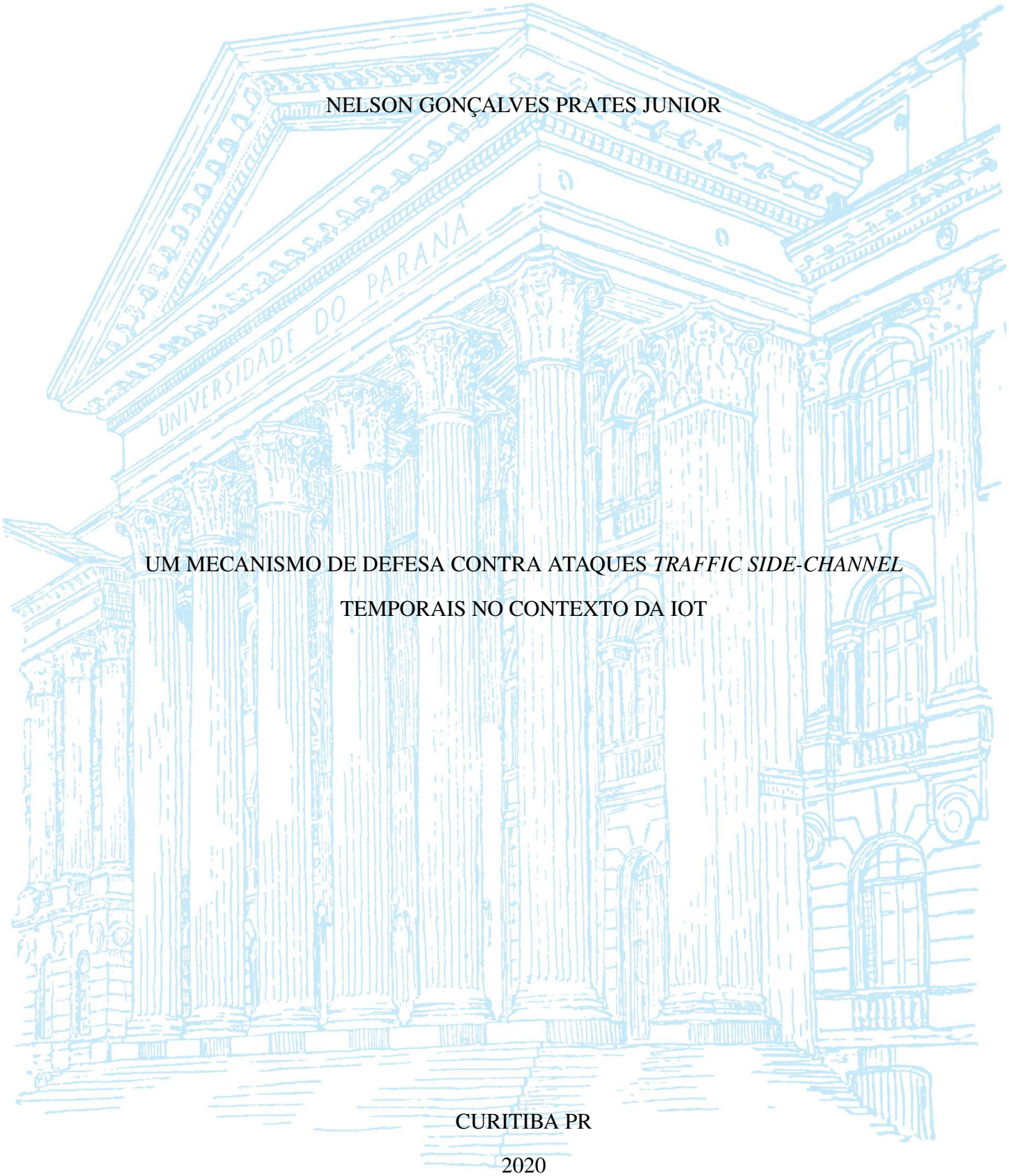
UNIVERSIDADE FEDERAL DO PARANÁ

NELSON GONÇALVES PRATES JUNIOR



UM MECANISMO DE DEFESA CONTRA ATAQUES *TRAFFIC SIDE-CHANNEL* TEMPORAIS NO CONTEXTO DA IOT

CURITIBA PR

2020

NELSON GONÇALVES PRATES JUNIOR

UM MECANISMO DE DEFESA CONTRA ATAQUES *TRAFFIC SIDE-CHANNEL* TEMPORAIS NO CONTEXTO DA IOT

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Michele Nogueira Lima.

Coorientador: Ricardo T. Macedo.

CURITIBA PR

2020

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

P912m Prates Junior, Nelson Gonçalves
Um mecanismo de defesa contra ataques *traffic side-channel temporais* no contexto da IOT/ Nelson Gonçalves Prates Junior. – Curitiba, 2020.

Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática, 2020.

Orientadora: Michele Nogueira Lima.
Coorientador: Ricardo Tombesi Macedo.

1. Internet das Coisas. 2. Vazamentos (Divulgação de informação). 3. Privacidade. I. Universidade Federal do Paraná. II. Lima, Michele Nogueira. III. Macedo, Ricardo Tombesi. IV. Título.

CDD: 004.678

Bibliotecária: Vanusa Maciel CRB- 9/1928

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **NELSON GONÇALVES PRATES JUNIOR** intitulada: **UM MECANISMO DE DEFESA CONTRA ATAQUES TRAFFIC SIDE-CHANNEL TEMPORAIS NO CONTEXTO DA IOT**, sob orientação do Prof. Dr. MICHELE NOGUEIRA LIMA, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 23 de Março de 2020.

MICHELE NOGUEIRA LIMA
Presidente da Banca Examinadora (UNIVERSIDADE FEDERAL DO PARANÁ)

RICARDO TOMBESI MACEDO
Coorientador - Avaliador Externo (UNIVERSIDADE FEDERAL DE SANTA MARIA)

ALDRI LUIZ DOS SANTOS
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

DANIEL SADOC MENASCHÉ
Avaliador Externo (UNIVERSIDADE FEDERAL DO RIO DE JANEIRO - DCC)

MINISTÉRIO DA EDUCAÇÃO · UNIVERSIDADE FEDERAL DO PARANÁ · Setor de Ciências Exatas · Coordenação de Pós-Graduação em Informática

"Be the change you want to see in the world." Mahatma Gandhi.

AGRADECIMENTOS

Primeiramente, eu gostaria de agradecer aos meus pais, Marta Medeiros Prates e Nelson Gonçalves Prates. Minha irmã Kely Angélica Medeiros Prates, meu cunhado Vinicius Martins Camargo, minha sobrinha Sophia Prates Martins e todos os demais membros da minha família. Vocês são as minhas principais fontes de motivação e energia nesta caminhada, onde apesar dos meus tropeços, prometo honrar todos os seus esforços. Também, gostaria de agradecer a todos os professores que fizeram parte da minha formação pessoal e profissional. Dentre estes professores, gostaria de destacar o Dr. Ricardo Tombesi Macedo, a Drª Michele Nogueira Lima eo Dr Aldri Luiz dos Santos por serem os meus principais guias na caminhada acadêmica. Aos meus pais de coração Noedi Lazzarotto Barbosa e Rui Barbosa por me proporcionarem momentos de conforto e descanso. Por fim, gostaria de agradecer a todos os meus amigos. Principalmente a Andressa Vergütz, que dedicou muito esforço e paciência ao impulsionar meus passos, compartilhando a experiência de como é estar nesta estrada a mais tempo. Aos demais amigos, Julia Maria Miranda, Bruno Henrique Schwengber, Ricardo Tombesi Macedo, Alisson Puska, Arthur Emilio Garcete Ferreira, Arthur Silva, Augusto Dantas, Benevid Felix Silva, Bruna Cardoso, Bruno Marquez Cremonezi, Cainã Passos, Carlos Alberto Pedroso Junior, Christian Cavalheiro, Daiana Corrêa, Danilo Rodrigo Possati, Davi Daniel Gemmer Euclides Peres Farias Junior, Everton Vilhena Cardoso, Fábio Desconsi, Fábio Miguel Knapp, Fagner Rontani, Fernando Nakayama, Gulherme Mariani, Gustavo Henrique Carvalho de Oliveira, Igor Steuck Lopes, Iulisloi Zacarias, Jerusa Grolli, Larissa Amaral da Silva, Ligia, Luiz Augusto B. Minchola, Marcos Antônio Dellazari, Mateus Boldrin, Mateus Pelloso, Matheus Batistella, Orlando Junior, Paulo Henrique Vianna, Paulo Lenz Junior, Paulo Vinicius Mendonça, Renato Melo, Sabrine Dellarmelin, Thainá Mariani, Vanessa F. Salomé, Yan Uehara de Moraes e Uelinton Brezolin por tornarem a caminhada da vida mais agradável.

RESUMO

A Internet das Coisas (IoT) visa conectar objetos à Internet para prestar serviços inovadores, como por exemplo, o monitoramento da saúde através de dispositivos vestíveis conectados. Entretanto, devido à natureza crítica dos dados transportados pela IoT somado a escassez recursos, ela vem sendo alvo de ataques que causam impactos como a comercialização e a divulgação indevida de dados privados. Em 2018 estes ataques geraram um custo médio global de US$ 3,86 milhões. Os ataques *side-channel* baseados no tráfego sondam dados como os intervalos entre pacotes, o tamanho dos pacotes, as taxas de bits, entre outros, com o objetivo de inferir informações pessoais que comprometem o direito de privacidade dos usuários da IoT. No contexto de segurança computacional, estes dados são denominados vazamentos *side-channel*, pois revelam informações a partir de dados observados por um canal marginal ao que de fato estão passando as informações almejadas. Na literatura, existem trabalhos que apresentam formas de realizar este tipo de ataque e técnicas de defesa. Entretanto, os poucos trabalhos consideram os ataques *side-channel* baseados no tráfego da IoT, ignoram as características contidas na temporização do tráfego de rede sem fio ou deixam potenciais vulnerabilidades em aberto. Este trabalho apresenta um estudo sobre os ataques *side-channel* que analisam os vazamentos temporais emitidos pelas transmissões de uma rede IoT, como os tempos de resposta, os intervalos entre as mensagens e os instantes de envio e recebimento de mensagens, para avaliar os impactos destes ataques na privacidade dos usuários. Além disso, apresenta o mecanismo FISHER (do inglês: a deFense mechanIsm against Side-cHannEl Attacks based on inteRnet of things traffic Timing) de Defesa Contra Ataques *Side-Channel* baseados na Temporização do Tráfego da IoT. O mecanismo FISHER atua como um serviço virtual e segue dois módulos para testar as vulnerabilidades e o proteger de privacidade dos dados. O módulo de teste de vulnerabilidade identifica os vazamentos temporais expostos através das técnicas de identificação de dispositivos empregadas pelos ataques e inicia o processo de defesa. O módulo de proteção de privacidade implementa as duas técnicas supracitadas em sincronia para mascarar os vazamentos temporais, mas diferente da literatura, pretende-se analisar o estado da rede. O estudo conduzido apresenta uma avaliação de desempenho sobre os ataques *side-channel* baseados na Temporização do Tráfego da IoT, considerando diferentes cenários experimentais. Os resultados apontam a relevância destes ataques, pois foi possível inferir informações sensíveis como os diferentes dispositivos e os seus sensores embarcados, considerando apenas os instantes de envio das mensagens e os tempos de resposta gerados pelo tráfego de rede. Em seguida, o mecanismo foi avaliado considerando ocultar estes vazamentos relacionados à temporização do tráfego. Os resultados revelam a eficiência das técnicas empregadas pelos módulos ao reduzindo a precisão das inferências implementada pelos ataques.

Palavras-chave: Internet das Coisas. Vazamentos Temporais. Ataques Side-Channel. Privacidade

ABSTRACT

The Internet of Things (IoT) aims to connect objects to the Internet to provide innovative services, such as health monitoring through connected wearable devices. However, due to the critical nature of the data transported by the IoT plus the scarcity of resources, it has been the target of attacks that cause impacts such as the commercialization and improper disclosure of private data. In 2018 these attacks generated an average global cost of $3.86 million. Traffic-based side-channel attacks poll data such as the intervals between packets, the size of the packets, and the bit rates, among others, to infer personal information that compromises IoT users' privacy rights. In the context of computational security, these data are called side-channel leaks, as they reveal information from data observed by a marginal channel to which the desired information is passing. The works that present ways to carry out this type of attack and defense techniques in the literature. However, few studies consider side-channel attacks based on IoT traffic, ignore the characteristics contained in the timing of wireless network traffic, or leave potential vulnerabilities open. This work presents a study on the impacts of side-channel attacks on users' privacy. The attacks analyze temporal leaks emitted by the transmissions of an IoT network, such as the response times, the intervals between the messages, and the moments of sending and receiving messages. Also, it presents the FISHER mechanism (from English: the deFense mechanIsm against Side-cHannEl Attacks based on inteRnet of things traffic Timing) for Defense Against Side-Channel Attacks based on IoT Traffic Timing. The FISHER mechanism acts as a virtual service and follows two modules to test vulnerabilities and protect users' data privacy. The vulnerability test module identifies the temporal leaks through the device identification techniques and initiates the defense process. The privacy protection module implements the two techniques above in sync to analyze the state of the network and mask the time leaks, unlike the literature. The conducted study presents a performance evaluation on side-channel attacks based on IoT Traffic Timing, considering different experimental scenarios. The results point out the relevance of these attacks since it was possible to infer sensitive information such as the different devices and their embedded sensors, considering only the moments of sending messages and the response times generated by the network traffic. Then, the mechanism evaluation considered hiding these leaks related to traffic timing. The results reveal the efficiency of the techniques of the modules by reducing the accuracy of the inferences implemented by the attacks.

Keywords: Internet of Things; Time Leaks; Side-Chanel Attack; Privacy.

