UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Bianca de Espíndola Manoel

Análise da viabilidade de uma camada de segurança para um dispositivo vestível cardíaco empregando conceitos de Internet das Coisas Médicas

Araranguá
2022

Bianca de Espíndola Manoel

Análise da viabilidade de uma camada de segurança para um dispositivo vestível cardíaco empregando conceitos de Internet das Coisas Médicas

Trabalho de Conclusão de Curso do Curso de Graduação em Engenharia de Computação submetido ao Centro de Ciências, Tecnologias e Saúde do Campus Araranguá da Universidade Federal de Santa Catarina para a obtenção do título de Bacharela em Engenharia de Computação.
Orientadora: Profa. Analúcia Schiaffino Morales, Dra
Coorientador: Vinícius Rodrigues Zanon, Eng.

Araranguá
2022

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

Manoel, Bianca de Espindola
Análise da viabilidade de uma camada de segurança para dispositivo vestível cardíaco empregando recursos de Internet das Coisas / Bianca de Espindola Manoel ; orientador, Analúcia Schiaffino Morales, coorientador, Vinícius Rodrigues Zanon, 2022.
45 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Araranguá, Graduação em Engenharia de Computação, Araranguá, 2022.

Inclui referências.

1. Engenharia de Computação. 2. Segurança. 3. Dispositivo vestível. 4. Eletrocardiograma. 5. Internet das Coisas Médicas. I. Schiaffino Morales, Analúcia. II. Rodrigues Zanon, Vinícius . III. Universidade Federal de Santa Catarina. Graduação em Engenharia de Computação. IV. Título.

Bianca de Espíndola Manoel

Análise da viabilidade de uma camada de segurança para um dispositivo vestível cardíaco empregando conceitos de Internet das Coisas Médicas

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharela em Engenharia de Computação e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Computação.

Araranguá, 17 de Março de 2022.

Documento assinado digitalmente
ANALUCIA SCHIAFFINO MORALES
Data: 17/03/2022 18:19:23-0300
CPF: 622.256.420-87
Verifique as assinaturas em https://v.ufsc.br

Profa. Analúcia Schiaffino Morales, Dra.
Coordenadora do Curso

Banca Examinadora:

Documento assinado digitalmente
ANALUCIA SCHIAFFINO MORALES
Data: 17/03/2022 18:18:31-0300
CPF: 622.256.420-87
Verifique as assinaturas em https://v.ufsc.br

Profa. Analúcia Schiaffino Morales, Dra
Orientadora

Documento assinado digitalmente
Vinicius Rodrigues Zanon
Data: 17/03/2022 18:07:31-0300
CPF: 052.473.311-25
Verifique as assinaturas em https://v.ufsc.br

Vinícius Rodrigues Zanon, Eng.
Coorientador

Documento assinado digitalmente
Fabricio de Oliveira Ourique
Data: 18/03/2022 05:41:00-0300
CPF: 916.167.860-00
Verifique as assinaturas em https://v.ufsc.br

Prof. Fabrício de Oliveira Ourique, Dr.
Avaliador
Universidade Federal de Santa Catarina

Documento assinado digitalmente
Jim Lau
Data: 17/03/2022 18:02:47-0300
CPF: 613.464.702-00
Verifique as assinaturas em https://v.ufsc.br

Prof. Jim Lau, Dr.
Avaliador
Universidade Federal de Santa Catarina

Documento assinado digitalmente
Marcelo Daniel Berejuck
Data: 17/03/2022 18:38:40-0300
CPF: 640.174.519-34
Verifique as assinaturas em https://v.ufsc.br

Prof. Marcelo Daniel Berejuck, Dr.
Avaliador Suplente
Universidade Federal de Santa Catarina

Análise da viabilidade de uma camada de segurança para um dispositivo vestível cardíaco empregando conceitos de Internet das Coisas Médicas

Bianca de Espindola Manoel* Vinícius Rodrigues Zanon †
Analúcia Schiaffino Morales‡

2022, Março

Resumo

Com o avanço da tecnologia da Internet das Coisas, a aplicação de dispositivos na área médica torna-se promissora. Devido à sua rápida expansão, muitas das arquiteturas propostas, em outras áreas do conhecimento, apresentam sérias falhas de segurança quando aplicadas à saúde, pois lidam com dados sensíveis de pessoas, o que as torna inadequadas na área médica. Este artigo tem como objetivo propor uma camada de segurança para um dispositivo cardíaco vestível que realiza o exame de eletrocardiograma remotamente. Para torná-lo menos suscetível aos principais ataques conhecidos, como espionagem e Man-in-the-Middle, métodos de criptografia foram utilizados para analisar sua viabilidade. Três métodos de criptografia (AES-CBC, SPECK e CLEFIA) foram comparados em um ambiente de comunicação seguro e autenticado, a fim de analisar o desempenho dos algoritmos quando submetidos à latência da rede e testes de carga. A verificação consistiu em mostrar o desempenho dos métodos de criptografia na arquitetura de rede proposta por meio de análise gráfica. O algoritmo de criptografia AES-CBC provou ser a melhor opção para a camada de segurança do dispositivo vestível. Assim, conclui-se que a inserção de uma camada de segurança baseada em criptografia é viável para o aprimoramento da troca de informações em dispositivos vestíveis cardíacos.

Palavras-chaves: Segurança. Privacidade dos dados. Dispositivo vestível. Eletrocardiograma. Internet das Coisas Médicas.

*bianca.espindola@grad.ufsc.br
†vinicius.zanon@posgrad.ufsc.br
‡analucia.morales@ufsc.br

The feasibility analysis of a security layer for the cardiac wearable using Internet of Medical Things

Bianca de Espindola Manoel* Vinícius Rodrigues Zanon †
Analúcia Schiaffino Morales‡

2022, Março

Abstract

With the advancement of Internet of Things technology, the application of devices in the medical field becomes promising. Due to its rapid expansion, many of the proposed architectures, in other areas of knowledge, present serious security flaws when applied to health, as they deal with sensitive data of people, which makes them inappropriate in the medical area. This article aims to propose a security layer for a wearable cardiac device that performs the electrocardiogram exam remotely. To make it less susceptible to major known attacks such as spying and Man-in-the-Middle, encryption methods were used to analyze its feasibility. Three encryption methods (AES-CBC, SPECK and CLEFIA) were compared in a secure and authenticated communication environment in order to analyze the performance of the algorithms when subjected to network latency and load tests. The verification consisted of showing the performance of the encryption methods in the proposed network architecture through graphical analysis. The AES-CBC encryption algorithm proved to be the best choice for the wearable security layer. Thus, it is concluded that the insertion of a security layer based on encryption is viable for the improvement of information exchange in cardiac wearable devices.

Key-words: Security. Data privacy. Wearable device. Electrocardiogram. Internet of Medical Things.

*bianca.espindola@grad.ufsc.br
†vinicius.zanon@posgrad.ufsc.br
‡analucia.morales@ufsc.br

1 Introdução

Com o surgimento da pandemia de COVID-19, em 2020, a área de saúde se tornou o centro das atenções em uma esfera mundial. Os sistemas de saúde estão incorporando tecnologias e o uso de dispositivos eletrônicos para realizar tratamentos mais precisos. As monitorações através de dispositivos tecnológicos permitem realizar diagnósticos personalizados, e consequentemente melhoram o tratamento e os cuidados com os pacientes (NASAJPOUR et al., 2020). Estes recursos tecnológicos fazem parte do ecossistema da Internet das Coisas, ou Internet of Things (IoT), em inglês. Trata-se de um novo paradigma que incorpora dispositivos móveis, tecnologias de comunicação, sistemas pervasivos e armazenamento em nuvem e tem sido amplamente pesquisado para a área da saúde e bem-estar (ISLAM et al., 2015).As aplicações mais comuns se estendem nos campos de monitoramento contínuo de doenças crônicas (VISHNU; RAMSON; JEGAN, 2020), telemedicina (YU; ZHOU, 2021), acompanhamento da saúde mental de pacientes (MANIYATH et al., 2021) e aprimoramento de exames médicos (MOHANRAJ et al., 2020).

No entanto, muitos dispositivos ainda não têm mecanismos de segurança implementados em camadas mais físicas, como é o caso de vestíveis que medem glicose no sangue, conhecidas como bombas de insulina sem fio que são amplamente utilizadas no cuidado médico (HEI et al., 2015).Como a comunicação sem fio dos sistemas invadidos não era criptografada os invasores puderam facilmente comprometer as informações do sistema em duas categorias de ataque. Ataques injeção de sobredoses de insulina no dispositivo sem fio e sobredose aguda única pode causar diversas complicações ao paciente podendo levá-lo à morte.

Em 2017 a FDA (Food and Drug Administration) órgão regulatório do governo dos Estados Unidos, emitiu uma nota sobre um marcapasso cardíaco que possuía inúmeras vulnerabilidades de segurança. Sendo necessária a visita dos cerca de 65.000 pacientes a seus médicos para estes poderem atualizar o firmware do dispositivo (STELLIOS et al., 2018). O artigo ainda cita ataques de diversas naturezas a dispositivos vestíveis na saúde. Destaca-se o ataque a rede de monitoramento domiciliar de pacientes. Onde é extraído o firmware dos dispositivos conectados ao corpo do paciente através da rede e realizado uma engenharia reversa e assim encontrar vulnerabilidades.

Com o avanço das novas tecnologias, praticamente, todas as instituições estão sujeitas aos ataques de segurança e privacidade de dados. Quando o assunto são dados médicos, a preocupação aumenta, e estas instituições podem ser alvos de diversas categorias de ataques, dentre eles o de injeção de dados. Neste caso, o atacante fornece informações falsas sobre usuários, de forma que pareçam reais, e como consequência, o impacto pode ser fatal. Esta categoria de ataque pode comprometer a integridade física e mental do indivíduo (BENDECHACHE; LE-KHAC; KECHADI, 2017). Isto pode ocorrer na camada de rede do dispositivo, através de um ataque man-in-the-middle (MITM), interceptando as informações verdadeiras do dispositivo antes de chegar na chamada de rede e as substituindo por dados falsos pelos atacantes. Este tipo de problema pode ser evitado através de implementações de políticas de segurança, que bloqueiem ou ignorem solicitações não autenticadas por meio de algoritmos criptográficos e mecanismos de troca de chaves (FIROUZI et al., 2018).

De acordo com o documento publicado em 2020, pela Sociedade Brasileira de Cardiologia (SBC), as cardiopatias lideram as causas de mortes desde 1960 em todo o território nacional (OLIVEIRA et al., 2020). De acordo com o site da SBC, são mais 1100 mortes diárias, cerca de uma morte a cada 90 segundos, somente em janeiro de 2022, foram registradas 27.227 mortes por doenças cardiovasculares no Brasil. Estes dados ajudam a

compreender o interesse de pesquisadores de novos dispositivos na área da cardiologia. Estas tecnologias costumam ser aplicadas na prática clínica cardiovascular visando promover melhores diagnósticos e tratamento mais precisos de doenças. Segundo Pevnick et al. (2018), no caso do monitoramento de saúde cardíaca através do eletrocardiograma (ECG), exame tradicionalmente realizado com inúmeros eletrodos adesivos colados na pele do paciente, ligados por fios ao sistema de captura de sinais promove ao paciente grande desconforto, podendo alterar o resultado do mesmo. Outros problemas gerados como a desconexão acidental dos fios durante o exame, eletrodos mal posicionados no corpo do paciente conduzem o ECG ser alvo de pesquisas de dispositivos vestíveis como potencial solução a estas questões (ARQUILLA; WEBB; ANDERSON, 2020).

Devido à sua natureza de informações confidenciais dos pacientes, garantir a segurança é uma questão fundamental no desenvolvimento de sistemas utilizando recursos de IoT e outras tecnologias. Como, por exemplo, tem sido pouco divulgados resultados relacionados à segurança na infraestrutura de redes onde os dados dos pacientes são transmitidos. Para que uma proposta de melhoria no ECG seja de fato usada no âmbito clinico, o sistema deve garantir que todas as informações geradas no exame, desde a aquisição, transmissão, armazenamento ou na associação aos dados pessoais do paciente correspondam aos critérios mínimos de segurança em todas as camadas da aplicação. Segundo Ghubaish et al. (2021) existem seis requerimentos de segurança a serem cumpridos, dos quais se destacam: confidencialidade, integridade, autenticação e disponibilidade.

O presente artigo tem por objetivo apresentar um mecanismo de segurança para o dispositivo com interface vestível para a aquisição, processamento e transmissão do sinal cardíaco desenvolvido em (ZANON et al., 2021). Além disso, deverá ser avaliada a influência do mecanismo proposto em um cenário que simule a coleta de dados de um exame de ECG. A análise do desempenho retratará possibilidades de implementar a solução em exames de tempo real e conectados à Internet, implementando alguns recursos de segurança para os dispositivos vestíveis sem comprometer a execução do exame.

A camada consiste em funções de autenticação empregando certificados de segurança. A estratégia empregada é garantir a troca de informações de forma segura validando-se de conceitos de autenticação por chaves criptográficas. Ao serem enviados ao servidor de aplicação, os dados da amostragem do sinal cardíaco passam por um processo de criptografia leve. Criando deste modo, uma camada de segurança entre o envido dos dados pelo dispositivo cardíaco e o recebimento pelo servidor. Essa comunicação será realizada apenas quando o servidor autenticar o dispositivo usado e criar uma espécie de túnel de comunicação segura na rede. Após os dados chegarem ao servidor, os mesmos serão descriptografados e exibidos na página do sistema em tempo real. Foram comparados três algoritmos diferentes: AES-CBC 3.5.1, SPECK 3.5.2 e CLEFIA 3.5.4 associados aos testes para avaliar o desempenho e os impactos gerados pelos mesmos sobre a realização do exame.

O artigo está estruturado em cinco seções. Os aspectos teóricos relacionados aos mecanismos de IoT e segurança, bem como os algoritmos empregados são discutidos na Seção 2. A Seção 3 apresenta o estado da arte descrevendo alguns trabalhos relevantes investigados na literatura científica nos últimos anos. A Seção 4 descreve o desenvolvimento do trabalho, os resultados obtidos e as discussões. Finalmente, seguem as considerações finais, com a conclusão e trabalhos futuros e na sequência, as devidas referências bibliográficas.

2 Internet das Coisas na Saúde

2.1 Internet das Coisas (IoT)

Considerado por Firouzi, Chakrabarty e Nassif (2020) o próximo estado da Internet, a Internet das Coisas (Internet of Things - IoT) abrange uma vasta quantidade de conexões entre dispositivos que podem interagir entre si e o meio em que estão inseridos. Compartilhando dados coletados em diferentes tipos de redes, eles podem ser processados, armazenados e auxiliar no processo de tomadas de decisão. Os autores apresentam definições, das quais se destacam a da palavra "coisas" no âmbito de sistemas IoT. São objetos inteligentes que podem detectar, atuar e interagir com outros objetos, sistemas ou pessoas. Para um dispositivo ser considerado IoT, ele necessita de uma unidade de processamento, ser um sensor ou atuador, estar conectado a uma rede e possuir uma forma de identificação única, ou seja, um tipo de endereçamento. Os dados brutos coletados por sensores devem ser tratados, formatados e enviados para análise, que ocorre muitas vezes na borda da rede, perto dos dispositivos ou camadas mais a cima. Por exemplo, a nuvem onde o sistema está hospedado ou uma camada intermediária (IQBAL et al., 2021) como tem sido proposto em diversas arquiteturas, seriam as camadas de *fog* ou *edge computing*.

Na IoT serão conectadas pessoas, dados e dispositivos com capacidade de sincronizar e acumular dados. O reporte anual da Cisco e Internet (2020) prevê que nos próximos anos, o número de dispositivos conectados às redes IPs seja três vezes a população global. Poderá haver 3,6 dispositivos em rede *per capita*, acima dos 2,4 dispositivos em rede *per capita* em 2018. Cerca de 29,3 bilhões de dispositivos em dois anos, acima dos 18,4 bilhões em 2018. O impacto que dispositivos inseridos em nosso cotidiano terão nos próximos anos pode melhorar os processos na indústria, economizando tempo, dinheiro e energia. A qualidade de vida poderá aumentar com aplicações de sistemas mais eficientes. O rápido crescimento de dispositivos interconectados no contexto da saúde permitirá melhorar a prestação de serviços e atendimento em saúde, a produtividade das equipes de profissionais e o próprio gerenciamento dos sistemas hospitalares.

O ecossistema da IoT compreende então, dispositivos desde a aquisição de dados, mecanismos de comunicação e seus diferentes protocolos, plataformas para a integração das aplicações e a computação distribuída e em nuvem, juntamente com o processamento de um grande volume de dados. Não existe ainda, uma padronização internacional para a arquitetura deste novo paradigma, mas alguns autores evidenciam que a infraestrutura necessária pode variar entre três a quatro camadas das quais, destacam-se: camada de coisas, uma camada de comunicação e a camada de distribuição e sincronização das informações (MORALES; OURIQUE; CAZELLA, 2021).

