VILMAR ABREU JUNIOR

GESTÃO DE IDENTIDADE E ACESSO MULTIFATOR PARA SMART GRID

Proposta de tese apresentado ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná, como requisito parcial à obtenção do título de doutor em Informática.

CURITIBA

2019

VILMAR ABREU JUNIOR

GESTÃO DE IDENTIDADE E ACESSO MULTIFATOR PARA SMART GRID

Proposta de tese apresentado ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná, como requisito parcial à obtenção do título de doutor em Informática.

Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. Altair Olivo Santin

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central
Luci Eduarda Wielganczuk – CRB 9/1118

A162g 2019

Abreu Júnior, Vilmar
Gestão de identidade e acesso multifator para smart grid / Vilmar Abreu Junior; orientador: Altair Olivo Santin. – 2019.
xvi, 103 f. : il. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2019
Bibliografia: f. 98-109

1. Informática. 2. Internet – Controle de acesso. 3. Internet das coisas. 4. Mineração de dados. I. Santin, Altair Olivo. II. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. III. Título.

CDD 22. ed. – 004

Pontifícia Universidade Católica do Paraná

PUCPR
GRUPO MARISTA

DECLARAÇÃO

Declaro para os devidos fins que o aluno VILMAR ABREU JUNIOR, defendeu sua tese de doutorado intitulada **"GESTÃO DE IDENTIDADE E ACESSO MULTIFATOR PARA SMART GRID"**, na área de concentração Ciência da Computação, no dia 24 de abril de 2019, no qual foi aprovado.

Declaro ainda que foram feitas todas as alterações solicitadas pela Banca Examinadora, cumprindo todas as normas de formatação definidas pelo Programa.

Por ser verdade, firmo a presente declaração.

Curitiba, 03 de maio de 2019.

Programa de Pós-Graduação em Informática
PPGIa
PUCPR

Prof. Dr. Emerson Cabrera Paraiso
Coordenador do Programa de Pós-Graduação em Informática
Pontifícia Universidade Católica do Paraná

Dedico este trabalho aos meus familiares, a minha esposa e meus amigos que sempre me apoiaram e aconselharam ao longo de minhas conquistas.

Agradecimentos

Agradeço ao meu orientador, Altair Olivo Santin, pelo apoio, não somente relativo à orientação desse trabalho, mas por toda a ajuda nesse longo período de formação acadêmica. Agradeço também aos meus prezados amigos Cleverton Vicentini, Eduardo Viegas, José Eduardo Nunes Lino e Rafael Ribeiro pela amizade e companhia durante essa jornada. Finalmente, agradeço ao apoio dos meus pais Vilmar Abreu e Sueli do Rocio Abreu, da minha esposa Mykaele Fortes Abreu, e das minhas irmãs, Andreia Abreu e Sandra Abreu, por sempre me apoiarem incentivarem.

Resumo

Apesar da literatura propor vários mecanismos de cibersegurança, inclusive com provas matemáticas de sua eficácia, vulnerabilidades que permitem o acesso não autorizado continuam acontecendo em infraestruturas críticas, como *smart grids*. Neste trabalho é proposto um mecanismo de gestão de identidade e acesso multifator (MIAM, *Multi-Factor Identity and Access Management*) que utiliza diversos fatores de autenticação e acesso para garantir a cibersegurança da *smart grid*. O MIAM é compatível com dispositivos da Internet das coisas (IoT, *Internet of Things*), estendendo as funcionalidades da gestão de identidade e acesso para os recursos limitados desses dispositivos e protocolos da IoT. Além disso, o MIAM é protegido por um mecanismo de detecção de intrusão inteligente. As principais contribuições do MIAM são: (i) mecanismo de autenticação multifator, que requer uma credencial de proximidade emitida pelo controle de admissão baseado em localização; (ii) mecanismo de controle de acesso multidomínios, que requer o endosso do controle de admissão baseado em criptografia e quórum; (iii) mecanismo que permite um usuário autenticado transportar suas credenciais para o contexto da IoT, mantendo a autenticidade e confidencialidade na comunicação; (iv) mecanismo de detecção de intrusão treinado em um conjunto de dados de *pentest*, que visa identificar as vulnerabilidades da proposta baseada no modelo de adversário. O protótipo foi implementado usando padrões amplamente adotados na literatura. A avaliação de segurança identificou vulnerabilidades nas tecnologias utilizadas no protótipo que foram classificadas de acordo com sua severidade. O mecanismo de detecção de intrusão tem acurácia de 99%.

Palavras-chave: *Smart grid; Gestão de Identidade e Acesso; Controle de acesso multidomínios; Autenticação Multifator; Detecção de Intrusão Inteligente.*

Abstract

The literature works have proposed protection mechanisms, including mathematical proofs of their security. Nonetheless, vulnerabilities allowing unauthorized access continue to happen in critical infrastructures, such as smart grids (SG). In this work, we propose a Multi-Factor Identity and Access Management (MIAM) approach that uses several authentication and access factors to protect a SG. The MIAM is IoT-friendly, extending IAM features to accommodate the constraints of IoT devices and protocols. In addition, the MIAM is protected by an intelligent Intrusion Detection (iID) engine. The proposed MIAM main contributions are: (i) a multi-factor authentication that requires a proximity credential issued by the device-based location admission control, (ii) a multi-domain access control that requires the endorsement issued by the cryptographic and quorum admission controls, (iii) a mechanism that allows a user authenticated in the MIAM to transport her credentials to the IoT context, maintaining authenticity and confidentiality in communication and a (iv) an intelligent intrusion detection mechanism trained from a pentest database, which identifies the proposed vulnerabilities based on the adversary model. The proposal's prototype was implemented with the use of IT standards. The security evaluation identified vulnerabilities in the prototype technologies that were classified according to their severity. The intrusion detection mechanism has 99% of accuracy.

Key-words: *Smart grid; Identity and Access Management; Multi-domain Access Control; Multi-factor Authentication; Intelligent Intrusion Detection.*

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

ABAC	*Attribute-based Access Control*
AM	*Access Management*
AMI	*Automatic Metering Infrastructure*
BLE	*Bluetooth Low Energy*
CA_dC	*Cryptographic Admission Control*
CH	*Context Handler*
CIA	*Central Intelligence Agency*
CLP	Controlador Lógicos de Processamento
CoAP	*Constrained Application Protocol*
CPS	*Cyber Physical Systems*
CS	*Central System*
CVE	*Common Vulnerabilities and Exposures*
DC	*Data Concentrator*
DCS	*Distributed Control System*
DK	*Data Key*
DLA_dC	*Device-based Location Admission Control*
DSoD	*Dynamic Separation of Duty*
DTLS	*Datagram Transport Layer Security*
FCA	*Formal Concept Analysis*
FIDO	*Fast IDentity Online*
FN	Falso Negativo
FP	Falso Positivo
HIDS	*Host-based Intrusion Detection System*
HMI	*Human Machine Interface*
IAM	*Identity and Access Management*
IdM	*Identity Management*
IdP	*Identity Provider*
IDS	*Intrusion Detection System*
iID	*intelligent Intrusion Detection*
IEC	*International Electrotechnical Commission*
IoT	*Internet of Things*
JKS	*Java KeyStore*
JWT	*JSON Web Token*
KDC	*Key Distributed Center*
KEK	*Key-Encrypting Key*
KH	*Key Hierarchy*
LAN	*Local Area Networking*
LMK	*Local Master Key*
M2M	*Machine-to-machine*
MAC	*Media Access Control*

M_dAC	*Multi-domain Access Control*
MFA	*Multi-Factor Authentication*
MIAM	*Multi-factor Identity and Access Management*
MW	Megawatts
NIDS	*Network-based Intrusion Detection System*
NCS	*Networked Control System*
OIDC	OpenID Connect
OTP	*One-Time Password*
OWL	*Ontology Web Language*
PAP	*Policy Administration Point*
PC	*Personal Computer*
PCAP	*Packet Capture*
PDP	*Policy Decision Point*
PEP	*Policy Enforcement Point*
PEM	*Privacy Enhanced Mail*
PenTest	*Penetration Test*
PIN	*Personal Identification Number*
PIP	*Policy Information Point*
QA_dC	*Quorum Admission Control*
QRCode	*Quick Response Code*
RBAC	*Role-based Access Control*
RC	*RBAC Controller*
REST	*Representational State Transfer*
SAN	*Sensor Actuator Network*
SAML	*Security Assertion Markup Language*
SCADA	*Supervisory Control And Data Acquisition*
SG	*Smart Grid*
SM	*Smart Meter*
SoD	*Separation of Duty*
SoS	*System of Systems*
SP	*Service Provider*
SQRL	*Secure Quick Read Login*
SRA	*Single Role Activation*
SSL	*Secure Socket Layer*
SSO	*Single Sign-On*
SSoD	*Static Separation of Duty*
TLS	*Transport Layer Security*
TOTP	*Time-based One Time Password*
UDP	*User Datagram Protocol*
WISN	*Wireless Industrial Sensor Network*
WSN	*Wireless Sensor Network*
XACML	*Extensible Access Control Markup Language*

Capítulo 1

Introdução

Redes elétricas inteligentes (SG, *Smart Grid*) consistem em sistemas de energia elétrica que utilizam inteligência computacional e tecnologias de comunicação de maneira integrada na geração, transmissão, distribuição e consumo de energia [1]. Ou seja, são responsáveis pelo controle e supervisão de infraestruturas críticas [2]. Infraestruturas críticas são sistemas e ativos, virtuais ou físicos, que são vitais a uma nação, de modo que a indisponibilidade ou destruição dessas infraestruturas podem causar grandes impactos na segurança e na economia de um país [3].

Pesquisas comprovam que nos últimos anos os sistemas de energia elétrica estão mais vulneráveis e, consequentemente, sujeitos a ciberataques [4]. Existem diversos motivos que justificam esse cenário, porém duas circunstâncias se destacam. Primeiramente, os controles físicos tradicionais estão sendo substituídos por microprocessadores, sistemas embutidos e principalmente por dispositivos da Internet das Coisas (IoT, *Internet of Things*), os quais permitem o acesso remoto interativo através da Internet. A utilização de dispositivos dessa natureza, que contêm sensores, poder de processamento e de atuação, é uma tendência natural adotada na Indústria 4.0 [5]. Entretanto, a exposição desses dispositivos na Internet pode comprometer toda infraestrutura crítica [3]. Além disso, no passado, os sistemas de energia elétrica eram compostos de *softwares* e *hardwares* que utilizavam protocolos proprietários. Porém, nos dias atuais os componentes são implantados em soluções convencionais (*commodity*) de TI que utilizam protocolos abertos.

A cibersegurança para SG emergiu como interesse crítico de organizações governamentais [2], devido a diversos incidentes. Por exemplo, o centro de controle de uma usina elétrica foi atacado em 2010, o que resultou numa perda de 900MW (*Megawatts*) em menos de 7 segundos [6]. No mesmo ano, a usina nuclear de *Natanz*, localizada no Irã, foi atacada pelo *Stuxnet* [7]. Os prejuízos foram incalculáveis e provocaram sérias deteriorações na infraestrutura física [8]. De acordo com um relatório da CIA (*Central Intelligence Agency*),

diversos sistemas de energia nos Estados Unidos foram invadidos por *hackers*, o que ocasionou apagões em diversas cidades [6].

Dessa maneira, os sistemas de energia elétrica possuem características peculiares que os diferem de sistemas corporativos convencionais. Por exemplo, a atualização de *software*, de maneira geral, não é uma atividade bem aceita [4] porque é economicamente difícil justificar a interrupção dos serviços para realizar uma atualização de segurança. Dessa maneira, uma simples atualização para correção de falhas de segurança pode exigir meses de planejamento, principalmente se for necessário reiniciar os serviços/servidores. Outro exemplo, em março de 2008, uma usina nuclear foi acidentalmente desligada após um servidor reiniciar por causa de uma atualização de *software* [6].

Os sistemas de energia elétrica estão expostos a ciberataques como qualquer sistema. Entretanto, os exemplos citados evidenciam que os ciberataques podem causar prejuízos incalculáveis na vida das pessoas e nos serviços públicos de uma nação. Portanto, a cibersegurança de sistemas de energia elétrica é de interesse científico.

1.1. Motivação

Para proteger os sistemas de energia elétrica, mais especificamente da SG, foram definidos requisitos padronizados por entidades conceituadas, como a IEC (*International Electrotechnical Commission*), NERC (*North American Electric Reliability Corporation*) e IEEE (*Institute of Electrical and Electronic Engineers*). Entretanto, esses requerimentos apresentam normas e procedimentos com granularidade grossa. Inclusive, a IEC realizou um mapeamento de normas para SG [9], no qual apresenta as diferentes aplicações de SG e suas respectivas normas e lacunas (*gaps*). Esse estudo concluiu que os padrões de cibersegurança existentes, inclusive os que apresentam relatórios técnicos, não são suficientes para cobrir toda a complexidade da arquitetura da SG. Além disso, o estudo apresenta que existe uma lacuna em relação ao gerenciamento de identidade e acesso em SG, uma vez que a SG é composta por diferentes entidades conectadas através de serviços.

A alta conectividade combinada com a heterogeneidade da SG é um desafio do ponto de vista de segurança. A arquitetura da SG é composta por sensores, atuadores, controladores e componentes de rede que integram desde dispositivos IoT (e.g. medidores elétricos inteligentes)

até nuvens computacionais. Essa arquitetura heterogênea aumenta a complexidade na definição de especificações ou mecanismos implementáveis para garantir a cibersegurança da SG.

A literatura apresenta algumas propostas de modelo de segurança [10, 11 e 12] cuja eficácia foi provada matematicamente, e que poderiam ser aplicados em um contexto de SG. Entretanto, quando esses modelos são implementados em um sistema, utilizam tecnologias que podem apresentar vulnerabilidades. Dessa maneira, permite o acesso não autorizado aos recursos protegidos da SG.

É essencial que novos mecanismos de cibersegurança, que adotam tecnologias consolidadas, sejam desenvolvidos para dificultar ao máximo que um adversário explore uma vulnerabilidade e comprometa a SG.

A alta conectividade presente na SG permite que os usuários realizem operações remotas em diferentes serviços. Contudo, é essencial que os serviços possuam a garantia da autenticidade do usuário. A autenticação multifator é uma tendência presente em diversos sistemas comerciais, pois permite a combinação de diferentes técnicas de autenticação independentes. Dessa maneira, caso o adversário consiga forjar apenas uma credencial específica, não conseguirá autenticar no sistema, pois não possui as demais credenciais. Entretanto, esses trabalhos não foram propostos para atuar em cenários de infraestrutura crítica, nos quais em diversos casos, um determinado conjunto de operações críticas devem ser acessadas a partir de um perímetro pré-definido, como por exemplo uma sala de operações em uma usina nuclear. Dessa maneira, é essencial a garantia da identidade do usuário e de sua localização.

Ademais, tipicamente a SG é um sistema de sistemas (SoS, *System of Systems*) que possui um comportamento emergente que agrega inteligência computacional. Nesse contexto, é usual o compartilhamento de informações e recursos protegidos. Dessa maneira, um recurso protegido pode ser acessado a partir de aplicações de um domínio externo, caracterizando uma operação multidomínios. Por exemplo, um sistema de gestão de energia nacional poderia acessar diversos sistemas de distribuição e geração de energia regionais (independentes) para dispor de uma visão macro da situação energética do país.

Nesse cenário, para garantir que apenas usuários autorizados acessem os recursos protegidos, é necessária a aplicação de um controle de acesso distribuído. Embora controle de acesso distribuído seja um tema relevante de pesquisa (capítulo 3), a literatura carece de

trabalhos que consideram diferentes fatores de controle de acesso. Assim, se o controle de acesso for comprometido, o adversário tem acesso a todo o sistema.

Finalmente, é essencial garantir a segurança dos dispositivos da IoT, pois comprometendo um único dispositivo é possível comprometer toda a SG [13]. A principal dificuldade em garantir a segurança está relacionada às restrições de poder computacional, que impossibilitam utilizar protocolos tradicionais de comunicação e segurança. Isso cria uma lacuna tecnológica de interoperabilidade e de padronização que impede a implantação de mecanismos tradicionais de autenticação e acesso.

1.2. Questão de pesquisa

A questão que esta pesquisa pretende responder é: **como é possível proteger o ecossistema de SG utilizando gestão de identidade e acesso e, adicionalmente, identificar ataques aos mecanismos de segurança utilizando detecção de intrusão?**

1.3. Objetivos

O objetivo geral desta pesquisa é **propor um mecanismo de gestão de identidade e acesso multifator para SG, que seja protegido por um mecanismo de detecção de intrusão inteligente criado a partir de um processo de pentest**.

Para cumprir o objetivo geral, os seguintes objetivos específicos devem ser atingidos:

i. Desenvolver um mecanismo de gestão de identidade que combina mecanismos de autenticação multifator de maneira flexível para reduzir a probabilidade de invasão à SG por parte de um adversário oriundo da Internet.

ii. Desenvolver um controle de admissão baseado na localização do dispositivo do usuário que mitiga a possibilidade de um adversário oriundo da Internet acessar o mecanismo de autenticação.

iii. Desenvolver um mecanismo de gestão de acesso que combine políticas de controle de acesso baseadas em papéis com um controle de admissão de criptografia e quórum com o objetivo de limitar o acesso a recursos locais e remotos.

iv. Conceber o modelo de adversário dos mecanismos de segurança propostos, considerando o cenário em que o adversário consegue controlar uma entidade por vez.

v. Conceber um cenário de ataque, composto por análise de ferramentas de vulnerabilidade, testes de intrusão de caixa branca e interceptações *man-in-the-middle* (MITM).

vi. Desenvolver um mecanismo de detecção de intrusão inteligente utilizando aprendizagem de máquina com utilização de registros gerados pelo cenário de ataque para treinar o modelo de detecção.

vii. Desenvolver um protocolo baseado em chaves criptográficas que assegura a autenticidade e confidencialidade na comunicação dos dispositivos da IoT.

viii. Testar e avaliar o protótipo que implementa a proposta.

1.4. Contribuições

As principais contribuições deste trabalho são listadas a seguir:

- Mecanismo de Gestão de Identidades que combina fatores de autenticação para atenuar as chances de sucesso de um adversário oriundo da Internet acessar a SG. Para utilizar esse mecanismo, o usuário precisa de uma credencial de proximidade emitida pelo controle de admissão baseado em localidade.
- Mecanismo de Gestão de Acesso que combina políticas de controle de acesso baseadas em papéis com um controle de admissão criptográfico e de quórum para limitar o acesso, local e remoto, a um recurso protegido.
- Modelo de detecção de intrusão baseado em técnicas de aprendizagem de máquina para proteção exclusiva do mecanismo de gestão de identidade e acesso. O modelo de detecção de intrusão foi treinado a partir de um processo de *pentest* que visa identificar as vulnerabilidades da proposta baseada no modelo de adversário.
- Mecanismos de gestão de identidade e acesso compatível com as restrições computacionais da IoT. Esse mecanismo permite que um usuário autenticado

transporte suas credenciais para o contexto da IoT, mantendo a autenticidade e confidencialidade na comunicação

1.5. Organização

Este documento está organizado da seguinte forma. O capítulo 2 trata da fundamentação teórica. O Capítulo 3 discute os trabalhos relacionados à proposta. O Capítulo 4 disserta sobre o modelo de gestão de identidade e acesso multifator. O Capítulo 5 apresenta um estudo de caso da proposta aplicada em um cenário de SG. O Capítulo 6 discute os resultados do trabalho. Finalmente, o Capítulo 7 apresenta a conclusão do trabalho.

Capítulo 2

Fundamentação

Este capítulo apresenta conceitos relacionados à gestão de identidades, controle de autenticação, controle de acesso, sistemas de detecção de intrusão, e *smart grid*. Tais conceitos são fundamentos substanciais para o entendimento da proposta.

2.1. Gestão de Identidade (IdM, *Identity Management*)

A identidade representa uma entidade em um contexto particular [14]. Tipicamente, uma identidade é formada por um identificador com credenciais e atributos que representam características da entidade. Por exemplo, ao considerar que a entidade é uma pessoa, seu identificador pode ser seu CPF e seus atributos são informações pessoais relevantes.

A relevância das informações depende do contexto da aplicação, por exemplo, os atributos exigidos em uma transação bancária são diferentes dos atributos exigidos para ouvir música *online*. Um sistema de gestão de identidade (IdM, *Identity Management*) tem a função de controlar identidades através da autenticação, transportar de maneira segura os atributos de identidade, autorizar o acesso a recursos protegidos, delegar as identidades entre domínios federados de maneira segura [14].

Um IdM é formado por três componentes [15]:

- **Provedor de Serviço (SP, *Service Provider*):** provê serviços ao usuário, como: serviço bancário, *e-commerce*, rede social;
- **Provedor de Identidade (IdP, *Identity Provider*):** provê identidade ao usuário para utilização dos serviços (SP). É responsável pela autenticação do usuário e pelo processamento das requisições dos SP;

- **Usuário**: é um cliente do SP e do IdP que necessita de uma identidade para utilizar os serviços. Na prática, um usuário pode ser uma pessoa, organização, entidade virtual etc.

Dessa forma, a utilização de um IdM possui quatro objetivos dominantes [15]: (i) fornecimento de identidade – baseado em registro único no IdP, diversos SPs podem utilizar diferentes identidades para o mesmo usuário; (ii) autenticação única (SSO, *Single Sign-On*) – baseado em uma única autenticação em um SP, diversos SPs acessam o mesmo IdP; (iii) compartilhamento de atributos – os atributos de identidade especificados em determinado SP podem ser reutilizados em outros SPs; (iv) autorização de acesso – restringe o acesso de um SP a um recurso protegido sem precisar acessar as credenciais do usuário.

A literatura adota três modelos principais de IdM que determinam a relação entre SP e IdP, sendo eles o modelo tradicional, centralizado e federado. As subseções a seguir especificam cada modelo.

2.1.1. Modelo Tradicional

No modelo tradicional, chamado também de isolado, cada SP tem o papel de provedor de serviço e de identidade. Dessa forma, a gestão de identidade e de serviço é gerenciada pela mesma entidade. As operações do IdM, como autenticação e autorização de acesso, são implementadas pelo SP. A Figura 2.1 ilustra o modelo tradicional.

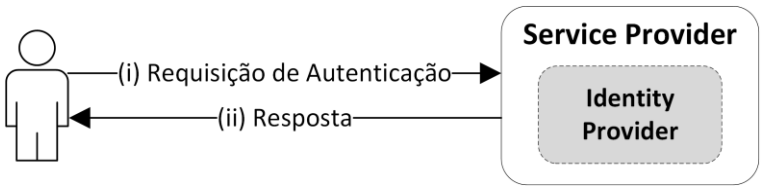

Figura 2.1. Modelo tradicional de IdM. Adaptado de [14].

Por ser um modelo elementar, traz consigo inúmeras desvantagens. O usuário necessita gerenciar várias identidades, uma para cada SP que precise utilizar. Além disso, todas as funcionalidades administrativas da gestão de usuários, como cadastro de usuários e recuperação de senha, devem ser implementadas em cada SP.

2.1.2. Modelo Centralizado

O modelo centralizado é baseado na arquitetura cliente/servidor. A implementação das funções de autenticação e a gestão de identidades são realizadas em servidores distintos. Dessa forma, o SP não armazena as credenciais do usuário localmente. Todas as identidades são enviadas para um IdP centralizado, permitindo o armazenamento de todos os usuários, em um único domínio. A Figura 2.2 ilustra o modelo centralizado.

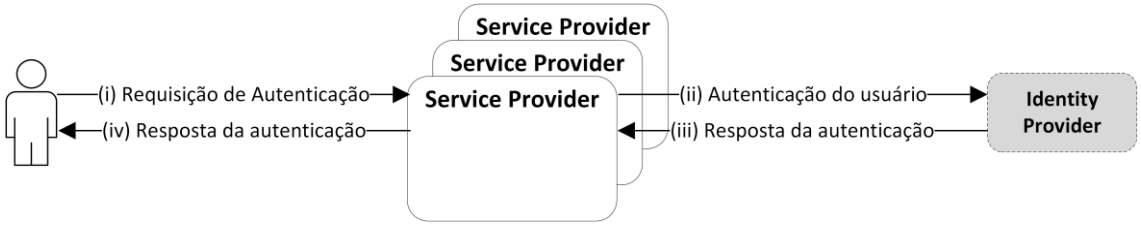

Figura 2.2. Modelo centralizado de IdM. Adaptado de [14].

O modelo centralizado atualmente é o mais popular em aplicações comerciais, porém possui algumas desvantagens por utilizar um único servidor. Esse modelo, em sua essência, não escala para um número elevado de usuários devido à complexidade da gestão de identidades para cada SP. Outro ponto negativo na disposição dos elementos dessa abordagem é tornar o provedor de identidades um ponto único de falhas.

2.1.3. Modelo Federado

O modelo federado integra diversos domínios, formando um domínio global virtualizado. Isso permite que diversos IdPs partilhem da identidade dos usuários, desde que exista uma relação de confiança pré-estabelecida entre os domínios. A Figura 2.3 ilustra o modelo federado.

Existe um mapeamento das identidades de um determinado usuário entre os diversos IdP. Dessa forma, quando um usuário está autenticado em um IdP, considera-se que ele está autenticado em todos, não necessitando uma segunda autenticação.

As desvantagens desse modelo estão relacionadas à necessidade de uma relação de confiança entre os domínios que compõem a federação, além da complexidade de implantação de mecanismos dessa natureza.

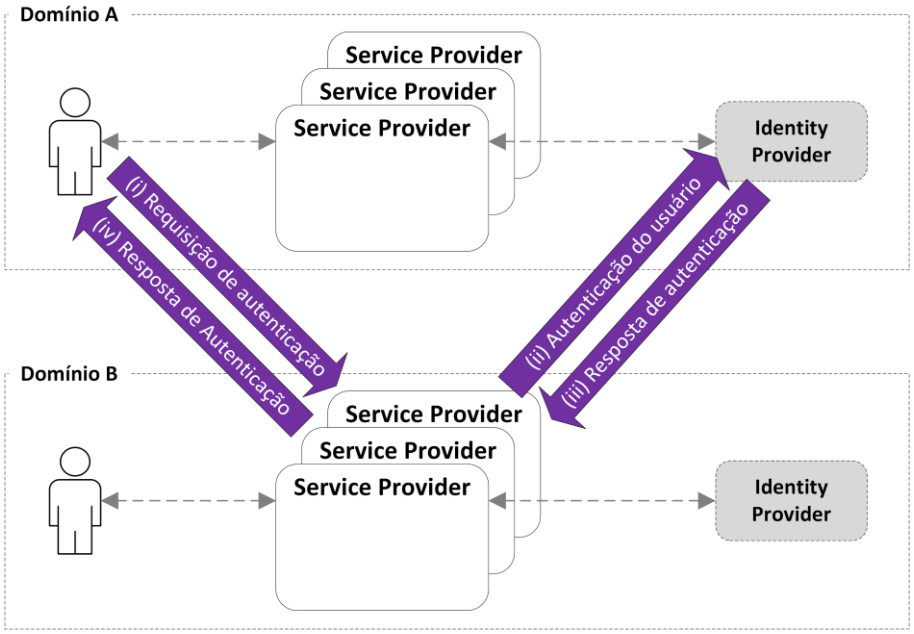

Figura 2.3. Modelo federado de IdM. Adaptado de [14].

2.1.4. OAuth

OAuth é um *framework open source* para autorização de acesso para ambiente *web*. Seu objetivo é permitir que um SP obtenha acesso restrito a um serviço, sem a necessidade de compartilhamento de credenciais [16]. O OAuth utiliza um *token* de acesso que define parâmetros para restrição das ações que o usuário é capaz de realizar no domínio do *token* de acesso, por exemplo: escopo de acesso, tipo do *token*, tempo de expiração e *token* de *refresh* (referência um novo *token* quando o atual expirar).

A Figura 2.4 representa o fluxo do protocolo OAuth. Inicialmente o SP solicita ao proprietário do recurso a autorização para acessá-lo *(Figura 2.4, evento i e ii).* Em seguida, já de posse da autorização emitida pelo proprietário do recurso, o SP troca a autorização emitida, que normalmente é um código temporário (*ticket*), por um *token de acesso (Figura 2.4, evento iii e iv).* Ao apresentar o *token de acesso* ao servidor (guardião) do recurso, é possível acessar o recurso, porém o *token de acesso* deve ser válido e possuir o escopo de acesso necessário *(evento v e vi).*

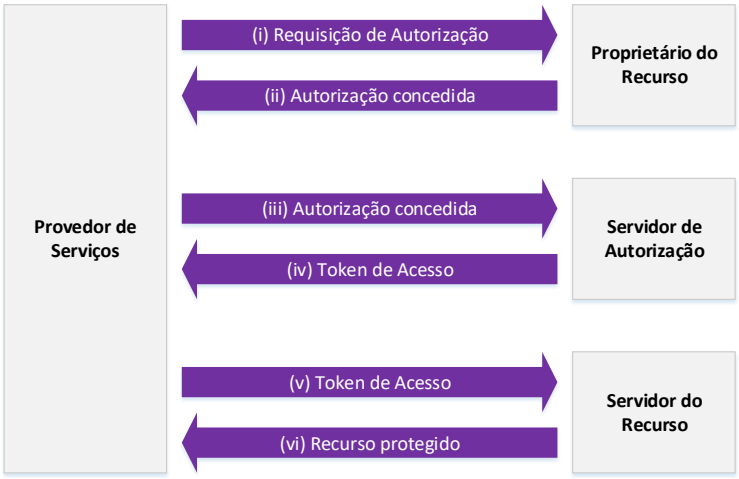

Figura 2.4. Fluxo do protocolo OAuth. Adaptado de [16].

2.1.5. OpenID Connect

O OpenID Connect (OIDC) é um protocolo de IdM desenvolvido sobre o OAuth 2.0. Por consequência, é uma extensão que permite ao SP a verificação da identidade de cada usuário [17]. O OIDC dispõe um *token de identidade* (*ID Token*) que contém informações sobre a autenticação e os atributos do usuário. Por se tratar de uma extensão do OAuth 2.0, além do *token de identidade*, o OIDC também fornece o *token de acesso*.

O processo de obtenção dos *tokens* através do OIDC é ilustrado na Figura 2.5. Inicialmente o SP solicita a autenticação do usuário no OIDC *(Figura 2.5, evento i)*. O usuário fornece suas credenciais *(Figura 2.5, evento ii) e*, caso sejam válidas, o mesmo será questionado se autoriza ou não que o SP acesse as suas informações definidas no escopo do *token (Figura 2.5, evento iii)*. O OIDC responde ao SP sobre o processo de autenticação e o consentimento de acesso do usuário *(Figura 2.5, evento iv)*. Finalmente, o SP pode obter do OIDC o *token de acesso e de identidade* do usuário (*eventos v e vi*, respectivamente).

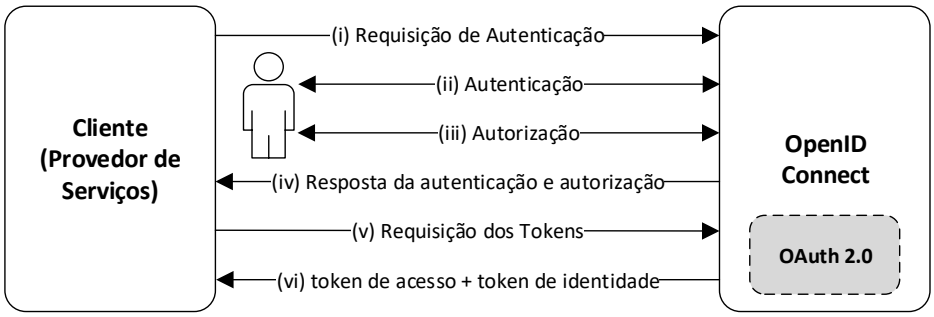

Figura 2.5. Visão geral das trocas do protocolo OIDC. Adaptado de [17].

2.2. Controle de Autenticação

A autenticação pode ser determinada como o processo de verificação de uma identidade alegada por uma entidade do sistema [18]. Esse processo é composto de duas etapas: (i) etapa de identificação: momento que o usuário apresenta um identificador ao sistema; (ii) etapa de verificação: momento que o usuário apresenta uma informação que certifica a validade da alegação de identidade. É essencial que a identidade seja cuidadosamente escolhida, como a subseção 2.1 que apresentou detalhes sobre os conceitos de identidade.