LISTA DE FIGURAS

2.1 Arquitetura da IoT. 22
2.2 Classificação dos Vazamentos *side-channel* 27

4.1 Rede IoT. 40
4.2 Arquitetura do Mecanismo 42
4.3 Funções do Módulo de Teste de Vulnerabilidade. 43

5.1 Cenário Experimental CCSC 47
5.2 Cenário Experimental IoTLab 48
5.3 Etapas da Análise dos Vazamentos 51
5.4 Comportamento do Tempo de Resposta dos Dispositivos 53
5.5 Desempenho dos Classificadores com Sensores de Iluminação 54
5.6 Desempenho dos Classificadores com Sensores de Temperatura. 54
5.7 Resultados sobre as Abordagens 56
5.8 Desempenho dos Classificadores na Identificação dos Dispositivos 56
5.9 Cenário de Rede IoT Lab 57
5.10 Comportamento do Tempo de Resposta dos Dispositivos por Sensor 59
5.11 Desempenho dos Classificadores - **C3** 60
5.12 Desempenho dos Classificadores - **C4** 60
5.13 Comportamento do Tempo de Resposta dos Dispositivos por Sensor - Abordagem **A1** 62
5.14 Desempenho dos Classificadores - Abordagem A1. 63
5.15 Comportamento do Tempo de Resposta dos Dispositivos por Sensor - Abordagem **A4** 64
5.16 Desempenho dos Classificadores 65

LISTA DE TABELAS

2.1 Protocolos Padronizados para IoT 24

3.1 Categorização dos Ataques por Alvo 34
3.2 Categorização das Defesas por Técnica. 37

4.1 Terminologia 40
4.2 Medidas Estatísticas para Caracterização do Tráfego (Selis e Marshall, 2017) . . 43

5.1 Medidas Estatísticas para Caracterização do Tráfego. 48
5.2 Detalhes dos Dispositivos IoT 53
5.3 Matriz de Confusão Referente ao Classificador KNN - **C3**. 60
5.4 Matriz de Confusão *Random Forest* - **C4**. 61
5.5 Matriz de Confusão KNN - **C4** 61
5.6 Matriz de Confusão *Multilayer Perceptron*- **C4** 61
5.7 Matriz de Confusão do Classificador *Random Forest*- Abordagem **A4** 63
5.8 Matriz de Confusão do Classificador *Multilayer Perceptron*- Abordagem **A4** . . . 63

LISTA DE ACRÔNIMOS

6LoWPAN	*IPv6 over Low power Wireless Personal Area Networks* IPv6 sobre Redes de baixa Capacidade Energética e de Área Pessoal
Ack	*Acknowledgement* Reconhecimento
ARPANET	*Advanced Research Projects Agency Network* Rede da Agência para Projetos de Pesquisa Avançada
Bluetooth LE	*Bluetooth Low Energy* *Bluetooth* para Baixa Capacidade de Energia
CIA	*Central Intelligence Agency* Agência Central de Inteligência
CoAP	*Constrained Application Protocol* Protocolo de Aplicação Restrita
Con	*Confirmable* Confirmável
CSMA/CA	*Carrier Sense Multiple Access with Collision Avoidance* Acesso Múltiplo com Verificação de Portadora com Prevenção de Colisão
DFC	Dispositivo de Função Completa
DFR	Dispositivo de Função Reduzida
DODAG	*Destination-Oriented Directed Acyclic Graph* Grafo Acíclico Direcionado Orientado ao Destino
GPS	*Global Positioning System* Sistema de Posicionamento Global
IBM	*International Business Machines*
ID	Identidade
IEEE	*Institute of Electrical and Electronics Engineers* Instituto de Engenheiros Eletricistas e Eletrônicos
IETF	*Internet Engineering Task Force* Força-tarefa de Engenharia para Internet
IoT	*Internet of Things* Internet das Coisas
IPv6	*Internet Protocol version* 6 Protocolo da Internet versão 6
ISO	*International Organization for Standardization* Organização Internacional para Padronização

KNN	*K-Nearest Neighbor* K Vizinhos mais Próximos
MAC	Media Access Control Controle de Acesso ao Meio
MQTT	*Message Queuing Telemetry Transport* Protocolo de Enfileiramento de Mensagens de Telemetria de Transporte
MTU	*Maximum Transmission Unit* Unidade Máxima de Transmissão
ONU	Organização das Nações Unidas
PHY	*Physical Layer* Camada Física
REST	*Representational State Transfer* Transferência Representacional de Estado
RFC	*Request for Comments* Pedido de Comentários
RPL	*IPv6 Routing Protocol for Low-Power and Lossy Networks* Protocolo de Roteamento para Redes IPv6 com Baixa Capacidade Energética e com Perdas
Serpro	Serviço Federal de Processamento de Dados
SUS	Sistema Único de Saúde
TCP	*Transmission Control Protocol* Protocolo de Controle de Transmissão
UDP	*User Datagram Protocol* protocolo de datagrama de uso
UE	União Européia
USB	*Universal Serial Bus* Barramento Serial Universal

LISTA DE SÍMBOLOS

T	Instante de tempo
G	Gateway de rede
d	Um Dispositivo
D	Subconjunto de Conjunto de Dispositivos
C	Subconjunto de Canais
C	Conjunto de Canais entre o Gateway e um Dispositivo Final
c	Um Canal
Tr	Transmissões
V	Conjunto de Características Estatísticas
VP	Verdadeiro Positivo
FP	Falso Positivo
VN	Verdadeiro Negativo
FP	Falso Positivo
min	Mínimo
max	Máximo
sum	Soma
μ	Média
LI	Limite Inferior
LS	Limite Superior
$mode$	Moda
r	Correlação de Pearson
X	Conjunto de Atrasos
$\hat{X}$	Conjunto de Atrasos Modificados
N	Rede IoT
m	Mensagem
req	Requisição
$resp$	Resposta
$Traf$	Tráfego de rede
Tr	Transmissão
$\mathcal{P}$	Família de Distribuições
p_ν	Distribuição
P	Probabilidade
$F1$	Primeira Função do Módulo de Teste de Vulnerabilidade
$F2$	Segunda Função do Módulo de Teste de Vulnerabilidade
$F3$	Terceira Função do Módulo de Teste de Vulnerabilidade

A	Conjunto de Amostras de Tempo de Resposta
η	Limiar de Requisições Capturadas
τ	Tempo de Resposta
w	Precisão
X	Variáveis de Tempo de Resposta
χ	Conjunto de Possíveis tempos de Resposta
γ	Atraso Inserido

SUMÁRIO

1 INTRODUÇÃO 15
1.1 MOTIVAÇÃO 16
1.2 DEFINIÇÃO DO PROBLEMA 17
1.3 OBJETIVO 19
1.4 CONTRIBUIÇÕES 20
1.5 ESTRUTURA DO TRABALHO 20
2 FUNDAMENTOS 21
2.1 INTERNET DAS COISAS 21
2.1.1 Arquitetura da IoT 22
2.1.2 Os Principais Protocolos para a IoT 24
2.1.3 Computação em Nuvem e IoT 26
2.2 PRIVACIDADE DIANTE DOS ATAQUES *SIDE-CHANNEL* 26
2.2.1 Definição dos Ataques 28
2.2.2 Métodos de Defesa 30
2.3 RESUMO 31
3 TRABALHOS RELACIONADOS 32
3.1 ATAQUES TRAFFIC *SIDE-CHANNEL* 32
3.2 DEFESAS A ATAQUES *TRAFFIC SIDE-CHANNEL* TEMPORAIS 36
3.3 RESUMO 38
4 UM MECANISMO DE DEFESA CONTRA ATAQUES TRAFFIC *SIDE-CHANNEL* TEMPORAIS NA INTERNET DAS COISAS 39
4.1 VISÃO GERAL 39
4.1.1 Modelo de Rede 39
4.1.2 Modelo do Ataque 41
4.2 DETALHAMENTO DO MECANISMO 41
4.2.1 Módulo de Teste de Vulnerabilidade 41
4.2.2 Proteção de Privacidade 44
4.3 RESUMO 45
5 AVALIAÇÃO 46
5.1 METODOLOGIA DE AVALIAÇÃO 46
5.1.1 Cenários Experimentais 46
5.1.2 Métricas de Avaliação 47
5.1.3 Caracterização do Tráfego 48
5.1.4 Métodos de Defesa 49

5.2 ANÁLISE DOS VAZAMENTOS TEMPORAIS 50
5.2.1 Avaliação de Desempenho - Cenário CCSC 50
5.2.2 Avaliação de Desempenho - Cenário IoTLab 57
5.3 RESUMO 65
6 CONCLUSÃO 67
6.1 TRABALHOS FUTUROS 67
6.2 PUBLICAÇÕES 67
REFERÊNCIAS 69

1 INTRODUÇÃO

A evolução tecnológica das redes de sensores sem fio culminou no surgimento da Internet das Coisas (do inglês, *Internet of Things* - IoT) potencializando a revolução para a era digital. Os dispositivos IoT, também conhecidos como "coisas", interconectam-se para desempenhar atividades muitas vezes sem a necessidade da intervenção humana. Estas atividades possibilitam diversas aplicações, tais como o monitoramento da saúde, do trânsito, a automação de atividades industriais, entre outros. Como exemplo didático, a coleta de dados como temperatura do ambiente, a intensidade do trânsito em uma cidade e a pressão arterial de um motorista, submetidos às técnicas de correlação para avaliá-los em tempo real e, assim, confirmar ou refutar a incidência de uma possível emergência de saúde com antecedência (Vergütz et al., 2017). O fundo de investimento lançado em parceria entre Ministério da Ciência, Tecnologia, Inovações e Comunicações (MCTIC), o Banco Nacional de Desenvolvimento Econômico e Social (BNDES) e a empresa Qualcomm motiva o desenvolvimento de pesquisas relacionadas à IoT e mostra o interesse de empresas públicas (ASCOM, 2019b). Desta forma, a IoT proporciona novos serviços, disponibiliza recursos, possibilita a criação de cidades mais inteligentes, além de ser atraente para os negócios devido ao volume de dados gerados que permite a melhor compreensão do comportamento de consumo, interesses pessoais, entre outros.

As aplicações mais populares para a IoT nos dias de hoje compreendem soluções para a automação de atividades humanas devido a problemas sociais relacionados aos cuidados com a saúde e a segurança pública. Este fato motivou o surgimento de novos conceitos como a indústria 4.0 (ou indústria inteligente), cidades inteligentes, casas inteligentes e *eHealth*. Neste contexto, os dispositivos monitoram o cotidiano dos seus usuários através da coleta de dados, transmissão e recebimento de mensagens pela Internet. Estes dispositivos englobam sensores vestíveis, equipamentos domésticos, *smartphones*, semáforos, câmeras IP, entre outros, que são conectados a servidores responsáveis pelo processamento dos dados coletados para fornecer informações visando a uma melhor qualidade de vida dos usuários.

Na IoT qualquer objeto é capaz de se comunicar através de redes sem fio. Estes objetos são equipados com microcontroladores, placas de rede, sensores e/ou atuadores para coletar, transmitir dados do ambiente e interagir com ele. Deste modo, os objetos atuam como sensores, atuadores e *gateways*. Os sensores captam informações, os atuadores interagem com o ambiente e o *gateway* age como uma entidade central de comunicação, que permite a interação entre os dispositivos locais e os conectados à Internet. Toda esta organização envolve a cooperação entre diferentes tecnologias de comunicação, protocolos e serviços, tornando o *gateway* um agente que também administra a heterogeneidade da IoT. Esta estrutura recebe requisições ou comandos para os dispositivos em rede, a fim de fornecer respostas com a informação desejada, conforme pré-estabelecido pelos protocolos de rede. Neste contexto, os provedores de serviços para a Internet utilizam a IoT como meio para coletar e processar dados sobre os usuários com o intuito de fornecer informações úteis e auxiliar os seus clientes.

A mobilidade e a portabilidade dos dispositivos IoT prejudicam o envio e o recebimento de mensagens. Estas características levam a problemas como redes com perdas de pacotes compostas por dispositivos com baixa capacidade computacional e alimentação de energia limitada (Alduais et al., 2016). Assim, implementar os protocolos padronizados para a Internet tradicional, os quais foram projetados para dispositivos mais robustos e com conexões confiáveis, se torna inviável na maioria dos cenários IoT (Prates et al., 2018). Diante destes desafios, a Força Tarefa de Engenharia para Internet (do inglês, *Internet Engineering Task Force* - IETF)

padronizou uma série de protocolos que levam em consideração as características supracitadas. Isto motiva o desenvolvimento de novos serviços através da disponibilidade de recursos que ampliam a capacidade de gerenciamento remoto e automático, possibilitando a criação de aplicações mais inteligentes e independentes da iteração humana.

As características supracitadas somadas à diversidade e à quantidade de dispositivos IoT resultam em um grande volume de dados, como os dados sensíveis relacionados à saúde e à segurança dos usuários, exigindo um nível maior de privacidade. Em 2018, foi registrado um aumento de 600% de ataques envolvendo dispositivos IoT (Symatec, 2018), evidenciando a necessidade de soluções técnicas. O conceito de segurança na Internet ainda é recente, pois quando o Departamento de Defesa dos EUA apresentou a criação da primeira versão da Internet (a ARPANET), em 1960, este requisito não foi considerado. Desde então, a segurança dos dispositivos baseados na Internet não acompanhou o ritmo acelerado da inovação (Wu, 2018). A IoT é mais suscetível a ataques do que a Internet tradicional, devido ao meio de comunicação sem fio e às limitações de recursos dos dispositivos, como baixa energia, capacidades computacionais limitadas, conexão através de canais com perdas, entre outras características (Cervantes et al., 2014; Park et al., 2011). Entretanto, a utilidade das aplicações e serviços não pode representar um risco aos direitos de segurança e privacidade para seus usuários, exigindo soluções eficientes para estas características.

1.1 MOTIVAÇÃO

Nos últimos anos, a privacidade diante da era digital vem sendo discutida nas principais organizações do mundo, como na Organização das Nações Unidas (ONU) e na União Europeia (UE). Este fato foi motivado pelo caso no qual, em 2013, um agente da Agência Central de Inteligência (do inglês, *Central Intelligence Agency* - CIA) dos Estados Unidos vazou informações sigilosas do país sobre alguns dos programas de vigilância utilizados para espionar a população americana (usando servidores de empresas como Google, Apple e Facebook) e de outros países, entre eles o Brasil (Grupo Globo, 2013). No mesmo ano, a ONU na III Comissão da 68ª Assembleia Geral das Nações Unidas aprovou, por consenso, o projeto de resolução "O direito à privacidade na era digital", apresentado pelos países Brasil e Alemanha (Ministério de Relações Exteriores, 2013). A UE e o Brasil tomaram posições ainda mais críticas, através do "Lei Geral de Proteção de Dados" e o "Marco Civil da Internet", respectivamente, sancionaram uma série de normas sobre a privacidade dos usuários de provedores de serviços e aplicações através da Internet ligadas aos seus territórios ou mercados. Desta forma, os provedores que apresentarem irregularidades estarão sujeitas a multas de até 4% do volume do mercado mundial (Comissão Européia, 2018). Estes fatos evidenciam a necessidade de medidas que esclareçam para a sociedade a importância do direito de privacidade na era digital. Além disso, as defesas transparentes que permitam os usuários escolher a forma em que seus dados são explorados e utilizados por terceiros.