2.2 Internet das Coisas Médicas – IoMT

Apontada como uma área prioritária, a Internet das Coisas Médicas, ou Internet of Medical Things (IoMT), em inglês, tem sido evidenciada pela literatura como uma alternativa para reduzir os custos e melhorar a saúde e bem-estar das pessoas. O vasto campo de implementações de IoT na saúde está criando inúmeras novas possibilidades na prestação de serviços de saúde para pacientes de médicos em todo o mundo, melhorando o acesso a ele. O monitoramento contínuo da saúde de crianças e idosos tornou-se mais fácil com o uso de dispositivos não invasivos, sem fio, que podem medir dados do usuário como pressão arterial, glicose, contagem de passos, etc. por profissionais da saúde e outros familiares, sem que os mesmo se sintam incomodados e removam os dispositivos (BHATIA;

PANDA; NAGPAL, 2020).

Devido a vários desafios práticos, a área de saúde tornou-se o principal ponto de discussão para abordagem dos vários aspectos da IoT nos últimos anos. Resultando em um esforço maior para desenvolver plataformas exclusivas atendendo as especificações da área, tanto no nível de *hardware* quanto no nível de *software* (GANDHI; GHOSAL, 2018). Esta visão está levando a uma diferenciação entre dispositivos IoT e IoMT, como habilitação de protocolos de comunicação existente apenas para a transferência de dados na saúde. As questões de segurança e privacidade de dados, que ainda são consideradas lacunas existentes dentro do IoMT. O aumento da procura por sistemas de tempo real com alta disponibilidade. A crescente demanda por dispositivos de monitoramento fisiológico continuo para controle de atividades físicas (QADRI et al., 2020). Essas diferenças estão apresentadas na Tabela 1.

Tabela 1 – Principais diferenças entre IoT e IoMT.

	IoT Genérico	IoMT
Implantação	Em uma grande área geográfica ou ambientes internos que serve um único proposito.	Ambientes pequenos fechados, ao redor do corpo humano ou em um centro de saúde.
Alimentação	Baterias, placas solares ou gerador eólico.	Extraem energia do meio, corpo humano e/ou movimento.
Monitoramento	Ambientes, usado em aplicações industriais e comerciais.	Pessoas e sinais vitais.
Tamanho	Desejável nodos pequenos mas, tamanhos variam de acordo com aplicação.	Nodos são miniaturizados e discretos.
Mobilidade	Podem ser estacionários ou móveis.	Móveis e associados ao corpo humano.
Integridade dos Dados	Tenta-se manter a integridade dos dados, Redundância compensa os eventuais erros	Os dados devem ser preservados, transmitidos com essencial integridade.
Complexidade de Implantação	Geralmente é simples.	Mais difícil, principalmente em implantes.

Nos últimos anos o investimento em IoMT vem aumentando por conta do crescimento da população de idade avançada. A combinação entre dispositivos inteligentes e assistência médica pode contribuir muito com a qualidade de vida da população em geral. O principal passo a ser dado em direção a um sistema de saúde inteligente é a combinação de dados coletados nos sensores com as informações já existentes no sistema de saúde para que doenças crônicas ou epidêmicas sejam monitoradas e prevenidas (AL-TURJMAN; NAWAZ; ULUSAR, 2020). Em sistemas de monitoramento remoto com grande tráfego de dados sensíveis, é necessário um amplo investimento em dispositivos com mecanismos de segurança. O vazamento destes dados e roubo de informações são problemas sérios, quando não há garantia de segurança a implementação real de tais sistemas fica comprometida.

Conforme Qadri et al. (2020) mostra que este novo paradigma modifica todas as áreas médicas e suas interfaces homem-maquina, com a evolução dos dispositivos de monitoramento de saúde, com melhor conectividade das infraestruturas IoT, levou ao

desenvolvimento de sistemas *healthcare-oriented* para o monitoramento de saúde e sinais vitais de pacientes, melhorando diagnóstico e qualidade do cuidado médico oferecido ao usuário. A saúde médica baseada em IoT supera os erros humanos e ajuda o médico a diagnosticar mais rapidamente e com maior precisão doenças de pacientes. Plataformas inteligentes associam sensores e circuitos eletrônicos ligados no corpo do paciente com análise dos dados através da rede visam diminuir o tempo de resposta humana em emergências dentro e fora dos hospitais (VISHNU; RAMSON; JEGAN, 2020).

2.3 Arquitetura IoMT

As tecnologias de IoMT, *Big Data*, inteligência artificial e dispositivos vestíveis são promissoras e poderão revolucionar a forma tradicional como a saúde é tratada. O modelo convencional tem uma abordagem curativa, onde o paciente apresenta uma doença, o médico efetua o diagnóstico e então se inicia o tratamento. Com essas tecnologias, um novo modelo de saúde focado em prevenção e predição é possível. Diferentes protocolos de comunicação e arquiteturas podem ser usados para a implementação deste novo modelo. Segundo os autores de Zhu et al. (2019) para ser possível a implementação na saúde, importantes recursos devem ser alcançados e eles são resumidos no que definem como "7P" (Figura 1).

Figura 1 – 7P do *saúde inteligente*

Fonte: Zhu et al. (2019)

Personalizada: O sistema de saúde inteligente deve ser capaz de fornecer um plano de tratamento único e adequado baseado nas condições de fisiológicas de cada indivíduo.

Persuasiva: O sistema de saúde inteligente pode também influenciar o usuário indiretamente por várias técnicas de pressuração. A ideia principal é mudar os hábitos do usuário para melhorar o gerenciamento de sua saúde.

Preditiva: A saúde inteligente possibilita uma categoria de manutenção preditiva em humanos. Com um sensor captura o eletrocardiograma (ECG) e transmite os dados para o sistema do hospital, um sistema pode detectar se o paciente está prestes a sofrer um ataque cardíaco. Neste caso, uma ambulância poderia ser chamada para a sua residência e o hospital pode alertar um cardiologista disponível.

Participativo: o sistema representa um novo paradigma de tratamento participativo, transformando a maneira com que pacientes se conectam, comunicam, compartilham informações de saúde, descobrem e acessam novas opções de tratamento. Tornando o paciente um agente ativo em seu tratamento.

Preventiva: O sistema deve fornecer soluções que ajudam pessoas a levarem um estilo de vida mais saudável e buscar tratamento apenas quando for necessário.

Perpétuo: Um sistema de conscientização perpétuo de saúde inteligente de maneira que haja monitoramento contínuo e detecção onipresente. É essencial para aplicações de segurança e alta criticidade na saúde pública, como vida assistida e assistência medica emergencial.

Programável: Os sistemas de saúde inteligente deve permitir usuários configurar programas quando estão lidando com casos médicos sofisticados.

Os "7P"requeridos abrem várias oportunidades no domínio de negócios e pesquisa. Como não há uma arquitetura definida para IoMT, é possível encontrar diversas propostas de fluxo de informações com números diferentes de camadas. Devido à coleta de dados de pacientes em tempo real e troca de dados sensíveis, outro desafio é elaborar sistemas com um alto nível de segurança e confiabilidade. Os autores Bhatia, Panda e Nagpal (2020) apresentam um fluxo de informações estendido em cinco níveis que são comuns a diversas propostas. São eles a camada de Sensoriamento, Transporte, Processamento, Aplicação e Negócios, como ilustrado pela Figura 2.

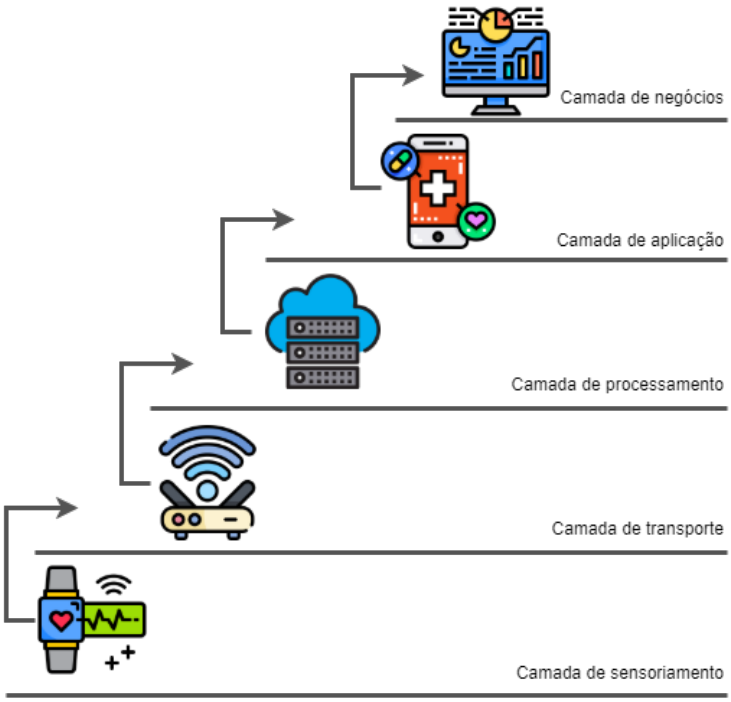
Até o momento não existe uma padronização para o número de camadas de uma arquitetura de IoMT, no entando algumas camadas são comuns a alguns autores, e que são mencionadas a seguir(ZHU et al., 2019), (OMONIWA et al., 2019), (BHATIA; PANDA; NAGPAL, 2020):

- Camada de Sensoriamento: Também chamada camada de percepção, consiste na camada de borda, onde é encontrada as “coisas” IoT. Essa camada consiste em sensores que detectam e coletam dados do usuário. Pode haver vários sensores que coletam diferentes conjuntos de dados, ou atuadores agindo em diversos aspectos. Por exemplo, sensores vestíveis coletando temperatura e sinal cardíaco de pacientes.

- Camada de Transporte: Também chamada camada de rede é usada para a transmissão sem fio dos dados da camada anterior. Dependendo da arquitetura, suporta diversos protocolos de comunicação, como 5G, LTE, bluetooth, WiFi etc.

- Camada de Processamento: A camada de processamento ou serviço, atuada como um *middleware*. Permite que os usuários possam ter vários sensores com funções distintas. Executa o processamento dos dados do usuário, podendo realizar a descriptografia de mensagens, detectar dados preocupantes e emitir avisos a camadas superiores.

Figura 2 – Modelo 5 camadas IoMT



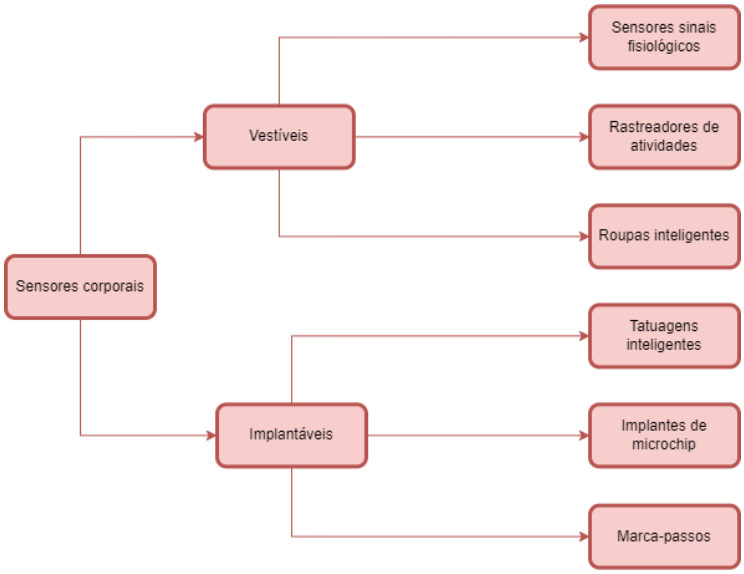
Fonte: Adaptada de Bhatia, Panda e Nagpal (2020)

- Camada de Aplicação: Fornece suporte baseado nas funcionalidades do usuário final do sistema, sendo o médico ou o próprio paciente. No modelo de 5 camadas, ela é responsável por instruir a camada de processamento. Está camada pode permitir uma interatividade através dos protocolos e serviços da Web, auxiliando na assistência médica inteligente.
- Camada de Negócios: A camada de negócios lida com todo o sistema IoT, que inclui aplicativos, modelos de negócios e informações confidenciais dos usuários. O grande desafio para a implementação destes sistemas está nessa camada, devido à segurança e privacidade das informações. Provedores de saúde que desejam usar o compartilhamento de dados dos usuários para outros fins devem deixar claro no modelo de negócio da aplicação essa intenção.

2.4 Dispositivos vestíveis

Na camada de sensoriamento podem ser encontrados diversos componentes eletrônicos capazes de converter informações físicas em impulsos elétricos. São empregados na detecção de vários parâmetros, tais como, temperatura corporal, batimentos cardíacos, suor na pele, níveis de glicose, pressão arterial (BHATIA; PANDA; NAGPAL, 2020). Com capacidade sem fio, são incorporados em *gadgets*, acessórios ou roupa, sensores dos mais diversos tipos permitem o monitoramento de dados fisiológicos de pacientes. Estes sensores podem ser dispositivos vestíveis, ou seja, usados sob o corpo humano ou implantáveis conforme a Figura 3. Podem ser invasivos ou não invasivos, podem ler sinais constantemente ou apenas durante um período de tempo (DIWAKER; JANGRA; RANI,).

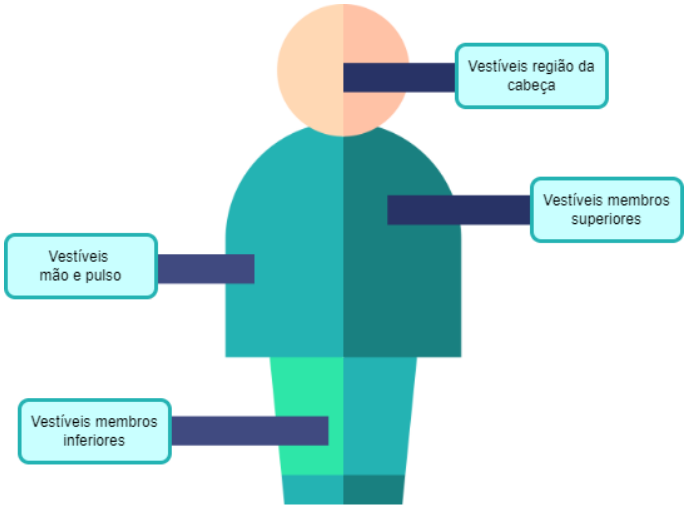
Figura 3 – Sensores corporais



Fonte: Própria autora

Em Ometov et al. (2021) é apresentada uma divisão destes sensores em quatro grupos: região da cabeça, membros superiores, mão ou pulso e membros inferiores, na Figura 4 é possível verificar que os dispositivos são classificados conforme a região do corpo do usuário em que estes vestíveis estão localizados.

Figura 4 – 4 categorias de vestíveis



Fonte: Própria autora

Além da classificação de posicionamento, os autores também apresentam definições sobre a energia consumida pelos dispositivos (OMETOV et al., 2021). Dispositivos que possuem display normalmente consomem mais bateria que dispositivos que não o possuem.

Porém, isso depende da natureza da aplicação e quanto de processamento de dados eles realizam. Desta forma, eles podem ser classificados em dispositivos vestíveis de baixa, média e alta potência.

- **Dispositivo de baixa potência**, possuem capacidades limitadas de processamento e sensoriamento, e precisam operar por um tempo maior, com foco único de sensoriamento e aquisição de dados, incluem dispositivos pequenos como um anel inteligente ou uma pulseira, dispositivos com uma pequena bateria, radio e poucos sensores.
- **Dispositivo de média potência**, começam a incluir pequenos *displays* para que seja possível interagir e ver de maneira mais prática o seu estado, podem ter múltiplos sensores a bordo, e podem ter maneira direta ou indireta conexão com a internet, neste conjunto estão *smart watches* por exemplo.
- **Dispositivo de alta potência**, categoria que possui alto poder de processamento, geralmente não funcionam a bateria, possuem grandes *displays* e comportam altas taxas de dados, podem aplicar técnicas de *machine learning*, dispositivos desta classe estão inseridos, óculos de realidade virtual, cameras térmicas com sensoriamento processado.

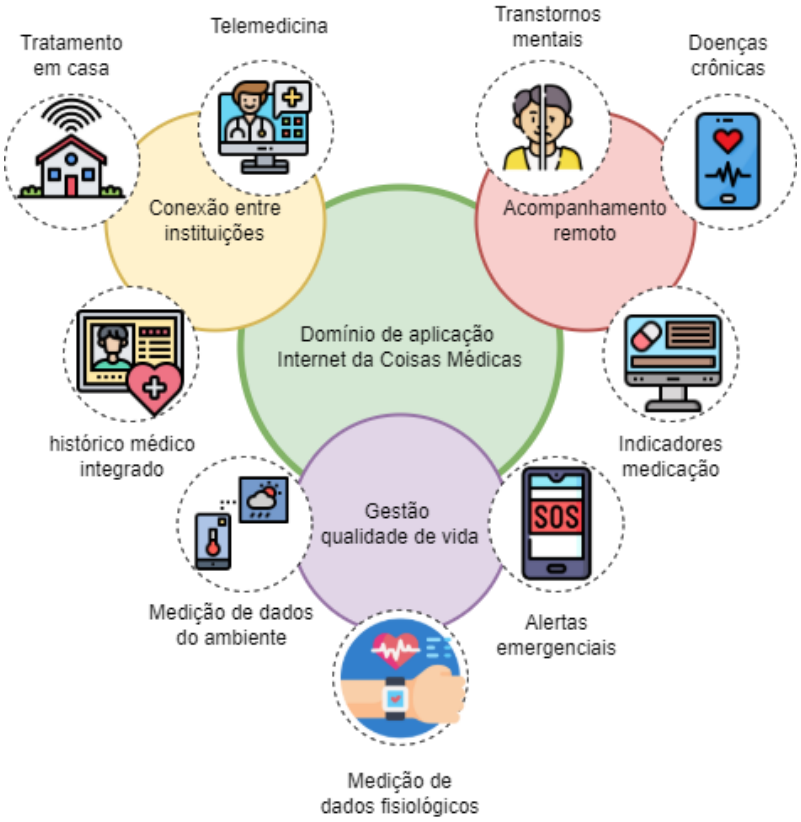
2.5 Aplicações IoMT

Na área da saúde e das ciências médicas, algumas das principais aplicações de IoMT incluem o avanço tecnológico no acompanhamento remoto de doenças, medicamentos e equipamentos. No gerenciamento de qualidade de vida através de dados da saúde, além da infraestrutura que permite conexão entre instituições para o controle da saúde móvel (MISHRA; RASOOL, 2019). Nestes domínios, conforme figura 5, podem ser apresentados diversas aplicações, tais como: telemedicina, tratamento em casa, histórico de médico integrado, acompanhamento de transtornos mentais, acompanhamento de doenças crônicas, etc.