A etapa de verificação tipicamente utiliza múltiplas abordagens denominadas fatores. Esses fatores podem ser aplicados individualmente ou combinados. A combinação de alguns fatores de autenticação é intitulada autenticação multifator [18]. Essa técnica reduz o sucesso de invasões a um determinado sistema. Os fatores mais populares de autenticação são: (i) algo que o usuário sabe, por exemplo: senha, número de identificação pessoal (PIN, *Personal Identification Number*); (ii) algo que o usuário tem, por exemplo: *token* ou *smartphone*; (iii) algo que o usuário é ou faz, por exemplo: características biométricas.

De forma adicional às técnicas tradicionais de autenticação citadas, novas técnicas situam-se em destaque, como o georreferenciamento e a baseada em leitura óptica. A técnica de autenticação por georreferenciamento fornece segurança ao sistema, em razão de definir a localização de um usuário a partir de um local predeterminado. Existem variados mecanismos que permitem esse tipo de autenticação, porém a forma mais popular é a autenticação utilizando recursos do *smartphone* do usuário. Utilizar o *smartphone* para realizar a autenticação propicia viabilidade da técnica de leitura óptica, que consiste na utilização da câmera para realizar a leitura de um código visual, usualmente um QRCode (Quick Response Code). Nessa técnica, chamada de SQRL (*Secure Quick Read Login*), o usuário digitaliza um QRCode de curta duração através de um aplicativo que notifica o provedor de identidades a respeito das credenciais do usuário. Ambas as técnicas são significativas na redução da probabilidade que um invasor externo (e.g. Internet) obtenha êxito no acesso ao sistema. Outro quesito relevante, é que o dispositivo *smartphone* possui o número de telefone, que é um atributo único e não é facilmente forjável, utilizado na implementação de técnicas com senhas descartáveis (OTP, *One-Time Password*).

2.3. Controle de Acesso

O controle de acesso é um mecanismo de segurança que permite limitar ações ou operações que determinado sujeito (humano ou máquina) pode realizar sobre um recurso [19]. O controle de acesso é tipicamente utilizado após um usuário estar devidamente autenticado, ou seja, limitará o acesso a usuários legítimos. Assim, deve existir um serviço de autenticação responsável por validar a identidade de um sujeito.

A arquitetura de um controle de acesso geralmente é formada por um monitor de referências, que é responsável por intermediar as tentativas de acesso a um determinado recurso, consultando uma base de autorização que é mantida pelo administrador do sistema. Assim, o monitor de referências encaminha sua decisão ao guardião do recurso (*enforcement*), que executa a decisão permitindo ou negando o acesso do usuário ao recurso protegido.

Uma base de autorização, em sua forma mais primitiva, pode ser representada conceitualmente como uma matriz de acesso, onde cada linha da matriz é um sujeito (usuário) e cada coluna é um objeto (recurso). Cada célula dessa matriz representa a autorização que o usuário tem sobre o recurso. Por exemplo, se o recurso for um arquivo, normalmente as autorizações são para leitura, escrita e execução. O objetivo do controle de acesso é permitir que o usuário realize apenas as permissões contidas nessa matriz de acesso.

Para uma solução de segurança mais completa é recomendada a utilização de um serviço de auditoria para registrar todas as requisições de acesso e atividades do sistema, permitindo realizar uma análise a *posteriori* de tentativas de invasão e possíveis violações no sistema. A Figura 2.6 ilustra um controle de acesso trabalhando em conjunto com outros serviços de segurança.

As políticas de um controle de acesso determinam o resultado da decisão de acesso. As políticas mais tradicionais são as discricionárias, mandatórias, baseadas em papéis ou em atributos. É importante ressaltar que as políticas não são exclusivas, ou seja, podem ser combinadas. As subseções a seguir apresentam as políticas de controle de acesso baseada em papéis e atributos, que são políticas amplamente utilizadas.

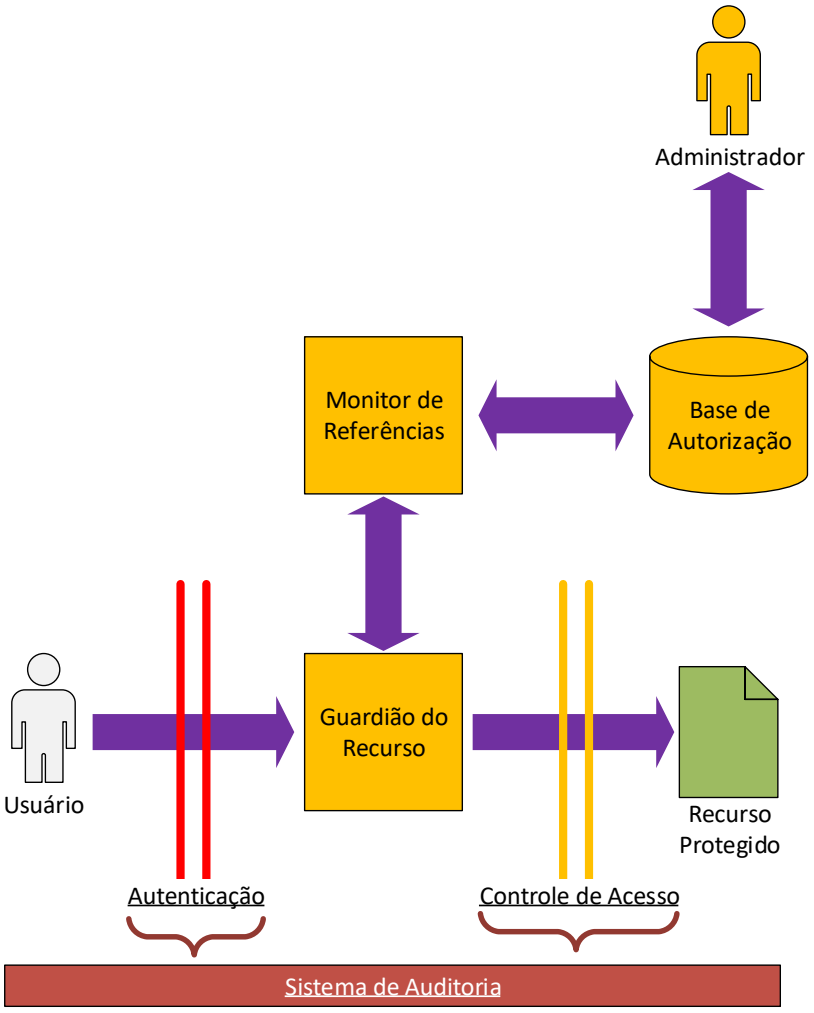

Figura 2.6. Controle de Acesso e outros serviços de segurança. Adaptado de [20].

2.3.1. Controle de Acesso baseado em Papéis (RBAC)

O Controle de Acesso Baseado em Papéis (RBAC, *Role-based Access Control*) utiliza papéis para intermediar a ligação entre usuários e permissões. O conceito de papéis é utilizado em sistemas há décadas, porém apenas nas últimas duas décadas se consolidou como um modelo de controle de acesso maduro [19]. O início desse processo foi realizado por Ferraiolo e Kuhn [21] em 1992, quando desenvolveram as primeiras especificações técnicas e formais do RBAC. Três anos mais tarde, os mesmos autores criaram um modelo expandido de RBAC, que além de suportar hierarquia de papéis, permitiu o tratamento de conflitos de interesses. Em meados de 2000, o NIST, liderado por Ravi Sandhu, propôs que o RBAC se tornasse um padrão, o que foi aceito em 2004. Após se tornar um padrão, o RBAC tem sido utilizado como modelo padrão de controle de acesso em diversos sistemas.

O modelo do RBAC é formado a partir dos seguintes componentes: principal, hierárquico e conflito de interesses. A primeira parte do modelo do RBAC, denominada *Core* RBAC, define os elementos e princípios básicos do RBAC. A Figura 2.7 ilustra os elementos e seus relacionamentos.

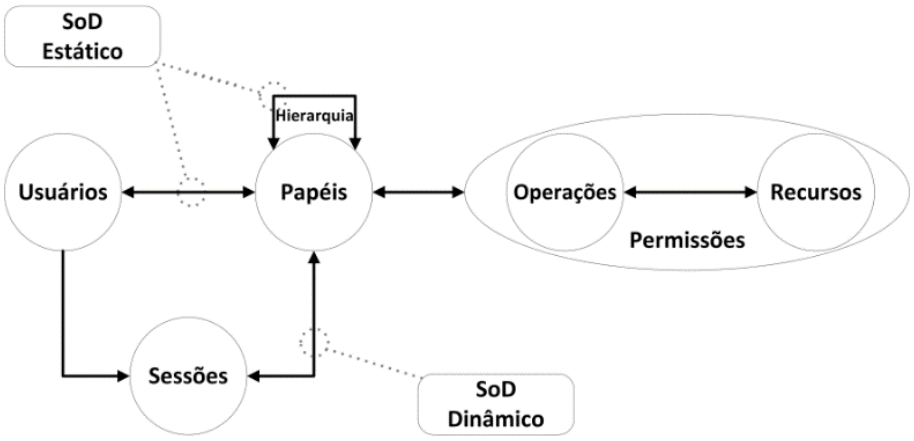

Figura 2.7. Modelo do RBAC. Adaptado de [19].

Os elementos do RBAC são definidos formalmente como [19]:

- **Usuário**: pode ser definido como um ser humano, máquina, rede, agente autônomo inteligente, dentre outros;
- **Papel**: uma função de trabalho no contexto de uma organização, com uma semântica associada a uma autoridade e a uma responsabilidade;
- **Sessão**: mapeamento entre um usuário e um subconjunto de papéis ativos;
- **Permissão**: aprovação de uma operação em um ou mais recursos protegidos;
- **Operação**: executa uma função para o usuário;
- **Recurso**: qualquer recurso/objeto do sistema.

- **Separação de Deveres Estática/Dinâmica**: Separação de Deveres define restrições para associação e ativação de papéis.

Observe que não existe uma relação direta entre o usuário e a permissão, o administrador do sistema associa usuários aos papéis, que por sua vez são associados às permissões. Isso acarreta vantagens para a manutenção, e também para o ciclo de vida do sistema. Caso um usuário seja desligado do sistema, basta desassociá-lo do papel em que está associado.

O modelo adota o conceito de privilégio mínimo em que o usuário não deve possuir privilégios superiores, mas apenas o necessário para realizar suas funções de trabalho. Isso

minimiza o problema de um indivíduo ter a capacidade de realizar funções indesejadas. Logo, mesmo que um usuário possua diversos papéis, apenas os papéis realmente necessários estarão ativos no sistema (através da sessão).

A segunda parte do modelo, denominada *Hierarchical* RBAC, define um meio natural para estruturação dos papéis com objetivo de refletir as linhas de autoridade e de responsabilidade de uma organização. Para que isso ocorra, existe uma ordem hierárquica entre os papéis, em que um papel sênior (pai) herda as permissões do papel júnior (filho) e o papel júnior herda os usuários do papel sênior. Exemplificando: um papel de diretor herda as permissões de um papel analista, e o papel analista herda os usuários do papel diretor.

As hierarquias estão divididas em dois tipos: Geral e Limitada. A hierarquia geral permite herança múltipla de papéis, no qual um determinado papel pode ter mais que um papel sênior. Por outro lado, a herança limitada permite apenas uma herança simples, estruturando os papéis como uma árvore.

A terceira parte do modelo, denominada de *Constrained* RBAC, determina que uma operação crítica deve ser realizada por duas ou mais pessoas. Esse procedimento evita que uma pessoa individualmente comprometa o sistema. Para reduzir a possibilidade de conluio, indivíduos de diferentes habilidades ou interesses divergentes são atribuídos a tarefas individuais necessárias na execução de uma determinada função. A motivação é garantir que fraudes e erros graves não ocorram sem conluio deliberado de múltiplos usuários. Assim, existem dois tipos de separação de deveres (SoD, *Separation of Duty*): estático e dinâmico.

A separação de deveres estática (SSoD, *Static Separation of Duty*) é implementada na associação de usuários com papéis e tem como objetivo limitar que um usuário detenha dois papéis conflitantes. Por exemplo, um sistema pode estabelecer que para acessar determinado cofre são necessários dois papéis: diretor administrativo e diretor financeiro. Nesse cenário, o usuário não pode ser associado ao papel de diretor administrativo e de diretor financeiro.

A separação de deveres dinâmica (DSoD, *Dynamic Separation of Duty*) tem o mesmo propósito da separação de deveres estática, porém é aplicada em local diferente. A separação de deveres dinâmica é aplicada na ativação do papel, ou seja, um usuário pode ter dois papéis possivelmente conflitantes desde que ele não ative ambos ao mesmo tempo. Por exemplo, um usuário pode ser autorizado a ter o papel de caixa e supervisor de caixa, sendo que o supervisor pode efetuar correções feitas pelo caixa. Se um usuário desejar ativar o papel de supervisor de caixa, ele precisará desativar o papel de caixa para ativar o papel de supervisor. Essa

propriedade da separação de deveres dinâmica reforça o princípio do privilégio mínimo, no qual cada usuário possui diferentes níveis de permissão em momentos distintos, dependendo do papel ativo. Isso assegura que as permissões não persistam além do tempo necessário para cumprimento do trabalho.

2.3.2. Controle de Acesso baseado em Atributos (ABAC)

O Controle de Acesso Baseado em Atributos (ABAC, *Attribute-Based Access Control*) é um modelo de controle de acesso que adota políticas que expressam regras *booleanas* utilizando atributos [22]. O usuário possui um conjunto de atributos que são utilizados para avaliar sua autorização. O ABAC evita que as permissões estejam ligadas diretamente a um usuário ou a um papel. Logo, quando uma requisição de acesso é solicitada, um mecanismo realiza a decisão baseada nos atributos do usuário, recursos, ambientes, entre outros. É possível utilizar papéis no ABAC, considerando que estes são atributos do sujeito.

O XACML é um popular *framework* do ABAC que define uma linguagem baseada em XML para escrita de políticas de controle de acesso, requisições e respostas. Adicionalmente, provê um mecanismo de avaliação das políticas de controle de acesso [23]. Por padrão, o modelo de avaliação do XACML é baseado em atributos. Entretanto, o XACML dispõe de um *profile* (extensão) para RBAC, cujo os papéis são atributos do sujeito [78].

As entidades dominantes do XACML são: (i) PAP (*Policy Administration Point*), que permite criar e armazenar as políticas de controle de acesso; (ii) PEP (*Policy Enforcement Point*), responsável por encaminhar as requisições de acesso dos usuários ao *Context Handler*, atuando como guardião dos recursos; (iii) PIP (*Policy Information Point*), que atua como a fonte de atributos para o *Context Handler*; (iv) PDP (*Policy Decision Point*), que avalia as políticas e produz a decisão de acesso, classificada como permitido ou negado; e (v) *Context Handler*, responsável por realizar a coordenação, a adequação e a interoperabilidade de atributos, requisições e credenciais entre as entidades do XACML.

A Figura 2.8 apresenta uma visão geral do protocolo XACML. Nela, um usuário requisita acesso a um recurso protegido *(Figura 2.8, evento i)*. O PEP intercepta a requisição de acesso e a encaminha para o *Context Handler (Figura 2.8, evento ii).* O *Context Handler* gerencia as informações (atributos) do usuário providas pelo PIP para criar uma requisição no formato XACML para o PDP *(Figura 2.8, evento iii).* O PDP requisita ao PAP as políticas

relacionadas aos recursos e realiza a decisão sobre a requisição de acesso. O PDP retorna a decisão (permitido/negado) utilizando o formato XACML para o *Context Handler (Figura 2.8, evento iv).* O *Context Handler* informa ao PEP a decisão *(evento v).* O PEP permite ou rejeita o acesso do usuário de acordo com a decisão do PDP.

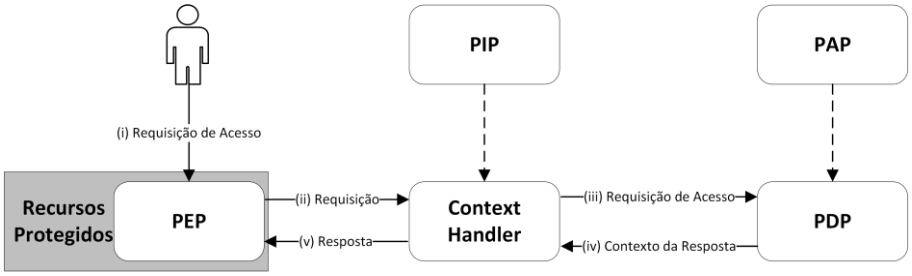

Figura 2.8. Arquitetura simplificada das entidades XACML. Adaptado de [23].

2.4. Sistema de Detecção de Intrusão (IDS, *Intrusion Detection System*)

Sistema de Detecção de Intrusão (IDS, *Intrusion Detection System*) é um serviço de segurança que monitora e analisa eventos do sistema com a finalidade de identificar e alertar, em tempo real, a ocorrência de tentativas de intrusão [18]. Caso uma intrusão seja detectada com rapidez, o intruso pode ser identificado antes de causar danos ou comprometimento do sistema. Assim, um IDS tem como premissa que o comportamento do intruso é distinto do comportamento do usuário legítimo. Entretanto, em muitos casos, não existem uma distinção nítida entre um ataque de um intruso de uma utilização normal de um usuário autorizado. Dessa maneira, um IDS pode emitir alertas: (i) Falso Positivo (FP): quando um usuário autorizado é identificado como intruso; (ii) Falso Negativo (FN): quando um intruso é identificado como usuário legítimo.

A arquitetura de um IDS tipicamente é formada por quatro módulos [24]: (i) Aquisição de eventos: responsável por efetuar a leitura dos eventos no ambiente monitorado (e.g. leitura dos pacotes de redes); (ii) Pré-processamento: responsável por preparar as informações antes do módulo de detecção ser executado (e.g. extração de características); (iii) Detecção: responsável por processar os dados e detectar se o evento é uma intrusão ou não; (iv) Alerta: caso o evento seja classificado como intrusão, é gerado um alerta.

A Figura 2.9 apresenta a árvore de classificação de técnicas de IDS [76]. O IDS pode ser classificado de acordo com a técnica de detecção e com a fonte de dados utilizadas na auditoria. A técnica de detecção pode ser baseada em assinatura ou em anomalia. A detecção

baseada em assinatura pretende extrair características dos padrões dos ataques conhecidos. Assim, a cada evento uma base é consultada para verificar se a ocorrência é uma intrusão ou não. A principal desvantagem dessa abordagem está relacionada à incapacidade de detectar ataques desconhecidos que não estão na base. Por outro lado, a detecção baseada em anomalia utiliza um modelo de ataque que pode ser utilizado para detectar novos ataques. A detecção de um ataque ocorre quando o mecanismo de classificação identifica similaridades entre o modelo de ataque e os atributos do evento. Os dados utilizados na auditoria podem ser providos baseados em rede (NIDS, *Network-based Intrusion Detection System*), baseado em aplicação (APIDS, *Application-based based Intrusion Detection System*) ou em *host* (HIDS, *Host-based Intrusion Detection System*). O NIDS detecta apenas o conteúdo no nível de rede (e.g. ataque de negação de serviço), já um HIDS detecta ataques em nível de sistema (e.g. escalação de privilégio), por fim, o APIDS detecta ataques específicos de uma aplicação.

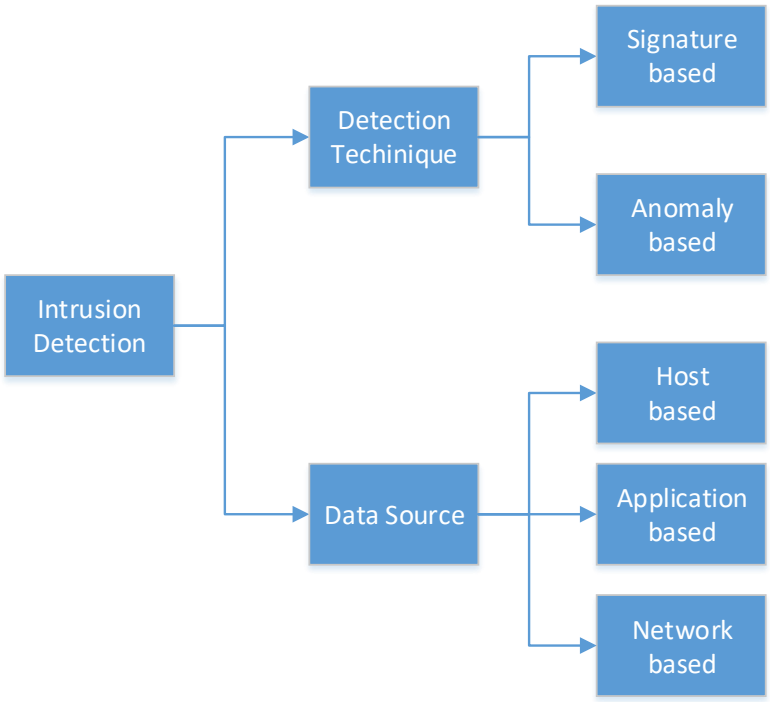

Figura 2.9. Classificação de IDS. Adaptado de [76].

É importante ressaltar que as técnicas de coleta de dados de auditoria podem ser combinadas para criar um IDS híbrido, que combina as técnicas de NIDS e HIDS. Assim, o IDS consegue detectar intrusões a nível de rede e *host*.

2.5. Smart grid

A *smart grid* (SG) utiliza fluxos bidirecionais de eletricidade e informação para criar um sistema automatizado de distribuição de energia. Mais especificamente, a SG pode ser considerada como um sistema que utiliza informações, tecnologias de comunicação bidirecionais, cibersegurança e inteligência computacional de maneira integrada para geração, transmissão, distribuição, subestações e consumo de energia [1]. A SG tem o propósito de disponibilizar um sistema limpo, seguro, confiável, resiliente, eficiente e sustentável, contemplando desde a geração da energia até o consumidor final.

A evolução da SG não está relacionada somente com os avanços tecnológicos, mas também na sofisticação do monitoramento, analise, otimização, segurança e controle a partir do sistema central (CS, *Central System*) da empresa de energia. A medição inteligente de energia é o mecanismo mais importante usado na SG, utilizado para coletar e gerenciar informações dos consumidores e dos dispositivos em campo. Essas atividades são de responsabilidade da AMI (*Automatic Metering Infrastructure*), sendo que uma de suas principais funcionalidades é coletar diagnósticos, consumo de energia e informações de estado dos medidores inteligentes de energia (SM, *Smart Meter*) para serem transferidos para o CS com o objetivo de cobrança, análise e resolução de problemas. A AMI, presente na SG, tem como grande diferencial a comunicação por duas vias, que permite a transmissão de informações quase em tempo real e sob demanda, possibilitando o aperfeiçoamento das operações do sistema e gerenciamento por parte do consumidor.

Na perspectiva do consumidor, a medição inteligente de energia oferece potenciais benefícios. Por exemplo, o consumidor pode estimar o valor de sua fatura no final do mês, assim podendo gerenciar o consumo para reduzir sua fatura. Na perspectiva da empresa de energia, as empresas podem disponibilizar em tempo real o valor da energia para encorajar os consumidores a reduzir a demanda em horários de picos ou aproveitarem horários com a tarifa reduzida.

2.5.1. Arquitetura de AMI

A arquitetura típica de AMI de uma SG, apresentada na Figura 2.10, é composta de SMs, que são instalados nas residências dos consumidores. Os SMs viabilizam a comunicação

de duas vias com o CS. Normalmente são medidores eletrônicos responsáveis por armazenar o consumo de energia em um intervalo de no máximo uma hora para ser enviado ao CS, pelo menos uma vez por dia, com propósito de monitoramento e cobrança. Adicionalmente, o SM pode conectar e desconectar remotamente um dispositivo do usuário (*appliance*) para gerenciar a carga da residência.

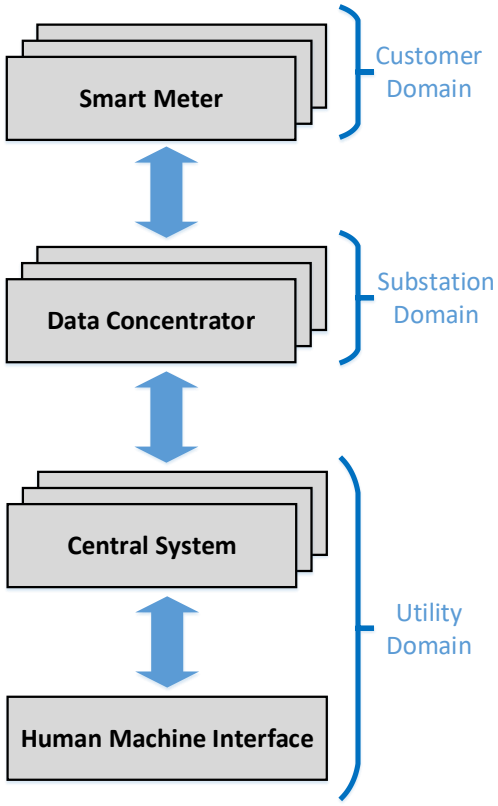

Figura 2.10. Arquitetura típica de um sistema de medição na SG. Adaptado de [1].

O SM possui diversos requisitos [26]: (i) frequentemente são dispositivos de baixo custo, com recursos computacionais limitados que restringem o uso de protocolos de comunicação e segurança tradicionais. Dessa maneira, devido às restrições de processamento, bateria, memória e rede conseguem apenas implementar algumas funcionalidades limitadas e unitárias.; (ii) devem ser acessíveis e configuráveis (parametrizáveis) remotamente; (iii) a segurança física e lógica é um aspecto essencial, pois comprometendo um único SM, os adversários podem comprometer toda a operação da SG.

O DC (*Data Concentrator*) é responsável por agregar e concentrar essas informações fornecidas pelos SMs para transmitir para o CS. O CS, por sua vez, é responsável por realizar a parte de negócios do sistema. Possui uma infraestrutura robusta, normalmente alocada em nuvens computacionais, possuindo processamento de larga escala. Suas principais

funcionalidades são: coletar, monitorar e supervisionar os dados. O CS é considerado o elemento crítico da arquitetura, uma vez que possui uma ampla visão da situação dos medidores, com potencial de tomada de decisões estratégicas a respeito do negócio. Finalmente, um usuário (operador) pode acessar o sistema para gerenciar a infraestrutura através da HMI (*Human Machine Interface*).

2.5.2. Frameworks

Sistemas de Controle Supervisório de Aquisição de Dados (SCADA, *Supervisory Control And Data Acquisition*) são responsáveis pela supervisão e controle de processos que exigem coleta de dados para tomada de decisões [27]. Esses sistemas são utilizados em diferentes setores, como energia, indústria, saneamento etc. Esses sistemas destacam-se pela diversidade de *frameworks* disponíveis. No setor de energia, o SCADA tipicamente atua de forma semelhante ao CS, se comunicando com os DCs e SMs.

Sistemas SCADA foram concebidos no início da década de 60 e permanecem presentes em indústrias e usinas até os dias atuais. A literatura categoriza os sistemas SCADA em quatro gerações [28]: (i) **monolítica**: essa geração está relacionada ao início da computação, cujo os sistemas eram centralizados em *mainframes*; (ii) **distribuída**: essa geração de sistemas SCADA aproveitou a evolução dos computadores pessoais (PC, *Personal Computer*) e da *Local Area Networking* (LAN) para distribuir o processamento entre várias máquinas. Dessa maneira, várias estações formam uma rede local para compartilhar informações em tempo real; (iii) **Internet**: A terceira geração do SCADA está intimamente relacionada com a segunda geração, no qual em síntese o diferencial é o emprego da Internet, a utilização de uma arquitetura e protocolos abertos. Nessa geração diversas vulnerabilidades foram expostas, e consequentemente diversos ataques foram identificados.

Apenas em 2012, mais de 200 ataques aos sistemas SCADA foram registrados [29]. O principal motivo da vulnerabilidade desses sistemas está relacionado à transição de gerações. A grande maioria desses sistemas não foi planejada para operar na Internet, apenas adaptados durante as diversas gerações. Para solucionar esses problemas, a segurança dos sistemas SCADA da próxima geração deve ser definida desde os primeiros estágios da construção. Assim, a quarta geração, chamada de geração da Internet das Coisas, pretende utilizar os limites das tecnologias da IoT, *big data*, aprendizagem de máquina e computação em nuvem.

Framework	Linguagem	Requisitos	Ambiente	Compatível com a IoT	Operações Multidomínios
ScadaBR[1]	Java	Tomcat v.6	Web	Não	Não
openSCADA[2]	Java	Eclipse	API	Não	Não
FreeSCADA[3]	C#	.NET 3.0	Desktop	Não	Não
PascalSCADA[4]	Delphi	Lazarus	Desktop	Não	Não
IndigoSCADA[5]	C, C++	Qt	Desktop	Não	Não
SZARP[6]	C, C++, Python	Linux	Desktop	Não	Não

Tabela 2.1. Comparação de *frameworks* SCADA de código aberto.

A quarta geração de sistemas SCADA encontra-se em processo de construção [28]. No mercado, diversos sistema SCADA de terceira geração estão disponíveis com o código aberto *(open source)*, conforme apresenta a tabela 2.1. É possível observar que nenhum desses *frameworks* permite a realização de operações multidomínios ou a conexão com dispositivos da IoT utilizando protocolos convencionais da SG, como o CoAP [69]. Ou seja, não permite o compartilhamento de recursos protegidos entre aplicações de outros domínios (externo).

2.6. Discussão

O termo gestão de identidades (IdM) apresentado na seção 2.1 está sendo substituído na literatura pelo termo gestão de identidades e acesso (IAM, *Identity and Access Management*).

[1] http://www.scadabr.com.br/

[2] http://openscada.org/

[3] https://sourceforge.net/projects/free-scada/

[4] http://www.pascalscada.com/

[5] https://sourceforge.net/projects/indigoscada/?source=directory

[6] https://szarp.org/en/

Essa alteração não é somente estética: o IAM tem a função de estabelecer o relacionamento entre uma pessoa e os recursos que deve acessar para exercer sua função de trabalho [3]. Dessa forma, o IAM disponibiliza identidades que são associadas a credenciais que, por sua vez, são utilizadas para controlar o acesso a recursos protegidos. Na literatura, ainda não existe um consenso em relação à utilização desse termo. Enquanto alguns autores consideram que os controles de acesso (RBAC, ABAC etc) não estão contemplados no IAM [30], esse trabalho considera que o IAM é responsável pelo controle de autenticação e de acesso, sendo condizente com a definição do NIST [3]. Dessa maneira, o IAM adota todas as funcionalidades de um IdM, além de integrar mecanismos de controle de acesso. É esperado que nos próximos anos existam especificações e padrões relacionados ao IAM. Como ainda não existem esses padrões, esse trabalho segue as taxonomias definidas pelo NIST [3].

A adoção de mecanismos de autenticação multifator no contexto de SG é essencial em virtude dos recentes ciberataques. Empregar mecanismos de autenticação robustos, baseados em biometria e georreferenciamento, tendem a reduzir a probabilidade de invasão ao sistema por parte de um adversário oriundo da Internet. Além disso, determinadas funções críticas devem ser acessadas a partir de um perímetro pré-definido, como uma sala de operações em uma usina nuclear. Entretanto, a segurança do processo de autenticação está diretamente relacionada à confiabilidade dos fatores de autenticação.

A autorização de acesso provida pelo OAuth limita o acesso a um recurso protegido, porém não suporta políticas para determinação de operações nos recursos. Dessa forma, não é possível obter um controle de granularidade fina, necessitando um controle de acesso para esse fim. Adotar um controle de acesso que combina variadas políticas de segurança, como atributos e papéis, amplia a flexibilidade na definição dos direitos e privilégios.

O controle de acesso baseado em papéis (RBAC) é amplamente utilizado em aplicações comerciais e dispõe de diversas vantagens em relação aos controles de acesso tradicionais (discricionário e obrigatório) [31]. Os papéis RBAC intermedeiam a associação entre o usuário e as permissões, na qual as restrições de acesso estão definidas. No controle de acesso baseado em atributo (ABAC) não ocorre a associação de permissões a sujeitos ou papéis no sistema, porém ocorre a associação de permissões a atributos. A avaliação de políticas é efetuada baseada nos atributos, em tempo real, constituindo-se como uma vantagem em relação ao RBAC. Entretanto, o ABAC pode adotar os papéis do RBAC, considerando que os papéis são atributos de um usuário, ou seja, como não são modelos conflitantes, podem ser utilizados juntos.

Existem padrões de segurança para SCADA, como a IEC 62351[1] que define que o controle de acesso deve ser baseado em papéis (RBAC) e deve adotar o princípio de privilégio mínimo. Além disso, diversos trabalhos propõem a implantação de IDS para SCADA. Os autores focaram na utilização de NIDS baseado em anomalia [32, 33, 34, 35 e 36] na qual os tipos de ataques utilizados eram baseados em ataques de negação de serviço, espionagem, *probing* e demais ataques presentes na KDD99 [37]. Esses trabalhos detectaram diversas vulnerabilidades e, consequentemente, ataques por utilizarem SCADA de segunda e terceira geração, que não foram concebidas para atuar na Internet. A utilização de IDS para a quarta geração de SCADA, que se encontra em construção, é um problema pouco explorado na literatura.