O problema de privacidade na IoT se agrava pois os dispositivos monitoram e controlam dados privados sobre a rotina dos usuários de forma autônoma. O *smartphone* é um bom exemplo de um dispositivo que coleta e transmite dados privados sobre seus usuários pela Internet. Ele pode ser equipado com sensores de GPS, câmera, entre outros e permite também agregar funcionalidades por meio da conexão com outros dispositivos IoT, oferecendo uma experiência ainda mais imersiva para seus usuários. No geral, estes dispositivos são controlados por sistemas operacionais que por padrão provêm serviços através da Internet de localização, pesquisa e armazenamento de dados. No entanto, os dados trafegados pelos dispositivos da IoT, se capturados e interpretados podem expor seus usuários a problemas sérios de privacidade e

segurança pessoal. A principal forma de capturar estes dados privados é através de ataques que exploram a inocência dos usuários e as brechas de segurança que podem existir nos sistemas de computação e comunicação. Estes dados são adquiridos com o objetivo de lucrar com a venda de informações, chantagem ou apelo político.

Em 2018, os impactos da comercialização e divulgação indevida de dados privados representaram um custo médio global de 3,86 milhões de dólares, além disso, o Brasil apresenta uma das maiores taxas de riscos de violação de dados com 43%, comparados a 27% da média global (Ponemon, 2018). O esquema milionário de venda de dados pessoais pelo Serviço Federal de Processamento de Dados (SERPRO) (Marília Marques, 2018) e a divulgação das informações pessoais, como nome completo e endereço dos usuários do Sistema Único de Saúde (SUS) (Márcio Padrão, 2018), em 2018, reforçam estes números. Os impactos dos vazamentos de dados não representam unicamente prejuízos financeiros, eles podem levar aos mais variados tipos de problemas. Por exemplo, eles podem influenciar nas decisões públicas de um país, como no caso em que ocorreu a publicação das conversas entre um juiz federal e um procurador de justiça. As conversas colocam em questão a legitimidade das decisões tomadas pelo juiz ao julgar um caso de corrupção, influenciando um possível golpe político. Portanto, os efeitos dos ataques contra o direito de privacidade geram impactos financeiros, políticos e sociais.

Em contrapartida, as iniciativas governamentais despertaram um maior interesse da população pela segurança e privacidade na era digital. A principal prova disso é o decreto, aprovado pelo atual presidente Jair Bolsonaro em junho de 2019, que institui o Plano Nacional de Internet das Coisas (ASCOM, 2019a). Neste plano, estudam-se os impactos da IoT na sociedade brasileira e incentiva-se a criação de sistemas de certificação para a segurança da informação na IoT considerando as características dos dispositivos. Isto, aliado ao Marco Civil da Internet, determina como as empresas provedoras de Internet e de aplicações devem tratar os dados de acesso e navegação dos usuários. O Plano Nacional de Internet das Coisas e o Marco Civil da Internet demonstram a relevância do assunto e o incentivo no desenvolvimento de soluções de segurança virtual relacionados ao controle e gestão de dados privados. Além disso, o Marco Civil garante a neutralidade da Internet, ou seja, as empresas que fornecem a conexão não devem ter acesso aos conteúdos acessados pelos usuários. Os principais alvos destas medidas de segurança são empresas que se aproveitavam das brechas na lei para comercializar os dados de acesso e navegação dos clientes. Entretanto, apesar das iniciativas dos órgãos públicos para normatizar e reforçar a defesa pelo direito de privacidade, é evidente a necessidade de soluções técnicas que compreendam as características específicas da IoT.

1.2 DEFINIÇÃO DO PROBLEMA

Os dispositivos da IoT possibilitam o vazamento dos dados ao coletar, processar e transmitir informações (Yan et al., 2017; Xiong et al., 2018). Os dados são gerados a partir dos rastros deixados pelo funcionamento dos dispositivos. Eles refletem os comportamentos dos algoritmos executados pelos dispositivos para desempenhar atividades como a coleta e a transmissão de dados. Estas atividades são controladas por blocos de códigos que seguem implementações de protocolos, serviços e aplicações. Os códigos são implementados em *loops* e ramificações, no qual, ao serem executados, geram dados capturáveis, tais como a ativação ou desativação de componentes embarcados, diferentes consumos de energia, tempos de execução, entre outros. Estes dados, podem ser coletados e explorados pelos ataques *side-channel*, a fim de identificar características sobre como os dispositivos operam, quais os seus objetivos, suas capacidades, aplicações em execução, rotina de utilização, entre outras. Assim, os vazamentos são submetidos a análises estatísticas e correlações com outros conjuntos de dados conhecidos,

podendo revelar informações privadas como a rotina, costumes e localização dos usuários quebrando a sua privacidade.

A maior parte das soluções de defesas encontradas na literatura empregam técnicas fundadas nos pilares da segurança computacional como confidencialidade, disponibilidade e integridade (Avizienis et al., 2004). O pilar da confidencialidade define que as informações só devem ser disponibilizadas para indivíduos, entidades ou processos autorizados. No pilar da integridade garante-se que a informação será transmitida sem sofrer alterações indevidas. O pilar de disponibilidade tem a responsabilidade de manter um serviço ou aplicação disponível o máximo de tempo possível, mesmo sob a incidência de falhas, sejam elas de *hardware* ou de *software*. Estes pilares apoiam o fato de que os sistemas computacionais devem operar sem oferecer riscos aos usuários, protegendo unicamente o conteúdo das mensagens trafegadas na rede e não o tráfego em si. Elas compreendem soluções baseadas em modelos de autenticação, detecção de intrusão, controle de acesso, criptografias e outros (Lima et al., 2009). Os ataques *side-channel* utilizam técnicas como análises estatísticas que driblam estas soluções, objetivando os diferentes comportamentos entre os dispositivos sem precisar acessar o conteúdo das mensagens.

Mesmo protegendo as informações trafegadas através das soluções acima, os vazamentos temporais do tráfego de rede não são considerados pelos modelos empregados por elas. Os ataques *side-channel* baseados na temporização do tráfego levam à quebra da privacidade dos usuários utilizando apenas os instantes das trocas de mensagens entre os dispositivos (Yan et al., 2017). Para este fim, os atacantes, mediante *sniffers*, capturam as transmissões de rede e extraem os instantes das trocas das mensagens. Estes dados são denominados vazamentos temporais que servem como base de cálculo para algumas características da rede como tempo médio de resposta e atraso entre os pacotes, refletindo o comportamento dos usuários por meio da utilização dos programas em execução que controlam os dispositivos. Dessa forma, os atacantes empregam algoritmos de classificação para processar os dados de tal forma que represente e identifique estes comportamentos. Assim, o atacante treina um classificador para analisar outras capturas de rede e identificar os dispositivos com comportamentos semelhantes. Portanto, este trabalho foca na confidencialidade das informações que podem ser obtidas a partir destas análises e não diretamente das informações trafegadas no conteúdo das mensagens. Este tipo de ataque quando comparado a outros ataques é diretamente nocivo aos usuários, visto que o atacante não precisa atingir os dispositivos ou os algoritmos de criptografia para adquirir informações privadas.

Na literatura, os vazamentos *side-channel* são explorados em diferentes perspectivas. O mais popular é o ataque que se beneficia dos vazamentos relacionados ao consumo de energia para inferir, através de análises estatísticas, informações críticas sobre os algoritmos de criptografia, como identificar a chave criptográfica. No entanto, estes tipos de ataques exigem a posse do dispositivo alvo, dificultando a execução em um cenário realístico. Em contraste, os ataques *side-channel* no tráfego de rede se aproveitam de dados contidos em um registro de tráfego de rede, como o tamanho dos pacotes, atrasos, entre outros, para inferir informações remotamente. Existem diversos trabalhos que abordam este tipo de ataque (Veysset et al., 2002; Srinivasan et al., 2008; Feghhi e Leith, 2016; Saltaformaggio et al., 2016; Apthorpe et al., 2017; Conti et al., 2018; Sivanathan et al., 2018; Taylor et al., 2018; Xiong et al., 2018). Porém, poucos exploram em redes compostas pelos protocolos padronizados para as características da IoT como baixa capacidade energética e redes com perdas (Yan et al., 2017; Prates et al., 2019a,b).

Identificar padrões comportamentais a partir da temporização das transmissões no tráfego de rede é a essência dos ataques *side-channel* considerados neste trabalho. Os dados relacionados ao tempo identificados em capturas de tráfego podem revelar informações sobre o comportamento dos dispositivos e seus usuários. Veysset et al. (2002) caracterizaram sistemas operacionais através de análises sobre o tempo entre as mensagens trafegadas em rede conforme

o protocolo na camada de transporte. Feghhi e Leith (2016) classificaram páginas *web* utilizando somente a informação de tempo capturada através dos instantes de envio e recebimento de mensagens em rede. Malik et al. (2017) identificaram diferentes características dos sistemas operacionais móveis para classificar as aplicações ativas. Através disso, os autores revelaram se o usuário realiza tratamento para AIDS, pois a caracterização e a identificação de aplicações de *smartphones* com precisão revelam a utilização de um aplicativo para controle medicinal. Na IoT, as análises realizadas sobre os dados temporais são ainda mais prejudiciais devido à diversidade de dispositivos IoT, como dispositivos vestíveis para cuidados com a saúde, equipamentos de segurança, monitoramento, eletrodomésticos, entre outros. Estes dispositivos estão diretamente relacionados ao cotidiano dos usuários e se atacados com efetividade podem revelar informações ainda mais críticas.

A solução mais comum entre os trabalhos que propuseram mecanismos de defesa para os ataques *side-channel*, é baseada em métodos para mascarar os vazamentos das análises estatísticas realizadas pelos atacantes (Patranabis et al., 2018). Os autores Li et al. (2017) e Yu e Köse (2017) modificaram os algoritmos de criptografia para que o consumo de energia de cada operação criptográfica não represente uma distribuição que revele características. Apesar destes trabalhos dificultarem a identificação dos vazamentos *side-channel*, eles não consideram o mesmo tipo de ataque e os vazamentos capturados a partir do tráfego de rede. Neste sentido, Xiong et al. (2018) implementaram uma técnica para ocultar o vazamento relacionado aos tamanhos dos pacotes e impedir que informações estatísticas sejam extraídas destes dados. Nestes ataques exploram-se diferentes propriedades exigindo uma técnica específica para cada uma delas, pois as soluções que evitam os vazamentos *side-channel* de tamanho dos pacotes normalmente não são as mesmas que evitam os vazamentos temporais. Existem poucos trabalhos que abordam soluções para os ataques *side-channel* baseados na temporização do tráfego no contexto dos protocolos padronizados para a IoT com baixa capacidade de recursos computacionais e energéticos (Yan et al., 2017). Isto mostra a necessidade de uma solução que explore as variáveis relacionadas ao tempo dos protocolos da IoT, a fim de evitar a ocorrência destes ataques.

As principais técnicas que mascaram os vazamentos temporais são baseadas na geração de pacotes falsos e a inclusão de atrasos de rede. Esses trabalhos abordam ocultar a frequência em que os dispositivos transmitem mensagens (Srinivasan et al., 2008; He et al., 2016, 2017; Apthorpe et al., 2019). O método mais eficiente para ocultar vazamentos baseados em tempo está na transmissão de pacotes falsos. Este método gera pacotes falsos para manter a frequência do tráfego e ocultar os padrões de tráfego originais de ataques. No entanto, isso não é suficiente porque há vazamentos cruciais contidos no tempo de resposta que revelam informações sobre o hardware dos dispositivos (por exemplo, os sensores embarcados). Esses vazamentos colocam em risco a privacidade do usuário. Portanto, é essencial projetar um novo mecanismo de defesa para mascarar vazamentos baseados em temporização do tráfego de forma eficiente.

1.3 OBJETIVO

Este trabalho tem como objetivo geral melhorar a confidencialidade dos dados sobre o tráfego de rede sem fio na IoT e a privacidade das informações sobre os usuários. De forma específica, este trabalho visa estudar os vazamentos temporais e prevenir que as análises realizadas pelos ataques *side-channel* baseados na temporização do tráfego da IoT sejam eficazes. Assim, este trabalho responde as seguintes questões de pesquisa: (i) Qual o impacto das análises realizadas sobre os vazamentos *side-channel* na privacidade da rede? (ii) Qual a eficácia ao implementar os métodos de geração de pacotes falsos e inclusão de atrasos no contexto da IoT? Portanto, este trabalho desenvolve um estudo sobre os vazamentos temporais para avaliar os

impactos na privacidade dos usuários e responder a primeira pergunta. Por fim, baseado no estudo, desenvolver uma solução para empregar os métodos de geração de pacotes falsos e inclusão de atrasos no contexto da IoT, a fim de mascarar os vazamentos *side-channel* e responder a segunda pergunta.

1.4 CONTRIBUIÇÕES

A contribuição deste trabalho compreende o mecanismo FISHER (do inglês: a deFense mechanIsm against Side-cHannEl Attacks based on inteRnet of things traffic Timing) de Defesa Contra Ataques *Side-Channel* baseados na Temporização do Tráfego da IoT. Desta forma, o mecanismo FISHER se baseia nas técnicas empregadas pelo estudo objetivado para identificar e mascarar os vazamentos temporais e, assim, empregar os métodos de geração de pacotes falsos e inclusão de atrasos para dificultar a aquisição dos dados relacionados a informações cruciais sobre a privacidade dos usuários, como nos ataques *side-channel* baseados na temporização do tráfego da IoT. O mecanismo FISHER segue dois módulos: teste de vulnerabilidades e proteção de privacidade. O primeiro módulo coleta o tráfego de rede, extrai os instantes das trocas das mensagens e calcula o tempo de resposta por requisição. Em seguida, ele divide em amostras para a caracterização e identificação dos vazamentos temporais *side-channel* (Prates et al., 2019a). O módulo de proteção de privacidade define regras individuais para os dispositivos a fim de mascarar o comportamento, ocultando os vazamentos temporais das análises estatísticas realizadas pelos ataques *traffic side-channel* (Prates et al., 2019b). Este módulo controla as variáveis relacionadas ao tempo através da aplicação de duas técnicas baseadas na literatura, a de geração de pacotes falsos e a inclusão de atrasos de rede.

