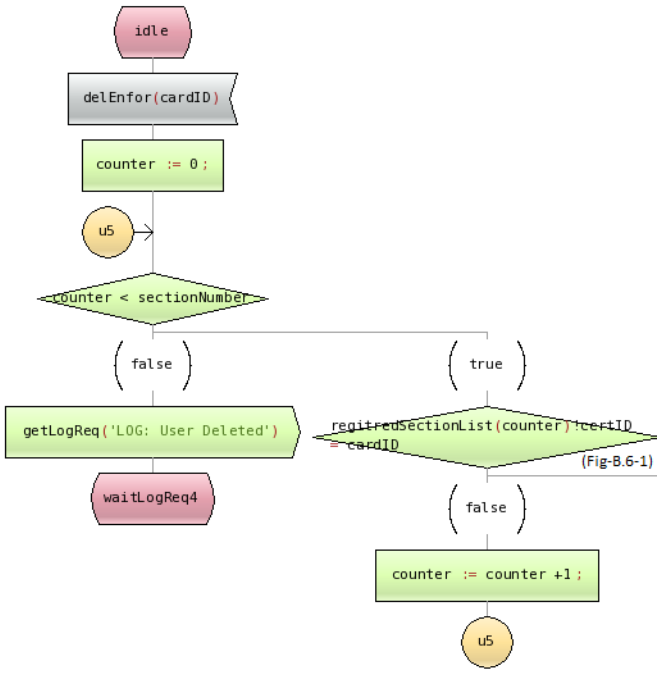


Figura 59 – Procedimento Eliminação de Aplicação de Política (a) (Fonte: Autor).

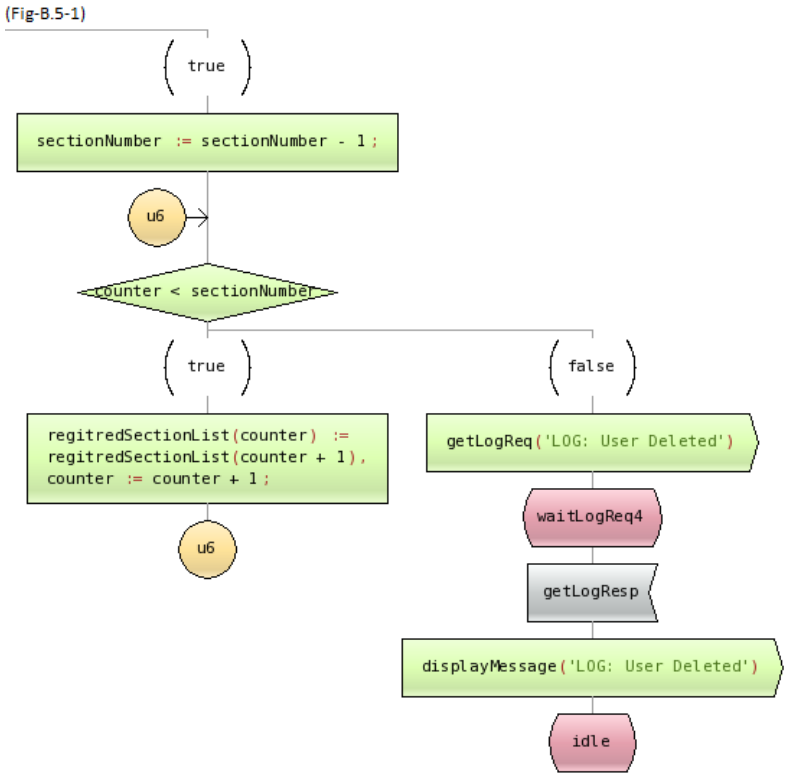


Figura 60 – Procedimento Eliminação de Aplicação de Política (b) (Fonte: Autor).

# Processo Serviço de Auditoria - ServAud

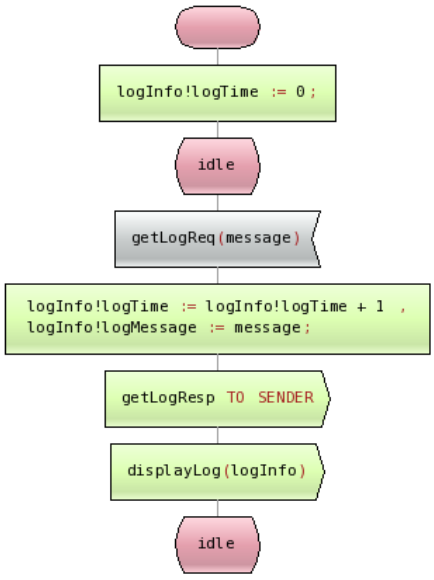


Figura 61 – Procedimento Realização de Registro de Eventos (Fonte: Autor).