É relevante destacar que o guia de boas práticas e taxonomias de cibersegurança definido pelo NIST [2] define unicamente as diretrizes de segurança. No entanto, não determina especificações ou mecanismos implementáveis para garantia da segurança dos elementos da SG. Implantar um mecanismo de IAM que realiza a gestão das identidades, das autenticações e do acesso, otimiza o processo administrativo de gerenciar as políticas de segurança. Além disso, permite a federação de identidades, autenticação única e rastreabilidade dos dispositivos acessados.

[1] https://webstore.iec.ch/publication/6905

Capítulo 3

Trabalhos Relacionados

A cibersegurança da SG é um desafio recorrente na literatura. Entretanto, de maneira geral os trabalhos apresentam desafios e tendências na segurança da SG [2, 8 e 38]. Esses, são trabalhos fundamentais para definir os padrões, procedimentos, especificações e melhores práticas para proteger a SG. Contudo, é essencial a existência de trabalhos que abordem definições de especificações ou mecanismos implementáveis para garantir a segurança da SG. A literatura carece de trabalhos dessa natureza para SG ou para os nomes correlatos. Os nomes correlatos encontrados na literatura são [25]: Sistemas Ciberfísicos (CPS, *Cyber Physical Systems*), Sistemas de Controle Distribuídos (DCS, *Distributed Control System*), Sistema de Controle de Rede (NCS, *Networked Control System*), Rede de Sensores e Atuadores (SAN, *Sensor Actuator Network*) ou Rede de Sensores Wireless Industriais (WISN, *Wireless Industrial Sensor Network*). As funções em comum são de aquisição e controle das informações.

A literatura apresenta inúmeros trabalhos relacionados à autenticação e autorização, que consideraram características da SG como: federado, distribuído, descentralizado e heterogêneo. Uma parcela desses trabalhos é discutida neste capítulo. Primeiramente será discutida a cibersegurança em *smart grid*. Posteriormente, as abordagens para segurança fim-a-fim no contexto da IoT. Na sequência, as abordagens de autenticação multifator e controle de acesso distribuído. Finalmente, ocorre a discussão do capítulo.

3.1. Cibersegurança em smart grid

A cibersegurança, de maneira geral, é uma disputa interminável entre atacantes e administradores de sistemas. A cibersegurança em *smart grid* (SG) não é diferente e é considerada um dos principais desafios da SG [39]. Vulnerabilidades podem permitir que um adversário obtenha acesso ao sistema central, obtendo dados privados, controlando os softwares

de monitoramento ou até mesmo alterando a carga da SG com o objetivo de desestabilizar a rede.

É importante ressaltar que os avanços tecnológicos da SG expõem à infraestrutura a novos desafios. Os controladores lógicos programáveis (CLP) estão sendo substituídos naturalmente por dispositivos da IoT. Como é o caso dos medidores inteligentes (SM), que são alvos atrativos para os atacantes (adversários), uma vez que as vulnerabilidades podem ser facilmente monetizadas [40]. Atacantes que comprometem os SMs podem manipular o consumo de energia ou até mesmo forjar a geração de energia para ganhar dinheiro. Além de ganhos financeiros, atacantes podem forjar informações de consumo energético em diversos SMs para fazer com que a empresa de energia elétrica tome decisões erradas em relação à capacidade da rede. Adicionalmente, é habitual que empresas de energia elétrica disponham de milhares de SM instalados em seus consumidores, que são controlados por alguns CS. Dessa maneira, caso o atacante obtenha o controle do CS, poderia interromper o fornecimento de energia de toda uma região [41].

Para aumentar a segurança dos elementos da SG, em um trabalho anterior [42] abordei a segurança física e lógica dos SM. Nesse trabalho, os autores propuseram mecanismos de violações físicas baseados em sensores e acelerômetros, além de um mecanismo de gestão de identidades para garantir a segurança lógica. O trabalho de Berthier et al. [43] apresentou a necessidade de um sistema de monitoramento e detecção de intrusão no contexto de SG. Entretanto, ressaltou a dificuldade de detectar a grande diversidade de ameaças e o alto custo de desenvolvimento de uma solução.

O trabalho de Lu et al. [44] apresentou contramedidas de ataques contra a integridade e confidencialidade das informações utilizando protocolos de autenticação e detecção de intrusão. O trabalho de Fang [1] apresentou que a interoperabilidade entre sistemas criptográficos na SG é um desafio, uma vez que é necessário um esquema de troca de chaves robusto para a troca de informações entre diferentes sistemas.

3.2. Segurança fim-a-fim no contexto de IoT

A IoT está presente em diversas áreas tais como: energia, saúde, transporte, automação industrial e residencial etc. Além das funcionalidades específicas ao contexto, um dispositivo da IoT tipicamente é capaz de informar ao fabricante que está com problemas técnicos. Assim,

o fabricante deverá acessar remotamente o dispositivo para reparar ou atualizar o *firmware*. Nesse caso, é comum que o fabricante necessite acessar centenas ou milhares de dispositivos remotamente, necessitando a autenticação individual em cada dispositivo. Para atender essa demanda, os trabalhos explorados na literatura isolam o dispositivo por questões de segurança [44 e 45]. Adicionalmente, esses trabalhos indicam que a comunicação e o acesso ao dispositivo devem ser controlados.

A autenticação e autorização no contexto da IoT é um desafio significativo discutido na literatura, uma vez que, devido às restrições de poder computacional, mecanismos tradicionais de segurança não podem ser aplicados na IoT. É habitual encontrar soluções de mercado que utilizam uma senha única para todos os dispositivos. Essa abordagem tem como vantagem o baixo consumo de recursos. Entretanto, uma vez que a senha é descoberta, todos os dispositivos tornam-se vulneráveis [46, 47 e 48]. Ademais, atualizar a senha ou chave de todos os dispositivos não é uma tarefa trivial [45], e esse desafio desencadeou diversas propostas para tratar essa demanda.

O trabalho de Liu et al. [49] define métodos de autenticação e controle de acesso para IoT. A proposta do trabalho utiliza o OpenID e o RBAC no contexto da IoT. Entretanto, apesar do trabalho selecionar mecanismos de segurança popularmente conhecidos, os autores não mencionam como realizar a integração no contexto da IoT, que possui diversas restrições de recursos computacionais. Além disso, a confidencialidade ou SSO não foram tratados.

O trabalho de Chin et al. [50], apresenta uma plataforma de autenticação baseada em M2M (*Machine-to-machine)* para redes inteligentes. A proposta é baseada em assinaturas digitais, nas quais a troca de chaves entre os contextos é realizada no concentrador. Isso é um problema, pois gera um ponto de falha único. O trabalho de Saxena [51] propôs um protocolo de autenticação para redes inteligentes que utiliza criptografia simétrica e assimétrica para garantia de autenticação e confidencialidade no período de comunicação. Apesar do trabalho considerar que a proposta é leve (*lightweight*), a mesma utiliza operações de criptografia assimétrica, que não são práticas recomendadas para dispositivos da IoT [42].

Referente ao tema de autenticação e autorização para IoT, esse autor que escreve, publicou dois trabalhos científicos. O primeiro trabalho [52] apresenta um esquema de autenticação baseada em chaves para IoT, que permite a autenticação única. O segundo trabalho [42] apresenta uma solução integral (*software*, *hardware* e rede) para proteger um medidor inteligente de energia, que é um dispositivo IoT. A proposta apresenta técnicas de prevenção

de adulteração física (*anti-tampering*), proteção da integridade multinível e proteção da comunicação. A proteção da comunicação utiliza um esquema baseado em chaves, o que garante a segurança fim-a-fim no contexto de redes inteligentes.

Vale ressaltar que, para ampliar o nível de confiabilidade do processo de autenticação, é fortemente recomendada a adoção do serviço de autenticação multifator, o que é discutido na próxima seção.

3.3. Autenticação Multifator

A autenticação multifator, que combina diversas técnicas de autenticação para reduzir a probabilidade de invasão, é desejada na maioria dos sistemas, devido aos seus benefícios e facilidade de implantação. Essa combinação de técnica é consolidada e disponibilizada por diversos serviços na web, que utilizam biometria, senha descartável por e-mail ou SMS, leitura óptica de QRCode etc. Dessa maneira, pesquisas recentes estão focadas em novos fatores de autenticação para ampliar a confiabilidade do sistema. Assim, o fator baseado em georreferenciamento do usuário encontra-se em destaque. O principal desafio em autenticar um usuário baseado em seu georreferenciamento está relacionado à confiabilidade dessa informação. O georreferenciamento é facilmente forjável, principalmente quando depende exclusivamente do GPS de um *smartphone*. Dessa maneira, diversos autores propõem técnicas para melhorar a confiabilidade dessa informação [53, 54, 55 e 56].

Técnicas rudimentares estimam a longitude e latitude a partir do endereço IP do *smartphone*. O fato de o endereço IP ser facilmente forjável implica na baixa confiabilidade no emprego dessa técnica. O trabalho de Zhang et al. [54] utiliza o georreferenciamento do *smartphone* para realizar a autenticação e a autorização. Para garantir a confiabilidade da informação, os autores empregam o endereço MAC (*Media Access Control*) do *access point*. Assim, quando o *smartphone* se autentica no *access point*, o *smartphone* registra o endereço MAC do *access point*. Essa técnica é interessante, pois utiliza o conceito de proximidade com um dispositivo que é fixo.

O trabalho de Jaros [53] apresenta duas técnicas de autenticação baseadas em geolocalização. A primeira é semelhante à abordagem de Zhang et al. [54], já a segunda considera um dispositivo âncora que constantemente se comunica com o *smartphone* utilizando um protocolo de baixo alcance (e.g. *bluetooth)*. O trabalho de Camenisch [55] considera as antenas das operadoras de rede móvel para determinar a localização do usuário. A proposta

objetiva a troca de mensagens entre *smartphone* e antena, baseando-se em um protocolo que adota o conceito de uma função de autenticação previamente combinada sobre o *nonce*. O trabalho não esclarece o comportamento em um cenário onde a rede móvel não existe.

Finalmente, o trabalho de Choi [56] apresenta um método de autenticação georreferenciada baseada em *Beacons*. *Beacons* são dispositivos da IoT que são fixados em uma localização e que transmitem dados de forma contínua através da rede sem fio [57]. A proposta define um protocolo de troca de mensagens baseado no conceito de OTP para melhorar a autenticação do usuário. A utilização dessas diferentes estratégias de maneira combinada amplia a confiabilidade da autenticação baseada em proximidade.

3.4. Controle de Acesso Distribuído

Desenvolver o controle de acesso distribuído é um desafio amplamente discutido na literatura. O trabalho de Lee e Luedemann [58] investigou uma série de trabalhos que abordam autorização multidomínios de maneira geral. Os autores identificaram algumas características desejadas que são discutidas a seguir.

- Autonomia: os domínios são heterogêneos em termos de infraestrutura de TI, sistemas e políticas. Dessa forma, todos os domínios almejam manter sua autonomia para definir o acesso a seus recursos protegidos.
- Privacidade: as informações privadas de um usuário não devem, em hipótese alguma, ser acessadas por pessoas não autorizadas. Em inúmeros casos, apenas os proprietários de um determinado recurso devem ter o controle sobre o mesmo.
- Representatividade: na maioria dos trabalhos existe de alguma forma, seja através de um usuário ou um papel global, um ponto de referência para realizar a gestão dos acessos multidomínios.
- Descentralização: é necessário um mecanismo descentralizado para gerir as políticas de segurança.
- Escalabilidade: o sistema de autorização necessita ser escalável e de simples instalação/configuração. Isso se justifica porque o objetivo é facilitar o procedimento de adesão no caso de novas organizações participantes.

Tradicionalmente, o RBAC foi concebido para controlar um único domínio. Assim, diversos trabalhos objetivaram resolver o desafio de utilizar o RBAC em ambientes multidomínios, baseados em duas possíveis abordagens para controlar o acesso: centralizado e descentralizado.

3.4.1. Abordagem centralizada

A abordagem centralizada em ambientes multidomínios tem o objetivo de criar um domínio global responsável por integrar os papéis de todos os domínios envolvidos. Os principais desafios envolvidos nessa abordagem estão relacionados ao mapeamento de papéis, à herança entre papéis e à consistência dos domínios individuais.

A principal referência relacionada a essa abordagem foi desenvolvida por Shafiq e Bertino em 2005 [11]. Eles propuseram um algoritmo que centraliza os papéis e os mapeia globalmente através de uma operação de junção *(merging)* de papéis. Essa operação verifica em cada papel se existe a intersecção de permissões com algum papel de outro domínio. A solução proposta por Shafiq et al. [11] permite interoperar papéis entre domínios. Dessa forma, tornou viável realizar hierarquia de papéis entre domínios. A hierarquia de papéis pode causar inconsistências no modelo RBAC, como uma herança cíclica de papéis. A herança cíclica ocorre quando um papel sênior se torna filho de um papel júnior. Para resolver isso, os autores desenvolveram um método de verificação de conflitos. Essa operação de junção centralizada dos papéis exige um custo computacional elevado, uma vez que é essencial comparar e mapear todos os papéis entre si. Além disso, em caso de atualização de papéis, é necessário, na maioria das vezes, refazer todas as junções. A solução necessita de uma área de compartilhamento dos recursos para que esses sejam visíveis a todos os domínios. Por fim, existe um problema ligado à semântica das permissões, pois as ações que uma permissão pode executar em um determinado domínio podem ser diferentes em outro domínio, ainda que tenham o mesmo nome.

Na proposta de Qi Li et al. [59] foi adotado o conceito de virtualização de papéis sob demanda. O administrador RBAC define os papéis que deseja utilizar em multidomínios e então é criado um *link* de referência desses papéis em um domínio global. Essa abordagem é uma evolução do trabalho de Shafiq et al. [11] e traz como principal vantagem a escolha de papéis sob demanda, sem a necessidade de incluir todos os papéis no domínio global. Entretanto, ainda

se mantêm como problemas a centralização dos papéis em um domínio global, a necessidade de compartilhamento de recursos e a semântica das permissões.

O trabalho de Mouliswaran [60] apresenta um modelo que utiliza análise de conceito formal (FCA, *Formal Concept Analysis*) para representar as permissões de acesso que um papel possui em diferentes domínios. Dessa forma, o autor considera que existem papéis globais (multidomínios) que possuem permissões locais em cada um dos domínios. Essas permissões são armazenadas em uma matriz, onde os papéis globais são as linhas e as permissões são as colunas. A solução não é escalável, visto que cada operação sobre determinado recurso é uma coluna da matriz, se considerar que uma organização possui diversos recursos e operações, torna-se inviável sua aplicação.

O trabalho de She et al. [61] apresenta uma solução que integra a proposta de Shafiq e Bertino (2005) com um mecanismo de proveniência de dados no contexto de *web services*. O objetivo do trabalho é permitir a rastreabilidade das informações durante a orquestração de serviços. Para isso, representam um recurso como uma estrutura, onde cada recurso possui um conjunto de atributos que é utilizado durante a construção do histórico de serviços utilizados. Além disso, consideram cada serviço como uma entidade, onde são definidas políticas de controle de acesso. Essa proposta é adequada para contextos onde as informações são trafegadas por diversos serviços e a confidencialidade é necessária.

3.4.2. Abordagem descentralizada

A abordagem descentralizada utiliza os mecanismos dos próprios domínios para fazer a integração de papéis, evitando a necessidade de criar um domínio. A principal referência relacionada à abordagem descentralizada é o trabalho de Freudenthal et al. [10]. O trabalho de Freudenthal et al. [10] adotou um repositório de credenciais denominado *Wallet* que armazena as delegações de autorizações através de papéis. A arquitetura utiliza-se de um monitor de provas responsável por avaliar as delegações. Cada domínio possui uma *Wallet* e na medida do possível todas as *Wallets* encontram-se sincronizadas entre si através de um serviço *publish*/*subscribe*. No caso de uma delegação não existir na *Wallet* local, é aplicado um serviço de descoberta de localização para a *Wallet* de origem do recurso envolvido na delegação, e se a delegação for encontrada, será inserida na *Wallet* local para fazer *cache*. Isso pode tornar o

processo de busca intenso e lento, pois um conjunto de delegações necessárias para autorizar um papel pode encontrar-se em várias *Wallets*.

O trabalho de Shehab e Bertino em 2005 [11] apresentou uma solução descentralizada que não necessita de uma visão global dos papéis. O trabalho tem três premissas: (i) cada domínio possui apenas suas próprias políticas de segurança, incluindo as políticas que permitem operações multidomínios; (ii) cada domínio é responsável por realizar as próprias decisões de acesso; (iii) os domínios estão dispostos a colaborar através de trocas de mensagens. A ideia do trabalho é, ao invés de permitir a hierarquia de papéis, o *framework* proposto concatena os papéis e domínios formando um caminho (*path*). No momento da avaliação, o monitor de referências verifica a autenticidade do caminho e em seguida a autorização. Caso esse caminho resulte em um ciclo, o acesso é negado. Além disso, o trabalho possui um mecanismo de descoberta de caminhos de papéis sob demanda.

O trabalho de Avita et al. [62] apresentou uma proposta de implementação do modelo RBAC com a utilização de ontologias. A complexidade da implementação do RBAC está relacionada à forma de representar os papéis. Os autores representaram cada papel como uma classe OWL (*Ontology Web Language*). Dessa forma, existe uma classe principal denominada "Papel" que dispõe das características e atributos pertencentes a todos os papéis, e os demais papéis são subclasses da classe principal. Assim, é possível representar hierarquias de papéis através das heranças das classes. Os autores propõem uma delegação de papéis, no qual um usuário que possui um determinando papel tem a possibilidade de delegar o papel, juntamente com as permissões relacionadas, a um outro usuário. Esse comportamento é um ponto fraco do trabalho, pois o administrador do sistema perde a autonomia da gestão de papéis. A partir do momento em que um usuário recebe a delegação de um papel, ele recebe a permissão para delegar suas permissões a outros usuários. Isso prejudica a visão gerencial do administrador sobre os papéis, que é uma das principais vantagens do RBAC.

Em uma pesquisa anterior [63], este autor que escreve desenvolveu um modelo de ativação de papéis multidomínios que considera diferentes semânticas de um papel, permitindo a ativação única de papel. A proposta mantém a autonomia do administrador de cada domínio, permitindo que ele mesmo defina as permissões associadas a cada papel. Isso possibilita ao administrador do domínio local definir políticas XACML referenciando papéis de outros domínios. Quando um usuário acessa a um domínio remoto e requisita acesso a um recurso protegido, são considerados os papéis ativos em seu domínio de origem. Se os papéis ativos no domínio de origem não forem conflitantes com os papéis do domínio remoto, o RBAC importa

a ativação do papel em questão e o ativa no domínio remoto. Como resultado, um usuário acessa domínios remotos de maneira transparente, sem precisar ativar papéis que estão em uso em seu domínio de origem, assim como acontece com SSO para autenticação. O ponto fraco desse trabalho é o mapeamento direto entre os papéis de diferentes domínios nas políticas XACML, pois, se o volume de papéis e domínios forem elevados, a proposta é pouco amigável para o administrador configurar.

3.5. Discussão

Esse capítulo discutiu os trabalhos relacionados a proposta. Apesar da literatura apresentar diversos trabalhos que abordam a cibersegurança na SG, poucos trabalhos discutem especificações ou mecanismos implementáveis para garantir a cibersegurança. Entretanto, a literatura apresenta diversos trabalhos que abordam autenticação e autorização em um contexto geral. Contudo, a SG apresenta características peculiares que diferem de sistemas convencionais. Em decorrência dessas características e dos frequentes ciberataques na SG, esse trabalho elencou algumas propriedades esperadas em mecanismos de cibersegurança na SG. A Tabela 3.1 apresenta uma comparação entre os principais trabalhos relacionados considerando as propriedades esperadas dos mecanismos de segurança da SG. Observe que alguns trabalhos adotam o controle de acesso com granularidade fina, utilizando uma abordagem centralizada ou descentralizada para atender multidomínios. Dentre esses trabalhos, nenhum adotou a autenticação e o acesso multifator para garantir a integridade do mecanismo caso seja parcialmente comprometido. Finalmente, não foram encontrados trabalhos que abordem todas as propriedades definidas na tabela abaixo. Esta proposta atende todas as propriedades relacionadas na Tabela 3.1, o que será discutido no próximo capítulo.

Trabalhos	Políticas com granularidade fina	Centralizado / Descentralizado	Autenticação Multifator	Controle Criptográfico	Acesso Multifator	Adota padrões	Avaliação de Segurança	Protegido por IDS
Shafiq [11]	Sim	Centralizado	Não	Não	Não	Sim	Sim	Não
Q. Li [59]	Sim	Centralizado	Não	Não	Não	Sim	Sim	Não
Freudental [10]	Sim	Descentralizado	Não	Não	Não	Não	Não	Não
She [61]	Sim	Descentralizado	Não	Não	Não	Não	Sim	Não
Abreu [63]	Sim	Descentralizado	Não	Não	Não	Sim	Sim	Não
Witkovski [52]	Não	Descentralizado	Não	Sim	Não	Sim	Não	Não
Abreu [42]	Não	Descentralizado	Não	Sim	Não	Sim	Sim	Não

Tabela 3.1. Comparação entre os trabalhos relacionados.

Capítulo 4

Gestão de Identidade e Acesso Multifator

Os incidentes de cibersegurança relatados evidenciam que ataques às *smart grids* (SG) podem causar prejuízos incalculáveis na vida das pessoas e/ou em organizações. O desafio em desenvolver soluções de cibersegurança para SG está relacionado à heterogeneidade presente em sua arquitetura, que pode conter milhares de sensores, atuadores, agregadores, controladores e outros componentes de rede que são acessados remotamente [42]. Algumas características intrínsecas da SG amplificam esse desafio: (i) o acesso remoto aos dispositivos deve exigir autenticação do usuário, porém autenticar individualmente milhares de dispositivos é impraticável; (ii) o compartilhamento de informações e de recursos protegidos com usuários de outros domínios federados deve ser controlados; (iii) a comunicação entre os elementos (sensores, atuadores, controladores etc.) que compõem a arquitetura da SG deve ser protegida; (iv) a SG e seus mecanismos de segurança são frequentemente alvo de ataques.

O trabalho de Fang et al. [1] apresentou uma revisão sistemática sobre a cibersegurança em SG. Os autores apresentaram os principais desafios de cibersegurança e privacidade existentes, e ressaltaram a necessidade do desenvolvimento de soluções robustas para garantir a cibersegurança de todos os elementos da SG, inclusive dos dispositivos (sensores e medidores inteligentes de energia) que estão fora do perímetro da empresa de energia elétrica.

Este trabalho propõe um mecanismo de gestão de identidade e acesso multifator (MIAM, *Multi-factor Identity and Access Management*), que objetiva impor uma proteção adicional na camada de autenticação e autorização na SG para evitar o acesso remoto não autorizado aos elementos da SG. O propósito de utilizar multifatores é combinar diferentes mecanismos de autenticação e acesso para evitar violações do sistema. Além disso, esse trabalho adota controles de admissão que emitem endossos para qualificar os usuários da SG a utilizarem o MIAM.

O mecanismo de gestão de identidade (IdM) proposto combina diversos fatores de autenticação tradicionais com um controle de admissão baseado na localização do dispositivo do usuário. Esse controle de admissão considera um dispositivo âncora que é colocado em uma localização física em um ambiente controlado para garantir que o usuário (operador, administrador etc.) está próximo do dispositivo. Assim, o dispositivo âncora emite uma credencial de proximidade que é utilizada como requisito para acessar ao mecanismo de autenticação. Dessa maneira, esse mecanismo evita que alguém na internet, que não esteja autorizado a acessar a SG, comprometa o mecanismo de autenticação.

Além disso, o adversário não consegue acessar um sistema da SG comprometendo um único fator (e.g. explorando uma vulnerabilidade do controle de acesso). Pois, a gestão de acesso (AM, *Access Management*) não é composta apenas de um controle de acesso multidomínios, mas também de um controle de admissão baseado em criptografia e quórum. Dessa maneira, antes de um usuário receber autorização para realizar operações nos recursos da SG, o mesmo deve receber o endosso desses controles de admissão.

Devido à SG ser frequentemente atacada, o MIAM será protegido por um mecanismo de detecção de intrusão inteligente (iID, *intelligent Intrusion Detection*) que aplica técnicas de aprendizagem de máquina baseado no modelo de adversário e nas características do MIAM. O iID será treinado com os logs gerados por um processo sistemático de *pentest*.

Finalmente, o MIAM foi concebido para ser compatível com dispositivos da IoT (e.g. medidores inteligentes de energia). Essa característica é desafiadora pois esses dispositivos geralmente possuem restrições computacionais que impedem a adoção de protocolos tradicionais de comunicação e segurança. Essa proposta permite que um usuário autenticado no MIAM transporte suas credenciais para o contexto da IoT, mantendo a autenticidade e confidencialidade na comunicação.

A visão geral da proposta é apresentada na Figura 4.1, na qual um usuário, local ou remoto, pode acessar um recurso protegido, desde que o MIAM permita esse acesso. A decisão de acesso realizada pelo MIAM é alimentada por dois mecanismos multifatores, um de autenticação e outro de acesso. Assim, o MIAM é composto pelo IdM e pelo AM. O objetivo de utilizar multifatores é manter-se íntegro mesmo que uma entidade específica (fator) seja comprometida. Para mitigar a possibilidade de comprometimento de uma entidade foi concebido um iID para monitorar exclusivamente esse mecanismo de IAM. As subseções a seguir descrevem os mecanismos de segurança propostos.

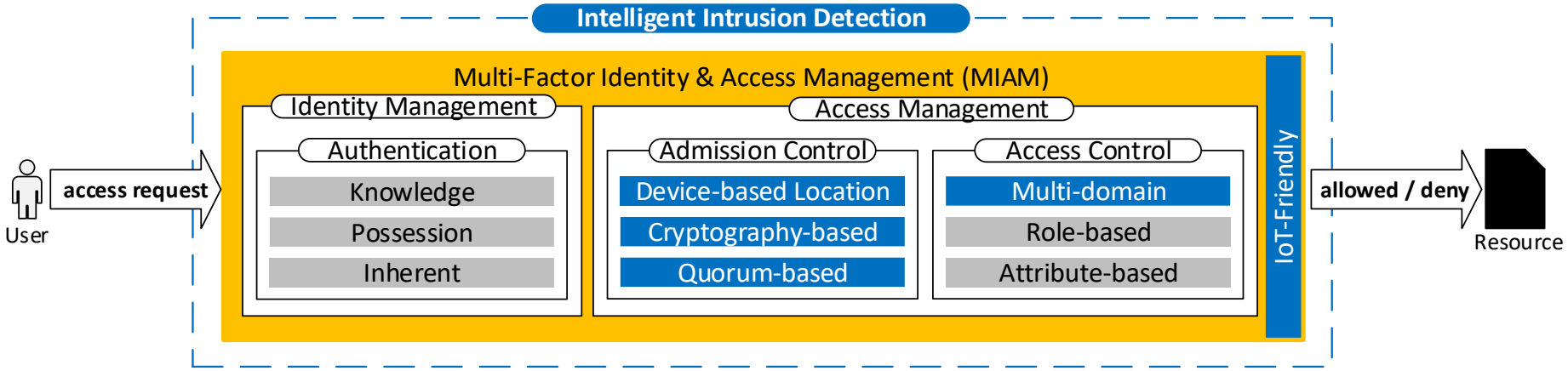

Figura 4.1. Modelo do MIAM.

No próximo capitulo, será discutido um estudo de caso que aplica detalhadamente esses mecanismos de segurança no contexto de SG.

4.1. Gestão de Acesso (AM, *Access Management*)

O AM é responsável por gerenciar o acesso (local e remoto) para proteger os recursos da SG. O AM é baseado na combinação de um controle de acesso multidomínios (M_dAC, *Multi-domain Access Control*) com um controle de admissão baseado em criptografia (CA_dC, *Cryptographic Admission Control*) e quórum (QA_dC, *Quorum Admission Control*). Essa combinação aumenta a robustez do mecanismo de segurança, já que utiliza três mecanismos independentes para autorizar o acesso aos recursos protegidos. Além disso, essa proposta possui um controle de admissão baseado na localização do dispositivo do usuário (DLA_dC, *Device-based Location Admission Control*) que mitiga a possibilidade de um adversário oriundo da internet realizar um acesso não autorizado, pois o adversário não tem posse do token de proximidade emitida pelo DLA_dC. As subseções seguintes exploram as principais características do MIAM (Figura 4.2).

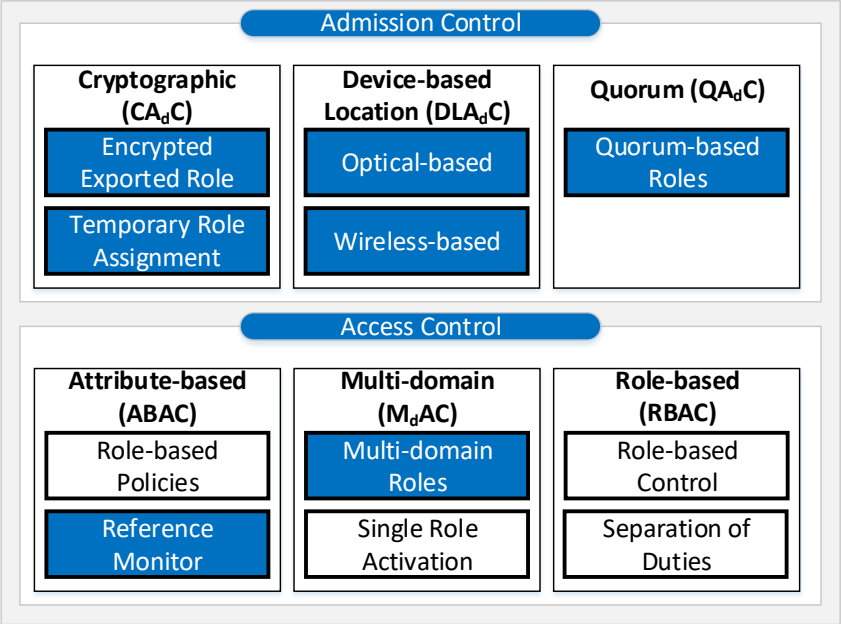

Figura 4.2. Gestão de Acesso.

4.1.1. Políticas baseadas em papéis

As políticas de controle de acesso podem utilizar o papel ativo do usuário. O papel do usuário representa o direito que um usuário tem dentro de uma organização. Os papéis são baseados no modelo RBAC [21]. Assim, é necessário adquirir os papéis ativos de um determinado usuário no controlador de papéis (RC, *RBAC Controller*) (subseção 2.3.1).

4.1.2. Políticas multidomínios

A federação de identidades é uma característica natural na SG. Frequentemente, as empresas de energia elétrica necessitam que parceiros, subsidiários, fornecedores ou clientes acessem recursos internos através de operações multidomínios. Para isso, é necessário a existência de relação de confiança entre esses domínios. O MIAM considera a federação de identidades entre diferentes domínios (organizações) da SG. Esse comportamento implica na necessidade do gerenciamento de identidade e acesso, que tem como principal objetivo validar a identidade do usuário e verificar se o usuário tem autorização de acesso a determinado serviço.

O M_dAC permite a realização de operações multidomínios, desde que os domínios façam parte da federação. As operações multidomínios são definidas pelos próprios detentores de papéis, que disponibilizam seus papéis de maneira flexível para acesso de outros domínios

da federação. Assim, é mantida a autonomia do domínio em definir os direitos associados a cada recurso.

4.1.3. Ativação única de papéis

A ativação única de papéis (SRA, *Single Role Activation*), realizada pelo M_dAC, permite a importação de papéis ativos de um domínio local para outros domínios federados. Dessa maneira, o usuário não precisa ativar seus papéis em cada domínio que realiza operações multidomínios. Na prática, espera-se que um usuário que usa SRA acesse a outros domínios de maneira transparente, sem a necessidade de ativar seu papel nestes domínios, assim como acontece com SSO para autenticação.

4.1.4. Exportação de papéis cifrados

A exportação de papéis, realizada pelo CA_dC, permite que um usuário (proprietário) possuinte de um papel ativo disponibilize o seu acesso a recursos associados para outros usuários, locais ou remotos ao domínio. O proprietário pode escolher se deseja exportar parcialmente ou completamente as permissões desse papel. Esse procedimento aumenta a flexibilidade do proprietário de selecionar as permissões e recursos que serão disponibilizados. O papel exportado é vinculado ao domínio que será utilizado, sendo trafegado de maneira cifrada na chave pública do usuário que deseja acessar o papel. Observe que a exportação de papéis difere da delegação de autorização [10], pois não é possível realizar a disseminação descontrolada de papéis, permitindo que o administrador tenha visão clara sobre os papéis vinculados aos usuários.