A seguir serão destacadas algumas destas aplicações, devido o enfrentamento da pandemia, que impulsionou o desenvolvimento de tecnologias aplicadas à saúde nos últimos dois anos:

- **Tratamento em casa**: A pandemia de COVID-19 trouxe muitas mudanças de paradigma para a IoMT e acelerou muitos processos. Um deles foi a necessidade de fazer o tratamento de pacientes de forma remota. Uma proposta de monitoramento domiciliar de pacientes com risco cardiovascular e pulmonar é apresentado em (SHAJI et al., 2021). Onde o paciente faz o uso de um dispositivo vestível conectado a um *smartphone* que analisa o ECG, extrai os parâmetros cardíacos e os exibe em um painel de monitoramento remoto.
- **Acompanhamento de transtornos mentais**: A avaliação e intervenção em saúde mental está se estendendo além das clínicas e consultórios para a vida cotidiana. Através do monitoramento frequente do bem-estar por longos períodos seria possível melhor avaliar crises e necessidades de intervenção. Para tal finalidade Maniyath et al. (2021) faz o uso de sensores vestíveis combinados com imagens de expressões faciais para detectar a emoção de um paciente. Essa nova abordagem pode além de ajudar o paciente e tutores entenderem quais os motivos que desencadeiam as crises, pode reunir informações relevantes sobre a saúde mental dos mesmos. Além de

Figura 5 – Exemplos de campos de aplicação IoMT.



Fonte: Própria autora

manter o registro diário de suas emoções, estes dados podem ser enviados a sistemas maiores para melhor avaliação e compreensão medica. Diminuindo desta forma o estresse gerado pela necessidade do paciente sair de sua rotina para ir a uma clínica.

- **Acompanhamento de doenças crônicas**: idosos têm maior probabilidade de adquirir doenças relacionadas à memória, gerando um grande impacto na qualidade de vida desta população. Com a expansão de tecnologias de RFID na IoT, tornou-se viável localizar pessoas de maneira prática e econômica. Uma forma de melhorar a segurança destas pessoas em áreas urbanas é apresentada em (RAAD; DERICHE; KANOUN, 2021). O artigo propõe um sistema de rastreamento através de pulseiras ou tornozeleiras vestíveis que portaria RFIDs passivos para a localização de idosos andando pela casa. Em caso de risco, ou seja, uma baixa proximidade alertar os cuidadores através de um aplicativo para smartphone.

- **Indicadores de medicação**: pacientes que sofrem com problemas diabéticos precisam verificar seu nível de glicose com frequência para adequar sua dieta conforme o resultado. Os dispositivos atuais disponíveis no mercado dependem da coleta e intervenção manual. Sistemas que permitem o controle em tempo real e intervenção automática diminuiriam erros humanos além de melhorar a qualidade de vida destas pessoas. O trabalho de Vishnu, Ramson e Jegan (2020) faz um levantamento de

sistemas cuja finalidade é além de monitorar a medicação, levantar dados para análise médica que melhorem o tratamento do paciente.

- **Vestíveis no combate ao COVID-19**: o avanço da COVID19, também influenciou o uso da tecnologia para combater a pandemia, os autores de Ates et al. (2021) apresentam a tecnologia de sensores vestíveis para ajudar na detecção e combate ao vírus. A pandemia de COVID-19 destacou o potencial dos dispositivos eletrônicos vestíveis na área da saúde. Com inovação e desenvolvimento contínuos, a próxima geração de sensores vestíveis (e sua capacidade de monitorar continuamente parâmetros físicos e marcadores bioquímicos) pode desempenhar um papel fundamental no combate à próxima pandemia. Para complementar, uma revisão sistemática de dispositivos vestíveis apresentada por Quer et al. (2021) discute o uso de inteligência artificial nos dados provenientes dos dispositivos de IoMT, como o uso de dados de termômetros, oxímetros e batimentos cardíacos, podem em conjunto evidenciar como as variantes do vírus se comportam. Essas informações, poderão auxiliar na detecção do vírus, aprimorando o processo de diagnóstico da doença.

3 Segurança em IoMT e dispositivos vestíveis

A cibersegurança tem se tornado uma questão estratégica para a implementação dessas novas tecnologias no meio da saúde (HATZIVASILIS et al., 2019). Além disso, a área da saúde é um constante alvo de busca e exploração de vulnerabilidades por conta da natureza de seus dispositivos e categoria de dados que circulam na rede. Os dados ligados aos serviços de saúde, são principalmente dados confidenciais como nomes, endereços, histórico médico e problemas de saúde. Garantir a segurança e privacidade dessas informações são os principais desafios não resolvidos devido ao tamanho, a sensibilidade e o alto nível de segurança exigido pelas aplicações (PERWEJ et al., 2022). Como os benefícios do uso do IoMT são inegáveis, a resolução destes desafios são fatores importantes para o uso dessas tecnologias pelo setor da saúde e seus pacientes.

3.1 Premissas de segurança

No contexto de IoMT, Papaioannou et al. (2020) são apontadas as principais premissas de segurança, muito similares aos pontos de segurança de sistemas mas com a limitação de recursos devido a natureza das redes de sensores e dispositivos empregados:

- Confidencialidade: Garante que as informações não são divulgadas a usuários, processos ou dispositivos, que não tenham sido autorizados a acessar as informações. As informações médicas do paciente não devem ser compartilhadas a terceiros de não autorizados. Caso a confidencialidade dos dados não for respeitada, criminosos podem interferir entre o receptor e remetente efetuando a captura deses dados. Há várias maneiras de garantir que as informações sejam confidencias, desde o uso de proteção física dos equipamentos até o uso de algoritmos criptográficos para que os dados se tornem ilegíveis fora do contexto do sistema.

- Integridade: É a garantia que as informações não foram modificadas ou destruídas de forma não autorizada. Na área da saúde a qualidade dos dados depende da integridade dos mesmos. Os dispositivos que operam coletando dados fisiológicos de pacientes precisam ser seguros, pois, estão sujeitos a ataques físicos que visam o comprometimento das informações. Manter a integridade dos dados é uma parte

significativa que muitas aplicações acabam negligenciando. Transmitir dados sensíveis de pacientes remotamente através da Internet traz potenciais perigos que precisam ser considerados (SABA et al., 2020). Se há alguma falha na arquitetura do sistema, o atacante realiza um ataque *man in the middle*. Pode efetk,mluar o uso *malwares* ou até com acesso físico ao dispositivo. Realizando assim ataques de injeção de código ou *firmware* malicioso.

- Não Repúdio: Impede que uma entidade consiga negar a realização de determinada ação em uma interação anterior ou negue compromissos. É garantir a imutabilidade dos dados do tratamento médico e a certificação de não repudio dos dados (N et al., 2021). Um dispositivo que coletou o eletrocardiograma de um paciente pode enviar os dados e posteriormente alegar que não foi a origem desses dados. Por isso a imutabilidade e o não repudio devem ser garantidos em todas as transações de dados no fluxo dispositivo-servidor.

- Autenticação: Processo ao qual uma entidade comunicante é assegura da identidade alegada por outra parte envolvida na comunicação. Quando aplicada a mensagens a autenticação é o processo pelo qual e entidade que emitiu a mensagem é verificada como a fonte. Permite que aplicações finais com usuários e dispositivos vestíveis realizem autenticação mútua e troquem chaves de sessão (WANG et al., 2021).

- Autorização: É a permissão concedida a outra entidade para que está possa obter acesso a determinados serviços na aplicação. Diferentes níveis de usuários e permissões que cada camada na arquitetura de rede tem ao acessar dados deve ser permitida conforme a necessidade, localização, categoria de acesso. Devido à arquitetura do sistema e a sensibilidade dos dados podem ser projetados duas estratégias de gerenciamento de identidades. Uma distribuída onde todas as entidades envolvidas devem entrar em consenso. A outra centralizada, que pode ser controlada facilmente, mas a torna um alvo de ataques (TRNKA et al., 2022).

- Disponibilidade: Garante que os sistemas não neguem serviços a usuários autorizados. Os dados devem estar sempre disponíveis para serem utilizados quando solicitados por uma entidade legitima e autorizada. Quando um paciente necessitar de uma intervenção sem interrupções, ataques do estilo "negação de serviço" podem custar sua vida (PAPAIOANNOU et al., 2020).

3.2 LGPD e dados sensíveis

A Lei n.º 13.706/2018 também conhecida como Lei Geral de Proteção de Dados (LGPD) é voltada a proteção de dados pessoais, que exige adequação de organizações de diversas áreas. Na LGPD o titular, ou seja, a pessoa física a qual os dados pessoais se referem tem direitos de proteção sobre os mesmos, tal como o respeito a privacidade (MONTEIRO et al., 2021). Segundo a lei, dados pessoais sensíveis são aqueles que podem gerar discriminação ou identificação do titular. São os dados que se referem, por exemplo, à orientação sexual, convicções, filiação a sindicato e saúde. Por este motivo as informações na IoMT precisam de uma atenção especial. O vazamento dessas informações pode gerar prejuízos gigantescos ao titular. O compartilhamento, processamento e armazenamentos de dados sensíveis em sistemas de saúde devem estar em adequação com a lei vigente desde agosto de 2021, gerando implicações legais as instituições caso ocorra falha na proteção e privacidade dos mesmos.

3.3 Desafios de segurança nos dispositivos vestíveis

Devido à falta de um padrão uniforme de arquitetura a ser seguido, a diversidade de dispositivos e muitas vezes a dependência de equipamentos de terceiros como *smartphones*, os dados gerados por dispositivos vestíveis podem ser facilmente interceptados e adulterados (JIANG; SHI, 2021b). Conforme apresentado em Jiang e Shi (2021a), os desafios de segurança de vestíveis em IoMT, podem ser relacionados em três aspectos: segurança tecnológica, gerenciamento de dados, leis e regulamentações:

- **Segurança tecnológica:** encontrar formar de evitar que o paciente consiga interromper a coleta. Promover o controle de acesso de dados a apenas usuários específicos. No processo de transmissão de dados realizar o uso mecanismos além do endereço MAC para identificação do dispositivo. Neste caso o atacante pode facilmente o simular em uma tentativa de ataque. Utilizar formato de dados simples (por exemplo, JSON) sem nenhuma medida de desfoque de dados criptografados.
- **Gerenciamento de dados:** como ainda não há um padrão estabelecido, no desenvolvimento de dispositivos vestíveis focados em saúde médica há inúmeras maneiras de gerenciar dados. Deve ser dada uma atenção especial a transmissão destes dados, as políticas de privacidade e confidencialidade. Assim como por quanto tempo a LGPD exige que este dado deve ser armazenado e qual a forma de exclusão no caso de quando for solicitado pelo proprietário (MONTEIRO et al., 2021).
- **Leis e regulamentações:** por se tratar de uma indústria emergente a de dispositivos vestíveis ainda não possui muitas políticas e regulamentos sobre a segurança dos dados. Entretanto, com a promulgação da LGPD vazamento de dados, mau gerenciamento ou ainda o não comprimento de algumas das disposições citadas em seus artigos. Poderia ocasionar multas e sanções conforme previsto em lei.

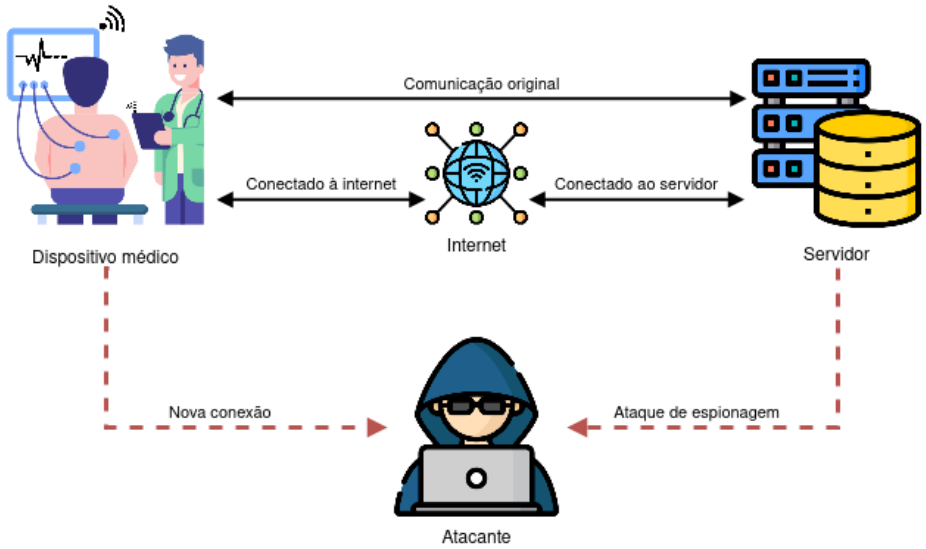
Estes desafios quando não resolvidos criam oportunidades para ataques cibernéticos, visto que os ataques são possíveis consequências de vulnerabilidades já existentes nas implementações. Os autores Strielkina, Kharchenko e Uzun (2018) afirmam que as principais categorias de ataques são destinadas ao controle de dados, aos dispositivos e às redes. O ataque de controle se refere ao invasor ter o controle do sistema, total ou parcial, ou dos dados que nele circulam. Ataque a dispositivos são focados nos nós finais da rede, neste caso, os equipamentos ligados ao corpo do paciente. Quando os ataques são focados na rede, os principais objetivos dos atacantes são a cópia de informações, sequestro de dados e injeção de *malwares*. Em Zakaria et al. (2019) apresentam ataques que atingem os dispositivos e infraestrutura da rede, focados em *gateways* e redes internas vulneráveis. Destacam-se alguns pontos importantes sobre ataques no contexto dos IoMT:

Ataques a *gateways*: Segundo o estudo de Rahmani et al. (2018) que cria uma arquitetura de *Gateway*. Esta arquitetura proposta considera um conjunto de metodologias de segurança e de funcionamento seguro. No entanto, alguns trabalhos mais recentes como o de Meneghello et al. (2019), mostra que questões de acesso ao equipamento físico não são cobertas nas propostas de *gateway* seguro. A importância de segurança ao nível de *hardware* é fundamental visto que um usuário malicioso pode estar fisicamente presente. Nesta situação realizar uma série de ataques, que exploram vulnerabilidades do *hardware* através de suas *interfaces* de comunicação como *USB*, Ethernet e Wi-Fi. Este atacante pode inclusive abrir fisicamente o dispositivo e tentar internamente mais uma abundância de ataques. Este ataque pode ser evitado com técnicas de *boot* seguro e assinado.

Ataques a dispositivos: Em Meneghello et al. (2019), Jiang e Shi (2021b), foram elaborados estudos de ataques nas mais variadas tecnologias. Dentre elas foram citadas, Zigbee, BLE, LoRaWAN e 6LowPAN. O estudo mostra que mesmo os mais modernos e seguros possuem processos de comunicação da camada de enlace, que não são completamente seguros. Podendo ocorrer ataques maliciosos de interceptação de dados usando a técnica *Man-in-the-Middle* ou de negação de serviço que pode drenar a bateria de dispositivos rapidamente. Geralmente para comunicação estes protocolos usam *block ciphers* menos eficientes e menos seguras que serão tratados com maior ênfase neste artigo.

Ataque de espionagem: Ataques de espionagem (do inglês, *eavesdropping attack*) são baseados principalmente na interceptação de informações e os autores Perwej et al. (2022) os classifica em duas categorias, espionagem ativa e passiva. Os atacantes rastreiam o dispositivo alvo e efetuam buscas por vulnerabilidades para os interceptar. Ocorrendo durante a transmissão de dados, como o exemplo da Figura 6, onde os sinais vitais de um paciente são interceptados por um atacante. Que pode usar ilegalmente os dados, comprometendo o exame e muitas vezes a vida do paciente. Para resolver este problema, é recomendado o uso de criptografia. Ao menos que o dispositivo seja de baixa potência, isso pode prejudicar o funcionamento do sistema. O atacante ainda pode adotar uma estratégia além do roubo de dados, o que é chamado ataque de chaves pré-compartilhados. Quando o sistema fornece troca de chaves fixas para validar os dispositivos e estas chaves são armazenadas sem proteção, o invasor pode interceptar a chave, ou então acessar onde estão salvas. Assim ele poderia se passar por um dispositivo salvo comprometendo o exame.

Figura 6 – Ataque de espionagem na IoMT



Fonte: adaptado de Perwej et al. (2022).