O mecanismo proposto foi avaliado em diferentes cenários experimentais de redes sem fio de área pessoal, executando os principais protocolos padronizados para a IoT. Esta avaliação foi dividida em duas etapas, a primeira etapa analisa a eficiência do módulo de teste de vulnerabilidade ao identificar os dispositivos que apresentam vazamentos temporais e responde a questão de pesquisa (i), confirmando a existência de vazamentos temporais significativos. A segunda etapa avalia a eficiência de três abordagens executadas pelo módulo de proteção de privacidade, ao ocultar os vazamentos temporais *side-channel* conforme os métodos empregados pela literatura. Os resultados mostram que as técnicas presentes na literatura, gerar pacotes falsos e manipular o tempo de resposta, comprometem a identificação dos vazamentos e, com isso, dificultam a realização dos ataques *traffic side-channel*, o que responde a questão de pesquisa (ii). Nesta avaliação, o módulo de teste de vulnerabilidade oferece informações detalhadas que possibilitam o processamento das variáveis temporais, e consequentemente provê informações precisas para o módulo de proteção de privacidade atuar de forma eficiente.

1.5 ESTRUTURA DO TRABALHO

O restante deste manuscrito está dividido em cinco capítulos. O Capítulo 2 apresenta os principais fundamentos para compreensão do funcionamento da IoT, seus principais protocolos padronizados, sua relação com a Internet e seus serviços fornecidos. Além disso, o capítulo descreve como os ataques *traffic side-channel* temporais ferem o direito de privacidade dos usuários da IoT e fundamenta as principais técnicas de defesa. O Capítulo 3 apresenta uma revisão bibliográfica sobre os trabalhos que implementam os ataques *traffic side-channel* em geral e o estado da arte sobre as defesas para os ataques que consideram os vazamentos temporais. O Capítulo 4 descreve o mecanismo proposto, sua arquitetura e os modelos de rede e ataque. O Capítulo 5 detalha as avaliações sobre o mecanismo e discute os resultados. Por fim, o Capítulo 6 apresenta as conclusões e os trabalhos futuros.

2 FUNDAMENTOS

Este capítulo apresenta os principais fundamentos das redes IoT e como os ataques *side-channel* podem atingir os dispositivos que as compõem. A Seção 2.1 introduz os principais conceitos e terminologias da IoT, sua arquitetura, protocolos e a sua relação com a computação em nuvem. A Seção 2.2 discute como a privacidade dos usuários da IoT pode ser quebrada pelos ataques *side-channel* e fundamenta as técnicas de defesa relacionadas com as metodologias de gerência de redes.

2.1 INTERNET DAS COISAS

Introduzido por Kevin Ashton no ano de 1998, o novo paradigma chamado *Internet of Things* (IoT) vem ganhando cada vez mais a atenção da academia e da indústria (Bandyopadhyay e Sen, 2011). Este paradigma compreende objetos comuns (coisas) sejam equipados com transceptores, sensores e atuadores, capazes de se interconectarem, gerando novas formas de comunicação entre humanos e coisas. Nos dias de hoje, estas coisas na sociedade graças ao desenvolvimento das tecnologias de comunicação em larga escala, da redução do tamanho e custo de sensores, atuadores e dispositivos computacionais. Assim, estas coisas em forma de produtos e aplicações automatizam atividades do cotidiano das pessoas. No entanto, questões como a interoperabilidade entre os dispositivos e a segurança no transporte e manipulação de informações ainda tem motivado a academia no desenvolvimento de soluções para este novo paradigma de rede que é a IoT. Desta forma, esta seção apresenta uma visão detalhada da IoT, sua arquitetura e os seus protocolos de rede padronizados.

O paradigma de Internet de Todas as Coisas (*Internet of Everything* – IoE) expande o conceito de IoT, pois visa proporcionar serviços ainda mais relevantes para as pessoas. A IoT, muitas vezes considerada como um sinônimo de IoE, representa uma vasta gama de dispositivos (coisas) capazes de se conectar à Internet para prover serviços inteligentes por meio da troca de uma grande quantidade de dados em tempo real (Atzori et al., 2010). Estas coisas podem ser computadores de bordo de um veículo, *smartphones*, refrigeradores ou fontes de energia elétrica, entre outros. No entanto, a IoE estende o conceito de IoT ao considerar a associação de pessoas, processos, dados e coisas (Schatten et al., 2016). Por meio desta associação, a IoE explora a relação entre estas entidades, gerando oportunidades econômicas sem precedentes para empresas, indivíduos e países (Miraz et al., 2015). Entretanto, o advento da IoE está intrinsecamente atrelado com a solução de questões abordadas pela IoT (Iannacci, 2018), sendo um dos principais pontos o gerenciamento do número massivo de coisas e sua integração com a Internet atual (Lamaazi et al., 2014).

Este novo modelo de rede conecta os mais variados dispositivos computacionais à Internet, exigindo redes flexíveis com suporte a um alto nível de escalabilidade e heterogeneidade. Estas características resultam em grandes quantidades de dados e potencializam desafios relacionados a *Big Data* e dados em *stream*. Além disso, devido a exigência de portabilidade e mobilidade, estas redes são compostas por dispositivos menores, possuindo conexões com perdas e dispositivos com características específicas, como escassez de recursos computacionais e energéticos. Por consequência, implementar os protocolos já padronizados pela Internet, que foram projetados para dispositivos mais robustos e conexões confiáveis, se torna inviável na maioria dos cenários IoT. Em vista disso, os grupos de padronização mais influentes, como a *Internet Engineering Task Force* (IETF), a *International Organization for Standardization*

(ISO) e o *Institute of Electrical and Electronics Engineers* (IEEE), padronizaram uma série de protocolos com o objetivo de atender aos dispositivos com recursos limitados da IoT (Alexander et al., 2012; Prates et al., 2018). Entretanto, estes protocolos ainda são recentes e demonstram possibilidades de aprimoramentos nas áreas de segurança e privacidade dos dados.

O principal desafio para o desenvolvimento de soluções para as redes IoT consiste na limitação da capacidade dos recursos computacionais dos dispositivos. As principais soluções para os desafios da IoT estão fundadas em tecnologias como a computação em nuvem, pois ela trata os desafios relacionados a *Big Data* e fornece recursos computacionais pela Internet. A fim de detalhar os principais conceitos da IoT e a sua respectiva relação com a computação em nuvem, a Subseção 2.1.1 apresenta a arquitetura conceitual de rede para IoT. A Subseção 2.1.2 descreve os principais protocolos padronizados para a IoT. Por fim, a Subseção 2.1.3 aborda a relação entre a computação em nuvem e a IoT.

2.1.1 Arquitetura da IoT

A IoT suporta uma ampla variedade de aplicações e vem sendo apoiada pela evolução das tecnologias de rede, protocolos, meios de comunicação, dispositivos e serviços de rede. No entanto, não existe um consenso sobre o modelo arquitetural a seguir. Na literatura existem diversos estudos, discussões e propostas para modelos adequados que abordam a divisão em camadas, inspirados no modelo TCP/IP (Miao Wu et al., 2010; Bandyopadhyay e Sen, 2011; Yang et al., 2011; Khan et al., 2012; Chen et al., 2018b). Este trabalho usou como referência o modelo de três camadas (Yan e Huang, 2009; Zhao e Ge, 2013). O modelo é dividido em camadas de **percepção**, **rede** e **aplicação**, como ilustra a Figura 2.1. A camada de percepção compreende dispositivos sensores e/ou atuadores que coletam dados e interagem com o ambiente físico. A camada de rede realiza as comunicações de dispositivo-a-dispositivo, para encaminhar as informações coletadas pela camada anterior. A camada de aplicação utiliza as informações adquiridas, tratadas e encaminhadas, respectivamente, pelas camadas de percepção e rede. Esta organização em camadas suporta o desempenho de atividades direcionadas aos requisitos de diferentes contextos, como casas inteligentes, cidades inteligentes e outros.

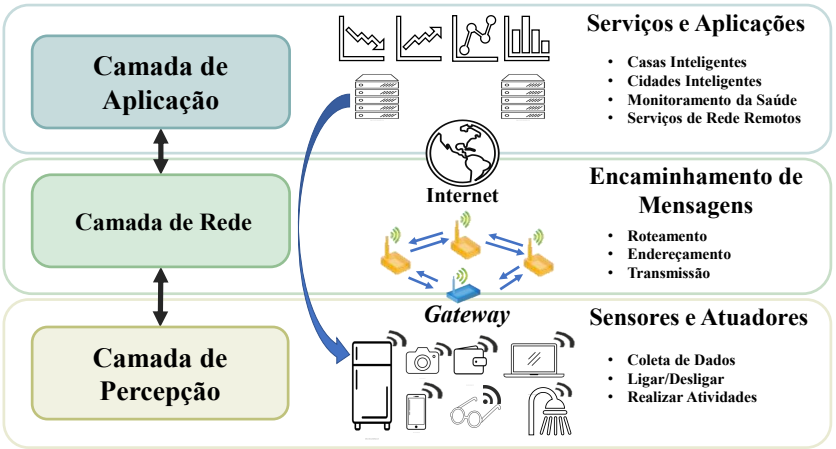


Figura 2.1: Arquitetura da IoT

2.1.1.1 Camada de Percepção

A camada de percepção define como os dispositivos interagem com o ambiente. Estes dispositivos são equipados com sensores e/ou atuadores. Os sensores captam os dados do ambiente

e os enviam para serem interpretados por um dispositivo com capacidade de processar esses dados e obter informações. Os atuadores interagem com o ambiente fisicamente, visto que eles recebem comandos para realizar determinadas atividades, como ligar uma lâmpada inteligente. Os dispositivos providos de maior capacidade computacional são capazes de captar, processar os dados, interagir com o ambiente e/ou compartilhar suas ações com a rede. Desta forma, eles podem assumir o papel de *gateway*, executando atividades de gerenciamento da rede. As aplicações para casas inteligentes implementam redes IoT através dos móveis e eletrodomésticos, que são equipados com sensores de temperatura, ruído, temporizadores, entre outros. Por exemplo, uma geladeira pode identificar a falta de insumos, gerar uma lista de compras e enviar para o *smartphone* do usuário. Os dispositivos pessoais podem ser equipados com sensores de localização (GPS) ou acompanhamento cardíaco e assim, promover um acompanhamento contínuo da saúde do usuário. Os dispositivos para controle de energia elétrica podem identificar oscilações na transmissão de energia e desligar os dispositivos, ou controlar o consumo excessivo de energia elétrica. Para diminuir os custos de mão de obra ou até otimizar a produção, as aplicações para fábricas inteligentes utilizam dispositivos como braços eletrônicos, equipados com sensores e atuadores de movimentação que recebem ordens remotamente. Os dispositivos de segurança permitem monitoramento à distância, desta forma câmeras IP capturam imagens e as transmite através da rede, ou ainda fechaduras que recebem ordens remotas.

2.1.1.2 Camada de Rede

A camada de rede na arquitetura conceitual da IoT identifica os dispositivos e controla como as mensagens são transmitidas de um emissor para um receptor. Ela segue dois principais tipos de comunicação, uma direta e outra virtual fim-a-fim entre dois dispositivos IoT. Na comunicação direta, os dispositivos são responsáveis pela transmissão das próprias mensagens e pela transmissão das mensagens de dispositivos terceiros, recebendo-as e encaminhando-as, dispositivo-para-dispositivo, normalmente com o objetivo de transportá-las até um destinatário. A comunicação virtual fim-a-fim abstrai os saltos da comunicação direta e define um padrão de troca de mensagens que garante a confiabilidade em conexões de longa distância. Deste modo, a camada de rede agrega funcionalidades e serviços presentes nas camadas física/enlace, rede e transporte da arquitetura TCP/IP. Para cada tipo de comunicação mencionado, existem diferentes protocolos padronizados, sendo atualmente os principais o 6LoWPAN (*IPv6 over Low power Wireless Personal Area Networks*) e o Bluetooth LE (*Low Energy*). Ambos os protocolos são padronizados pela IETF e definem que o *gateway* deve atuar como uma entidade centralizadora que fornece a interface entre os protocolos da Internet tradicional e da IoT, promovendo heterogeneidade. Além disso, a camada de rede gerencia e controla dados massivos em tempo real para reorganizá-los, filtrá-los e integrá-los de modo que eles sejam transformados em algum serviço (Cervantes et al., 2014). Portanto, essa camada transmite as informações coletadas na camada de percepção e entrega os serviços de rede entre longas distâncias.

2.1.1.3 Camada de Aplicação

A camada de aplicação tem como objetivo integrar todas as funções das camadas inferiores e fornecer serviços específicos e inteligentes para os usuários finais da IoT (Cervantes et al., 2014). Estes serviços podem ser oferecidos através de um servidor de aplicações local ou pela Internet. No servidor analisam-se e processam-se os dados para a tomada de decisões e para desempenhar atividades que cumpram os objetivos dos serviços oferecidos aos usuários. Deste modo, a camada de aplicação coordena e controla as camadas inferiores a fim de prover serviços como monitoramento de ambientes, cuidados com a saúde, casas inteligentes, entre

outros. Além disso, a camada de aplicação pode oferecer serviços de gerência de redes como segurança, monitoramento, controle de acesso, entre outros. Portanto, a camada compreende as aplicações e os serviços baseados em computação em nuvem, descritos na Subseção 2.1.3.

2.1.2 Os Principais Protocolos para a IoT

As principais características dos dispositivos IoT consistem na mobilidade, escassez de recursos energéticos e redes assumindo perdas de pacotes. Devido a isso, implementar os protocolos padronizados para a Internet, que foram projetados para dispositivos mais robustos e conexões confiáveis, se torna inviável na maioria dos cenários IoT. Em vista disso, a IETF, normatizou uma série de protocolos com o objetivo de interconectar estas redes através da Internet e de atender tais dispositivos com recursos limitados. Estes novos protocolos possibilitam a gerência dos dispositivos a distância e assim, motivam o surgimento de novas aplicações e serviços. Com base nisso, esta subseção apresenta estes novos protocolos seguindo a organização do modelo de cinco camadas, tendo como referência a pilha de protocolos TCP/IP. A Tabela 2.1 mostra os principais protocolos padronizados para a IoT. Todavia, além destes protocolos, existem outros que implementam técnicas alternativas, os quais são apresentados em paralelo.