4.1.5. Associação temporária de papéis

Para que um determinado usuário consuma um papel exportado, faz-se necessário que esse usuário esteja associado ao papel. Essa associação é temporária (curta duração), realizada

utilizando técnicas de criptografia e assinatura digital. Assim, o usuário precisa possuir permissões no sistema e possuir a chave privada para utilizar um papel.

4.1.6. Papéis baseados em quórum

Baseado no conceito de separações de deveres (seção 2.3), o quórum de papéis, realizado pelo QA_dC, determina que operações críticas devem ser realizadas por duas ou mais pessoas. Esse procedimento evita que uma pessoa individualmente comprometa o sistema. Dessa forma, a política possui uma regra que referencia a quantidade necessária de usuários com papéis ativos. Assim, no momento da avaliação, o QA_dC requisita o endosso dos usuários relacionados.

4.1.7. Token de proximidade por wireless

O DLA_dC é responsável por emitir credenciais de proximidade que são utilizadas para admitir o usuário para acessar o MIAM. Essa emissão de credenciais pode ser realizada de duas formas, através de wireless ou por leitura óptica. A emissão de tokens de proximidade por wireless, que é um canal de comunicação de curto alcance, é realizada por um dispositivo acoplado em uma localização fixa em um ambiente controlado, denominado de Beacon. O Beacon é um dispositivo que continuamente transmite o token de proximidade utilizando a rede *Bluetooth Low Energy* (BLE). Para mitigar a possibilidade de o adversário forjar o valor transmitido foi adotado o protocolo Eddystone [64]. Esse protocolo, desenvolvido pela Google, possui um modo de comunicação chamado de EID (*Ephemeral Identifier*) que permite transmitir, em broadcast na BLE, um identificador modificado em um determinado intervalo de tempo.

4.1.8. Token de proximidade por leitura óptica

Esse método disponibiliza o token de proximidade de forma gráfica através de um código QRCode exibido através de um display LCD fixado fisicamente em um determinado local. Esse dispositivo LCD pode ser qualquer equipamento com conectividade a Internet e que possua um display, por exemplo: *Personal Computer* (PC), *smartphone*, *tablet* etc. A credencial

disponibilizada através do QRCode é gerada por um algoritmo TOTP (*Time-based One Time Password*). O TOTP [65] é um método comumente encontrado em mecanismos de autenticação multifator que necessitam de uma credencial de curta duração e dinâmica.

4.1.9. Token de Acesso

O AM é responsável por emitir *tokens de acesso* que autorizam os usuários a acessar serviços protegidos. O *token de acesso* contém um escopo de acesso que permite definir a visibilidade do recurso e seu tempo de vida (duração). Quando o tempo de duração é expirado, o *token de acesso* é invalidado e o AM deve emitir um novo. Portando, o *token de acesso* garante que apenas usuários autenticados e autorizados podem ser admitidos no sistema.

4.2. Gestão de Identidades (IdM)

A autenticação multifator (MFA, *Multi-Factor Authentication*) é o principal componente do IdM, responsável pela autenticação do usuário. Essa entidade provê cinco métodos de autenticação (Figura 4.3), agrupados da seguinte maneira:

- **Fatores de Conhecimento**: (i) usuário e senha; (ii) número de identificação pessoal (PIN);
- **Fatores de posse**: (iii) senha descartável (TOTP); (iv) leitor de SQRL;
- **Fatores inerentes**: (v) leitura de impressão digital;

Esses métodos de autenticação foram escolhidos pela facilidade de utilização no contexto de SG. Entretanto, é possível adicionar facilmente outros fatores de autenticação como um *plugin*.

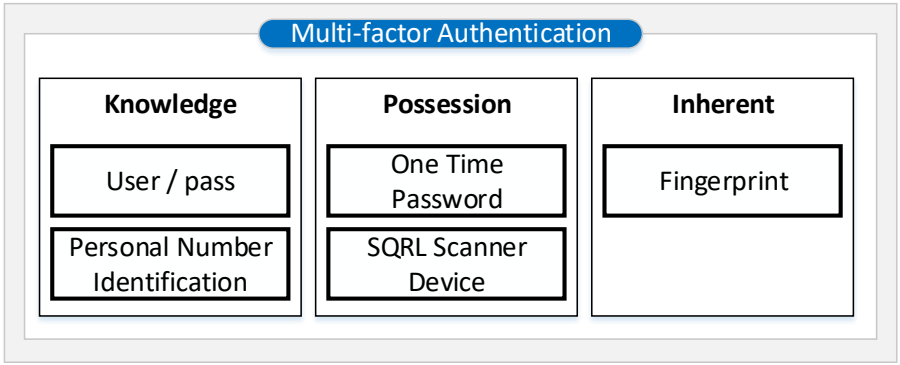

Figura 4.3. Gestão de Identidades.

Para que o usuário obtenha êxito ao ser autenticado é necessário que o dispositivo utilizado por ele seja admitido pelo DLA_dC. Ou seja, o MFA tem como requisito que o usuário esteja fisicamente próximo do dispositivo âncora onde está localizado (e.g. sala de operações). A comunicação entre o dispositivo âncora e o dispositivo do usuário é multicanal (óptico ou wireless), para evitar que um canal comprometido comprometa todo o sistema de autenticação. Adotar esse requisito evita que um adversário da Internet controle a infraestrutura crítica explorando uma vulnerabilidade do MFA, porque precisa da credencial de proximidade emitida pelo dispositivo âncora.

Esses métodos de autenticação são consumidos de maneira flexível pelos demais mecanismos de segurança na arquitetura. Através da combinação dos métodos de autenticação, é possível estabelecer o nível de autenticação do usuário. Esse é um atributo que indica em quais métodos (fatores) o usuário está autenticado. As políticas de acesso podem definir o nível necessário de autenticação do usuário para acessar um determinado recurso. Logo, o MFA é responsável por gerenciar esse atributo e mantê-lo atualizado.

Um *token de identidade* é emitido pelo MFA após o usuário estar autenticado. Esse é uma cadeia de caracteres (*string*) assinada digitalmente pelo MFA, que contém informações relativas à autenticação do usuário. Os principais atributos do *token de identidade* estão identificados na tabela abaixo:

Atributo	**Descrição**
sub	Identificador do usuário, deve ser único dentro do domínio.
iss	Identificador do MIAM, deve ser único entre a federação.
aud	Identificador do provedor de serviço (SP, *Service Provider*) que utilizará o *token de identidade*. Deve ser único dentro do domínio.
exp	Data e hora da expiração do *token de identidade*. Após essa data e hora, o *token de identidade* não deve ser aceito.
iat	Data e hora da emissão do *token de identidade*.
amr	Método de autenticação utilizado durante a autenticação. Esse atributo pode conter uma lista de valores, que são os possíveis fatores.

kpu	Chave pública do usuário. Cada sujeito possui um par de chaves pública e privada, sendo que a chave pública fica armazenada no CA_dC.

Tabela 4.1. Atributos do *token de identidade.*

4.3. Detecção de Intrusão Inteligente (iID)

A iID monitora apenas o MIAM, utilizando a abordagem de detecção baseada em anomalias com o propósito de identificar novos ataques. A iID monitora o tráfego de rede e os logs emitidos pelo MIAM. Assim, o iID é considerado híbrido devido aos seus dados de auditoria serem coletados a partir da rede (NIDS) e da aplicação (HIDS). A Figura 4.4 apresenta o modelo de detecção de intrusão, responsável por determinar se um evento é normal ou ataque.

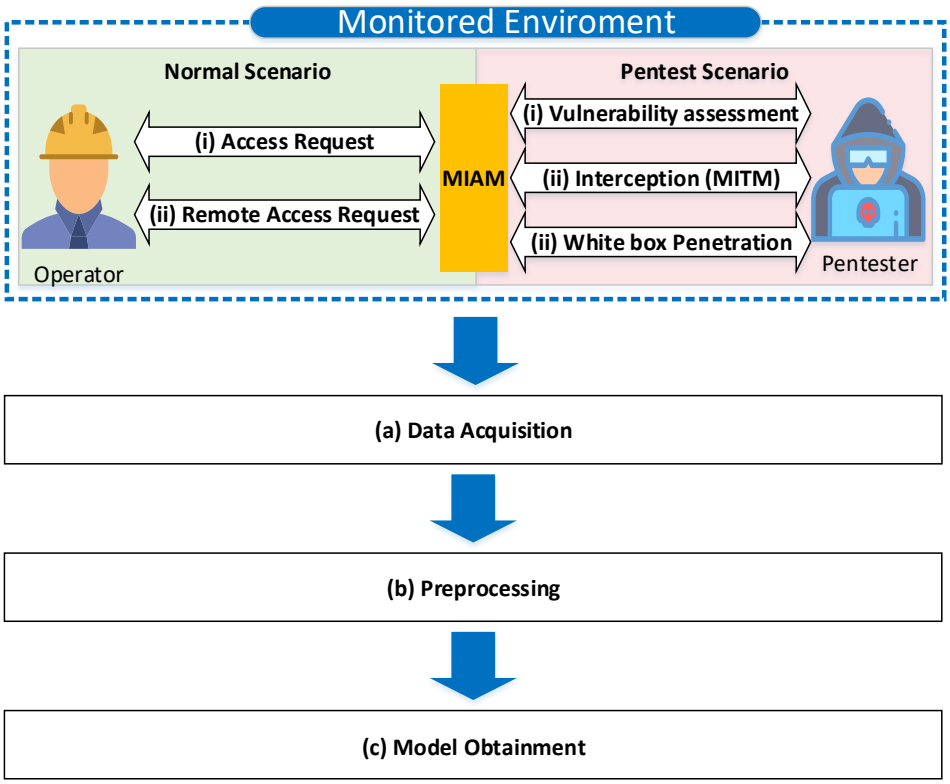

Figura 4.4. Mecanismo de detecção de intrusão inteligente.

Para treinar o modelo, dois cenários foram construídos, denominados de cenário normal e de *pentest*. O cenário normal possui processos de carga de trabalho (*workload*) que imitam os procedimentos realizados pelos usuários em sua rotina de trabalho. Esses processos possuem comportamentos imprevisíveis, realizando requisições pseudoaleatórias, porém válidas e reais, em um conjunto de serviços protegidos, papéis, operações e recursos. É

importante ressaltar que esses processos podem requisitar operações inválidas ou não autorizadas, que devem ser negadas pelos mecanismos de segurança.

O cenário de *pentest* tem dois principais objetivos: (i) identificar vulnerabilidades na atual proposta; (ii) treinar o modelo de detecção do iID. Para identificar potenciais vulnerabilidades na proposta, um processo de *pentest* foi definido. Esse processo tem o objetivo de utilizar ferramentas de *pentest* e o conhecimento de um profissional de *pentest* para identificar as vulnerabilidades, baseado no modelo do adversário (seção 6.2). Dessa maneira, com base no tráfego normal e no tráfego de ataque (gerado durante o *pentest*), é possível treinar o classificador do iID para detectar se um determinado evento é normal ou ataque.

4.4. IoT-Friendly

Para garantir a autenticidade e confidencialidade durante a comunicação dos dispositivos da IoT com os demais elementos presentes na arquitetura da SG, o MIAM possui duas abordagens que garantem a segurança fim-a-fim na comunicação entre essas entidades. O objetivo é permitir que um usuário autenticado no MIAM transporte suas credenciais para o contexto da IoT, mantendo a autenticidade e confidencialidade na comunicação.

A abordagem baseada em hierarquia de chaves (KH, *Key Hierarchy*) é baseada na ANSI X.9.17 [66] e possui dois níveis de criptografia para garantir a segurança na comunicação *unicast* fim-a-fim. Por outro lado, a abordagem OTP é baseada em senhas descartáveis, que são utilizadas para cifrar a comunicação *unicast* e *multicast*. Apesar dessas abordagens garantirem a segurança na comunicação, não são capazes de garantir políticas de controle de acesso fino nos recursos protegidos da IoT. Assim, o MIAM possui um módulo de controle de acesso leve (*lightweight*) adequado aos dispositivos da IoT que aproveitam das configurações da abordagem OTP para controlar o acesso dos usuários aos recursos protegidos da IoT.

4.5. Discussão

Esse capítulo dissertou os objetivos dos mecanismos de segurança propostos. O IdM tem como principal objetivo combinar fatores de autenticação com um controle de admissão baseado na localização do dispositivo do usuário. Essa combinação evita que um adversário oriundo da Internet comprometa o mecanismo de autenticação. O AM tem como principal

objetivo controlar o acesso, local ou remoto, a recursos protegidos. Para isso, combina um controle de acesso multidomínios com dois controles de admissões (criptográfico e quórum). Essa combinação tem como objetivo evitar que uma vulnerabilidade do sistema comprometa o controle de acesso, uma vez que existe a dependência do controle de admissão criptográfico. Ou seja, mesmo que o adversário comprometa o controle de acesso, o mesmo não possui a chave criptográfica necessária para ganhar autorização. O mesmo conceito é utilizado para o quórum, no qual é solicitada a assinatura digital dos responsáveis pelo recurso protegido antes de o usuário ganhar autorização para realizar uma operação no recurso.

Para proteger o MIAM, foi desenvolvido um mecanismo de detecção de intrusão inteligente construído a partir das características do MIAM. Esse mecanismo foi treinado a partir de técnicas de aprendizagem de máquina baseado no modelo de adversário, tráfego de rede (normal e ataque) e logs de auditoria. Finalmente, para conceber uma solução integral de autenticação e acesso para SG, foram propostas duas abordagens de segurança fim-a-fim na comunicação entre os dispositivos da IoT e entidades na Internet. Além de um controle de acesso leve para dispositivos da IoT. O próximo capítulo detalha a aplicação desses mecanismos de segurança no contexto de SG.

Capítulo 5

Estudo de caso em *smart grid*

Neste capítulo é apresentado o MIAM aplicado no contexto de SG. O MIAM pode ser aplicado em diferentes contextos que necessitem de proteção ao acesso local e remoto em recursos críticos. Entretanto, o foco desse capítulo é na aplicação do MIAM em uma arquitetura típica de medição da SG (Figura 2.11). As seções seguintes apresentam detalhes dos mecanismos de segurança propostos aplicados em operações rotineiras na SG. Os processos de criptografia descritos a seguir consideram a cifração autenticada, com o objetivo de garantir a confidencialidade e autenticidade dos dados.

5.1. Autenticação

O mecanismo de IdM, responsável pela autenticação do usuário, é implantando no domínio da empresa de energia elétrica. O usuário (operador) precisa estar endossado pelo DLA_dC antes de se autenticar no IdM. Para isso, através de uma aplicação executada em um computador ou *smartphone*, o operador deve realizar a leitura da credencial de proximidade emitida pelo dispositivo âncora. Essa credencial pode ser obtida por um canal *wireless* ou óptico [67]. Após a obtenção da credencial, o operador encaminha a mesma para o DLA_dC (Figura 5.1). Se a credencial for válida, o operador recebe a autorização expressa através de um *token de proximidade*, que é encaminhada para o SP que deseja utilizar.

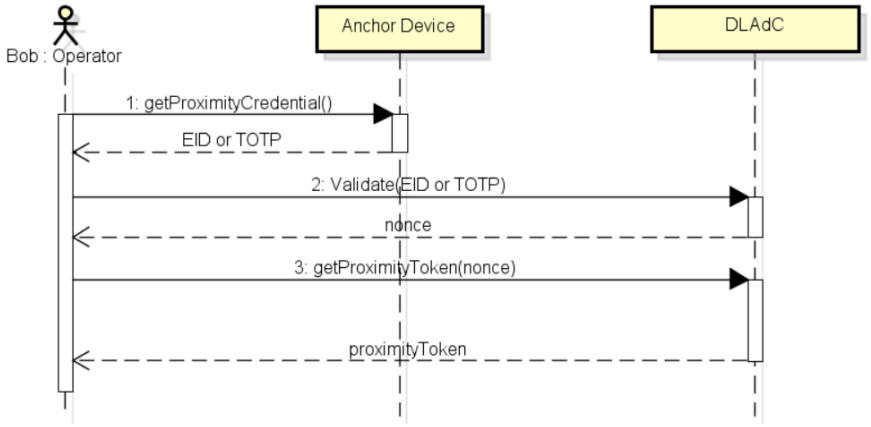

Figura 5.1. Processo de admissão por proximidade.

O processo de obtenção do *token de identidade* e *token de acesso* é representado na Figura 5.2. Primeiramente, o usuário solicita a autenticação ao SP (*Figura 5.2, evento 1*). O SP é o serviço em que o usuário deseja se autenticar, podendo ser instanciado como CS, SCADA etc. O SP recebe por parâmetro o fator que o usuário deseja/necessita autenticar e responde com o endereço do serviço de autenticação para o fator desejado. Após isso, o usuário realiza a autenticação fornecendo suas credencias e o *token de proximidade* emitido pelo DLA_dC (*Figura 5.2, evento 2*). Caso sejam válidas, o MIAM retorna um código pseudoaleatório de curta duração chamado de *nonce*. O objetivo do *nonce* é provar a autenticação do usuário e evitar ataques de replay [18]. Assim, o usuário apresenta o *nonce* ao SP (*Figura 5.2, evento 3*) e o utiliza para obter o *token de identidade* e *token de acesso*.

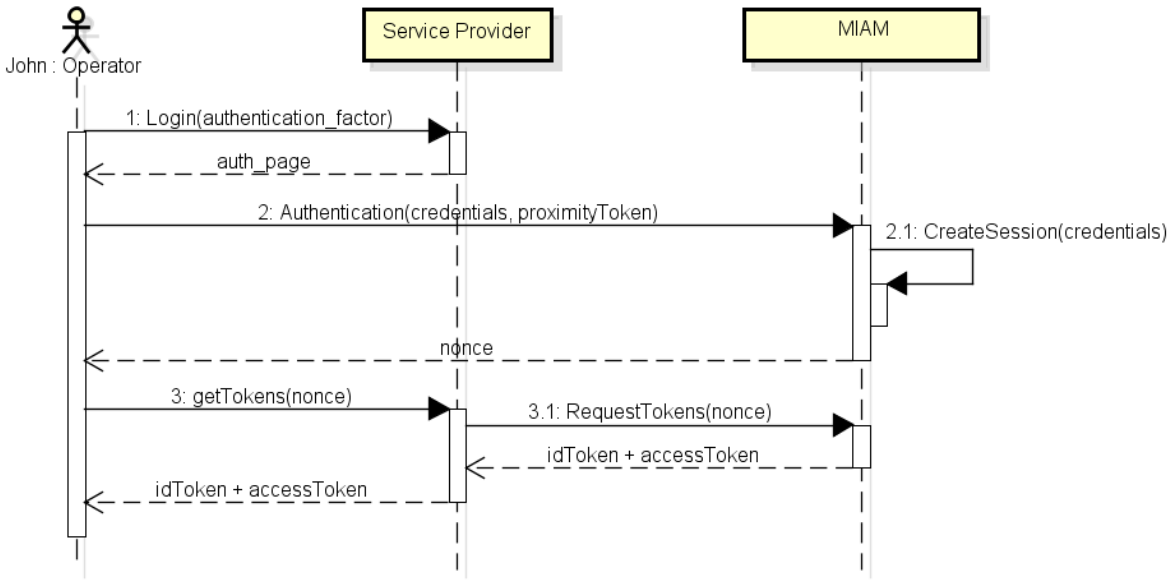

Figura 5.2. Processo de autenticação.

É possível observar que o SP retorna ao usuário a página de autenticação. Dessa forma, caso o SP seja comprometido, poderia retornar uma página de autenticação falsa para realizar *fishing*. A implementação do protótipo, discutida na seção 6.1, evita esse problema na adoção de certificados digitais.

5.2. Controle de papéis

O RC é responsável pela gestão dos papéis. A gestão de papéis consiste em criar, relacionar papéis com os usuários e ativá-los. Além disso, é possível definir hierarquias e conflitos de interesse entre papéis. O RC atua como um provedor de serviços (SP), no qual o usuário é identificado a partir de seu *token de acesso*.

Assim, para consumir os serviços disponibilizados pelo RC, o usuário deve possuir um *token de acesso* com o escopo adequado. Para criar o conflito de interesses, o administrador do sistema seleciona dois papéis, classificando-os como conflito de interesse estático ou dinâmico. Quando o administrador associa um papel a um usuário, o RC verifica se existe algum conflito de interesse estático (SSoD). Quando o usuário ativa um papel, inicia-se o processo de verificação de um possível conflito de interesse dinâmico (DSoD) e caso isso ocorra, o papel não é ativado.

O processo de ativação de papel é representado na Figura 5.3. O usuário solicita a ativação de um determinado papel fornecendo seu *token de acesso* (*Figura 5.3, evento 1*). O RC valida o *token de acesso* no AM (*Figura 5.3, evento 1.1*). Caso o *token de acesso* seja válido, o RC obtém o usuário a partir do *token de acesso* (*Figura 5.3, evento 1.2*). Após isso, o RC verifica se o usuário possui associação com o papel (*Figura 5.3, evento 1.3*), e se existe algum conflito de interesses dinâmico (*Figura 5.3, evento 1.4*). Finalmente, caso as verificações anteriores forem verdadeiras, o RC ativa o papel do usuário (*Figura 5.3, evento 1.5*).

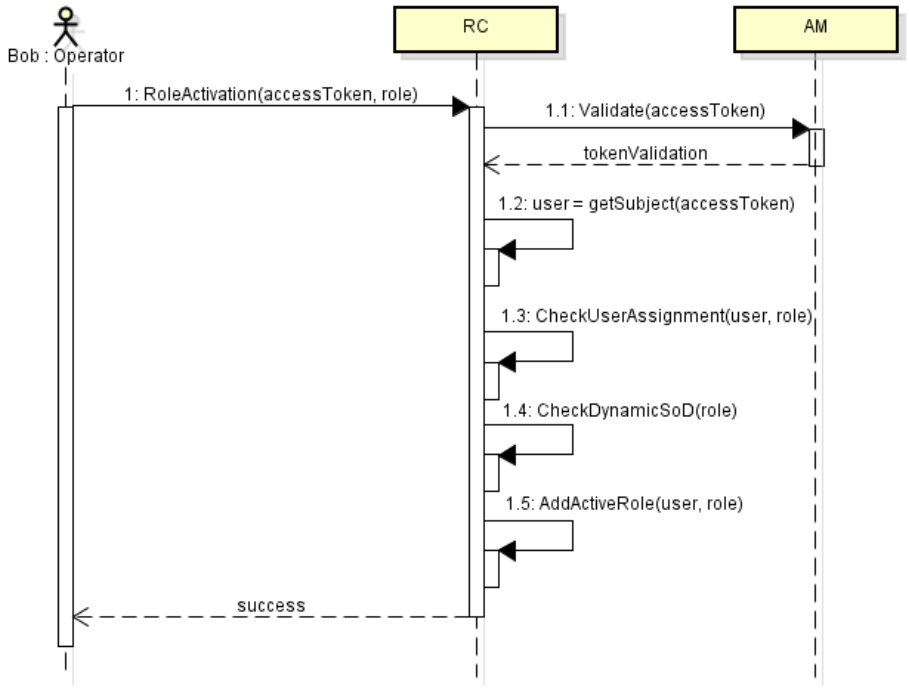

Figura 5.3. Ativação de um papel no RC.

5.3. Operações no domínio local

O ABAC é responsável pelo controle de acesso com granularidade fina que utiliza políticas baseadas nos atributos do usuário, ambiente e recurso. Dessa forma, torna-se o guardião dos recursos protegidos na arquitetura, atuando como um SP.

O ABAC é responsável por deliberar as permissões que um papel terá sobre determinado recurso. Para isso, o administrador do sistema define políticas de segurança vinculando ações, recursos e papéis ativos que estão armazenados no RC. Além disso, o administrador do sistema pode associar atributos do ambiente. Por exemplo, determinado recurso só pode ser acessado das 8h até às 18 horas.

A Figura 5.4 representa o processo de decisão de acesso. O usuário solicita a execução de uma ação sobre um recurso protegido, informando seu *token de acesso* (*Figura 5.4, evento 1*). O ABAC valida o *token de acesso* no AM (*Figura 5.4, evento 1.1*). Caso seja válido, o ABAC obtém os papéis ativos do usuário no RC (*Figura 5.4, evento 1.2*). Finalmente, de posse dos papéis ativos, recurso e ação, o ABAC avalia o direito de acesso baseado em suas políticas de acesso (*Figura 5.4, evento 1.3*).

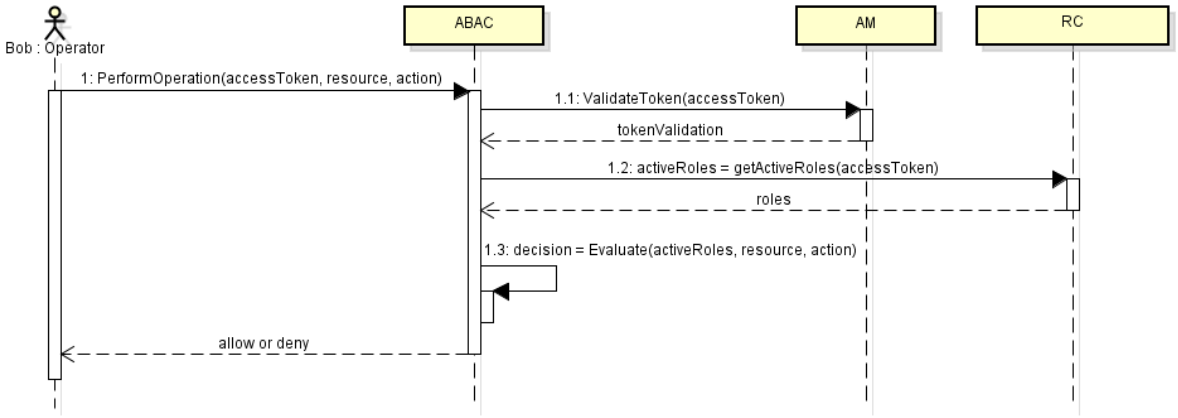

Figura 5.4. Acesso local a um recurso protegido.

Para executar uma operação multidomínios, o ABAC deve consultar os papéis ativos no M_dAC ao invés do RC. A subseção a seguir apresenta o processo de operação multidomínios.

5.4. Operações multidomínios

O M_dAC é responsável por realizar a gestão das operações multidomínios, disponibilizando ao ABAC os papéis que estão associados a usuários de outros domínios da federação. Dessa maneira, domínios que possuem uma relação de confiança podem realizar operações multidomínios sobre recursos protegidos, desde que exista uma política para permitir esse acesso. Cada domínio da federação possui a arquitetura apresentada na Figura 4.1. Assim, cada domínio possui os controles de segurança: IdM, AM e iID.

Para criar uma política que permita operações multidomínios, o usuário deve exportar um de seus papéis ativos. A exportação de papel tem como objetivo viabilizar operações remotas sobre os recursos vinculados a esse papel, sendo que no momento da exportação o usuário pode realizar a exportação parcial desse papel, selecionando de maneira flexível as operações e os recursos.

Para apresentar o processo de exportação e importação de papéis, considere dois domínios/organizações independentes de energia elétrica: Regional e Nacional. Cada domínio possui seus próprios controles de segurança, serviços e usuários. Considere que Alice é uma engenheira da Regional, que pode realizar a leitura e parametrização do consumo de energia elétrica de uma subestação. Alice precisa disponibilizar somente a leitura do consumo da subestação para os Operadores da Nacional. Assim, Alice deve exportar seu papel de

Engenheira para os Operadores da Nacional, permitindo apenas a leitura do consumo da subestação. O M_dAC é responsável por garantir que essa operação multidomínios seja possível. Assim, quando Bob, que é um Operador da Nacional, solicitar a leitura do consumo da subestação, o ABAC consultará o M_dAC. Para facilitar a compreensão, considere que a Regional é um domínio local que disponibiliza seus recursos para Nacional, que é o domínio remoto que deseja realizar operações multidomínios, e considere que o M_dAC se comunica com o CA_dC e QA_dC para realizar operações de criptografia e quórum, respectivamente.

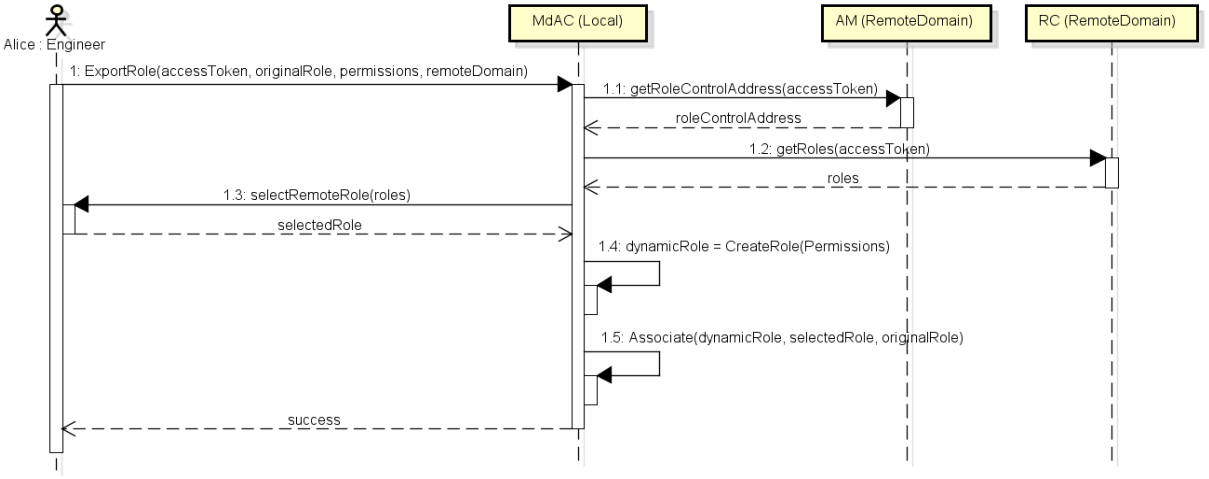

Figura 5.5. Exportação de papel para um domínio remoto.

A primeira etapa consiste na exportação do papel. Considere que Alice possui o papel de Engenheira ativa na Regional e deseja que um determinado papel da Nacional realize a leitura do consumo de uma subestação. Assim, Alice deve solicitar ao M_dAC de seu domínio a exportação do papel Engenheiro, fornecendo seu *token de acesso,* o papel, as permissões sobre os recursos e o domínio remoto (*Figura 5.5, evento 1*). O M_dAC da Regional valida o *token de acesso*. Caso seja válido, solicita ao AM da Nacional o endereço do RC da Nacional (*Figura 5.5, evento 1.1*). Com o endereço do RC da Nacional, o M_dAC da Regional consulta os papéis disponíveis no RC da Nacional, caso o *token de acesso* disponibilizado seja válido (*Figura 5.5, evento 1.2*). O M_dAC da Regional apresenta para Alice os papéis disponíveis e ela escolhe o papel de Operador (*Figura 5.5, evento 1.3*). O M_dAC da Regional cria um papel dinamicamente com a permissão de leitura do consumo de uma subestação (*Figura 5.5, evento 1.4*) e associa ao papel de Operador da Nacional e a Engenheiro da Regional (*Figura 5.5, evento 1.5*). Assim é estabelecido o vínculo entre os papéis de ambos domínios com o papel dinâmico.