Man-in-the-Middle: Um dos ataques de autenticação mais comuns é chamado *Man-in-the-Middle* (MitM). Seu objetivo é controlar e monitorar a comunicação entre duas partes legitimas, enquanto modifica os dados enviados, explorando as vulnerabilidades e recursos limitados dos sensores. Este ataque pode ser passivo, quando o interceptador apenas lê as mensagens entre as entidades, ou ativo quando o invasor consegue realizar a alteração, manipulação e substituição do dispositivo sem que as entidades tenham conhecimento

(PERWEJ et al., 2022). Em um cenário de monitoramento de saúde, onde o atacante obteve sucesso em ultrapassar as medidas existentes na infraestrutura da aplicação. O invasor consegue interceptar e dependendo da criptografia usada, descriptografar os dados trocados entre as entidades. Podendo assim, desabilitar remotamente sensores, ou então impedir que o sistema notifique um alarme quando o paciente necessitar de assistência. Modificar as medições, e no caso de bombas de insulina, provocar uma sobredose, colocando a vida do paciente em risco (SALEM et al., 2022).

3.4 Criptografia

Diferentes categorias de soluções para criptografia estão disponíveis para proteger os dados, algumas podem ser visualizadas na Figura 7. Infelizmente, nem todas são adequadas aos dispositivos de IoMT, devido a sua limitada capacidade de processamento e memória. Soluções criptográficas destinadas ao uso de recursos limitados estão sendo pesquisadas continuamente (DUTTA; GHOSH; BAYOUMI, 2019).

3.4.1 Criptografia de chave assimétrica

Também conhecida como criptografia de chave pública, onde é necessário a criação de uma chave privada e uma chave pública. A chave pública criptografa o dado, mas apenas a chave privada é capaz de descriptografá-la. Recentemente, o foco de criptografias leves virou para este tipo de criptografia, mas ainda não existem disponíveis soluções estáveis e eficientes. Algoritmos assimétricos são muito complexos em termos de operações e não são eficientes no quesito tempo para serem processados. Os mais importantes algoritmos de criptografia assimétrica são segundo Dutta, Ghosh e Bayoumi (2019):

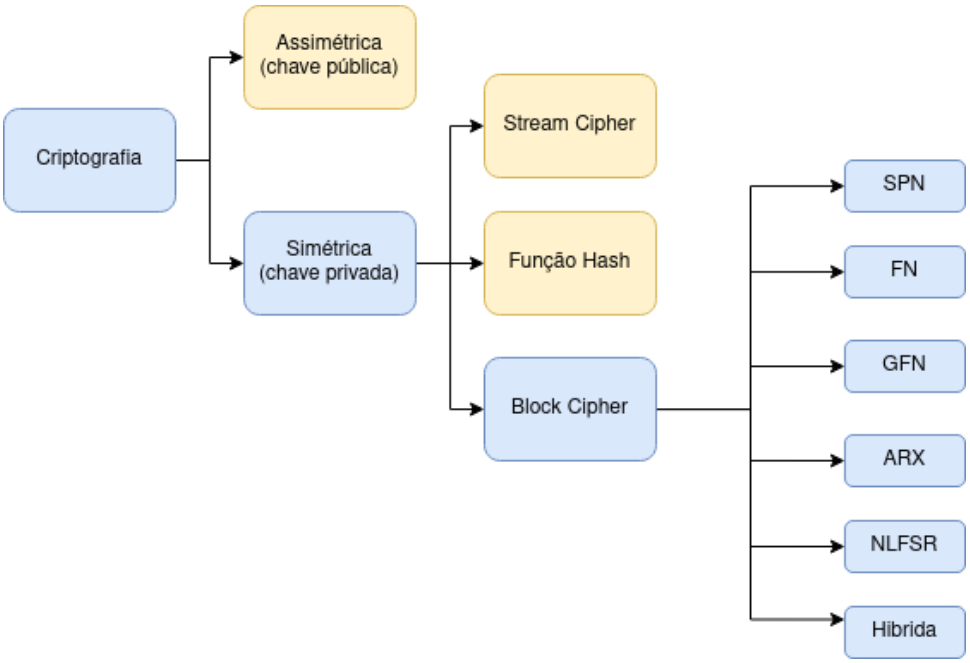
- **Rivest-Shamir Adleman (RSA):** O processo de reverso é muito difícil para um atacante, também é difícil produzir a chave pública por meio da chave privada, por mais que este método seja bem seguro, o processo de gerar chaves é complexo e pesado para um dispositivo com recursos escassos.
- **Diffie-Hellman:** A chave privada é pequena, como resultado o processo é mais rápido, mas devido a esta chave privada ser menor, é vulnerável a ataques *man in the middle*.
- **Digital Signature Algorithm (DSA):** O processamento é mais rápido e que outros algoritmos assimétricos. Porém assinaturas digitais possuem tempo de vida pequeno, e o compartilhamento deste de maneira segura é complexo (CHANDRA et al., 2014).
- **Elliptical Curve Cryptography (ECC):** Apesar de ser mais complexo e difícil de implementar, consome menos energia. Entre todas as técnicas assimétricas, ECC é a favorita para dispositivos com processamento restrito. Utiliza curvas elípticas para processamento, sendo matematicamente mais leves de serem processadas (LARA-NINO; DIAZ-PEREZ; MORALES-SANDOVAL, 2018). A criptografia ECC, está sendo um importante tópico de pesquisa recente em IOT, a maioria do ponto de vista de *software*. Um estudo de Goyal e Sahula (2016) realiza um comparativo entre RSA, ECC, ECDH, concluindo que ECDH é melhor que outros algoritmos em termos de eficiência energética.
- **Elliptic Curve Diffie-Hellman (ECDH):** Paradigma que mistura criptografia ECC com técnicas de trocas de chaves Diffie-Hellman, quando usadas em conjunto

permitem utilizar curvas elípticas mais leves e de maneira geral diminuir o tempo de processamento mantendo o nível de segurança.

3.4.2 Criptografia de chave simétrica

Criptografia de chave simétrica, também conhecida como criptografia de chave secreta. Neste processo, o destinatário e o destino possuem a mesma chave, capaz de criptografar e descriptografar os dados. Criptografia simétrica é mais indicada para aplicações de IOT, devido a suas operações mais rápidas. Tratando-se apenas de operações XOR, *bit-shift* e de permutação, rápido de processar e não necessitando de muitos recursos. Uma distinção importante de algoritmos simétricos está entre os conceitos de *stream* e *block ciphers*. *Stream ciphers* usam uma chave que é do mesmo tamanho do dado, neste processo o texto cifrado é obtido por operações bit a bit no texto. *Blocks ciphers* possuem um tamanho fixo de bits, e diferentes estados de transformação determinados pela chave simétrica. *Block ciphers* são muito versáteis, sendo muito úteis de um ponto de vista IoT. Outra vantagem é que este processo possui métodos idênticos encriptar e decriptar, possibilitando implementações com menos recursos.

Figura 7 – Categorias de criptografia, e onde serão direcionados os estudos.



Fonte: Adaptado de Thakor, Razzaque e Khandaker (2021)

O principal fator de segurança dessa técnica de criptografia é o tamanho da chave utilizada, técnicas quânticas de análise criptográfica podem reduzir a "complexidade" da chave utilizada. Fazendo com que seja possível em tempo hábil, quebrar a criptografia dos dados, por exemplo, o trabalho de Rao et al. (2017), mostra que uma chave simétrica de 256 bits, quando tratada no ambiente quântico equivale a uma chave de 128 bits, de forma que ainda se mantém complexa demorando $5{\times}10^{22}$ anos para ser quebrada. Com isso, é interessante que pelo menos, tenha-se uma chave criptográfica de 192 bits com recomendado de 256 bits ou maiores.

3.4.3 Tipos de *blockcipher*

Para cifrar um texto, por uma chave, existem múltiplas técnicas de difusão e cifragem dos dados. Algumas podem ser citadas a seguir, com base nos estudos de Thakor, Razzaque e Khandaker (2021), Hatzivasilis et al. (2018), Mohajerani et al. (2021).

- **Substitution Permutation Network (SPN):** modifica o dado, por *substituition box* e tabelas de permutação, formulando os dados para o próximo *round*
- **Feistel Network (FN):** quebra o bloco de entrada, em partes iguais e aplica um processo de difusão em cada *round*. Para uma das metades, além disto, a troca entre às metades acontece no começo de cada *round*.
- **Generalised Feistel Network (GFN):** é uma versão extrapolada do FN padrão. Quebrando o *input* em uma série de sub blocos e aplica FN para cada par de sub-bloco, em seguida aplicando um *shift* proporcional ao número de sub-blocos (SUZAKI; MINEMATSU, 2010).
- **Add-Rotate-Xor (ARX):** realiza encriptação e decriptamento usando adição, rotação e funções XOR, sem o uso de S-Box. Implementações de ARX são rápidas e compactas, porém limitadas em propriedades de segurança comparadas as SPN e FNs.
- **Nonlinear-Feedback Shift Register (NLFSR):** aplicam-se ambas a *stream* e *block ciphers*, que basicamente são *shift registers* que são unidades lógicas que aplicação a operação *bit shift* com base no *input* e além disto usa como *feedback* o valor anterior (BOGDANOV, 2007).
- **Hybrid:** cifradores híbridos, combina qualquer um dos três tipos ou até misturam propriedades de *block ciphers* e *stream ciphers* para melhorar qualidades específicas como, *throughput*, energia, entre outros, com base nos requisitos da aplicação.

Uma das estruturas de dados mais comuns utilizadas em criptografias são as *substituition box* ou S-Box. Tratando-se basicamente de tabelas de substituição, cujo objetivo é criar um resultado não linear de saída, não importando a sequência de entrada dos dados. Este requisito é necessário para não ser possível realizar a criptoanálise do dado e conseguir extrair a chave criptográfica do mesmo.

Outra técnica muito utilizada em algoritmos criptográficos é o uso de operações *bitwise XOR* que por si só é uma operação não linear binária.

3.4.4 Criptografias leves e dispositivos de IoT

O conceito de criptografias leves, conhecida como LWC - *Lightweight Cryptography* tornou-se relevante com a crescente demanda por criptografia em dispositivos IOT. Estes com limitado poder de processamento, o trabalho de Thakor, Razzaque e Khandaker (2021) discute muito bem isso abordando uma série de desafios deste campo de pesquisa. Com um extensivo teste de mais de 50 algoritmos de criptografias leves. Estes são algoritmos criptográficos executáveis em dispositivos com baixo poder computacional, como, por exemplo, um micro-controlador de 8 bits que envia dados via rádio, rodando em uma pilha moeda. Este dispositivo precisa realizar o mínimo processamento possível para um máximo proveito de sua pequena bateria. Esta categoria de dispositivo não pode ficar

muito tempo acordado realizando uma complexa cifragem de dados. Criptografias leves oferecem a possibilidade de dispositivos limitados possuírem condições de segurança e durabilidade de bateria.

Na literatura recente, muitos algoritmos leves foram propostos por pesquisadores, no entanto, muitos trabalhos como o de Thakor, Razzaque e Khandaker (2021) revelam falhas de segurança nestas novas implementações. Já os autores de Hatzivasilis et al. (2018) realizam uma análise profunda de diferentes algoritmos de criptografia leve propícios para IoT, mas sem abordar o escopo de segurança. O trabalho Mohajerani et al. (2021) envolve uma série de algoritmos promovidos por uma competição do NIST (*National Institute of Standanrds and Technology*), escritos em cima de uma API para dispositivos de baixa potência. No trabalho Thakor, Razzaque e Khandaker (2021) foram analisados 41 algoritmos de criptografia simétrica, em cima de 7 métricas de desempenho, estas sendo: requisito de memória, uso de flash, latência, *throughput*, eficiência energética e uso de portas lógicas. Os destaques que tem-se neste artigo são os seguintes algoritmos, SIMON e SPECK, com PRESENT e CLEFIA sendo oficialmente os mais bem votados no quesito segurança e leveza de implementação. Por fim o estudo de Sevin e Mohammed (2021) com foco apenas em *block ciphers* para IoT, com 20 implementações ao todo, conclui que a melhor *block cipher* equilibrando métricas de velocidade ao custo de RAM/ROM seriam: PRESENT , SPECK , SIMON e CLEFIA.

3.5 Algoritmos de criptografia

Os algoritmos que serão utilizados no presente trabalho, serão descritos nas próximas seções, discorrendo sobre suas capacidades criptografias e recentes análises de segurança.

3.5.1 AES-CBC

AES Selent (2010) é um exemplo clássico de algoritmo do tipo SPN. Padronizado pelo NIST, funciona em blocos de 128 bits com chaves de 128,192 ou 256 bits (MORADI et al., 2011). Possui um bom desempenho quando adicionado alguns recursos de *software* como demonstra (SALLAM; BEHESHTI, 2018). Também segundo Mouha, Dworkin et al. (2021) um trabalho do NIST, levanta como a criptografia é segura. Por usar uma chave de 256 bits mostrou-se resistente até a técnicas quânticas de análise criptográfica (RAO et al., 2017).

3.5.2 SPECK

Algoritmos com blocos de 128 bits implementados por Beaulieu et al. (2015), da mesma família do SIMON. Também produzido pela NSA, trata-se de uma criptografia leve com foco em ser implementado usando *software* suporta chaves de até 256 bits. É um algoritmo da classe (ARX), possui uma das implementações mais simples, com apenas uma operação principal que executa toda a característica de difusão e cifração. A operação denominada pelo algoritmo 2 de **R** com significando "rotação" referente a seguinte sequência de código, considerar que as variáveis são todos inteiros de 64 bits.

3.5.3 PRESENT

Algoritmo eficiente focado em *hardware*, aprovado pela ISO/IEC, é um algoritmo SPN, que possui duas variantes, com chaves de 80 bits e de 128 bits, é profundamente estudado e possui inclusive versões aprimoradas em *software*. Apesar de que o seu verdadeiro

Algoritmo 1 Algoritmo de criptografia AES-CBC

```
procedure encrypt(in: bytes[8],out:bytes[8], key: bytes[16], rounds: int)
    var estado = in;
    AddRoundKey(estado,key);
    for estado = 0, 1, ..., 14
        SubBytes(estado);
        ShiftRows(estado);
        MixColumns(estado);
        AddRoundKey(estado,key);
    end
    SubBytes(estado)
    ShiftRow(estado)
    AddRoundKey(estado,key)
    out = estado;
end
```

Algoritmo 2 Função **R**

```
#define ROR(x, r) ((x » r) | (x « (64 - r)))
#define ROL(x, r) ((x « r) | (x » (64 - r)))
#define R(x, y, k) (x = ROR(x, 8), x + = y, x ^= k, y = ROL(y, 3), y ^= x)
```

Algoritmo 3 Algoritmo de criptografia SPECK

```
procedure encrypt(in: bytes[16],out:bytes[16], key: bytes[32], rounds: int)
    var x,y,a,b;
    y = in[0..7];
    x = in[8..15];
    b = key[0..7];
    a = key[8..15];
    R(x,y,b)
    while rounds − −
        R(a, b, rounds);
        R(x, y, b);
    end
    out[0..7] = y;
    out[8..15] = x;
end
```

forte se baseia em implementações de *hardware* Bogdanov et al. (2007), o algoritmo 4 mostra o seu funcionamento.

A cada *round* do algoritmo, será realizada uma operação XOR do dado de entrada com a chave criptográfica, este processo é chamado **addKey**. Em seguida será realizado o passo de substituição, que passará por todos os bytes. Cada byte terão os seus 4 *nibbles* inferiores e superiores são substituídos por um s-box de 4bits. Após essa substituição, será realizado o passo de permutação, em que é realizado a operação *bitwise shift*, embaralhando os dados permutados. Por fim a chave criptográfica é atualizada utilizando o mesmo s-box de 4 bits em seguida realizando uma rotação por meio operadores *bitwise shift* um número *round* de vezes.

Algoritmo 4 Algoritmo de criptografia PRESENT

```
procedure encrypt(in: bytes[8],out:bytes[8], key: bytes[16], rounds: int)
    while rounds − −
        addKey(in,key);
        substituition(in);
        permutation(in);
        updatekey(key,round);
    end
end
```

3.5.4 CLEFIA

Assim como PRESENT é um algoritmo aprovado pela ISO/EIC 259, possui compatibilidade com chaves de 128, 192 ou 256 bits (SHIRAI et al., 2007). Demonstra alto desempenho e forte imunidade contra vários ataques (HATZIVASILIS et al., 2018). A sua grande capacidade de difusão e complexo processo de embaralhamento dos dados faz com que necessite de um pouco mais de memória. Devido a este fato possui um uso limitado em aplicações de dispositivos que são muito restritos (BANSOD; RAVAL; PISHAROTY, 2014). Trata-se de um algoritmo GFN, seu algoritmo pode ser simplificado pela seguinte sequência de funções mostradas no Algoritmo 5.