CAMADA	**PROTOCOLO**
Aplicação	CoAP
Transporte	UDP
Rede	6LoWPAN
Enlace	IEEE 802.15.4 / MAC
Física	IEEE 802.15.4 / PHY

Tabela 2.1: Protocolos Padronizados para IoT

Os principais protocolos da camada de aplicação compreendem o Protocolo de Enfileiramento de Mensagens de Telemetria de Transporte (do inglês, *Message Queuing Telemetry Transport* - MQTT) (Banks e Gupta, 2014) e o Protocolo de Aplicações Restritas (do inglês, *Constrained Application Protocol* - CoAP) (Shelby et al., 2014). O protocolo MQTT utiliza o modelo *Publish/Subscribe*, onde os dispositivos geradores de dados (comumente sensores – *Publishers*) os enviam para um *gateway*, que por sua vez os encaminha para os dispositivos interessados (*subscribers*). A ISO/IEC 20922 padronizou o MQTT e definiu regras destinadas às camadas inferiores para o controle da qualidade do serviço. O protocolo CoAP, padronizado pela IETF (RFC 7252), destina-se as aplicações *web* para dispositivos com baixa capacidade computacional e energética. Sua principal característica consiste em utilizar o modelo de Transferência Representacional de Estado (do inglês, *Representational State Transfer* - REST), permitindo que sistemas solicitantes acessem e manipulem representações textuais dos recursos através de comandos básicos como GET, PUT, POST e DELETE. Entretanto, a principal diferença entre estes protocolos consiste em como as conexões são estabelecidas. No CoAP, o *gateway* atua como um intermediário que encaminha mensagens entre um cliente e um servidor, enquanto no MQTT o *gateway* agrega os dados coletados e os clientes apenas têm acesso ao que é disponibilizado nele. Entretanto, devido à arquitetura estabelecida pelo modelo REST, o CoAP também pode atuar como no modelo *Publish/Subscribe*.

A camada de transporte abstrai a estrutura física da rede e determina como dois dispositivos vão se comunicar à distância, ou seja, ela estabelece um padrão para a forma e envio das mensagens em uma comunicação virtual fim-a-fim. Existem dois modelos de comunicação

virtual fim-a-fim, orientada à conexão e não-orientada à conexão. Quando a aplicação exige maior confiabilidade, emprega-se o Protocolo de Controle de Transmissão (do inglês, *Transmition Control Protocol* - (TCP)) da Internet, padronizado pela IETF (RFC 793) (Postel, 1981), para oferecer uma comunicação virtual fim-a-fim orientada à conexão. O TCP garante a entrega dos dados entre dois dispositivos. Em contrapartida, o Protocolo de Datagrama do Usuário (do inglês, *User Datagram Protocol* - (UDP)), padronizado pela RFC 768 (Postel, 1980), tem como principal característica oferecer uma comunicação virtual fim-a-fim não orientada à conexão. Em geral, utiliza-se o UDP quando a aplicação não possui restrições em relação à confiabilidade da entrega. No contexto da IoT, emprega-se o UDP quando a rede é formada por dispositivos com alta mobilidade e baixa capacidade computacional e energética, pois esses dispositivos precisam fazer um uso eficiente dos seus recursos. Desta forma, ao utilizar o protocolo UDP, os dispositivos não precisam manter conexões e podem economizar energia por meio do estado de hibernação sem prejudicar as aplicações e seus serviços (Sonar e Upadhyay, 2014).

Devido ao problema clássico sobre o limite de endereços do protocolo da Internet legada, da versão 4 do protocolo IP, na camada de rede adota-se uma solução que implementa o IPv6 para redes sem fio e com baixo consumo de energia, o protocolo 6LoWPAN. O 6LoWPAN, padronizado pela IETF (RFC 4919) (Montenegro et al., 2007b), permite que a estrutura de rede de curto alcance e baixa disponibilidade de banda se comunique com dispositivos na Internet através do protocolo IPv6. A principal técnica utilizada é a definição e compressão dos cabeçalhos IPv6. Isso permite que dentro dos limites físicos de comunicação, mais dados sejam inseridos no conteúdo (*payload*) dos pacotes sem exceder a Unidade Máxima de Transmissão (do inglês, *Maximum Transmission Unit* - MTU) do protocolo da camada física e enlace na IoT. Além disso, o 6LoWPAN define que um ou mais dispositivos sejam responsáveis pela descoberta e roteamento das mensagens, no geral, estes dispositivos compreendem o *gateway* (descrito na Subseção 2.1.1.2). Desta forma, são necessárias comunicações de múltiplos saltos, exigindo uma maior quantidade de trocas de mensagens e por consequência, um maior consumo de energia.

Em virtude do consumo de energia ocasionado pelo 6LoWPAN, foi padronizado o Protocolo de Roteamento para Redes IPv6 com Baixa Capacidade Energética e com Perdas (do inglês, *IPv6 Routing Protocol for Low-Power and Lossy Networks* - RPL) (RFC 6550) (Alexander et al., 2012). O RPL constrói uma rede com topologia em árvore como um Grafo Acíclico Direcionado Orientado ao Destino (do inglês, *Destination-Oriented Directed Acyclic Graph* - DODAG). Assim, o RPL classifica os dispositivos em nós raízes e nós folhas. Um DODAG está ligado a um ou mais nós raízes e servem como um ponto de trânsito que vincula a rede IoT às redes IPv6. Enquanto, os nós folhas são os dispositivos finais. O RPL também define dois tipos de rotas, descendentes e ascendentes. As descendentes são as rotas direcionadas para os nós raízes e as descendentes são as direcionadas para qualquer nó folha. Para traçar as rotas e montar o grafo, o RPL implementa um padrão de trocas de mensagens que realizam a manutenção das rotas e a inclusão de novos dispositivos na estrutura (Zhao et al., 2017).

Os protocolos IEEE 802.15.4 MAC e PHY (802.15.4, 2016) são os responsáveis pelas camadas de enlace e física, respectivamente. Todos os protocolos das camadas anteriores são adaptados e implementados respeitando os limites estabelecidos por estes dois protocolos. Eles normatizam as redes de área pessoal, ou seja, até 10 metros de baixo custo energético e fácil instalação. O protocolo IEEE 802.15.4 MAC determina as topologias de baixo nível, as classes de dispositivos e o controle de acesso ao meio. A fim de determinar uma topologia inicial que auxilia a descoberta e o reconhecimento entre os dispositivos, o protocolo IEEE 802.15.4 MAC define três tipos de topologia, a topologia em estrela, a topologia em *peer-to-peer* e a topologia mista. Nestas topologias, dividem-se os dispositivos em dispositivo de função completa (DFC) e dispositivo de função reduzida (DFR). Os DFC são os dispositivos mais robustos que implementam todas

as camadas de protocolos e podem desempenhar o papel de coordenador da estrutura, logo são compatíveis com os três tipos de topologias. Enquanto, os DFRs consistem dos dispositivos mais simples, por isso, implementam um conjunto reduzido dos protocolos e compreendem topologias em estrela, ou ainda atuam como um dispositivo final da topologia *peer-to-peer*. Para o controle de acesso ao meio, o IEEE 802.15.4 MAC implementa o acesso múltiplo com verificação de portadora com anulação/prevenção de colisão (do inglês, *Carrier Sense Multiple Access with Collision Avoidance* - CSMA/CA), ele coordena as transmissões conforme a disponibilidade do meio físico. O IEEE 802.15.4 PHY realiza o controle de transmissão e recepção de bits sobre o meio físico conforme a disponibilidade apontada pela camada MAC. Assim, ele determina a frequência de operação, a taxa de transmissão, os canais de rádio e a coexistência entre outros canais de transmissão.

2.1.3 Computação em Nuvem e IoT

Mesmo com alguns dispositivos assumindo a posição de coordenador da topologia da rede, nem sempre eles possuem a capacidade de realizar tarefas mais complexas. Incluir dispositivos mais robustos pode sair muito custoso, tanto financeiramente como também em termos de tempo com instalação e manutenção. Isso trouxe a necessidade de implementar arquiteturas que ofereçam recursos extras e sob demanda para a IoT. Esses recursos incluem poder de processamento, armazenamento, serviços de rede, aplicações completas e até ganho no consumo de bateria, visto que a IoT terceirizaria os recursos e as atividades de processamento. O modelo de computação em nuvem proporciona o fornecimento de recursos computacionais necessários para a IoT através da Internet. Além de tratar Big Data e a heterogeneidade de dispositivos e tecnologias (Botta et al., 2016), os serviços em nuvem são adquiridos através de contratos de nível de serviço (do inglês, *Service Level Agreement* - SLA) que quantificam os recursos e especificam as regras de uso e valores entre o fornecedor do serviço e o cliente. A principal vantagem de utilizar a computação em nuvem consiste na possibilidade de integrar recursos computacionais e serviços que na maior parte das estruturas IoT são escassos.

No entanto, devido ao constante aumento no número de dispositivos, a demanda de serviços de rede também cresceu e acabou gerando problemas de escalabilidade e latência para os serviços ofertados em nuvem. Desta forma, o paradigma de computação em névoa (*fog*) surgiu para aliviar a sobrecarga dos servidores e enlaces responsáveis por interligar a borda da Internet com a nuvem. A computação em *fog* aproxima da borda parte dos serviços oferecidos originalmente em nuvem (Alrawais et al., 2017), permitindo o pré-processamento de dados ou a pré-seleção dos recursos da nuvem. Além disso, a computação em *fog* alivia a sobrecarga e permite um melhor desempenho em aplicações em tempo real.

O amadurecimento e o ganho de desempenho dos serviços oferecidos através da Internet motivaram o surgimento do conceito segurança como um serviço (do inglês, *security as a service*). Este conceito visa a contratação de serviços que forneçam segurança através da Internet. Portanto, os serviços estabelecem uma conexão com os dispositivos dos seus usuários, realizam monitorias e atuam avisando-os sobre os possíveis problemas de segurança, ou tomando medidas de proteção forma automática.

2.2 PRIVACIDADE DIANTE DOS ATAQUES *SIDE-CHANNEL*

A privacidade é o direito do indivíduo de excluir do conhecimento de terceiros aquilo que só é pertinente a ele e que diz respeito a seu modo de ser exclusivo no âmbito de sua vida

privada (Ferraz Júnior, 1993). Nos conformes da era tecnológica, a garantia da privacidade não é dada somente por leis e normas de uso, mas também exige soluções técnicas eficientes para que o usuário tenha opções na forma de tratamento dos dados gerados por ele. As principais técnicas de proteção à privacidade compreendem ocultar os dados que são convenientes a quebra dela. Por exemplo, a criptografia simétrica cifra os dados com base em uma chave secreta, transformando-os em uma sequência de caracteres ininterpretáveis para quem lê em primeira mão. Ainda assim, existem formas de quebrar a privacidade dos usuários, mesmo com os dados criptografados, ou sem ter acesso aos dados na íntegra, como por exemplo nos ataques *side-channel*. Este ataque utiliza dados que vazam inevitavelmente por um dispositivo ao realizar as suas atividades básicas de processamento e transmissão, possibilitando que um usuário mal-intencionado consiga, através de análises estatísticas, inferir informações sobre seus usuários. Esta seção apresenta uma classificação sobre estes dados vazados, também descreve como eles podem ser adquiridos e os detalhes de como ataques *side-channel* podem quebrar a privacidade dos usuários da IoT.

Um dado passa por etapas como a coleta, o processamento, empacotamento, entre outros para ser gerado e transmitido em uma rede de dispositivos computacionais. Estas etapas produzem dados em uma escala diferente, o qual são chamados vazamentos *side-channel*. Eles são o tempo ou a energia que um dado gasta para passar por essas etapas. Nos ataques *side-channel*, os vazamentos são capturados através de técnicas específicas e submetidos a análises estatísticas, podendo revelar características sobre as aplicações, serviços, dispositivos, sistemas operacionais e, consequentemente, informações sobre seus usuários (quebra da privacidade). Estes vazamentos podem ser capturados em diferentes camadas. A Figura 2.2 apresenta uma classificação sobre os dados, gerados por um usuário e seus dispositivos que podem revelar informações privadas dos usuários, quebrando a confidencialidade da rede.

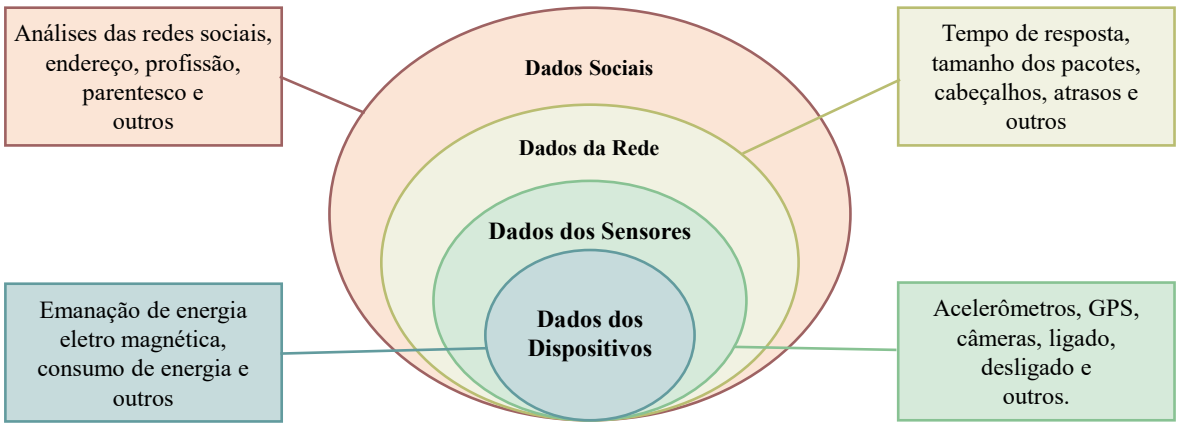


Figura 2.2: Classificação dos Vazamentos *side-channel*

Os vazamentos *side-channel* ocorrem em diferentes camadas. Nas capturas realizadas diretamente sobre os dispositivos, as análises são feitas sobre vazamentos como o consumo de energia, onde o principal objetivo destes ataques normalmente é revelar a chave criptográfica. Nos sensores, as capturas podem ser realizadas através da observação e/ou análises de padrão de imagens e ativação/desativação, com o objetivo de descobrir senhas, padrões comportamentais e posicionamento. As análises realizadas sobre as capturas do tráfego de rede revelam informações como o sistema operacional, aplicações em execução e até os cômodos de uma casa. Através dos

dados sociais podem ser observadas determinadas informações populacionais a fim de descobrir os perfis sociais e padrões de massa publica. No entanto, nas duas primeiras classificações, mostradas na Figura 2.2 (dados dos dispositivos e dados dos sensores), a posse dos dispositivos sujeitos a serem atacados é necessária, já os dados sociais e da rede podem ser capturados remotamente, sendo mais vulneráveis a ataques, facilitando a quebra de privacidade dos usuários através dos ataques *side-channel*.