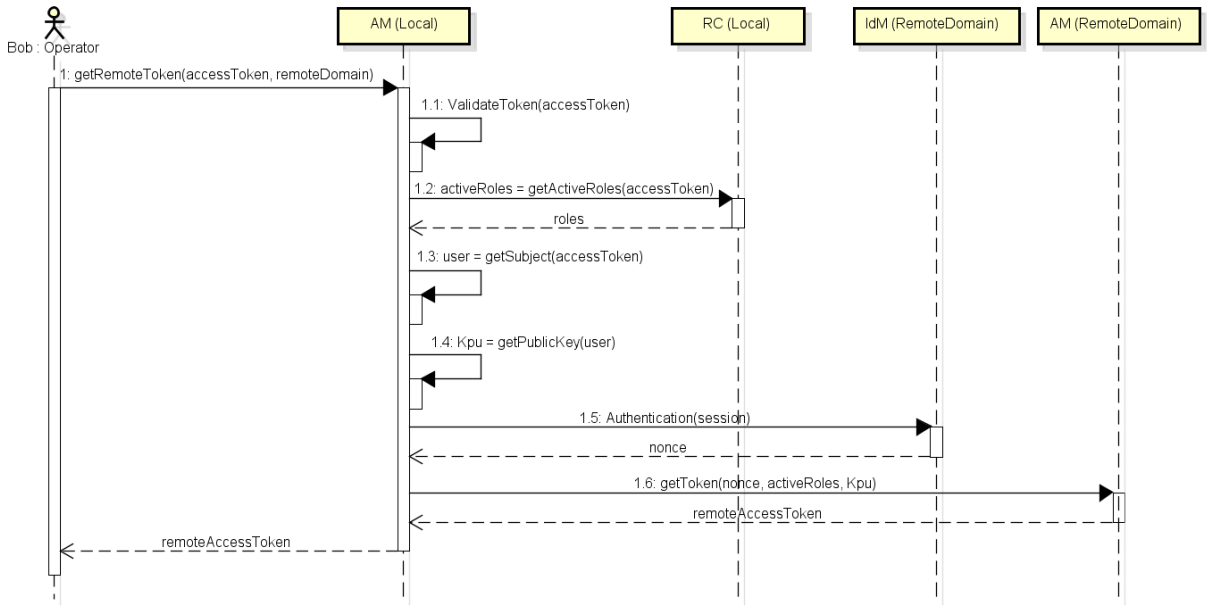

Figura 5.6. Obtenção do *token de acesso* remoto.

Para que Bob, Operador da Nacional, consiga acessar aos serviços protegidos da Regional é necessário que ele obtenha um *token de acesso remoto*. Para isso, Bob solicita ao AM de seu domínio um *token de acesso* para acessar serviços da Regional (*Figura 5.6, evento 1*). O AM valida o *token de acesso* de Bob, caso seja válido, o AM da Nacional solicitará ao AM da Regional um *token de acesso* que contém no escopo os papéis ativos de Bob e sua chave pública. Primeiramente, o AM da Nacional valida o *token de acesso* (*Figura 5.6, evento 1.1*). Caso seja válido, obtém os papéis ativos do usuário no RC da Nacional (*Figura 5.6, evento 1.2*). O AM da Nacional recupera a chave pública de Bob (*Figura 5.6, evento 1.3*) no CA_dC. Com posse dessas informações, o AM da Nacional realiza a autenticação única (SSO) no IdM da Regional, que retorna o *nonce* (*Figura 5.6, evento 1.5*). Finalmente, o AM da Nacional solicita o *token de acesso remoto* ao AM da Regional, informando o *nonce*, os papéis ativos de Bob (Operador) e a sua chave pública. O AM da Regional retorna o *token de acesso remoto* para o AM da Nacional (*Figura 5.6, evento 1.6*).

Com posse do *token de acesso remoto*, torna-se possível que Bob consulte os papéis exportados disponíveis para o seu papel de Operador. Para isso, Bob solicita os papéis exportados para o M_dAC da Regional, fornecendo seu *token de acesso remoto* (*Figura 5.7, evento 1*). O AM da Regional valida seu *token de acesso remoto* (*Figura 5.7, evento 1.1*). Caso seja válido, O AM extrai do *token de acesso remoto* o usuário (Bob), os papéis ativos do usuário em seu domínio de origem (Operador) e sua chave pública (*Figura 5.7, evento 1.2, 1.3 e 1.4* respectivamente). Após obter os papéis ativos do usuário, o M_dAC da Regional pesquisa os

papéis exportados para esses papéis ativos (*Figura 5.7, evento 1.5*). Ou seja, pesquisa os papéis exportados para Operador da Nacional. O M_dAC da Regional cifra os papéis exportados na chave pública de Bob (*Figura 5.7, evento 1.6*). Bob recebe os papéis cifrados em sua chave pública. Como ele é o único, em teoria, a ter posse da sua chave privada, consegue decifrar os papéis cifrados para obter os papéis exportados.

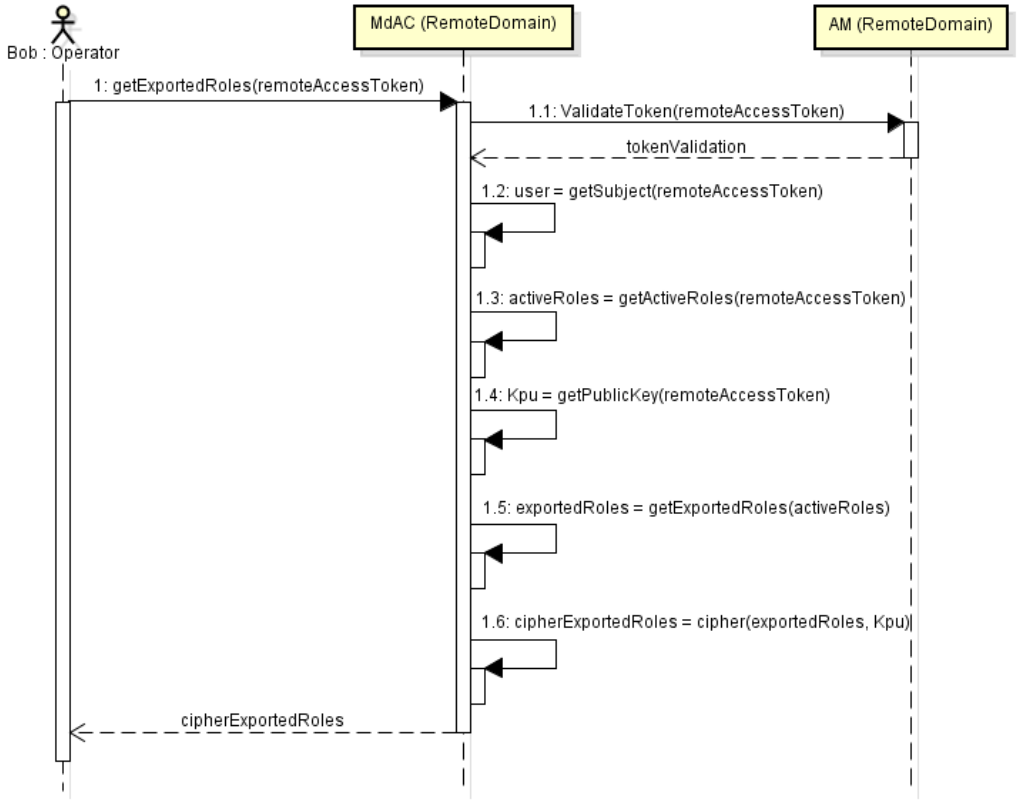

Figura 5.7. Consulta de papéis exportados.

Nesse momento, Bob identificou que existe um papel em Regional que pode acessar, para realizar a leitura do consumo de energia elétrica de uma subestação. Entretanto, antes de Bob solicitar acesso ao recurso, ele deve ativar esse papel remoto. Para isso, Bob precisa selecionar o papel de Regional que deseja acessar e assiná-lo utilizando sua chave privada (*Figura 5.8,* evento 2). Após isso, solicita a ativação do papel remoto para o M_dAC da Regional, fornecendo o *token de acesso remoto,* o identificador do papel e o papel assinado (*Figura 5.8, evento 3*). O AM da Regional valida o *token de acesso remoto*, caso seja válido, o M_dAC da Regional extrai do *token de acesso remoto* o usuário (Bob) e sua chave pública (*Figura 5.8, evento 3.2 e 3.3* respectivamente). O M_dAC da Regional verifica a assinatura do papel para garantir a autenticidade do usuário (*Figura 5.8, evento 3.4*). Após isso, o M_dAC da Regional identifica se o papel que Bob deseja ativar está vinculado ao papel de Engenheiro (*Figura 5.8,*

evento 3.5). Assim, o M_dAC da Regional solicita ao QA_dC da Regional os usuários com o papel de Engenheiro ativo (*Figura 5.8, evento 3.6*), com a finalidade de realizar o quórum. Finalmente, o M_dAC de Regional identifica que Alice está com o papel de Engenheira ativa, então a associa e ativa o papel de Bob (*Figura 5.8, evento 3.7 e evento 3.8*, respectivamente).

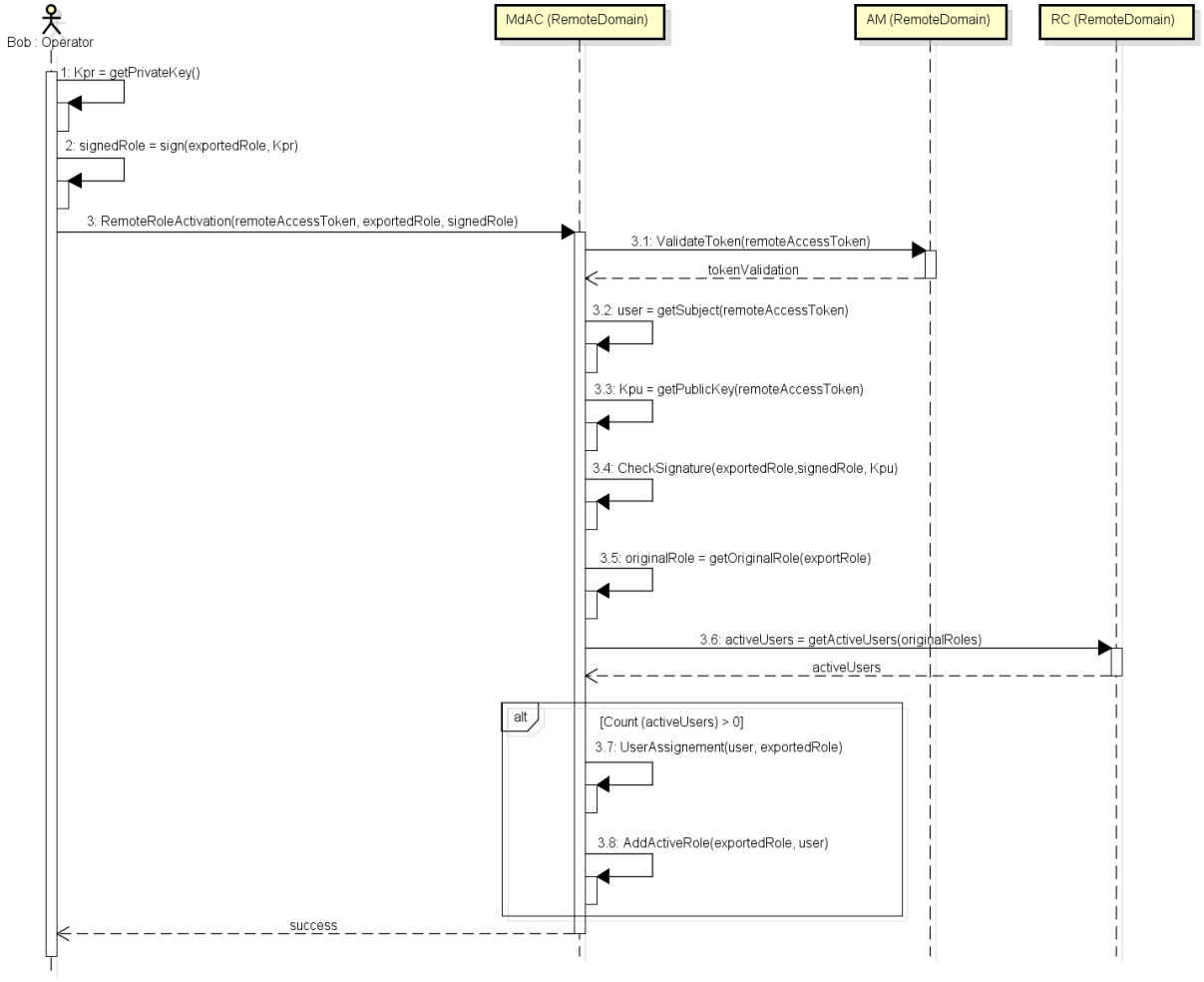

Figura 5.8. Ativação de papel remota.

Finalmente, após Bob, Operador da Nacional, estar com um papel ativo no M_dAC da Regional, ele poderá realizar uma operação multidomínio nos recursos protegidos da Regional. Para isso, Bob solicita a leitura do consumo de energia elétrica da subestação para o ABAC da Regional, fornecendo seu *token de acesso remoto* (*Figura 5.9, evento 1*). O ABAC valida o *token de acesso remoto* (*Figura 5.9, evento 1.1*). Caso seja válido, recupera os papéis ativos no M_dAC (*Figura 5.9, evento 1.2*). Com posse dos papéis ativos, recursos e ações, o ABAC avalia, baseado nas políticas, se a decisão é válida ou não (conforme discutido na subseção 5.3). Caso seja válida, o ABAC ainda não permite o acesso ao recurso (*Figura 5.9, evento 1.3*). Antes disso, identifica-se que o papel fornecido por Bob está vinculado ao papel de Engenheiro

(*Figura 5.9, evento 1.4*). Assim, O ABAC solicita ao QA_dC os usuários que estão com o papel Engenheiro ativo (*Figura 5.9, evento 1.5*). Finalmente, o QA_dC realiza uma operação de quórum (*Figura 5.9, evento 1.6*), para verificar o endosso dos usuários que estão com o papel do Engenheiro ativo. Caso a decisão do quórum seja positiva, o acesso ao recurso é permitido pelo ABAC. Se não, o acesso é negado.

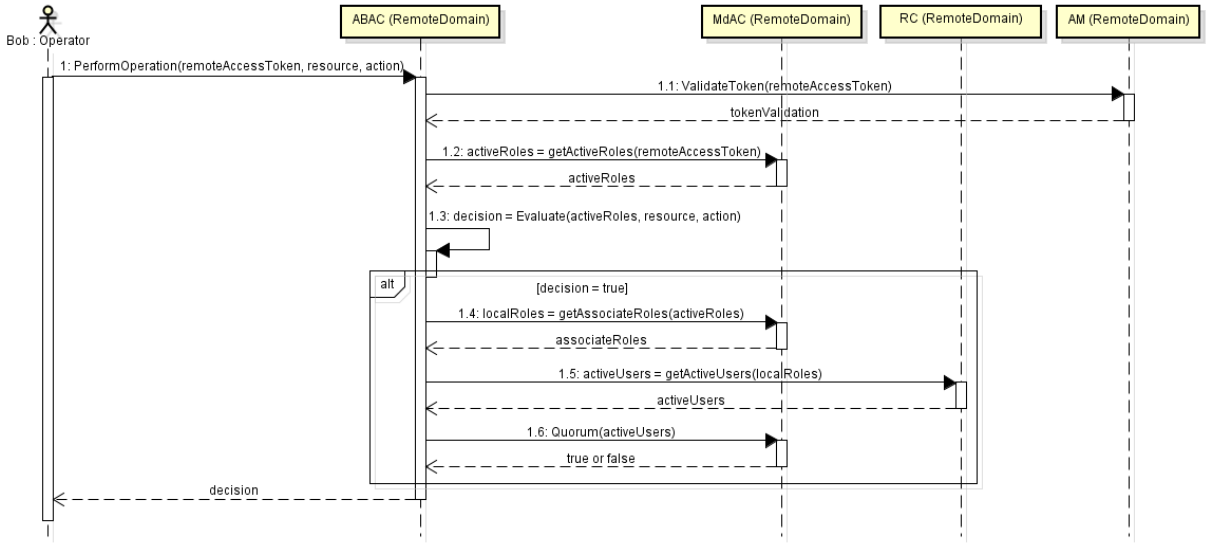

Figura 5.9. Acesso remoto a um recurso protegido.

Os papéis exportados não são armazenados no RC. Dessa forma, o RC segue os padrões definidos em [21]. Esses papéis exportados tem uma curta duração, parametrizado pelo administrador do sistema. É importante ressaltar que uma operação multidomínio a um recurso protegido é realizada da mesma maneira que uma operação interna ao ABAC. Em ambos os casos, o ABAC solicita os papéis vinculados à política, que podem ser fornecidos pelo RC ou pelo M_dAC. Além disso, é possível que o administrador do sistema imponha regras que impeçam a exportação (disponibilização) de determinados recursos críticos para outros domínios. Da mesma forma, é possível colocar uma restrição no sistema, para que seja necessário solicitar o endosso, através de assinatura digital, do administrador do sistema para exportar o papel. Finalmente, a qualquer momento, um usuário que realizou a exportação do papel pode revogar no M_dAC a exportação de seu papel.

5.5. Cibersegurança nos dispositivos IoT

Essa seção apresenta duas abordagens de segurança fim-a-fim na comunicação dos medidores inteligentes (SM) com o sistema central (CS) da empresa de energia elétrica. Ambas abordagens são baseadas na utilização do MIAM, considerando as restrições da IoT. Dessa maneira, o MIAM permite que as mesmas credenciais que um usuário utilizou para autenticar no CS, através da HMI, sejam utilizadas nos SMs. Posteriormente serão avaliadas ambas abordagens. Além disso, essa seção apresenta um controle de acesso leve adequado às restrições computacionais da IoT.

5.5.1. Abordagem baseada em hierarquia de chaves

Devido às restrições de poder computacional, característica intrínseca da IoT (discutido na seção 2.5), não é possível adotar os padrões de segurança tradicionais da Internet (e.g. TLS). Dessa forma, a segurança do SM foi desenvolvida para atender aos requisitos da IoT. Com objetivo de garantir a confidencialidade na comunicação, a criptografia de chave simétrica foi empregada, pois exige menor poder de processamento, quando comparado com a criptografia de chave pública [68].

Dessa maneira, cada SM possui uma chave simétrica que é compartilhada com o CS (Figura 5.10). Para mitigar a possibilidade de comprometimento da chave secreta, foi adotado o conceito de hierarquia de chaves, definido na ANSI X.9.17 [66]. O referido padrão apresenta um esquema de gerenciamento de chaves simétricas, baseado em três tipos de chaves secretas. A primeira chamada de chave mestre (LMK, *Local Master Key*) nunca trafega pela rede, pois é armazenada em linha de produção no SM de maneira protegida por *hardware* (e.g. *smartcard*). Já no CS, a LMK é armazenada em uma base de chaves com proteção adequada, similar à que é empregada no *Key Distribution Center (KDC)* do Kerberos [68]. A LMK é utilizada para distribuir de maneira cifrada a chave de sessão (KEK, *Key-Encrypting Key*), com duração personalizada pelo administrador do sistema de, por exemplo, 24 horas. A chave de sessão, KEK, é armazenada em memória e tem a finalidade de distribuir a chave que será utilizada para proteger o conteúdo, a chave de dados (DK, *Data Key*). A DK é utilizada para proteger os conteúdos na comunicação fim-a-fim entre o SM e CS. A duração da DK pode ser

personalizada pelo administrador do sistema, sendo recomendada uma duração curta. Por exemplo, a cada 10 mensagens a DK deve ser renovada, ou a cada 10 minutos.

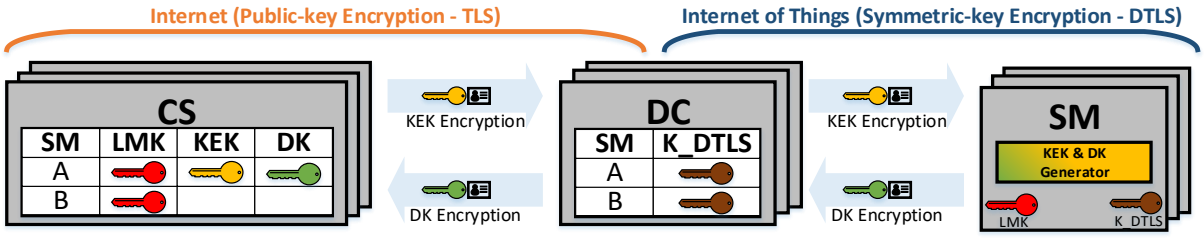

Figura 5.10. Abordagem baseada na hierarquia de chaves.

A geração das chaves (KEK e DK) sempre é realizada pelo CS, com o objetivo de diminuir a possibilidade de comprometimento em larga escala. Por exemplo, caso o SM seja comprometido, apenas esse dispositivo específico será acessível. Por outro lado, se a geração de chaves ocorre no CS e se torna comprometido, todos os dispositivos vinculados poderiam ser acessados.

Essa abordagem possui dois níveis de criptografia que garantem proteção da comunicação fim-a-fim. O nível interno de criptografia utiliza a criptografia de chave simétrica baseada na ANSI X.9.17 [66]. O nível externo utiliza criptografia de chave pública para as entidades hospedadas na Internet e utiliza a criptografia de chave simétrica para as entidades da IoT. O DC é o elemento responsável por gerenciar esse nível externo de criptografia, atuando como um *gateway* na arquitetura, realizando a conversão (*gateway*) das mensagens entre o SM e CS, e vice-versa. Ou seja, o DC intermedeia as conversas, que podem ser iniciadas pelo SM ou CS. O DC possui duas interfaces de comunicação, uma para atender à Internet e outra para atender à IoT. Assim, o papel do DC é converter as mensagens da Internet que utilizam a criptografia de chave pública para a IoT que utiliza a criptografia simétrica, e vice-versa.

É indispensável destacar que o DC não tem acesso ao conteúdo das mensagens trafegadas, visto que no nível interno estão cifradas via chave simétrica. Assim, o DC não é considerado um ponto único de falha na arquitetura. Caso seja comprometido, o adversário não é capaz de visualizar o conteúdo trafegado. Além disso, é comum que mais de um DC atenda um SM para ser tolerante a falha.

O CS é responsável por intermediar o acesso do usuário no SM, sendo responsável por armazenar a chave mestre (LMK), a chave de sessão (KEK) e os dados (DK) de cada SM.

Quando um usuário desejar acessar ao SM, o CS informa a KEK (caso já exista) ou solicita a geração de uma nova.

Para o usuário obter a KEK deve-se estar autenticado no MIAM (*Figura 5.2*) e com posse do *token de acesso*. De posse do *token de acesso*, o usuário pode, de maneira transparente, solicitar para o CS a KEK de um determinado SM, informando seu número de serial (*Figura 5.11, evento 1*). O CS valida o *token de acesso* no MIAM (*Figura 5.11, evento 1*.1). Caso seja válido, obtém a chave pública do usuário (*Figura 5.11, evento 1.2*). Após isso, o CS verifica se existe algum SM vinculado ao número de serial informado pelo usuário. Em caso afirmativo, o CS verifica se existe uma KEK válida, ou seja, não expirada (*Figura 5.11, evento 1.3*). Caso exista, retorna a KEK cifrada na chave pública do usuário (*Figura 5.11, evento 1.4*). Esse processo garante que apenas o detentor da chave privada possa decifrar a KEK. Observe que o administrador do CS pode definir políticas que definem a expiração de uma chave KEK. Por padrão, o tempo de vida de uma KEK é de 24 horas.

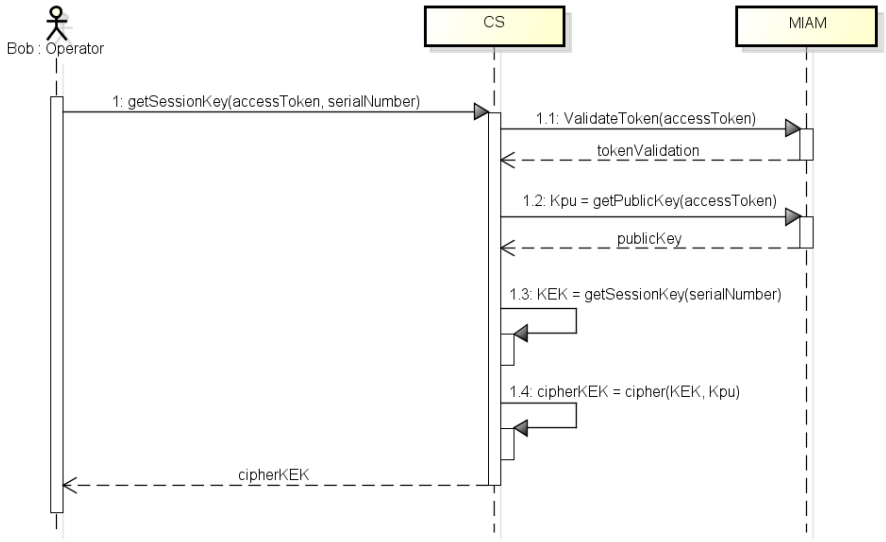

Figura 5.11. Consulta da KEK de um SM específico.

Se a KEK for inválida ou inexistente, o usuário deverá solicitar a geração de uma nova chave de sessão. Para gerar a KEK, o usuário solicita ao CS, informando seu *token de acesso* e o número do serial (*Figura 5.12, evento 1*). O CS valida o *token de acesso* (*Figura 5.12, evento 1.1*) e caso seja válido, recupera a LMK do SM vinculada ao número de serial (*Figura 5.12, evento 1.2*). Após isso, o CS cifra na LMK o pedido de geração de uma KEK (*Figura 5.12, evento 1.3*). Com posse dessas informações, realiza uma requisição ao DC, informando o *token de acesso*, número de serial e o pedido de geração de KEK cifrado (*Figura 5.12, evento 1.4*). O DC valida o *token de acesso* e deposita a requisição em uma fila (*Figura 5.12, evento 1.4.2*),

retornando um ticket ao CS. O CS aguarda seu ticket ser resolvido. Frequentemente, o SM solicita ao DC requisições pendentes, informando seu número de serial (*Figura 5.12, evento 2*). Quando existe uma requisição pendente para seu número serial, o SM obtém a requisição cifrada e o respectivo *ticket*. Assim, decifra a requisição utilizando sua LMK (*Figura 5.12, evento 3*) e identifica o pedido de geração de KEK. O SM gera a KEK (*Figura 5.12, evento 4*), cifra com sua LMK (*Figura 5.12, evento 4*) e deposita a resposta no DC, informando o *ticket* correspondente. O CS que estava frequentemente consultando o DC, identifica que o *ticket* foi resolvido e recebe a KEK cifrada. Em seguida, decifra utilizando a LMK do SM. A KEK é associada ao número serial do SM. Finalmente, o usuário é informado que a KEK foi gerada com sucesso, para obtê-la deve solicitar a KEK novamente conforme o processo da Figura 5.11.

Figura 5.12. Processo de geração de KEK.

Essa abordagem permite que a interação seja realizada de maneira assíncrona e desacoplada. O CS não é capaz de injetar informações diretamente no SM, apenas deposita as informações no DC. Essa abordagem mitiga a possibilidade de um adversário injetar código malicioso ou controlar o SM, pois não há sessão interativa. Para atender às requisições, frequentemente o SM solicita requisições pendentes ao DC, que a informa baseado em um sistema de *ticket*. Da mesma maneira, após depositar no DC, o CS frequentemente consulta seus

tickets pendentes. Esse desacoplamento entre as entidades pode adotar o paradigma de publicador e assinante *(publish/subscriber).* Observe que, no diagrama o processo de criptografia de chave pública baseado nos padrões tradicionais da Internet (e.g. TLS) foram ocultados para simplificar a representação. Assim como o processo de criptografia de chave simétrica entre o DC e o SM. Ambos processos são identificados como criptografia externa nessa abordagem.

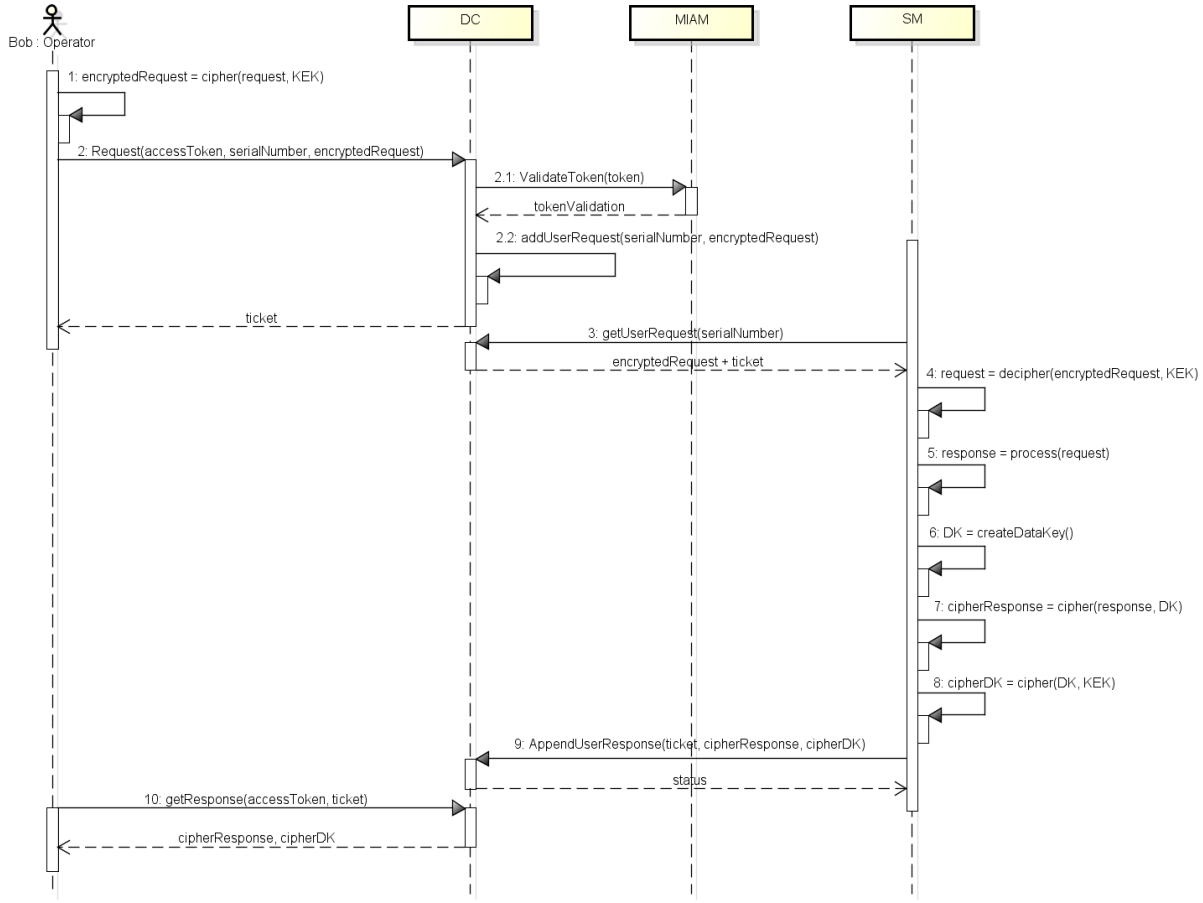

Figura 5.13. Processo de requisição ao SM.

Com posse da chave de sessão decifrada, o usuário pode interagir com o SM. O processo de interação também é realizado de maneira assíncrona e desacoplada. Primeiramente, o usuário cifra a requisição utilizando a KEK (*Figura 5.13, evento 1*). Após isso, solicita uma requisição ao DC, informando o *token de acesso*, o número de serial e a requisição cifrada (*Figura 5.13, evento 2*). O DC valida o *token de acesso* (*Figura 5.13, evento 2.1*), caso seja válido, deposita a requisição do usuário na fila, associando o número serial (*Figura 5.13, evento 2.2*). O SM frequentemente consulta o DC para obter a requisição do usuário cifrada e seu *ticket* (*Figura 5.13, evento 3*). Assim, o SM decifra e processa a requisição (*Figura 5.13, evento 4 e 5* respectivamente). O SM cria uma DK para cifrar a resposta. Além disso, cifra a DK utilizando

a KEK. Finalmente, o SM deposita a resposta e a DK cifrada no DAS. Dessa maneira, o usuário obtém, utilizando seu *ticket*, a resposta e a DK cifrada. Assim, o usuário decifra a DK e em seguida a utiliza para decifrar a resposta.

A abordagem baseada em chaves secretas garante a proteção fim-a-fim da comunicação entre os elementos da SG, e permite que um usuário autenticado no MIAM transporte suas credenciais obtidas na Internet para o contexto da IoT. Esse processo é adequado às restrições da IoT, garantindo a confidencialidade da comunicação. Além disso, esse esquema por utilizar SSO permite realizar auditorias de acesso, pois é utilizado o mesmo *token de acesso* para acessar diversos dispositivos. Apesar desse processo garantir a comunicação segura durante a comunicação, ele não consegue estabelecer um controle fino sobre o acesso aos recursos protegidos.

5.5.2. Abordagem baseada em OTP

Esta abordagem é baseada no conceito de senha descartável, que é utilizada como chave simétrica para cifrar e decifrar a comunicação entre entidades da Internet e IoT. O valor gerado pelo OTP é baseado em um contador dinâmico e incremental compartilhado entre entidades da IoT e da Internet. Dessa maneira, um contador é compartilhado entre o CS e o SM é usado para garantir a comunicação fim-a-fim, e outro contador *é* compartilhado entre o DC e o SM é usado para proteger a comunicação *multicast*. O objetivo dessa comunicação multicast é reduzir o número de trocas de mensagem, uma vez que é comum que o CS requisite informações de diversos SMs. Assim, a senha descartável gerada pelo OTP é usada como chave de grupo na comunicação [68].

