UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Wesley dos Reis Bezerra

# Um mecanismo de autenticação com reputação e multi-fator para dispositivos restritos em IoT e LPWAN

Florianópolis
2024

Wesley dos Reis Bezerra

# Um mecanismo de autenticação com reputação e multi-fator para dispositivos restritos em IoT e LPWAN

Tese submetida ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do título de Doutor em Ciência da Computação.
Orientador: Prof. Carlos Becker Westphall, Dr.

Florianópolis
2024

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Bezerra, Wesley dos Reis
Um mecanismo de autenticação com reputação e multi-fator para dispositivos restritos em IoT e LPWAN / Wesley dos Reis Bezerra ; orientador, Carlos Becker Westphall, 2024.
111 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Ciência da Computação, Florianópolis, 2024.

Inclui referências.

1. Ciência da Computação. 2. IoT. 3. Autenticação. 4. Segurança. 5. LPWAN. I. Westphall, Carlos Becker. II. Universidade Federal de Santa Catarina. Programa de Pós Graduação em Ciência da Computação. III. Título.

Wesley dos Reis Bezerra

**Um mecanismo de autenticação com reputação e multi-fator para dispositivos restritos em IoT e LPWAN**

O presente trabalho em nível de doutorado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Profª. Janine Kniess, Drª.
UDESC

Prof. Jean Everson Martina, Dr.
INE/CTC/UFSC

Prof. Mario de Noronha Neto, Dr.
Instituto Federal de Santa Catarina

Prof. Rafael de Santiago, Dr.
INE/CTC/UFSC

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Doutor em Ciência da Computação.


Documento assinado digitalmente
**Cristina Meinhardt**
Data: 23/01/2024 08:02:13-0300
CPF: ***.850.380-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Márcio Bastos Castro, Dr
Coordenador do Programa


Documento assinado digitalmente
**Carlos Becker Westphall**
Data: 22/01/2024 20:04:30-0300
CPF: ***.747.990-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Carlos Becker Westphall, Dr.
Orientador

Florianópolis, 2024.

Dedico este trabalho a minha família e meus ancestrais, todos que trilharam caminhos árduos para pavimentar a estrada para as próximas gerações. Em especial aos meus filhos: "ela"(*in memorian*), Felipe (*in memorian*), Pedro (*in memorian*) e Lavínia.

## AGRADECIMENTOS

Primeiramente, ao meu Orientador, Prof. Dr. Carlos B. Westphall, por sua dedicação e guiança durante todo o processo; à Prof[a]. Dr[a]. Carla M. Westphall, pelos conselhos e orientações durante as publicações; ao Dr. Cristiano A. Souza, pela parceria em diversas publicações; e também aos demais membros do LRG, pelo companheirismo durante esse período tão importante. Em especial, agradeço a minha esposa, Prof[a]. Dr[a]. Laís Machado Bezerra, pela ajuda em diversos momentos da pesquisa, pela cumplicidade e pelo suporte em diversos momentos difíceis - sem você eu não teria conseguido.

## RESUMO

A autenticação em dispositivos restritos IoT tem se apresentado como um dos desafios de segurança em sistemas com restrições energéticas e de transmissão de dados. Este trabalho apresenta um mecanismo inédito de autenticação multi-fator para dispositivos de internet das coisas e redes de baixa potência e longa distância (LPWAN) que tem atrelado a si um sistema de reputação. Para isso, traz-se algumas contribuições que podem ser organizadas em dois momentos: uma preparação para o desenvolvimento e o seu desenvolvimento. Na sua fase preparatória, foi trazida uma modelagem prévia do problema e seus desafios; uma revisão sistemática da literatura; e uma visão geral de tecnologias por meio de uma pequena experimentação. Em um segundo momento, foram realizados detalhamentos e validações quanto ao modelo proposto; e algumas revisões de literatura foram realizadas para áreas específicas, assim como experimentações para a validação do modelo. Alguns dos resultados podem ser observados na seção de apêndices, onde estão os artigos a eles relacionados, tendo sido possível confirmar a relevância do objeto de estudo para a área da computação e também verificar a eficácia no mecanismo proposto diante do cenário delimitado.



**Palavras-chave:** mecanismo de autenticação. fog computing. lpwan. iot. constrained devices.

## ABSTRACT

Authentication in restricted IoT devices has been one of the security challenges in systems with energy and data transmission restrictions. This work presents an unprecedented multi-factor authentication mechanism for Internet of Things devices and low-power, long-distance networks (LPWAN) that is linked to a reputation system. To this end, we bring some contributions that can be organized into two moments: preparation for development and its development. In its preparatory phase, a preliminary modeling of the problem and its challenges, a systematic review of the literature and an overview of technologies were carried out through a small experimentation. In a second step, details and validations were carried out regarding the proposed model, some literature reviews were carried out for specific areas, as well as experiments to validate the model. Some of the results can be seen in the appendices section where the articles related to them are located. In this way, it could confirm the relevance of the object of study for the area of computing and also verify the effectiveness of the proposed mechanism in the delimited scenario.



**Keywords:** authentication mechanism. fog computing. lpwan. iot. constrained devices.

## LISTA DE ILUSTRAÇÕES

**SUMÁRIO**

# 1 INTRODUÇÃO

Ultimamente, vários desafios de segurança atrelados a dispositivos restritos IoT (*Internet of Things –Internet* das Coisas) têm surgido, e tais desafios estão ligados ao baixo consumo de energia e a limitações na transmissão de dados, levando à necessidade de tecnologias de segurança mais adequadas para tratar dessas limitações. Esses dispositivos são alimentados por baterias, placas solares ou fazem "colheita"de energia; têm *hardwares* limitados; pouca capacidade de armazenamento; características de baixa capacidade energética; e limitações quanto à capacidade de processamento, velocidade e capacidade de transmissão de dados. Diante desse cenário, um tipo de protocolo com menor capacidade de transmissão, projetado para dispositivos restritos, tem sido mais utilizado para atender a soluções com essas características, o LPWAN - que tem modificado o cenário IoT (OLSSON, 2014).

O LPWAN é um conceito que agrega protocolos de comunicação para dispositivos com pouco potencial energético, que devem ter um baixo consumo de energia, transmitir poucos dados e atuarem em longas distâncias. Tal conceito está alinhado à eficiência energética e a dispositivos amigáveis ao meio ambiente, ou seja, à aplicação de tecnologias verdes e sustentáveis (ROUTRAY; HUSSEIN, 2019). Algumas das características de tais protocolos são o baixo número de mensagens trocadas e a baixa frequência com que ocorrem essas trocas. No entanto, esse tipo de protocolo não é aplicável a todas as soluções de IoT, tendo seu nicho específico de atuação nos casos em que a frequência e a velocidade da transmissão são menos importantes. Esse é o caso do nosso cenário: o monitoramento do meio ambiente por meio de sensores, além de outras aplicações de *smart metering*, como a agricultura de precisão.

Este trabalho contribui para a área de segurança da informação em três pontos: (i) com um mecanismo de autenticação multifator adequado à LPWAN e dispositivos restritos; (ii) um sistema de reputação para sensores e seus dados; e (iii) um fator de autenticação baseado no comportamento do sensor enquanto parte de um arranjo. A principal e maior contribuição é o mecanismo de autenticação, o qual engloba as demais contribuições. Entretanto, notou-se a oportunidade de estender o conceito de reputação aplicando-o em situações como *public* ou *open sensors networks* (SCARCELLI et al., 2020), utilizadas para o bem comum (ROBINSON; FRANKLIN, 2020; OKIGBO et al., 2020). Não obstante, diferentes fatores de autenticação têm sido pesquisados nos últimos anos e nossa contribuição soma-se a esse conjunto de inovações.

## 1.1 MOTIVAÇÃO

O tipo de dispositivo restrito proposto apresenta desafios para a segurança, pois é um mecanismo que necessita de uma complexa criptografia ou de algoritmos complexos que podem não ser suportados por dispositivos restritos (ZHANG et al., 2019). A baixa frequência na troca de mensagens e o canal com pouca capacidade de transmissão de dados são fatores limitantes para a utilização de protocolos padrão de gestão de identidade amplamente conhecidos, como

OIDC[1], XACML[2] e OAuth2. Ressalta-se que o grande número de trocas de mensagens entre as partes envolvidas no OAuth2 e no OpenID pode fazer com que o dispositivo exceda a quantidade de mensagens determinada em seu pacote de dados, além de que a sobrecarga trazida por uma política descrita em XML[3] também pode vir a extrapolar a quantidade de *bytes* contratados.

## 1.2 JUSTIFICATIVA

A segurança é uma ferramenta necessária para manter e habilitar a privacidade (OMETOV et al., 2018) em qualquer área da IoT. Dessa forma, faz-se necessário um mecanismo de autenticação que seja apropriado aos protocolos de LPWAN (YI et al., 2015; ROMAN; LOPEZ; MAMBO, 2018; CAO et al., 2019) e que garanta uma segurança em nível adequado ao dispositivo, ao sistema e ao seu possuidor. Um mecanismo que promova a compatibilidade com dispositivos legados e assegure a identificação dos sensores envolvidos no sistema IoT prevenirá alguns dos mais notáveis ataques de segurança nesse âmbito, tais como: *man-in-the-middle* (MITM), DDoS, replay e acesso não identificado.

## 1.3 OBJETIVOS

Este estudo apresenta um objetivo geral e alguns específicos, conforme explicitado nas subseções a seguir.

### 1.3.1 Objetivo geral

O objetivo geral deste trabalho é propor um mecanismo, original e inédito, de autenticação com reputação e multifator para dispositivos restritos em IoT e LPWAN. Tal mecanismo deve oferecer flexibilidade na autenticação, atrelando-a a um sistema de reputação dos sensores participantes do sistema, sendo capaz de autenticar os dispositivos legados ou que implementem seu mecanismo de autenticação de maneira transparente. O objetivo geral pode ser entendido a partir da seguinte pergunta de pesquisa:

*É possível a implementação de uma autenticação leve que tome como princípios as limitações, acima citadas, dos dispositivos restritos e das redes de baixa potência e longas distâncias com um desempenho e nível de segurança adequados para os sistemas IoT?*

### 1.3.2 Objetivos específicos

Para este trabalho, temos os seguintes objetivos específicos:

[1] OpenID Connect
[2] eXtensible Access Control Markup Language
[3] eXtensible Markup Language

- Criar um mecanismo de autenticação multifator que se distancie dos problemas em aberto (anteriormente citados);
- Adotar uma abordagem multifator (MFA) em conjunto com um sistema de reputação para os sensores;
- Desenvolver um experimento para validação do mecanismo proposto;
- Levantar o estado da arte de IoT, LPWAN e autenticação multifator;
- Fazer uma revisão sistemática de literatura sobre o tema;
- Analisar os desafios encontrados;
- Apresentar uma proposta preliminar de mecanismo de autenticação;
- Fazer a modelagem da proposta de um mecanismo de autenticação;
- Desenvolver o experimento em memória, utilizando um protocolo de mensagens.

## 1.4 CONTRIBUIÇÕES E RESULTADOS OBTIDOS

Ao longo do trabalho são detalhadas as fases do desenvolvimento do mecanismo de autenticação, contudo, alguns resultados foram listados nos apêndices e fazem parte de artigos já publicados. Dessa forma, a seguir listamos um resumo de algumas contribuições trazidas como resultado do processo de construção desse mecanismo. São elas:

- Criação de um ambiente para avaliação dos experimentos com protocolos e suas configurações de segurança (Apêndice A.1);
- Avaliação dos protocolos de mensagem a serem utilizados (Apêndice A.2);
- Revisão de literatura sobre paradigmas de computação sob demanda (Apêndice A.3);
- Revisão de literatura sobre tendências, oportunidades e desafios para autenticação de dispositivos restritos na *fog computing* (Apêndice A.4);
- Revisão de literatura sobre características e principais ameaças para autenticação multifator (Apêndice A.5);
- Experimentação utilizando o mecanismo proposto e avaliando o seu desempenho em um cenário simulado (Apêndice A.6);
- Verificação formal do mecanismo proposto utilizando autômatos temporizados e lógica temporal (*safety*) (Apêndice A.7).

## 1.5 ORGANIZAÇÃO DO TRABALHO

Este trabalho, além desta breve introdução, está estruturado da seguinte maneira: no segundo capítulo é apresentado o referencial teórico que norteou o estudo, seguido pela revisão

sistemática da literatura, no terceiro capítulo, em que foi aplicada a metodologia ProKnow-C de maneira customizada para atender aos objetivos propostos. No quarto capítulo são listados os desafios de segurança encontrados no portfólio bibliográfico elegido na revisão. No capítulo cinco é apresentada a proposta de solução por meio de algoritmos que descrevem seus processos. Já no quinto capítulo é apresentada a modelagem da solução por meio da UML, o que permite uma representação mais próxima da implementação. No sétimo capítulo é apresentada uma conclusão do estudo e as sugestões para trabalhos futuros. Em adição, tem-se os apêndices, que trazem uma breve visão dos artigos que apresentam os resultados obtidos nas experimentações, revisões e validações do mecanismo.

# 2 CONCEITOS BÁSICOS

Neste capítulo serão apresentados conceitos importantes para o entendimento do modelo. Alguns autores selecionados como base, assim como a visão que eles delimitam, podem sem encontrados ao longo deste capítulo. Uma vez que sabemos que nem todos os conceitos apresentam uma visão única dentro da computação, e que alguns dos mais atuais estão ainda se firmando, é importante o mapeamento que realizamos para um melhor entendimento das decisões tomadas durante o processo de construção do mecanismo proposto.

## 2.1 IOT

*Internet* das coisas é uma tecnologia emergente, sendo pesquisada de maneira notável há mais de uma década (DEEP; ZHENG; HAMEY, 2019a). Tal tecnologia integra diferentes tipos de equipamentos com conectividade à *Internet* e oferece, de maneira geral, uma plataforma para o cruzamento de dados, criação de perfis de uso e extração de inteligência dos dados obtidos. Sua adoção e seu crescimento têm sido expressivos, tendo sido citada como a *Internet* do futuro, de acordo com autores como Yi *et al.* (YI et al., 2015), entretanto, ela deve ser encarada tanto como uma oportunidade quanto uma ameaça.

A oportunidade provém do fato de novos serviços, equipamentos e comodidades estarem disponíveis aos usuários a um preço mais acessível e com uma forma de utilização mais facilitada para o público geral. Já a ameaça tem relação com a quantidade de equipamentos, muitos deles mal configurados e com o poder computacional para usuários maliciosos (i.e., com intenções de degradar o desempenho de sistemas), o que tem aumentado significativamente. Ataques massivos de negação de serviço, serviços de rede inseguros, *interfaces* de *cloud*/*mobile*/*web* inseguras, entre outros, são alguns dos exemplos (MIESSLER, 2015) de problemas decorrentes de equipamentos que implementam pouca segurança ou estão mal configurados.

A IoT está presente em diversas áreas, tais como: indústria, automação residencial, automação predial, segurança e vigilância residencial, cidades inteligentes e agricultura inteligente. Tal tecnologia tem se mostrado um fator-chave para o melhor desempenho na produção de bens e alimentos, e na melhoria da qualidade de vida das pessoas em todo o mundo. Como promotora de um melhor desempenho na agricultura, a IoT tem sido utilizada em diversas áreas da agricultura, tendo trazido benefícios financeiros e no atendimento à legislação.

Algumas das áreas da agricultura que têm colhido frutos com a utilização da IoT são: agropecuária, maricultura, carcinicultura, piscicultura, agricultura de precisão, entre outras. Ressalta-se que soluções para uma melhor utilização de insumos agrícolas têm melhorado os custos, economizado água e diminuído a utilização de agrotóxicos. Por exemplo, com mais informações sobre o solo, é possível fazer correções e diminuir os efeitos de pragas, o desgaste do solo e a contaminação das águas. Outra importante aplicação é no monitoramento e utilização de recursos hídricos. Dessa forma, é possível preservá-lo, diminuindo custos de bombeamento

e evitando a lixiviação do solo.

Devido a sua grande entrada em diversos setores, a IoT é tanto uma realidade promissora como um desafio em relação às questões de segurança, o que está ligado à emergência dessa tecnologia e ao fato de ainda não existirem mecanismos de segurança bem consolidados para tal aplicação, conforme Tiburski *et al.* (TIBURSKI et al., 2019). A IoT tem sido aplicada em diversos setores, que variam desde áreas que não têm um passado próximo com a tecnologia, até áreas que utilizam tecnologias de ponta em seu dia a dia. Essas últimas, muitas vezes, têm a elas associados profissionais mais capacitados à implementação de tais mecanismos, o que não acontece comumente em uma situação de agricultura familiar.

Cada solução de sistema de IoT envolve diversos parâmetros, que poderão influenciar nos mecanismos de segurança a serem utilizados. São parâmetros como o *link* de dados utilizado, a confiabilidade necessária na transmissão de dados, as restrições temporais e os atrasos. Outro fator importante é a capacidade de fornecimento de energia aos dispositivos envolvidos, pois alguns dispositivos estarão ligados diretamente à rede de fornecimento de energia, outros utilizarão baterias e ainda existirão alguns que farão a colheita energética (*i.e.* solar); para utilização no processamento e transmissão de dados.

Há ainda o foco em dispositivos restritos (*constrained devices*), que funcionarão com baterias, além de terem um *link* de rádio (sem fio) com pouca capacidade de transmissão de dados a longas distâncias (LPWAN). São dispositivos que estarão esparsamente distribuídos e potencialmente isolados geograficamente.

## 2.2 COMPUTAÇÃO EM NUVEM E NEVOEIRO

Neste estudo, adotou-se o conceito do NIST (*National Institute of Standards and Technologies*), que apresenta uma visão componentizada da computação em nuvem. Tal conceito (MELL; GRANCE et al., 2011) determina que a computação em nuvem é “um modelo para habilitar um conjunto de recursos computacionais compartilhados de maneira onipresente, conveniente e com acesso sob demanda, através da rede”. Para Bonomi *et al.* (BONOMI et al., 2014), a computação em nuvem liberta a empresa e o usuário final de uma especificação muito detalhada para funcionamento de um serviço. Também há uma visão herdada da computação distribuída, que apresenta a *cloud* como uma extensão da computação distribuída, da computação paralela e da computação em grade (AAZAM; HUH, 2014).

Como se pôde notar, alguns termos do conceito estão associados à ideia inicial de capacidade virtualmente infinita da nuvem de prover serviços quando demandada, entretanto, a capacidade de prover comunicação não foi amplamente atendida. Dessa forma, criou-se um gargalo de comunicação, em consequência do surgimento em grande quantidade dos dispositivos inteligentes conectados à *Internet*.

A computação em nuvem (*Cloud Computing* - CC) está encontrando crescentes desafios para atender requisitos novos e complexos, advindos das tecnologias emergentes de IoT (CHIANG; ZHANG, 2016). Um desses desafios é em relação à quantidade de dispositivos

conectados, o que tem aumentado consideravelmente, e muitos deles fornecendo pouca ou nenhuma configuração de segurança para os usuários finais, tornando-se alvos fáceis de ataques de *hackers* mal-intencionados (OWASP. . . , 2019).

Outro grande desafio é o gargalo de rede gerado pela posição centralizada na arquitetura provida pelos serviços em CC. Os dispositivos que desejam fazer computação, armazenamento e comunicação, entre outros, devem sempre se comunicar com o nó central na nuvem, e isso faz com que haja tráfego intenso de dados de monitoramento e controle entre a nuvem e os dispositivos em questão. Para resolução dos problemas que a centralização no provimento de serviços trouxe, foi proposto o modelo de computação em nevoeiro (*Fog Computing* - FC) (STOJMENOVIC, 2014).

Alguns autores propõem a FC como um complemento da computação em nuvem (CHIANG; ZHANG, 2016), entretanto, seu propósito vai muito além, pois ela permite ao sistema executar ações de maneira descentralizada, através de seus *fog nodes* (FN). Um FN pode trocar informação com outros FNs, de modo que informações sobre controle de acesso, autenticação e outras endereçadas à gestão de identidades possam trafegar entre os nodos. Cabe ressaltar também que os FNs podem estar federados (IORGA et al., 2018).

Outra importante característica trazida pela FC é que, diferentemente da computação em nuvem, o modelo de *cloud* se propõem ser um modelo em camadas, promovendo abstração e desacoplamento para as partes do sistema. Os sensores não precisam guardar informações e conhecer a *cloud*, tampouco precisam se comunicar diretamente com ela, o que possibilita um isolamento maior da camada de sensores, permitindo implementar melhores políticas de segurança.

Aliado a isso, há o fato de o processamento acontecer próximo aos sensores e atuadores, fazendo com que a tomada de decisão possa acontecer mais rapidamente. Em algumas situações, é importante que o sensor decida em um tempo determinado e, caso esse tempo seja excedido, o processo falhará. Tais casos sãos mais comuns em ambientes industriais (Indústria 4.0), ou mesmo em cidades inteligentes (*smart cities*), como em sinais de trânsito e no processamento de placas de automóveis, entre outras situações.

## 2.3 PROTOCOLOS DE MENSAGENS EM IOT

Um grande influenciador no possível gargalo obtido pela IoT são os protocolos que trafegam dados entre as partes envolvidas. Em um sistema que utiliza CC, o sensor pode comunicar-se diretamente com a *Cloud* ou com um intermediário, o *Gateway* (GW), cuja responsabilidade é centralizar dados e fornecer uma capacidade de conectividade, o que para alguns dispositivos é inalcançável, devido a restrições orçamentárias do projetos ou outro tipo de restrições que limitam a conectividade.

Tal cenário, conforme apresentado na Figura 1, foi elegido devido a características comuns, tanto para CC quanto FC, sendo esta última o foco deste estudo. No caso da *fog*, o GW deverá comunicar-se com o FN, e não diretamente com a *cloud*, sendo esta a principal

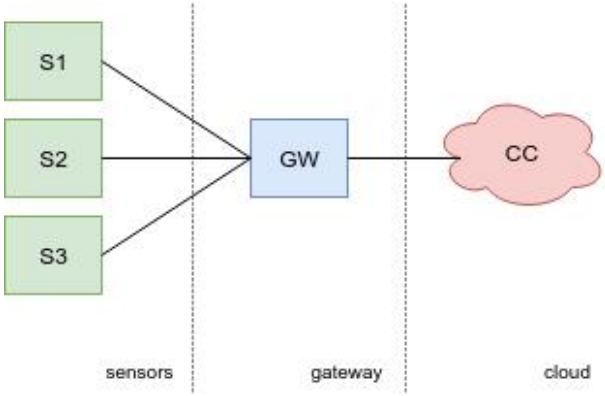



Figura 1 – Comunicação de sensores em nuvem

diferença. Nesse caso, teremos dois segmentos da comunicação: sensor para *gateway* (n1) e *gateway* para *fog* (n2). No primeiro segmento (n1), o sensor terá um *link* sem fio com pouca capacidade de transmissão, sendo tais características importantes para a avaliação e para os resultados obtidos nos experimentos, desse modo, o *link* será LPWAN. Já no segundo segmento (n2), o *link* terá um maior potencial de transmissão, podendo ser uma rede cabeada, *Wi-Fi* ou qualquer outro tipo de protocolo que não sofra restrições de dispositivos mais modestos.

Serão enumerados protocolos com o foco no segmento sensor-*Gateway*, mas considerações sobre segmentos com maior processamento também serão realizadas, quando necessárias. Os protocolos descritos podem ser classificados de diferentes maneiras: quanto ao uso de URI[1] ou tópicos; quanto à QoS em cada protocolo; quanto à segurança, entre outros aspectos. Alguns protocolos importantes são: MQTT, CoAP, DDS, AMQP, XMPP, STOMP e HTTP. Tais protocolos foram selecionados devido a sua notoriedade na utilização em sistemas IoT.

O protocolo MQTT (*Message Queueing Telemetry Transport*) é um protocolo leve para tráfego de dados de sensoriamento (telemetria), utilizado com frequência em sistemas SCADA (*Supervisory Control and Data Acquisition*), tendo sido criado em 1999 (SONI; MAKWANA, 2017). Por ser bem conhecido e validado, tanto na academia quanto na indústria, ele é o protocolo sobre o qual há mais estudos e artigos publicados. Tal protocolo apresenta três níveis de QoS; pode ter segurança no canal de comunicação SSL/TLS; em alguns servidores é implementada autenticação baseada em *login*/senha (STANDARD, 2014); utiliza a arquitetura *publish/subscriber*; os dados são publicados em tópicos; é um protocolo binário; utiliza um *broker* centralizado; e seu transporte acontece utilizando TCP.

O CoAP é um protocolo que tem ganhado notoriedade devido a ser ainda mais leve do que o protocolo MQTT. O *Constrained Application Protocol* é leve para comunicação máquina-máquina (M2M) do IETF CoRE Working Group (NAIK, 2017). Como não utiliza TCP para transporte, mas UDP, seu desempenho em redes de baixa qualidade (onde há muito reenvio de pacotes) é superior ao MQTT. Apesar de utilizar um protocolo como o UDP, que não envia confirmação de pacotes, é possível ter-se um protocolo confiável por meio da utilização das opções de QoS fornecidas pelo CoAP. Esse protocolo apresenta duas opções de QoS; implementa

[1] *Universal Resource Identifier* - identificador universal de recurso (tradução livre.)

segurança de canal utilizando DTLS; utiliza a arquitetura cliente-servidor, mas com a opção de *observer*; utiliza URI para publicação de recursos e seu acesso; é um protocolo binário; utiliza um servidor centralizado; e seu transporte acontece utilizando UDP.

DDS (*Data Distribution System*) é um padrão de interoperabilidade em tempo real (PARDO-CASTELLOTE, 2003), desenvolvido em 2004 pela Object Management Group (OMG). Diferentemente de outros protocolos, ele não apresenta uma figura central, como um servidor ou um *broker*, sendo um protocolo descentralizado. O DDS é dito um protocolo centrado em dados (PARDO-CASTELLOTE, 2003) e de fato é um protocolo de MOM (*Message Oriented Middleware*) (ALKHAWAJA; FERREIRA; ALBANO, 2012). Sua comparação com o CORBA é inevitável, pois ambos foram desenvolvidos pela OMG e utilizam IDL (CORSARO, 2014). Ele oferece mais de 20 opções de QoS, tais como: configurações padrão; segurança de canal com DTLS quanto à SSL/TLS; permite autenticação; utiliza a arquitetura descentralizada *publish/subscriber*; é um protocolo binário; não tem componente central, pois é baseado em *middleware*; e seu transporte pode acontecer tanto por UDP quanto por TCP.

AMQP (*Advanced Message Queueing Protocol*) é um protocolo para troca de mensagens originário do mercado financeiro e criado pela JPMorgan em 2003 (NAIK, 2017), por John O'Hara. Sua arquitetura é de *publish/subscriber* ou cliente/servidor, com a utilização de *exchanges* e *queues* para a distribuição de mensagens entre os assinantes, possuindo diferentes tipos de *exchange* de mensagens, como *fan-out*, tópicos e cabeçalhos. O protocolo oferece dois tipos de QoS (*unsettle format* e *settle format*); permite a utilização de TLS/SSL; sua autenticação é utilizando SASL; tem a figura central do *broker*/servidor; seu protocolo é binário; e para transporte utiliza o TCP (STANDARD, 2012).

XMPP (*eXtensible Message and Presence Protocol*) é um protocolo que tem ganhado espaço em IoT, principalmente devido as suas características de segurança (DIZDAREVIĆ et al., 2019). Para utilização em IoT, é necessário que seja utilizada a extensão XEP060 (pubsub), que permite a utilização do protocolo com a arquitetura *publish/subscriber* (SAINT-ANDRE, 2011). Um importante ponto negativo a se considerar é que o protocolo não implementa QoS, o que o torna menos atrativo em relação a outras opções. O protocolo não apresenta QoS; a segurança de canal é utilizando SSL/TLS; a autenticação é feita utilizando SASL; a arquitetura é *publish/subscriber*; tem a figura central do servidor; seu protocolo é textual e baseado em marcações; e para o transporte utiliza TCP.

STOMP (*Simple/Streaming Text Oriented Message Protocol*) é um protocolo textual baseado em *frames* modelados sobre HTTP (SZYDŁO; SUDER; BIBRO, 2013). Propõe-se a ser um protocolo de fácil utilização, tanto para implementação do lado servidor quanto do lado cliente. Ele não implementa opções de QoS; tem segurança de canal TLS/SSL e autenticação com usuário/senha; sua arquitetura é *publish/subscriber*; tem a figura central do *broker*; seu protocolo é textual; e o transporte é feito utilizando TCP.

HTTP (*HyperText Transfer Protocol*) é um protocolo para *web*, proposto por Tim Bernes-Lee e padronizado em 1997 (NAIK, 2017). Seu ponto positivo é que por ser um protocolo amplamente utilizado para sistemas *web* e construção de *sites*, é bem conhecido pelos

desenvolvedores (DIZDAREVIĆ et al., 2019). Entretanto, por ser um protocolo baseado em texto e utilizar o transporte TCP, ele não tem um bom desempenho em redes e dispositivos com restrições energéticas e de transmissão. O protocolo não implementa QoS; tem segurança de canal com SSL/TLS e diferentes mecanismos de autenticação; sua arquitetura é cliente/servidor; tem a figura central do servidor; seu protocolo é textual; e utiliza para transporte o TCP.

Existe uma grande complexidade na decisão sobre quais tecnologias adotar para os projetos IoT. Com diferentes opções de utilização de protocolos de mensagem, os sistemas IoT trazem consigo características variáveis de segurança, robustez e capacidade de entrega de dados. A escolha do protocolo mais adequado deve levar em conta características da solução a ser implementada.

## 2.4 GESTÃO DE IDENTIDADES

A gestão de identidades digitais (*Identity Management* - IdM) é um fator fundamental para sistemas computacionais (BERTINO; TAKAHASHI, 2010). Tal gestão permite fazer customizações de acordo com as preferências e informações do usuário associadas à identidade, ou ainda propor melhoramentos na experiência e na proteção de privacidade, entre outros aspectos. Este estudo tem foco nos aspectos de segurança relacionados à gestão de identidade, como controle de acesso, identificação e questões associadas à privacidade.