2.2.1 Definição dos Ataques

Os ataques *side-channel* realizam análises sobre dados vazados não intencionalmente com o objetivo de revelar informações pessoais quebrando ou levando a comprometer a privacidade dos usuários. Estes vazamentos representam o comportamento dos alvos em funcionamento normal ou em situações criadas pelos atacantes. Este ataque pode focar em diferentes alvos, como os dispositivos, os protocolos, as aplicações ou os serviços. Eles são classificados ao longo de dois eixos ortogonais: A) **invasivos** ou **não-invasivos**; B) **ativos** ou **passivos** (M.R., 2010). No eixo A, os ataques invasivos exigem o acesso físico e direto ao dispositivo, para observar vazamentos como o consumo de energia e a ativação ou desativação dos sensores e outros. Um ataque não-invasivo explora os dados disponíveis externamente, neste caso as observações podem ser realizadas através de capturas de rede, possibilitando que vazamentos temporais, de tamanho de pacotes, entre outros sejam explorados. O eixo B define que nos ataques ativos o atacante tem controle sobre as entradas e acesso às saídas, nos passivos ele só observa as saídas.

Neste trabalho, consideram-se os ataques **não-invasivos** que exploram os dados da rede, com um atacante que pode ser tanto **ativo**, quanto **passivo**. Este ataque foi escolhido pois o atacante consegue adquirir informações à distância, eliminando a barreira do acesso físico aos dispositivos. Os ataques **não-invasivos** normalmente exploram dados referentes às características comportamentais dos dispositivos e algoritmos distribuídos. Uma das formas de adquirir os dados para a realização de um ataque **não-invasivo** é através do tráfego de rede. Neste tipo de ataque três etapas principais são características. Na primeira etapa, os vazamentos encontrados em uma captura de rede, como os instantes de transmissão e os tamanhos dos pacotes são extraídos e dispostos em uma lista. A segunda etapa a análise dos vazamentos na busca por padrões que revelem informações sobre a rede ou sobre o comportamento dos dispositivos, protocolos ou aplicações. Na terceira etapa, os atacantes empregam ferramentas de classificação para que possam revelar um conjunto de informações passíveis de quebrar ou comprometer a privacidade dos usuários. Esta variedade é denominada ataque *traffic side-channel* e são considerados mais perigosos devido à fácil implementação, pois não exigem acesso físico aos dispositivos e os vazamentos de rede muitas vezes são inevitáveis (Yan et al., 2017). Um ataque *traffic side-channel* ativo compreende o atacante com controle sobre o envio e recebimento das mensagens trocadas em rede logo, ele pode confeccionar as próprias mensagens para observar como os dispositivos se comportam em diferentes situações. Os ataques passivos podem ser implementados com uma simples captura de rede, independente da camada.

Todos os dispositivos IoT existentes executam os programas em *loop*, portanto, os fragmentos de código seguem ramificações que se repetem durante o seu funcionamento. Cada fragmento de código consome recursos computacionais, energia e até levam tempos diferentes para serem executados. Esse comportamento pode ser capturado, analisado estatisticamente e a distribuição resultante armazenada. Estas distribuições representam o comportamento do dispositivo e podem ser consideradas como impressões digitais capazes de identificar os dispositivos Xiong et al. (2018). Estas impressões digitais são utilizadas para diferenciar os dispositivos e servem de base de conhecimento prévio para ser utilizada como referência. Aproveitando-se disso, um atacante *side-channel* traça impressões digitais a partir dos dispositivos

que estão no mercado, adquirindo um pré-conhecimento dos programas em potencial, agilizando a identificação simplificando a execução do ataque. Para identificar um programa/dispositivo desconhecido em execução, o atacante coleta uma nova impressão digital e faz a correspondência com as adquiridas previamente.

Identificar os padrões comportamentais dos algoritmos e dispositivos é a essência dos ataques *traffic side-channel*. Então, a partir das capturas de rede, tecnologias aprendizagem de máquina são aplicadas com o objetivo de encontrar padrões que identifiquem informações sobre os alvos. Assim, os ataques *side-channel* utilizam como principal ferramenta algoritmos de classificação. Os algoritmos de aprendizagem de máquina podem ser supervisionados ou não-supervisionados. Os supervisionados compreendem etapas de treinamento e teste exigindo que as amostras estejam rotuladas. No treinamento normalmente um classificador procura aprender os padrões nas amostras, calibrando as variáveis pertinentes à filtragem dos campos de entrada conforme os rótulos estabelecidos. A etapa de teste compreende a avaliação da aprendizagem, onde o classificador treinado recebe como entrada as amostras, em seguida, verifica se o resultado da classificação está correto conforme os rótulos. Nos classificadores não supervisionados, as amostras não são rotuladas. Diante disso, os classificadores não supervisionados procuram padrões entre os dados amostrados sem nenhum tipo de rótulo.

2.2.1.1 Vazamentos temporais side-channel

Dentre os diversos vazamentos *side-channel*, uma análise de tempo revela o comportamento dos dispositivos em rede, levando à violação da privacidade dos dados em diferentes escalas. Devido ao comportamento do dispositivo tratar de uma informação extremamente valiosa, os atacantes buscam encontrar padrões para inferir informações sobre cada dispositivo presente na rede. Estes padrões são explorados através do tempo de execução de todo o conjunto de programas dos dispositivos. Com base na literatura, as análises sobre o vazamento temporal *side-channel* de fato revelam informações tanto das camadas inferiores no nível de *hardware* (por exemplo, sobre os sensores embarcados), como sobre as camadas superiores no nível de *software* (por exemplo, sobre os aplicativos em execução) (Yan et al., 2017).

Os programas de computador contêm ramificações e laços de repetições condicionais para manipular entradas e produzir a saída pretendida. Dependendo dos valores de entrada, o caminho de execução de um programa pode diferir (Sayakkara et al., 2019). Em um cenário composto por dispositivos IoT, estes padrões se destacam, pois a escassez de recursos exige que os programas desenvolvidos sejam compostos por blocos de código menores e que envolvam operações mais específicas. Além disso, o tempo de execução desses códigos podem ser diferentes de um dispositivo para o outro devido à divergência de capacidade computacional, como poder de processamento, memória, bateria, entre outros. Alguns parâmetros estruturais dos cenários de redes também são explorados, como a portabilidade dos dispositivos que motivam a geração de novas características no decorrer do tempo.

As especificações definidas pelo protocolo 6LoWPAN exigem que os dispositivos estejam conectados a uma estação base (*gateway*). Os atacantes exploram esta configuração, pois o *gateway* possibilita que conexões externas à estrutura de rede sejam realizadas através de um ponto central. Assim, a captura do tráfego deste dispositivo, mesmo que criptografado, gera brechas através da exploração do tempo de resposta. Explorar este tempo revela informações sobre o posicionamento e o comportamento dos dispositivos. Em contrapartida, esta captura auxilia a gerência de monitoramento das redes. Um exemplo prático é a necessidade de identificar sensores de monitoramento de saúde para oferecer prioridade de transmissão na rede. A próxima subseção apresenta como os ataques *traffic side-channel* temporais ocorrem nas redes IoT.

2.2.1.2 Ataques traffic side-channel temporais contra a IoT

Os ataques *traffic side-channel* seguem três etapas: *(i)* amostragem, *(ii)* extração de características e *(iii)* identificação. Na fase *(i)* que realiza a amostragem os instantes das trocas de mensagens e o tempo de resposta são dispostos em um conjunto de dados. Em seguida, na fase *(ii)* extraem-se as medidas estatísticas convenientes com o objetivo do ataque, como média, mediana, entre outros. A entrada para a computação dessas medidas estatísticas compreende os dados extraídos na amostragem, ou seja, os instantes das trocas de mensagens e o tempo de resposta, elas servem para refinar os dados e melhorar as observações realizadas. Na fase *(iii)* as amostras são submetidas a algoritmos classificadores supervisionados e não supervisionados. A escolha no tipo de classificação depende do objetivo do atacante e das informações já observadas por ele. Desta forma, as informações adquiridas a partir dos vazamentos temporais são relacionadas a outros conjuntos ou bases de conhecimento em geral. Os ataques *traffic side-channel* temporais revelam informações sobre as aplicações, sistemas operacionais, dispositivos e outros, quebrando ou comprometendo a privacidade dos usuários.

2.2.2 Métodos de Defesa

As metodologias de defesa para os ataques *traffic side-channel* temporais compreendem técnicas para mascarar o comportamento dos dispositivos ou da rede. Conforme anteriormente exposto, as variáveis relacionadas ao tempo são analisadas com o objetivo de encontrar padrões nas distribuições dos dados. Assim, os autores utilizam técnicas que destacam estas distribuições, assim, os algoritmos de classificação as detectam e as agrupam de tal forma que representem alguma informação. Os mecanismos de defesa inserem ruídos aleatórios ou controlados para prejudicar a eficácia dos classificadores. Estes ruídos podem ser inseridos seguindo dois métodos: 1) diretamente nos dispositivos, modificando o atraso de processamento, ou 2) na rede, através de técnicas que geram um tráfego falso (do inglês, dummy). No entanto, estes métodos geram um custo de recursos extras como um maior consumo de energia e de banda. Neste trabalho consideramos a sinergia dos dois métodos aplicados tanto de forma aleatória como controlados para implementar uma forma robusta de dificultar a classificação das informações.

Os métodos que visam implementar as técnicas controladas seguem os parâmetros previstos pelos modelos de gerências de segurança e configuração. A gerência de segurança visa identificar os dados sensíveis através do monitoramento da rede e aplicar métodos de proteção. A gerência de configuração, compreende a instalação e controle dos estados da rede, alterados ou não. Desta forma, para identificar a discrepância entre os vazamentos *side-channel*, os mecanismos de defesa compreendendo a gerência de segurança, realizam testes de vulnerabilidades e calculam a discrepância entre as distribuições dos dados. A partir da identificação positiva do vazamento, calcula-se um valor que define um novo comportamento para a rede, sejam eles inserindo um novo atraso ou gerando tráfego falso. Então, através da gerência de configuração, o novo comportamento da rede é aplicado e controlado se necessário.

A IoT representa um novo paradigma de redes, o qual a simplicidade e a escassez de recursos fragilizam a segurança e dificultam a atuação dos mecanismos de defesa. Os ataques *traffic side-channel* realizados contra a IoT são adaptados para se aproveitar da simplicidade dos códigos desenvolvidos para este paradigma, da mobilidade e, principalmente, da especificidade das funções dos dispositivos. Este fato simplifica a realização e potencializa os efeitos dos ataques, pois um dispositivo é mais fácil identificar a função de um dispositivo que não possui múltiplas funcionalidades. Além do mais, os mecanismos que implementam técnicas de defesa para a Internet tradicional compreendem dispositivos que normalmente possuem maiores capacidades computacionais e energéticas, o que inviabiliza a atuação na IoT. Devido a este fato, novos

modelos de mecanismos de defesa precisam ser desenvolvidos compreendendo dispositivos com limitações de memória, processamento, energia e também os novos protocolos padronizados para este tipo de rede. Desta forma, este trabalho propõe um mecanismo de defesa que impede que os ataques *traffic side-channel* temporais sejam eficientes no contexto da IoT, considerando as limitações dos recursos e diferentes dispositivos.

2.3 RESUMO

Este capítulo apresentou os conceitos relacionados ao funcionamento da tecnologia da Internet das coisas, assim como também sua arquitetura, principais protocolos padronizados e a comunicação empregada pelos dispositivos com a computação em nuvem. O capítulo também apresentou como a confidencialidade da IoT pode ser prejudicada para quebrar a privacidade dos usuários através dos ataques *side-channel*. Além disso, foram abordados como a IoT vaza dados, como estes vazamentos são classificados para revelar as informações trafegadas na rede. O capítulo apresentou também os detalhes de como ocorrem os ataques *traffic side-channel* temporais em etapas. Por fim, as técnicas de defesa foram fundamentadas e como elas se relacionam com as gerências de segurança e configuração.

3 TRABALHOS RELACIONADOS

Na literatura, diferentes trabalhos descrevem os ataques *traffic side-channel* e propõem mecanismos de defesa. Neste tipo de ataque, os vazamentos em capturas de tráfego de rede são identificados e analisados, permitindo a aquisição de dados de forma remota, mesmo que os conteúdos transportados pelos pacotes da rede estejam criptografados. Os ataques *traffic side-channel* exploram os vazamentos temporais e podem quebrar a confidencialidade da rede, revelando informações críticas sobre o comportamento dos dispositivos. Na IoT, os dispositivos estão diretamente relacionados à rotina dos usuários, coletando, enviando e recebendo dados, o que torna este tipo de ataque ainda mais grave, podendo levar ao comprometimento da privacidade dos usuários. Este capítulo tem o objetivo de apresentar os principais tipos de ataques *traffic side-channel* e defesas relacionadas com os que exploram os vazamentos temporais. Assim, este capítulo está organizado como segue. A Seção 3.1 apresenta as diferentes implementações dos ataques *traffic side-channel*, destacando os que exploram os vazamentos temporais. Na Seção 3.2, é apresentado o estado da arte das principais técnicas de defesas para os ataques *traffic side-channel* temporais.