# Processo Serviço de Autenticação - ServAuth

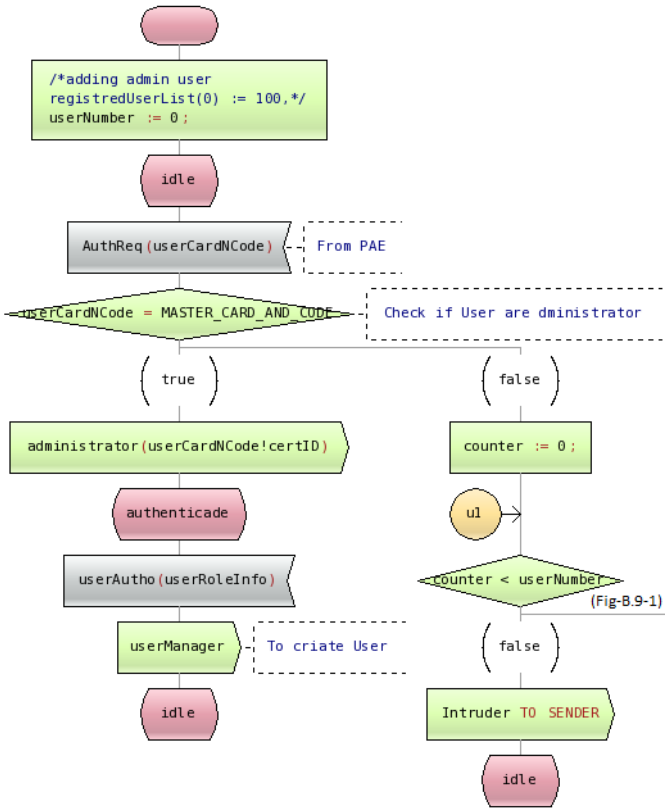


Figura 62 – Procedimento Realização de Solicitação de Autenticação (a) (Fonte: Autor).

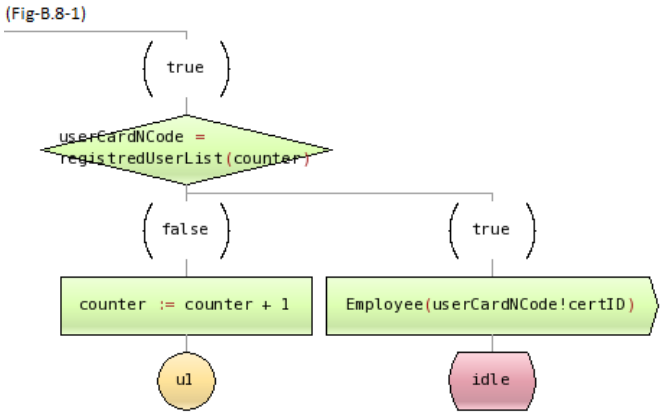


Figura 63 – Procedimento Realização Solicitação de Autenticação (b) (Fonte: Autor).

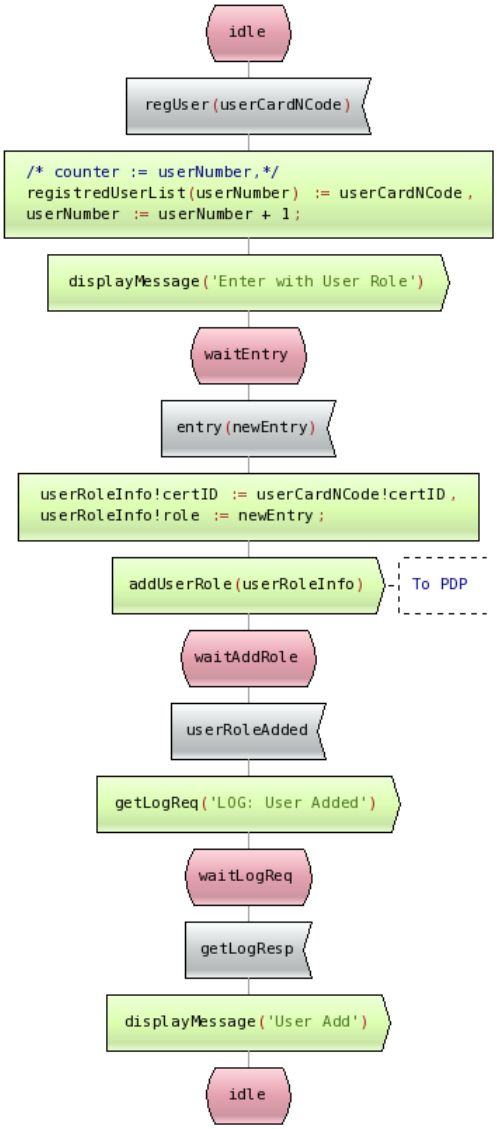


Figura 64 – Procedimento Realização de Registro de Usuário (Fonte: Autor).