Para realizar a comunicação fim-a-fim entre o CS e o SM é utilizado o método TOTP, que é baseado no HMAC (*Hashed Message Authentication Code*). Esse método utiliza um contador e uma chave simétrica (LMK) para produzir um número de 160 bits (*nonce*). Esse valor pode ser truncado para ter um tamanho customizado. O TOTP é um método simples, leve e seguro para produzir senhas dinâmicas que podem proteger a comunicação entre as entidades na Internet com os dispositivos da IoT.

Similar a abordagem de hierarquia de chaves, a comunicação entre entidades da Internet é protegida pela criptografia de chave pública (TLS), enquanto as comunicações das entidades da IoT são protegidas pela criptografia simétrica (DTLS). Essa abordagem requer

uma chave secreta (LMK) e um *contador* compartilhado entre o CS e o SM, sendo que o LMK e o *contador* nunca são transmitidos pela rede. Além dessas premissas, essa abordagem requer uma chave secreta e um *contador* compartilhado entre o DC e os SMs. Entretanto, essas informações são transmitidas dinamicamente de maneira segura pela rede.

A Figura 5.14 apresenta o protocolo para definição da chave DTLS (K_DTLS). Para isso, o DC gera uma nova K_DTLS (*Figura 5.14, evento 1*) e cifra a mesma na chave pública do CS (*Figura 5.14, evento 2*). Em seguida, encaminha a requisição para o CS cifrar a K_DTLS com a LMK (*Figura 5.14, evento 3*). Assim, o CS decifra a mensagem usando sua chave privada (*Figura 5.14, evento 3.1*) para poder cifrar na LMK (*Figura 5.14, evento 3.2*). O DC recebe a mensagem cifrada em sua chave pública e decifra com sua chave privada (*Figura 5.14, evento 4*). Finalmente, o DC encaminha a K_DTLS cifrada para o SM (*Figura 5.14, evento 5*), no qual será decifrada na LMK do SM (*Figura 5.14, evento 5.1*).

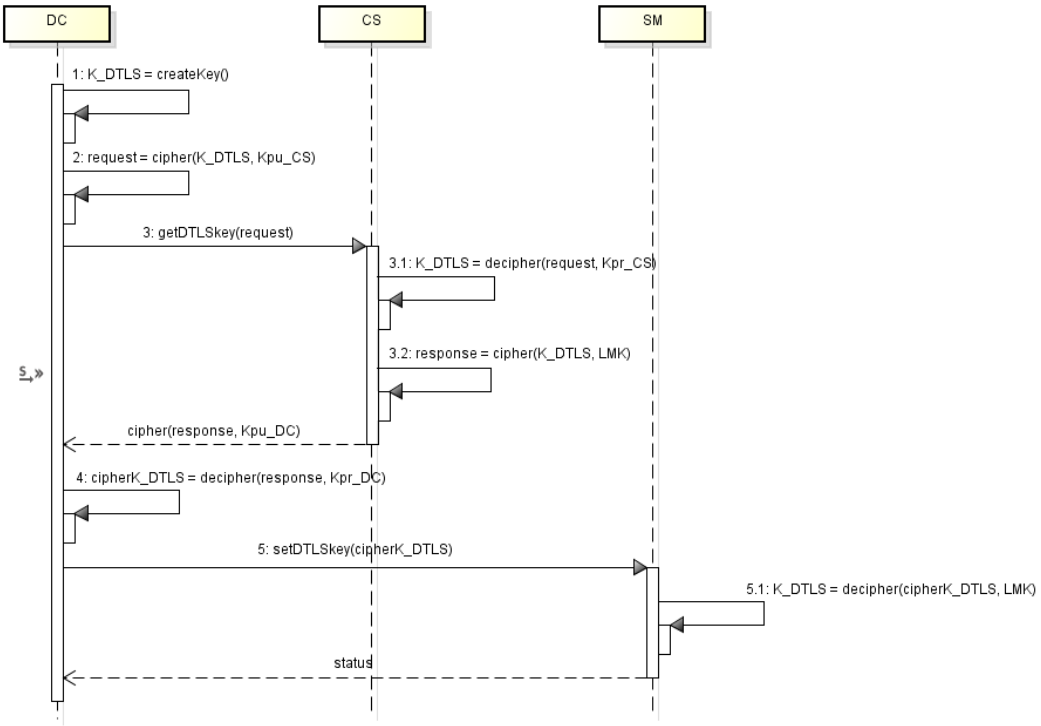

Figura 5.14. Processo de definição da chave DTLS.

É importante observar que toda comunicação é cifrada, sendo que a comunicação entre o CS e o DC é protegida pela criptografia de chave pública. Por outro lado, a comunicação entre o CS e o SM é protegido por uma criptografia de chave simétrica, no qual a LMK nunca é transmitida/exposta na rede. Esse mesmo processo é utilizado para definir o *contador* compartilhado entre o DC e o SM.

A comunicação *multicast*, adotada entre o DC e os SM, utiliza o método convencional de OTP, ao invés do TOTP. Esse método não utiliza uma chave criptográfica para produzir a senha. Essa escolha foi motivada por duas razões: (i) evita a necessidade do compartilhamento de uma chave criptográfica entre o DC e o SM; (ii) de maneira geral as requisições não contem conteúdos sensíveis, apenas as respostas que estão cifradas em duas camadas de criptografia. Assim, a camada mais interna é cifrada utilizando a senha (chave) produzida pelo TOTP, enquanto a camada mais externa é cifrada no K_DTLS. Assim, o DC consegue apenas decifrar a camada externa, não sendo considerado um ponto único de falhas (Figura 5.15).

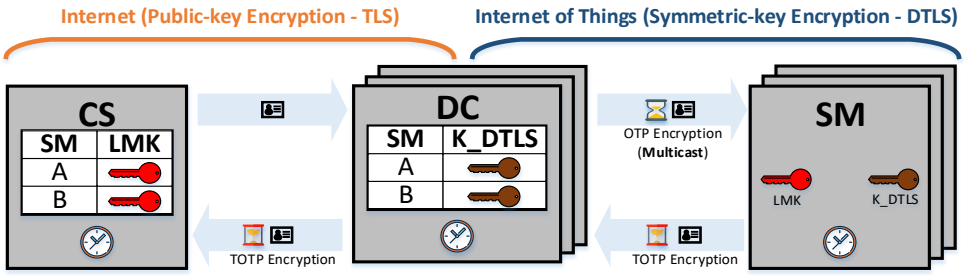

Figura 5.15. Abordagem baseada em OTP.

O protocolo de segurança fim-a-fim utilizando a abordagem OTP é apresentado na Figura 5.16. Primeiramente, um operador autenticado no CS requisita, por exemplo, o consumo de energia atual de todos os SMs de uma determinada região, fornecendo seu token de acesso (*Figura 5.16, evento 1*). O DC valida o *token de acesso* e caso seja válido gera uma nova chave utilizando o OTP para ser utilizada como chave de grupo (*Figura 5.16, evento 1.1*). Assim, a requisição do CS é cifrada na chave de grupo (*Figura 5.16, evento 1.2*) e transmitida para todos os SMs através do *multicast* (*Figura 5.16, evento 1.3*).

Todos SMs pertencentes ao grupo *multicast* recebem a mensagem cifrada na chave de grupo. Cada SM gera uma nova chave de grupo utilizando o OTP (*Figura 5.16, evento 1.3.1*) para decifrar a requisição (*Figura 5.16, evento 1.3.2*). Assim, após processar a requisição do CS, o SM gera uma chave utilizando o TOTP, parametrizado pela LMK (*Figura 5.16, evento 1.3.3*). Essa chave é usada para cifrar a resposta (*Figura 5.16, evento 1.3.4*) que será enviada para o DC, cifrada no K_DTLS.

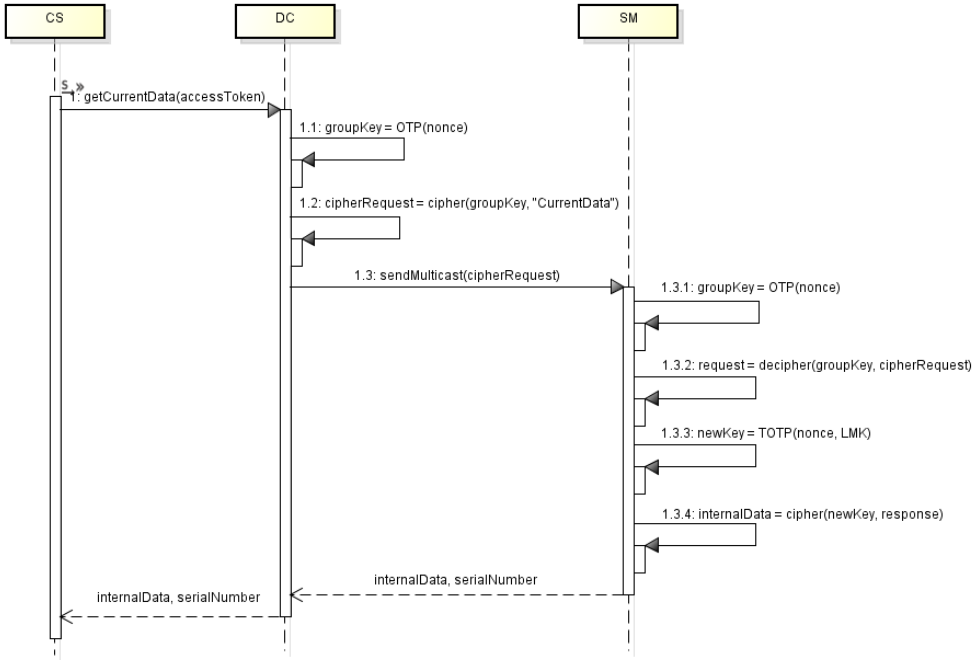

Figura 5.16. Comunicação fim-a-fim utilizando a abordagem do OTP.

Observe que esse processo garante a segurança fim-a-fim na comunicação, pois todas as comunicações estão cifradas. Nesse contexto, o DC é capaz de ler o conteúdo das requisições realizadas pelo CS para os SMs. Entretanto, não é capaz de ler as respostas dos SMs porque são cifradas utilizando a chave gerada pelo TOTP que utiliza a LMK compartilhada entre o CS e o SM. Uma possível variação nesse processo seria permitir que o DC atue como concentrador de dados. Nessa variação, a resposta do SM seria cifrada apenas na K_DLTS ao invés do TOTP. Assim, o DC é capaz de decifrar o conteúdo para agregar e consolidar informações para serem transmitidas para o CS, reduzindo seu processamento.

5.5.3. Controle de acesso leve para IoT

Apesar das abordagens propostas proverem segurança fim-a-fim na comunicação entre dispositivos da IoT, o controle de acesso para IoT continua sendo um desafio a ser resolvido. Isso ocorre porque a autorização provida pelo *token de acesso* limita o acesso a um recurso protegido, mas não suporta políticas determinando operações sobre esses recursos. Assim, não é possível estabelecer um controle de acesso fino nos recursos protegidos.

A arquitetura tradicional de um controle de acesso é tipicamente composta por um monitor de referência, uma base de autorização e um mecanismo guardião (seção 2.3).

Entretanto, essa arquitetura não é adequada às restrições de recursos computacionais dos dispositivos da IoT. Com base nisso, é proposto um controle de acesso leve, do ponto de vista de processamento, transmissão, memória e criptografia simétrica, baseado em OTP. Esse mecanismo aproveita as configurações estabelecidas na seção 5.5.2 para propor um controle de acesso baseado em papéis, no qual o usuário possui direitos de acesso de acordo com o seu papel na organização. Entretanto, esse mecanismo não adota o conceito de sessões, definido no modelo RBAC, para simplificar a implementação e implantação em dispositivos da IoT.

Para acessar um determinado SM, três papéis podem ser definidos, por exemplo: (i) *operador*: realiza consultas (operações de leitura) nos recursos protegidos; (ii) *administrador*: realiza parametrizações (operações de escrita) nos recursos protegidos; (iii) *fabricante*: realiza operações de leitura e escrita nos SM, inclusive atualizações de *firmware*. Outros papéis e permissões poderiam ser definidos, porém para simplificar considere somente esses três. Cada um desses papéis possui um *contador* de TOTP associado, tanto no CS quanto no SM (Figura 5.17). Assim, quando um determinado administrador desejar acessar um SM, ele deve cifrar a requisição utilizando a senha produzida pelo TOTP associado, que utiliza um *contador* de administrador com a LMK de um SM específico para produzir a senha.

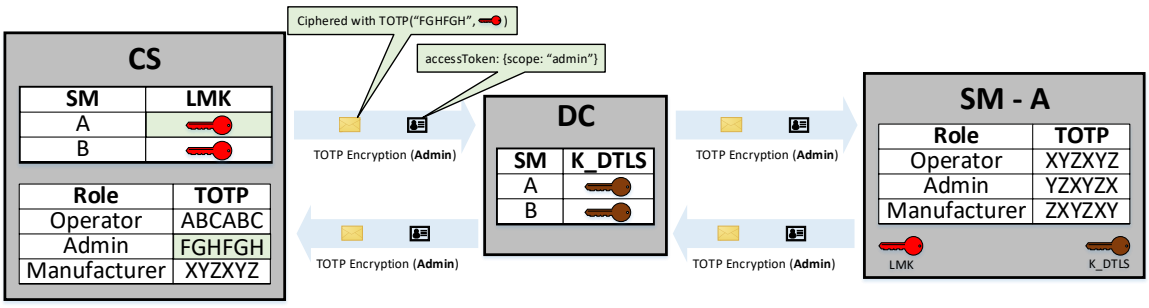

Figura 5.17. Administrador acessando um recurso protegido da IoT.

Quando o SM recebe a requisição, o mesmo tentará decifrar a mensagem utilizando cada *contador* associado aos papéis, começando pelo operador, seguido do administrador e fabricante. Essa sequência poderia ser ordenada pela frequência de operações de cada papel, por exemplo. Para garantir que o SM obtenha sucesso na decifração, o CS encaminha o identificador da requisição decifrado e o mesmo identificador cifrado na chave gerada pelo TOTP. Se o identificador decifrado for igual ao identificador recebido, isso significa que o *contador* utilizado está correto, sendo possível inferir no papel vinculado.

Além disso, esse controle de acesso é baseado em duas etapas. Pois para que um usuário obtenha autorização no controle de acesso, ele deve atender a dois requisitos: (i) possuir

um *token de acesso* com o escopo apropriado a requisição realizada; (ii) conseguir decifrar a resposta do SM, que foi cifrada na LMK e no *contador* do papel. Ou seja, para que o adversário consiga comprometer o dispositivo da IoT, deve possuir o *token de acesso* e o *contador* do papel.

5.6. Discussão

Esse capítulo dissertou o MIAM (proposto no capítulo 4) aplicado a uma arquitetura típica de medição de SG (seção 2.5). Para que o usuário utilize o mecanismo de autenticação multifator (MFA), primeiramente deve obter o endosso do controle de admissão baseado em localidade (DLA_dC). Para isso, deve estar fisicamente próximo de um dispositivo âncora, pois isso evita que um adversário da Internet comprometa o MFA. O controle de papéis realizado é baseado no padrão do RBAC, no qual é possível realizar separação de deveres estático e dinâmico. Esses papéis são avaliados por um controle de acesso de granularidade fina que consulta os papéis ativos do usuário no momento da avaliação.

O MIAM permite que operações multidomínios sejam realizadas, desde que o domínio pertença à federação. O usuário que deseja compartilhar um determinado recurso tem autonomia para definir as políticas de acesso. O mecanismo de AM foi concebido para manter-se integro, mesmo que uma das entidades seja comprometida. Assim, um usuário remoto precisa obter autorização do controle de acesso multidomínios, do controle de admissão criptográfico e do controle de admissão de quórum para acessar a um determinado recurso.

Para garantir a segurança fim-a-fim na comunicação entre as entidades da arquitetura, foram desenvolvidas duas abordagens baseadas em chaves criptográficas. A abordagem baseada em hierarquia de chaves é composta por dois níveis de criptografia. No nível interno utiliza a criptografia simétrica baseada na ANSI X.9.17 [66]. No nível externo, utiliza a criptografia simétrica para as entidades da IoT, e criptografia assimétrica para as entidades na Internet. A abordagem baseada em OTP é similar a baseada em hierarquia de chaves, entretanto no nível interno de criptografia utiliza-se o conceito de senhas descartáveis. Essa abordagem emprega comunicação *multicast* com objetivo de reduzir a quantidade de mensagens trafegadas. Assim, as senhas (chaves) produzidas pelo OTP são utilizadas como chave de grupo no momento da requisição. E a chave produzida pelo TOTP é utilizada na resposta para o sistema central. O próximo capítulo apresenta uma comparação entre as duas abordagens.

Em razão desses protocolos serem integrados ao MIAM, é possível utilizar suas funcionalidades como autenticação única (SSO) e autenticação multifator. Assim, um usuário autenticado no MIAM pode acessar diversos SM com a mesma autenticação (desde que tenha autorização).

Capítulo 6

Resultados

Este capítulo apresenta os resultados da proposta que incluem a implementação e avaliação da proposta. O protótipo tem como objetivo implementar o mecanismo de MIAM integrado ao mecanismo que garante a segurança fim-a-fim na comunicação das entidades. Este capítulo detalha a especificação e as bibliotecas utilizadas na implementação do protótipo e no cenário de implantação. São realizadas duas avaliações, uma de desempenho e outra de segurança. A avaliação de desempenho tem o objetivo de comparar as abordagens de comunicação fim-a-fim no contexto de IoT. A avaliação de segurança é baseada no modelo do adversário e em um processo de *pentest*. Os resultados da avaliação permitem identificar as vulnerabilidades da proposta e treinar o modelo de detecção do iID.

6.1. Protótipo

O protótipo foi desenvolvido com a utilização de padrões, tecnologias consolidadas e bibliotecas de código aberto. As subseções a seguir detalham a implementação dos principais mecanismos da proposta.

6.1.1. MIAM

A implementação do MIAM utilizou como base a plataforma *WSo2 Identity Server*[1], que é um servidor de gestão de identidade e acesso que permite a utilização de diversos

[1] https://wso2.com/identity-and-access-management

protocolos, como: *OpenID*, *OpenID Connect*, *SAML*, *Passive STS* etc. Para realizar a autenticação, o protocolo *OpenID Connect* (seção 2.1.5) foi configurado como IdP no WSo2. O protocolo *OAuth 2.0*, presente no *OpenID Connect* é responsável pela autorização de acesso que emite *tokens de acesso* no formato JWT (*JSON Web Token*[1]). Foi necessário modificar a plataforma do *WSo2 Identity Server* para permitir que o escopo de um *token de acesso* possua mais de 60 caracteres. Essa ação foi indispensável porque o *token de acesso remoto* possui os papéis ativos do usuário e sua chave pública (seção 5.4).

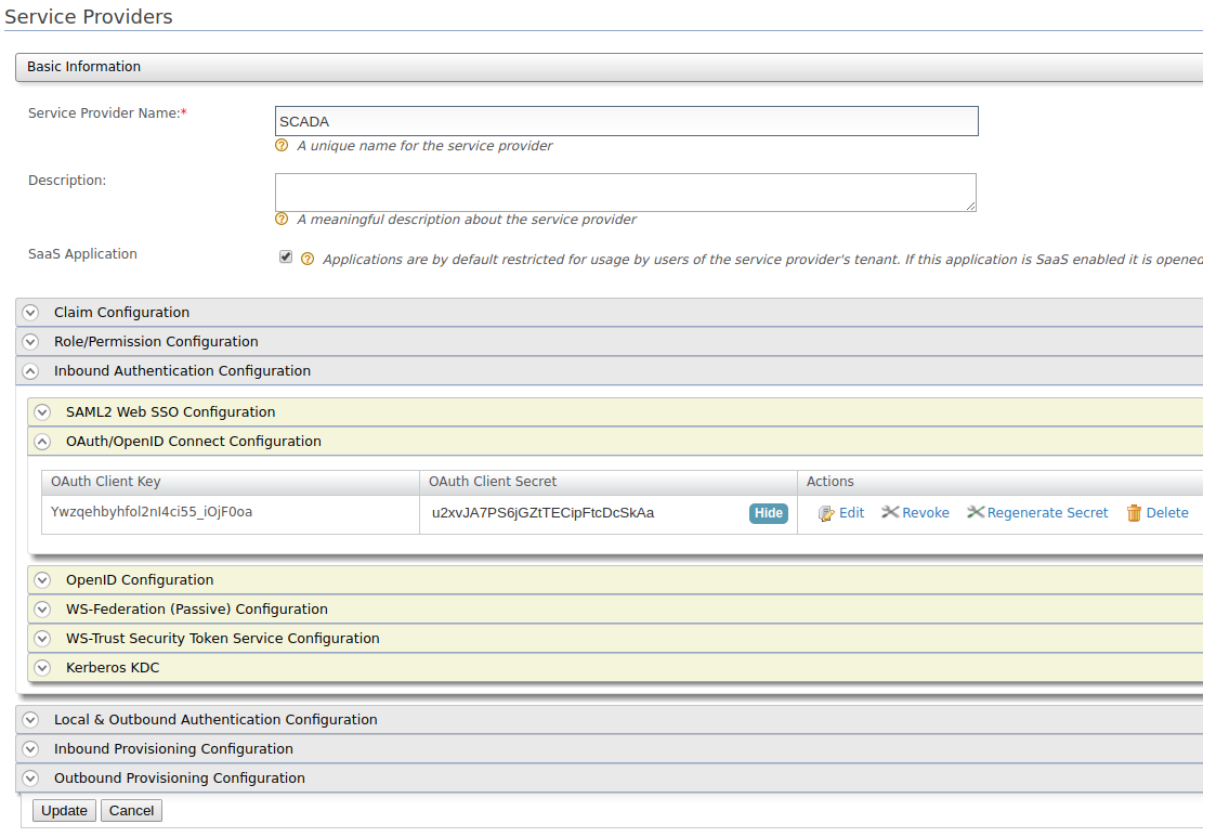

Figura 6.1. Criação de um provedor de serviços no *WSo2 Identity Server*.

Foram criados diversos SP's em cada *WSo2 Identity Server*, que são responsáveis por disponibilizar os serviços de segurança da arquitetura. Para estabelecer a federação de identidades, cada domínio possui seu próprio *WSo2 Identity Server* que contém outros WSo2 *Identity Server* cadastrados como IdP. A Figura 6.1 apresenta a tela de configuração de um SP

[1] https://jwt.io/

no *WSo2 Identity Server*. Note que foi configurada a autenticação como OpenID Connect, na qual é gerada uma chave e senha para autenticar o SP.

Relacionado à criptografia, cada WSo2 *Identity Server* possui um certificado que foi importado no repositório de chaves do Java (*KeyStore*) utilizando a ferramenta *keytool*[1]. Para cada usuário do domínio, foi gerado um par de chaves pública e privada utilizando a biblioteca de segurança nativa do *Java* com o algoritmo RSA (chaves de 4096 bytes). Toda a comunicação ocorre via HTTPS utilizando certificados autoassinados gerados pelo *keytool*.

Para implementar o MFA foi necessário adicionar métodos de autenticação no WSo2 utilizando o FIDO[2] (*Fast IDentity Online*). O FIDO define um mecanismo aberto, escalável e interoperável para realizar a autenticação segura de usuários em *web services*. Para implementar o método de autenticação por SQRL, foi utilizado a linguagem Python com a biblioteca pyqrcode[3] para disponibilizar graficamente o QRCode. Para implementar o método de OTP, foi utilizada a biblioteca pyotp[4] e flask[5]. Para implementar o método baseado em wireless do DLA_dC, um simulador de *beacon* foi desenvolvido na plataforma Android, utilizando o protocolo Eddystone-EID[6].

Os serviços de segurança foram implementados como *web services* RESTful, utilizando a biblioteca *Jersey*[7], hospedados no apache Tomcat 8.0[8]. *Jersey* implementa a especificação JAX-RS[9], que define diretrizes para implementação de serviços RESTful em Java.

---

[1] https://docs.oracle.com/javase/6/docs/technotes/tools/windows/keytool.html

[2] https://docs.wso2.com/display/IS510/Multifactor+Authentication+using+FIDO

[3] https://pypi.org/project/PyQRCode/

[4] https://github.com/pyauth/pyotp

[5] http://flask.pocoo.org/

[6] https://developers.google.com/beacons/eddystone-eid

[7] https://jersey.github.io/

[8] http://tomcat.apache.org/

[9] https://docs.oracle.com/javaee/6/tutorial/doc/giepu.html

A arquitetura de avaliação do ABAC foi implementada com a utilização da biblioteca WSo2 Balana[1], biblioteca em Java baseada na conhecida biblioteca sun-xacml[2], que suporta a versão 3.0 do XACML. Para a integração e funcionamento da arquitetura, o PIP foi estendido para coletar os papéis ativos do usuário no RC e M_dAC. Por padrão, o WSo2 Balana realiza cache dos valores dos atributos que o PIP fornece. Entretanto, como os papéis dos usuários são voláteis, tornou-se necessário limpar o cache do PIP toda vez que um usuário ativa/desativa um papel. Dessa forma, garantindo que o PIP sempre forneça os papéis que estão realmente ativos.

Para implementar o M_dAC, foi necessário desenvolver um gerenciador de políticas XACML que escreve políticas no PAP e publica políticas no PDP. A Figura 6.2 apresenta um exemplo de políticas disponíveis no PDP. Note que existem políticas locais e as políticas dinâmicas criadas pelo M_dAC.

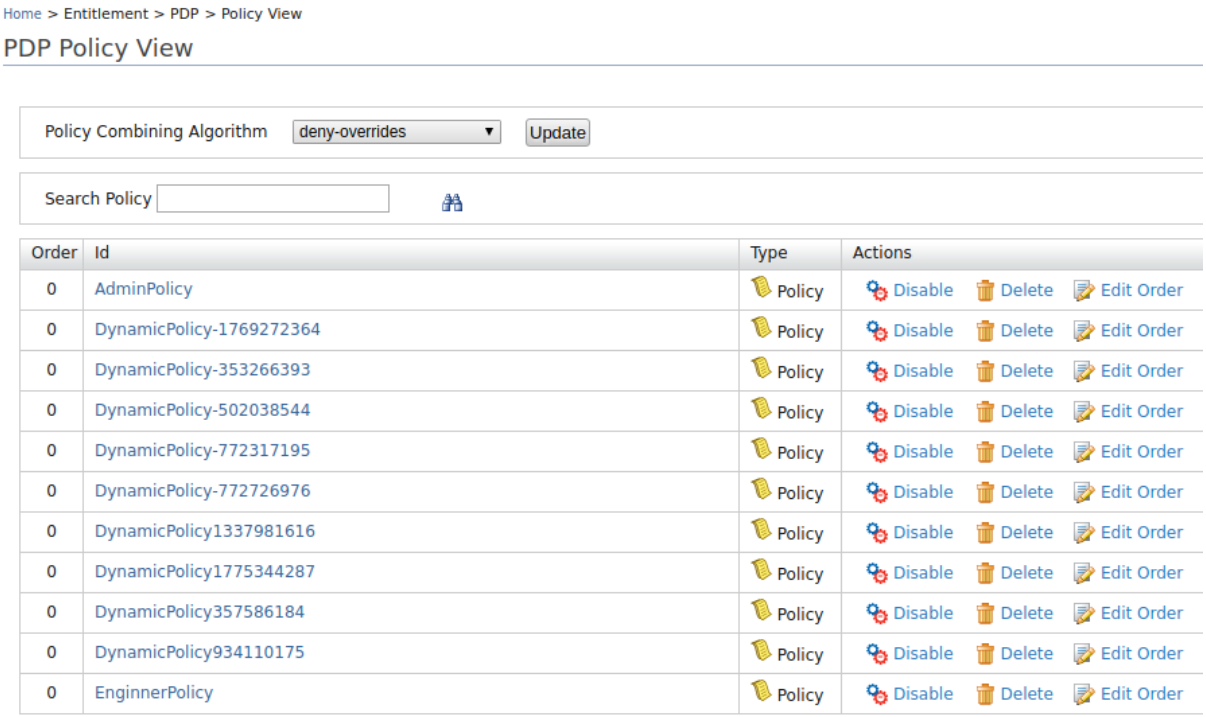

Figura 6.2. Políticas disponíveis no PDP.

[1] https://github.com/wso2/balana

[2] http://sunxacml.sourceforge.net/

6.1.2. *Segurança fim-a-fim na IoT*

A Figura 6.3 apresenta a arquitetura de IoT proposta ressaltando a pilha de protocolos utilizada em cada uma das entidades para garantir a comunicação segura. Na perspectiva da Internet, é utilizado o HTTPS. Já na perspectiva da IoT, é utilizado o CoAP (Constrained Application Protocol) [69] com o DTLS (*Datagram Transport Layer Security*), que é popularmente chamado de CoAPs [70]. O protocolo CoAP é baseado na arquitetura REST, onde os recursos são acessados a partir de uma URL. O CoAP utiliza o UDP na camada de transporte, pois é um protocolo adequado aos dispositivos com poucos recursos computacionais.

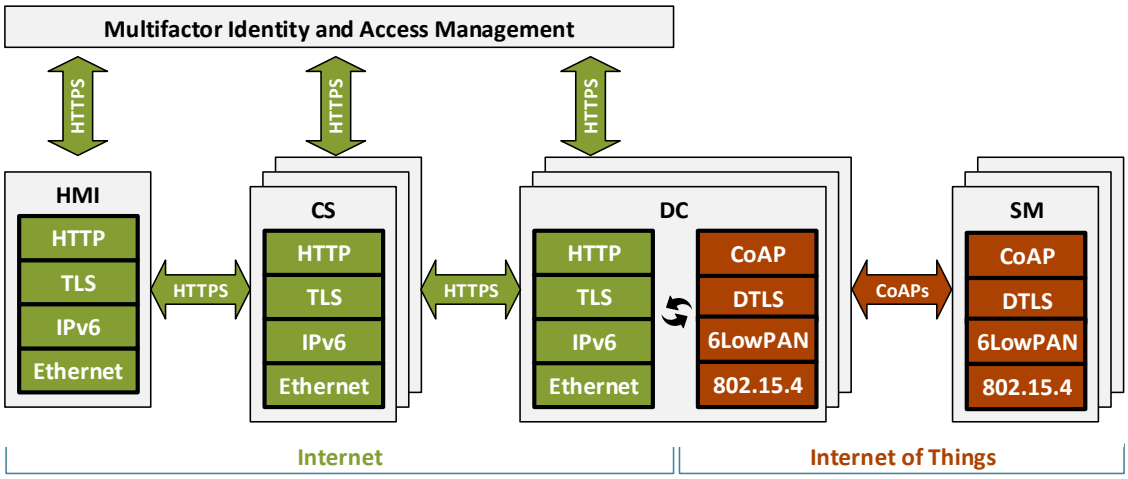

Figura 6.3. Visão geral da pilha de protocolos de rede.

Na abordagem baseada em hierarquia de chaves, o DC foi implementado em Java. Sua interface com a Internet utiliza um servidor HTTP, enquanto a interface com a IoT utiliza a biblioteca Californium[1] que disponibiliza um servidor CoAP. Dessa maneira, o DC pode converter as mensagens entre HTTP e CoAP, e vice-versa. O SM foi implementado em Java, utilizando também a biblioteca Californium, e executado no ContikiOS[2]. Foi utilizado o algoritmo AES com chaves de 128 bits para criptografar a KEK e a DK. A biblioteca

[1] https://www.eclipse.org/californium/

[2] http://www.contiki-os.org/

Scandium[1], pertencente a um subprojeto da Californium, foi utilizada para realizar o DTLS versão 1.2.

Para implementar a abordagem baseada em OTP, foi utilizada a linguagem Python. Para realizar a geração de senhas descartáveis, foi utilizada a biblioteca PyOTP, e a biblioteca Flask para disponibilizar serviços REST.

6.2. Modelo de Adversário

O modelo de adversário é um procedimento para avaliar os riscos da proposta. A proposta é baseada em uma combinação de mecanismos que garantem a segurança durante o acesso a recursos protegidos. A combinação de mecanismos de segurança tem objetivo de garantir a segurança da arquitetura mesmo que uma entidade específica seja comprometida. Nesse cenário, considere que foi explorada a vulnerabilidade CVE-2016-8735[2] presente no Apache Tomcat 8.0 e versões anteriores. Essa vulnerabilidade permite a execução de um código remoto. A tabela 6.1 apresenta o possível impacto no sistema caso essa vulnerabilidade seja explorada em cada uma das entidades. A tabela considera que as entidades são implementadas de maneira distribuída, através do conceito de microsserviços [79]. Dessa maneira, essas entidades estão hospedadas em sistemas operacionais distintos (Máquinas Virtuais, Containers etc.). O modelo de adversário proposto tem como premissa que o adversário consegue controlar somente uma entidade por vez.