Violações de identidade podem ocorrer devido a diversos fatores (BERTINO; TAKAHASHI, 2010), como clonagem de identidade com intuito criminoso em relação a finanças. Em razão disso, ter uma boa implementação de gestão de identidades é também um fator fundamental para a segurança de sistemas. No tocante aos sistemas que devem ser assegurados, não se excluem aqueles que utilizam comunicação máquina-máquina, tendo como exemplo os que utilizam *Internet* das Coisas.

Atrelada ao escopo de IoT, a gestão de identidade pode acontecer tanto na identificação do usuário que está acessando os dados coletados ou informações processadas a partir de tais dados, quanto na permissão de utilização de uma "coisa" que é vista como um recurso, ou ainda em relação à própria atuação da "coisa" como sujeito produtor/consumidor de dados e atuador no mundo físico, mediante alguma configuração específica de dados.

No caso específico dos sensores, a perda de produtividade, os danos ao patrimônio ou as violações de acesso são alguns exemplos decorrentes da incorreta utilização da gestão de identidade. Por exemplo, no caso de um equipamento que envie informações sobre a umidade de uma estufa de morangos de maneira não segura, ele pode sofrer uma violação, como no caso de um atacante interceptar e alterar o valor de medição da umidade, fazendo com que não haja uma ativação do sistema de irrigação e os morangos não sejam produzidos de maneira adequada. Outro exemplo é o do *worm* Stuxnet (LANGNER, 2011; FALLIERE; MURCHU; CHIEN, 2011), que alterou a referência para a rotação das centrífugas de urânio no Irã, fazendo com que elas fossem danificadas. Sendo assim, a não implementação de gestão adequada de identidade em IoT é um grande fator de risco.

## 2.5 AUTENTICAÇÃO MULTIFATOR

Autenticação multifator é, segundo Johansson (JOHANSSON et al., 2018), uma abordagem que provê um grau maior de segurança na autenticação em relação às abordagens baseadas em somente um passo, as quais são mais comumente utilizadas. Existem diversas formas de se realizar autenticação multifator, mas podemos reduzi-las a três categorias: (i) algo que se sabe; (ii) algo que se tem; ou (iii) algo que se é.

Em relação a algo que se sabe (i), pode-se entender isso como um código ou segredo compartilhado entre as partes. Uma senha, uma frase ou qualquer segredo equivalente pode ser interpretado como algo que se sabe. Esse é o fator usado mais comumente em abordagens de autenticação clássicas. Uma abordagem clássica baseada em somente segredo não é suficientemente segura.

Um código secreto pode ser obtido por meio de engenharia social, ataques de dicionário ou mesmo ser adivinhado (quando fraco). Basear a segurança de um sistema em somente tal fator torna-o mais fácil de implementar, mais barato e mais rápido para o usuário se autenticar, entretanto, não é segura suficientemente em situações em que se deve garantir o mais alto grau de certeza na identificação do usuário.

Outra situação que ocorre quanto à autenticação atrelada a equipamentos de IoT, é quando usuários menos versados em informática tendem a não alterar a configuração padrão de autenticação dos equipamentos que são instalados, além de utilizarem senhas fracas em serviços por meio dos quais tais equipamentos se comunicam.

Já em relação a algo que se tem (ii), a referência é à posse de algo, como um *token* ou outro dispositivo físico, que é um pré-requisito. Esse dispositivo pode conter/gerar uma chave que deverá ser validada por uma entidade reconhecidamente confiável, como, por exemplo, uma Autoridade Certificadora. Dessa forma, a emissão do *token* está condicionada a uma terceira parte, ou seja, uma parte além do sujeito e do provedor de serviço onde ele quer se autenticar. Violações de segurança nesse caso são menos prováveis, mas ainda assim podem acontecer (ESINER; DATTA, 2019).

No caso de violações desse segundo tipo, é necessário que o invasor tenha acesso físico ao dispositivo que é utilizado. Por exemplo, ao receber um código de acesso em seu celular como segundo fator, o usuário garante uma maior segurança no momento do acesso. Entretanto, caso o usuário tenha seu celular roubado, ficando esse em posse do invasor, ele pode terá acesso a esse segundo fator. Dessa forma, o acesso físico ao dispositivo é um risco para o usuário, sendo mais uma questão a ser considerada quanto a sua segurança.

No caso de dispositivos IoT que estão implantados em campos ou florestas para monitoramento, não há como garantir facilmente que não houve acesso físico a estes dispositivos por um invasor. Garantias desse tipo tornam o preço dos sensores mais elevados, o que o pode torná-los menos atrativos aos usuários finais. Dessa forma, partimos da premissa de que garantir que o dispositivo não foi acessado fisicamente é impossível, sendo assim, os dois fatores iniciais não garantem 100% de certeza na identificação de sujeitos (sensores).

Por último, tem-se o que se é (iii). Para seres humanos é comum a utilização de fatores biométricos, tais como digitais, identificação baseada na retina, na voz, entre outras formas. Esse é um importante fator na identificação de sujeitos humanos, pois, dada a vasta quantidade de informações que o sistema pode ter sobre seus usuários e suas características, tal fator torna-se muito difícil de sofrer um ataque, devido à complexidade de reprodução dessas características. Entretanto, mesmo características físicas inerentes ao sujeito do acesso podem ser falseadas ou replicadas, o que torna o fator (iii) frágil, quando utilizado isoladamente.

É sabido da utilização de engôdos para falsear dados biométricos, como dedos de silicone condutivo ou borracha (MATSUMOTO, 2002), no entanto, tais procedimentos requerem um nível maior de sofisticação para sua implementação. A replicação de características analógicas, tais como impressão digital, assinatura vocal, entre outras, traz um risco para situações onde exista maior sensibilidade das informações, a exemplo, podemos citar os dados médicos e os dados financeiros.

Entretanto, o principal problema para a autenticação em um dispositivo IoT utilizando o fator “o que se é”, é que os dispositivos são produzidos em série e compartilham o mesmo projeto e componentes de milhares de produtos produzidos. Dessa forma, suas características são iguais às de todos seus modelos, nas mesmas versões, tornando a "digital"de um dispositivo algo difícil de se encontrar. Como exemplo de fatores do tipo (iii), podem ser elegidos o UUID do *hardware*, uma identificação gerada no momento de seu registro, ou mesmo um certificado digital. No entanto, utilizar um certificado, em nosso contexto, excederia o número de *bytes* trafegados em alguns protocolos LPWAN, tendo em vista que o tamanho mínimo de um certificado é de 1024 *bytes*.

## 2.6 LPWAN

Protocolos de redes para redes dispositivos IoT evoluíram para protocolos baseados em LPWAN (*Low Power Wide-Area-Network*), em decorrência das limitações de processamento e energia em dispositivos móveis (KHAN; SALAH, 2018). O LPWAN tem uma menor taxa de transmissão, o que faz com que haja um menor consumo de energia, sendo uma solução adequada para dispositivos IoT que dependam de baterias e não necessitem transmitir grande quantidade de dados frequentemente. Essa é uma característica essencial e que diferencia esses tipos de protocolos de outros protocolos sem fio com maior taxa de transferência, como *Wi-Fi*. Alguns dos protocolos LPWAN mais comumente utilizados são LoRAWAN, NB-IoT e Sigfox. Tais tipos de protocolos, mesmo tento uma baixa taxa de transmissão, podem alcançar longas distâncias em suas comunicações.

LoRaWAN é uma tecnologia emergente que permite aos sistemas transmitirem uma baixa carga de dados a grandes distâncias e com um baixo custo (LIN; SHEN; MIAO, 2017). Essa tecnologia tem capacidades de transmissão de 0.3kpbs a 50kbps, dependendo da distância requerida e da interferência a qual o sinal está sujeito, e tem como principais características positivas a segurança, o tempo de vida de bateria, múltiplos canais de comunicação e diferentes

classes de dispositivos (NAOUI; ELHDHILI; SAIDANE, 2016). É um protocolo que provê meios de adição de novos nodos à rede de maneira segura (STEFANO; SIMONE; LORENZO, 2017).

*Narrowband Internet of Things* (NB-IoT) é um protocolo proposoto pela Third Generation Partnership Project (3GPP) em 2015 e padronizado em 2016, que apresenta as seguintes vantagens: aumento da capacidade e da cobertura do protocolo, que foi baseado no Long Term Evolution (LTE) (RASTOGI et al., 2020). Esse protocolo consome menos energia e largura de banda, enquanto tem desempenho comparável a outras formas de transmissão (ROUTRAY et al., 2019), além disso, pode usar a infraestrutura legada do LTE, visto que compartilham uma arquitetura similar, ou pode ser implantado de modo não celular (ROUTRAY; HUSSEIN, 2019), sendo essa uma escolha de projeto.

Sigfox é um protocolo que foi projetado para aplicações com baixo orçamento de transmissão de dados e para dispositivos de baixo custo, suportados por baterias e transmitindo dados a longas distâncias (ANANI; OUDA; HAMOU, 2019). As transmissões de dados acontecem com uma quantidade pequena de dados, a uma baixa taxa de transmissão e com pouca frequência. A taxa de transmissão é de 100 bps, com um número de transmissão de mensagens limitadas por dia, como 4 mensagens para comunicações DL e 140 para UL (LALLE et al., 2019). Existe também uma limitação de 12 *bytes* de tamanho para o *payload* de cada envio (MEKKI et al., 2019), o que faz com que a vocação deste protocolo seja para *smart metering* ou algum contexto similar.

Após se observar as características dos protocolos LPWAN, é possível notar a necessidade de uma abordagem de segurança leve nesse contexto. A utilização de mecanismos adotados em redes com maior capacidade de transmissão de dados não é adequada a um contexto com tanta limitação na quantidade de *bytes* trafegados e com velocidade de transmissão tão baixa. Entretanto, as redes LPWAN atrelam baixo custo de operação, baixo custo de projeto de dispositivos e longa distância de transmissão (até 50km com LoRa), o que as tornam de grande importância na evolução de *smart metering*, especificamente no nosso caso de monitoramento ambiental (*Environmental Sensing*). Em vista disso, é primordial que sejam adequados os mecanismos existentes para as restrições exigidas em sistemas LPWAN.

# 3 REVISÃO SISTEMÁTICA

Para um melhor entendimento da área de pesquisa desejada, foi proposta uma revisão sistemática, por meio da qual foi possível identificar desafios, oportunidades, outros autores que pesquisam a mesma área, principais periódicos onde acontecem as publicações, entre outras informações expostas neste estudo.

Esta seção apresenta uma pesquisa documental que utiliza revisão sistemática (SR –*Systematic Review*) como base, um método que diminui o viés em pesquisas científicas, pois, aplicando uma metodologia validada, é possível garantir a execução dos processos de maneira replicável. Por meio de passos bem estabelecidos, o pesquisador pode rastrear a aderência de interesse, sem ter um viés no processo metodológico (MOHER et al., 2009).

A SR auxilia em diversas áreas, desde a pesquisa bibliográfica para produção de um livro ou de aulas (KITCHENHAM et al., 2009), até a seleção de um portifólio de artigos para pesquisa de uma tese, dissertação ou trabalho de conclusão de curso. Alguns métodos de SR baseiam-se em evidências para criação de um documento vivo (MOHER et al., 2009), ou seja, um documento que evoluirá e deverá sofrer manutenção ao longo do tempo. Existem muitos motivos para a realização de uma SR, dentre os quais podemos citar (BUDGEN; BRERETON, 2006): a) sumarizar evidências existentes; b) identificar *gaps*; c) ajudar a posicionar pesquisas; e d) analizar suportes em evidências de uma hipótese.

Existem diversos métodos de revisão sistemática na literatura, e neste estudo analisaremos quatro. Os métodos escolhidos foram: Protocolo de Viera (VIEIRA et al., 2017), PRISMA *Statement* (MOHER et al., 2009), EBSE (*Evidence-based Software Engineering*) (KITCHENHAM; DYBA; JORGENSEN, 2004) e ProKnow-C (CHAVES et al., 2012). Cada metodologia foi criada com o mesmo propósito, mas em diferentes áreas do conhecimento, compartilhando algumas etapas/passos entre si e todas permitindo um levantamento sistemático do estado da arte.

Vieira et al. (2017) apresentam um protocolo em quatr passos distintos: (i) definição do objetivo da revisão; (ii) identificação da literatura da área; (iii) seleção do portifólio de arquivos; e (iv) análise e discussão sobre desafios, oportunidades e trabalhos relacionados. É um protocolo simples e de fácil execução, que permite ao pesquisador utilizar uma abordagem mais leve de revisão de literatura. Entretanto, tal protocolo não fornece uma análise gráfica dos dados levantados, não implementa definição de escopo através de filtros e não propõe uma exclusão parcial das publicações antes da leitura completa.

O PRISMA *Statement* é um processo de revisão sistemática que vem da área da saúde e é baseado em evidências. Inicialmente conhecido como QUOROM *Statement* (*QUality Of Reporting Of Meta-analyses*), ele teve seu nome modificado, com o intuito de incorporar também a revisão sistemática. Esse processo consiste de um *checklist* de 27 itens (MOHER et al., 2009) e um diagrama de fluxo de quatro fases, sendo mais orientado à área da saúde, por isso utiliza as SR com a finalidade de investigar custo-efetividade, diagnóstico ou perguntas prognósticas, entre outros aspectos. Apesar de ser aplicável a qualquer área do conhecimento, por meio de

algumas adaptações, sua origem na área médica pode ser um desafio para pesquisadores que não estão acostumados a ela.

*Evidence-Based Software Engineering* (EBSE) é uma metodologia de SR nascida com a finalidade de auxiliar o processo de decisão na manutenção e desenvolvimento de *software* (KITCHENHAM; DYBA; JORGENSEN, 2004), sendo composta por cinco fases e algumas tabelas de resultados. Suas fases são: (i) perguntas de pesquisa; (ii) processo de pesquisa; (iii) seleção dos estudos; (iv) garantia de qualidade; e (v) processo de extração de dados. Cada fase tem sua importância, mas cabe ressaltar que uma boa escolha das perguntas de pesquisa guiará o restante do processo de maneira adequada. Um revés desse método é a exigência de um grupo de pesquisadores para revisões por pares (KITCHENHAM et al., 2009). Tal fato pode ser um contratempo no caso de projetos com poucos participantes ou em um esforço solitário em uma pesquisa sem financiamento.

O ProKnow-C, desenvolvido para a área acadêmica, está disponível para utilização sem custos, gera dados estatísticos e portifólio, e não envolve um número grande de participantes na sua realização (CHAVES et al., 2012). O método ProKnow-C (*Knowledge Development Process - Constructivist*) foi desenvolvido em 2009 na UFSC (Universidade Federal de Santa Catarina), para a tomada de decisão na área de engenharia, e está organizado em quatro macroetapas: (i) seleção de portifólio bibliográfico; (ii) análise sistêmica; (iii) bibliometria; e (iv) pergunta de pesquisa.

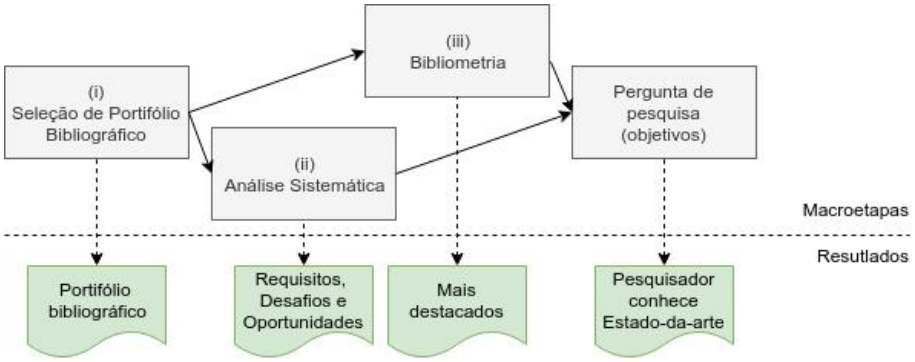



Figura 2 – Macroetapas do ProKnow-C

Cada método ou protocolo traz contribuições e desafios. Em decorrência disso, optou-se pela adoção, neste estudo, de um método-base que apresentasse o maior número de características desejadas para a revisão sistemática de literatura, tendo sido escolhido o processo ProKnow-C. Essa escolha se baseia no fato de o ProKnow-C ser um processo possível de ser realizado com somente um pesquisador, sob orientação de um pesquisador mais experiente. Outro fator importante é que esse método não é advindo da área de saúde ou mesmo foi proposto para seleção de tecnologias para manutenção/desenvolvimento de *software*, mas nasceu dentro da área acadêmica e com propósito acadêmico. Também é um método desenvolvido pela própria UFSC, instituição onde se ambienta e desenvolve-se esta pesquisa.

## 3.1 ADAPTAÇÕES FEITAS NO PROKNOW-C

Para este estudo, foi adaptada somente a fase de seleção de portifólio bibliográfico do ProKnow-C (CHAVES et al., 2012). As adaptações foram feitas com contribuições do EBSE (KITCHENHAM; DYBA; JORGENSEN, 2004) e do protocolo de Vieira (VIEIRA et al., 2017). Algumas fases foram integradas, algumas tiveram sua ordem alterada e outras sofreram pequenos acréscimos e adapatações em suas etapas.

Como principais modificações, podemos citar as seguintes alterações realizadas: a seleção das bases de dados como primeiro passo; a definição de palavras-chave transformou-se em engenharia da *query-string* e integra o teste de aderência das palavras-chave escolhidas; a coleta de dados é feita logo após a seleção inicial dos artigos e foi tornada mais completa; e os passos que descartam artigos no ProKnow-C foram integrados no estabelecimento de critérios (inclusão e exclusão). Os demais passos obedecem à mesma proposta do ProKnow-C. O procedimento adaptado pode ser visto na Figura 3.

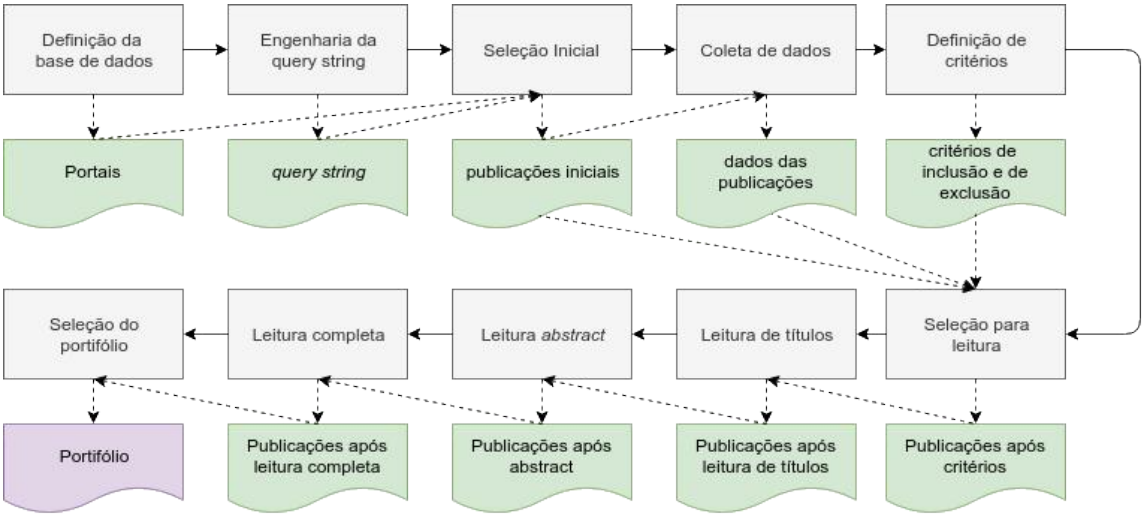



Figura 3 – Macroetapas do ProKnow-C

A definição das bases de dados é uma fase comum a diversos métodos. Para o PRISMA, esse é o sétimo item de seu *checklist*, denominado "fontes de informação"(MOHER et al., 2009). Para o EBSE, essa é uma sub-tarefa do processo de busca, ocorrida na segunda fase do método, logo após a definição das palavras de busca. Neste estudo, trouxemos essa fase para o início, sem que houvesse qualquer penalização ao restante do processo.

O passo de engenharia de *query string* é um dos primeiros, e a qualidade com que ele será executado impactará os demais passos. Nesse passo, são realizados os seguintes procedimentos: (a) definição da palavra de busca (*query string*); (b) relação entre as suas partes; (c) utilização de caracteres coringas; e (d) realização de um teste de aderência da *query string*.

Feita a busca, é hora de realizar a seleção inicial dos artigos. Nessa fase, é indicada a utilização de uma ferramenta de suporte à gestão de artigos acadêmicos, como o Mendeley[1].

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

Tal ferramenta permite a gestão dos artigos em pastas, tornando possível organizar a base de artigos que foi selecionada em cada fase, formando as bases de artigos e permitindo que haja rastreabilidade nas decisões de inclusão e exclusão dos estudos.

A coleta de dados dos artigos é a identificação do número de citações, ano de publicação, autores e fator de impacto do veículo onde o artigo foi publicado, e qualquer outro dado peculiar ao estudo em questão. Desse modo, é possível estabelecer um ponto de corte quanto à qualidade esperada de um artigo selecionado. O fator de impacto que foi adicionado nessa etapa do processo é uma adaptação feita no ProKnow-C. Originalmente, tal variável seria analisada na macroetapa bibliometria, no entanto, viu-se a necessidade de coleta desse dado para utilização nos critérios de filtragem.

A definição de critérios estabelece os parâmetros para inclusão e exclusão de artigos encontrados na primeira busca, além de auxiliar na tomada de decisão. Essa é uma adaptação que incorpora os critérios do EBSE (KITCHENHAM; DYBA; JORGENSEN, 2004) no ProKnow-C, e que trabalha somente com limites nessa fase. Os limites originais do ProKnow-C, de atualidade e representatividade de um artigo, são, fatores importantes em um primeiro contato com uma grande massa de artigos, e foram incluídos como critérios, além da disponibilidade para *download*. Por se tornar uma etapa mais restritiva do que a original, acredita-se que tal adaptação trará um grupo inicial de artigos mais conciso, com maior qualidade e com maior aderência ao tema.

Na seleção para leitura de títulos, os artigos são escolhidos de acordo com critérios preestabelecidos no passo anterior. Tais critérios serão confrontados com os dados adquiridos na quarta etapa do processo e submetidos aos critérios de inclusão e exclusão. Na fase de leitura de *abstracts* deverá ser possível verificar o alinhamento do artigo com a pesquisa pretendida, e a leitura completa do texto buscará confirmar esse alinhamento. Nessa fase será possível verificar detalhes da publicação, permitindo que qualquer artigo que tenha sido mal compreendido e selecionado de maneira errônea possa ser removido do processo.

Ao final, tem-se o portfólio de artigos selecionados e com a máxima aderência dentro do conjunto inicial. Frente ao conjunto inicial, baseado somente em palavras de busca, a seleção final do portfólio apresenta de forma sistemática um subconjunto de artigos com mais qualidade e aderência. Após passar por vários crivos, os artigos poderão ser utilizados para descrever o cenário atual, oportunidades de pesquisa e avanços nas áreas selecionadas pelo pesquisador que realiza a SR.

## 3.2 SELEÇÃO DE PORTFÓLIO

A escolha dos portais, primeiro passo do processo adaptado do ProKnow-C, deu-se baseada na disponibilidade de acesso a artigos por meio dos convênios da CAPES com portais de periódicos, todos com acesso para a leitura integral dos artigos. Foram selecionados os seguintes portais: (1) ACM Digial Libray, (2) IEEExplorer, (3) SciELO, (4) ScienceDirect e (5) Scopus.

O processo de engenharia de *query strings* deu-se conforme segue. Inicialmente foram buscadas as publicações dos últimos anos em cada área de interesse deste estudo, tendo sido elegidas as seguintes áreas: IoT, *Edge Computing*, *Fog Computing*, *Mist Computing*, *Multi Factor Authentication* e LWPAN. Os três paradigmas (*Edge*, *Fog* e *Mist*) foram escolhidos para tomada de decisão e para conferir suporte quantitativo sobre qual paradigma escolher como contexto do desenvolvimento e aplicação da proposta aqui descrita. Os resultados podem ser vistos na Figura 4.


Figura 4 – Evolução das áreas nos últimos três anos

Como se pode observar na Figura 4, a área de IoT já está com a quantidade de pesquisas em redução nos últimos anos. Já as três áreas de paradigmas de computação têm tido um crescimento maior em relação a outras áreas, como, por exemplo, a área de autenticação multifator. A autenticação multifator e a LWPAN têm se mostrado com pouca variação em relação à quantidade de pesquisas. Tal fato pode indicar que essas áreas ainda carecem de melhores soluções e apresentam um espaço para pesquisas.

Após a definição da relação entre os conjuntos de publicações obtidas a partir das áreas, foram feitas as consultas para cada paradigma. O resultado pode ser observado na Tabela 1. Para cada *string* de pesquisa, foram realizadas três consultas: com os dados de 2018, 2019 e 2020. Foram subtraídos dos valores acumulados de dados dos anos posteriores, objetivando-se identificar quais termos têm tido tendências crescentes de publicação ao longo dos últimos anos.

| nro | em | 2018 | 2019 | 2020 |
|---|---|---|---|---|
| 1 | "multi-factor authentication" iot "fog computing" | 32 | 50 | 17 |
| 2 | "multi-factor authentication" iot "edge computing" | 21 | 55 | 19 |
| 3 | "multi-factor authentication" iot "mist computing" | 0 | 2 | 0 |
| 4 | "multi-factor authentication" iot "fog computing"lwpan | 4 | 7 | 5 |
| 5 | "multi-factor authentication" iot "edge computing"lwpan | 4 | 5 | 4 |
| 6 | "multi-factor authentication" iot "mist computing"lwpan | 0 | 0 | 0 |

Tabela 1 – Consultas realizadas no Google Acadêmico para avaliação das strings de busca (acervo próprio)

Pode-se notar que as grandes áreas da pesquisa têm muitos artigos publicados, são elas: IoT, *edge computing*, *fog computing* e *multi-factor authentication*. Entretanto, áreas mais recentes e específicas, como LWPAN e *mist computing*, ainda estão com publicações menos expressivas. Com foco específico, foram encontradas zero publicações para o caso da *mist computing* (linha 6 da Tabela 1) e seis para a *fog computing* (linha 4 da Tabela 1). *Fog computing* (linha 5 da Tabela 1) trouxe 16 artigos publicados desde 2018[2].

| Portal | Publicações (2018 a 2020) | (2019 a 2021) |
|---|---|---|
| ACM | 01 | 02 |
| IEEExplorer | 00 | 02 |
| SciELO | 00 | 00 |
| ScienceDirect | 06 | 19 |
| Scopus | 17 | 03 |
| Total | 24 | 26 |

Tabela 2 – Publicações por portal (acervo próprio)

Por meio da validação da *query string*, apresentada na Tabela 1, optou-se pela frase: ("*multi factor authentication*"and "*fog computing*"and iot), linha 1 dessa mesma tabela. Efetuadas as consultas nos portais escolhidos, obteve-se um total de vinte e quatro publicações envolvendo o tema escolhido. A distribuição dos resultados por portal pode ser avaliada na Tabela 2.

Os critérios de inclusão escolhidos foram listados na Tabela 3, enquanto os critérios de exclusão pode ser averiguados na Tabela 4.

Dentre os artigos que atenderam ao critério de inclusão, alguns foram eliminados por meio dos critérios de exclusão (Tabela 4). Tais critérios permitem uma melhor delimitação de escopo e evitam o viés durante o processo (KITCHENHAM; DYBA; JORGENSEN, 2004).

[2] No final do ano de 2021 foi feita uma atualização na revisão sistemática contida neste capítulo

| | |
|---|---|
| i1 | o artigo deve estar disponível para leitura completa; |
| i2 | o artigo deverá ter no máximo três anos desde sua publicação; |
| i3 | o artigo deverá estar na língua inglesa; |
| i4 | artigos completos. |

Tabela 3 – Critérios de inclusão

Artigos repetidos, publicações em diferentes veículos sobre o mesmo experimento, entre outros, foram removidos do conjunto de artigos durante essa fase do processo.

| | |
|---|---|
| e1 | artigo repetido; |
| e2 | relato repetido, artigo diferente relatando o mesmo experimento em diferentes momentos; |
| e3 | artigo informal e sem relevância científica; |
| e4 | patentes ou relatórios técnicos; |
| e5 | veículos de publicação deverão ter fator de impacto acima de 1,0; |
| e6 | deve atender ao número mínimo de citações. |

Tabela 4 – Critérios de exclusão

Após o crivo dos critérios ter sido aplicado aos 24 artigos selecionados, restaram somente 19 para seleção inicial. Durante o processo de leitura de títulos, foram eliminados 2 artigos, restando somente 17 publicações nessa fase. Já na leitura de *abstracts*, restaram somente 12 publicações. Seguindo-se com a leitura completa, mais 2 artigos foram removidos. Ao final, o portfólio obteve um total de 10 publicações. Uma visão completa do processo é apresentada na Tabela 5.

| Etapa | Início | Fim | 2021 Início | 2021 Fim |
|---|---|---|---|---|
| Seleção inicial (critérios) | 24 | 19 | 26 | 24 |
| Leitura de títulos | 19 | 17 | 24 | 15 [3] |
| Leitura de abstracts | 17 | 12 | 15 | 10 |
| Leitura completa | 12 | 10 | 10 | 08 |
| Portfólio final | 10 | 10 | 08 | 08 |

Tabela 5 – Publicações por fase do processo

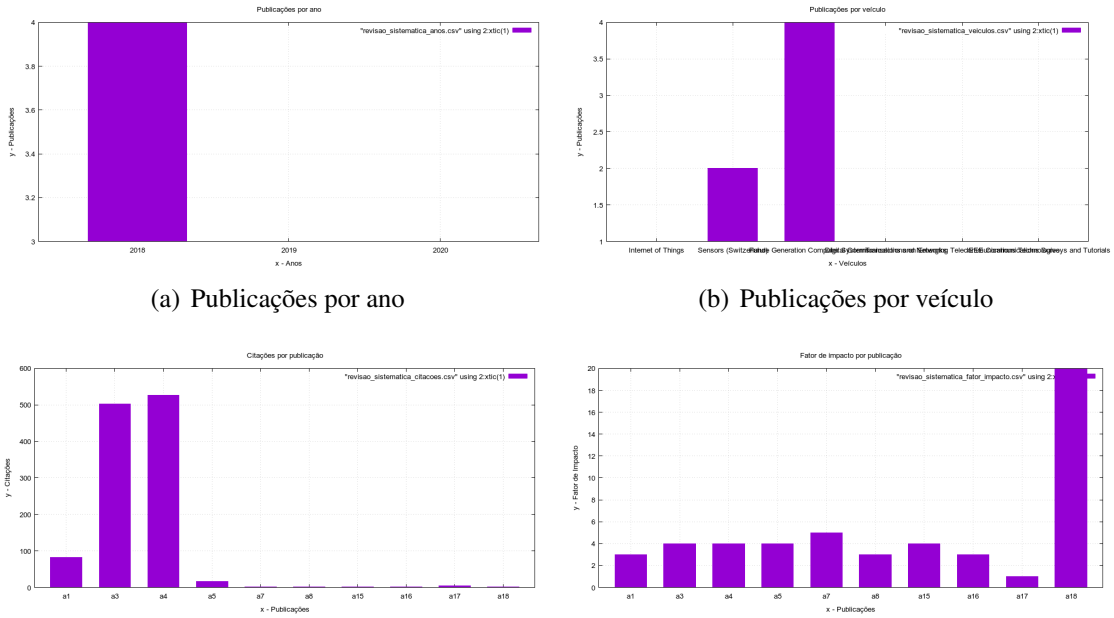
## 3.3 BIBLIOMETRIA

Essa macroetapa apresenta quantitativamente os dados do portfólio, trazendo as informações colhidas apresentadas de forma gráfica, permitindo uma visualização e uma análise rápida das principais variáveis do nosso portfólio bibliográfico. As variáveis analisadas em nosso estudo são: ano, autor, veículo, citações e fator de impacto. Como análise de variáveis,

[3] Artigos com mais de dois anos devem ter mais de 10 citações; artigos com um ano devem ter ao mínimo uma citação; artigos do ano corrente não devem ser excluídos mediante este critério. Observação: as duas últimas colunas são referentes a atualização na revisão sistemática que aconteceu no fim de 2021

temos os seguintes gráficos: publicações por ano, publicações por veículo, citações por publicação e fator de impacto por publicação. O gráfico de publicações por autor não foi inserido neste estudo, pois cada autor aparece somente em uma publicação do portfólio, o que torna o gráfico pouco representativo.