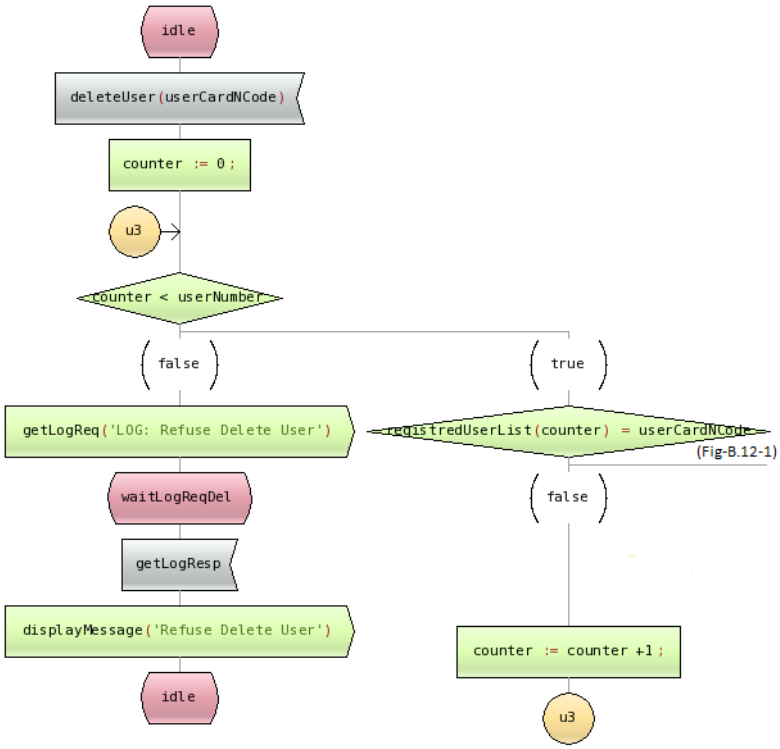


Figura 65 – Procedimento Realização de Eliminação de Usuário (a) (Fonte: Autor).

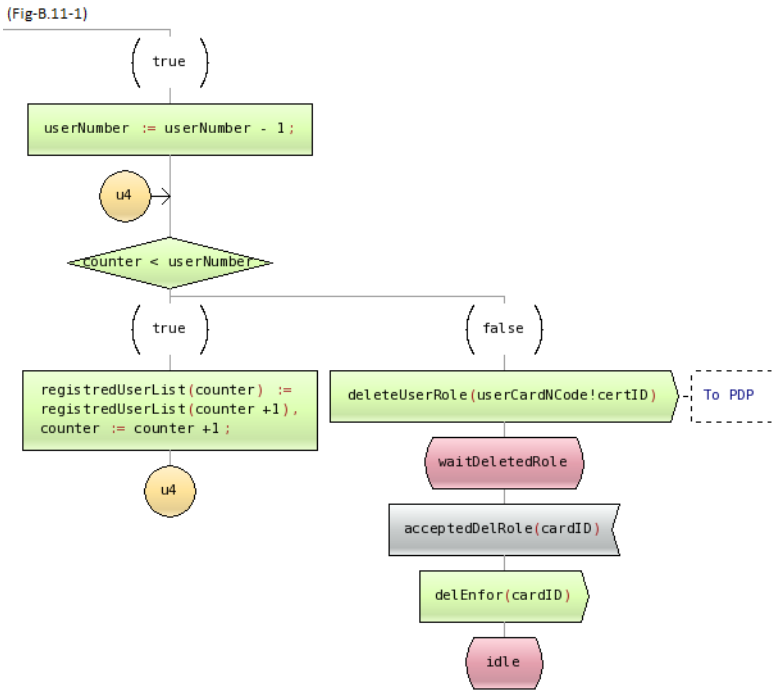


Figura 66 – Procedimento Realização de Eliminação de Usuário (b) (Fonte: Autor).