[1] https://github.com/eclipse/californium.scandium

[2] https://www.cvedetails.com/cve/CVE-2016-0714/?q=CVE-2016-0714

Entidade	Impacto caso seja comprometida
MFA	O adversário não consegue se autenticar nos SPs porque não possui o *token de proximidade* emitido pelo DLA_dC.
RC	O adversário consegue ativar papéis, entretanto não consegue modificar políticas de controle de acesso, pois estão armazenadas no ABAC.
M_dAC	O adversário pode ativar papéis exportados, porém não pode forjar novas políticas de acesso, pois dependem de chaves criptográficas. Além disso, não consegue modificar políticas de controle de acesso existentes, pois estão armazenadas no ABAC.
ABAC	O adversário pode modificar as políticas de controle de acesso para escalar privilégios. Entretanto, não pode realizar operações, pois não possui papel ativo, *token de acesso*, chave privada e endosso do quórum.
CA_dC	O adversário possui acesso às chaves públicas, porém não consegue realizar operações, pois não possui papel ativo, *token de acesso*, chave privada e endosso do quórum.
QA_dC	O adversário pode forjar o endosso do quórum, porém não consegue realizar operações, pois não possui papel ativo, *token*, chave privada.
DLA_dC	O adversário pode forjar o *token de proximidade*, porém não consegue se autenticar no MFA, pois não tem credenciais válidas.
iID	O adversário pode passar pela detecção de intrusão, porém não possui as credenciais e autorizações exigidas pelo MIAM.
SM	O adversário tem acesso à LMK, que é armazenada no SM. O adversário não consegue injetar código malicioso no CS porque não possui sessão interativa. O adversário consegue apenas responder a requisições previamente armazenadas no DC. Idealmente esse dispositivo também poderia possuir proteções físicas [42].
DC	O adversário não consegue acessar ao conteúdo confidencial das mensagens porque estão cifradas na DK ou TOTP.
CS	O adversário tem acesso à LMK de todos os SMs. Entretanto, o adversário não consegue acessar, pois não tem o *token de acesso*.
HMI	O adversário não consegue acessar nenhum SP, pois não possui *token de acesso* e chave privada.

Tabela 6.1. Modelo de Adversário

É possível observar que o *token de acesso* e a chave privada são elementos críticos na proposta. Conforme discutido anteriormente, o *token de acesso* tem a finalidade de identificar e acessar informações protegidas do usuário sem a necessidade de acessar suas credenciais. Já a chave privada é utilizada para garantir a confidencialidade e autenticidade do usuário no momento do acesso remoto a recursos protegidos. Ou seja, ambos os elementos são críticos na proposta. O mecanismo de segurança proposto tem o objetivo de permanecer íntegro e confidencial mesmo que um desses elementos seja comprometido.

Apesar da proposta armazenar de maneira segura esses elementos, armazenando a chave privada na KeyStore[1] do Java e o *token de acesso* em formato JWT, é possível que o adversário obtenha um desses elementos. Dessa maneira, a tabela 6.2 apresenta uma avaliação do impacto na operação multidomínios caso o adversário possua um, e somente um, desses elementos. É importante ressaltar que a proposta adota o conceito de papéis temporários que precisam ser ativados. Dessa maneira, para realizar a operação multidomínio em um recurso protegido, o usuário deve realizar os seguintes passos:

1. Ativar papéis locais;
2. Obter o *token de acesso* remoto;
3. Obter os papéis exportados cifrados;
4. Decifrar os papéis exportados;
5. Assinar digitalmente o papel que deseja utilizar;
6. Ativar o papel remotamente;
7. O acesso remoto.

Elemento Comprometido	Adversário pode acessar	Adversário não pode acessar
Token de Acesso	1,2,3	4,5,6,7
Chave privada		1,2,3,4,5,6,7

[1] https://docs.oracle.com/javase/8/docs/api/java/security/KeyStore.htm

Tabela 6.2. Avaliação de impacto considerando o *token de acesso* ou a chave privada esteja comprometida.

6.3. Avaliação da comunicação fim-a-fim em IoT

Esse trabalho apresentou duas abordagens que garantem a comunicação fim-a-fim entre entidades da Internet e da IoT. Essa seção avalia o desempenho e a segurança das duas abordagens. Ambas as abordagens adotam dois níveis de criptografia para evitar as vulnerabilidades conhecidas do DTLS e TLS [71]. Dessa maneira, ambas abordagens utilizam o DTLS para proteger a camada interna. A Figura 6.4 apresenta a comparação entre requisições CoAP e CoAPs (com DTLS), em que o eixo Y apresenta o tempo de resposta em milissegundos, e o eixo X apresenta o tamanho da requisição em bytes. Esse consumo está relacionado ao procedimento de autenticação, comum às duas abordagens. Observe que o impacto decorrente do DTLS é pequeno em relação às vantagens do ponto de vista de segurança.

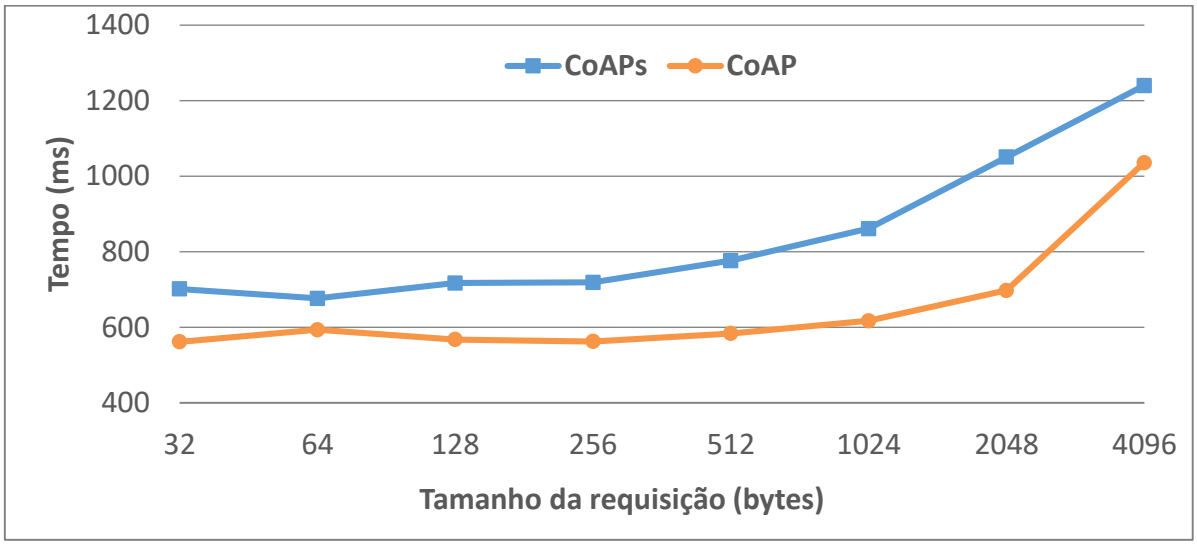

Figura 6.4. Comparação de requisições CoAPs e CoAP.

Na abordagem baseada em hierarquia de chaves (seção 5.5.1), a comunicação é realizada de maneira desacoplada e assíncrona. Dessa maneira, o CS não consegue injetar informações diretamente no SM, apenas deposita informação no DC. Tal abordagem mitiga a possibilidade de o adversário injetar código malicioso ou controlar o SM, uma vez que não possui sessão interativa.

Para estabelecer a comunicação, o SM frequentemente consulta o DC, verificando se existem requisições pendentes. Da mesma maneira, após o CS realizar uma requisição no DC,

frequentemente consulta pela resposta de sua requisição baseado no conceito de ticket. Esse desacoplamento entre entidades amplifica a robustez da solução. Por outro lado, aumenta significativamente o número de mensagens trafegadas, devido ao *pooling* realizado. Essa característica é agravada quando o contexto de IoT é considerado, pois possui restrições de largura de banda e bateria. Além disso, para o CS e o SM obter a DK, diversas mensagens são necessárias, conforme mostraram as figuras 5.11, 5.12 e 5.13. Considerando um contexto cujo o número de SM é elevado, a situação se agrava ainda mais, uma vez que a comunicação é *unicast*.

A abordagem baseada em OTP possui três principais vantagens em relação a abordagem baseada em hierarquia de chaves. Primeiramente, reduz o número de mensagens trocadas utilizando a comunicação *multicast* entre o CS e os SMs. É importante ressaltar que a abordagem mantém a segurança da comunicação utilizando criptografia de grupo nas requisições *multicast*. E as respostas das requisições são criptografadas em uma comunicação fim-a-fim utilizando chaves geradas pelo TOTP. Observe que a chave criptográfica utilizada na requisição (OTP) é mais fraca do que a chave criptográfica (TOTP) utilizada na resposta, uma vez que a resposta contém informações sensíveis.

Além disso, a abordagem baseada em OTP permite que a chave K_DLTS (Figura 5.14) seja facilmente atualizada, enquanto na abordagem baseada em hierarquia de chaves a K_DTLS deve ser definida manualmente. Finalmente, a abordagem baseada em OTP adiciona flexibilidade no processo, pois permite que o DC atue como *gateway* ou como concentrador. A tabela 6.3 apresenta a avaliação de parâmetros entre as abordagens, considerando *n* o número de SMs na comunicação.

Característica	Hierarquia de Chaves	OTP
Quantidade de mensagens	*n* * 13	6 + (*n**3)
Criptografia simétrica no contexto de IoT	*n* * 4	*n* * 4
Funções hash no contexto de IoT	0	n
Geração de chaves no contexto da IoT	*n* * 2	*n* * 3
Compartilhamento de *contador*	0	*n* * 2

Tabela 6.3. Comparação entre as abordagens de comunicação fim-a-fim.

A comparação do número de mensagens necessárias nas abordagens é apresentada na Figura 6.5. Observe que a relação do custo de processamento (número de cifras, geração de

chaves e funções hash) é semelhante entre as abordagens, sendo que a maior diferença entre as abordagens é apresentada quando o número de SM é elevado. É importante destacar, que apesar de uma arquitetura de SG tipicamente ser composta por milhões de SMs, cada DC não possui um número elevado de SM.

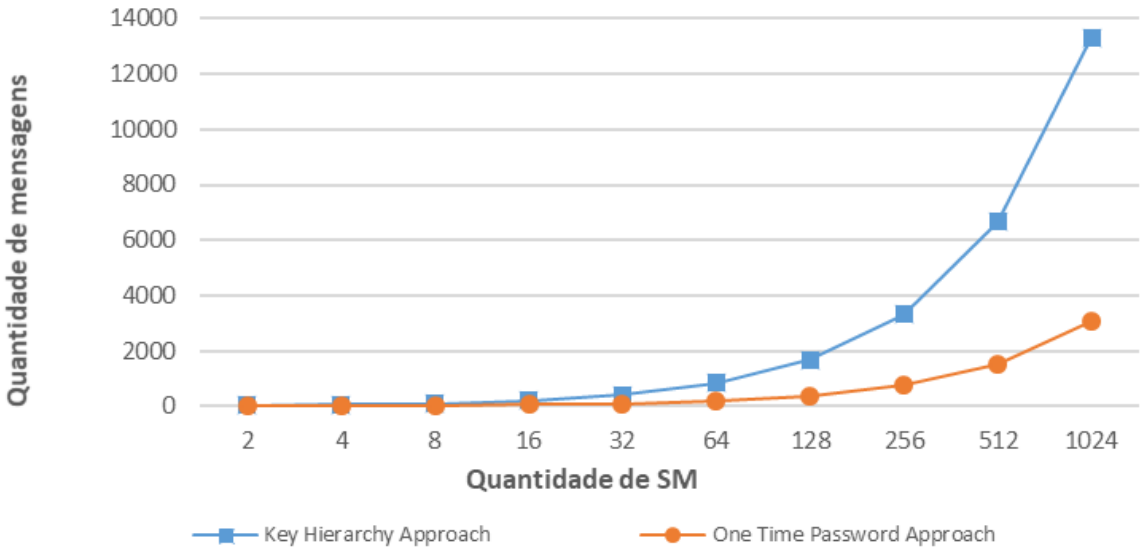

Figura 6.5. Comparação entre as abordagens de comunicação fim-a-fim.

Em termos de configuração inicial, a abordagem baseada em OTP possui mais parâmetros, entretanto são configurados de maneira dinâmica. Por outro lado, na abordagem baseada em hierarquia de chaves os parâmetros são definidos manualmente. A Figura 6.6 apresenta o tempo de configuração inicial da abordagem baseada em OTP. Essa configuração inclui a distribuição segura da chave DTLS (Figura 5.14).

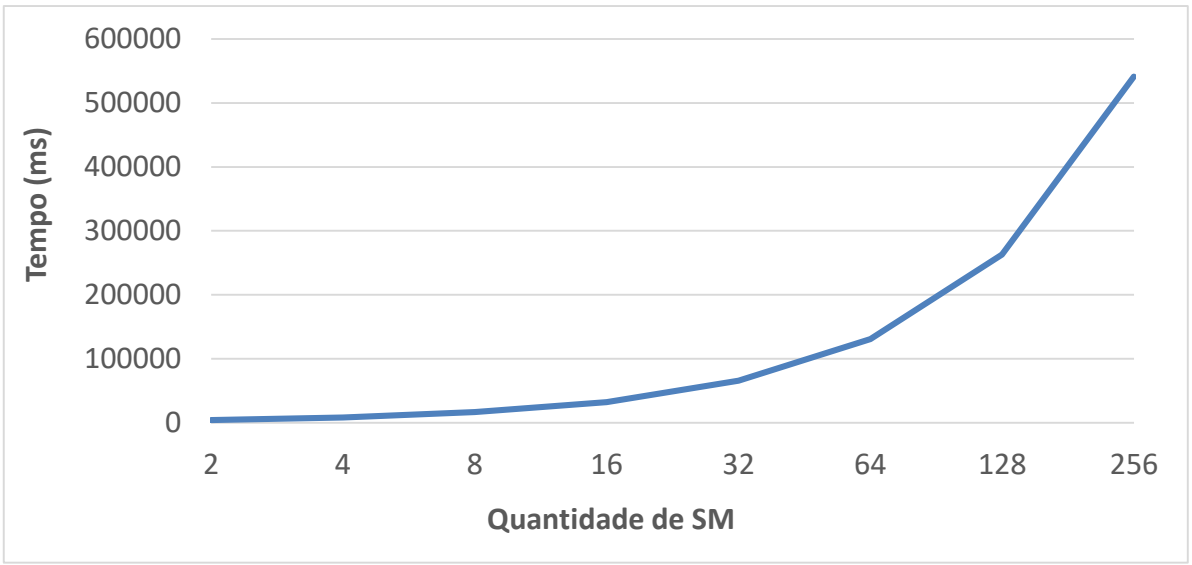

Figura 6.6. Configuração inicial da abordagem baseada em OTP.

6.4. Avaliação de segurança

A avaliação da proposta objetiva analisar a eficácia e eficiência do MIAM no contexto de SG. Conforme mencionado na seção 2.5.2 que explora *frameworks* da SG, não foi encontrado um *framework* disponível que contemple as características da SG. Dessa maneira, o presente autor que escreve considera que não é interessante modificar um *framework* para contemplar as características essenciais da SG, como: federação de identidades, operações multidomínios e conexão com protocolos da IoT para aplicar os mecanismos de segurança propostos. Essa modificação perde a referência de comparação, além do resultado ser questionável. É esperado que os mecanismos desenvolvidos sejam implantados em SG de maneira fácil e flexível, devido à adoção do conceito de microsserviços.

Por conseguinte, foram desenvolvidos processos que realizam a orquestração dos serviços de segurança, com o intuito de imitar o comportamento imprevisível do usuário. Esses processos foram definidos baseados nos diagramas de sequência presentes no capítulo 5. Dessa forma, foram criados dois cenários que serão explorados nas próximas subseções.

6.4.1. Cenário normal

Para avaliar a proposta, foi construído o cenário normal, que possui três processos que automatizam os principais procedimentos dos usuários. Esses processos têm comportamento imprevisível, pois efetuam requisições aleatórias, porém válidas e reais, a um conjunto de serviços, papéis, ações e recursos. O conjunto é composto de 03 domínios, cada um com 100 usuários, 10 papéis, 10 recursos e 10 ações. Assim, esse programa de *workload* implementa interações pseudoaleatórias com o mecanismo de MIAM. Os processos são detalhados a seguir.

A) Processo de acesso ao recurso local (Figura 5.1, 5.2, 5.3 e 5.4)

1) Seleciona de maneira aleatória um domínio federado. Por exemplo: Nacional;
2) Seleciona de maneira aleatória um usuário do domínio da Nacional. Por exemplo: Bob;
3) Seleciona de maneira aleatória um SP do domínio da Nacional. Por exemplo: *Dashboard*;
4) Bob solicita a autenticação para acessar a aplicação *Dashboard*;

5) Bob é redirecionado para o DLA_dC;
6) Bob verifica se está próximo de algum dispositivo âncora.
7) Bob recebe a credencial de proximidade emitida pelo dispositivo âncora. Por exemplo: *Beacon014*;
8) Bob valida a credencial de proximidade no DLA_dC, recebendo um *nonce*.
9) Bob fornece o *nonce* ao DLA_dC para comprovar sua proximidade e recebe o *token de proximidade*;
10) Bob solicita a autenticação para acessar à aplicação *Dashboard*, forcendo o *token de proximidade*;
11) Bob é redirecionado para a página de autenticação no MIAM;
12) Bob fornece suas credenciais para o MIAM, e recebe um *nonce* caso as credenciais sejam válidas;
13) Bob fornece o *nonce* ao *Dashboard* para comprovar sua autenticação, e recebe o *token de identidade e acesso*;
14) Bob solicita ao RC os papéis que estão associados à sua conta, fornecendo o *token de acesso*;
15) Bob seleciona de maneira aleatória um papel, fornecendo o *token de acesso*. Por exemplo: Operador;
16) Bob solicita ao RC a ativação do papel, fornecendo o *token de acesso*;
17) Bob seleciona aleatoriamente uma operação e um recurso que deseja acessar, fornecendo o *token de acesso*. Por exemplo: consumo da subestação 3;
18) Bob solicita o consumo da subestação 3 ao *Dashboard*, fornecendo o *token de acesso*;
19) O *Dashboard* solicita a decisão de acesso ao ABAC, que pode permitir ou negar o acesso.

B) Processo de Exportação de papel (Figura 5.5)

1) Seleciona de maneira aleatória um domínio federado. Por exemplo: Regional.

2) Seleciona um usuário do domínio Regional de maneira aleatória. Por exemplo: Alice;

3) Seleciona um papel associado a Alice de maneira aleatória. Por exemplo: Engenheira;

4) Seleciona de maneira aleatória os recursos e ações que serão exportados do papel Engenheira. Por exemplo: consumo da subestação;

5) Seleciona de maneira aleatória um domínio remoto. Por exemplo: Nacional;

6) Alice, engenheira da Regional, solicita ao seu M_dAC a exportação parcial do seu papel de Engenheira, permitindo o consumo da subestação para o domínio da Nacional;

7) O M_dAC da Regional solicita a AM da Nacional o endereço do RC da Nacional;

8) O M_dAC da Regional solicita os papéis disponíveis ao RC da Nacional, informando seu *token de acesso*;

9) Alice seleciona de maneira aleatória um papel do domínio da Nacional. Por exemplo: Operador;

10) O M_dAC cria um papel dinâmico com as permissões definidas por Alice;

11) O M_dAC associa o papel dinâmico com Engenheiro de Regional e Operador da Nacional;

C) Processo de acesso ao recurso remoto (Figura 5.6, 5.7, 5.8 e 5.9)

1) Seleciona de maneira aleatória um domínio federado. Por exemplo: Nacional.

2) Seleciona um usuário do domínio Nacional de maneira aleatória. Por exemplo: Bob;

3) Seleciona um papel associado ao Bob de maneira aleatória. Por exemplo: Operador;

4) Seleciona de maneira aleatória um domínio remoto. Por exemplo: Regional;

5) Bob, operador da Nacional, precisa de um *token de acesso remoto* para acessar o M_dAC da Regional;

6) Bob solicita ao seu AM um *token de acesso* emitido pela Regional;

7) O AM obtém os papéis ativos de Bob no RC e a chave pública em seu repositório de chaves;

8) O AM da Nacional realiza a autenticação única no IdM da Regional, através da recuperação da sessão;
9) O AM da Nacional solicita o *token de acesso remoto* para o AM da Regional, fornecendo os papéis ativos e a chave pública de Bob;
10) Bob solicita ao M_dAC da Regional os papéis que estão exportados para seu papel de Operador, fornecendo o *token de acesso remoto*;
11) O M_dAC da Regional valida o *token de acesso* no AM da Regional;
12) O M_dAC da Regional extrai do *token de acesso:* usuário, papéis ativos e a chave pública de Bob;
13) O M_dAC da Regional obtém os papéis exportados associados ao papel de Operador;
14) O M_dAC cifra os papéis exportados na chave pública de Bob;
15) Bob identifica que existe um papel na Regional que pode utilizar, porém ele precisa ativá-lo antes;
16) Bob seleciona de maneira aleatória um papel exportado (dinâmico) que pode utilizar;
17) Bob assina o papel dinâmico e solicita a ativação remota para o M_dAC da Regional, fornecendo o *token de acesso remoto*;
18) O M_dAC da Regional valida o *token de acesso* e a assinatura do papel;
19) O M_dAC da Regional verifica se existe algum Engenheiro com o papel ativo. Se existir, ativa o papel exportado;
20) Finalmente, com o papel ativo, Bob pode solicitar o consumo da subestação da Regional para o ABAC da Regional;
21) O ABAC da Regional valida o *token de acesso* no AM da Regional;
22) O ABAC da Regional obtém os papéis ativos no M_dAC da Regional;
23) O ABAC da Regional avalia se existe uma política que permite o consumo da subestação;
24) Se o resultado for permitido, o ABAC solicita ao QA_dC uma operação de quórum para os usuários que estão com o papel Engenheiro ativo;

25) Bob acessa o consumo da subestação da Regional, se houver o endosso do QA_dC;

É importante ressaltar que os processos podem solicitar operações inválidas, que devem ser negadas pelos mecanismos de segurança. Dessa maneira, é possível executar diversas instâncias dos processos em paralelo para imitar o comportamento de diversos usuários utilizando o sistema. É esperado que seja possível avaliar a eficácia e eficiência dos mecanismos de segurança utilizando esses processos.

6.4.2. Cenário de ataque

O cenário de ataque possui dois objetivos: (i) identificar vulnerabilidades da proposta; (ii) treinar e avaliar o iID. Para identificar possíveis vulnerabilidades da proposta é utilizado um processo de *pentest*. Esse processo tem como objetivo utilizar ferramentas de análise de vulnerabilidade e o conhecimento de um especialista em *pentest* para identificar vulnerabilidades da proposta baseando-se no modelo de adversário (seção 6.2). Dessa maneira, é possível utilizar o tráfego normal (seção 6.4.1) e o tráfego de ataque (durante o *pentest*) para treinar o classificador que detecta se um evento é normal ou ataque.

O cenário de ataque é composto por três processos: análise de vulnerabilidades, testes de intrusão caixa branca, interceptações *main-in-the-midle* (MITM). Esses testes têm como o objetivo identificar e explorar as vulnerabilidades da proposta, gerando um guia para mitigar possíveis intrusões. Além disso, foi utilizado esse cenário de ataque para treinar o modelo de detecção de intrusão.

Um ambiente controlado foi desenvolvido, no sentido de prevenir possíveis interferências nas medições. Um total de quatro máquinas físicas foram conectadas em rede Gigabit. Todas as máquinas físicas possuem a mesma configuração de hardware, com processadores core i7 e 8 GB memória RAM. Duas máquinas executaram o sistema operacional Ubuntu[1] x64 versão 17.04, com o Java 1.8 e Tomcat 9. Uma máquina executou o sistema

[1] https://www.ubuntu.com/

operacional Kali Linux[1] versão 2018.1 para realizar os testes de intrusão. Finalmente, uma máquina executou o sistema operacional Windows 10 para também realizar testes de intrusão.

Para realizar a análise de vulnerabilidades, oito ferramentas comerciais e de código aberto foram utilizadas (Acutenix[2], Dependecy Check[3], Nessus[4], Nexpose[5], Nikto[6], WSSAT[7], TestSSL[8] e SoapUI[9]). Tais ferramentas de análise de vulnerabilidades, conceituadas pela comunidade, otimizam a execução dos testes, uma vez que não necessitam de interação humana. Entretanto, os resultados gerados por essas ferramentas geram um número elevado de FP, situação na qual a ferramenta aponta erroneamente a presença de vulnerabilidade, e vice-versa (FN). A tabela 6.4 apresenta as vulnerabilidades identificadas pelas ferramentas, classificadas de acordo com a OWASP Top Ten do ano de 2017 [72]. Essa classificação elaborada pela OWASP, entidade internacional de segurança de *software*, apresenta os dez riscos de segurança mais recorrentes de 2017.

[1] https://www.kali.org/

[2] https://www.acunetix.com/

[3] https://www.owasp.org/index.php/OWASP_Dependency_Check

[4] https://www.tenable.com/downloads/nessus

[5] https://www.rapid7.com/products/nexpose/

[6] https://cirt.net/Nikto2

[7] https://github.com/YalcinYolalan/WSSAT

[8] https://github.com/drwetter/testssl.sh

[9] https://www.soapui.org/

OWASP TOP 10	Acunetix	Dependency Check	Nessus	Nexpose	Nikto	WSSAT	TestSSL	SoapUI
(A1) Injeção de Código								X
(A2) Autenticação vulnerável								
(A3) Exposição de dados sensíveis	X		X	X	X	X	X	
(A4) XML externa a entidade (XXE)								X
(A5) Controle de Acesso vulnerável								
(A6) Configuração inadequada de segurança	X		X		X	X		X
(A7) Cross-Site Scripting (XSS)								X
(A8) Desserialização Insegura								
(A9) Utilização de componentes com vulnerabilidades conhecidas	X	X	X	X			X	
(A10) Monitoramento e Registro de informações insuficientes								

Tabela 6.4. Análise de vulnerabilidades classificadas pela OWASP Top Ten [72].

A análise de vulnerabilidade identificou vulnerabilidades presentes no código fonte e na infraestrutura, tanto do lado do cliente quanto do servidor. Observe na tabela 6.4 que o risco (A6) foi recorrente em várias ferramentas. Esse risco está relacionado à configuração inadequada de segurança, mais especificamente à divulgação de arquivos de configurações. Essa vulnerabilidade permite que um adversário acesse aos arquivos de configurações da aplicação, normalmente um arquivo web.xml. Assim, essa vulnerabilidade foi identificada na aplicação de boas vindas do Tomcat, que tem como objetivo notificar se o servidor está funcionando. É interessante ressaltar que muitos desenvolvedores não removem aplicações dessa natureza, permitindo que o servidor seja suscetível a uma intrusão devido a uma aplicação padrão (*default*) do servidor. Além dessa vulnerabilidade, outras vulnerabilidades estavam relacionadas a configurações padrões do WSo2 Identity Server e do Tomcat.

O relatório gerado pelas ferramentas Nikto e TestSSL não categorizam as vulnerabilidades identificadas por nível de severidade, apenas informaram a presença ou não das vulnerabilidades. A Figura 6.7 apresenta as vulnerabilidades identificadas por nível de severidade, sendo que o eixo Y representa as ferramentas utilizadas, e o eixo X representa o número de vulnerabilidades identificadas pelas ferramentas de acordo com sua severidade. Cada vulnerabilidade pode ser classificada de quatro maneiras: aviso, baixo risco, médio risco ou alto risco.

A ferramenta Dependency Check, que analisa dependências do código fonte (bibliotecas e componentes) e verifica se existem vulnerabilidades conhecidas (CVE, *Common Vulnerabilities and Exposures*) associadas ao código fonte, apresentou 81 vulnerabilidades classificadas como alto risco. Dentre essas 81 vulnerabilidades, 25 foram relacionadas a API de upload de arquivos que está presente no WsO2. Apesar do protótipo desenvolvido não realizar upload de arquivos, a biblioteca vulnerável está presente na arquitetura, permitindo uma possível violação de segurança.

Observe que nenhuma ferramenta apontou vulnerabilidades relacionadas à autenticação (A2) e ao controle de acesso (A5). Entretanto, não é possível assumir que não existem vulnerabilidades nesses mecanismos. Assim, testes de intrusão de caixa branca foram realizados por um profissional de *pentest*, que tinha conhecimento *a priori* da aplicação, infraestrutura, rede, roteadores, servidores e demais configurações. O teste de intrusão tem uma característica peculiar diferente da análise de vulnerabilidade. Essa diferença consiste em que o teste de intrusão tem como objetivo não somente identificar as vulnerabilidades, mas também

explorar com o intuito de obter acesso a recursos protegidos. O teste de intrusão seguiu a metodologia definida em [73].

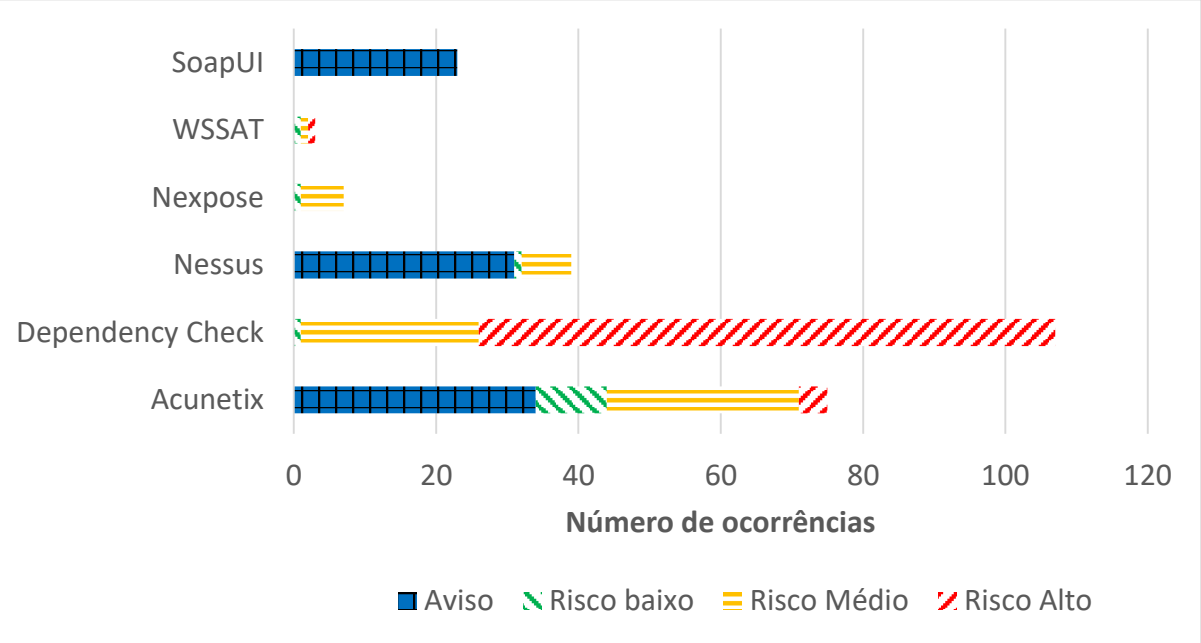

Figura 6.7. Análise de vulnerabilidades classificadas por nível de severidade.

Inicialmente foi utilizado o Nmap[1], ferramenta utilizada para mapear as portas abertas no servidor e os serviços em execução. Assim, foi possível identificar as versões das aplicações utilizadas pelo sistema. Tais vulnerabilidades não possibilitaram a invasão ou comprometimento do sistema, tratam-se de vulnerabilidades de baixa severidade, que permitem a obtenção de informações referentes às tecnologias empregadas no sistema, e que podem ser utilizadas para auxiliar o adversário em um futuro ataque.

Com relação à classificação proposta pela OWASP Top Ten, foram identificados três riscos: (A3) Exposição de dados sensíveis – as credenciais de usuários armazenadas no WsO2 estão em texto às claras (sem criptografia); (A6) Configuração inadequada de segurança – serviços com senha padrão de fábrica. Através dessas vulnerabilidades foi possível acessar a base de dados do WsO2 Identity Server, permitindo o acesso às credenciais de usuário, que são configurações do sistema, logs de acesso, inclusive aos *tokens de acesso* utilizados. Entretanto,

[1] https://nmap.org/

essas vulnerabilidades não permitiram a intrusão ou o comprometimento dos SPs, devido às características multifatores de autenticação, acesso e criptografia.