A distribuição dos artigos por ano pode ser vista na Figura 5-(a), contendo quatro ocorrências no ano de 2018, três ocorrências no ano de 2019 e três ocorrências no ano de 2020. A autenticação tem apresentado estudos significativos na área, considerando que esse é um tema novo e um dos desafios da computação de borda e, por sua vez, da computação em nevoeiro (*Fog Computing*) também.




(a) Publicações por ano

(b) Publicações por veículo

(c) Citação por publicação

(d) Fator de impacto por publicação

Figura 5 – Resultados da bibliometria

No portfólio podemos notar a concentração de publicações relevantes por veículo na Figura 5-(b). Com maior número de publicações, tem-se o *Journal Future Generation Computer Systems*, com 4 artigos (KHAN; SALAH, 2018; ROMAN; LOPEZ; MAMBO, 2018; AU et al., 2018; ESINER; DATTA, 2019); em segundo lugar, está o *Sensors*, com 2 artigos; e os demais veículos estão empatados em terceiro lugar, todos com uma publicação. Pode-se notar, por meio desses dados, que os veículos selecionados abordam tecnologias mais recentes, o que reforça a hipótese de que esse é um tema em aberto e relevante para a academia.

Já quanto ao número de citações, conforme apontado na Figura 5-(c), as publicações que obtiveram mais destaque foram as a4 (ROMAN; LOPEZ; MAMBO, 2018) e a3 (KHAN; SALAH, 2018) (com 527 e 503 citações, respectivamente). Nota-se que tais publicações têm somente dois anos de existência e um significativo número de citações. Em terceiro e quarto lugar estão a1 (ALI; AWAD, 2018) e a5 (AU et al., 2018), com 82 e 17 citações, respectivamente, e com o ano de publicação também em 2018.

Quanto ao fator de impacto de cada publicação, conforme apontado na Figura 5-(d) , a publicação a18 (CAO et al., 2019) apresenta um valor de 20,23 de FI. Em segundo lugar temos o artigo a7 (HADDADPAJOUH et al., 2019), com 5,768, seguido pelos quatro artigos da *Future Generation Computer Systems* (a3 (KHAN; SALAH, 2018), a4(ROMAN; LOPEZ; MAMBO, 2018), a5 (AU et al., 2018), a15 (ESINER; DATTA, 2019)), todos com 4,639. É importante notar que todos os artigos selecionados no portfólio apresentam um fator de impacto relevante, sendo que somente um está abaixo de 3, o artigo a17 (DEEP; ZHENG; HAMEY, 2019b).

## 3.4 REVISÃO SISTEMÁTICA

Com o portfólio definido (Tabela 6), pode ser feita a análise das publicações selecionadas. A análise foi feita seguindo a lente dos desafios e oportunidades de segurança de IoT em *fog computing*. Como desafios, pode-se considerar todas as barreiras trazidas pela adoção desse novo paradigma de FC, junto a restrições de IoT e LWPAN. Por oportunidades de pesquisa, entende-se os *gaps*, adequações de tecnologias, necessidades de mercado e criação de novas tecnologias; com a finalidade de atender uma demanda futura para pesquisas ou empresas.

| | |
|---|---|
| a1 | ALI, Bako; AWAD, Ali Ismail. Cyber and physical security vulnerability assessment for IoT-based smart homes. Sensors, v. 18, n. 3, 2018. |
| a3 | KHAN, Minhaj Ahmad; SALAH, Khaled. IoT security: Review, blockchain solutions, and open challenges. Future Generation Computer Systems, v. 82, p. 395-411, 2018. |
| a4 | ROMAN, Rodrigo; LOPEZ, Javier; MAMBO, Masahiro. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. Future Generation Computer Systems, v. 78, p. 680-698, 2018. |
| a5 | AU, Man Ho et al. Privacy-preserving personal data operation on mobile cloud—Chances and challenges over advanced persistent threat. Future Generation Computer Systems, v. 79, p. 337-349, 2018. |
| a7 | HADDADPAJOUH, Hamed et al. A survey on internet of things security: Requirements, challenges, and solutions. Internet of Things, p. 100129, 2019. |
| a8 | DEEP, Gaurav et al. Authentication Protocol for Cloud Databases Using Blockchain Mechanism. Sensors, v. 19, n. 20, p. 4444, 2019. |

| | |
|---|---|
| a15 | ESINER, Ertem; DATTA, Anwitaman. Two-factor authentication for trusted third party free dispersed storage. Future Generation Computer Systems, v. 90, p. 291-306, 2019. |
| a16 | SHA, Kewei et al. A survey of edge computing based designs for IoT security. Digital Communications and Networks, 2019. |
| a17 | DEEP, Samundra; ZHENG, Xi; HAMEY, Len. A survey of security and privacy issues in the Internet of Things from the layered context. arXiv preprint arXiv:1903.00846, 2019. |
| a18 | CAO, Jin et al. A Survey on Security Aspects for 3GPP 5G Networks. IEEE Communications Surveys & Tutorials, 2019. |
| a19 | KAMBOU, Samy; BOUABDALLAH, Ahmed. A Strong Authentication Method for Web/Mobile Services. Cyber Security and Cloud Computing (CSCloud), 2019. |
| a20 | KHALLID, Haqi et al. SELAMAT: A New Secure and Lightweight Multi-Factor Authentication Scheme for Cross-Platform Industrial IoT Systems. Sensors. 2021. |
| a21 | DE SMET, Ruben et al. Lightweight PUF based authentication scheme for fog architecture. Wireless Networks, 2021. |
| a22 | BANERJEE, Soumya et al. Multi-Authority CP-ABE-Based user access control scheme with constant-size key and ciphertext for IoT deployment. Journal of Information Security and Applications, 2020. |
| a23 | KALARIA, Rudri et al. A Secure Mutual authentication approach to fog computing environment. Computers & Security, 2021. |
| a24 | LOFFI, Leandro et al. Mutual authentication with multi-factor in IoT-Fog-Cloud environment. Journal of Network and Computer Applications, 2021. |
| a25 | LI, Jianhua et al. A fast and scalable authentication scheme in IOT for smart living. Future Generation Computer Systems, 2021. |

| | |
|---|---|
| a23 | FOTOUHU, Mahdi et al. A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT. Computer Networks, 2020. |

Tabela 6 – Portfólio gerado a partir da revisão sistemática

Uma visão mais detalhada da revisão sistemática pode ser apreciada na seção 3.5, onde são expostos os desafios e as oportunidades encontradas no portfólio bibliográfico para a área de pesquisa proposta. A partir da lente elegida, é possível fazer um recorte na literatura e visualizar os desafios de maneira sistemática e pela luz do método científico.

## 3.5 DESAFIOS E PROBLEMÁTICA

Iniciando com o resultado da análise da revisão bibliográfica sistemática apresentada anteriormente, temos a seção de Desafios e Problemática. Como podemos observar, a Tabela 7 nos demonstra que os desafios mais importantes para IoT, segundo a literatura escolhida, são: autenticação, privacidade e controle de acesso, empatados em primeiro lugar. Logo em seguida temos a confidencialidade, em segundo lugar, e a disponibilidade, em terceiro. Na sequência, em quarto lugar, temos as restrições energéticas, as restrições de rede e a integridade. Por último, sendo citadas somente por dois artigos cada, estão a escalabilidade, *openness* e flexibilidade, virtualização, latência e mobilidade. Esses são fatores ligados a temas que suportam a IoT, como sistemas distribuídos e computação de borda.

Sendo uma área mais recentemente explorada, há um grande número de oportunidades em aberto em IoT, tendo sido listadas pelos autores as seguintes: alta vulnerabilidade a DoS; possibilidade de violação; e pertubação e acesso não autorizado de dispositivos na *edge*, *building blocks* simples e com pouca sobrecarga (TIBURSKI et al., 2019). Para os autores, as pesquisas apresentam uma visão mais ampla da IoT e de seu funcionamento geral, tendo em vista a consideração dos *building blocks* e mesmo violações físicas em dispositivos nas proximidades da borda do sistema (BERTIN et al., 2019; ALI; AWAD, 2018; KHAN; SALAH, 2018). Um importante aspecto é a necessidade da resolução da vulnerabilidade quanto a ataques de negação de serviço, devido a problemas de segurança.

De uma maneira mais detalhada, podemos analisar cada trabalho que aparece no portfólio e que representa o estado-da-arte na área de pesquisa. Este trabalho segue analisando o portfólio a partir das lentes de desafios, trabalhos futuros e problemas não resolvidos completamente, e que são apresentados nessa literatura selecionada. Ao todo, são analisados diversos tipos de trabalhos que contribuem de maneira importante para construção do cenário e motivação do nosso trabalho. Ainda, tais trabalhos nos permitem caracterizar e justificar os aspectos de inovação necessários para resolução dos problemas aqui apresentados.

Também acerca dos desafios em *fog computing*, que estão diretamente associadas ao

| Requisito | Publicações |
|---|---|
| Escalabilidade | Bertin et al. (2019), Cao et al. (2019) |
| Heterogeneidade | Bertin et al. (2019), Deep, Zheng e Hamey (2019a), Vaquero e Rodero-Merino (2014), Cao et al. (2019) |
| *Openness* e flexibilidade | Bertin et al. (2019), Cao et al. (2019) |
| Autenticação | Bertin et al. (2019), Deep, Zheng e Hamey (2019a), Yi et al. (2015), Cao et al. (2019), Ali e Awad (2018), Roman, Lopez e Mambo (2018), HaddadPajouh et al. (2019), Sha et al. (2019), Deep et al. (2019), Deep, Zheng e Hamey (2019b), Khan e Salah (2018) |
| Privacidade | Bertin et al. (2019), Deep, Zheng e Hamey (2019a), Yi et al. (2015), Cao et al. (2019), Ali e Awad (2018), Roman, Lopez e Mambo (2018), HaddadPajouh et al. (2019), Sha et al. (2019), Deep et al. (2019), Deep, Zheng e Hamey (2019b), Khan e Salah (2018) |
| Controle acesso | Bertin et al. (2019), Deep, Zheng e Hamey (2019a), Yi et al. (2015), Cao et al. (2019), Ali e Awad (2018), Roman, Lopez e Mambo (2018), HaddadPajouh et al. (2019), Sha et al. (2019), Deep et al. (2019), Deep, Zheng e Hamey (2019b), Khan e Salah (2018) |
| Ease to use security | Bertin et al. (2019), Cao et al. (2019), Ali e Awad (2018), HaddadPajouh et al. (2019), Khan e Salah (2018) |
| Consumo de rede | Deep, Zheng e Hamey (2019a), Yi et al. (2015), Vaquero e Rodero-Merino (2014), Cao et al. (2019), Deep, Zheng e Hamey (2019b), Khan e Salah (2018) |
| Consumo de energia | Deep, Zheng e Hamey (2019a), Yi et al. (2015), Vaquero e Rodero-Merino (2014), Cao et al. (2019), Deep, Zheng e Hamey (2019b), Khan e Salah (2018) |
| Confidencialidade | Deep, Zheng e Hamey (2019a), Yi et al. (2015), Vaquero e Rodero-Merino (2014), Cao et al. (2019), Ali e Awad (2018), Roman, Lopez e Mambo (2018), HaddadPajouh et al. (2019), Deep et al. (2019), Deep, Zheng e Hamey (2019b), Khan e Salah (2018) |
| Disponibilidade | Deep, Zheng e Hamey (2019a), Yi et al. (2015), Vaquero e Rodero-Merino (2014), Cao et al. (2019), HaddadPajouh et al. (2019), Deep, Zheng e Hamey (2019b), Khan e Salah (2018) |
| Integridade | Deep, Zheng e Hamey (2019a), Yi et al. (2015), Vaquero e Rodero-Merino (2014), Cao et al. (2019), Deep, Zheng e Hamey (2019b), Khan e Salah (2018) |
| Virtualização | Yi et al. (2015), Roman, Lopez e Mambo (2018) |
| Latência | (YI et al., 2015; ROMAN; LOPEZ; MAMBO, 2018) |
| Busca e sincronização | Vaquero e Rodero-Merino (2014), Cao et al. (2019), Khan e Salah (2018) |
| Mobilidade | Vaquero e Rodero-Merino (2014), Cao et al. (2019) |
| Detecção de intrusão | Vaquero e Rodero-Merino (2014), Cao et al. (2019), Ali e Awad (2018), Roman, Lopez e Mambo (2018), Khan e Salah (2018) |

Tabela 7 – Desafios por publicações

sistema IoT, Vaquero e Rodero-Merino (2014) trazem os seguintes: busca; sincronização; limitações de computação e armazenamento; gerenciamento; segurança; padronização; responsabilidade; monetização; e mobilidade dos programas. Mesmo sendo uma visão bem focada na infraestrutura que suporta a IoT, os autores trazem questionamentos relevantes sobre como garantir a responsabilização em nodos da *fog*; quem será responsável por sua gestão e faturamento; a segurança de tais nodos; e a infraestrutura de rede em que houve a implantação.

Yi et al. (2015) apontam desafios em aberto nas questões relacionadas à infraestrutura da *fog computing*, também muito importantes para este estudo. O autor lista como desafios: a escolha de opções de virtualização; a luta contra o problema da latência; a gestão da rede; a segurança; e a privacidade. Em específico, a segurança e privacidade despontam como aspectos relevantes de pesquisa em nosso contexto.

A publicação de Ali e Awad (2018) foca nos riscos de segurança para *smarthomes* e faz uma avaliação de risco utilizando OCTAVE. Além disso, ela traz a identificação de recursos na modelagem de ameaças que devem ser avaliados quanto ao risco de segurança, sendo eles: credenciais de usuário; dados móveis pessoais e de *apps*; informações coletadas pelos dispositivos; inventário dos dispositivos; informação de *log*; informação transmitida pelo *gateway*; informações de configuração; acesso a vídeos de segurança; rastreamento de localização; e acesso a recursos de informação. Como soluções para mitigação dos problemas, os autores apresentam os mecanismos de autenticação, segurança na transmissão de dados, substituição de informações iniciais de configuração, segurança física dos dispositivos e sistemas de detecção de intrusão.

Segundo Roman, Lopez e Mambo (2018), as pesquisas de segurança no paradigma de borda ainda estão nascendo. Alguns desafios desse paradigma são organizados pelos autores em quatro grupos: infraestrutura de rede, infraestrutura de serviço, infraestrutura de virtualização e dispositivos de usuário. A segurança aparece como fator importante em cada um dos cinco grupos, pois qualquer dispositivo controlado por um adversário pode ser reprogramado para inserir informações erradas no sistema (ROMAN; LOPEZ; MAMBO, 2018). No estudo citado, também são propostos mecanismos de segurança para resolução dos problemas listados, tais como: identidade e autenticação; sistemas de controle de acesso; protocolos e segurança de rede; gestão de confiança; detecção de intrusão; privacidade; virtualização; tolerância a falhas e resiliência; e forense.

Na mesma linha temos Khan e Salah (2018), que trazem um conjunto de contribuições: taxionomia e categorização de questões de segurança em IoT; análise paramétrica de ameaças de segurança; características de *blockchain*; e problemas de segurança em aberto na IoT. Como alguns dos requisitos, os autores trazem: privacidade, confiabilidade e integridade dos dados; autenticação, autorização e *accounting*; disponibilidade de serviços; eficiência energética; e ponto único de falha.

Continuando com Bertin et al. (2019), os autores apresentam pesquisas em aberto quanto à escalabilidade, gestão de heterogeneidade, *openness* e flexibilidade; resolução de identidade de objetos, gerenciamento de dados pessoais, provimento de políticas de controle de acesso dinamicamente e segurança de fácil utilização para o usuário final. Tais autores levam em conta o sistema como um todo, mas sem entrar no aspecto físico, e consideram desde o aspecto da usabilidade que prevenirá o usuário de uma má configuração no dispositivo, também levando em conta aspectos de sistemas distribuídos que são de mais baixo nível, como *openness*, escalabilidade e heterogeneidade. Além de aspectos de segurança, como políticas de controle de acesso e identificação de objetos, trazendo à tona a necessidade da gestão de identidade.

Deep, Zheng e Hamey (2019a) afirmam que os problemas em aberto são: consumo de largura de banda e energia; complexidade; sensoriamento; e computação leve. E ainda acrescentam requisitos de segurança para os sistemas IoT, como confidencialidade, disponibilidade e integridade, que são os três pilares da segurança de informação, de acordo com Stallings (STALLINGS et al., 2012). Os autores focam em aplicações que exigem restrições quanto a uso de recursos, tais como energia e largura de banda. Mesmo que a segurança tenha sido levantada como uma questão de requisitos em seu trabalho, a necessidade de um mecanismo de segurança que dê suporte às restrições e desafios citados pelos autores é de suma importância.

No trabalho de Cao et al. (2019) são listados alguns dos problemas existentes na utilização de IoT com 3GPP 5G. Dentre eles, figuram o acesso massivo e concorrente de dispositivos IoT, a necessidade de uma acesso diferenciado para diferentes tipos de dispositivos, a proteção de privacidade e os mecanismos leves de segurança. O trabalho demonstra os desafios da 3GPP 5G e das tecnologias associadas, como a LoWPAN e partes constituintes da arquitetura 5G. O estudo ainda reforça que essa nova tecnologia que está chegando vem ampliar a perspectiva da quantidade de dispositivos conectados.

Trazendo uma abordagem em três camadas, HaddadPajouh et al. (2019) apresentam os desafios estratificados por camadas, levantando a problemática da má configuração de dispositivos, a falta de boa gestão de identidades, o controle de acesso inapropriado, a violação da infraestrutura de segurança e a violação de privacidade. Além disso, os autores reforçam a importância da autenticação e postulam que a implementação de um bom mecanismo de segurança em todo o ambiente computacional garante que dados privados serão acessíveis somente por partes (entidades computacionais ou pessoas) que tenham permissão de acesso e que sejam confiáveis.

Sha et al. (2019) pressupõem que muitos dos mecanismos de segurança existentes não são aplicáveis a dispositivos fim de IoT. Os autores apresentam um cenário onde há a utilização de dispositivos inteligentes funcionando integradamente com o paradigma de computação em borda. Eles apresentam também algumas pesquisas em aberto, tais como: segurança da camada de borda e acesso não confiável a ela; segurança *cross-domain* distribuída com *machine-learning*; protocolos leves de comunicação *device-edge*; sistema operacionais seguros; e máquinas virtuais leves. Os autores afirmam ainda que a autenticação e a autorização são mecanismos cruciais de segurança para se evitar muitos tipos de ataques, entre eles DDoS.

O trabalho de Deep et al. (2019) traz uma proposta de utilização de *blockchain* para segurança no armazenamento de dados na *cloud* e apresenta as falhas de segurança nos atuais mecanismos de segurança, quanto a ataques internos, mesmo com utilização de autenticação em dois fatores, propondo um mecanismo para autenticação que permita evitar esses ataques internos. Seus principais interesses são a garantia da autorização de acesso a informações e à privacidade dos dados armazenados na nuvem.

Deep, Zheng e Hamey (2019b) revisam e analisam a segurança em IoT utilizando uma arquitetura em quatro camadas. Para a camada de aplicação, eles apontam como preocupações a privacidade de dados e o controle de acesso; para a camada de *middleware*, listam integridade

e confiabilidade; já na camada de redes, autenticação e integridade; e, por último, na camada de percepção, os autores trazem como requisitos a integridade, a autenticação e a confidencialidade. Os autores ainda observam que soluções leves são uma área importante de pesquisas futuras, devido à natureza restrita dos dispositivos.

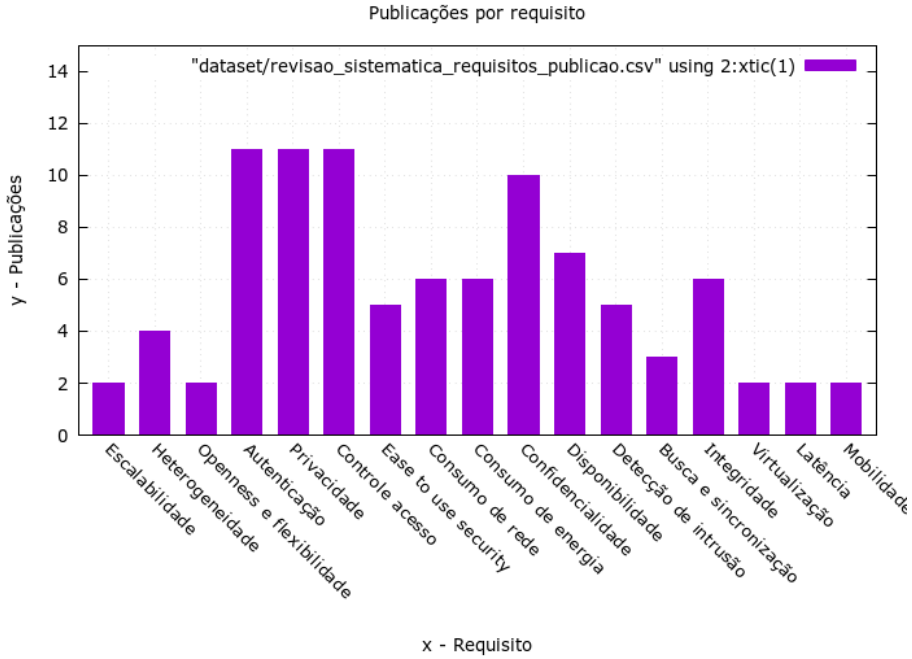



Figura 6 – Número de publicações por requisito encontrado

A partir desse levantamento de desafios feito no portfólio bibliográfico obtido na Revisão Bibliográfica Sistemática (Figura 6), foi possível confirmar a relevância do tema escolhido, ou seja, a autenticação para IoT em dispositivos restritos. De acordo com Khan e Salah (2018), as soluções de segurança precisam ser adaptadas às arquiteturas, sendo essa adaptação um desafio que se renova a cada salto tecnológico que ocorre, como é o caso das LPWAN e o advento da 5G/6G, entre outras evoluções tecnológicas atuais.

# 4 SOLUÇÃO PROPOSTA

Neste capítulo são apresentados os fundamentos do nosso objeto de pesquisa, o mecanismo de autenticação multifator com reputação (MFA_R). Com base na exposição da motivação e dos trabalhos existentes, este capítulo traz um detalhamento da solução do problema. As Soluções Existentes apresentam trabalhos relevantes na área e que são soluções similares ou substitutas na resolução dos problemas citados na seção de desafios. E, por fim, é detalhada a nossa solução proposta e implementada para os mesmos problemas e desafios.

Ressalta-se que a seção de Soluções Existentes se baseia na revisão bibliográfica sistemática já apresentada neste trabalho. Para mais detalhes sobre a metodologia e os dados da revisão bibliográfica sistemática, o leitor deverá acessar o capítulo de Revisão Bibliográfica Sistemática (3). É importante também esclarecer que alguns trabalhos correlatos (Soluções Existentes) são provenientes do processo de *Backward Snowbolling* (BSB) and *Forwarding Snowbolling* (FSB), não estando tais processos contemplados no corpo desta obra.

## 4.1 SOLUÇÕES EXISTENTES E TRABALHOS CORRELATOS

Este trabalho tem o foco na segurança entre a comunicação e a utilização de recursos em sistemas IoT por parte dos dispositivos. Muitos trabalhos têm o foco em autenticação, privacidade e mecanismos de controle de acesso a recursos do sistema IoT por parte dos usuários, ou seja, do cliente para a *cloud/fog*. Acreditamos que a segurança desses sistemas também deverá levar em conta a relação dispositivo-dispositivo e dispositivo-*gateway*, focando-se, dessa forma, na segurança da camada de percepção (sensores) e de rede (comunicação sensor-*gateway-fog*), que, de acordo com Grade, Zhang e Kumar (GADRE; ZHANG; KUMAR, 2019), é um dos primeiros problemas que deve ser resolvido.

Atualmente, vários esforços foram empreendidos para buscar maior segurança na autenticação de sistemas IoT, o que traz a necessidade de analisarmos as soluções atuais e correlacioná-las às necessidades aqui encontradas (Figura 6), na busca das soluções existentes e problemas semelhantes que estão presentes na literatura. Como consequência, trazemos aqui alguns trabalhos relacionados à análise do problema e suas possíveis soluções.

Lee et al. (2009) propõem um mecanismo de autenticação baseado em assinatura de grupo. Tal mecanismo promove uma autenticação anônima e herda características da assinatura em grupo, tais como: corretude, incapacidade de forja, anonimidade, não rastreabilidade e exclusão de conluio de parte do grupo para geração de chaves. Além disso, exime os provedores de serviço de coletarem informações de cada indivíduo do grupo para fornecer o serviço, provendo privacidade para o membro do grupo que está utilizando o serviço em questão.

Tsai e Lo (2015) propõem um esquema de autenticação para dispositivos móveis na *Mobile Cloud Computing* (MCC) distribuída, que leva em conta a capacidade limitada dos dispositivos móveis. O esquema proposto suporta autenticação mútua e anonimato sem SSL, utilizando uma chave para múltiplos provedores de serviço. A terceira parte somente participa

| # | Publicação | Solução | Tipo | Contexto |
|---|---|---|---|---|
| 1 | Lee et al. (2009) | Método | Autenticação | Assinatura em grupo |
| 2 | Tsai e Lo (2015) | Esquema | Autenticação | MCC distribuída |
| 3 | Fremantle e Aziz (2016) | Aplicação | Privacidade | OAuth e IoT |
| 4 | Ali et al. (2018) | Mecanismo | Criptografia | Dados biométricos |
| 5 | Ometov et al. (2018) | Mecanismo | Autenticação | MFA |
| 6 | Anakath, Rajakumar e Ambika (2019) | Mecanismo | Autenticação | MFA e OTP |
| 7 | Chaudhry et al. (2019) | Melhoria | Autenticação | Esquema Tsai e Lo |
| 8 | Esiner e Datta (2019) | Mecanismo | Privacidade | 2FA e OTP |
| 9 | Gomi et al. (2019) | Mecanismo | Autenticação | MCC |
| 10 | Shahraki, Rudolph e Grobler (2019) | Modelo | Cont.acesso | e-health |
| 11 | Kambou e Bouabdallah (2019) | Método | Autenticação | Strong Authentication |
| 12 | Banerjee et al. (2020) | Esquema | Cont.acesso | Baseado em CP-ABE |
| 13 | Fotouhi et al. (2020) | Esquema | Autenticação | Two-Factor + healthcare |
| 14 | Khalid et al. (2021) | Esquema | Autenticação | MFA cross-plataforma |
| 15 | Smet et al. (2021) | Esquema | Autenticação | Fog + PUF |
| 16 | Kalaria et al. (2021) | Abordagem | Autenticação | Autenticação mútua |
| 17 | Loffi et al. (2021) | Método | Autenticação | Autenticação mútua + MFA |
| 18 | Li et al. (2021) | Esquema | Autenticação | Smart living |

Tabela 8 – Trabalhos correlatos de segurança, autenticação e controle de acesso

do registro e o processo de autenticação não requer muitos recursos computacionais. Chaudhry et al. (2019) trazem uma melhoria no esquema de autenticação de Tsai e Lo para autenticação anônima.

A proposta de Fremantle e Aziz (2016) traz uma solução utilizando privacidade para usuários e dispositivos. Tal solução deve ser implementada em quatro partes: (i) uma arquitetura e um modelo de sistema; (ii) a definição clara do processo de registro de dispositivos e usuários; (iii) um modelo para o provimento de identidade anônima para usuários; e (iv) uma demonstração do funcionamento do OAuthing, nome dado à proposta. Tais requisitos são pressupostos para o OAuth2, que atualmente tem (USING. . . , 2019) *authorization code* de 256 *bytes*, *access token* de 2048 *bytes* e *refresh token* de 512 *bytes*. Dessa forma, o consumo energético e de banda podem ser impeditivos para dispositivos com as restrições anteriormente citadas.

Em Ali et al. (2018) é proposto um novo método para encriptar/desencriptar *templates* biométricos e a desencriptação com processamento dos dados biométricos na *cloud*. Essa é uma abordagem utilizada para computação de borda e leva em conta a validação de dois de três (2/3) dados biométricos fornecidos, tolerando um pequeno grau de incerteza. São extraídas e avaliadas duas características da voz e uma da face, o que proporciona ao mecanismo proposto uma maior maleabilidade no momento da decisão.