3.1 ATAQUES TRAFFIC *SIDE-CHANNEL*

Existem estudos que implementaram os ataques *traffic side-channel* para inferir informações sobre os usuários, dispositivos, serviços ou aplicações que compõem as estruturas de rede em geral (Saltaformaggio et al., 2016; Gu et al., 2019; Feghhi e Leith, 2016; Srinivasan et al., 2008). Eles têm o objetivo de mensurar seus efeitos na privacidade dos usuários. No entanto, existem poucos trabalhos que os exploram no contexto da IoT (Xiong et al., 2018; Prates et al., 2019a). Os efeitos destes ataques se agravam na IoT, pois ela é composta por dispositivos simples, com funções específicas, que coletam e transmite dados críticos sobre o cotidiano dos usuários. Por exemplo, as lâmpadas de uma casa inteligente podem ser identificadas pelos ataques *traffic side-channel* e, com isso, revelar informações sobre a quantidade de cômodos, quais estão ocupados e até os trajetos percorridos pelos usuários (Lin e Bergmann, 2016). Em contrapartida, no contexto da Internet tradicional os dispositivos, no geral, são utilizados para navegação, representando apenas o comportamento virtual do usuário. Com isso, as principais diferenças entre os ataques *traffic side-channel* nestes dois contextos consistem no tipo de informação almejada pelo atacante, ou seja, na Internet tradicional as técnicas de caracterização e classificação são adaptadas para identificar o comportamento virtual e na IoT o comportamento cotidiano dos usuários (Srinivasan et al., 2008). Portanto, esta seção, em um primeiro momento, apresenta dois exemplos de diferentes ataques *traffic side-channel* na Internet tradicional. Em seguida, apresenta os trabalhos que identificam os vazamentos temporais para adquirir as informações privadas dos usuários nas redes de curto alcance. Por fim, apresenta as críticas e uma comparação entre os trabalhos descritos e o mecanismo proposto.

Saltaformaggio et al. (2016) implementaram um ataque **passivo** que analisa os vazamentos de rede para identificar as atividades dos usuários ao navegar em aplicações móveis. Os autores construíram um conjunto de dados de treino por aplicação. Este conjunto é formado por capturas de tráfego gerada em uma simulação de navegação na Internet. Na etapa de amostragem, as amostras foram compostas pelos seguintes vazamentos: instantes de tempo, tamanhos e quantidades de pacotes. A partir delas, na etapa de caracterização, foram extraídas características estatísticas como os tempos médios de resposta e as taxas do tamanho, envio e recebimento

dos pacotes. Nas análises empregadas pela etapa de identificação, os autores identificaram os comportamentos dos usuários por meio de classificadores estatísticos supervisionados e não-supervisionados. Em um primeiro momento, as amostras foram avaliadas e divididas por similaridade na distribuição dos dados através do algoritmo de classificação *K-means*. Por fim, eles treinaram um algoritmo supervisionado chamado Máquina de Vetores de Suporte (do inglês, support vector machine - SVM) com as novas amostras. Em seguida, as amostras compostas por capturas de um tráfego real passam pelas etapas de amostragem e as atividades dos usuários é identificada pelo SVM. Assim, os autores identificaram 35 aplicativos amplamente populares (como redes sociais) com precisão média de detecção de 78,04%. A identificação dos aplicativos fere o direito de privacidade dos usuários podendo levar a um problema de segurança, visto que alguns são utilizados para objetivos específicos, como o controle medicamentoso de uma patologia específica, transmissão de conteúdo adulto ou monitoramento residencial.

Gu et al. (2019) comprometeram a confidencialidade de transmissões de vídeo Dinâmicas e Adaptativas via HTTP (do inglês, *Dynamic Adaptive Streaming over HTTP* - DASH), padrão utilizado pelas plataformas *Netflix* e *Youtube*. Para isso, os autores projetaram um ataque *traffic side-channel* **passivo** de extração de recursos de vídeo para tráfego de transmissão capturado na camada física da rede. Este método empregou uma técnica que gera impressões digitais das transmissões com base no vazamento *side-channel* encontrado na taxa de bits variável. Assim, os autores propuseram um método de correspondência parcial baseado no algoritmo de comparação *Dynamic Time Warping* (DTW) para calcular as semelhanças entre impressões digitais do vídeo e o fluxo de bits capturados na camada física do tráfego de rede. O método foi avaliado e identificou com 90% de precisão os diferentes conteúdos dos vídeos, comprometendo a confidencialidade da rede e consequentemente ferindo o direito de privacidade dos usuários. Através deste trabalho, nota-se a adaptação das etapas do ataque conforme o vazamento explorado. Neste caso, os vazamentos explorados estão em uma camada de abstração muito baixa sendo eles os bits trafegados na camada física.

Dentre os trabalhos que exploram os vazamentos temporais, Feghhi e Leith (2016) classificaram páginas *web* utilizando somente os vazamentos relacionados ao tempo capturados através de um tráfegos de rede criptografado. Os autores exploraram os vazamentos relacionados aos instantes de tempo em que o canal de envio transmite informações. Este trabalho segue as três principais etapas fundamentadas por este tipo de ataque. Na etapa de amostragem, os dados foram dispostos em uma sequência temporal e rotulados por site. Diante das restrições de dados e das diferenças entre os tamanhos das amostras, na etapa de caracterização, os autores utilizaram as técnicas derivadas do DTW e Medida de Distância-F (do ingês, *F-Distance Measure*). O DTW derivado mensura a distância, insensível aos tipos de distorção causadas por possíveis atrasos rede, entre duas amostras de um mesmo site, de tal modo que seja traçado um caminho de deformação entre elas. A técnica de medida de distância-F mapeia a distância entre dois caminhos de deformação. Desta forma, eles calcularam um valor ótimo que define a distorção entre os intervalos de tempo de um determinado site. A etapa de identificação, compreende os classificadores supervisionados *K-Nearest Neighbours* (KNN) e Naive Bayes para a identificação dos sites. As amostras foram divididas em 90% para o treinamento e 10% para o teste, o qual os resultados mostraram uma taxa de acerto médio de 95.01%. Para o Naive Bayes, respeitando as mesmas configurações de treinamento e teste, os resultados mostraram uma taxa de acerto médio de 53.3% ao identificar os diferentes sites.

Os estudos supracitados analisaram o comportamento das transmissões de rede por meio de diferentes vazamentos *side-channel* como tamanhos dos pacotes, as taxas de bits, os dados dos cabeçalhos dos pacotes e os instantes de tempo. Existem diversas outras formas para este tipo de ataque que compreendem objetivos diferentes. Arp et al. (2015) revelaram conexões Tor,

um redirecionador de tráfego que deveria proporcionar uma comunicação anônima e segura para seus usuários. Também, Zhang et al. (2011) propuseram inferir as atividades dos usuários ao navegar na Internet. No entanto, poucos exploraram os vazamentos adquiridos no contexto da IoT. A Tabela 3.1 demonstra todos os trabalhos que implementam técnicas diversas de amostragem e classificação agrupando-os pelas tecnologias almejadas. O restante desta seção apresenta os ataques que exploraram os vazamentos temporais no contexto da IoT.

Tabela 3.1: Categorização dos Ataques por Alvo

Alvo	**Trabalhos**	
Internet Tradicional	Sung et al. (2018) Chen et al. (2018a) Arp et al. (2015) Bates et al. (2012)	Gu et al. (2019) Feghhi e Leith (2016) Wang et al. (2015)
Aplicativos Móveis	Kausar et al. (2019) Atkinson et al. (2018) Saltaformaggio et al. (2016)	Chaddad et al. (2018) Copos et al. (2016)
Internet da Coisas	Prates et al. (2019a) Srinivasan et al. (2008)	Yan et al. (2017)

Srinivasan et al. (2008) implementaram um ataque **passivo**, que inferiu informações sobre uma casa inteligente. Nele, os autores capturam somente o instante em que os dispositivos IoT transmitem dados via rádio. Os autores conseguiram inferir os cômodos, as atividades dos residentes e os dispositivos que a equipam. Neste trabalho foram analisadas oito casas inteligentes equipadas com doze dispositivos ou mais, sendo quatro contendo um residente e as quatro restantes dois ou mais residentes. As amostras compreendem os instantes de transmissão dos dispositivos. Para cada informação inferida, uma variação do ataque *traffic side-channel* temporal é implementada. O ataque completo é dividido em quatro camadas. No entanto, os protocolos utilizados para o estabelecimento de conexão de rede **não correspondem aos protocolos padronizados para a IoT**. As camadas são detalhadas nos parágrafos a seguir.

Na primeira camada, eles identificaram se as casas estão ocupadas ou se os ocupantes estão dormindo. Devido à simplicidade da informação, não foi realizado nenhum tipo de caracterização, desta forma as amostras representavam apenas os momentos ociosos de transmissão. Assim os autores utilizaram o classificador KNN obtendo acurácia entre 85% e 100%. O objetivo da segunda camada é definir quais dispositivos estão em um mesmo cômodo. Na etapa de extração de características estatísticas os autores calcularam a aproximação dos instantes de tempo através do algoritmo de caminho mais curto Dijkstra. O algoritmo recebeu como entrada uma matriz, composta pelos limites mínimos das diferenças temporais aproximadas de cada par de dispositivos. Nas quais, o classificador não-supervisionado *k-means* processa as amostras não rotuladas, para que os dados sejam agrupados. Como resultado, a quantidade de grupos com maior acurácia representa o número de cômodos. Então, as amostras divididas pelos grupos são rotuladas por cômodo.

A terceira camada classificou os cômodos da casa como o banheiro, cozinha, quarto e sala de estar. Uma vez que os dispositivos foram agrupados, representando diferentes cômodos, Srinivasan et al. (2008) assumiram que (i) casas diferentes têm cômodos semelhantes; (ii) os cômodos semelhantes entre as casas podem ser identificados usando as características específicas de uso de cada um. Assim, os autores definiram sete características estatísticas ao representar o comportamento exclusivo de cada cômodo, como segue:

- **Por cômodo:**
  - O agrupamento como o número de transmissões por dia
  - A mediana entre cada instante de transmissão
  - A distância mediana de tempo entre as transmissões
- **Por casa:**
  - O número total de transmissões durante o dia
  - O número total de transmissões durante a noite
  - O primeiro cômodo a transmitir depois de longos períodos
  - Um histograma das transmissões com granularidade de quatro horas

Desta forma, os autores utilizaram como base de treinamento uma captura rotulada. Considerando que o atacante também pode ter uma casa inteligente de teste composta por dispositivos comprados no mercado. Então, um classificador supervisionado que implementa um algoritmo de *correspondência bipartida de custo mínimo* foi treinado e testado com as amostras compostas pelas características extraídas. Esta camada identificou os cômodos com acurácia entre 95 e 100%.

A quarta camada define a função de cada dispositivo e, com isso, as atividades que estão sendo executadas, naquele intervalo de tempo. Elas são identificadas como cozinhando, tomando banho, dormindo, entre outros. Nesta camada, os autores reutilizaram como base de conhecimento as características extraídas das camadas anteriores para classificá-los supervisionadamente através de uma *análise discriminante linear*. Além disso, assumem que todos os tipos de sensores de uma casa de testes foram observados pelo menos uma vez em uma casa de treinamento. Assim, os dispositivos foram amostrados e rotulados individualmente conforme as camadas anteriores, compreendendo o cenário de oito casas. Nas quais, para identificar as informações almejadas nesta camada, o classificador que foi treinado com as amostras de sete casas é usado para classificar os dispositivos de uma oitava. Esta camada conseguiu identificar os dispositivos com acurácia de até 80%. Srinivasan et al. (2008) demonstraram o grau de periculosidade de um ataque *traffic side-channel* temporal. Onde, apesar da complexidade, um atacante mapeia a estrutura de uma casa inteligente quase que por completo, simplesmente com capturas da camada física de rede, podendo levar a problemas de segurança não só virtual como física.

Dentre os trabalhos que implementam os ataques *traffic side-channel* temporais contra os protocolos padronizados para a IoT. Yan et al. (2017) observaram as características extraídas dos vazamentos referentes aos tamanhos dos pacotes e os tempos de resposta. O cenário de teste era composto por dois dispositivos diferentes IoT. Este trabalho destacou a criticidade dos vazamentos temporais. Isso é explicado, pois as capturas realizadas sobre o tempo apresentam menor quantidade de ruído quando comparada ao tamanho do pacote, e com isso, alcança uma precisão maior na classificação do tráfego. Neste sentido, este trabalho considera implementar um mecanismo de defesa para tipo de ataque apresentado em Prates et al. (2019a), onde os autores mostraram uma análise sobre os vazamentos temporais *side-channel*, a fim de caracterizar e identificar dispositivos idênticos em uma rede IoT. Motivados pela violação de privacidade e consequências causadas por tais ataques, os autores destacaram a relevância dos vazamentos temporais *side-channel* na caracterização do tráfego de dispositivos IoT idênticos executando os mesmos protocolos e aplicações. Para este fim, a análise compreendeu a coleta do tráfego de três dispositivos IoT, a extração dos vazamentos temporais e a identificação dos dispositivos. As informações *side-channel* consideradas englobaram o instante de envio e o tempo de resposta, bem como suas medidas estatísticas como média, mediana, limites máximos e mínimos. Por meio desse conjunto de informações a análise apresentou um comportamento único e específico de cada dispositivo mesmo idênticos, alcançando taxas de precisão de até 100%.

Esta seção descreveu os trabalhos que implementaram as variantes dos ataques *traffic side-channel* e os detalhes daqueles mais eficientes no contexto geral de redes. Diante disso, identificou-se que (i) o classificador supervisionado mais eficiente é o KNN; (ii) o classificador não-supervisionado K-means teve unanimidade entre os trabalhos. As técnicas de caracterização normalmente se adaptam à informação buscada pelo atacante, agrupando ou dispersando as distribuições. Os trabalhos de Saltaformaggio et al. (2016), Gu et al. (2019) e Feghhi e Leith (2016) aproveitam-se dos vazamentos encontrados no tráfego de diferentes camadas das redes tradicionais, não considerando os protocolos e as características dos dispositivos IoT. Em especial, Feghhi e Leith (2016) identificaram com sucesso informações utilizando unicamente os vazamentos temporais. No entanto, as técnicas empregadas por eles, nas etapas de caracterização e identificação, procuram comprometer a confidencialidade de páginas web e não das informações privadas sobre os usuários. Em contrapartida, Srinivasan et al. (2008) implementaram um ataque que desvenda com sucesso as informações sobre uma casa inteligente equipada por dispositivos IoT, apenas com os vazamentos temporais sobre os instantes de atividade dos dispositivos. Entretanto, o cenário de avaliação considerado não corresponde aos protocolos atualmente padronizados para este contexto. Yan et al. (2017) e Prates et al. (2019a) consideraram os protocolos padronizados para a IoT, apesar disso, eles apenas Prates et al. (2019a) apresentaram resultados concretos considerando os vazamentos temporais em um cenário com dispositivos idênticos, deixando em aberto as questões sobre heterogeneidade.