As intercepções do MITM consistem em "envenenar" a tabela ARP do cliente e servidor. Assim, os clientes acreditam que o adversário é o servidor, e o servidor acredita que o adversário é o cliente. Dessa maneira, todo o tráfego gerado pelo cliente e servidor é interceptado pelo cliente. Para realizar o teste, as seguintes ferramentas foram utilizadas: Ettercap[1], SSLstrip [2]and MITM framework[3]. Essas ferramentas permitiram interceptar o tráfego realizado pelo cliente e servidor. Entretanto, devido às características dos mecanismos de segurança, não foi possível visualizar o tráfego por conta da criptografia imposta.

6.4.3. Modelo de detecção de intrusão

O cenário normal e de ataque têm como objetivo identificar as vulnerabilidades da proposta e treinar o modelo de detecção de intrusão a fim de detectá-las. Para isso, foi necessário utilizar a ferramenta tcpdump[4] (libpcap), que foi responsável pela captura dos pacotes de rede e armazenamento em um arquivo no formato PCAP (*packet capture*). Esse tráfego capturado está criptografado, uma vez que a proposta utiliza SSL, dificultando a utilização dos pacotes para treinamento do modelo. Desta maneira, apesar de ferramentas como o Wireshark[5], que utiliza o libpcap, possuírem a funcionalidade de decifrar comunicações criptografadas, houve um grande esforço para conseguir configurar adequadamente a etapa de decifração dos pacotes.

Para tanto, primeiramente foi necessário adequar o formato do certificado digital, uma vez que o Wireshark aceita somente certificados no formato *pem* (*Privacy Enhanced Mail*), e o Tomcat aceita somente certificados no formato *jks* (*Java KeyStore*). Dessa maneira, todos os

[1] https://www.ettercap-project.org/

[2] https://www.guiadoti.com/2017/09/sslstrip-2-0-hsts-bypass/

[3] https://github.com/byt3bl33d3r/MITMf

[4] https://www.tcpdump.org/

[5] https://www.wireshark.org/

certificados precisaram ser atualizados. A ferramenta openssl[1] foi utilizada para gerar o certificado *pem* e a chave privada, e em seguida converter para jks. Após a configuração adequada do Wireshark, foi possível capturar o tráfego de maneira decifrada, utilizando a chave privada gerada pelo openssl. Outra dificuldade encontrada na avaliação foi em relação ao volume de informações capturadas, uma vez que as ferramentas de análise de vulnerabilidade e os testes de intrusão não geraram grande volume de tráfego de dados. Dessa maneira, para realizar a prova de conceito, foram coletadas 04 pcaps (base de dados), conforme apresenta a tabela 6.5.

Ferramentas	Tempo de coleta (em segundos)	Quantidade de pacotes
Normal	1574	346683
Acunetix	7020	617931
Nessus	1803	97281
Nikto	1622	135094

Tabela 6.5. Características da base de dados utilizada para treinamento e teste dos modelos.

A ferramenta BigFlow [74] foi utilizada para extrair as características dos pacotes de redes. Essa ferramenta considera um cenário de big data, característica comum no contexto de SG. Após a extração das características foram criados quatro *datasets* (detalhados na Tabela 6.5). Para cada ferramenta de *pentest* foram criados *datasets* de treinamento específicos. O objetivo é validar a capacidade de generalização dos modelos, uma vez que em produção não é possível garantir que o modelo foi treinado/testado com todos os eventos possíveis. Sendo assim, três modelos distintos foram gerados, um para cada ferramenta de pentest: *Acunetix, Nessus e Nikto*. Enquanto que os modelo obtidos são validados em todas as ferramentas, permitindo assim validar a capacidade de generalização.

Finalmente, devido ao desbalanceamento da quantidade de instâncias para cada ferramenta de *pentest* (cenário de treinamento), utilizou-se a abordagem de *random stratified undersampling*. Desta maneira, adotou-se a proporção de 60% das instâncias para treinamento,

[1] https://www.openssl.org/

enquanto que os restantes das instâncias são utilizadas para o teste. Adicionalmente, quando o cenário não é utilizado para treinamento, o mesmo é utilizado inteiramente para testes. Por exemplo, quando o modelo é obtido utilizando o cenário da ferramenta Nessus, apenas 40% das instâncias do Nessus são utilizadas para teste, enquanto que 100% das instâncias dos outros cenários são utilizados para teste. Finalmente, 60% das instâncias do cenário *Normal* são utilizadas para treinamento em todos os cenários.

O classificador *Random Forest* [75] foi escolhido para os testes, uma vez que é comumente utilizado por trabalhos correlatos no contexto de Big Data [74]. Para tanto, o classificador foi treinado através da ferramenta Weka[1], com 101 árvores de decisões e profundidade ilimitada por árvore. A tabela 6.6 exibe os valores da acurácia do classificador *Random Forest* para os cenários avaliados

Ferramenta utilizada para treinamento	**Cenário**	**Acurácia**	**Falso-Positivo**	**Falso-Negativo**
Acunetix	**Acunetix**	**99,21%**	**2,86%**	**0,28%**
	Nessus	97,21%		2,77%
	Nikto	96,61%		3,70%
Nessus	Acunetix	99,73%	**0,06%**	0,29%
	Nessus	**99,96%**		**0,03%**
	Nikto	98,87%		2,59%
Nikto	Acunetix	99,15%	**0,19%**	1,14%
	Nessus	99,07%		1,66%
	Nikto	**99,80%**		**0,43%**

Tabela 6.6. Acurácia do iID. Falso-positivo denota a taxa de eventos normais classificados como ataques. Falso-negativo denota a taxa de eventos ataques classificados como normais.

[1] https://www.cs.waikato.ac.nz/ml/weka/

É possível notar que em todos os cenários, a taxa de acerto permaneceu acima de 99%, para quando o classificador é avaliado com o mesmo comportamento evidenciado durante o treinamento. Adicionalmente, a taxa de FN nestes casos, é significativamente baixa, 0,43% no pior caso (*Nikto*). Nota-se ainda que o *Acunetix* apresentou a pior acurácia quando avaliado no próprio cenário de treinamento: 99,21%. Sendo assim, o classificador gerado é capaz de detectar comportamentos semelhantes à aqueles evidenciados durante a etapa de treinamento (exibido em negrito na tabela 6.6).

Por outro lado, quando o classificador é avaliado utilizando eventos desconhecidos, ou seja, ferramentas não utilizadas durante o treinamento, é possível notar um aumento na taxa de FN. Porém, o aumento na taxa de FN não é significativo, uma vez que, em média a detecção de ataques desconhecido acarreta em apenas 1,78% de aumento na taxa de FN. Sendo assim, nota-se que a abordagem proposta é capaz de detectar ataques desconhecidos ao modelo, com baixo impacto em sua acurácia.

6.5. Discussão

Esse capítulo dissertou o protótipo que foi desenvolvido utilizando padrões e tecnologias consolidadas. O MIAM foi construído sobre a plataforma do WSo2 Identity Server, e modificado para atender as peculiaridades da proposta. Em relação a IoT, foi avaliado o impacto da utilização do DTLS comparando diferentes tamanhos de pacotes de rede. Observou-se que o DTLS aumenta o tempo das requisições devido à criptografia, porém esse impacto é pequeno em relação às vantagens de segurança propiciadas. As abordagens baseadas em hierarquia de chaves e baseada em OTP foram comparadas. Apesar da hierarquia de chaves apresentar maior robustez do ponto de vista de segurança, a abordagem baseada em OTP possui vantagens quando o número de SM é elevado.

O modelo de adversário desenvolvido considerou o impacto dos mecanismos de segurança propostos serem comprometidos. Observou-se que o adversário não consegue comprometer o sistema, mesmo que tenha posse da chave privada ou do *token de acesso*.

Para avaliação da proposta, foram desenvolvidos dois cenários. O cenário normal apresentou um programa que imita o comportamento imprevisível do usuário, realizando requisições pseudoaleatórias nos serviços de segurança. Já o cenário de ataque utilizou ferramentas de análise de vulnerabilidades, testes de intrusão de caixa branca e interceptações

MITM com o objetivo de identificar e explorar as vulnerabilidades da proposta. Os testes identificaram vulnerabilidades que foram classificadas de acordo com sua severidade. Observa-se que as vulnerabilidades mais severas foram identificadas a partir de bibliotecas provenientes dos servidores WSo2 Identity Server e do Tomcat. Apesar dessas bibliotecas não terem sido utilizadas diretamente na construção do protótipo, as mesmas apresentam riscos que podem possibilitar violações de segurança.

Apesar de ser uma tendência natural que a SG adote tecnologias e protocolos abertos (*commododity*) ao invés de proprietários, é necessária uma análise profunda em relação às bibliotecas e às dependências associadas, pois podem comprometer todo o sistema apenas por estarem presentes nos serviços e servidores. Sugere-se aplicar um processo de *Hardening* [80] que mapeie as ameaças de toda a infraestrutura.

Finalmente, o modelo de detecção de intrusão criado foi treinado a partir da coleta dos pacotes de dados de ambos cenários apresentando acurácia acima de 99%, para quando é avaliado com o mesmo comportamento evidenciado durante o treinamento. Entretanto, é importante ressaltar que foi realizado apenas uma prova de conceito, e que o modelo idealmente deve ser treinado com base em registros de um ambiente de produção.

Capítulo 7

Conclusão

A SG provê inúmeros benefícios para a sociedade através do emprego de inteligência computacional e tecnologias de comunicação de maneira integrada. Entretanto, esses benefícios implicam em desafios de cibersegurança que requerem mecanismos mais eficientes no combate aos frequentes ciberataques. A literatura carece de mecanismos de cibersegurança adequados às características da SG. Esse trabalho apresentou o MIAM, que considera diferentes fatores de autenticação e acesso para mitigar violações de segurança na SG. Nesse sentido, caso uma entidade (fator) seja comprometida, o MIAM contínua integro.

O IdM combinou fatores de autenticação para atenuar as chances de sucesso de um adversário oriundo da Internet acessar à SG. Para utilizar esse mecanismo, o usuário precisa de uma credencial de proximidade emitida pelo controle de admissão baseado em localidade. O AM combinou políticas de controle de acesso baseadas em papéis com um controle de admissão criptográfico e de quórum para limitar o acesso, local e remoto, a um recurso protegido. O MIAM, que é composto pelo IdM e AM, tem como objetivo permitir que um usuário autenticado transporte suas credenciais para o contexto da IoT, mantendo a autenticidade e a confidencialidade na comunicação.

O MIAM foi aplicado em uma arquitetura típica de SG. No estudo de caso apresentado, foram consideradas as operações rotineiras da SG, que envolvem o uso de autenticação, administração de papéis e políticas de acesso. Além disso, o estudo de caso considerou as operações que envolvem a comunicação com dispositivos da IoT, como os medidores inteligentes.

Para ampliar a robustez da proposta, o MIAM foi protegido por um mecanismo de detecção de intrusão baseado em anomalias chamado de iID. O modelo de detecção de intrusão foi treinado a partir de um processo de *pentest,* que visa identificar as vulnerabilidades da proposta baseada no modelo de adversário.

A avaliação de segurança da proposta identificou as vulnerabilidades do protótipo, que foram classificadas de acordo com sua severidade. As ferramentas e técnicas de pentest que identificaram as vulnerabilidades foram utilizadas para treinar o modelo de detecção do iID que apresentou uma acurácia acima de 99%, quando o classificador é avaliado com o mesmo comportamento do treinamento. Além disso, foi possível observar que a proposta é capaz de detectar ataques desconhecidos ao modelo, com baixo impacto em sua acurácia.

O código fonte do trabalho está disponível no GitHub através do link https://github.com/vilmarabreujr/smartgrid.

7.1. Limitações da proposta

Esta seção tem como objetivo discutir algumas limitações do trabalho. O trabalho adotou o modelo federado de identidade, que integra diversos domínios. Entretanto, esse trabalho não discutiu como é possível estabelecer a relação de confiança entre os domínios. Apenas estabeleceu o relacionamento entre os IdPs de cada domínio.

A validação do token de acesso foi realizada de forma *online*. Essa abordagem tem como desvantagem a necessidade de consultar o AM a cada requisição. Seria possível adotar uma validação de *token de acesso off-line*, prevista na RFC 6749 [16]. Porém, essa alteração impacta diretamente na revogação do *token de acesso*.

Durante a execução do cenário normal foram criadas, em um único dia, milhares de papéis para avaliar a escalabilidade da proposta. Para que a avaliação desses papéis esteja em conformidade com o modelo RBAC, este trabalho adotou a estratégia de escrever cada papel em uma política, conforme discutido em [63]. A implementação do MIAM utilizou como base a biblioteca WsO2 Balana, que implementa o mecanismo de avaliação do XACML (PDP). Para avaliar se uma determinada requisição é válida, o PDP deve consultar todas as políticas vinculadas ao recurso desejado. Caso o número de políticas seja elevado, o tempo dessa avaliação pode ser impactante na arquitetura. Para solucionar essa limitação, seria possível descentralizar o PDP com o objetivo de aumentar o desempenho, e aplicar técnicas mais eficientes de processamento distribuído.

Finalmente, não foi realizado um aprofundamento nos testes relacionados a aprendizagem de máquina, no qual poderia ser testado diferentes classificadores para treinar o IDS. Além disso, seria possível considerar a especificação proposta no treinamento do IDS.

7.2. Publicações

Esse trabalho possui três publicações parciais de seus resultados, sendo que os periódicos que contemplam a proposta foram submetidos, estando em fase de revisão. A tabela 7.1 apresenta as publicações relacionadas a esse trabalho durante o período de doutoramento.

Periódico publicado
ABREU, VILMAR; SANTIN, ALTAIR; XAVIER, ALEX; LANDO, ALISON; WITKOVSKI, ADRIANO; RIBEIRO, RAFAEL; STIHLER, MAICON; ZAMBENEDETTI, VOLDI; CHUEIRI, IVAN. A Smart Meter and Smart House Integrated to an IdM and Key-based Scheme for Providing Integral Security for a Smart Grid ICT. MOBILE NETWORKS AND APPLICATIONS, v. 22, p. 1, 2017. **(Qualis A2)**
Conferências publicadas
ABREU, VILMAR; SANTIN, ALTAIR O.; VIEGAS, EDUARDO K.; STIHLER, MAICON . A multi-domain role activation model. In: ICC 2017 2017 IEEE International Conference on Communications, 2017, Paris. 2017 IEEE International Conference on Communications (ICC), 2017. p. 1. **(Qualis A1)**
Conferência aceita para publicação
ABREU, VILMAR; SANTIN, ALTAIR O.; VIEGAS, EDUARDO K.; STIHLER, MAICON . An end-to-end Secure Communication Protocol and a Lightweight Access Control for Cyber-Physical Systems. In: IFIP/IEEE International Symposium on Integrated Network Management (IM 2019). **(Qualis A2)**
Periódicos submetidos
Multi-Factor IAM for Smart Grid. Submetido em IEEE Transaction on Smart Grid. **(Qualis A1)**
Proximity-based Admission Control and Intelligent Intrusion Detection for IAM on Smart Grids. Submetido em IEEE Communication Magazine. **(Qualis A1)**

Tabela 7.1. Lista de publicações relacionadas a esse trabalho.

A Figura 7.1 apresenta a relação das publicações com a proposta de trabalho. O conceito de operações multidomínios utilizando RBAC e ABAC foi publicado no ICC (2017). A abordagem de hierarquia de chaves foi publicada em MONET (2017). A abordagem baseada em OTP foi aceita para publicação em IM (2019). O controle de admissão baseado em localidade foi submetido para o ISCC (2019).

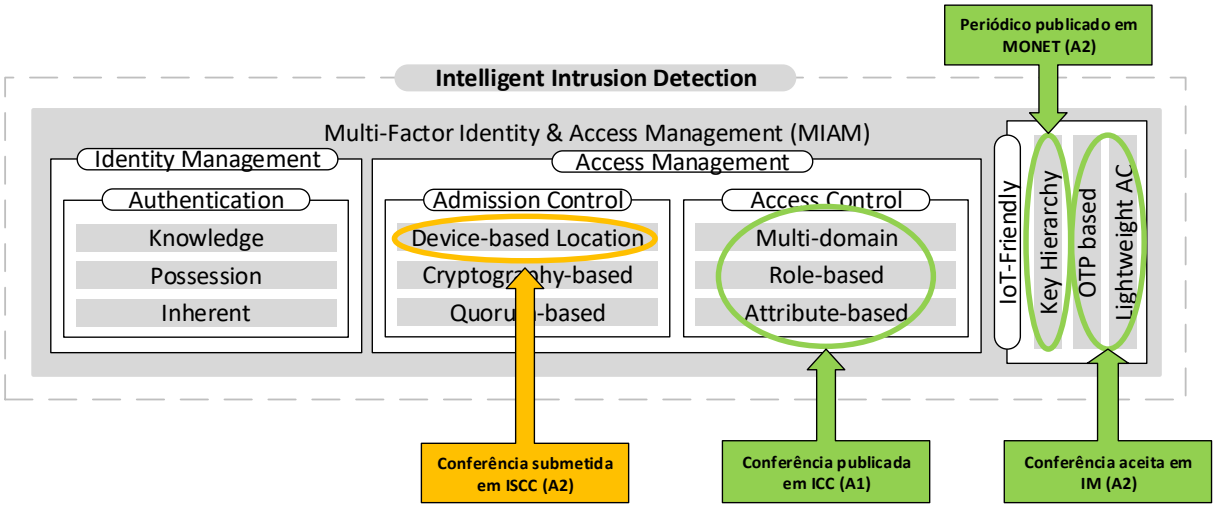

Figura 7.1. Publicações de artigos relacionados à proposta.

Referências

[1] X. Fang, S. Misra, G. Xue, and D. Yang, “Smart grid - The new and improved power grid: A survey,” IEEE Communications Surveys and Tutorials, vol. 14, no. 4. pp. 944–980, 2012.

[2] NIST, “Framework for Improving Critical Infrastructure Cybersecurity,” 2014.

[3] NIST, “Identity and Access Management for Electric Utilities,” 2014.

[4] A. Cárdenas, S. Amin, and S. Sastry, “Research Challenges for the Security of Control Systems,” Netw. Secur., p. 6, 2008.

[5] J. Lee, B. Bagheri, and H. A. Kao, “A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems,” Manuf. Lett., vol. 3, pp. 18–23, 2015.

[6] Y. Ashibani and Q. H. Mahmoud, “Cyber physical systems security: Analysis, challenges and solutions,” Comput. Secur., vol. 68, pp. 81–97, 2017.

[7] R. Langner, “Stuxnet: Dissecting a cyberwarfare weapon,” IEEE Secur. Priv., vol. 9, no. 3, pp. 49–51, 2011.

[8] Y. Peng, T. Lu, J. Liu, Y. Gao, X. Guo, and F. Xie, “Cyber-physical system risk assessment,” in Proceedings - 2013 9th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2013, 2013, pp. 442–447.

[9] IEC Smart Grid Standardization Roadmap. [Online]. Disponível em: https://www.iec.ch/smartgrid/downloads/sg3_roadmap.pdf.

[10] E. Freudenthal, T. Pesin, L. Port, E. Keenan, and V. Karamcheti, “dRBAC: Distributed Role-based Access Control for Dynamic Coalition Environments,” Proc. 22nd Int. Conf. Distrib. Comput. Syst., pp. 411–420, 2002.

[11] B. Shafiq, J. B. D. Joshi, E. Bertino, and A. Ghafoor, “Secure interoperation in a multidomain environment employing RBAC policies,” IEEE Trans. Knowl. Data Eng., vol. 17, no. 11, pp. 1557–1577, 2005.

[12] M. Shehab, A. Ghafoor, and E. Bertino, “Secure collaboration in a mediator-free distributed environment,” IEEE Trans. Parallel Distrib. Syst., vol. 19, no. 10, pp. 1338–1351, 2008.

[13] A. M. Gamundani, “An impact review on internet of things attacks,” in Proceedings of 2015 International Conference on Emerging Trends in Networks and Computer Communications, ETNCC 2015, 2015, pp. 114–118.

[14] Y. C. Y. Cao and L. Y. L. Yang, “A survey of Identity Management technology,” 2010 IEEE Int. Conf. Inf. Theory Inf. Secur., pp. 287–293, 2010.

[15] S. Clauβ, D. Kesdogan, and T. Kölsch, “Privacy enhancing identity management,” Proc. 2005 Work. Digit. identity Manag. - DIM ’05, vol. 9, no. 1, pp. 35–44, 2005.

[16] D. Hardt, “[OAuth 2.0] The OAuth 2.0 Authorization Framework [RFC 6749],” RFC 6749, pp. 1–76, 2012.

[17] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore, “Openid connect core 1.0,” OpenID Found., p. S3, 2014.

[18] W. Stallings and M. Bauer, Computer Security - Principles and Practice, vol. 56, no. 2. 2014.

[19] D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli, “Role-Based Access Control,” Components, vol. 2002, no. 10, p. 338, 2003.

[20] R. S. Sandhu and P. Samarati, “Access Control: Principles and Practice,” IEEE Commun. Mag., vol. 32, no. 9, pp. 40–48, 1994.

[21] D. Ferraiolo, J. Barkley, and R. Kuhn, “Role-Based Access Controls,” ACM Trans. Inf. Syst. Secur., vol. 2, no. 1, pp. 34–64, 1992.

[22] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo, “Attribute-based access control,” Computer (Long. Beach. Calif)., vol. 48, no. 2, pp. 85–88, 2015.

[23] B. Parducci and H. Lockhart, “eXtensible Access Control Markup Language (XACML) Version 3.0,” OASIS Stand., no. January, pp. 1–154, 2013.

[24] C. F. Tsai, Y. F. Hsu, C. Y. Lin, and W. Y. Lin, “Intrusion detection by machine learning: A review,” Expert Systems with Applications, vol. 36, no. 10. pp. 11994–12000, 2009.

[25] R. Mitchell and I.-R. Chen, “A Survey of Intrusion Detection Techniques for Cyber-physical Systems,” ACM Comput. Surv., vol. 46, no. 4, p. 55:1-55:29, 2014.

[26] V. C. Gungor, B. Lu, and G. P. Hancke. Opportunities and challenges of wireless sensor networks in smart grid. IEEE Trans. Ind. Electron., 57(10):3557–3564, 2010.

[27] D. J. Gaushell and H. T. Darlington, “SUPERVISORY CONTROL AND DATA ACQUISITION.,” Proc. IEEE, vol. 75, no. 12, pp. 1645–1658, 1987.

[28] S. Karnouskos and A. W. Colombo, “Architecting the next generation of service-based SCADA/DCS system of systems,” in IECON Proceedings (Industrial Electronics Conference), 2011, pp. 359–364.

[29] B. Miller and D. Rowe, “A survey SCADA of and critical infrastructure incidents,” in Proceedings of the 1st Annual conference on Research in information technology - RIIT ’12, 2012, p. 51.

[30] C. Gunter, D. Liebovitz, and B. Malin, “Experience-based access management: A life-cycle framework for identity and access management systems,” IEEE Secur. Priv., vol. 9, no. 5, pp. 48–55, 2011.

[31] S. Osborn, R. Sandhu, and Q. Munawer, “Configuring role-based access control to enforce mandatory and discretionary access control policies,” ACM Trans. Inf. Syst. Secur., vol. 3, no. 2, pp. 85–106, 2000.

[32] P. Duessel, C. Gehl, P. Laskov, J.-U. Busser, C. Stoermann, and J. Kaestner, “Cyber-Critical Infrastructure Protection Using Real-Time Payload-Based Anomaly Detection,” Crit. Inf. Infrastructures Secur., vol. 6027, pp. 85–97, 2010.

[33] D. Hadžiosmanović, L. Simionato, D. Bolzoni, E. Zambon, and S. Etalle, “N-gram against the machine: On the feasibility of the N-gram network analysis for binary protocols,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2012, vol. 7462 LNCS, pp. 354–373.

[34] S. Shin, T. Kwon, G. Y. Jo, Y. Park, and H. Rhy, “An experimental study of hierarchical intrusion detection for wireless industrial sensor networks,” IEEE Trans. Ind. Informatics, vol. 6, no. 4, pp. 744–757, 2010.

[35] S. Cheung and K. Skinner, “Using Model-based Intrusion Detection for SCADA Networks,” Sci. Technol., vol. 329, no. 7461, pp. 1–12, 2006.

[36] Chi-Ho Tsang and Sam Kwong, “Multi-Agent Intrusion Detection System in Industrial Network using Ant Colony Clustering Approach and Unsupervised Feature Extraction,” in 2005 IEEE International Conference on Industrial Technology, 2005, pp. 51–56.

[37] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009, 2009.

[38] NIST, “Framework for Cyber-Physical Systems,” 2017.

[39] T. Baumeister. Literature review on smart grid cyber security, Technical Report, http://csdl.ics.hawaii.edu/techreports/10-11/10-11.pdf. 2010.

[40] P. McDaniel and S. McLaughlin. Security and privacy challenges in the smart grid. IEEE Security & Privacy, 7(3):75–77, 2009.

[41] R. Anderson and S. Fuloria. Who controls the off switch? IEEE SmartGridComm’10, pages 96–101, 2010.

[42] V. Abreu et al., “A Smart Meter and Smart House Integrated to an IdM and Key-based Scheme for Providing Integral Security for a Smart Grid ICT,” Mobile Networks and Applications, pp. 1–15, 2017

[43] R. Berthier, W. H. Sanders, and H. Khurana. Intrusion detection for advanced metering infrastructures: Requirements and architectural directions. IEEE SmartGridComm’10, pages 350–355, 2010.

[44] Z. Lu, X. Lu, W. Wang, and C. Wang. Review and evaluation of security threats on the communication networks in the smart grid. Military Communications Conference'2010, pages 1830–1835, 2010.

[44] S. Babar, P. Mahalle, A. Stango, N. Prasad, and R. Prasad, "Proposed security model and threat taxonomy for the Internet of Things (IoT)," in Communications in Computer and Information Science, 2010, vol. 89 CCIS, pp. 420–429.

[45] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," Ad Hoc Networks, vol. 10, no. 7. pp. 1497–1516, 2012.

[46] ZDnet, "Smart meter hacking tool released," 2012. [Online]. Disponível em: http://www.zdnet.com/article/smart-meter-hacking-tool-released/.

[47] Infoworld, "Millions of embedded devices use the same hard-coded SSH and TLS private keys," 2015. [Online]. Disponível em: http://www.infoworld.com/article/3009667/security/millions-ofembedded-devices-use-the-same-hard-coded-ssh-and-tls-privatekeys.html. [Accessed: 01-Apr-2018].

[48] Hacker News, "Millions of IoT Devices Using Same Hard-Coded CRYPTO Keys," 2015. [Online]. Disponível em: http://thehackernews.com/2015/11/iot-devicecrypto-keys.html. [Accessed: 01-Apr-2018].

[49] J. Liu, Y. Xiao, and C. L. P. Chen, "Authentication and Access Control in the Internet of Things," in 2012 32nd International Conference on Distributed Computing Systems Workshops, 2012, pp. 588–592.

[50] W. L. Chin, Y. H. Lin, and H. H. Chen, "A Framework of Machine-to-Machine Authentication in Smart Grid: A Two-Layer Approach," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 102–107, 2016.

[51] K. Ammayappan, A. Saxena, and A. Negi, "Mutual authentication and key agreement based on elliptic curve cryptography for GSM," in *Proceedings - 2006 14th International Conference on Advanced Computing and Communications, ADCOM 2006*, 2006, pp. 183–186.

[52] A. Witkovski, A. Santin, V. Abreu, and J. Marynowski, "An IdM and key-based authentication method for providing single sign-on in IoT," in 2015 IEEE Global Communications Conference, GLOBECOM 2015, 2015.

[53] D. Jaros and R. Kuchta, "New location-based authentication techniques in the access management," in Proceedings - 6th International Conference on Wireless and Mobile Communications, ICWMC 2010, 2010, pp. 426–430.

[54] F. Zhang, A. Kondoro, and S. Muftic, "Location-Based Authentication and Authorization Using Smart Phones," 2012 IEEE 11th Int. Conf. Trust. Secur. Priv. Comput. Commun., pp. 1285–1292, 2012.

[55] J. Camenisch, D. A. Ortiz-yepes, and F. Preiss, “Strengthening Authentication with Privacy-Preserving Location Verification of Mobile Phones,” Wpes, pp. 37–48, 2015.

[56] M. Choi, J. Lee, S. Kim, Y.-S. Jeong, and J.-H. Park, “Location based authentication scheme using BLE for high performance digital content management system,” Neurocomputing, vol. 209, pp. 25–38, 2016.

[57] C. Anthony, LaMarca; Yatin, Chawathe; Sunny, “Place Lab: Device Positioning Using Radio Beacons in the Wild,” in Pervasive Computing, 2005, pp. 116–133.

[58] H. K. Lee and H. Luedemann, “Lightweight Decentralized Authorization Model for Inter-Domain Collaborations,” in Proceedings of the 2007 ACM workshop on Secure web services, 2007, pp. 83–89.

[59] Q. Li, X. Zhang, S. Qing, and M. Xu, “Supporting ad-hoc collaboration with group-based RBAC model,” in 2006 International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom, 2006.

[60] S. C. Mouliswaran, C. A. Kumar, and C. Chandrasekar, “Representation of multiple domain role based access control using FCA,” in Proceedings of 2015 IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT 2015, 2015.

[61] W. She, W. Zhu, I. L. Yen, F. Bastani, and B. Thuraisingham, “Role-Based Integrated Access Control and Data Provenance for SOA Based Net-Centric Systems,” *IEEE Trans. Serv. Comput.*, vol. 9, no. 6, pp. 940–953, 2016.

[62] K. Avita et al., “Authentication and Authorization Domain Specific Role Based Access Control Using Ontology,” Intell. Syst. Control, pp. 439–444, 2012.

[63] V. Abreu, A. O. Santin, E. K. Viegas, and M. Stihler, “A multi-domain role activation model,” in IEEE International Conference on Communications, 2017.

[64] Eddystone, https://developers.google.com/beacons/eddystone.

[65] D. M'Raihi, S. Machani, M. Pei and J. Rydell, "TOTP: Time-Based One-Time Password Algorithm", RFC 6238, DOI 10.17487/RFC6238, May 2011, <https://www.rfc-editor.org/info/rfc6238>.

[66] ANSI, “X9 Encryption Collection,” 2009. [Online]. Disponível: https://webstore.ansi.org/packages/x9.aspx.

[67] Cesar, R. “Modelo De Autenticação Multicanal Baseado Em Proximidade”, 2018.

[68] W. Stallings, Cryptography and Network Security: Principles and Practices, 2008.

[69] Z. Shelby, K. Hartke, and C. Bormann, “Constrained Application Protocol (CoAP),” CoRE Work. Gr., pp. 1–118, 2013.

[70] E. Rescorla and N. Modadugu, “RFC 6347 - Datagram Transport Layer Security Version 1.2,” IETF RFC, pp. 1–32, 2012.

[71] O. Garcia-Morchon, S. L. Keoh, S. Kumar, P. Moreno-Sanchez, F. Vidal-Meca, and J. H. Ziegeldorf, “Securing the IP-based internet of things with HIP and DTLS,” in Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks - WiSec ’13, 2013, p. 119.

[72] OWASP Project Top 10 – 2017. [Online]. Disponível em: https://www.owasp.org/index.php/Top_10-2017_Top_10

[73] Network Penetration Testing. [Online]. Disponível em: https://www.redteamsecure.com/network-penetration-testing/

[74] VIEGAS, E. K.; SANTIN, A. O.; BESSANI, Alysson Neves; NEVES, N. F. BigFlow: Real-time and Reliable Anomaly-based Intrusion Detection for High-Speed Networks. Future Generation Computer Systems-The International Journal of eScience, Accepted for publication. 2018.

[75] Breiman, L. Machine Learning (2001) 45: 5. Disponível em: https://doi.org/10.1023/A:1010933404324

[76] Wood, Mark, and Michael Erlinger. Intrusion detection message exchange requirements. No. RFC 4766. 2007.

[77] Newman, Sam. Building microservices: designing fine-grained systems. " O'Reilly Media, Inc.", 2015.

[78] A. Anderson, “XACML Profile for Role Based Access Control (RBAC), Version 2.0,” 2004.

[79] Newman, Sam. Building microservices: designing fine-grained systems. " O'Reilly Media, Inc.", 2015.

[80] Wang, Shuzhen, Zonghua Zhang, and Youki Kadobayashi. "Exploring attack graph for cost-benefit security hardening: A probabilistic approach." Computers & security 32 (2013): 158-169.