Ometov et al. (2018) trazem uma ferramenta de MFA para o ecossistema veículo-para-tudo(*vehicle-to-everything* - V2X), que utiliza diferentes fatores biométricos para autenticar o usuário no carro, e ainda apresenta diferentes aspectos da MFA, incluindo fatores cognitivos como elegíveis para autenticação. A ferramenta traz o conceito de fator de bloqueio, a exemplo do sensor que detecta quando o condutor consumiu álcool e bloqueia o veículo.

Anakath, Rajakumar e Ambika (2019) propõem um mecanismo de autenticação e trazem uma visão geral sobre os desafios da *Multi Factor Authentication* (MFA). Em sua proposta, eles utilizam a geração de um número pseudoaleatório que somente é válido em uma janela temporal. Essa geração é feita mediante a autenticação prévia de dois fatores: a utilização de OTP e uma senha.

Outro trabalho analisado é o de Esiner e Datta (2019), que traz a utilização de dois fatores na autenticação do armazenamento disperso de dados. Esse trabalho evita a utilização de um terceiro confiável na autenticação, de modo que não é possível que as partes que fazem o armazenamento e o terceiro confiável façam conluio para roubo de dados. A não necessidade de um terceiro confiável ocorre devido à utilização de *On Time Password* (OTP) e um segredo, dois fatores utilizados para a autenticação e acesso aos dados armazenados dispersamente.

Em Gomi et al. (2019), a proposta é de um novo mecanismo de autenticação baseado no contexto, que parte do conceito de que é possível confiar em dispositivos próximos. O contexto é estabelecido a partir do dispositivo primário e estende-se pelos dispositivos subordinados. Os autores também apresentam um modelo de ameaças e uma experimentação com uma implementação do modelo proposto.

Em Shahraki, Rudolph e Grobler (2019) é proposto um modelo de controle de acesso descentralizado e multi-autoridade baseado em atributos (DMC-ABAC) para segurança em saúde. Nesse modelo, vários atributos e chaves secretas são atribuídas aos usuários por diversas autoridades, e o acesso é decido a partir de políticas de acesso a atributos ou satisfatibilidade de atributos para acesso a dados.

Kambou e Bouabdallah (2019) apresentam uma pesquisa focada em *cloud services* para autenticação *web* e de *smartphones*. O trabalho citado traz uma abordagem de dois fatores baseada em uma autenticação na *cloud* e outra através de um OTP, a ser recebido em um dispositivo IoT. Tal trabalho utiliza um canal criptografado para transferência de dados sensíveis. Em se tratando de *hardware*, utiliza uma FIPY *board*[1] como dispositivo para implementação de sua prova de testes.

No trabalho Banerjee et al. (2020), os autores propõem uma ferramenta de controle de acesso de grão fino. Essa solução foca no acesso e uso de dados em um ambiente de IoT, e ainda utiliza *multi-authority attribute-based encription* para implementação do controle de acesso, utilizando três fatores para a autenticação proposta.

O trabalho de Fotouhi et al. (2020) tem um foco em autenticação de dispositivos médicos e visa permitir acesso aos dados dos sensores existentes no corpo dos pacientes, por parte dos médicos e do pessoal de atendimento. Tal proposta utiliza em conjunto uma estrutura de *hash* e XORs para realizar a autenticação das pessoas citadas. Seu esquema foi validado utilizando o ProVerif e a comunicação de rede foi simulada através do OPNET.

Em Kalaria et al. (2021), os autores propõem uma solução para autenticação mútua de sensores na *Fog Computing*. A abordagem proposta utiliza criptografia de curvas elípticas

[1] Dispositivo que tem capacidade de comunicação com cinco diferentes redes - https://pycom.io/product/fipy/.

(ECC) e funções *hash* para um menor custo computacional. Os autores também fornecem uma verificação formal através da ferramenta AVISPA. De forma geral, a solução proposta visa ser um protocolo leve e seguro de troca mútua de chaves para o ambiente *edge-fog-cloud*.

Já em Khalid et al. (2021), os autores trazem um esquema de autenticação mútua que estende o fluxo de trabalho do Kerberos e se propõe a ser multifator. Apesar de o trabalho tentar solucionar os problemas de autenticação *cross-plataform* de dispositivos da borda (*edge*), sua proposta de autenticação utiliza um fator biométrico. Ainda sobre os fatores utilizados, eles são usuário/senha, *smartcard* e impressão digital. Sua área de aplicação é a Industrial IoT (IIoT), ou seja, a Indústria 4.0.

Em Li et al. (2021), os autores propõem trazer a autoridade certificadora mais para proximidade da borda (*Edge Computing*), lugar onde os dados são produzidos. Tal abordagem visa diminuir a latência durante a autenticação e também a superfície de ataque disponível para adversários. Seu foco está na necessidade de autenticação massiva de dispositivos (grande quantidade de dispositivos), em específico para *Smart Living*.

O trabalho de Loffi et al. (2021) propõem uma melhoria na autenticação mútua por meio da incorporação de multifator durante o processo de autenticação. Como fatores, são propostos a utilização de um tempo ajustável de resposta, um fator do tipo *Challenge*/*Response* e um *nonce*. Tal proposta foca em autenticação de dispositivos IoT nas camadas de *Cloud* e *Fog Computing* e sua verificação formal foi realizada utilizando-se a ferramenta ProVerif.

Smet et al. (2021) trazem uma autenticação multifator leve que integra autenticação mútua e *Physical Unclonable Function* (PUF) como fatores de autenticação. Seu trabalho foca na autenticação automática de sensores na *Fog Computing*, não abordando a de usuários. Se faz importante citar que o trabalho tem como base algoritmos de criptografia assimétrica.

Diversos trabalhos apontam a necessidade de uma solução, esquema ou mecanismo de autenticação para dispositivos com menor poder computacional. Algumas propostas (TSAI; LO, 2015; GOMI et al., 2019; CHAUDHRY et al., 2019) trazem soluções para dispositivos móveis com pouco processamento; já outras (ALI et al., 2018) focam na importância de transferência de dados biométricos para autenticação segura; e também há algumas que têm foco em v2x (OMETOV et al., 2018). Entretanto, dois aspectos são consenso na maioria dos trabalhos citados: a utilização de um fator para autenticação traz insegurança ao sistema e dispositivos restritos não suportam algoritmos pesados de criptografia e segurança

## 4.2 O MECANISMO DE MULTI-FACTOR AUTHENTICATION COM *SCORE* DE REPUTAÇÃO (MFA_R)

### 4.2.1 Conceitos iniciais

Antes de detalharmos os componentes, é necessário uma complementação quanto ao funcionamento do mecanismo. Os aspectos que devem ser esclarecidos são: o modo compatibilidade, o conceito de reputação, valores esperados e a API de consulta. Esses conceitos não

estão aparentes na modelagem de componentes, pois estão distribuídos em diferentes componentes.

Sobre o **modo de compatibilidade**, esse mecanismo de autenticação visa garantir compatibilidade com dispositivos legados, e, para isso, permite o registro de dispositivos antigos com uma *flag* de "dispositivo em modo compatibilidade". Essa marcação permite que o *gateway* registre os dados do dispositivo sem um nível de reputação associado a eles. Como resultado, caso o consumidor dos dados os utilize, saberá que eles não são assegurados pelo mecanismo.

Quanto ao conceito de **reputação**, neste trabalho ele difere do conceito clássico de sistemas distribuídos. Reputação, dentro do contexto desse mecanismo, expressa um *score* de publicações não identificadas, como violações, ou seja, dentro do padrão esperado de publicações. Esse conceito é representado por um *score* de reputação associado ao dispositivo e sua sessão de uso do sistema, ou seja, ele é reiniciado a cada ciclo de nova autenticação do dispositivo. Ressalta-se que existem ainda limiares associados ao *score* de reputação que definem como o mecanismo de autenticação evolui ao longo de seu funcionamento e que serão discutidos mais a frente neste capítulo.

Esse **padrão de dados** pode variar de acordo com o domínio de problema (Vide Algoritmo 1). Por exemplo, vejamos três diferentes casos (expressos nas três condições das linhas 3,10 e 15): (i) temperatura em uma estufa; (ii) chuva em um talhão; e (iii) dados complexos em uma planta de produção. Em situações que envolvem temperatura, podemos utilizar limites, como no caso da temperatura em uma estufa (i) passar de 1000 graus Celsius, o que foge de um padrão normal. Outro caso é utilizar uma análise estatística, pois, se um pluviômetro (ii) indica 100mm de chuva em um talhão que outros indicam uma média de 3mm, ele pode estar comprometido. Ainda, como último caso (iii), podemos utilizar aprendizado de máquina e submeter as medições recebidas a seu crivo, a fim de identificar quais variáveis correlacionados estão fugindo do padrão que foi aprendido como normal. Observa-se que são diversas as formas de implementar a análise da validade dos dados e que elas variam de acordo com o escopo do problema.

**Algoritmo 1:** Avaliação do dado publicado quanto a sua validade (exemplo utilizando estatística, limites e *machine learning*).

```
Result: result
Function default_value(data: real) : boolean is
    result = FALSE;
    if type = "statistical" then
        mean = calculate_mean();
        max = calculate_max();
        min = calculate_min();
        std = calculate_std();
        result = data ≤ max and data ≥ min and |data − mean| ≤ std;
    end
    if type = "limits" then
        max = config("max");
        min = config("min");
        result = data ≤ max and data ≥ min;
    end
    if type = "machine learning" then
        generate_clusters();
        c = classify(data);
        result = is_valid(data);
    end
    return result;
end
```

Por último, temos a **API de consulta**, que é utilizada como fator do tipo "algo que se tem"durante a autenticação. Ainda que os dados de consulta sejam baseados no registro, eles são meramente um meio de assegurar que a mesma API está presente tanto no AuthInspector [2] quanto no *firmware* do sensor que está se comunicando, combatendo a possibilidade de ataques de *impersonation* [3]. Contudo, observa-se que a utilização dessa API como segundo fator pode ser substituída, dependendo da necessidade por *hardware* seguro (KULIK et al., 2022; COMAN et al., 2019), *tokens* (SAMMOUD et al., 2020; BHAWIYUGA; DATA; WARDA, 2017) ou mesmo *smart contracts* (BUCCAFURRI; ROMOLO, 2019).

[2] Componente melhor detalhado nas seções posteriores.

[3] Neste tipo de ataque o adversário pode produzir um processo para passar-se pelo sensor e assim publicar dados.

**Algoritmo 2:** Query API - query processor

```
   Result: query_response
1  Function process_query(a: String, device_data: List) : String is
2      a = split(query);
3      if a[0]==0 then
4          data = device_data['serial_number'];
5      else
6          if a[0]==1 then
7              data = device_data['capacity'];
8          else
9              other_data
10         end
11     end
12     s = slice(data,a[1],a[2]);
13     if a[3]==1 then
14         invert(s);
15     else
16     end
17     a[4]==1 complemento(s);
18     return s;
19 end
```

O algoritmo 2 traz um exemplo de processador de consultas, um método que recebe a consulta a ser processada junto aos dados do dispositivo e entrega uma resposta no formato *string* (linha 1). O método inicia quebrando a *string* de consulta em seus caracteres (linha 2) e, em seguida, analisa qual informação deverá ser processada (linhas 3 a 11), qual fragmento da informação deverá ser utilizado (linha 12) e se o resultado deverá sofrer alguma alteração em seu consulto de *bytes* (linhas 13 a 17). Por fim, retorna o resultado na linha 18.

Esses conceitos trazem uma complementação necessária para um melhor entendimento das próximas subseções: processos e componentes. Com eles em mente, é possível fazermos um melhor aproveitamento do conteúdo exposto nos algoritmos e compreender mais facilmente o desenvolvimento do processo de autenticação. Uma vez que esse processo se dá em várias etapas, tais conceitos podem ser revisitados em diferentes momentos da leitura do texto, para um melhor esclarecimento do seu conteúdo.

### 4.2.2 Processos do mecanismo

Nesta subseção são comentados os processos do MFA_R. Para tal, eles foram divididos em quatro partes: processo de registro, processo de autenticação, processo de *enforcement* e processo de publicação de dados. Uma vez que a autenticação se dá por meio de multifator,

utilizando reputação, um detalhamento maior desse processo será feito ao decorrer desta parte do texto.

De maneira geral, cada processo tem seu algoritmo ou conjunto de algoritmos comentado e detalhado. Contudo, cada algoritmo não se atém a detalhes de construção de uma linguagem de programação, mas serve como um guia para implementação em qualquer linguagem do paradigma estruturado ou seu subconjunto. Ainda que por meio de adaptações necessárias, é possível aplicar o MFA_R a qualquer paradigma de programação e a qualquer linguagem, pois o nível de abstração dos algoritmos dá suporte a isso. O conjunto de símbolos e termos utilizados para os algoritmos pode ser consultado na Tabela 9.

| Código | Descrição |
|---|---|
| $dev_{\text{reg}}$ | dispositivo registrado |
| $dev_{\text{unreg}}$ | dispositivo não registrado |
| $gw$ | *gateway*, dispositivo onde os serviços são publicados |
| $db_{\text{gw}}$ | banco de dados localizado no *gateway* |
| $regServ$ | serviço de registro |
| $authServ$ | serviço de autenticação |
| $authInspector$ | inspetor de autenticação |
| $dataServ$ | serviço de dados |
| $auth\_code$ | código de autenticação |
| $authetication\_query$ | consulta utilizada como segundo fator de autenticação |

Tabela 9 – Tabela de símbolos

#### *4.2.2.1 Processo de registro*

O registro é o primeiro passo para o uso da proposta e pode ser realizado no momento do desempacotamento para utilização do dispositivo, devendo ocorrer através de uma rede com maior capacidade de tráfego, ou mesmo através de um canal seguro (usando SSL), diminuindo a possibilidade de cópia dos dados de registro trafegados.

O registro é um processo que demanda transmissão de um número maior de dados, portanto, recomenda-se a não utilização de uma rede tarifada por pacotes ou com quotas de tráfego diário de dados (BERGMANN; ROBINSON, 2012; SIVANATHAN et al., 2016), o que pode aumentar expressivamente os custos ao se ativar múltiplos dispositivos em lote.

São demandados três componentes e dois dispositivos diferentes para o processo de registro. Os dois dispositivos são: o dispositivo a ser registrado ($dev_{\text{reg}}$) e o *gateway* ($gw$); e os três componentes são: o *firmware* cliente do dispositivo ($dev_{\text{reg}}$), o servidor de registro ($regServ$) e a base de dados do *gateway* ($db_{\text{gw}}$). Observa-se que não é mandatório que o serviço de registro esteja alocado no $gw$, entretanto, em nossa proposta utilizamos essa arquitetura com o intuito de reduzir os custos de implantação em uma situação real.

O processo inicia após a conexão no serviço de registro ($regServ$) por parte do cliente

não registrado ($dev_{\text{unreg}}$). Em seguida, o $dev_{\text{unreg}}$ envia uma mensagem[4] para o *regServ* do *gw* que, ao receber a mensagem, consulta seu banco de dados ($db_{\text{gw}}$) para verificar se o dispositivo já foi registrado previamente e, caso ele tenha sido, o registro é negado e uma mensagem com $-1$ é devolvida para o $dev_{\text{unreg}}$, o que se dá com o intuito de evitar a clonagem de dispositivos e ameaças relacionadas a esse aspecto.

A consulta é feita por meio da geração de um *hash* dos dados enviados que, se já estiver registrado no banco de dados, o dispositivo em questão já foi registrado. Essa solução não tem uma recomendação para situações de re-registro de dispositivos, sendo esse um procedimento a ser resolvido para cada situação no futuro. Caso o $dev_{\text{unreg}}$ não tenha sido registrado previamente, é gerada uma senha aleatória e um usuário baseado no seu *hash*. Os dados encapsulados em um objeto de registro, junto àqueles vindos do dispositivo (4.1), são enviados para o banco de dados ($db_{\text{gw}}$).

$$obj\_reg < user, passwd, reg\_data > \tag{4.1}$$

Ao final, são retornados para o $dev_{\text{unreg}}$ o *timestamp*, o usuário e a senha. Quanto ao formato, o *timestamp* utiliza sua codificação numérica (número do tipo inteiro) e não pressupõe sincronização entre as partes, ou seja, o *timestamp* gravado é o gerado no serviço de registro. Esses dados são encapsulados em uma mensagem de confirmação de registro (4.2) e, a partir desse momento, o dispositivo é considerado registrado ($dev_{\text{reg}}$).

$$reg\_confirm < timestamp, user, passwd > \tag{4.2}$$

Esse é um processo que deve acontecer no máximo uma vez. Para dispositivos no modo compatibilidade, o registro deverá ser feito por meio de uma *interface* de usuário. O serviço que fornecerá a capacidade de registro é o mesmo, contudo, uma camada *web* deve ser construída para permitir que o administrador do sistema IoT possa registrar os dispositivos legados participantes. Dessa forma, para o serviço a diferença não deve ser notada e, caso seja feito o registro por uma camada *web* ou uma API RESTful, isso deve ser transparente para o serviço.

Para uma melhor entendimento, são apresentados aqui os dois algoritmos de registro (3 e 4), respectivamente do *client-side* (do *firmware*) e do *server-side* (servidor). Por meio desse pseudocódigo é possível notar alguns detalhes de implementação que são mais facilmente apresentados na forma de código (i.e., *callback* das funções).

[4] Existem diversos formatos utilizados para troca de mensagens, e em nosso estudo utilizamos JSON, devido ao baixo *overhead* gerado.

**Algoritmo 3:** Processo de registro de um dispositivo - *client-side*.

```
   Result: result
1  Function register(registration_data: Device) : boolean is
2      result = FALSE;
3      connect_tls(regServ);
4      send_data(registration_data);
5      wait(callback) receive(reg_confirm);
6      if reg_confirm = -1 then
7          exit (1);
8      end
9      save_data(reg_confirm);
10     result = TRUE;
11     return result;
12 end
```

No primeiro algoritmo apresentado (3), o registro do *client-side* traz um fluxo simples desse processo. Contudo, é importante ressaltar alguns detalhes no código, como a conexão segura (linha 3) e o salvamento dos dados do registro ao final do processo (linha 9). Esses são detalhes que reforçam a necessidade de utilização de uma canal seguro no momento do registro e também a necessidade de salvar dados de confirmação do registro (seção 4.2) do lado do cliente também.

**Algoritmo 4:** Processo de registro de um dispositivo - *server-side*.

```
   Result: result
1  Function register_device() : boolean is
2      result = FALSE;
3      receive(device_data);
4      hash_device = hash(device_data);
5      if exists(hash_device) then
6          return -1;
7      end
8      timestamp = generate_timestamp();
9      user = generate_user();
10     passwd = generate_password();
11     db.save_device(hash_device,device_data,timestamp,user,passwd);
12     return (timestamp, user, passwd);
13 end
```

Já o algoritmo (4) apresenta o processo do ponto de vista do servidor (*gw*), explicitando os processos de verificação da existência de um registro prévio (já registrado - linha 5), a geração dos dados do registro (linha 8 a 10) e o salvamento dos dados em uma base de dados do lado servidor (linha 11). Dessa forma, foi possível formalizar os passos necessários no servidor para

o processo de registro.

O processo de registro é a primeira etapa de toda a autenticação, onde os dados entre cliente e servidor são trocados, sendo de suma importância o entendimento completo desse processo, assim como das funções e dos componentes demandados (i.e., funções *hash* e estrutura de dados) para um bom funcionamento. Com esse entendimento poderemos perceber melhor os detalhes dos demais processos de autenticação (4.2.2.2), de *enforcement* (4.2.2.3) e de publicação de dados (4.2.2.4).

#### *4.2.2.2 Processo de autenticação*

A autenticação se inicia com um dispositivo registrado ($dev_{\text{reg}}$) enviando dados de autenticação ao serviço relacionado (*authServ*), podendo acontecer três fluxos distintos a partir desse momento: (i) autenticação falha; (ii) autenticação com sucesso - modo compatibilidade; e (iii) autenticação com sucesso - modo reputação.

Quando ocorrer uma falha na autenticação (i), o *authServ* deverá armazenar um registro de auditoria [5]. Na autenticação com sucesso em modo de compatibilidade (ii), o *authServ* retorna uma mensagem confirmando a autenticação e também se comunica com o inspetor de autenticação (*authInspector*), registrando o *auth_code* [6] do dispositivo como modo compatibilidade.

Em seguida o *authServ* gera (2) uma *authetication_query* para o $dev_{\text{reg}}$ a partir dos dados trocados no momento do registro e, ao recebê-la, o $dev_{\text{reg}}$ deverá gerar uma resposta para a *query* e enviá-la ao *authServ*, que a comparará com o resultado esperado, previamente calculado. Caso os valores sejam iguais, o *authInspector* é informado de que o $dev_{\text{reg}}$ subiu ao nível AUTH2. É importante destacar que ao subir ao AUTH2, o dispositivo torna-se elegível para o AUTH3, de acordo com seu comportamento no envio de dados. No caso de resultado negativo durante a autenticação, é retornado $-1$ como código de erro [7].

Durante a publicação de dados, dá-se a validação do **terceiro fator**. Após atingir o nível AUTH2, o sensor recebe um *score* de reputação. Ao passo que as publicações são feitas e os dados permanecem dentro do desvio padrão, o $dev_{\text{reg}}$ ganha mais reputação e, quando ele atinge o limiar superior (4.3) necessário, torna-se elegível para o nível AUTH3.

$$limiar\_superior \leq score - salt \tag{4.3}$$

Existe também a possibilidade de um dispositivo descer do nível AUTH3 para o nível AUTH2, o que ocorrerá caso o sensor apresente falha e envie medições anômalas, levando-o a

[5] Este registro deve ser em formato textual para ser processado por um IDS, que trabalha em conjunto com o mecanismo de autenticação para evitar DDoS, entre outros ataques.

[6] O código de autenticação é armazenado no componente de *software authInspector*, e este código funciona como a credencial do dispositivo autenticado.

[7] Observamos que o valor $-1$ como padrão de erro de autenticação evita a identificação do nível de acesso que o dispositivo tem, pois, uma vez que os erros são iguais para os diferentes níveis, alguém que esteja escutando a rede não será capaz de obter a informação do nível de acesso baseado na captura dos retornos falhos.

perder reputação. Se isso for recorrente, o sensor pode atingir o limiar inferior (4.4) e cair para o nível AUTH2. Detalhadamente, o limiar inferior é sempre comparado com o score atual do dispositivo em questão, somado a um *salt*, que é um valor ajustável por meio de configuração, de modo a permitir adequações à forma de uso do limite. Tais modificações de nível devem gerar *audit log*.

$$limiar\_inferior \geq score + salt \tag{4.4}$$

Desse modo, são contemplados os aspectos gerais do processo de autenticação em suas três etapas. Ressalta-se que, durante o processo de autenticação de dispositivos, foram trazidos os limiares e seu funcionamento junto ao score de reputação. Para um maior esclarecimento e formalização do processo, a seguir apresentamos três algoritmos para esclarecimento do processo de autenticação: autenticação do lado cliente (5); primeiro passo da autenticação do lado servidor (6); e segundo passo da autenticação do lado servidor (7).

O algoritmo que descreve o comportamento do dispositivo durante a autenticação (5) tem aspectos importantes que devem ser observados. Alguns desses aspectos são as funções de *callback*, que devem ser tratadas (linha 4 e 9), e a utilização da Query API para resolução do segundo fator de autenticação (linha 7). Deve-se observar que o algoritmo de resolução da *query* já foi apresentado neste capítulo (seção 4.2.1).

**Algoritmo 5:** Processo de Autenticação de um Dispositivo - *client-side*.

```
   Result: result
1  Function authenticate() : boolean is
2      result = FALSE;
3      resp_step1 = send_step1(login,senha);
4      wait(callback_step1);
5      query = resp_step1.query;
6      auth_code = resp_step1.auth_code;
7      step2 = query_api.resolve(device_data,query);
8      resp_step2 = send_step2(auth_code,step2);
9      wait(callback_step2);
10     result = resp_step2;
11     return result;
12 end
```

O segundo algoritmo (6) traz a autenticação baseada em segredo (*password-based*), garantindo a compatibilidade com dispositivos legados. Como se pode notar, após validar o *login* do dispositivo (linha 6), o servidor verifica se ele está em modo compatibilidade (linha 8). Caso seja um dispositivo legado, o dispositivo estará registrado no *authInspector* como modo compatibilidade (COMPATIBILITY_MODE). Caso não seja legado, um fator Challenge-Response será gerado por meio da Query API (linha 12), salvo no *authInspector* junto ao código de acesso do dispositivo (linha 14), onde também é registrado que o dispositivo está no

primeiro nível de autenticação (linha 15). Ao final, a *query* é enviada para que o dispositivo a resolva (linha 16).

**Algoritmo 6:** Processo de Autenticação de um Dispositivo (STEP1) - *server-side*.

```
   Result: result
 1 Function do_step1(request: Request) : boolean is
 2     result = FALSE;
 3     step1 = receive_step1();
 4     login = step1.login;
 5     senha = step1.senha;
 6     if valid(login,senha) then
 7         device_data = retrieve (login);
 8         if device_data.legacy is TRUE then
 9             auth_code = login;
10             authInspector.level(auth_code,COMPATIBILITY_MODE);
11         else
12             query,resp = query_api.generate(device_data);
13             auth_code = authInspector.generate_code();
14             authInspector.save(auth_code,resp);
15             authInspector.level(auth_code,AUTH1);
16             resp_step2(query);
17         end
18         result = TRUE;
19     end
20     return result;
21 end
```

Por último, tem-se o algoritmo de resolução do segundo fator de autenticação (7), que é analisado através de um único algoritmo, o *server-side*, que traz os passos do servidor na validação do fator Challenge-Response (CR). Após receber a resposta do dispositivo e decodificá-la (linha 3 a 5), o *authInspector* é consultado para validação da resposta através da utilização do *auth_code* como chave de busca. Uma vez validada a chave, é gerado um registro de que o dispositivo está autenticado no segundo fator (linha 7), sendo a ele atribuído um *score* de reputação inicial (linha 8). Note-se que a interação entre o *authServ* e o *authInspector* é primordial para o sucesso do processo de autenticação e de manutenção do *score* de reputação, como evidenciaram os algoritmos anteriores.

**Algoritmo 7:** Processo de Autenticação de um Dispositivo (STEP2) - *server-side*.

```
Result: result
1  Function do_step2(request: Request) : boolean is
2      result = FALSE;
3      step2 = receive_step2();
4      auth_code = step2.auth_code;
5      resp = step2.query_response;
6      if authInspector.valid(auth_code,resp) then
7          authInspector.level(auth_code,AUTH2);
8          authInspector.reputation_score(auth_code,INITIAL_REPUTATION);
9          result = TRUE;
10     end
11     return result;
12 end
```

#### *4.2.2.3 Processo de enforcement*

Ainda que a autenticação ocorra com sucesso, faz-se necessário um procedimento para reforçar a segurança do sistema quanto à identificação de dispositivos. Esse processo é realizado após um número específico de mensagens, ou um tempo específico de autenticação do segundo passo, ou seja, quando deve haver um reforço da autenticação (*enforcement*). Tal passo tem a finalidade de mitigar ataques, tais como *spoofing*[8] e *sybil*[9], entre outros. Uma vez que a consulta do segundo passo será refeita, é praticamente impossível um dispositivo (real ou virtual) malicioso "advinhar"a resposta (fator Challenge-Response) e continuar participando do sistema autenticado.

[8] Passar-se por outro sujeito, componente de *software* ou dispositivo; imitando suas características físicas (SCHUCKERS, 2002; WU et al., 2015) ou de rede (BREMLER-BARR; LEVY, 2005; WHALEN, 2001).

[9] É um ataque onde um nó finge ser vários nodos (DOUCEUR, 2002). Isto é particularmente problemático em redes de sensores (NEWSOME et al., 2004).

**Algoritmo 8:** Processo de *Enforcement - server-side*.

```
   Result: result
1  Function do_enforcement(request: Request) : boolean is
2      result = FALSE;
3      if score < LIMIAR_INFERIOR then
4          query,resp = query_api.generate(device_data);
5          auth_code = authInspector.generate_code();
6          authInspector.save(auth_code,resp);
7          authInspector.level(auth_code,AUTH2);
8          resp_step2(query);
9          wait(step2);
10         step2 = receive_step2();
11         auth_code = step2.auth_code;
12         resp = step2.query_response;
13         if authInspector.valid(auth_code,resp) then
14             authInspector.level(auth_code,AUTH2);
15             authInspector.reputation_score(auth_code,INITIAL_REPUTATION);
16             result = TRUE;
17         else
18             authInspector.remove(auth_code);
19         end
20     end
21     return result;
22 end
```

O processo de *enforcement* inicia-se quando o serviço de dados (*dataServ*) recebe os dados do dispositivo autenticado. O *dataServ* consulta o *authServ* para verificar se existe a necessidade de *enforcement*. Caso haja, uma nova consulta é enviada de "carona"(*pigback*) no retorno da publicação de dados. Caso não haja necessidade de *enforcement*, o campo de consulta retornará vazio. Uma vez recebida a consulta, o $dev_{\text{reg}}$ deverá respondê-la corretamente, ou terá sua autenticação expirada pelo *authInspector* e descerá ao nível de compatibilidade.

Adicionalmente, é importante citar alguns aspectos do algoritmo de *enforcement* (8). Ele é um algoritmo *server-side* que tem por função revalidar o acesso de um dispositivo que potencialmente cometeu faltas, por exemplo. Nota-se que esse algoritmo inicia quando o *score* de reputação está abaixo do *LIMIAR_INFERIOR* (equação 4.4), linha 3. De modo similar ao processo de autenticação, uma *query* é gerada e enviada ao dispositivo (linhas 4 a 8), que, quando recebida de volta pelo servidor (linha 11), passa novamente por uma validação de sua resposta. Em caso positivo, o dispositivo é reinserido como autenticado em dois fatores e com uma reputação inicial (linhas 14 a 16). Porém, no caso de a resposta a *query* não ser a correta, o *authInspector* removerá o acesso do dispositivo ao sistema (linha 19).