Algoritmo 5 Algoritmo de criptografia CLEFIA

```
procedure encrypt(in: bytes[16],out:bytes[16], key: bytes[32], rounds: int)
    bytexor(in[4],in[4],key[0],4);
    bytexor(in[12],in[12],key[4],4);
    clefiaGFN4(out,int,key[12],rounds);
    bytexor(out[4],out[4],key[rounds ∗ 8],4);
    bytexor(out[12],out[12],key[round ∗ 8 + 4],4);
end
```

Com base no Algoritmo 5 observa-se que ele possui duas grandes operações principais, uma delas a operação XOR em blocos de 32 bits, chamada bytexor e a sua função GFN reversível, para cifrar e decifrar os dados, demonstrada no Algoritmo 6:

- **Bytexor**: Esta função executa uma operação *bitwise XOR* em um conjunto de bytes. Recebe como primeiro argumento o vetor que irá receber o resultado. Em seguida os 2 vetores que serão realizados a operação XOR, por fim o número de elementos a serem operados. Esta operação fornece característica criptográfica importante que é ocultar a chave utilizada no primeiro bloco, forte alvo de criptanálise.
- **Clefia GFN4**: Aqui tem-se a principal operação do algoritmo CLEFIA, a implementação desta função pode ser observado no Algoritmo 6. Em que são aplicados duas camadas de difusão F0 e F1, a primeira camada é aplicada nos 8 bytes iniciais, a segunda nos 8 bytes finais. Ambas camadas realizam uma substituição s-box nos 4 primeiros bytes trabalhados. Também realizará uma operação xor matricial dos primeiros 4 bytes, com os 4 últimos, na forma de uma matriz 2x2. Esta camada garante que os dados não sejam de maneira nenhuma lineares e ofusque completamente os blocos seguintes. No total 2 s-box de 256bytes são necessárias, com espaço para uma terceira para otimizar o *throughput*.

Algoritmo 6 Algoritmo GFN4 Clefia

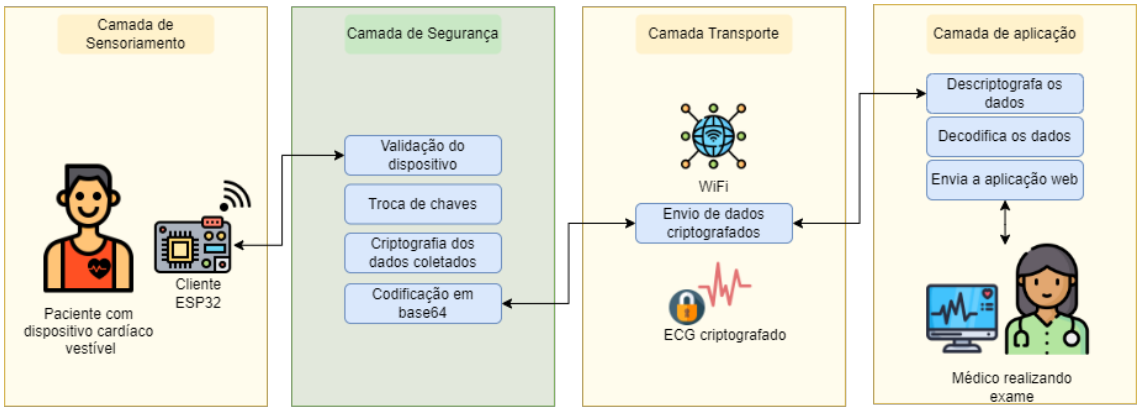
```
procedure clefiaGFN4(in: bytes[16],out:bytes[16], subkey: bytes[32], rounds: int)
    var i = 0;
    while rounds − −
        ClefiaF0Xor(out[0],in[0],subkey[i])
        ClefiaF1Xor(out[8],in[8],subkey[i+4]
        i+ = 8
    end
end
```

4 Desenvolvimento do Trabalho

4.1 Modelagem

A Figura 8 ilustra em uma alto nível de abstração, as interações relacionadas a modelagem da camada de segurança que será validada. É possível observar também as camadas de IoMT e a relação com os dispositivos do protótipo. O fluxo das informações desde a camada de sensoriamento até a camada de aplicação. Estas interações serão explicadas no desenvolvimento da implementação e realização dos testes.

Figura 8 – Modelagem da camada e interações com os dispositivos

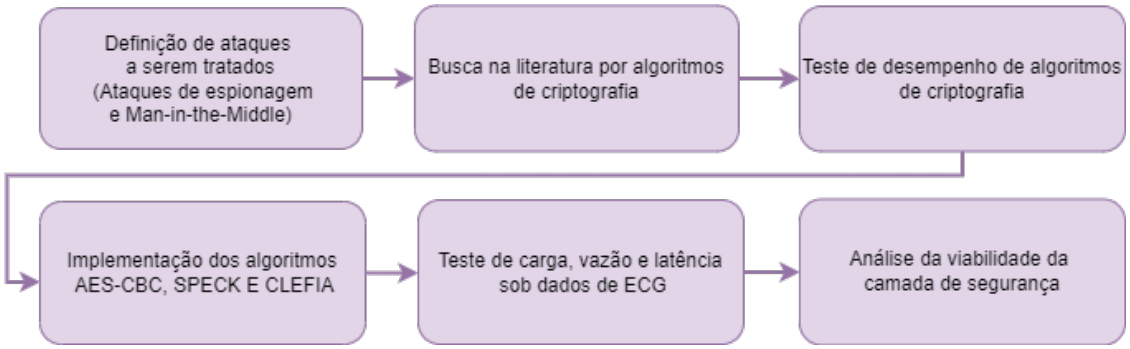


Fonte: A própria autora.

4.2 Metodologia

Um breve resumo da metodologia pode ser acompanhada pelo fluxograma da Figura 9. As etapas mais importantes para a execução do trabalho e obtenção dos resultados foram agrupadas de forma que seja possível compreender o estudo e implementação realizada. Nas próximas seções serão discutidos aprofundadamente cada uma das etapas.

Figura 9 – Fluxograma de Metodologia



Fonte: A própria autora.

Após análise dos ataques possíveis de serem testados foram escolhidos ataques de espionagem e *man-in-the-middle*. Com relação aos algoritmos de criptografia foram selecionados:

- AES256 (*hardware* esp32)
- SPECK (software)
- CLEFIA (software)
- PRESENT (software)

Além disso, destaca-se a que foram definidas as métricas de *throughput* ou vazão e latência para avaliar o desempenho entre os diferentes algoritmos de criptografia implementados na camada de segurança.

4.3 Implementações algoritmos

Foram utilizadas a arquitetura do ESP32 e a linguagem de programação C, com as seguintes configurações para a implementação dos algoritmos.

- GCC 8.4.0 para arquitetura Xtensa
- Flags de compilação: -O2
- Configuração de CPU: 240 MHz
- Configuração de Flash: 80 MHz
- Sistema operacional: FreeRTOS
- Framework: Arduino e ESP-IDF

Para a implementação no servidor, esta será por meio de um executável compilado em C++ que será chamado pela aplicação Node.js do servidor, este código é compilado com a seguintes configurações:

- GCC 10.2.0 MinGW/Posix para arquitetura x86-x64
- Flags de compilação: -O3
- Configuração de CPU: Intel Core i7 9750H a 2.6GHz 6 núcleos 12 threads
- Sistema operacional: Microsoft Windows® 11 64 bits

4.3.1 Resultados de performance

Na Tabela 2 são apontadas as principais características de cada algoritmo. Primeiramente, a criptografia integrada no microcontrolador ESP32 - a AES-CBC, apresenta o melhor desempenho em comparação com os demais algoritmos com relação *throughput* ou vazão dos dados. OU seja, a quantidade de dados transferidos foi muito superior aos demais, tanto para encriptar quando para decriptar. Já que se trata de uma unidade de *hardware* dedicado, é esperado que esta possua uma boa eficiência, no entanto, ela está limitada a um pequeno conjunto de algoritmos fixos, quaisquer outros que sua aplicação desejar usar, terá de ser implementado através de código em *software*.

Tabela 2 – Algoritmos analisados e selecionados

Criptografia	AES CBC	PRESENT	SPECK	CLEFIA
Throughput Encrypt ESP32	7.03 MB/s	0.07 MB/s	2.02 MB/s	0.65 MB/s
Throughput Decrypt ESP32	7.2 MB/s	0.07 MB/s	1.88 MB/s	0.65 MB/s
Throughput Servidor	300 MB/s	15 MB/s	290 MB/s	120 MB/s
Throughput Decrypt Servidor	300 MB/s	15 MB/s	290 MB/s	120 MB/s
Tamanho Chave	256 bits	128 bits	256 bits	256 bits
Tamanho Bloco	128 bits	64 bits	128 bits	128 bits
Uso Flash (ESP32)	512 bytes	3.5 KB	3.2 KB	2.9 KB
Uso RAM (ESP32)	892 bytes	1.3 KB	1.1 KB	2.5 KB
Otimizações ESP32	Acelerado por *Hardware*	N/A	Operações 32 bits	Lookup tables
Segurança Quântica	Sim	Não	Sim	Sim

Nas implementações de *software* observa-se o desempenho extremamente baixo do algoritmo PRESENT, devido a seu funcionamento totalmente focado em otimização para pouco uso de silício, sua implementação de *software* é extremamente limitada, pois, arquiteturas modernas de 32 bits, precisam de dezenas de ciclos para lidar com acessos de

4 bits, com uma segunda característica negativa de suportar chaves de apenas 128 bits no máximo.

Ao contrário do algoritmo SPECK, que por usar tipos de dados de 32 e 64 bits, favorecem arquiteturas modernas com operações consumindo poucos ciclos, faz com que possua um excelente *throughput* com baixo uso de Flash e RAM, além disto possibilitando usar chaves de 256 bits. Por fim o algoritmo CLEFIA, bastante estudado na literatura, possui um *throughput* mediano, por se dar categoria de *Feistel network*, trata-se de um algoritmo com elevada segurança a ataques, mas com custo de RAM considerável.

Com o resultado muito inferior da implementação de *software* da criptografia PRESENT, esta será descartada dos testes de rede, visto que não representa uma solução interessante, tanto do ponto de vista de *throughput* quanto de segurança.

Já o servidor devido à arquitetura de *hardware* mais robusta de um microprocessador para aplicações desktop, possui um conjunto de instruções mais completo dando possibilidade para que o compilador otimize mais a execução dos algoritmos. Este conjunto de fatores faz com que o servidor possua um *throughput* muito maior.

4.3.2 Implementações de comunicação

Com relação às transmissões entre as camadas, foram necessárias implementações para com relação à comunicação entre o ESP32 e o servidor. Buscando uma comunicação confiável, tolerante à falha, sincronizada e inteligível por ambas as partes, foram necessárias as seguintes implementações:

Client Socket.io: Foi utilizado uma biblioteca de terceiros para possibilitar a comunicação com o protocolo socket.io que funciona em cima de *websockets* possibilitando uma série de funcionalidades a mais além da implementação padrão de *websockets* puros.

Cliente NTP de alta precisão: Para ser possível mensurar a latência dos pacotes foi necessário utilizar uma biblioteca que implementa clientes NTP para o ESP32 com maior precisão. Pois, a implementação padrão do fabricante garante apenas precisão de alguns segundos, utilizado um NTP preciso e usando um mesmo servidor brasileiro de NTP para sincronizar ambos esp32 e servidor, garantimos uma maior proximidade de relógios dos dispositivos.

Biblioteca JSON: Para que os dados sejam formatados de maneira a serem processados mais facilmente pelo servidor optou-se por utilizar uma biblioteca de alto desempenho e agnóstica chamada ArduinoJSON para o ESP32, o servidor por ser *javascript*, naturalmente já trabalha com este tipo de dado.

Bibliotea base64: Quando se tem texto criptografado, estes deixam meramente de serem texto e se tornam sequencias de *bytes* inteligíveis, e com isso complica-se para que linguagens de alto nível façam o *parsing* destes dados, para isso, estes são enviados em formato base64, para que o servidor descriptografe diretamente para texto JSON e possa trabalhar com este.

Biblioteca Crypto: Biblioteca de criptografia que implementa funcionalidades para execução da troca de chaves. Por meio do algoritmo Diffie-Hellman. Este é usado puramente para adquirir um segredo compartilhado seguro entre servidor e ESP32 no processo de autenticação. Este processo ocorre em 2 etapas, a primeira com cada uma das entidades gerando uma chave pública, que será trocada entre sí. A segunda etapa, cada um irá mixar as chaves públicas recebidas com sua chave privada, resultando em uma chave

igual para ambos, e secreta.

O executável em C++ do servidor que gerencia a criptografia dos dados, divide a implementação dos algoritmos em C com o ESP32. Exceto pelo algoritmo AES, que por utilizar implementação de *hardware*, optou-se por utilizar uma implementação open source para o servidor. Também destacam-se algumas bibliotecas que se utiliza para esta aplicação.

Biblioteca simdjson: Biblioteca de altíssimo desempenho utilizada para processar JSON o mais rápido possível.

jsoncpp: Para criar um arquivo JSON de resposta precisamo de outra biblioteca, para isso foi escolhido uma biblioteca que implementa de uma forma moderna na linguagem C++ suporte a JSON.

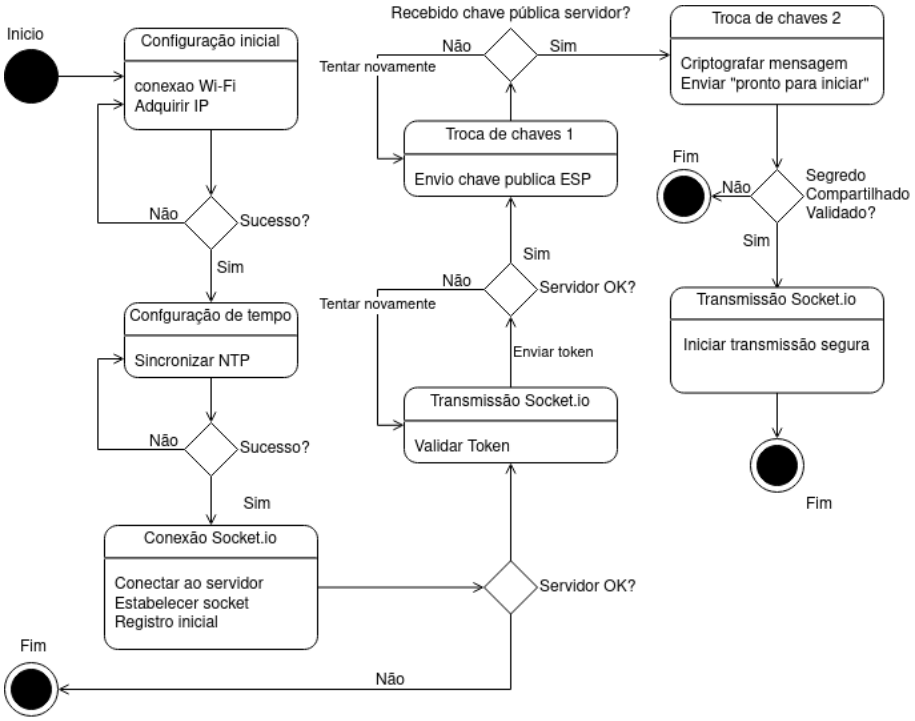
4.4 Comunicação ESP32

A seguir serão apresentados o processo de comunicação do dispositivo com o servidor. Abordando todos os aspectos do código e como serão executadas as etapas para alcançar uma comunicação segura. Além do formato de comunicação empregado pelo ESP32 e o servidor.

4.4.1 Máquina de estados

A maquina de estados representada pela Figura 10, demonstra todas as etapas de funcionamento do dispositivo em um processo de mensuração normal.

Figura 10 – Fluxo de execução do ESP32.



Fonte: Própria autora.

Configuração inicial: ESP32 irá realizar a inicialização do seu sistema operacional e configurar suas interfaces. Após este passo ele irá conectar-se a uma rede Wi-Fi previamente configurada. Ao conectar com sucesso adquirirá um IP através de DHCP. Caso o processo falhe, conectará em outra rede pré configurada, se houver, e tentará novamente adquirir um IP. Com Rede e IP Validados está pronto para a próxima etapa.

Configuração de tempo: Neste passo o ESP32 tentará realizar uma conexão com o servidor de tempo brasileiro **pool.ntp.br**. Este passo valida que a rede possui uma conexão com internet e durante um período o ESP32 irá comunicar-se para sincronizar com precisão a data e hora do seu relógio interno.

Conexão Socket.io: Após validar conexão e seu relógio interno, o dispositivo iniciará o processo de estabelecer uma comunicação com o servidor. Primeiramente acessando o IP do servidor já configurado, e efetuando algumas transações HTTP. Estas transações são necessárias para validar que o servidor está utilizando o protocolo Socket.io. Após a validação inicial o dispositivo iniciará uma conexão *websocket* com o servidor que prontamente, configurará para ambos uma sessão Socket.io. Com esta seção iniciada o ESP32 ira se registrar no domínio inicial "/". Notificando o servidor de uma nova conexão com sucesso.

Validar Token: Com a conexão estabelecida, o dispositivo de medição irá enviar o seu *token* interno pré registrado, de modo a validar com o servidor que se trata de um dispositivo conhecido. Ao verificar com sucesso, o servidor responderá que está OK para operar a comunicação.

Troca de chaves 1: A primeira etapa do procedimento de troca de chaves ocorre após a validação do *token*. Em que ambos os servidor e o ESP32, geram internamente um conjunto de chaves, uma chave pública e uma chave privada.

Troca de chaves 2: A segunda etapa, ambos trocam suas chaves públicas, de forma que ambos derivam um segredo compartilhado. Ao combinarem as chaves públicas recebidas com a chave interna gerada, ambos terão como resultado a mesma chave. Esta mesma chave compartilhada, será usada para criptografar a comunicação entre dispositivos. Com esta chave compartilhada o ESP32 enviará uma mensagem criptografada avisando ao servidor que está pronto para comunicar. Caso o processo de troca de chaves ocorra sem problemas, o servidor conseguirá descriptografar a mensagem e identificar que o procedimento ocorreu com sucesso.

Transmissão Segura: Com todo o processo de segurança validado no passo anterior, agora dados podem ser trocados seguramente. O dispositivo irá despachar todos os dados que necessitar até finalizar o processo.

4.4.2 Formato de mensagem

A comunicação Socket.io espera uma formato de mensagem específico, em que se trata de uma lista de argumentos, no formato JSON, como pode ser visto no Algoritmo 7. O primeiro objeto da lista é uma string, que irá ser interpretada como o evento do Socket.io. Já os argumentos seguintes da lista, estes podem ser quaisquer objetos válidos JSON.

Algoritmo 7 Formato da mensagem

```
["event", {"argumento1": "dados_argumento1"}, "argumento2"]
```

Um exemplo de comunicação pode ser observado no Algoritmo 8. Este são os dados puros sem nenhum tratamento ou proteção. No exemplo destacam-se as amostras com os valores em milivolts de um eletrocardiograma.

Algoritmo 8 Mensagem padrão sem criptografia e sem codificação

```
["amostragem", {
        "amostras": [2500, 2430, 2652, 2654, 2521]
}, "1646587707"]
```

Os dados da amostragem são então codificados no padrão base64 como mostrado no Algoritmo 9. Utiliza-se deste formato para ser interpretado como *bytes* de maneira mais fácil por aplicações em javascript.

Algoritmo 9 Mensagem padrão sem criptografia e codificado em base64

```
["amostragem", {
    "amostras": "WzI1MDAsIDI0MzAsIDI2NTIsIDI2NTQsIDI1MjFd"
}, "1646587707"]
```

Por fim as amostras serão criptografadas usando a chave compartilhada, como pode ser constatado no Algoritmo 10. Ao tentar decodificar está sequencia base64 não será possível identificar o conteúdo devido à criptografia.

Algoritmo 10 Mensagem padrão criptografada e codificada em base64

```
["amostragem", {
    "amostras": "R2o4Mm0zWnFnMDVGNExJZmhpZnR4OXJUaXZ4UGNT"
}, "1646587707"]
```

O servidor, que possuirá a mesma chave criptografia do dispositivo. Transformará o dado codificado em base64 e criptografado, em uma sequência de caracteres válidos JSON.

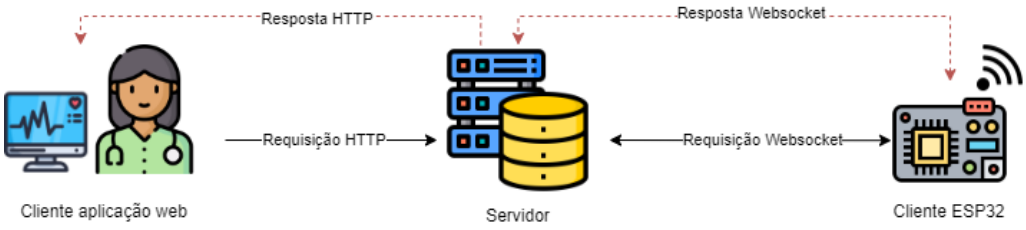
4.5 Visão geral do servidor

O servidor da aplicação foi desenvolvido em Node.js um framework de desenvolvimento em *javascript* que se baseia no interpretador V8 da Google®. É responsável por receber as requisições do usuário que irá executar o exame da aplicação web, e fazer o gerenciamento dos dados recebidos do exame em tempo-real.

4.5.1 Arquitetura cliente-servidor

Este modelo de arquitetura de rede que possui duas entidades, os clientes que executam tarefas baseadas no usuário e os servidores que providencia tarefas com base nos serviços requisitados pelo usuário. São dois clientes destintos nesta aplicação, a aplicação web a qual o usuário interage e os EPS32 localizados no colete usado pelo paciente. As requisições entre a aplicação web e o servidor como Figura 11 são HTTP. Para a comunicação ESP32 e servidor foi usado o protocolo de comunicação baseado em cliente-servidor *Websocket*.

Figura 11 – Arquitetura cliente-servidor da aplicação



Fonte: Própria autora.

4.5.2 Aplicação web

O usuário previamente cadastrado realiza *login* com seu *username* e senha. Seleciona o paciente cadastrado e inicia o exame, podendo visualizar a exibição tem tempo real do mesmo sendo realizado. Quando o usuário finaliza o exame, é gerado um arquivo com os dados do paciente selecionado e do usuário que realizou o exame, que pode ser um enfermeiro ou médico. O *front-end* da aplicação foi desenvolvido com React, uma biblioteca *open-source* da linguagem de programação *Javascript*, desenvolvida pelo *Facebook*. A comunicação entre o servidor e a interface é feita através de requisções HTTP como GET, POST, PUT e DELETE.

4.5.3 Autenticação de usuários

A autenticação de usuários da aplicação web faz o uso do mecanismo chamado *JSON Web Token* (JWT). JWT são métodos RFC7519 padrão para realizar autenticação de forma entre duas partes, onde é gerado um token em Base64 e fará com que o token seja assinado corretamente. a interface efetua uma requisição do servidor com o *username* e senha do usuário. O servidor retorna o token JWT e então a interface responde à requisição do token e o servidor responde à confirmação do token.

4.5.4 Autenticação de dispositivos

Os dispositivos possuem um token único criado em um cadastro prévio. este é validado assim que a conexão websocket é inicializada. Ao ter sua identidade verificada inicia-se o processo de troca de chaves. Este processo de duas etapas visa criar uma chave segura, conhecível apenas por ambas as partes. A primeira etapa é a geração de chaves privadas e públicas em ambos cliente e servidor. Em seguida ambas as partes efetuam uma troca de chaves públicas, a chave final será derivada da chave privada secreta e da chave pública recebida. A chave final será igual para ambos graças ao algoritmo Diffie-Hellman. Assim toda comunicação agora será criptografada usando esta chave final. Mesmo que um usuário malicioso capture ambas as chaves públicas, ele não conseguirá decifrar a chave final.

4.5.5 WebSocket

Assim como o HTTP, o *WebSocket* é usado na comunicação cliente-servidor. Enquanto o HTTP é unidirecional, ou seja o cliente envia uma requisição e o servidor uma resposta, o *WebSocket* é bidirecional, *full-duplex*. Depois que a conexão for aberta, ela

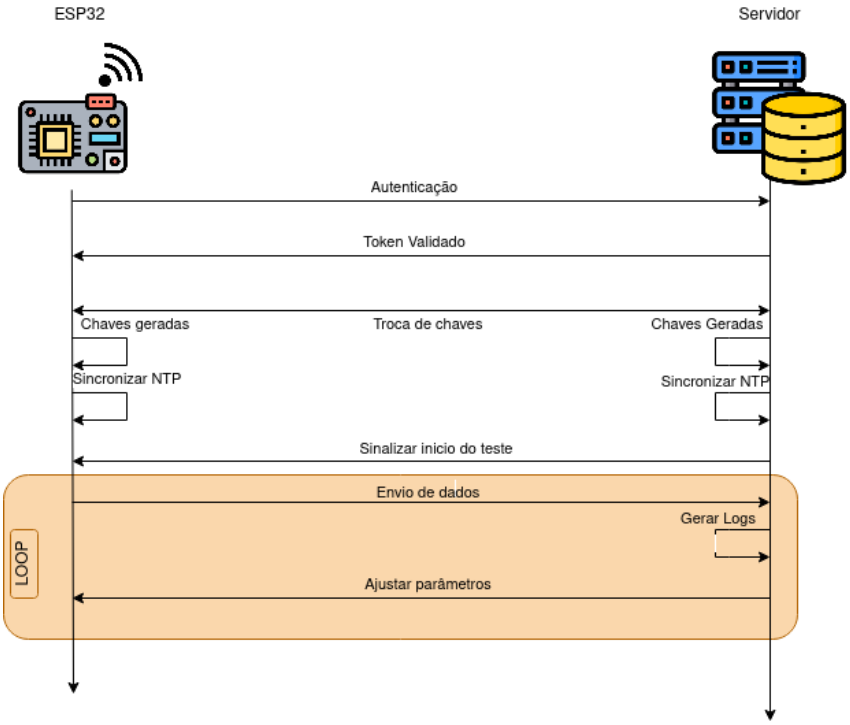
somente será encerrada se alguma das partes solicitar. Como a aplicação trata de dados em tempo real o uso de um soquete que faz a conexão entre o cliente ESP32 e o servidor seria recomendada para que o servidor não tenha sobrecarga de requisições.

4.6 Testes

Para realizar a análise completa de todo o sistema em execução, foi criado um roteiro de testes automatizado com foco em automaticamente ajustar parâmetros. Como sincronização de relógios, ajuste de quantidade de amostras, ajuste de criptografia e armazenar resultados.

A figura 12 demonstra como o teste foi configurado, primeiro inicia-se todo o processo de autenticação, foi adicionado um processo de sincronização de relógios para que ambos o ESP32 e Servidor estejam com seus relógios sincronizados, de modo a conseguir mensurar a latência da comunicação, após um período recebendo os dados e gerado *logs* o servidor irá configurar o ESP para outro conjunto de número de amostras e criptografia desejada, e o teste se reinicia novamente.

Figura 12 – Comunicação da camada de Segurança.



Fonte: Própria autora.

A estrutura dos dados do teste, possui o formato JSON, como descrito na seção 4.4.2. O servidor ao receber este pacote, irá realizar a descriptação das amostras, utilizando a chave que compartilha com o ESP32. No momento em que este dado é recebido ele irá adquirir um *timestamp* com precisão de microssegundos, e comparar com o *timestamp* contido na mensagem. Desta forma é possível inferir a latência desde a criação do pacote até seu recebimento com sucesso no servidor.

Os testes são executados por um tempo pré definido, até que seja completo uma rodada de testes e uma ou mais variáveis do teste sejam modificadas, as variáveis são:

- Quantidade de amostras, de 100 a 1000 amostras por pacote, alterando 10 em 10 do mínimo ao máximo;
- Criptografia utilizada, com opções: sem criptografia, AESCBC, SPECK e CLEFIA;

A cada rodada de teste são contabilizados os seguintes parâmetros. Estes são armazenados em arquivos de log para serem contabilizados assim que todos os testes terminarem.

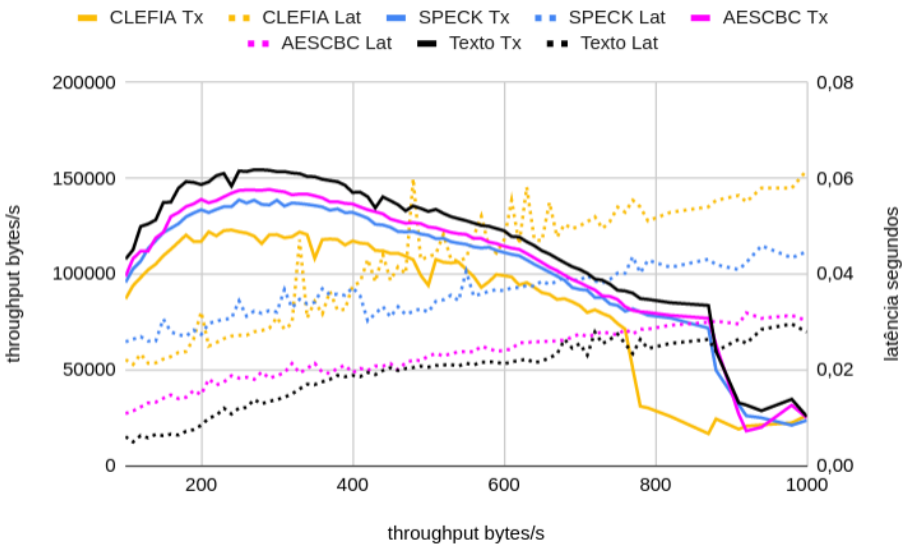
- Quantidade média de bytes recebidos;
- Quantidade média de pacotes recebidos;
- Latência média dos pacotes;

Além dos dados citados, também são armazenadas outras variáveis internas de modo a validar que o teste aconteceu com sucesso. Sem ter acontecido nenhuma intermitência de rede, problemas na execução do teste, etc.

5 Resultados e Discussões

Uma série de testes foram realizados e os resultados obtidos serão discutidos nesta seção. Para mapear o comportamento de cada um dos algoritmos, dado um tamanho de amostra, o gráfico da Figura 13 ilustra os resultados dos testes com todos os algoritmos. A seguir serão apresentados os resultados de cada algoritmo individualmente.

Figura 13 – Resultado do conjunto de testes.



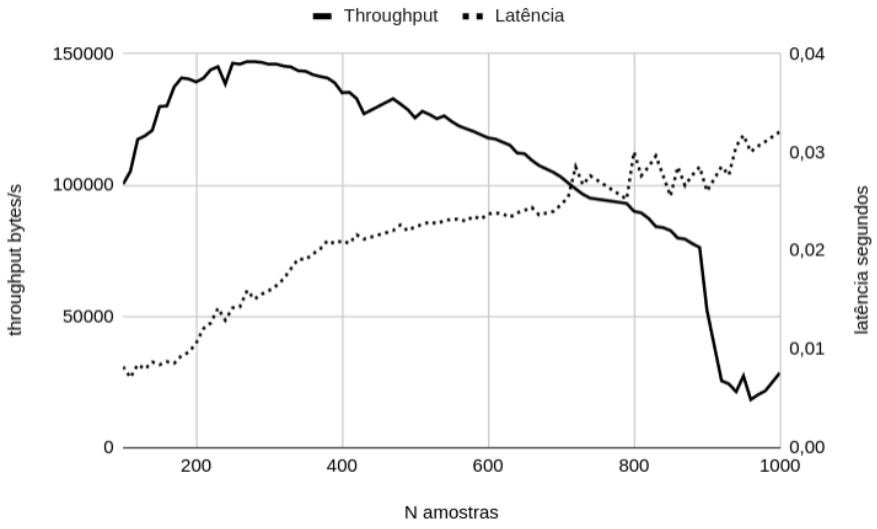
Fonte: Própria autora.

Além disso, para fins de análise e de comparação, um teste foi realizado sem o uso da criptografia. Para verificar o funcionamento dos dispositivos sem a função implementada e assim ter resultados de medição para comparar com os algoritmos que estão sendo avaliados.

5.1 Sem criptografia

Neste cenário, representado pela figura 14 trata-se do cenário ótimo. Onde o dispositivo apenas envia as amostras como texto puro, sem criptografia. Apresenta o maior *throughput* possível com a menor latência. Este é o resultado base que seria o melhor cenário para nossa aplicação. É possível observar que o pico de *throughput* encontra-se com o tamanho da amostra entre 200 a 400 amostras. Este pico de vazão de dados está relacionado a proximidade do tamanho do pacote em bytes com o valor do MTU da rede, padrão de 1500 bytes. Quando este valor está entre 1500 e 3000, está num nível ótimo de fragmentação, o ESP32 executa rapidamente. Acima deste valor apresenta uma perda de desempenho. É possível observar neste gráfico que existe uma variação grande de latência, esta variação se eleva assim que tem-se um pacote maior. Devido às dinâmicas de transmissão wifi e TCP. Além destas dinâmicas há um pequeno incremento de processar um pacote maior.

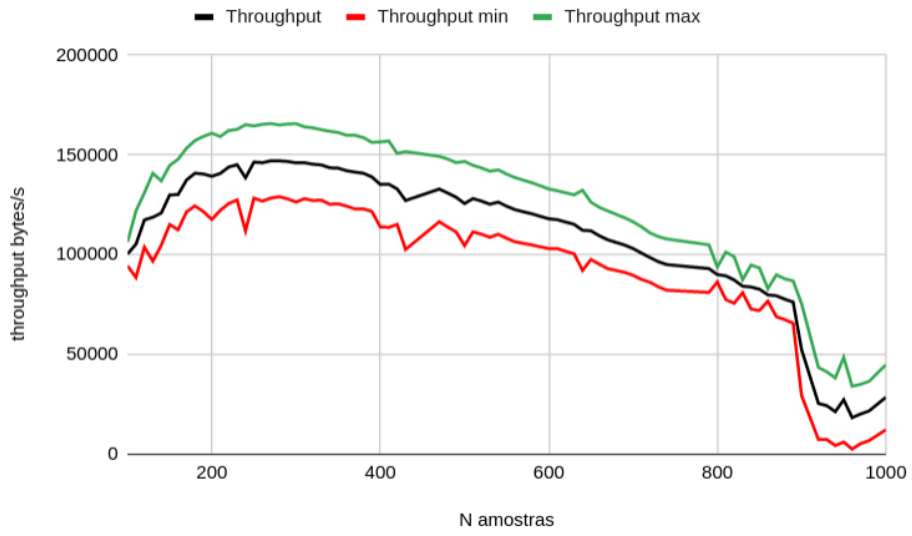
Figura 14 – Resultado *throughput* e latência sem criptografia.



Fonte: Própria autora.

Durante o período de testes também pode ser observado as variações de *throughput* a cada tamanho de amostra. É possível ver na Figura 15 que o *throughput* oscila mais quando o tamanho do pacote é menor, mais sensível a situações de uso da rede sem fio, porém mantém-se variando menos com amostras maiores. Observa-se que existem momentos em que com grandes amostras o *throughput* pode chegar a zero por pelo menos um segundo.

Figura 15 – Resultado de desvio padrão do throughput.