#### 4.2.2.4 Publicação de dados

$$data\_package < data, auth\_code > \tag{4.5}$$

Durante a publicação de dados, os $dev_{\text{reg}}$ enviam pacotes de dados (4.5) ao *dataServ* que, ao recebê-los, avalia o *auth_code* junto ao *authInspector* e verifica se é necessário fazer o *enforcement*. Caso não haja *enforcement*, existem quatro caminhos possíveis: (i) quando o nível de autenticação AUTH3 é ativado e o valor está dentro do desvio padrão; (ii) quando o nível de autenticação é AUTH3 e o valor é considerado inválido; (iii) quando o nível de autenticação é AUTH2 e está dentro do desvio padrão; e (iv) quando o nível é AUTH2 e está fora do desvio padrão.

Na situação (i), o valor é coletado e passa a compor a média do grupo monitorado. O valor da reputação é incrementado e uma mensagem de "OK", que é enviada ao $dev_{\text{reg}}$. No caso (ii), o valor é coletado e utilizado para o cálculo da média e do desvio padrão. O valor da reputação é decrementado e atualizado. Caso o $dev_{\text{reg}}$ seja rebaixado para o nível AUTH2, a rotina de *enforcement* é executada para esse dispositivo. Os casos (iii) e (iv) já foram abordados anteriormente neste texto.

Sobre perda de reputação, observa-se que, caso uma anomalia aconteça em vários sensores, ela não afetará o sistema de reputações, que se baseia nos dados estatísticos e, caso uma catástrofe ocorra, todos os sensores do grupo sofrerão variação em conjunto. O mecanismo de reputação foi projetado para identificar sensores com comportamento anômalo e o envio de dados falsos/alterados/forjados.

### 4.2.3 Partes do mecanismo

Como citado na seção anterior, o *server-side* é composto por quatro partes principais: RegServ, AuthServ, DataServ e AuthInspector. Nesta subseção comentaremos suas responsabilidades, atribuições e demais detalhes relevantes. Diferentemente do texto da seção anterior, aqui não apresentaremos a visão dinâmica do processo, mas uma visão estrutural de responsabilidades e de relações entre as partes do mecanismo proposto —iniciaremos pelo RegServ.

O **RegServ** é responsável pelo registro dos dispositivos no mecanismo. O registro precisa ser executado somente uma vez e pode ser feito preferencialmente antes de o equipamento ir a campo, utilizando uma rede mais robusta e por meio de uma *interface* segura (como HTTPS[10]) - o que diminui a quantidade de dados trafegados pelo *link* de dados LPWAN[11].

Ao se registrar um dispositivo, envia-se seus dados para o *gateway*, que o registra no artefato de dados dos dispositivos (*device_data*). Após o registro, são gerados *login* e *password*

[10] HTTP over SSL/TLS.

[11] É importante salientar que, caso o dispositivo esteja configurado como modo compatibilidade, seu *login* e *password* não são gerados, mas informados no momento do registro. O registro de dispositivos legados acontece por meio de uma *interface* diferente das demais.

para o dispositivo, que são enviados na resposta de confirmação do registro para o dispositivo. Também na resposta de confirmação é enviado o *timestamp* do registro e as informações citadas e dos dados do registro que serão utilizados como fatores na autenticação do dispositivo posteriormente.

Esse serviço se comunica somente com o dispositivo cliente que solicita o registro. Contudo, internamente ele faz adições na base de dados que está no *gateway* ($db_{gw}$). Tal base de dados é posteriormente utilizada pelos demais serviços do *server-side*.

Já o **AuthServ** é o componente responsável pela autenticação dos dispositivos, *queries* e pelo *enforcement*. Esse componente tem a responsabilidade de autenticar o primeiro fator e identificar se o dispositivo está no modo compatibilidade[12]. Os dados dos dispositivos em modo compatibilidade estarão disponíveis para consumo, entretanto, todos os dispositivos no modo compatibilidade têm seus dados publicados com reputação em zero. Isso acontece devido ao fato de o mecanismo identificá-los com grande potencial de falsificação de dados. O consumo desses dados pode ser feito por um sistema que não avalie como problemático esse fato, ou ainda, em um momento de transição na incorporação de novos sensores.

Caso o dispositivo não esteja no modo compatibilidade, o AuthServ gerará uma *query* (CR) utilizando a API comum (Query API) a ele e ao dispositivo, e a enviará como prova do segundo fator. O dispositivo solicitante, que tem a API do mecanismo implementada, resolverá a *query* enviada e conseguirá autenticar-se com o segundo fator, passando para o nível AUTH2. Nesse momento, o AuthServ informa o AuthInspector que o dispositivo está autenticado em AUTH2, atribuindo-lhe um *score* (reputação).

Ainda como responsabilidade do AuthServ, apresentamos o *enforcement*. Esse processo acontece quando um dispositivo apresenta discrepâncias nos dados publicados ou já está autenticado há muito tempo (determinado por meio de configuração). O *enforcement* solicita que o dispositivo execute novamente o segundo passo de autenticação, ou seja, responda novamente à *query* gerada e enviada pelo AuthServ. Após esse processo ser concluído com sucesso, o dispositivo estará habilitado novamente para envio de dados. Se o dispositivo falhar no *enforcement*, ele será removido do sistema e somente voltará a ele após refazer todo o fluxo de autenticação e ganhar um novo *auth_code*.

O **DataServ** é responsável pela publicação e consumo dos dados. Nosso mecanismo foca na qualidade dos dados e na reputação dos sensores, em decorrência disso, a autenticação foi implementada e proposta para garantia da segurança na autenticação para publicação de dados. Para publicar os dados, o dispositivo deverá estar autenticado com dois fatores, ou no modo compatibilidade.

Quando o serviço recebe dados de um dispositivo, ele consulta junto ao AuthInspector se o código de autenticação enviado é válido (*auth_code*). O AuthInspector pode responder com três valores distintos: *OK* (dados podem ser recebidos), *enforcement* (o acesso precisa ser renovado) e *auth_code* inválido. Nesse último caso, será gerado um registro de tentativa de

[12] Ou seja, se é um dispositivo legado que continua em uso.

publicação indevida de dados, um *log* de segurança.

Continuando com o serviço de dados, ele é responsável por avaliar se os dados publicados estão dentro dos valores válidos e esperados. Uma vez identificado um padrão aleatório ou inválido na recepção de dados, o dispositivo é penalizado com a diminuição de reputação para o dispositivo que enviou os dados. Caso o dispositivo atinja um nível mínimo de reputação, o serviço de dados solicita um *enforcement* ao AuthServ. Similarmente, quanto mais valores dentro do padrão esperado, mais aumentará a reputação no sistema. Ressalta-se que o DataServ avaliará o padrão dos dados e incrementará a reputação do dispositivo até o limite superior configurado e, ao ultrapassá-lo, o dispositivo ascenderá ao nível AUTH3 —passando a estar autenticado com o **terceiro fator**. Caso o dispositivo em AUTH3 se comporte de maneira inadequada (publicação de dados inválidos —padrão de dados 4.2.1), ele sofrerá penalidades na reputação e, eventualmente, cairá para o nível AUTH2, após passar por um *enforcement* sem sucesso.

O fator **parte-do-todo** não se baseia em algo intercambiado com o dispositivo, mas em seu funcionamento como parte do todo, como parte de um sistema. Propôs-se esse fator inovador de autenticação para identificação de uma real parte do todo, real parte do sistema. Os fatores nesta proposta identificam o conhecimento de um segredo, a posse de um artefato (artefato de *software* - Query API) e o comportamento adequado para uma parte real do arranjo onde esse se encontra inserido, podendo esse ser classificado como um fator comportamental, que é um dos fatores de inovação deste trabalho. Seu uso em conjunto com o *score* de reputação são a chave para a inovação do MFA_R e podem dar suporte a outros aspectos de segurança, como modelos de autorização baseados em risco (CALVO; BELTRÁN, 2022; PRISCILA et al., 2022; AMEER et al., 2022), modelos de autorização adaptativos (INSHI et al., 2023; IBRAHIM; MOHAMED; HASSAN, 2022; CAO et al., 2022), *zero trust authentication* (LIU et al., 2022; MENG et al., 2022; SYED et al., 2022), ou mesmo autenticação contínua (MENG et al., 2022; SAHU; SHARMA; RAJA, 2022).

Como última parte do *gateway*, tem-se o AuthInspector. Mesmo que ele já tenha sido abordado neste documento, convêm listar as suas responsabilidades: (i) manter o registro de todos os dispositivos autenticados e seus níveis de autenticação (COMPATIBILIDADE, AUTH1, AUTH2 e AUTH3); (ii) manter registro da reputação dos sensores; (iii) gerar *logs* de auditoria de tentativas de acesso indevido; e iv) executar *enforcements*. Esse é um componente acessado pelos demais no *server-side*, não apresenta uma *interface* externa para acesso ou consulta, foi implementado sem acoplamento, ou seja, não depende dos demais componentes, e a dependência se dá somente no sentido dos serviços para o AuthInspector.

Todos os componentes apresentados cooperam para o funcionamento do mecanismo de autenticação, porém alguns componentes serão acessados somente uma vez durante o ciclo de vida do dispositivo, como no caso do RegServ. Já outros terão um acesso recorrente a cada publicação de dados e interagirão entre si. O bom funcionamento do mecanismo de autenticação está nas responsabilidades divididas entre os três serviços listados anteriormente e o AuthInspector. Algumas descrições de processos feitas nesta seção são complementadas pelas

máquinas de estados apresentadas posteriormente neste estudo.

# 5 MODELAGEM DA PROPOSTA

Nesta seção serão apresentados os artefatos que deram suporte à implementação da solução proposta. Com a utilização de tais artefatos é mostrada uma visão mais procedimental e incremental no entendimento do processo de construção da proposta e de sua implementação. A organização deste capítulo é composta por:

- Visão geral 5.1;
- Premissas e requisitos 5.2;
- Escopo 5.3;
- Diagramas de sequência 5.4;
- Diagramas de componentes 5.5;
- Diagramas de estado 5.6;
- Resultados da modelagem 5.7.

Esse conjunto de artefatos faz parte de diferentes etapas da construção e implementação da solução apresentada nos próximos capítulos deste trabalho. Como exemplo, o diagrama de estados serviu de suporte para criação dos autômatos temporizados, utilizados para verificação formal (*safety*). Já os diagramas de componentes foram importantes na criação do modelo de ameaças, utilizado na modelagem dos componentes e na análise de segurança. Portanto, este é um capítulo que mostra detalhadamente as decisões tomadas durante o projeto e é também um suporte de projeto aos capítulos posteriores.

Por meio dessa modelagem geral da solução, é possível argumentar sobre alguns aspectos formais de desenvolvimento e de desempenho que afetaram diretamente o sucesso da proposta. Uma vez que este trabalho intenta ser uma solução peso-leve para a autenticação de dispositivos restritos em um ambiente com baixa vazão de dados, se faz necessário atenção a detalhes arquiteturais e comportamentais do projeto. Desse modo, tais detalhes foram modelados utilizando-se a linguagem unifica de modelagem (UML2), comum a desenvolvedores, projetistas e pesquisadores da área de Ciências da Computação.

## 5.1 VISÃO GERAL

Conforme demonstrado anteriormente (Capítulo 4), existe uma necessidade de autenticação mais leve para dispositivos restritos em LPWAN e, ainda que existam diferentes propostas e implementações, nenhuma tem o mesmo foco desta aqui apresentada. Assim posto, ressaltamos que a nossa proposta apresenta diferentes aspectos de projeto que caracterizam requisitos e funções inovadoras, tais como: um suporte a *score* de reputação e compatibilidade com sistemas legados. Em sua maioria, as propostas adéquam soluções robustas de outras áreas à área de IoT, soluções que não levaram em conta todas as restrições envolvidas na criação das propostas.

Em vista disso, faz-se necessário a criação de um mecanismo de autenticação que fuja dos problemas das soluções atuais, a saber: *overhead* no protocolo de troca de credenciais, grande quantidade de mensagens trocadas e muitas partes envolvidas (YI et al., 2015; ROMAN; LOPEZ; MAMBO, 2018). Tal solução deverá ser adequada para utilização em ambiente com dispositivos restritos quanto a recursos, como energia, processamento, armazenamento e largura de banda (DEEP; ZHENG; HAMEY, 2019a; YI et al., 2015; VAQUERO; RODERO-MERINO, 2014; CAO et al., 2019; DEEP; ZHENG; HAMEY, 2019b; KHAN; SALAH, 2018) .

Devendo também tal solução fazer uso de um mecanismo de autenticação mais robusto do que somente *password-based*, mas também não oneroso a ponto de consumir muitos recursos computacionais. Se analisarmos a literatura que apresenta mecanismos de autenticação, existem muitos deles que podem ser adotados, tais como: autenticação baseada em *PIN*, *password-based*, SMS, chave simétrica, biometria, assinaturas digitais, *zero knowledge proof*, autenticação mútua, PKI, dois-fatores (2FA) e multifator (MFA) (ANAKATH; RAJAKUMAR; AMBIKA, 2019). Dentro do portfólio e dos trabalhos correlatos, a utilização de MFA tem se mostrado um importante campo de pesquisa, com trabalhos que levam em conta fatores do contexto ou cognitivos. Cabe ressaltar que a flexibilidade da utilização de fatores é um ponto-chave para o sucesso em nosso estudo.

Neste trabalho adotaremos a MFA junto com um sistema de reputação para os sensores. Tal proposta tem a pretensão de garantir a compatibilidade com sistemas legados que autenticam com somente um fator, e também oferecer uma reputação específica para tal tipo de sensor. A adoção desse mecanismo de autenticação nos permite prevenir diversas ameaças de segurança e fornecer a possibilidade de um *ranking* de reputação para os sensores. Esse fato permite ao ator que consome os dados decidir com base nas medições e na classificação do sensor, de acordo com o nível de autenticação atingido.

## 5.2 PREMISSAS E REQUISITOS

As premissas foram os pilares da fase de *inception* no processo de construção da solução proposta. Com as premissas definidas, pode-se derivar e desenvolver um conjunto de artefatos que suporte a solução; e manter a unicidade da proposta preliminar. Dessa maneira, as premissas servem como guias para o desenvolvimento da proposta e foram obtidas a partir do estado-da-arte da área de autenticação para IoT em LPWAN. Desse modo, nesta seção foi estabelecido um conjunto de premissas que norteou o desenvolvimento deste trabalho, servindo como de base para a medição de sua completude e sucesso.

Os critérios foram estabelecidos com base na revisão sistemática de literatura anteriormente aqui exposta, sendo a intersecção das áreas de autenticação, IoT e LPWAN. Essas áreas são muito vastas e atualmente possuem diversos desafios (vide seção 3.5), sendo necessário estabelecer quais características seriam abordadas neste trabalho. Algumas características são essenciais para o sistema, em decorrência das limitações dos dispositivos para os quais esse mecanismo foi modelado, assim como para o meio de transmissão e suas restrições. As premissas

para proposta estão dispostas na Tabela 10:

| # | Descrição |
|---|---|
| P01 | autenticar por multifator |
| P02 | ter o *overhead* de mensagens reduzido |
| P03 | ter um número limitado de mensagens |
| P04 | evitar problemas de *tampering* |
| P05 | mitigar *man-in-the-middle* (MITM) |
| P06 | mitigar clonagem de equipamento |
| P07 | evitar injeção de dados aleatórios |
| P08 | suportar compatibilidade com sistemas legados |
| P09 | suportar um sistema de reputação para sensores |

Tabela 10 – Premissas da proposta do autor

Detalhando-se um pouco mais as premissas trazidas, a autenticação por multifator (**P01**) é um padrão que faz parte das *guidelines* da NIST para identidade digital (GRASSI; GARCIA; FENTON, 2017), permitindo ao sistema que a utiliza fornecer uma autenticação mais robusta e em múltiplos passos. Ainda que a adoção de autenticação multifator para dispositivos seja um processo mais complexo do que a autenticação de usuários, é possível adotar esse tipo de abordagem com sucesso em diversos contextos de uso. Mesmo que alguns fatores sejam características inerentes a sujeitos humanos (i.e. biometria digital), alguns trabalhos demonstram que mesmo fatores do tipo "algo que se é"tem seu equivalente para dispositivos IoT.

Seguindo com a necessidade de um *overhead* reduzido de comunicação (**P02**) e a necessidade de um número limitado de mensagens (**P03**), essas premissas estão associadas à economia de bateria e à limitação de quantidade de dados enviados. Ao iniciar um envio de dados, o dispositivo deve ativar seus componentes de rede consumindo, assim, mais bateria. Adicionalmente, em alguns protocolos LPWAN, existem limitações de quantidade de dados trafegados diariamente ou um *payload* muito restrito em tamanho, portanto, quanto menor o número de mensagens trocadas na autenticação, melhor, pois, desse modo, um maior número de mensagens enviadas pode ter o foco maior na transmissão de dados.

Já quanto a evitar *tampering* (**P04**) e mitigar MITM (**P05**), o mecanismo deverá prover meios de detectar uma alteração nos dados dos sensores, o reenvio de mensagens e demais ameaças associadas. Uma vez identificado que a identidade do dispositivo está comprometida, o mecanismo deverá ser capaz de remover a credencial dele do sistema, assim ele será capaz de evitar que tais ameaças se consolidem.

Para a mitigação de clonagem de equipamento (**P06**) e a mitigação de injeção de dados aleatórios (**P07**), devem ser implementados algoritmos que suportem a identificação de tais ameaças. Ao identificar-se que um equipamento foi clonado, seja por uma dupla autenticação ou por um comportamento anômalo durante a publicação de dados, o dispositivo deverá ter sua credencial removida do sistema. Já para a identificação de injeção de dados aleatórios, o mecanismo deve ser capaz de identificar que os dados publicados estão divergindo de um padrão

esperado. Tal identificação deverá levar em conta os dados já publicados pelo mesmo sensor e pelos demais sensores dentro de seu escopo de atuação.

Por último, temos o suporte para compatibilidade com sistemas legados (**P08**) e o suporte a um sistema de *score* de reputação para os sensores (**P09**). O primeiro permite que se utilize os sensores existentes na planta de atuação do sistema, sem a necessidade de atualização imediata de *firmware* ou troca por sensores mais atuais. O segundo, permite a aplicabilidade desse mecanismo de autenticação a um sistema de *ranking* para avaliação e consumo de dados, como Crowd-IoT (OGU et al., 2022; FENG et al., 2022; DUAN et al., 2022); ou a posterior utilização desse *score* em outro processo de segurança, a saber, um modelo de autorização adaptativo baseado nesse *score*.

Esse conjunto de premissas e requisitos expressa a necessidade encontrada na literatura do estado-da-arte (Capítulo 3), necessidade essa associada a restrições de processamento, bateria e transmissão de dados existentes dentro do contexto de atuação explorado neste trabalho —dispositivos restritos e LPWAN. Dessa forma, um mecanismo projetado para atender a esse tipo de cenário deverá respeitar as premissas aqui impostas e atender aos requisitos necessários listados nesta seção.

## 5.3 ESCOPO

O escopo, por sua vez, nivela as expectativas e define limitações de até onde o projeto irá. Uma vez que a autenticação de dispositivos IoT é uma área vasta e muito rica, é impossível para um único trabalho abordá-la em sua plenitude, ou mesmo conseguir abarcar uma sub-área sua de maneira completa. Portanto, nesta seção são apresentadas as fronteiras deste trabalho, estabelecendo limites e adequando expectativas para que este trabalho pontual seja mais facilmente mensurável quanto a sua completude e qualidade.

Acredita-se que, por meio da adoção de tais limitações, é possível o projeto e a implementação de um mecanismo de autenticação para dispositivos restritos no contexto de monitoramento de grandes áreas por meio de IoT. Tal solução é proposta para o monitoramento ambiental, de florestas e da agricultura, mas também pode ser é aplicável a qualquer ambiente que tenha as seguintes características:

i. alimentação energética baseada em baterias (*ultra long life battery*);
ii. *link* de rádio com baixa taxa de transmissão (LPWAN);
iii. os dados coletados serem medições (*smart metering*);
iv. dados não críticos (*non-critial mission*);
v. sensores com *hardware* de baixo custo de implementação (*constrained devices*).

Dispositivos que utilizam baterias de longa vida (**i**) são um desafio para a adoção de processos mais sofisticados de autenticação, como o *Multi Factor Authentication*. Qualquer utilização extra de energia, que vá além de seu propósito principal de medição, pode comprometer

o tempo de vida do sensor em campo. Cada novo processo adotado deve ser o mais leve possível, em prol de fazer coleta de dados —missão principal do sensor. Devido a essa limitação e ao foco de consumo energético, a segurança em alguns dispositivos é implementada de maneira fraca e aquém do ideal.

Fechando mais o escopo, tem-se a delimitação do tipo de rede utilizada - a rede LPWAN (**ii**). Sua principal característica é ter um *link* de rádio de baixa potência e ser adequada a dispositivos restritos com suporte energético baseado em baterias de longa vida. Devido a suas limitações de tráfego de dados, a adoção de soluções existentes, como autenticação mútua, infraestrutura de chave publica (PKI) e criptográfica complexa, não é uma solução adequada para esse tipo de rede. Contudo, em sua maioria, os protocolos LPWAN já adotam soluções de segurança de rede em suas transmissões (MOHAMED et al., 2022; COMAN et al., 2019), como criptografia com AES-128, MAC e mitigação de *replay*. Desse modo, a solução aqui proposta pôde focar em aspectos diretamente relacionados à autenticação, deixando aspectos de integridade e sigilo para a camada de rede.

Quanto ao tipo de medição abarcada pela nossa proposta, ela se restringe a resolver a autenticação para dispositivos de *smart metering* (**iii**). Ou seja, ela é adequada somente para dados de medições simples, do tipo escalar ou categórico (i.e., temperatura, pressão, nível de fluidos). Dados complexos ou *stream* de dados não são abarcados pela proposta deste documento. Ainda assim, uma adequação da atual proposta para atender esses tipos de dados pode ser feita futuramente, desde que os componentes associados a isso sejam reprojetados e reimplementados.

Adicionalmente, esta proposta não é adequada a dados críticos (**iv**), fato que decorre da característica de latência das redes LPWAN e do tipo de solução projetada para o mecanismo de autenticação. Esse mecanismo não tem o foco em soluções de missão crítica e de tempo real, o que leva a identificação de violações de autenticação a ocorrer "eventualmente". Consequentemente, o acesso é interrompido sempre que uma violação de acesso acontece, contudo, não no mesmo momento (não instantaneamente).

Por último, tem-se o foco nos sensores de baixo custo e com disponibilidade de recursos restrita (*constrained devices* [**v**]), o que traz limitações para o projeto, como a não implementação de muitas casas decimais, a falta de suporte a algoritmos complexos de criptografia, a baixa capacidade de memória e outras limitações nas capacidades do processador ou microprocessador utilizados. Algumas facilidades promovidas por *System on Chip*(SoC) —CubieBoard e Raspberry Pi —como a capacidade de multiprocessamento e a extensão de memória, não fazem parte do escopo deste projeto e devem ser evitadas.

Em resumo, este projeto se dedica a solucionar a autenticação de dispositivos restritos de baixo custo, utilizando redes LPWAN, com dados simples e sem propósito de missão crítica. Contudo, mediante considerações específicas de cada área, e adaptações necessárias, esse mecanismo é aplicável a outras áreas de atuação de IoT. Vale lembrar que seu desempenho nessas situações deverá ser inferior a soluções projetadas especificamente para tais situações.

## 5.4 DIAGRAMAS DE SEQUÊNCIA

Nesta seção são apresentados os diagramas de sequência, em que são identificados atores, fluxo de troca de mensagens, tempo de vida de requisições e demais itens representáveis nesse tipo de modelagem de *software*. Sua adoção possibilitou a identificação de mensagens trocadas por cada instância de objeto dentro do nosso mecanismo e, por conseguinte, foi possível refatorar e dividir responsabilidades dentro de cada conceito criado, por meio do desenvolvimento de um conjunto de diagramas que descreve e explicita o processo de autenticação através do MFA_R.

Ainda, nesta seção são trazidos os diagramas do processo de registro do sensor (Figura 7) e de autenticação desse dispositivo (Figuras 8 e 10), assim como o processo de *enforcement* da autenticação (Figura 9). Por meio desses diagramas é possível ter uma visão dinâmica geral das principais trocas de mensagens que acontecem durante os processos do MFA_R.

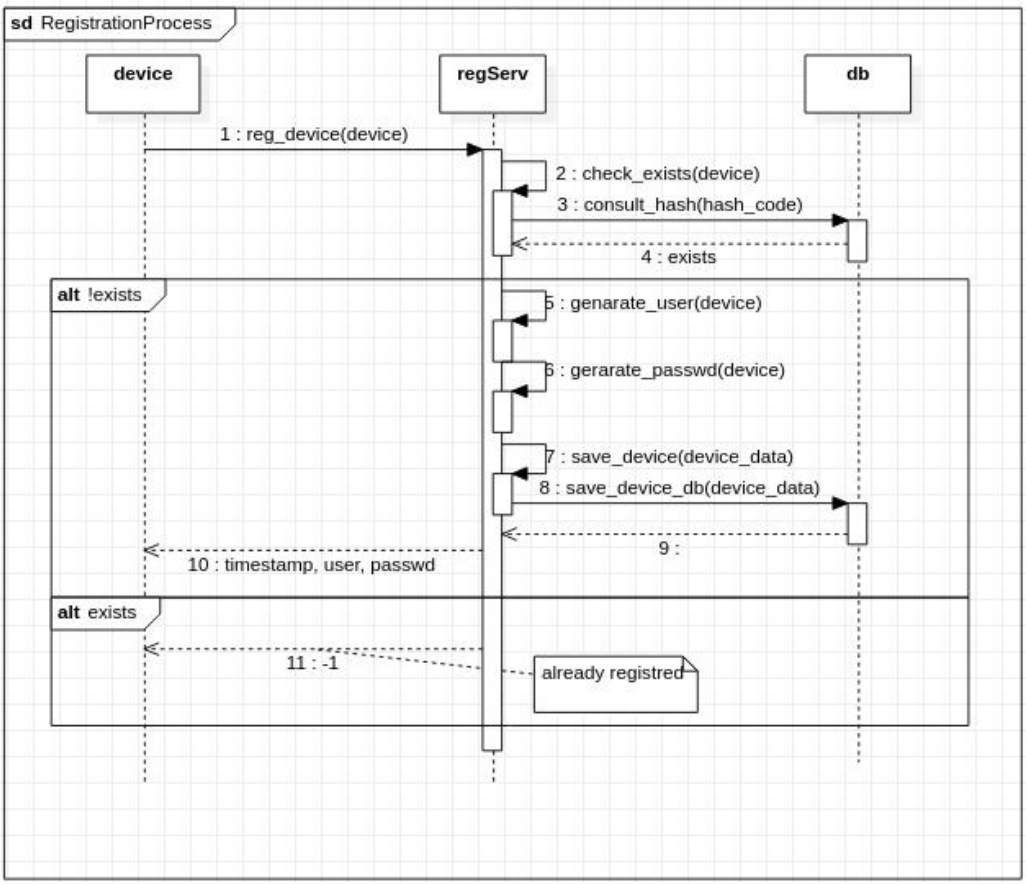



Figura 7 – Processo de registro de dispositivo

No diagrama apresentado (Figura 7), pode-se notar a comunicação entre os três objetos (*device*, *regServ*, e *db*) e seus dois fluxos alternativos. O *device* representa o sensor/dispositivo que está solicitando a validação de sua identificação para o mecanismo de autenticação. Já o *regServ* é o Serviço de Registro que está publicado no Servidor, dentro do *gateway*. Por último, o *db* representa a base de dados, dentro do servidor, que é utilizada para armazenamento dos

dados de registro dos dispositivos solicitantes.

Quanto à sequência de mensagens, ela inicia com a solicitação de registro ($reg_device(device)$) a partir do dispositivo ($device$). Sendo esse registro analisado no Serviço de Registro por meio da $check\_exists(device)$ que consulta o $db$, utilizando um *hash* dos dados do dispositivo ($consult\_hash(hash\_code)$), no caso de o dispositivo já estar registrado.

O primeiro fluxo alternativo ([**alt** $!exists$]) é onde acontece a geração do usuário, da senha e da persistência no banco de dados, com informações do dispositivo registrado. A geração de senha ($generate_passwd(device)$) e de usuário ($generate_user(device)$) utilizam os dados do dispositivo como base para origem dessas informações e, após a geração, o objeto $device\_data$, e não mais $device$, é salvo no $db$. Essa diferenciação acontece para sinalizar que os dados salvos no $db$ foram acrescidos das informações de autenticação e do *timestamp* do registro, antes de serem salvos. Já no segundo fluxo ([**alt** $exists$]), há somente uma sinalização $-1$ é enviada indicando que o dispositivo já estava registrado no mecanismo de autenticação.

A parte principal do mecanismo é o processo de autenticação e depois os passos que lhe dão suporte. Tal processo é expresso por meio dos diagramas 8 e 10 que, junto com o processo de *enforcement* (Figura 9), cobre todos os aspectos da autenticação.

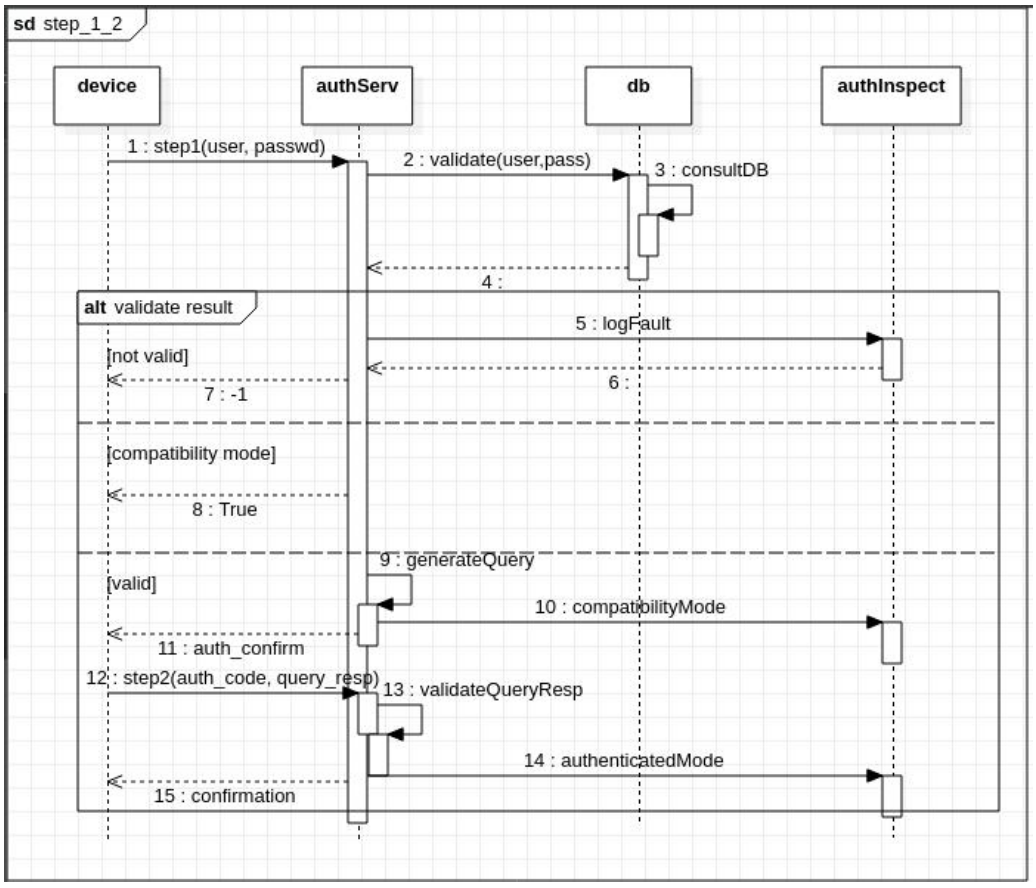



Figura 8 – Passos 1 e 2 da autenticação

O processo de autenticação se inicia com a mensagem de autenticação do primeiro fator ($step1(user, passwd)$) enviada ao $authServ$. Nesse processo estão presente quatro objetos:

o *device*, o *authServ*, o *db* e o *authInspector*. Tanto o *device* como o *db* já foram abordados anteriormente e mantêm a mesma definição ao longo desta seção. Já o *authServ* é definido como o serviço de autenticação e é responsável pela autenticação dos dois primeiros fatores. Por sua vez, o *authInspector* é o objeto responsável por dar suporte aos serviços envolvidos na autenticação e por gerir os processos de autenticação. Mais detalhes podem ser encontrados no capítulo anterior (Capítulo 4).

Após a validação, existem três possíveis fluxos alternativos: resultado inválido ($[notvalid]$), modo compatibilidade ($[compatibilitymode]$), e válido ($[valid]$). O primeiro fluxo é o login inválido que se inicia com a mensagem *logFault* entre o *authServ* e o *authInspector*, que sinaliza para esse segundo registrar o erro para um futuro processamento por outro mecanismo de segurança. Já o segundo é o modo compatibilidade, que registra o acesso e retorna a mensagem $TRUE$ para o dispositivo autenticado. Finalmente, no caso de os dados de autenticação serem válidos e o dispositivo não estar em modo de compatibilidade, o fluxo continuará da seguinte maneira: o *authServ* gera uma *query* (*generateQuery*), informa o *authInspector* que o dispositivo está autenticado como compatibilidade (*compatibilityMode*) e envia o código de acesso e a *query* para o dispositivo responder o segundo fator ($auth\_confirm$). A resposta do dispositivo é enviada contendo seu código de autenticação previamente atribuído e a resposta ao desafio (*query*) solicitada ($step2(auth\_code, query\_response)$). Uma vez recebida, essa mensagem é validada (*validateQueryResp*) e, ao ser confirmada sua validade, o dispositivo é considerado autenticado (*authenticatedMode*).

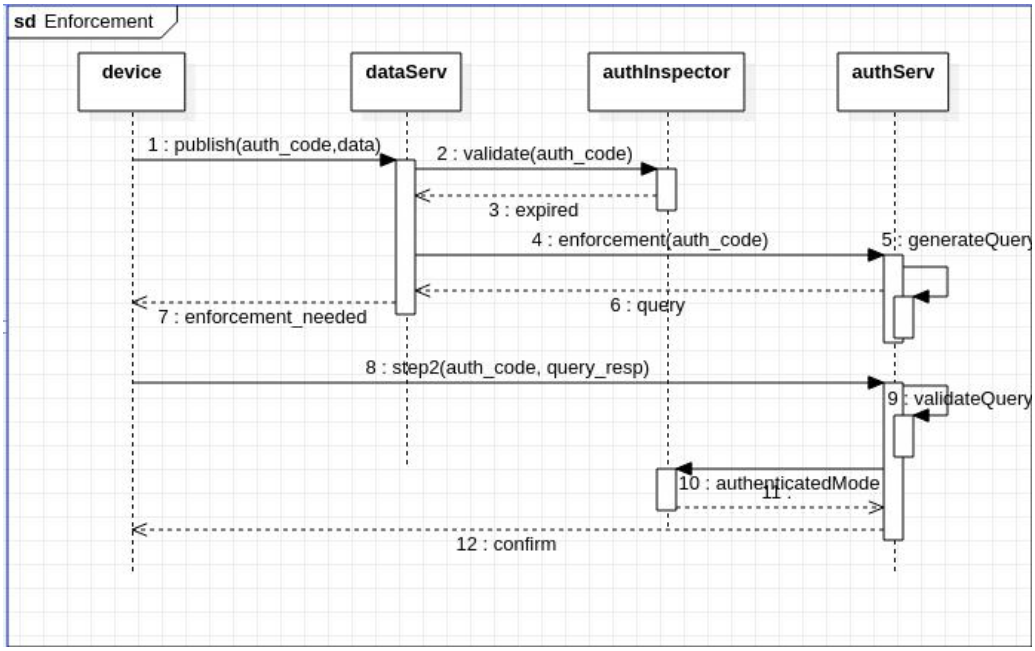



Figura 9 – *Enforcement* - segundo fator

Quanto ao processo de *enforcement* (Figura 9), ele acontece no momento em que o dispositivo tenta publicar dados. Após essa tentativa ($publish(auth\_code, data)$), ocorre uma validação do código de acesso do dispositivo ($validate(auth\_code)$) no *authInspector*. Caso

seja necessário o *enforcement* (i.e., código expirado), o *dataServ* solicitará ao *authServ* que inicie o processo de *enforcement* ($enforcement(auth\_code)$), enviando a resposta da publicação para o dispositivo, junto com a solicitação desse processo. Em seguida, o *device* responde com o segundo fator de autenticação novamente, o qual contém a reposta da *query* e o seu código de acesso ($step2(auth\_code, query\_resp)$). No caso de a resposta ser validada ($validateQuery$), o *authInspector* é notificado de que o *device* voltou ao modo autenticado (AUTH2) —$authenticatedMode$. Por fim, o *device* é notificado do sucesso do processo ($confirm$).

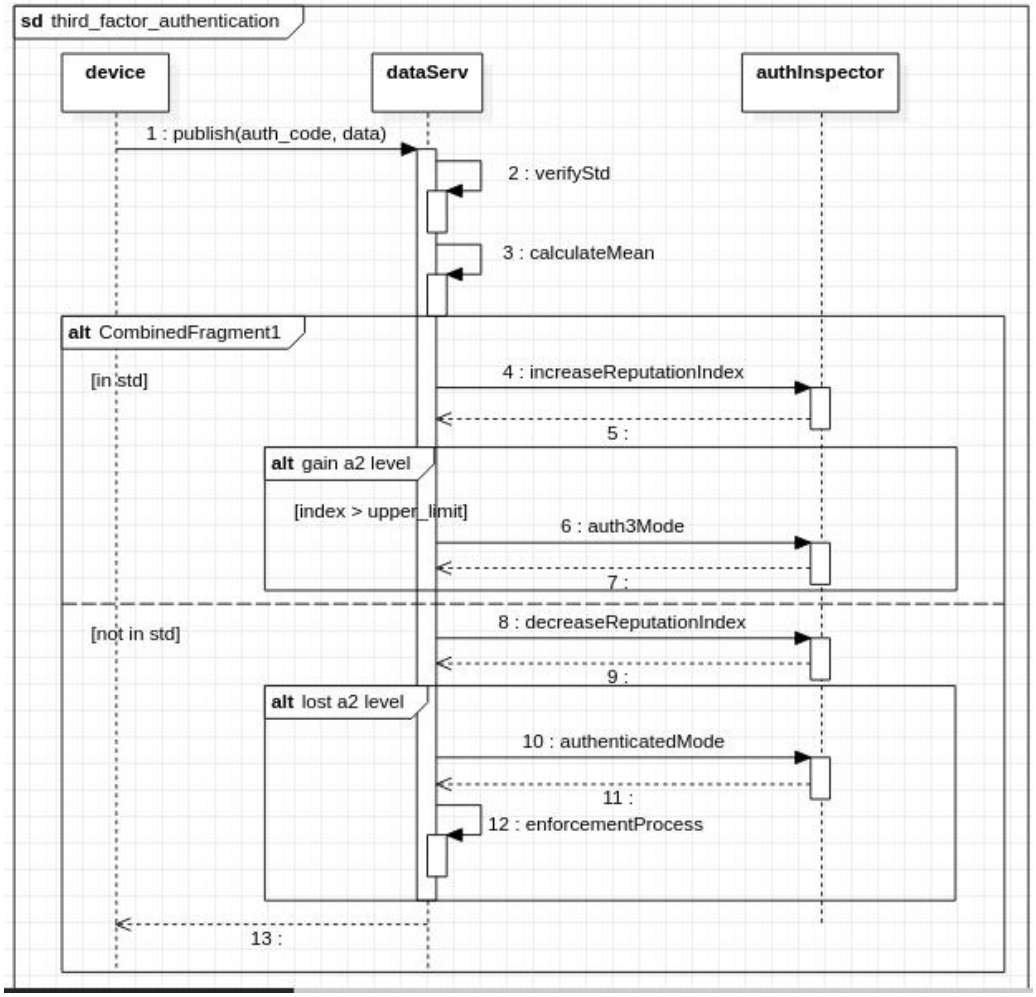



Figura 10 – *Third factor authentication*

Um ponto que deve ser bem esclarecido é o funcionamento do terceiro-fator, ou fator parte-todo. No diagrama apresentado (Figura 10), o *device* inicia o processo enviando ao *dataServ* uma mensagem de publicação de dados ($publish(auth\_code, data)$) que, ao ser recebida, desencadeia uma verificação da validade dos dados ($verifyStd$ e $calculateMean$). Desse modo, a partir dessa verificação dois possíveis caminhos podem ser seguidos pelo fluxo de dados: os dados enviados são válidos (i) ou os dados enviados são inválidos (ii). Ambos são discutidos mais detalhadamente a seguir.

Caso os dados sejam válidos (i), o *dataServ* solicitará ao *authInspector* que o valor

do *score* de reputação seja aumentado (*increaseReputationIndex*) para o dado dispositivo. Ao aumentarmos o *score* de reputação, ele ser suficiente para que o dispositivo fique elegível a se tornar autenticado no terceiro fator ($index > upper\_limit$). Especificamente, quando essa condição é atingida, o *authInspector* é informado de que o dispositivo subiu para o nível três de autenticação (AUTH3), ou seja, o dispositivo está no nível máximo de autenticação. Ao atingir esse nível, seu *score* de reputação se mantém estável, contudo, caso haja faltas, esse *score* sofrerá decrementos em seu valor.

Entretanto, caso os dados sejam inválidos (ii), o *dataServ* solicitará ao *authInspector* que o valor do *score* de reputação seja decrementado (*decreaseReputationIndex*), penalizando assim o dispositivo faltoso. Se o dispositivo estiver autenticado em AUTH3 e sua reputação atingir o limiar inferior ($index < lower\_limit$), ele é rebaixado para AUTH2. Em seguida, uma solicitação de *enforcement* é lançada (*enforcementProcess*). Contudo, caso o dispositivo já esteja em AUTH2, um *enforcement* é lançado imediatamente após a publicação faltosa. Em ambos casos, se o processo de *enforcement* não for respondido de maneira correta, o dispositivo será removido do sistema.

Como se pôde notar por meio dos diagramas de sequência aqui apresentados, foi possível demonstrar o funcionamento da proposta em seus aspectos dinâmicos, assim como de troca da mensagens entre algumas partes do sistema. Dessa forma, por meio de diagramas UML de sequência, foi possível perceber alguns aspectos dos fluxos de dados condicionados à evolução do sistema, como exemplo da autenticação do terceiro fator, que de outra forma não seria tão intuitivo de apresentar. E ainda, tais diagramas explicitam também os componentes lógicos necessários para o funcionamento da proposta e, como uma complementação, a próxima seção apresenta aspectos estruturais desta proposta por meio do diagrama de componentes dela.

## 5.5 DIAGRAMAS DE COMPONENTE

O multifator proporciona mais segurança para a autenticação (JOHANSSON et al., 2018), contudo, também traz maior complexidade, o que pode ser notado nos diversos componentes que colaboram para a execução (Figura 11). Detalhando, o *firmware* do dispositivo restrito se comunica com três componentes no lado do servidor: o serviço de registro (*RegServ*), o serviço de autenticação (*AuthServ*) e o serviço de publicação de dados (*DataServ*). Já os componentes do lado do servidor colaboram com o inspetor de autenticação (*AuthInspector*) e mantêm os dados de registro dos dados dos dispositivos (*device_data*).

De maneira geral, no diagrama de componentes (Figura 11) é mostrada uma visão geral do arcabouço de componentes do MFA_R. Nesse diagrama são apresentados dois nodos, cinco componentes, um artefato de dados e sete diferentes relações entre esses componentes e os artefatos. Ressalta-se que tal diagrama é de suma importância para o projeto, pois apresenta uma visão "caixa preta"dos componentes e permite ao leitor entender as principais relações entre o dispositivo e o *gateway*, assim como entre o dispositivo e as subpartes do *gateway*.

Quanto aos nodos, são apresentados dois nesse diagrama: o nodo do dispositivo e o

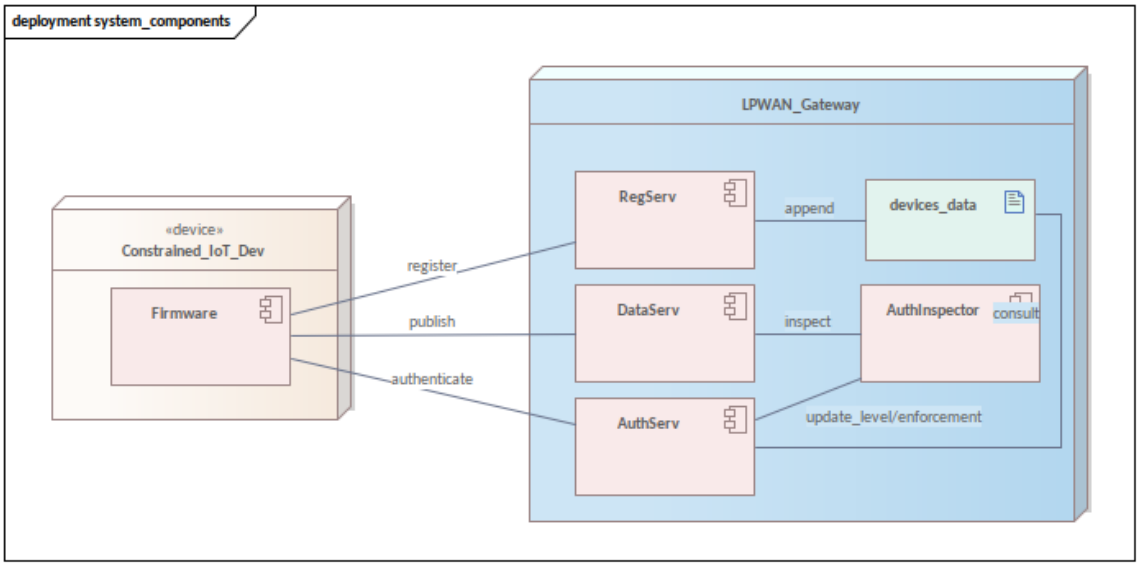



Figura 11 – Diagrama de componentes com principais componentes do mecanismo de autenticação multifator com reputação - À esquerda estão os nodos que representam o dispositivo restrito IoT e à direita o *gateway* LPWAN com os componentes do mecanismo de autenticação

nodo do *gateway*. Primeiramente, o nodo do dispositivo representa o sensor que está instalado no campo fazendo medições. Tal sensor tem um *firmware* que é responsável pela comunicação com os componentes do outro nodo (*gateway*) e pela execução da lógica de medição para envio dos dados. Esse componente deve implementar em seu *firmware* os processos necessários para a autenticação requerida para a utilização do MFA_R.

Já o segundo nodo, por sua vez, é o que modela o *gateway*. Observa-se que esse *gateway* pode ser replicado ou pode representar um Fog Node que está após um conjunto de *gateways*, de acordo com a implementação física utilizada. Em nossa modelagem, faremos uso dessa simplificação para que o cenário de uso demonstrado e a explicação do funcionamento do MFA_R não tenham que abarcar a complexidade de um conjunto de recursos maior. Contudo, para o funcionamento e explicação das partes do MFA_R, essa simplificação não traz problemas ou perda de representatividade. Quanto aos serviços publicados no servidor, eles são quatro:

- *RegServ* - Serviço de Registro;
- *AuthServ* - Serviço de Autenticação;
- *DataServ* - Serviço de Publicação de Dados;
- *AuthInspector* - Inspetor do Processo de Autenticação;

Assim posto, o nodo de LPWAN_Gateway, representado em azul, encapsula quatro componentes e um recurso de dados. Esse último armazena o conjunto de dados dos sensores registrados para a autenticação no sistema. Esses dados serão utilizados como fonte de dados para o segundo fator, que é um Challenge-Response (CR). Ainda, esse recurso de dados tem acesso disponível ao Serviço de Registro que executa atualizações e ao Serviço de Autenticação que faz consultas, ambos publicados no servidor.

Iniciando-se com o **RegServ**, ele tem a responsabilidade de fazer o registro dos dispositivos que serão autenticados no sistema, para posterior publicação de dados. Esse componente aparece no diagrama relacionando-se com o *firmware* do dispositivo através do método de *register*. É possível notar também que o RegServ faz a manutenção do recurso *devices_data* que está localizado no *gateway*. Essa manutenção se dá por meio da invocação do método de *append* desse mesmo recurso. Sendo assim, esse componente somente adiciona informações novas aos registros, não sendo de sua responsabilidade fazer alterações e exclusões de registro —dentro do contexto da nossa proposta.

Seguindo com o **AuthServ**, esse componente é responsável por se comunicar com o dispositivo para autenticação de multifator durante o uso e com o *AuthInspector* para executar seus processos de autenticação multifator. Com o dispositivo, a comunicação está representada pelo método *autenticate*, que encapsula os passos da autenticação. Já com o AuthInspector, ela se dá de duas formas: por meio da atualização dos níveis de autenticação do dispositivo (*update_level*) e do processo de *enforcement* de autenticação, que é solicitado pelo AuthInspector. Adicionalmente, o AuthServ consulta o recurso *devices_data* para o fator CR, o que ocorre através da Query API, apresentada no capítulo anterior.

Em terceiro lugar, tem-se o **DataServ** que recebe os dados, os processa e avalia a validade dos dados publicados, junto com o último componente. Os dados são enviados pelo dispositivo (*publish*) e, ao serem recebidos, é verificado pelo *AuthInspector* se o dispositivo está autenticado devidamente (*inspect*) e se os dados foram processados. Esse componente não se comunica diretamente com os demais ou com o recurso de *devices_data*, estando ele ligado somente ao *AuthInspector*, que monitorará e acompanhará a evolução dos processos de publicação de dados.

Por fim, como o **AuthInspector** não tem *interface* externa ao *gateway*, ou seja, seus serviços são consumidos diretamente pelos outros serviços do *gateway*, ele age como um intermediário, tendo uma visão geral de todo o processo de autenticação e publicação de dados, podendo, assim, propor intervenções quando for necessário (i.e. processo de *enforcement*). Nesse componente são publicados os níveis de autenticação e o *score* de reputação de todos os dispositivos que fazem parte do sistema IoT.

| Componente | Responsabilidades |
|---|---|
| **RegServ** | responsável pelo registro dos dispositivos no mecanismo |
| **AuthServ** | responsável pela autenticação dos dispositivos, *queries* e *enforcement* |
| **DataServ** | responsável pela publicação e consumo dos dados |
| **AuthInspector** | (i) manter o registro de todos os dispositivos autenticados e seus níveis de autenticação;<br>(ii) manter o registro da reputação dos sensores;<br>(iii) gerar *logs* de auditoria de tentativas de acesso indevido; e<br>(iv) executar *enforcements*. |

Tabela 11 – Componentes e suas responsabilidades

Como expresso na Tabela 11, existe uma clara divisão de responsabilidades para cada

serviço ou componente do sistema. Para nosso caso de validação do MFA_R, a utilização de todos os serviços e componentes em somente um *gateway* foi suficiente e atendeu a todas as necessidades da validação. Entretanto, essa não é a única forma de distribuição possível para esses componentes e, caso seja necessário, é possível a replicação e distribuição de alguns serviços/componentes em diferentes nodos durante a implantação. Dessa forma, é possível efetuar customizações para cada situação de distribuição ou implantação do MFA_R em um sistema IoT.

Por meio da visão de componentes, foi possível analisar a relação estática entre os componentes e os serviços. Adicionalmente, verificou-se para cada componente seu nível de dependência para recursos e para os demais componentes, assim como sua comunicação com esses componentes. Essa visão proposta pelo diagrama de componentes nos permitiu expressar a forma de implantação adotada nesta proposta e também comentar o potencial de adequação dessa forma de distribuição a futuras e diversas implantações do MFA_R.

## 5.6 DIAGRAMAS DE ESTADO - *STATE MACHINES*

Máquinas de estados (*State Machine* - STM) expressam a dinâmica dos estados internos de processos e componentes de *software* (AGGARWAL; SABHARWAL, 2012), sendo a ferramenta adequada para a modelagem do ciclo de vida dos componentes do MFA_R. A sequência de estados é chamada de *lifecycle* e evolui mediante a geração de eventos pelo componente e pelo sistema. Esse tipo de diagrama nos permitiu traçar a evolução de processos e componentes durante a autenticação em seus diferentes fatores, além de permitir expressar restrições (guardas) para a evolução de estados dentro do processo de autenticação, algo que seria muito complexo de ser explicitado de outra forma. Isso fica mais evidente na modelagem da evolução de estados para se chegar à autenticação no fator três (AUTH3).

Esta seção apresenta o comportamento dos componentes de *software* participantes do mecanismo. São apresentados quatro STMs: o processo de registro; a autenticação nos três fatores e o sistema de reputação; o ciclo de vida do sensor; e a evolução dos níveis de autenticação. Ao acompanharmos a evolução dessas STMs, foi possível visualizar a evolução de cada processo e do componente de *software* dentro do mecanismo.

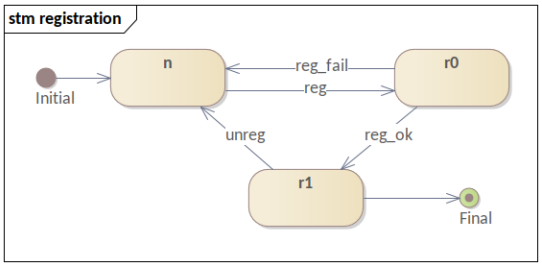



Figura 12 – Processo de registro de um dispositivo no mecanismo proposto

Durante o processo de registro (Figura 12) é possível ao sistema representar o disposi-

tivo por meio de três estados: não registrado ($n$), em registro ($r0$) e registrar ($r1$). A evolução entre os estados ocorre por meio da solicitação de registro enviada pelos dispositivos ($reg$), que pode ser aceita ($reg\_ok$) ou não ($reg\_fail$). Sendo aceita, o dispositivo torna-se registrado, saindo desse estado somente quando solicitado o processo de desregistro ($unreg$). Uma vez registrado no mecanismo, o dispositivo estará habilitado a executar o processo de autenticação e a posterior publicação dos dados. As máquinas de estado desses processos estão descritas mais adiante neste documento.

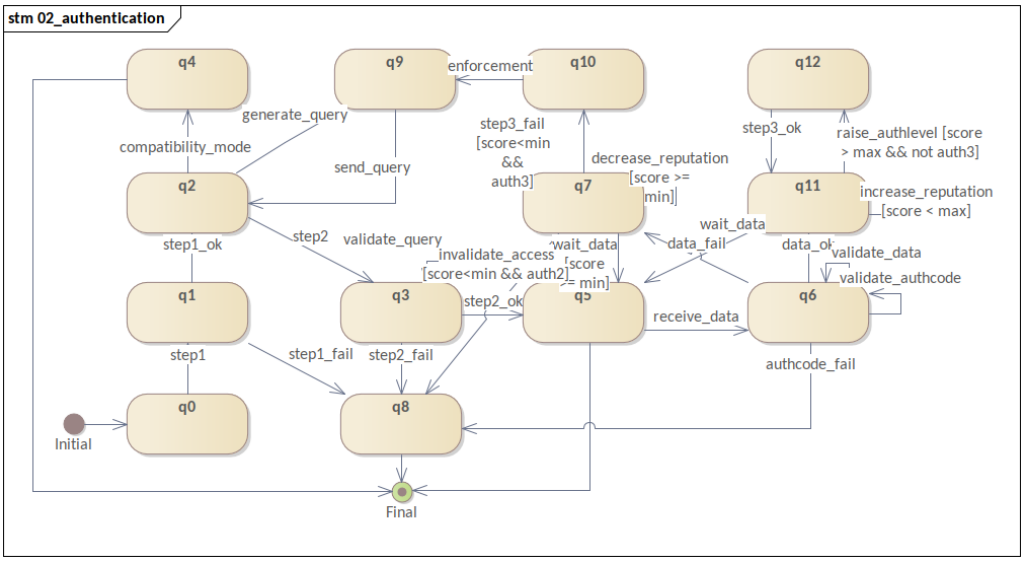



Figura 13 – Máquina de estados da autenticação com três fatores, *enforcement* e publicação de dados

No diagrama apresentado na Figura 13 estão descritos os processos de autenticação com fator um, autenticação com fator dois, publicação de dados, autenticação com fator três, sistema de reputação e possíveis casos de insucesso no processo. O diagrama fornece uma visão ampla do mecanismo de autenticação e, devido a isso, tornou-se extenso e um pouco mais complexo do que os demais.

A autenticação no primeiro fator começa com uma mensagem contendo as credenciais do dispositivo ($step1$) e uma validação é feita pelo AuthServ, que gera o evento de senha válida ($step1\_ok$). Após a autenticação com o fator, o dispositivo pode evoluir para um estado de autenticado em modo de compatibilidade ($q4$), ou o mecanismo pode gerar a *query* e enviá-la ao dispositivo ($q2 -> q9 -> q2$).

Caso haja uma falha na autenticação, o mecanismo evoluirá a representação do dispositivo para um estado de falha crítica ($q8$). Quanto aos próximos passos, uma vez enviada a *query*, em conjunto com o $auth\_code$, o mecanismo de autenticação esperará para receber a resposta da *query* e continuará a evoluir.

Para o fator dois, após receber a resposta da *query* ($step2$), o mecanismo evolui para seu estado de validação da *query* ($q3$). Caso a resposta da *query* esteja errada, o STM vai para um estado de falha crítica ($q8$); caso esteja correta, o mecanismo pode aguardar a publicação de dados ($q5$), que deverá ocorrer por meio do DataServ e com a utilização do $auth\_code$ obtido

pelo dispositivo.

Uma vez em estado de espera de dados ($q5$), já no nível de autenticação AUTH2, o sistema de reputação começa a funcionar e dispositivo recebe um valor inicial de reputação (*score*), que será mantido pelo processo de avaliação da publicação de dados. Após as publicações corretas de dados ($q11$), o *score* é incrementado ($increase\_reputation$) e, similarmente, será decrementado ($decrease\_reputation$), caso haja publicações de dados fora do padrão configurado ($q7$). Após sucessivas publicações válidas, a reputação do dispositivo passará do limite superior configurado ($score > max$) e o dispositivo estará elegível ($raise\_authlevel$) para subir de nível ($q12$) por meio do evento $step3\_ok$.

Como fluxos de falha, pode-se listar: (i) a recepção de dados inválidos e (ii) a recepção de um *auth_code* inválido. Quando há recepção de dados inválidos (i), o *score* do dispositivo é decrementado. Após sucessivas publicações de dados inválidos, o *score* atingirá o limiar inferior de reputação e será solicitado a ele um *enforcement*, quando esse estiver em AUTH3; quando em AUTH2, o dispositivo terá seu acesso invalidado ($invalidate\_access$), indo para o estado de falha crítica ($q8$). Caso o *enforcement* aconteça com sucesso, o dispositivo voltará a enviar dados para o sistema, mas com o nível de autenticação AUTH2. Caso o *enforcement* falhe, seu estado evoluirá para falha crítica ($q8$) e o dispositivo sairá do sistema. Já quando da recepção de um *auth_code* inválido (ii), o mecanismo invalidará o acesso ($authcode\_fail$) e evoluirá para o estado de falha crítica ($q8$).