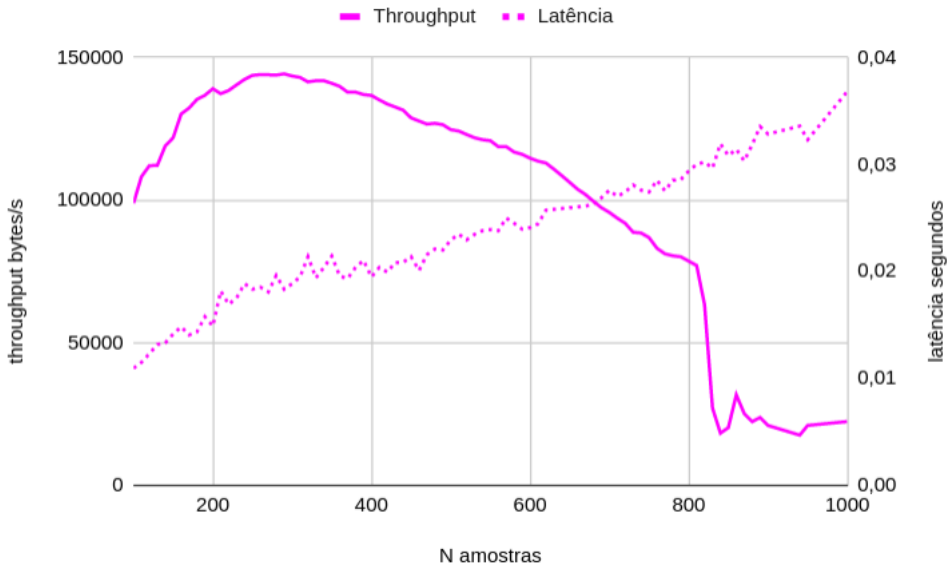


Fonte: Própria autora.

5.2 AESCBC

O primeiro algoritmo de criptografia, implementado em *hardware*. Que anteriormente apresentava o maior *throughput* teórico, mostrou que afeta pouco a taxa de dados, porém é possível observar na Figura 16 que a partir de 500 amostras, a latência aumentou consideravelmente.

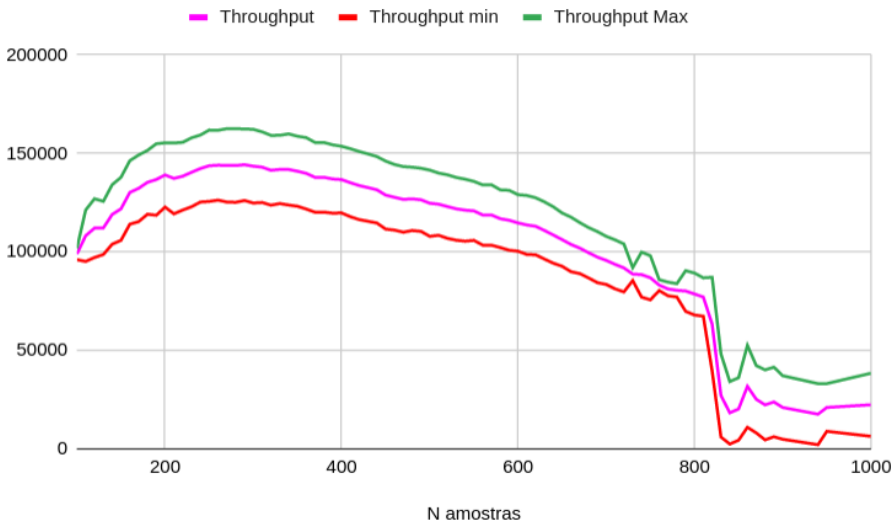
Figura 16 – Resultado *throughput* e latência com AES-CBC.



Fonte: Própria autora.

A variação de *throughput* e latência segue uma tendência ao texto puro, com algumas ressalvas quando o número de amostras é pequeno, variando para quase 50 KB/s. É possível identificar na Figura 17 o mesmo padrão e menor variação com número de amostras maior.

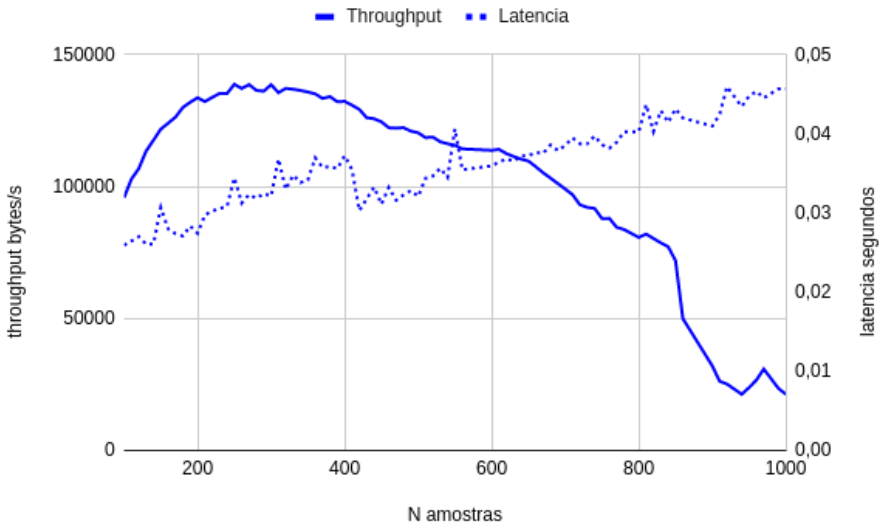
Figura 17 – Resultado de desvio padrão do throughput AES-CBC.



Fonte: Própria autora.

5.3 SPECK

Figura 18 – Resultado *throughput* e latência com SPECK.

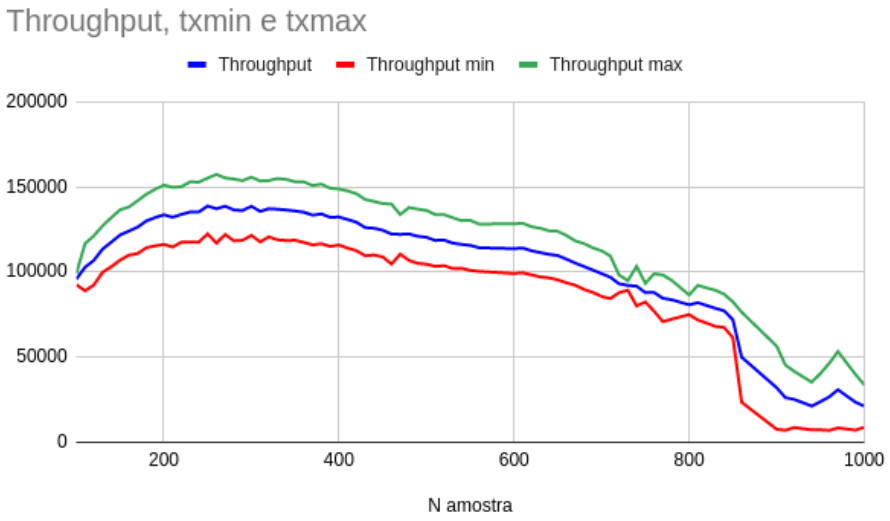


Fonte: Própria autora.

A segunda melhor implementação de criptografia, com detalhe desta funcionar em software. Apresenta um desempenho ainda satisfatório. é possível observar na figura 18 que o *throughput* máximo começou a diminuir, e observamos a latência com amostras acima de 500 a tomar um tempo considerável de latência. Ainda sim, amostras pequenas ainda mantém a latência próximo ao texto puro e criptografia AESCBC. é observável que durante o período do teste obtiveram-se menos variações bruscas, tanto de latência como de *throughput*.

A Figura 19 nos mostra o comportamento do algoritmo SPECK, no quesito de variação de throughput. Com uma variação de dados ainda ótima quando utilizado de 200 a 500 amostras, o algoritmo mesmo em software em pouco afetou a capacidade de transmissão.

Figura 19 – Resultado de variação de throughput *throughput* SPECK .

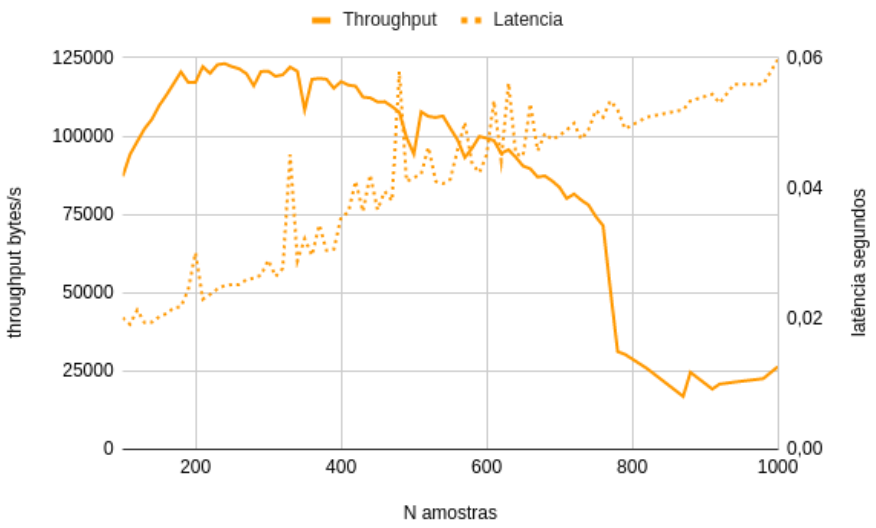


Fonte: Própria autora.

5.4 CLEFIA

O último algoritmo testado CLEFIA, obteve o menor desempenho dos algoritmos testados. Observável na Figura 20. Seu *throughput* ficou o menor entre os algoritmos, não chegando nem ultrapassar 125000 bytes por segundo. Apesar de também ter a maior latência, esta ainda está sob controle mostrando pouca diferença quando enviado menos de 400 amostras, após está marca segue com uma latência duas vezes maior que o algoritmo SPECK.

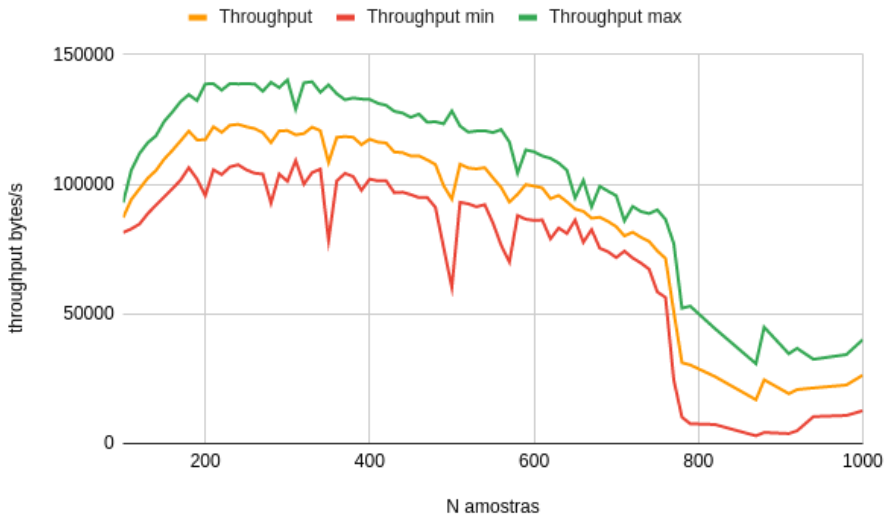
Figura 20 – Resultado *throughput* e latência com CLEFIA.



Fonte: Própria autora.

Por último observamos no gráfico da Figura 21 a variação de *throughput* do algoritmo CLEFIA. Em um cenário não ótimo de transmissão, este não conseguirá mais que 100000 bytes por segundo. O algoritmo mostra que afeta consideravelmente o *throughput* em toda quantidade de amostras.

Figura 21 – Variação de *throughput* CLEFIA.



Fonte: Própria autora.

6 Considerações Finais

6.1 Conclusão

A segurança e privacidade de dados são questões cruciais quando o assunto é Internet das Coisas Médicas ou IoMT. Soluções escaláveis para os desafios encontrados na coleta, transferência e armazenamento de dados tem sido encontradas na literatura científica mas poucos avanços em ambientes reais tem sido registrados. Ressaltando que os dados são considerados sensíveis nestas aplicações, e que podem comprometer a integridade física, mental e emocional de um paciente, ao ser exposto ou passar por uma experiência de ataque cibernético. Implementações de medidas de segurança em todos os níveis da arquitetura de IoMT tem sido amplamente pesquisados, pois existem, trabalhos recentes que abordam o tema com bastante propriedade. Visando analisar a viabilidade de uma camada de segurança quando os dados de dispositivo vestível cardíaco para eletrocardiograma proposto em Zanon e Ourique (2020), transmite seus os sinais adquiridos para um servidor proposto. Este trabalho realizou a análise de três algoritmos de criptografia empregados nas mensagens enviadas entre o dispositivo e o servidor. A partir dos resultados apresentados é possível comprovar a viabilidade da implementação desta camada de segurança para realizar uma solução de comunicação segura entre as entidades da arquitetura de IoMT. Para auxiliar na construção da camada foi realizada a implementação de *WebSocket*, troca de chaves não fixas, autenticação e criptografia.

Os testes realizados mostram valores de *throughput* suficientes para a realização de um exame em um paciente em tempo real. Como demonstra Zanon e Ourique (2020) com um ECG de 500Hz. Visto que um pacote com 500 amostras possui por volta de 3500 *bytes* e o *throughput* desta categoria de pacote segundo os ensaios realizados, gira em torno de 100000 *bytes* por segundo no pior caso. Sendo assim suficiente para amostrar a cada segundo 28 pontos de ECG, podendo ser implementado na aplicação proposta, melhorando assim a segurança o dispositivo vestível. Para tornar a coleta de dados menos suscetível a ataques MitM e de espionagem, como observado nos resultados, fazer o uso do algoritmo de criptografia AES-CBC se mostra uma opção viável para o caso de estudo. Seu uso é justificável pela questão de simplicidade de implementação, já que está presente em *hardware* no processador do dispositivo e também pelo desempenho elevado, em que não prejudica a comunicação de maneira significativa. Mostrando uma capacidade de transmissão similar ao dado sem criptografia.

Caso o dispositivo de estudo não possuir uma unidade de criptografia embutida, é possível optar por uma solução em software utilizando SPECK. Apesar de possuir teoricamente um terço da velocidade do algoritmo AES-CBC, nos testes o impacto foi relativamente pequeno. Garantindo ainda as capacidades de transmissão acima de 100000 bytes com amostras grandes. A latência em amostras pequenas praticamente dobrou, um ponto para ser considerado dependendo das necessidades da coleta de dados fisiológicos. Em amostras grandes a latência representa um aumento de 50%.

Uma segunda opção de criptografia via *software* CLEFIA, mostrou um desempenho reduzido. Apesar da latência equivalente ao algoritmo SPECK em amostras menores. A medida que se aumenta o número de amostras rapidamente o *throughput* é afetado e a latência sobe para os maiores níveis de todos os algoritmos. Mesmo com desempenho reduzido, ainda é uma opção viável para dispositivos que não enviam quantidades de dados elevadas e não demandam aplicações em tempo real. Estes dados, criptografados por quaisquer algoritmos, possuem suas chaves seguras e únicas apenas para cliente e servidor,

isso se deve ao procedimento de autenticação segura. Fazendo com que tentativas de adquirir dados ou chaves durante o processo de transmissão das mensagens seja dificultado, criando desta forma uma camada a mais de segurança para o paciente e a entidade que está realizando o exame.

6.2 Trabalhos Futuros

Para o desenvolvimento de trabalhos futuros, a camada de segurança posposta poderia ser submetida a simulações de ataque de espionagem, MitM e outros relacionados. Visando testar sua resistência a tentativas de invasão, pois devido às limitações do desenvolvimento do trabalho, não foi possível submeter os algoritmos de criptografia testados a estas simulações.

Uma melhoria a ser aplicada seria o uso de um banco de dados mais seguro que o SQLite usado na aplicação. Foi adotado o uso desse banco de dados por sua praticidade, mas isso torna inseguro o sistema contra outros ataques. O recomendado para este caso seria o uso de um banco criptografado com *login* e senha, que apenas os administradores do sistema tenham acesso. Os *tokens* de autenticação de usuários do JWT, as senhas dos usuários e os dados pessoais dos pacientes são salvos sem segurança adicional. Sendo necessário mais estudos visando compreender e implementar melhores e mais modernas formas de armazenar estas informações na aplicação.

Quando aplicado à IoMT, é de grande interessante o servidor da aplicação estar hospedado em alguma nuvem e novos testes serem realizados considerando a transferência de dados entre o dispositivo e o endereço onde a aplicação estiver hospedada. Tendo em vista que aplicações reais de IoT são multiplataforma e o cuidado médico está evoluindo para conectar pacientes em suas casas com médicos em clínicas, seria de grande interesse para tornar esta aplicação real a hospedagem em nuvem.

Como a aplicação web foi desenvolvida como protótipo, existe muito espaço para melhorias e aprimoramentos. Estudos de como deixar a *interface* mais atrativa aos usuários, tendo em vista que o público alvo são profissionais da saúde e algumas demandas podem não estar sendo atendidas pela aplicação. Uma tela para o gerenciamento de usuários pelos administradores, pois atualmente é necessário fazer requisições diretamente no banco de dados para o cadastro de novos usuários e elaborar formas de tratar erros quando os dispositivos não estão operantes. Adequação da aplicação as novas demandas que a LGPD exige, considerando as trativas dessa lei, seria necessário adequar a aplicação as suas normas, para evitar eventuais multas.