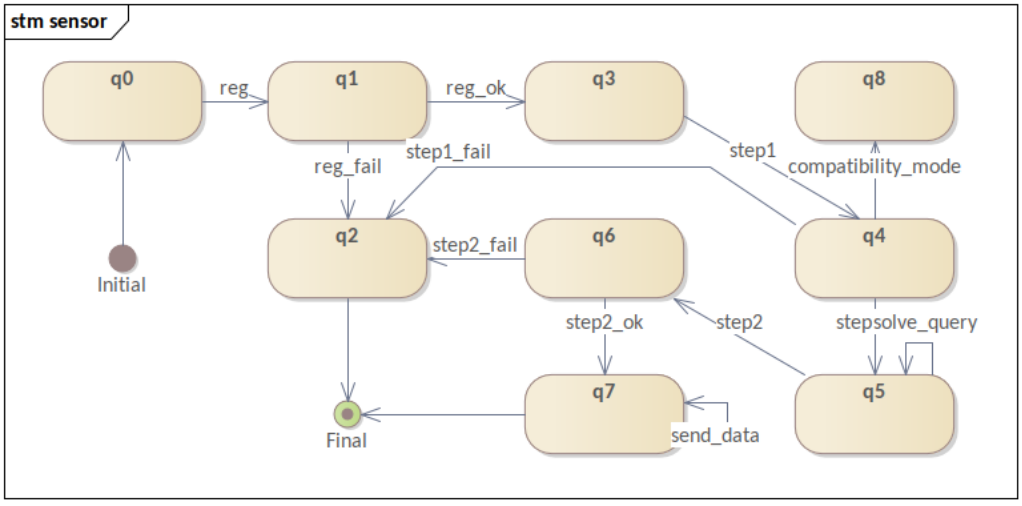



Figura 14 – Máquina de estados do ciclo de vida do sensor

Na Figura 14 observa-se que o STM do sensor demonstra o ciclo de vida do dispositivo e reflete a influência da evolução do mecanismo de autenticação no sensor e, mesmo que já tenham sido abordados alguns aspectos do sensor em outros diagramas, é importante salientar alguns fatos. O sensor inicia seu ciclo de vida não registrado ($q0$) e o finaliza de duas maneiras: por meio de uma falha crítica ($q2$) ou após seu ciclo de envio de dados ($q7$). Outra observação importante é que o dispositivo somente enviará dados após o sucesso na autenticação do segundo fator ($step2\_ok$) e que o funcionamento em modo compatibilidade não tem

uma representação de estados internos. Ainda assim, mesmo sem uma representação de estados internos, o mecanismo precisa saber quais são todos os dispositivos que estão funcionando em modo compatibilidade ($q8$).

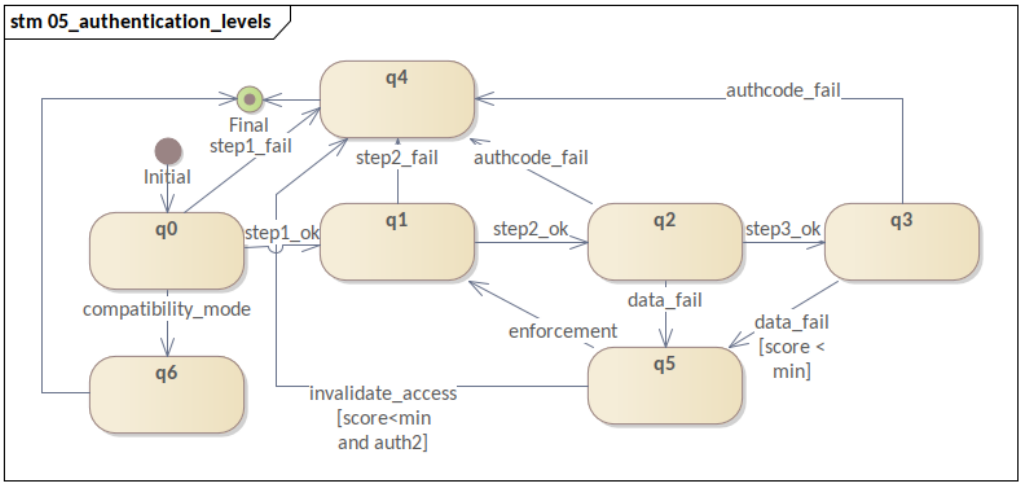



Figura 15 – Máquina de estados da evolução dos níveis de autenticação dentro do mecanismo proposto

Por fim, tem-se a STM que representa a evolução entre os níveis de autenticação (Figura 15). Seu principal objetivo é mostrar de maneira mais clara os passos essenciais para evolução nos níveis de autenticação e sua ordem. Dessa forma, ela representa de maneira transparente os principais fluxos de insucesso que levam a uma falha crítica ($q4$), entre os quais se pode citar a falha no primeiro fator ($step1\_fail$), a falha no segundo fator ($step2\_fail$), a falha no *auth_code* ($authcode\_fail$) e a invalidação de acesso ($invalidate\_access$).

Observe, ainda nesse diagrama, que não há uma transição de falha de terceiro fator (falha de reputação), o que pode levar a um *enforcement*. Por sua vez, a falha no **enforcement** pode levar a uma falha do segundo fator durante o processo de *enforcement*. Um resultado disso é que quando autenticado no nível AUTH3, um dispositivo somente sai do sistema se não executar um *enforcement* corretamente.

De modo geral, com a utilização de STMs foi possível detalhar o aspecto dinâmico de alguns componentes do mecanismo. Tais STMs proveram o nível de detalhamento necessário para a construção dos autômatos que serão descritos na próxima seção deste documento e que conferem suporte à validação formal de algumas propriedades às quais o mecanismo atende.

## 5.7 RESULTADOS DA MODELAGEM

Como resultado, este capítulo nos trouxe uma visão mais formalizada do processo de construção do MFA_R. De modo mais específico, mediante os diagramas de sequência, os diagramas de componentes e os diagramas de estados; pode-se expressar o conjunto de artefatos que participam do funcionamento correto desse mecanismo de autenticação multifator.

Os **diagramas de sequência** trouxeram os aspectos dinâmicos da comunicação entre os objetos do mecanismo e serviram para esclarecer detalhes da autenticação quanto ao com-

portamento dos objetos e quais mensagens são desencadeadas em cada momento da execução. Alguns aspectos mais complexos puderam ser esclarecidos durante a explicação desses diagramas, como exemplo o processo de autenticação do terceiro fator.

Já o **diagrama de componentes** nos trouxe uma visão das possibilidades de *deployment* do MFA_R. Adicionalmente, explicou-se a relação entre os componentes e o recurso existente no *gateway* para armazenamento dos dados dos dispositivos. Esse diagrama nos permite uma visão geral das possibilidades de distribuição, replicação e *avaibility* para futuras aplicações da MFA_R.

Por fim, o conjunto de **diagramas de sequência** permitiu observar o ciclo de vida de cada componente do sistema e como esse é dependente dos demais. Sem esses diagramas não teria sido possível a modelagem e a verificação das propriedades formais através de autômatos (vide Apêndice A.7) que garantiram o não intertravamento entre as partes distribuídas do sistema. Resultados adicionais podem ser encontrados nos artigos localizados nos apêndices, em especial as seções A.1 e A.6.

# 6 CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS

Este trabalho apresentou um mecanismo de autenticação multifator com suporte à reputação, tendo sido trazidas diferentes etapas do processo de criação desse mecanismo, tais como: revisão sistemática, proposta e modelagem. Adicionalmente, foram publicados alguns trabalhos que reportam etapas do desenvolvimento desse mecanismo, etapas como a escolha das tecnologias utilizadas, experimentação, custos computacionais de segurança, criação de uma ferramenta de simulação, validação formal da proposta (*safety*), entendimento das ameaças que podem afetá-la, seu desempenho em relação a outras opções, entre outras. De modo mais sucinto, como contribuições para área de segurança temos a especificação do mecanismo de autenticação e sua implementação, experimentação e validação formal.

A revisão sistemática nos permitiu ter uma visão geral da área e dos desafios encontrados em relação à segurança/autenticação, além de entender melhor a abordagem adotada pelos trabalhos que abordam o mesmo tema (trabalhos correlatos). Ainda na revisão sistemática, foi utilizada com sucesso uma adaptação do método ProKnow-C para obtenção do portfólio bibliográfico (Tabela 6). Como resultados, obtivemos uma confirmação da relevância do tema (Figura 6) e uma análise do estado-da-arte, por meio de gráficos e quadros relacionados.

No capítulo 4 foi exposta a solução proposta pelo mecanismo de autenticação, que se traduz em um texto descritivo explicando cada parte principal da solução e como ela se comunica com as demais partes. Não menos relevante, foram trazidos algoritmos que formalizam, de uma maneira sistêmica, a solução anteriormente exposta como texto. Dessa forma, o capítulo citado apresenta de modo formal uma descrição dos diversos processos internos do mecanismo.

Por último, este trabalho apresentou uma formalização do mecanismo por meio de premissas, escopo e diversos diagramas. Um desses diagramas foi o de sequência, que apresentou uma visão de cada processo e uma hierarquia de mensagens. Além desse, também apresentou outros diagramas, como o de estado, que mostrou o ciclo de vida das requisições, e o de componentes que contribuiu com uma visão da relação entre os componentes do trabalho. Como resultado, criou-se um suporte de artefatos de *software* que permitem a implementação desse mecanismo em qualquer linguagem.

É importante notarmos que alguns resultados práticos foram atingidos durante o tempo de construção desse mecanismo. Durante o seu trajeto, o presente trabalho permitiu a construção do portfólio de documentos utilizados para entender o estado-da-arte em algumas fases do desenvolvimento, como revisões e bibliometrias, um simulador (BEZERRA; WESTPHALL, 2020a) e experimentos (BEZERRA; WESTPHALL, 2020b; BEZERRA et al., 2023; BEZERRA; MARTINA; WESTPHALL, 2023). Como detalhamento da primeira parte, podemos listar: um estudo sobre a área de computação como serviço (BEZERRA; KOCH; WESTPHALL, 2020); uma visão da intersecção das áreas de autenticação; Fog Computing e dispositivos restritos (BEZERRA; WESTPHALL, 2022); as principais ameaças na autenticação multifator (BEZERRA; WESTPHALL, 2023); e uma análise bibliométrica dos modelos de ameaças para autenticação dos últimos 28 anos (BEZERRA et al., 2022). Mais detalhes sobre as publicações encontram-se

nos Apêndices.

Como trabalhos futuros, sugere-se uma complementação na fase experimental, iniciando com uma implementação de protótipo físico utilizando CoAP e LoRaWAN para avaliação em um ambiente com real e com interferências reais. Sugere-se também a avaliação do protocolo de autenticação implementado no mecanismo de autenticação através de um verificador de protocolos de segurança, como o Syther, AVISPA ou HOL. Por último, sugere-se uma experimentação em um ambiente simulado de redes (i.e., ns3) para avaliar a utilização desse mecanismo em larga escala. Entente-se que, por meio da execução desses trabalhos, será possível um refinamento do mecanismo proposto e a criação de artefatos documentais que auxiliem na continuidade da pesquisa nessa área de conhecimento.

# REFERÊNCIAS

AAZAM, M.; HUH, E.-N. Fog computing and smart gateway based communication for cloud of things. In: IEEE. **2014 International Conference on Future Internet of Things and Cloud**. [S.l.], 2014. p. 464–470.

AGGARWAL, M.; SABHARWAL, S. Test case generation from uml state machine diagram: A survey. In: IEEE. **2012 Third International Conference on Computer and Communication Technology**. [S.l.], 2012. p. 133–140.

ALI, B.; AWAD, A. I. Cyber and physical security vulnerability assessment for iot-based smart homes. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 18, n. 3, p. 817, 2018.

ALI, Z. et al. Edge-centric multimodal authentication system using encrypted biometric templates. **Future Generation Computer Systems**, Elsevier, v. 85, p. 76–87, 2018.

ALKHAWAJA, A. R.; FERREIRA, L. L.; ALBANO, M. Message oriented middleware with qos support for smart grids. In: **INForum 2012-Conference on Embedded Systems and Real Time.** [S.l.: s.n.], 2012.

AMEER, S. et al. Bluesky: Towards convergence of zero trust principles and score-based authorization for iot enabled smart systems. In: **Proceedings of the 27th ACM on Symposium on Access Control Models and Technologies**. [S.l.: s.n.], 2022. p. 235–244.

ANAKATH, A.; RAJAKUMAR, S.; AMBIKA, S. Privacy preserving multi factor authentication using trust management. **Cluster Computing**, Springer, v. 22, n. 5, p. 10817–10823, 2019.

ANANI, W.; OUDA, A.; HAMOU, A. A survey of wireless communications for iot echo-systems. In: IEEE. **2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)**. [S.l.], 2019. p. 1–6.

AU, M. H. et al. Privacy-preserving personal data operation on mobile cloud—chances and challenges over advanced persistent threat. **Future Generation Computer Systems**, Elsevier, v. 79, p. 337–349, 2018.

BANERJEE, S. et al. Multi-authority cp-abe-based user access control scheme with constant-size key and ciphertext for iot deployment. **Journal of Information Security and Applications**, Elsevier, v. 53, p. 102503, 2020.

BERGMANN, N. W.; ROBINSON, P. J. Server-based internet of things architecture. In: IEEE. **2012 IEEE Consumer Communications and Networking Conference (CCNC)**. [S.l.], 2012. p. 360–361.

BERTIN, E. et al. Access control in the internet of things: a survey of existing approaches and open research questions. **Annals of Telecommunications**, Springer, p. 1–14, 2019.

BERTINO, E.; TAKAHASHI, K. **Identity management: Concepts, technologies, and systems**. [S.l.]: Artech House, 2010.

BEZERRA, W. D. R. et al. An experimentation on coap multi factor authentication mechanism with reputation for internet of things constrained devices and low power wide area network. In: IEEE. **2023 International Conference on Information Networking (ICOIN)**. [S.l.], 2023. p. 67–72.

BEZERRA, W. dos R.; KOCH, F. L.; WESTPHALL, C. B. Models of computing as a service and iot: an analysis of the current scenario with applications using lpwan. **Revista de Sistemas de Informaçao da FSMA**, n. 25, p. 56–65, 2020.

BEZERRA, W. dos R. et al. A bibliometrics analysis on 28 years of authentication and threat model area. **arXiv e-prints**, p. arXiv–2209, 2022.

BEZERRA, W. dos R.; WESTPHALL, C. B. Ambiente de experimentação para avaliação protocolos de mensagem para iot na fog. In: SBC. **Anais do XI Workshop de Pesquisa Experimental da Internet do Futuro**. [S.l.], 2020. p. 1–6.

BEZERRA, W. dos R.; WESTPHALL, C. B. Avaliação de desempenho de protocolos de mensagens com arquitetura publish/subscribe no ambiente de computação em nevoeiro: um estudo sobre desempenho do mqtt, amqp e stomp. In: SBC. **Anais do Workshop de Pesquisa Experimental da Internet do Futuro**. [S.l.], 2020. p. 7–12.

BEZERRA, W. dos R.; WESTPHALL, C. B. Trends, opportunities, and challenges in using restricted device authentication in fog computing. **Revista de Sistemas de Informaçao da FSMA**, n. 30, p. 38–45, 2022.

BEZERRA, W. dos R.; WESTPHALL, C. B. Characteristics and main threats about multi-factor authentication: A survey. **Revista de Sistemas de Informaçao da FSMA**, n. 31, p. 69–75, 2023.

BEZERRA, W. R.; MARTINA, J. E.; WESTPHALL, C. B. A formal verification of a reputation multi-factor authentication mechanism for constrained devices and low-power wide-area network using temporal logic. **Sensors**, MDPI, v. 23, n. 15, p. 6933, 2023.

BHAWIYUGA, A.; DATA, M.; WARDA, A. Architectural design of token based authentication of mqtt protocol in constrained iot device. In: IEEE. **2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)**. [S.l.], 2017. p. 1–4.

BONOMI, F. et al. Fog computing: A platform for internet of things and analytics. In: **Big data and internet of things: A roadmap for smart environments**. [S.l.]: Springer, 2014. p. 169–186.

BREMLER-BARR, A.; LEVY, H. Spoofing prevention method. In: IEEE. **Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.** [S.l.], 2005. v. 1, p. 536–547.

BUCCAFURRI, F.; ROMOLO, C. A blockchain-based otp-authentication scheme for constrainded iot devices using mqtt. In: **Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control**. [S.l.: s.n.], 2019. p. 1–5.

BUDGEN, D.; BRERETON, P. Performing systematic literature reviews in software engineering. In: **Proceedings of the 28th international conference on Software engineering**. [S.l.: s.n.], 2006. p. 1051–1052.

CALVO, M.; BELTRÁN, M. A model for risk-based adaptive security controls. **Computers & Security**, Elsevier, v. 115, p. 102612, 2022.

CAO, J. et al. A survey on security aspects for 3gpp 5g networks. **IEEE Communications Surveys & Tutorials**, IEEE, 2019.

CAO, Y. et al. Specification and adaptive verification of access control policy for cyber-physical-social spaces. **Computers & Security**, Elsevier, v. 114, p. 102579, 2022.

CHAUDHRY, S. A. et al. An improved anonymous authentication scheme for distributed mobile cloud computing services. **Cluster Computing**, Springer, v. 22, n. 1, p. 1595–1609, 2019.

CHAVES, L. C. et al. Gestão do processo decisório: mapeamento ao tema conforme as delimitações postas pelos pesquisadores. **Revista Eletrônica de Estratégia & Negócios**, v. 5, n. 3, p. 3–27, 2012.

CHIANG, M.; ZHANG, T. Fog and iot: An overview of research opportunities. **IEEE Internet of Things Journal**, IEEE, v. 3, n. 6, p. 854–864, 2016.

COMAN, F. L. et al. Security issues in internet of things: Vulnerability analysis of lorawan, sigfox and nb-iot. In: IEEE. **2019 Global IoT Summit (GIoTS)**. [S.l.], 2019. p. 1–6.

CORSARO, A. The data distribution service tutorial. **Technical Report 4.0**, PrismTech, 2014.

DEEP, G. et al. Authentication protocol for cloud databases using blockchain mechanism. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 19, n. 20, p. 4444, 2019.

DEEP, S.; ZHENG, X.; HAMEY, L. A survey of security and privacy issues in the internet of things from the layered context. **arXiv preprint arXiv:1903.00846**, 2019.

DEEP, S.; ZHENG, X.; HAMEY, L. A survey of security and privacy issues in the internet of things from the layered context. **arXiv preprint arXiv:1903.00846**, 2019.

DIZDAREVIĆ, J. et al. A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration. **ACM Computing Surveys (CSUR)**, ACM, v. 51, n. 6, p. 116, 2019.

DOUCEUR, J. R. The sybil attack. In: SPRINGER. **International workshop on peer-to-peer systems**. [S.l.], 2002. p. 251–260.

DUAN, J.-H. et al. Forecasting fine-grained city-scale cellular traffic with sparse crowdsourced measurements. **Computer Networks**, Elsevier, v. 214, p. 109156, 2022.

ESINER, E.; DATTA, A. Two-factor authentication for trusted third party free dispersed storage. **Future Generation Computer Systems**, Elsevier, v. 90, p. 291–306, 2019.

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

FENG, D. et al. Blockchain-based secure crowdsourcing in wireless iot. **Journal of Communications and Information Networks**, PTP, v. 7, n. 1, p. 23–36, 2022.

FOTOUHI, M. et al. A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care iot. **Computer Networks**, Elsevier, v. 177, p. 107333, 2020.

FREMANTLE, P.; AZIZ, B. Oauthing: privacy-enhancing federation for the internet of things. In: IEEE. **2016 Cloudification of the Internet of Things (CIoT)**. [S.l.], 2016. p. 1–6.

GADRE, A.; ZHANG, D.; KUMAR, S. Towards enabling city-scale internet of things–challenges and opportunities. In: IEEE. **2019 11th International Conference on Communication Systems & Networks (COMSNETS)**. [S.l.], 2019. p. 72–79.

GOMI, H. et al. Context-aware authentication using co-located devices. In: IEEE. **2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)**. [S.l.], 2019. p. 304–311.

GRASSI, P. A.; GARCIA, M. E.; FENTON, J. L. Draft nist special publication 800-63-3 digital identity guidelines. **National Institute of Standards and Technology, Los Altos, CA**, 2017.

HADDADPAJOUH, H. et al. A survey on internet of things security: Requirements, challenges, and solutions. **Internet of Things**, Elsevier, p. 100129, 2019.

IBRAHIM, M.; MOHAMED, B.; HASSAN, M. F. An adaptive authentication and authorization model for service-oriented enterprise computing. **Kuwait Journal of Science**, v. 49, n. 1, 2022.

INSHI, S. et al. Secure adaptive context-aware abe for smart environments. **IoT**, MDPI, v. 4, n. 2, p. 112–130, 2023.

IORGA, M. et al. **Fog computing conceptual model**. [S.l.], 2018.

JOHANSSON, J. M. et al. **Approaches for providing multi-factor authentication credentials**. [S.l.]: Google Patents, 2018. US Patent 9,864,852.

KALARIA, R. et al. A secure mutual authentication approach to fog computing environment. **Computers & Security**, Elsevier, v. 111, p. 102483, 2021.

KAMBOU, S.; BOUABDALLAH, A. A strong authentication method for web/mobile services. In: IEEE. **2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)**. [S.l.], 2019. p. 124–129.

KHALID, H. et al. Selamat: A new secure and lightweight multi-factor authentication scheme for cross-platform industrial iot systems. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 21, n. 4, p. 1428, 2021.

KHAN, M. A.; SALAH, K. Iot security: Review, blockchain solutions, and open challenges. **Future Generation Computer Systems**, Elsevier, v. 82, p. 395–411, 2018.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering–a systematic literature review. **Information and software technology**, Elsevier, v. 51, n. 1, p. 7–15, 2009.

KITCHENHAM, B. A.; DYBA, T.; JORGENSEN, M. Evidence-based software engineering. In: IEEE. **Proceedings. 26th International Conference on Software Engineering**. [S.l.], 2004. p. 273–281.

KULIK, T. et al. A survey of practical formal methods for security. **Formal Aspects of Computing**, ACM New York, NY, v. 34, n. 1, p. 1–39, 2022.

LALLE, Y. et al. A comparative study of lorawan, sigfox, and nb-iot for smart water grid. In: IEEE. **2019 Global Information Infrastructure and Networking Symposium (GIIS)**. [S.l.], 2019. p. 1–6.

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

LEE, Y.-k. et al. Anonymous authentication system using group signature. In: IEEE. **2009 International Conference on Complex, Intelligent and Software Intensive Systems**. [S.l.], 2009. p. 1235–1239.

LI, J. et al. A fast and scalable authentication scheme in iot for smart living. **Future Generation Computer Systems**, Elsevier, v. 117, p. 125–137, 2021.

LIN, J.; SHEN, Z.; MIAO, C. Using blockchain technology to build trust in sharing lorawan iot. In: **Proceedings of the 2nd International Conference on Crowd Science and Engineering**. [S.l.: s.n.], 2017. p. 38–43.

LIU, Y. et al. A blockchain-based decentralized, fair and authenticated information sharing scheme in zero trust internet-of-things. **IEEE Transactions on Computers**, IEEE, v. 72, n. 2, p. 501–512, 2022.

LOFFI, L. et al. Mutual authentication with multi-factor in iot-fog-cloud environment. **Journal of Network and Computer Applications**, Elsevier, v. 176, p. 102932, 2021.

MATSUMOTO, T. Gummy and conductive silicone rubber fingers importance of vulnerability analysis. In: SPRINGER. **International Conference on the Theory and Application of Cryptology and Information Security**. [S.l.], 2002. p. 574–575.

MEKKI, K. et al. A comparative study of lpwan technologies for large-scale iot deployment. **ICT express**, Elsevier, v. 5, n. 1, p. 1–7, 2019.

MELL, P.; GRANCE, T. et al. The nist definition of cloud computing. Computer Security Division, Information Technology Laboratory, National . . . , 2011.

MENG, L. et al. A continuous authentication protocol without trust authority for zero trust architecture. **China Communications**, IEEE, v. 19, n. 8, p. 198–213, 2022.

MIESSLER, D. Securing the internet of things: Mapping attack surface areas using the owasp iot top 10. In: **RSA Conference**. [S.l.: s.n.], 2015.

MOHAMED, A. et al. Enhancing cyber security of lorawan gateways under adversarial attacks. **Sensors**, MDPI, v. 22, n. 9, p. 3498, 2022.

MOHER, D. et al. Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. **Annals of internal medicine**, Am Coll Physicians, v. 151, n. 4, p. 264–269, 2009.

NAIK, N. Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http. In: IEEE. **2017 IEEE international systems engineering symposium (ISSE)**. [S.l.], 2017. p. 1–7.

NAOUI, S.; ELHDHILI, M. E.; SAIDANE, L. A. Enhancing the security of the iot lorawan architecture. In: IEEE. **2016 International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)**. [S.l.], 2016. p. 1–7.

NEWSOME, J. et al. The sybil attack in sensor networks: analysis & defenses. In: IEEE. **Third international symposium on information processing in sensor networks, 2004. IPSN 2004**. [S.l.], 2004. p. 259–268.

OGU, R. E. et al. An iot solution for air quality monitoring and hazard identification for smart city development. In: IEEE. **2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON)**. [S.l.], 2022. p. 1–5.

OKIGBO, C. A. et al. Low cost air quality monitoring: comparing the energy consumption of an arduino against a raspberry pi based system. In: **Proceedings of the 2nd International Conference on Intelligent and Innovative Computing Applications**. [S.l.: s.n.], 2020. p. 1–8.

OLSSON, J. 6lowpan demystified. **Texas Instruments**, v. 13, 2014.

OMETOV, A. et al. Multi-factor authentication: A survey. **Cryptography**, Multidisciplinary Digital Publishing Institute, v. 2, n. 1, p. 1, 2018.

OWASP TOP 10 2018 Internet of Things. 2019. https://www.owasp.org/index.php/OWASP$_{I}nternet_{o}f_{T}hings_{P}roject.Accessed$ : $2019-12-03$.

PARDO-CASTELLOTE, G. Omg data-distribution service: Architectural overview. In: IEEE. **23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.** [S.l.], 2003. p. 200–206.

PRISCILA, S. S. et al. Risk-based access control mechanism for internet of vehicles using artificial intelligence. **Security and Communication Networks**, Hindawi, v. 2022, 2022.

RASTOGI, E. et al. Narrowband internet of things: A comprehensive study. **Computer Networks**, Elsevier, p. 107209, 2020.

ROBINSON, C.; FRANKLIN, R. S. The sensor desert quandary: What does it mean (not) to count in the smart city? **Transactions of the Institute of British Geographers**, Wiley Online Library, 2020.

ROMAN, R.; LOPEZ, J.; MAMBO, M. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. **Future Generation Computer Systems**, Elsevier, v. 78, p. 680–698, 2018.

ROUTRAY, S. K.; HUSSEIN, H. M. Narrowband iot: An appropriate solution for developing countries. **arXiv preprint arXiv:1903.04850**, 2019.

ROUTRAY, S. K. et al. Satellite based iot for mission critical applications. In: IEEE. **2019 International Conference on Data Science and Communication (IconDSC)**. [S.l.], 2019. p. 1–6.

SAHU, A. K.; SHARMA, S.; RAJA, R. Deep learning-based continuous authentication for an iot-enabled healthcare service. **Computers and Electrical Engineering**, Elsevier, v. 99, p. 107817, 2022.

SAINT-ANDRE, P. Extensible messaging and presence protocol (xmpp): Core. 2011.

SAMMOUD, A. et al. A secure and lightweight three-factor authentication and key generation scheme for direct communication between healthcare professionals and patient's wmsn. In: IEEE. **2020 IEEE Symposium on Computers and Communications (ISCC)**. [S.l.], 2020. p. 1–6.

SCARCELLI, A. et al. Smart app for personal dosimeter. In: IEEE. **2020 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)**. [S.l.], 2020. p. 1–6.

SCHUCKERS, S. A. Spoofing and anti-spoofing measures. **Information Security technical report**, Elsevier Advanced Technology, v. 7, n. 4, p. 56–62, 2002.

SHA, K. et al. A survey of edge computing based designs for iot security. **Digital Communications and Networks**, Elsevier, 2019.

SHAHRAKI, A. S.; RUDOLPH, C.; GROBLER, M. A dynamic access control policy model for sharing of healthcare data in multiple domains. In: IEEE. **2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)**. [S.l.], 2019. p. 618–625.

SIVANATHAN, A. et al. Low-cost flow-based security solutions for smart-home iot devices. In: IEEE. **2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)**. [S.l.], 2016. p. 1–6.

SMET, R. D. et al. Lightweight puf based authentication scheme for fog architecture. **Wireless Networks**, Springer, v. 27, n. 2, p. 947–959, 2021.

SONI, D.; MAKWANA, A. A survey on mqtt: a protocol of internet of things (iot). In: **International Conference On Telecommunication, Power Analysis And Computing Techniques (ICTPACT-2017)**. [S.l.: s.n.], 2017.

STALLINGS, W. et al. **Computer security: principles and practice**. [S.l.]: Pearson Education Upper Saddle River (NJ, 2012.

STANDARD, O. Oasis advanced message queuing protocol (amqp) version 1.0. **International Journal of Aerospace Engineering Hindawi www. hindawi. com**, v. 2018, 2012.

STANDARD, O. Mqtt version 3.1. 1. **URL http://docs. oasis-open. org/mqtt/mqtt/v3**, v. 1, 2014.

STEFANO, T.; SIMONE, Z.; LORENZO, V. Security analysis of lorawantm join procedure for internet of things networks. In: **IEEE Wireless Communications and Networking Conference Workshops (WCNCW)**. [S.l.: s.n.], 2017.

STOJMENOVIC, I. Fog computing: A cloud to the ground support for smart things and machine-to-machine networks. In: IEEE. **2014 Australasian telecommunication networks and applications conference (ATNAC)**. [S.l.], 2014. p. 117–122.

SYED, N. F. et al. Zero trust architecture (zta): A comprehensive survey. **IEEE Access**, IEEE, v. 10, p. 57143–57179, 2022.

SZYDŁO, T.; SUDER, P.; BIBRO, J. Message-oriented communication for ipv6-enabled pervasive devices. **Computer Science**, v. 14, 2013.

TIBURSKI, R. T. et al. Lightweight security architecture based on embedded virtualization and trust mechanisms for iot edge devices. **IEEE Communications Magazine**, IEEE, v. 57, n. 2, p. 67–73, 2019.

TSAI, J.-L.; LO, N.-W. A privacy-aware authentication scheme for distributed mobile cloud computing services. **IEEE systems journal**, IEEE, v. 9, n. 3, p. 805–815, 2015.

USING OAuth 2.0 to Access Google APIs. 2019. https://developers.google.com/identity/protocols/oauth2. Accessed: 2020-03-19.

VAQUERO, L. M.; RODERO-MERINO, L. Finding your way in the fog: Towards a comprehensive definition of fog computing. **ACM SIGCOMM Computer Communication Review**, ACM, v. 44, n. 5, p. 27–32, 2014.

VIEIRA, K. M. M. et al. Gerenciamento autonômico de segurança em cloud: provendo respostas à intrusão e considerando big data. 2017.

WHALEN, S. An introduction to arp spoofing. **Node99 [Online Document], April**, 2001.

WU, Z. et al. Spoofing and countermeasures for speaker verification: A survey. **speech communication**, Elsevier, v. 66, p. 130–153, 2015.

YI, S. et al. Fog computing: Platform and applications. In: IEEE. **2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)**. [S.l.], 2015. p. 73–78.

ZHANG, J. et al. Physical layer security for the internet of things: Authentication and key generation. **IEEE Wireless Communications**, IEEE, v. 26, n. 5, p. 92–98, 2019.

# A PUBLICAÇÕES E RESULTADOS

Aqui são elencadas as publicações feitas no período e que estão ligadas à pesquisa sobre o mecanismo de autenticação aqui apresentado. Algumas dessas publicações estão diretamente associadas a decisões na escolha de tecnologias que foram utilizadas durante as experimentações, algumas estão relacionadas a etapas do processo de construção e modelagem, e outras associadas a validações e experimentações.

A primeira página de cada artigo será apresentada, precedida de uma breve introdução. Na introdução é possível encontrar a referência para o artigo na íntegra, caso o leitor queira complementar sua busca. Isto acontece devido à restrição de direito de distribuição por parte de algumas editoras científicas, ou seja, mesmo sendo o autor, eu não tenho o direito de distribuição da obra, devendo o leitor buscá-la no portal da editora.

## A.1 AMBIENTE DE EXPERIMENTAÇÃO PARA AVALIAÇÃO PROTOCOLOS DE MENSAGEM PARA IOT NA FOG

Este trabalho (BEZERRA; WESTPHALL, 2020a) foi parte inicial da pesquisa e permitiu a criação da primeira versão do ambiente de experimentação, ambiente esse que foi utilizado durante todo o período em que se desenvolveu a pesquisa, evoluindo com ela.

O código-fonte foi disponibilizado de forma online no github do projeto[1] e está acessível para consulta. Sua construção utilizou a linguagem Python e bibliotecas que deram suporte de comunicação entre os protocolos de mensagem selecionados para experimentação.

Como resultado, obteve-se uma ferramenta que permitiu reproduzir em memória experimentos para validação de características de segurança dos protocolos e de diferentes métodos de autenticação.

[1] https://github.com/wesleybez/wpeif2020

# Ambiente de experimentação para avaliação protocolos de mensagem para IoT na Fog

**Wesley R. Bezerra, Carlos B. Westphall**

[1]INE – Universidade Federal do Santa Catarina (UFSC)
Florianópolis – SC – Brazil

{wesleybez,carlosbwestphall}@gmail.com

***Abstract.*** *Choose a message protocol to use in an IoT application involves several factors and a number of experiments. This work aims to share information about setting up an experimentation environment for IoT message protocols. It provides a description of the technologies used in programming, such as brokers and support for the generation of graphics. It also provides a narrative on challenges and choices in creating the experimentation environment. At the end it is possible to use the experience brought here to help in the creation of an experimentation environment, with some challenges already overcome.*

***Resumo.*** *A escolha de qual protocolo de mensagem usar em uma aplicação IoT envolve vários fatores e certo número de experimentações. Este trabalho visa compartilhar informações sobre a montagem de uma ambiente de experimentação para protocolos de mensagem em IoT. Traz um descrição das tecnologias utilizadas na programação, como brokers e suporte a geração de gráficos. Também traz um relato sobre os desafios e escolhas na criação do ambiente de experimentação. Ao fim é possível utilizar-se a experiência aqui trazida para ajudar na criação de um ambiente de experimentação, já com alguns desafios vencidos.*

## 1. Introdução

*Internet of Things* (IoT) é um tema que tem ganhado muita atenção dos pesquisadores. Suas tecnologias estão fragmentadas em diversas áreas do conhecimento, o que pode ser um desafio[Dizdarević et al. 2019]. Aliado este tema, temos paradigmas emergentes de computação como a computação em nevoeiro (FG-*Fog Computing*) que traz a proposta de aproximar recursos computacionais, armazenamento dos dispositivos finais (da borda)[Iorga et al. 2018], não sendo um substituto para a *cloud*, mas um complemento[Ahmed and Rehmani 2017]. A computação de borda é um novo foro para implantação de aplicações IoT novas e compreensíveis[Sha et al. 2019].

Devido a ainda ser uma área em desenvolvimento, existem poucos simuladores e ambientes de experimentação para soluções de IoT. Um desenvolvedor de aplicações que anseie validar uma tecnologia, protocolo ou biblioteca tem a tarefa árdua de implementar seu experimento do zero, sem contar com a experiência de outras desenvolvedores que já tiveram problemas parecidos.

Este trabalho tem dois objetivos:

i. descrição das tecnologias utilizadas no ambiente de experimentação;

## A.2 AVALIAÇÃO DE DESEMPENHO DE PROTOCOLOS DE MENSAGENS COM ARQUITETURA PUBLISH/SUBSCRIBE NO AMBIENTE DE COMPUTAÇÃO EM NEVOEIRO: UM ESTUDO SOBRE DESEMPENHO DO MQTT, AMQP E STOMP

Utilizando-se como base o ambiente de experimentação do artigo anterior, este artigo (BEZERRA; WESTPHALL, 2020b) avalia propriedades de segurança de alguns protocolos de mensagem. Tal experimento em memória utilizou um número diferente de instâncias de dispositivo para avaliar o crescimento do consumo de recursos em proporção ao número de dispositivos conectados.

A avaliação foi feita para cada protocolo selecionado e permitiu avaliar o impacto das escolhas de segurança e configurações na utilização energética (traduzida em tempo de processamento).

Nesse trabalho foram avaliados protocolos de *publish/subscribe*, como MQTT, STOMP e AMQP. O protocolo MQTT foi o que teve o melhor desempenho no geral, mas, mesmo assim, foi apresentada uma avaliação de todos os protocolos quanto a aspectos de desempenho.

# Avaliação de desempenho de protocolos de mensagens com arquitetura *publish/subscribe* no ambiente de computação em nevoeiro: um estudo sobre desempenho do MQTT, AMQP e STOMP

**Wesley R. Bezerra, Carlos B. Westphall**

[1]INE – Universidade Federal do Santa Catarina (UFSC)
Florianópolis – SC – Brazil

{wesleybez,carlosbwestphall}@gmail.com

***Abstract.*** *Secure solutions for IoT and agribusiness are critical. However, it is necessary to adopt the appropriate technology for the development of systems in this area. With a focus on pub/sub message protocols, our work provides an overview of the message protocols AQMP, MQTT and STOMP, a use case description of security application in agribusiness and an experiment to evaluate the temporal performance of the mentioned protocols. Finally, the results are discussed and it is verified which protocol is the most suitable for situations analogous to the mentioned use cases.*

***Resumo.*** *Soluções seguras para IoT e agronegócios são primordiais. Entretanto se faz necessária a adoção da tecnologia adequada para o desenvolvimento de sistemas nesta área. Com foco nos protocolos de mensagem pub/sub, nosso trabalho traz uma visão geral dos protocolos de mensagem AQMP, MQTT e STOMP, uma descrição de caso de uso de aplicação de segurança no agronegócio e uma experimentação para avaliação do desempenho temporal dos protocolos citados. Ao final, são discutidos os resultados e avaalido qual protocolo é o mais indicado para as situações análogas aos casos de uso citados.*

## 1. Introdução

O crescimento da adoção da *Internet* das Coisas (IoT - *Internet of Things*), no dia-a-dia das pessoas, tem exposto tanto os benefícios de tal tecnologia quanto os problemas de segurança [Yi et al. 2015, Aazam and Huh 2014, Bertin et al. 2019, Dizdarević et al. 2019] em diversas áreas. Uma dessas áreas é o agronegócio, onde a IoT tem sido utilizada para com várias finalidades, como: melhoria da gestão e rastreabilidade no setor agropecuário, melhor utilização de recursos no setor de agricultura de precisão, na gestão de estufas, entre outros. Tais aplicações demandam tecnologias seguras e que garantam a integridade dos dados transmitidos. Também deve garantir a proveniência dos dados a partir de sensores reais e pertencentes ao sistema.

Um desenvolvedor de soluções que integram IoT e agronegócios deve levar em conta fatores de segurança nos protocolos utilizados para mensagens de dados. Os protocolos de mensagem podem ser utilizados para comunicar tanto dispositivo-dispositivo, como dispositivo-*fog node*. Se faz relevante uma análise dos protocolos existentes, suas características e tempo de processamento de cada passo da sua utilização. Este último

## A.3 MODELS OF COMPUTING AS A SERVICE AND IOT: AN ANALYSIS OF THE CURRENT SCENARIO WITH APPLICATIONS USING LPWAN

Devido à grande quantidade de opções dos paradigmas de computação como serviço, optou-se por fazer uma análise de literatura (BEZERRA; KOCH; WESTPHALL, 2020) para a escolha do ambiente computacional para se aplicar o mecanismo proposto. Mediante um levantamento dos paradigmas e suas características, foi possível aprofundar em detalhes as diferenças entre eles, qualificando melhor e descrevendo o cenário proposto.

Desse modo, baseando-se nas características providas pela literatura, pôde-se selecionar o paradigma de computação em nevoeiro (*fog computing*) como o mais adequado para o nosso mecanismo. Isso se deu em decorrência de características como capacidade de processamento, distribuição local e menor latência em relação a outros paradigmas, como a *cloud computing*.

Revista de Sistemas de Informação da FSMA
n. 25 (2020) pp. 56-65

SISTEMAS DE INFORMAÇÃO
http://www.fsma.edu.br/si/sistemas.html

TRILHA PRINCIPAL

# Models of Computing as a Service and IoT: an analysis of the current scenario with applications using LPWAN

Wesley dos Reis Bezerra, *Ph.D Candidate, PPGCC, UFSC*,
Fernando Luiz Koch, *Visiting Researcher, PPGCC, UFSC*, Carlos Becker Westphall, *Prof. Dr., PPGCC, UFSC*,

***Resumo*—A computação em nuvem é um paradigma que transformou a forma de entrega de computação em sistemas distribuídos. Entretanto ela tem encontrado alguns desafios com a chegada massiva de dispositivos IoT, fato que demandou uma evolução e a criação de novos paradigmas baseados na cloud. Neste novo cenário com muitos diferentes paradigmas, é necessário uma análise de suas características e desafios para saber qual o mais adequado a aplicação desejada. Este trabalho trás um levantamento sobre os paradigmas derivados da *cloud* e que são aplicáveis a IoT, assim como uma comparação de suas características e desafios; por fim, faz uma análise utilizando um cenário hipotético de piscicultura com IoT para avaliar os paradigmas elencados. Através deste estudo é possível ter-se uma base de análise dos paradigmas e seus desafio para um escolha adequada no desenvolvimento de soluções IoT utilizando LPWAN e *contrained devices*.**

***Palavras-chave*—Computação em Nuvem, Low Power WAN, Internet of Things**

## Models of Computing as a Service and IoT: an analysis of the current scenario with applications using LPWAN

***Abstract*—**
**This work provides the basis to understand and select Cloud Computing models applied for the development of IoT solutions using Low-Power Wide Area Network (LPWAN). Cloud Computing paradigm has transformed how the industry implement solution, through the commoditization of shared IT infrastructures. The advent of massive Internet of Things (IoT) and related workloads brings new challenges to this scenario demanding malleable configurations where the resources are distributed closer to data sources. We introduce an analysis of existing solution architectures, along with an illustrative case from where we derive the lessons, challenges, and opportunities of combining these technologies for a new generation of Cloud-native solutions.**

***Index Terms*—Cloud Computing, Low Power WAN, Internet of Things**

Corresponding author: Wesley dos Reis Bezerra, *wesley-bez@gmail.com*

## I. Introduction

Cloud Computing is the engine to modern Internet of Things (IoT) solution development. However, this paradigm was originally designed for the purpose of shared IT infrastructure, primarily allowing for business of any size to trade on-premise infrastructure by a rented resources [1], [2].

As workloads start to migrate to IoT-based solutions, there are new challenges around distribution, heterogeneity, volume, velocity, variety, security, vulnerability and others [3]–[6]. Hence, there is a need to evolve the Cloud Computing paradigm with malleability, distribution, and closer proximity to the data sources. This is the origin of mixed models like Edge Computing, Fog Computing, Mist Computing and others.

On the other hand, the utilisation of Low-Power Wide Area Networks (LPWAN) and *publish-subscribe protocols* like *AMQP*, *MQTT*, *STOMP*, and *CoAP*, is increasingly popular in Cloud-IoT solution design. New challenges in combining these models revolve around issues of synchronisation, configuration, security, vulnerability and others [4], [5], [7]. For instance, more sophisticated security mechanisms require larger computing capacity, such as processing, memory utilisation, and power consumption. Engineers must measure the trade-offs between performance, security, and expected device cost while designing secure IoT devices and deploying distributed Cloud Computing configurations.

We argue that these issues can be mitigated by selecting the appropriate model combination for the solution demand. Solution designers and application developers need to understand the characteristics and capability of the diverse configurations and how they align with the system requirements in hands. Therefore, our research question is:

> What is the best Cloud Computing model to be applied for a given application scenario involving Cloud-IoT-LPWAN?

We introduce an analysis of the existing models of Cloud Computing applicable for the development of solutions involving IoT, LPWAN and constrained devices. Our goal

## A.4 TRENDS, OPPORTUNITIES, AND CHALLENGES IN USING RESTRICTED DEVICE AUTHENTICATION IN FOG COMPUTING

Esta é uma revisão literatura (BEZERRA; WESTPHALL, 2022) que permitiu criar a base documental para entender o cenário de autenticação para dispositivos restritos. Tal revisão permitiu detalhar os desafios, entender quais são as reais necessidades da área, e como essas necessidades estão sendo atendidas. Foi possível, ainda, traçar um paralelo acerca de como as tendências da área atendem essas necessidades e identificar oportunidades de pesquisa que não foram atendidas.

Por meio desse mapeamento foi possível confirmar e refinar a nossa hipótese de pesquisa, que é a necessidade uma autenticação leve para dispositivos restritos que utilizam LPWAN.

Revista de Sistemas de Informação da FSMA
n. 30 (2022) pp. 38-45

SISTEMAS DE INFORMAÇÃO
http://www.fsma.edu.br/si/sistemas.html

TRILHA PRINCIPAL

# Trends, Opportunities, and Challenges in Using Restricted Device Authentication in Fog Computing

Wesley dos Reis Bezerra, *PhD Candidate, PPGCC/UFSC*,
Carlos Becker Westphall, *Prof. Dr. PPGCC/UFSC*,

***Abstract*—The few resources available on constrained devices in Internet of Things are an important issue when we think about security. In this perspective, our work proposes an agile systematic review literature on works involving the Internet of Things, authentication, and Fog Computing. As a result, related works, opportunities, and challenges found at the intersection of those areas were brought, supporting other researchers and developers who work in those areas.**

***Index Terms*—internet of things, authentication, constrained devices, fog computing**

## I. Introduction

Security is a challenge in several areas of computing, especially for the Internet of Things (IoT)[1]. Specifically, when it has few resources[2]–[4], such as the battery, processing, storage, and throughput, security concerns may be relegated to the role of a minor and unimportant nuisance.. Thus, devices that have fewer resources tend to implement weaker security.

However, even with few resources, it is important to properly implement security in IoT[5] systems – in our case the implementation of device authentication. For that, it is necessary to know this specific area's trends, challenges, and opportunities. There was a lack of a preliminary study with the desired focus.

As a solution, this work aims to bring documentary resources that represent possible future developments in the area but also exposes problems that researchers/developers must avoid. This material was obtained from an agile systematic literature review (SLR) to answer the following questions.

- what are the opportunities for IoT authentication in Fog Computing?
- what are the challenges for IoT authentication in Fog Computing?
- what are the main works related to the topic of IoT authentication at Fog Computing?

The work is organized as follows: in the second section, the materials and methods used were presented; then, in the third section, a agile systematic literature review was presented; in the next section, there is the quantitative analysis of the data obtained from the researched works; finally, the conclusion and future works were brought in the sixth section.

## II. Materials and Methods

This study utilized the customization of the ProKnow-C [6] and EBSE [7] systematic review method focusing on the state of the art opportunities in the researched area, Figure 1. The systematic review is a structural investigation which uses systematic procedures for searching, synthesis, and analysis [8] of the collected evidence. This methodology enabled the reduction of bias in surveying the bibliographic portfolio [9], obtaining quantitative data and a more focused, higher quality portfolio.

Significant tools contributed to greater quality and replication of the process. Mendeley[1], was used for managing the bibliographic portfolio, which allowed for the creation of folders to organize articles and store them while synchronized with the cloud. With respect to creating datasets, LibreOffice Calc [2] - was used to create the spreadsheets that documented the project and the datasets' generation in the CSV format to read later and create charts. Lastly, concerning the generation of charts, GNUPlot[3] - enabled the automation of charts in this work.

## III. Systematic Review of Literature

The ACM Digital Library, IEEE Xplorer, Scopus, ScienceDirect, and Scielo portals were selected due to its expressiveness in the area of computing. All chosen portals allow access to many publications in journals and conferences. Moreover, such portals permit free access to many publications through partnerships between universities and CAPES[4].

Concerning the engineering process of query-string, four main areas were chosen. The selected areas were IoT, authentication, fog computing, and message protocols. The results can be seen in Table I, which offers an overview of the last three years of publications in each area. A three-year time window (2018-2020) was chosen due to the present study because we are only seeking new works and trends in the mentioned areas. For the initial exploration of the number of

Autor correspondente: Wesley dos Reis Bezerra, *wesleybez@gmail.com*

[1]https://www.mendeley.com, which is an important research tool[10] among both students and other researchers[11]

[2]https://pt-br.libreoffice.org/ - an open-source software project for office automation with a strong community[12]

[3]http://www.gnuplot.info/ a command line tool for the generation of charts [13] used by different IoT researchers[14]–[16] as well

[4]https://www.periodicos.capes.gov.br/

## A.5 CHARACTERISTICS AND MAIN THREATS ABOUT MULTI-FACTOR AUTHENTICATION: A SURVEY

Nesta revisão de literatura (BEZERRA; WESTPHALL, 2023) foram levantadas as principais características e ameaças que afetam autenticação multifator. Esse documento serviu como base para a criação de um modelo de ameaças para o mecanismo proposto e que foi utilizado em outras publicações.

Revista de Sistemas de Informação da FSMA
n. 31 (2023) pp. 69-75

SISTEMAS DE INFORMAÇÃO
http://www.fsma.edu.br/si/sistemas.html

TRILHA PRINCIPAL

# Characteristics and Main Threats about Multi-Factor Authentication: A Survey

Wesley dos Reis Bezerra, *PhD Candidate, PPGCC/UFSC*,
Carlos Becker Westphall, *Prof. Dr. PPGCC/UFSC*,

***Abstract*—This work reports that the Systematic Literature Review process is responsible for providing theoretical support to research in the Threat Model and Multi-Factor Authentication. However, different from the related works, this study aims to evaluate the main characteristics of authentication solutions and their threat model. Also, it intends to list characteristics, threats, and related content to a state-of-art. As a result, we brought a portfolio analysis through charts, figures, and tables presented in the discussion section.**

***Index Terms*—internet of things, authentication, constrained devices, fog computing**

## I. Introduction

Our work reports the Systematic Literature Review process responsible for providing theoretical support to research in the intersection of Threat Model (TM) and Multi-Factor Authentication (MFA) areas. Additionally, it aims to build a bibliographic portfolio capable of guiding the discussions in that area and being the theoretical support necessary for put forward the previously cited research areas.

Specifically, the present work focuses on multi-factor authentication. That focus is one of many sub-areas of authentication, a classical area of security that has evolved into wide-ranging sub-areas. Some related works in this sub-area range from authentication schemes [1], biometrics storage [2], the diversity between authentication factors [3]–[5]. However, different from our work, the selected documents (portfolio) evaluated the main characteristics of authentication solutions and their threat model.

This work searches for publications on threat models for multi-factor authentication. With this publication, we intend to list characteristics, threats, and related content to state-of-art research in MFA. Consequently, white papers, patents, or fewer academic documents were removed from the portfolio selection. In general, this review aims to answer the following research questions:

- Q1 what are the **main articles** in the selected research area?
- Q2 what are the **main characteristics** intended by the analyzed authentication solutions?
- Q3 what are the **main threats** listed in the threat models that appear in the selected portfolio?

As for paper organization, this work continues with the Systematic Literature Review in section II; the sequence discussed the results in III. The IV, the conclusions, and future work for this section are brought.

Autor correspondente: Wesley dos Reis Bezerra, *wesleybez@gmail.com*

TABLE I
List of Abbreviations

| Abbreviation | Meaning |
|---|---|
| CR | Challenge-Response |
| DDoS | Distributed Denial of Service |
| DFD | Data Flow Diagram |
| MFA | Multi-Factor Authentication |
| OTP | One-Time Password |
| SLR | Systematic Literature Review |
| SSO | Single Sign-On |
| TM | Threat Model |
| U2F | Universal 2nd Factor |
| TTS | Text-To-Speech |

## II. Systematic Literature Review (SLR)

The systematic literature review is a process/methodology whose goals is to promote the reduction of bias in scientific research [6]. However, it is not limited to this type of research [7], [8], but expands its results to constructing didactic material, classes, and books, a solid base for building a knowledge base.

In this work, we used the adapted ProKnow-C [9] methodology for the Systematic Literature Review methodology. Such methodology consists of four macrosteps: (i) portfolio selection, (ii) systematic review, (iii) bibliometrics, and (iv) research questions. Nevertheless, the research questions were already established a priori, and the answer to the research questions replaced the last step.

In this section, the macrosteps of (i) portfolio selection and (ii) systematic review will be further explored. The portfolio was chosen through these macro-steps in a documented and replicable manner. Additionally, the lenses (points of view) were chosen, analyzed, and contributed to the successful conclusion of this study. Thus, this review phase is the most important in this work.

The search term used in this review is given by the following condition:

## A.6 AN EXPERIMENTATION ON COAP MULTI FACTOR AUTHENTICATION MECHANISM WITH REPUTATION FOR INTERNET OF THINGS CONSTRAINED DEVICES AND LOW POWER WIDE AREA NETWORK

Este documento (BEZERRA et al., 2023) apresenta uma experimentação que compara o mecanismo proposto com uma referência de autenticação clássica, e com a utilização do sistema sem autenticação. A experimentação avalia memória e tempo de processamento (que pode ser traduzido em consumo de energia) com diferentes cargas de trabalho e diferentes opções de segurança.

Ao fim, foi possível comprovar que, apesar da perda de desempenho em comparação a uma autenticação somente utilizando login/senha, o MFA_R apresentou um desempenho adequado. Adicionalmente, o mecanismo trouxe um maior nível de segurança, devido à utilização de múltiplos fatores para autenticação. Dessa forma, o desempenho temporal e espacial mostrou-se adequado para dispositivos restritos.

# An Experimentation on CoAP Multi Factor Authentication Mechanism with Reputation for Internet of Things Constrained Devices and Low Power Wide Area Network

Wesley dos Reis Bezerra*,Ricardo do Nascimento Boing†, Cristiano Antonio de Souza‡, Carlos Becker Westphall§
*PPGCC —UFSC - Federal University of Santa Catarina*
University Campus - Trindade, Florianópolis/SC, Brazil, Florianópolis
E-mail: {*wesleybez,†ricardoboing.ufsc,‡cristianoantonio.souza10,§carlosbwestphall}@gmail.com

***Abstract*—The security of constrained devices in Internet of Things presents itself as a challenge due to the limitation of existing resources. It is important to analyze appropriate security mechanisms for this resource-constrained environment, specifically for authentication. This study presents an experiment that analyzes a proposal for an original Constrained Application Protocol Multi-Factor Authentication with Reputation, in comparison to simple authentication and a reference with no authentication. From this experience it was possible to prove that multi-factor authentication with reputation is also an adequate solution for Low Power Wide Area Network and constrained devices and does not require much more resources than simple authentication. With this work it is possible to evaluate the adoption of Multi Factor Authentication with Reputation on Constrained Devices and to subsidize choices of Internet of Things projects with this type of configuration.**

***Index Terms*—security, authentication, network security, lpwan**

## I. Introduction

Identity management on Internet of Things (IoT) devices is a security challenge to be overcome. Specifically, the issue of authentication has not been exhaustively addressed in some areas of IoT and needs to be further investigated in the search for solutions tailored to some limitations [1]–[3]. Device capacity limitations pose challenges in the use of complex algorithms and cryptography during the authentication process. Correspondingly, limitations on data transmission are common to some Low Power Wide Area Network (LPWAN) protocols. Such restrictions create further challenges for a more robust authentication [4], [5].

The Multi-Factor Authentication with Reputation (MFA_R) exposed here is part of a Ph.D. project where this complex multi-factor authentication mechanism aims to provide authentication at a low computational cost (battery, processing, and network) to Constrained Devices (CD). The experimentation presented here is a crucial architecture validation due to the MFA_R mechanism proposal. Therefore, the same architecture in this study could be applied to different message protocols for IoT.

### A. Contribution

This study presents an experiment on the multi-factor authentication mechanism with reputation for restricted IoT devices in LPWAN networks. Even in networks in which the use of SSL/TLS[1] is not recommended due to data traffic limitations, with the presented mechanism it will be possible to more securely authenticate the mentioned devices. In this work, we present the following contributions:

- a overview about the MFA_R proposal;
- a high-level modeling of the experiment on a multi-factor authentication for constrained devices (CD) and LPWAN;
- the presentation of the results of the experiment comparing a reference implementation with no authentication, with secret-based authentication and with MFA_R authentication;

### B. Paper Outline

The document is organized as follows: in Section II we brought a related work brief analysis. In Section III there are a theoretical background and a MFA_R proposal overview. In Section IV, there is a modeling of the MFA and the experiment with other solutions. In Section V, a discussion about the results of the experiment are presented. Lastly, in Section VI, we have the proposed conclusion and future work.

## II. Related Works

In literature, other authors have addressed this same problem. Li et. al. [6] takes the low interactivity approach to the key exchange protocol to mitigate latency. Its implementation is validated through the Bellare-Pointcheval-Rogaway model, together with theoretical and experimental evaluations. Anakath et al. [7] argue about the importance of ensuring authentication factors in the cloud; for this, they propose a trust model and submit it to experiments, claiming about its efficiency.

Next, Khalid et al. [8] present an MFA scheme for industrial IoT for preventing unauthorized access to fog servers; this

[1]Secure Sockets Layer/Transport Layer Security

## A.7 A FORMAL VERIFICATION OF A REPUTATION MULTI-FACTOR AUTHENTICATION MECHANISM FOR CONSTRAINED DEVICES AND LOW-POWER WIDE-AREA NETWORK USING TEMPORAL LOGIC

Nesta publicação (BEZERRA; MARTINA; WESTPHALL, 2023) foi verificada formalmente a segurança (*safety*) do funcionamento do mecanismo de maneira coordenada entre as suas partes. Foi utilizada uma modelagem que evolui desde as máquinas de estado até os modelos com autômatos temporizados. Também foi feita uma modelagem de ameaças, que depois foi traduzida para linguagem temporal para posterior verificação.

Como resultado, foi comprovada a segurança (*safety*) do mecanismo e algumas outras propriedades no modelo, como *deadlock free*. Portanto, podemos afirmar que esse mecanismo é seguro (*safety*) quanto a sua proposta e modelagem, evitando problemas de intertravamento entre as suas partes e garantindo algumas propriedades de segurança.

sensors

MDPI

*Article*

# A Formal Verification of a Reputation Multi-Factor Authentication Mechanism for Constrained Devices and Low-Power Wide-Area Network Using Temporal Logic

**Wesley R. Bezerra *, Jean E. Martina and Carlos B. Westphall**

Campus Universitário—Trindade, UFSC—Federal University of Santa Catarina, Florianópolis 88040-380, SC, Brazil; jean.martina@ufsc.br (J.E.M.); carlosbwestphall@gmail.com (C.B.W.)
* Correspondence: wesleybez@gmail.com

**Abstract:** There are many security challenges in IoT, especially related to the authentication of restricted devices in long-distance and low-throughput networks. Problems such as impersonation, privacy issues, and excessive battery usage are some of the existing problems evaluated through the threat modeling of this work. A formal assessment of security solutions for their compliance in addressing such threats is desirable. Although several works address the verification of security protocols, verifying the security of components and their non-locking has been little explored. This work proposes to analyze the design-time security of the components of a multi-factor authentication mechanism with a reputation regarding security requirements that go beyond encryption or secrecy in data transmission. As a result, it was observed through temporal logic that the mechanism is deadlock-free and meets the requirements established in this work. Although it is not a work aimed at modeling the security mechanism, this document provides the necessary details for a better understanding of the mechanism and, consequently, the process of formal verification of its security properties.

**Keywords:** multi-factor authentication; timed automata; formal verification; security



**Citation:** Bezerra, W.R.; Martina, J.E.; Westphall, C.B. A Formal Verification of a Reputation Multi-Factor Authentication Mechanism for Constrained Devices and Low-Power Wide-Area Network Using Temporal Logic. *Sensors* **2023**, *23*, 6933. https://doi.org/10.3390/s23156933

Academic Editor: Alessandra Rizzardi

Received: 29 June 2023
Revised: 17 July 2023
Accepted: 28 July 2023
Published: 3 August 2023



## 1. Introduction

There is an inherent need for a lightweight approach to authenticating restricted devices in the Internet of Things (IoT). This is a research topic of growing importance today [1–4], stemming from different motivations, ranging from the fact that IoT devices are used as support for Distributed Denial of Service (DDoS) (Distributed Denial of Service is an attack where zombie devices are commonly used to send packets to a target, thus causing an unavailability of access to the target service [5–8]) to the lack of adequate security support for surveillance systems [9,10]. Thus, security in IoT systems presents challenges, especially for constrained devices (CD) and with connectivity through low-power wide-area networks (LPWAN).

Many advances have been suggested in the field of IoT. Examples such as the IoT softwarization [11]—a tendency that came from the network area—evidenced a need for updates of security strategies for IoT, as corroborated by the work of Kaur et al. [12] presenting challenges from different areas. It is still possible to cite the application of IoT in the solution of security problems, as demonstrated in the work of Gupta et al., which used watermarking and lightweight encryption to encrypt images.

However, IoT challenges must be addressed in different ways to resolve and guarantee the effective and proper functioning of the system, for example, through the formal validation of security proposals to be implemented. Thus, a formal assessment of the modeling of components and systems is of paramount importance, especially systems related to security [13]. Beyond the validation of the security protocol while verifying the system on a whole, evaluation can occur through several formal verification approaches.