Referências

AL-TURJMAN, F.; NAWAZ, M. H.; ULUSAR, U. D. Intelligence in the internet of medical things era: A systematic review of current and future trends. *Computer Communications*, v. 150, p. 644–660, 2020. ISSN 0140-3664. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0140366419313337>. Citado na página [6].

ARQUILLA, K.; WEBB, A. K.; ANDERSON, A. P. Textile electrocardiogram (Ecg) electrodes for wearable health monitoring. *Sensors (Switzerland)*, v. 20, n. 4, p. 1–14, 2020. ISSN 14248220. Citado na página [4].

ATES, H. C. et al. Wearable devices for the detection of COVID-19. *Nature Electronics 2021 4:1*, Nature Publishing Group, v. 4, n. 1, p. 13–14, jan 2021. ISSN 2520-1131. Disponível em: <https://www.nature.com/articles/s41928-020-00533-1>. Citado na página [13].

BANSOD, G.; RAVAL, N.; PISHAROTY, N. Implementation of a new lightweight encryption design for embedded security. *IEEE Transactions on Information Forensics and Security*, IEEE, v. 10, n. 1, p. 142–151, 2014. Citado na página [22].

BEAULIEU, R. et al. The simon and speck lightweight block ciphers. In: *Proceedings of the 52nd Annual Design Automation Conference*. New York, NY, USA: Association for Computing Machinery, 2015. (DAC '15). ISBN 9781450335201. Disponível em: <https://doi.org/10.1145/2744769.2747946>. Citado na página [20].

BENDECHACHE, M.; LE-KHAC, N.-A.; KECHADI, T. *False Data Injection Attacks in Healthcare*. Springer Singapore, 2017. v. 845. 38–56 p. ISBN 9789811302923. Disponível em: <http://dx.doi.org/10.1007/978-981-13-0292-3_12>. Citado na página [3].

BHATIA, H.; PANDA, S. N.; NAGPAL, D. Internet of things and its applications in healthcare-a survey. In: *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. Noida, India: International Conference on Reliability, Infocom Technologies and Optimization, 2020. p. 305–310. Citado (3) vezes nas páginas [6, 8 e 9].

BOGDANOV, A. *Cryptanalysis of the KeeLoq block cipher*. 2007. Cryptology ePrint Archive, Report 2007/055. <https://ia.cr/2007/055>. Citado na página [19].

BOGDANOV, A. et al. Present: An ultra-lightweight block cipher. In: PAILLIER, P.; VERBAUWHEDE, I. (Ed.). *Cryptographic Hardware and Embedded Systems - CHES 2007*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 450–466. ISBN 978-3-540-74735-2. Citado na página [21].

CHANDRA, S. et al. A comparative survey of symmetric and asymmetric key cryptography. In: *2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE)*. Kolkata, India: International Conference on Electronics, Communication and Computational Engineering, 2014. p. 83–93. Citado na página [17].

CISCO, T.; INTERNET, A. Cisco: 2020 CISO Benchmark Report. *Computer Fraud Security*, v. 2020, n. 3, p. 4–4, 2020. ISSN 1361-3723. Citado na página [5].

DIWAKER, C.; JANGRA, A.; RANI, A. Survey on iot health care techniques. In: *2019 5th International Conference on Signal Processing, Computing and Control (ISPCC)*. Solan, India: 5th IEEE International Conference on Signal Processing, Computing and Control. Citado na página [9].

DUTTA, I. K.; GHOSH, B.; BAYOUMI, M. Lightweight cryptography for internet of insecure things: A survey. *2019 IEEE 9th Annual Computing and Communication Workshop and Conference, CCWC 2019*, IEEE, p. 475–481, 2019. Citado na página [17].

FIROUZI, F.; CHAKRABARTY, K.; NASSIF, S. *Intelligent internet of things: From device to fog and cloud*. Cham, Switzerland: Springer, 2020. ISBN 978-3-030-30366-2. Citado na página [5].

FIROUZI, F. et al. Keynote paper: From EDA to IoT eHealth: Promises, challenges, and solutions. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 37, n. 12, p. 2965–2978, 2018. ISSN 02780070. Citado na página [3].

GANDHI, D. A.; GHOSAL, M. Intelligent healthcare using iot:a extensive survey. In: *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*. Pune, India: IEEE Xplore Compliant, 2018. p. 800–802. Citado na página [6].

GHUBAISH, A. et al. Recent Advances in the Internet-of-Medical-Things (IoMT) Systems Security. *IEEE Internet of Things Journal*, v. 8, n. 11, p. 8707–8718, jun 2021. ISSN 2327-4662. Disponível em: <https://ieeexplore.ieee.org/document/9298452/>. Citado na página [4].

GOYAL, T. K.; SAHULA, V. Lightweight security algorithm for low power iot devices. In: IEEE. *2016 international conference on advances in computing, communications and informatics (ICACCI)*. Jaipur, India, 2016. p. 1725–1729. Citado na página [17].

HATZIVASILIS, G. et al. A review of lightweight block ciphers. *Journal of cryptographic Engineering*, Springer, Berlin, Heidelberg, v. 8, n. 2, p. 141–184, 2018. Citado (3) vezes nas páginas [19, 20 e 22].

HATZIVASILIS, G. et al. Review of security and privacy for the internet of medical things (iomt). In: *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. Crete, Greece: 5th International Conference on Distributed Computing in Sensor Systems, 2019. p. 457–464. Citado na página [13].

HEI, X. et al. Patient Infusion Pattern based Access Control Schemes for Wireless Insulin Pump System. *IEEE Transactions on Parallel and Distributed Systems*, IEEE, v. 26, n. 11, p. 3108–3121, 2015. ISSN 10459219. Citado na página [3].

IQBAL, M. A. et al. Internet of things (iot) fundamentals. In: _____. *Enabling the Internet of Things*. John Wiley Sons, Ltd, 2021. cap. 1, p. 1–28. ISBN 9781119701460. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119701460.ch1>. Citado na página [5].

ISLAM, S. M. R. et al. The internet of things for health care: A comprehensive survey. *IEEE Access*, v. 3, p. 678–708, 2015. Citado na página [3].

JIANG, D.; SHI, G. Data security and privacy protection of wearable equipment in healthcare. 2021. Disponível em: <https://doi.org/10.1155/2021/6656204>. Citado na página [15].

JIANG, D.; SHI, G. Research on data security and privacy protection of wearable equipment in healthcare. *Journal of Healthcare Engineering*, Hindawi, v. 2021, 2021. Citado (2) vezes nas páginas [15 e 16].

LARA-NINO, C. A.; DIAZ-PEREZ, A.; MORALES-SANDOVAL, M. Elliptic curve lightweight cryptography: A survey. *IEEE Access*, IEEE, v. 6, p. 72514–72550, 2018. Citado na página [17].

MANIYATH, S. R. et al. Evaluation of mental health using iot based wearables. In: *2021 International Conference on Design Innovations for 3Cs Compute Communicate Control*

(ICDI3C). Banglalore, India: IEEE Xplorer, 2021. p. 239–242. Citado (2) vezes nas páginas [3 e 11].

MENEGHELLO, F. et al. Iot: Internet of threats? a survey of practical security vulnerabilities in real iot devices. *IEEE Internet of Things Journal*, IEEE, v. 6, n. 5, p. 8182–8201, 2019. Citado (2) vezes nas páginas [15 e 16].

MISHRA, S. S.; RASOOL, A. Iot health care monitoring and tracking: A survey. In: *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. Bhopal, India: IEEE Xplorer, 2019. p. 1052–1057. Citado na página [11].

MOHAJERANI, K. et al. Hardware Benchmarking of Round 2 Candidates in the NIST Lightweight Cryptography Standardization Process. *Proceedings -Design, Automation and Test in Europe, DATE*, v. 2021-February, p. 164–169, 2021. ISSN 15301591. Citado (2) vezes nas páginas [19 e 20].

MOHANRAJ, T. et al. A review on internet of things (iot) based pulse rate, blood pressure, body temperature monitoring system. In: *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. Coimbatore, India: IEEE Xplorer, 2020. p. 78–79. Citado na página [3].

MONTEIRO, A. et al. Diretriz para o registro de dados de pacientes na vigência da lei geral de proteção de dados (lgpd). Revista de Saúde Digital e Tecnologias Educacionais, 2021. Citado (2) vezes nas páginas [14 e 15].

MORADI, A. et al. Pushing the limits: A very compact and a threshold implementation of aes. In: SPRINGER. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Berlin, Heidelberg, 2011. p. 69–88. Citado na página [20].

MORALES, A. S.; OURIQUE, F. d. O.; CAZELLA, S. C. A Comprehensive Review on the Challenges for Intelligent Systems Related with Internet of Things for Medical Decision. In: MARQUES, G. et al. (Ed.). *Enhanced Telemedicine and e-Health: Advanced IoT Enabled Soft Computing Framework*. Cham: Springer International Publishing, 2021. p. 221–240. ISBN 978-3-030-70111-6. Disponível em: <https://doi.org/10.1007/978-3-030-70111-6_11>. Citado na página [5].

MOUHA, N.; DWORKIN, M. et al. Review of the advanced encryption standard. NIST Interagency/Internal Report (NISTIR), National Institute of Standards ..., 2021. Citado na página [20].

N, K. et al. Iot secure framework for wearable sensor data for e-health system. In: *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. Chennai, India: IEEE Xplore, 2021. p. 211–215. Citado na página [14].

NASAJPOUR, M. et al. Internet of Things for Current COVID-19 and Future Pandemics: an Exploratory Study. *Journal of Healthcare Informatics Research*, v. 4, p. 325–364, 2020. Disponível em: <https://doi.org/10.1007/s41666-020-00080-6>. Citado na página [3].

OLIVEIRA, G. M. M. d. et al. Cardiovascular statistics–brazil 2020. *Arquivos Brasileiros de Cardiologia*, SciELO Brasil, v. 115, p. 308–439, 2020. Citado na página [3].

OMETOV, A. et al. A survey on wearable technology: History, state-of-the-art and current challenges. *Computer Networks*, Elsevier, v. 193, p. 108074, 2021. Citado na página [10].

OMONIWA, B. et al. Fog/edge computing-based iot (feciot): Architecture, applications, and research issues. *IEEE Internet of Things Journal*, v. 6, n. 3, p. 4118–4149, 2019. Citado na página [8].

PAPAIOANNOU, M. et al. A survey on security threats and countermeasures in internet of medical things (iomt). *Transactions on Emerging Telecommunications Technologies*, Wiley Online Library, p. e4049, 2020. Citado (2) vezes nas páginas [13 e 14].

PERWEJ, Y. et al. A methodical analysis of medical internet of things (miot) security and privacy in current and future trends. *Journal of Emerging Technologies and Innovative Research*, v. 9, n. 1, p. d346–d371, 2022. Citado (3) vezes nas páginas [13, 16 e 17].

PEVNICK, J. M. et al. Wearable technology for cardiology: An update and framework for the future. *Trends in Cardiovascular Medicine*, Elsevier, v. 28, n. 2, p. 144–150, 2018. ISSN 18732615. Disponível em: <http://dx.doi.org/10.1016/j.tcm.2017.08.003>. Citado na página [4].

QADRI, Y. A. et al. The future of healthcare internet of things: a survey of emerging technologies. *IEEE Communications Surveys & Tutorials*, IEEE, v. 22, n. 2, p. 1121–1167, 2020. Citado na página [6].

QUER, G. et al. Wearable sensor data and self-reported symptoms for covid-19 detection. *Nature Medicine*, Nature Publishing Group, v. 27, n. 1, p. 73–77, 2021. Citado na página [13].

RAAD, M. W.; DERICHE, M.; KANOUN, O. An rfid-based monitoring and localization system for dementia patients. In: *2021 18th International Multi-Conference on Systems, Signals Devices (SSD)*. Saudi, Arabia: IEEE Xplorer, 2021. p. 1–7. Citado na página [12].

RAHMANI, A. M. et al. Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach. *Future Generation Computer Systems*, Elsevier, v. 78, p. 641–658, 2018. Citado na página [15].

RAO, S. K. et al. The AES-256 Cryptosystem Resists Quantum Attacks. *International Journal of Advanced Research in Computer Science*, v. 8, n. 3, p. 404–408, 2017. ISSN 0976-5697. Disponível em: <www.ijarcs.info>. Citado (2) vezes nas páginas [18 e 20].

SABA, T. et al. Secure and energy-efficient framework using Internet of Medical Things for e-healthcare. *Journal of Infection and Public Health*, Elsevier, v. 13, n. 10, p. 1567–1575, oct 2020. ISSN 1876-0341. Citado na página [14].

SALEM, O. et al. Man-in-the-middle attack mitigation in internet of medical things. *IEEE Transactions on Industrial Informatics*, v. 18, n. 3, p. 2053–2062, 2022. Citado na página [17].

SALLAM, S.; BEHESHTI, B. D. A survey on lightweight cryptographic algorithms. In: IEEE. *TENCON 2018-2018 IEEE Region 10 Conference*. Jeju, Korea: IEEE Xplorer, 2018. p. 1784–1789. Citado na página [20].

SELENT, D. Advanced encryption standard. *Rivier Academic Journal*, v. 6, n. 2, p. 1–14, 2010. Citado na página [20].

SEVIN, A.; MOHAMMED, A. A. O. A survey on software implementation of lightweight block ciphers for IoT devices. *Journal of Ambient Intelligence and Humanized Computing*, Springer Berlin Heidelberg, n. 0123456789, 2021. ISSN 18685145. Disponível em: <https://doi.org/10.1007/s12652-021-03395-3>. Citado na página [20].

SHAJI, S. et al. Heart lung health monitor: Remote at-home patient surveillance for pandemic management. In: *2021 IEEE Global Humanitarian Technology Conference (GHTC)*. Kochi, India: IEEE Xplorer, 2021. p. 127–130. Citado na página [11].

SHIRAI, T. et al. *The 128-bit Blockcipher CLEFIA (Extended Abstract). FSE 2007. LNCS, vol. 4593*. Heidelberg: Springer, 2007. Citado na página [22].

STELLIOS, I. et al. A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services. *IEEE Communications Surveys Tutorials*, v. 20, n. 4, p. 3453–3495, 2018. Citado na página [3].

STRIELKINA, A.; KHARCHENKO, V.; UZUN, D. Availability models for healthcare iot systems: Classification and research considering attacks on vulnerabilities. In: IEEE. *2018 IEEE 9th international conference on dependable systems, services and technologies (DESSERT)*. Kyiv, Ukraine: IEEE Xplorer, 2018. p. 58–62. Citado na página [15].

SUZAKI, T.; MINEMATSU, K. Improving the generalized feistel. In: *International Workshop on Fast Software Encryption*. Berlin, Heidelberg: Springer, 2010. p. 19–39. Citado na página [19].

THAKOR, V. A.; RAZZAQUE, M. A.; KHANDAKER, M. R. Lightweight cryptography algorithms for resource-constrained iot devices: A review, comparison and research opportunities. *IEEE Access*, IEEE, 2021. Citado (3) vezes nas páginas [18, 19 e 20].

TRNKA, M. et al. Systematic review of authentication and authorization advancements for the internet of things. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 22, n. 4, p. 1361, 2022. Citado na página [14].

VISHNU, S.; RAMSON, S. J.; JEGAN, R. Internet of medical things (iomt) - an overview. In: *2020 5th International Conference on Devices, Circuits and Systems (ICDCS)*. Guntur, India: IEEE Xplorer, 2020. p. 101–104. Citado (3) vezes nas páginas [3, 7 e 12].

WANG, Z. et al. A three-party mutual authentication protocol for wearable iot health monitoring system. In: *2021 IEEE International Conference on Smart Internet of Things (SmartIoT)*. Zhangjiakou, China: IEEE Xplorer, 2021. p. 344–347. Citado na página [14].

YU, H.; ZHOU, Z. Optimization of iot-based artificial intelligence assisted telemedicine health analysis system. *IEEE Access*, v. 9, p. 85034–85048, 2021. Citado na página [3].

ZAKARIA, H. et al. Iot security risk management model for secured practice in healthcare environment. *Procedia Computer Science*, Elsevier, v. 161, p. 1241–1248, 2019. Citado na página [15].

ZANON, V. et al. Dispositivo com interface vestível para a aquisição, processamento e transmissão do sinal cardíaco em exame de eletrocardiograma. In: *Anais do XXI Simpósio Brasileiro de Computação Aplicada à Saúde*. Porto Alegre, RS, Brasil: SBC, 2021. p. 48–59. ISSN 2763-8952. Disponível em: <https://sol.sbc.org.br/index.php/sbcas/article/view/16052>. Citado na página [4].

Zanon, V. R.; Ourique, F. de O. ***Dispositivo com Interface Wearable para a Aquisição,Processamento e Transmissão do Sinal Cardíaco em Exame de Eletrocardiograma***. 45 p. Monografia (Trabalho de Conclusão de Curso) — Universidade Federal de Santa Catarina — UFSC, Santa Catarina, 2020. Citado na página [38].

ZHU, H. et al. Smart Healthcare in the Era of Internet-of-Things. *IEEE Consumer Electronics Magazine*, Institute of Electrical and Electronics Engineers Inc., v. 8, n. 5, p. 26–30, sep 2019. ISSN 21622256. Citado (2) vezes nas páginas [7 e 8].