Em busca de uma compreensão dos ataques *traffic side-channel* temporais contra uma rede IoT, esta seção auxilia a criação de um cenário experimental para a avaliação do mecanismo proposto. Esta avaliação tomou também como referência outro trabalho que apresenta uma forma eficiente de caracterização através dos vazamentos temporais. Selis e Marshall (2017) implementaram um mecanismo de identificação de dispositivos virtualizados intrusos que falsificam o tempo de resposta para simular um dispositivo IoT real. Então, ele caracteriza os tempos capturados e aplica cálculos estatísticos sobre os dados vazados como o instante de envio e o tempo de resposta. Estes dados são separados em amostras e submetidos a cálculos como média, mediana, correlação de Pearson, limites inferiores e superiores. Em seguida, as novas amostras servem como entrada para os classificadores KNN e Random Forest. Desta forma, o mecanismo proposto identificou dispositivos falsos com até 100% de precisão. Apesar de considerar um cenário da IoT, Selis e Marshall (2017) não avaliaram os ataques *traffic side-channel*. Assim, o cenário experimental visa implementar um ataque de máxima eficiência capaz de identificar tanto os diferentes dispositivos IoT em rede, quanto os sensores embarcados em cada um. Em vista disso, avaliam-se as diferentes formas de caracterizar e classificar os vazamentos temporais. Além disso, o mecanismo proposto aproveita estas técnicas para um módulo de teste de vulnerabilidade que identifica os vazamentos temporais vulneráveis nas redes IoT.

3.2 DEFESAS A ATAQUES *TRAFFIC SIDE-CHANNEL* TEMPORAIS

A maioria dos trabalhos que propuseram soluções de defesa para os ataques *traffic side-channel* temporais utilizaram métodos para mascarar os vazamentos. A escassez de trabalhos no contexto da IoT, caracterizada na seção anterior, justifica a carência de trabalhos que implementam mecanismos de defesa para este tipo de ataque. Além disso, as diferentes naturezas dos vazamentos *side-channel* exigem técnicas de defesa exclusivas para as variáveis relacionadas a cada um. Esta seção considera com exclusividade os métodos que compreendem mascarar os vazamentos temporais *side-channel*. Desta forma, classificamos os trabalhos pelas técnicas implementadas. A Tabela 3.2 mostra as propostas divididas pelos métodos de inserção de atrasos e/ou que geram tráfego falso. Estas técnicas podem ser aplicadas de forma controlada ou aleatória. A característica predominante entre as técnicas encontradas é a sobrecarga de recursos,

sejam eles da rede, como maior consumo de banda e tempos de resposta; ou dos dispositivos como um maior consumo de energia. Assim, além de descrever as principais técnicas de defesa, os trabalhos são comparados e criticados conforme as avaliações de consumo de recursos, as técnicas empregadas e o tipo de rede considerada.

Tabela 3.2: Categorização das Defesas por Técnica

Trabalhos	**Atraso Aleatório/Controlado**	**Tráfego Falso Aleatório/Controlado**	**IoT**
Prates et al. (2019b)	Ambos	Controlado	✓
Feghhi e Leith (2019)	-	Controlado	-
Wang et al. (2008)	-	Controlado	-
Srinivasan et al. (2008)	Aleatório	Controlado	✓
Shmatikov e Wang (2006)	Controlado	Controlado	-

Feghhi e Leith (2019) apresentaram um mecanismo de defesa eficiente contra um ataque *traffic side-channel* temporal que revela páginas web (Feghhi e Leith, 2016). Os autores consideraram uma implementação que auxilia um mecanismo de tunelamento criptográfico, como uma rede virtual privada (VPN), ao garantir a confidencialidade das conexões através da técnica de geração de tráfego falso. Desta forma, os autores consideraram duas políticas de gerência de configuração, *adaptativa* e *capacitiva*. Na *adaptativa* o número de pacotes falsos transmitidos se adapta com a carga de tráfego. Quando a carga de tráfego é baixa, o mecanismo insere pacotes fictícios conforme necessário para preencher os instantes vagos de trocas de pacotes e, assim, mascarar as características causadas pelos instantes de transmissão. A *capacitiva* reduz a geração de pacotes falsos para zero à medida que o tráfego aumenta, ou seja, os usuários podem fazer uso total da capacidade de transferência do túnel de rede conforme o aumento no fluxo de pacotes. Assim, os autores conseguiram reduzir a acurácia dos ataques em até 82%.

Wang et al. (2008) implementaram um algoritmo de preenchimento de canal, ou seja, os autores visaram padronizar o comportamento de envio e recebimento de pacotes preenchendo os espaços de tempo em que não há transmissões. O algoritmo implementa tanto o cálculo e atraso dos pacotes como a geração de tráfego falso. Assim, um padrão de intervalos silenciosos é identificado e preenchido através da fabricação de um tráfego falso, manipulando as capturas de rede. No entanto, os intervalos entre os tempos de resposta ainda são identificáveis. Seguindo essa afirmação, os autores calculam um valor de tempo que define um limite para os atrasos de rede, onde caso o tempo previsto para o pacote falso seja menor que o limite calculado, um atraso controlado é inserido. Este trabalho avaliou o algoritmo utilizando as distribuições de Pareto e Poisson. No entanto, os resultados não foram efetivos diante de um ataque *traffic side-channel*. Shmatikov e Wang (2006) propuseram um algoritmo que visa destruir as possíveis impressões digitais de forma adaptativa. Com isso, os autores preencheram as lacunas de tempo sem adicionar pacotes onde o tráfego já é denso. O algoritmo segue duas regras de gerência. O primeiro *burst* define que depois de um pacote ter sido recebido, um novo intervalo esperado entre os pacotes é amostrado. O *gap* define um novo intervalo a partir da expiração do atraso amostrado pelo *burst*. Esta técnica se mostrou mais efetiva diminuindo a capacidade de identificação dos vazamentos temporais em até 44.7%.

Srinivasan et al. (2008) apresentou uma avaliação sobre todas as técnicas de defesas fundamentadas, sendo a inclusão de atrasos de forma aleatória e o envio de pacotes falsos de forma controlada. Além disso, avaliou a inclusão de um atenuador de sinal, diminuindo o alcance das transmissões e, também, a capacidade da interceptação do sinal sem fio. A avaliação é

pertinente ao ataque apresentado na Seção 3.1. Desta forma, ao avaliar a inclusão dos atrasos aleatórios, a técnica reduziu a acurácia da primeira camada em 60% e nas demais em média de 40%. Na avaliação que se refere ao envio de pacotes falsos, os autores compreendem que os dispositivos transmitem um pacote falso a cada oito segundos, inviabilizando todas as camadas do ataque proposto. Também foi avaliado o custo energético do envio extra de pacotes, no qual reduziu em 8.75% o ciclo de vida do dispositivo avaliado. Os autores avaliaram as técnicas aplicadas em conjunto, onde a redução da acurácia média de todas as camadas foi de 40%.

Prates et al. (2019b) apresentou um mecanismo de defesa contra os ataques *traffic side-channel* temporais no contexto da IoT. O mecanismo o mecanismo segue dois módulos: teste de vulnerabilidades e proteção de privacidade. O primeiro módulo realiza uma rotina de requisições, coleta o tráfego de rede, extrai o instante de tempo em que foi enviada uma requisição e o tempo de resposta por requisição. Em seguida, divide em amostras para a caracterização e identificação dos vazamentos temporais *side-channel*. O módulo de proteção de privacidade define regras individuais para os dispositivos a fim de mascarar as diferenças estatísticas de forma controlada. Desta forma, os autores avaliaram três abordagens. A primeira compreende mascarar os instantes de envio realizando a duplicação das mensagens, ou seja, duas requisições são geradas, exigindo duas respostas simultâneas. A segunda inseriu os atrasos de forma aleatória considerando o ciclo da fila de processos implementada pelo sistema operacional Contiki. A terceira abordagem inseriu operações de atraso na função de envio das mensagens aproximando os tempos médios de resposta dos dispositivos. Os resultados mostram que a primeira abordagem foi mais eficiente, onde, reduziu a precisão do algoritmo de classificação em até 63%, a terceira foi marginalmente eficiente reduzindo menos que 10% e a segunda é ineficaz pois os classificadores não apresentaram dificuldade ao identificar os dispositivos.

Todos os trabalhos apresentados nesta seção compreendem defesas contra os ataques *traffic side-channel* temporais. No entanto, apenas Prates et al. (2019b) apresentou um mecanismo apropriado aos protocolos padronizados para a IoT. Apesar disso, apenas Srinivasan et al. (2008) realizou uma análise sobre o consumo de energia, considerando a característica da escassez de recursos energéticos na IoT. Wang et al. (2008), Feghhi e Leith (2019) e Shmatikov e Wang (2006) apresentaram um conceito para geração de pacotes falsos baseados no estado atual da rede a fim de economizar largura de banda. Esta abordagem pode ser útil visto pois gera menor quantidade de mensagens comparado a uma abordagem sem esse tipo de controle. Este trabalho propõe um mecanismo de defesa que considera a sinergia entre as técnicas de geração de pacotes falsos e a inclusão de atrasos. O mecanismo, diferentemente dos trabalhos da literatura, atua de forma sustentável, consciente do consumo de energia e de recursos de rede.

3.3 RESUMO

Este capítulo apresentou uma revisão da literatura referente aos ataques *traffic side-channel* e das técnicas de defesa contra para os vazamentos temporais. Na primeira seção, foram apresentados os ataques conforme as etapas *(i)* amostragem, *(ii)* extração de características e *(iii)* identificação (fundamentadas na Subseção 3.1), onde mostra o estado da arte diante dos diferentes tipos de ataques *traffic side-channel* nas redes tradicionais e na IoT, bem como seus efeitos. O objetivo da seção foi identificar as técnicas empregadas por este tipo de ataque, principalmente os relacionados aos vazamentos temporais, para servir como base de conhecimento para a avaliação do mecanismo de defesa proposto. Em seguida, o estado da arte das técnicas de defesa contra os ataques *traffic side-channel* temporais são classificados e descritos. A partir deste capítulo também é possível observar a carência de trabalhos considerando as defesas para este tipo de ataque nas redes IoT padronizadas.

4 UM MECANISMO DE DEFESA CONTRA ATAQUES TRAFFIC *SIDE-CHANNEL* TEMPORAIS NA INTERNET DAS COISAS

Este capítulo apresenta o mecanismo FISHER (do inglês: A De**f**ence Mecan**i**sm against Time Traffic **S**ide-C**h**annel in Int**er**net of Things) de Defesa Contra ataques *Traffic Side-Channel* Temporais no contexto da IoT. A Seção 4.1 apresenta uma visão geral do mecanismo proposto, bem como suas principais características, o modelo de rede segundo o ataque considerado. A Seção 4.2 ilustra e descreve com detalhes a arquitetura do mecanismo bem como seus módulos e as técnicas empregadas.

4.1 VISÃO GERAL

O mecanismo de defesa contra ataques *traffic side-channel* na IoT atua para identificar e diminuir a eficiência destes ataques através da identificação e do mascaramento dos vazamentos temporais. Este ataque fere o direito de privacidade dos usuários de dispositivos IoT, sem precisar acessar o conteúdo dos pacotes. Os vazamentos temporais são capturados pelos instantes entre as trocas de mensagens e os tempos de resposta. Estes dados são analisados através de algoritmos classificadores e comparados com informações de conhecimento geral, isto revela os objetivos dos dispositivos em rede. Então, para proteger a privacidade de uma rede sem fio e de área pessoal, o mecanismo atua como um serviço virtual executando no *gateway* para identificar e mascarar os vazamentos *side-channel*. O mecanismo atua na rede pelos módulos de teste de vulnerabilidades e proteção de privacidade. O primeiro módulo realiza as operações de coleta do tráfego e amostragem, com o intuito de identificar os vazamentos temporais *side-channel* por dispositivo. O segundo módulo insere atrasos e gera mensagens falsas a fim de mascarar o comportamento observável dos dispositivos. O mecanismo não procura esconder o fato de que os dispositivos estão trocando mensagens em rede, ele procura esconder as características individuais de cada dispositivo. Portanto, a privacidade é incorporada na indistinguibilidade das características individuais dos dispositivos fornecidas em relação às sequências de tempos de resposta. Assim, o mecanismo coordena a sinergia dos módulos através de ciclos de identificação, proteção e teste dos dispositivos vulneráveis. A Figura 4.1 demonstra a rede IoT e onde o mecanismo é executado. A Tabela 4.1 apresenta toda a terminologia usada neste capítulo.

4.1.1 Modelo de Rede

O mecanismo FISHER considera proteger a privacidade uma rede de sensores sem fio, conforme o padrão IEEE 802.15.4 sob os protocolos 6LoWPAN e CoAP. Esta rede de dispositivos IoT se conecta à uma rede de longo alcance através de um *gateway* de maior poder computacional. Os protocolos utilizados definem uma rede de área pessoal (distância nominal de 10 metros) em topologia estrela, conforme a Figura 4.1, denotada por um conjunto de dispositivos $N = \{G, D\}$ e um conjunto de canais $C = \{c_1, c_2, c_3, ..., c_{|D|}\}$. Nesta rede, N é composto pelos subconjuntos G e D, onde, $|G| = 1$, $|D| \geq 1$ e $D = \{d_1, d_2, ..., d_i\}$. G é o *gateway* no centro da topologia estrela, ele atua como o dispositivo coordenador, ou seja, recebe demandas de dados através da rede de longo alcance e coleta informações através de uma rede de sensores sem fio que monitoram o ambiente. O subconjunto D representa todos os dispositivos finais com sensores embarcados, eles recebem as requisições de G e respondem com os dados solicitados. Cada $c \in C$ representa os canais da comunicação direta entre

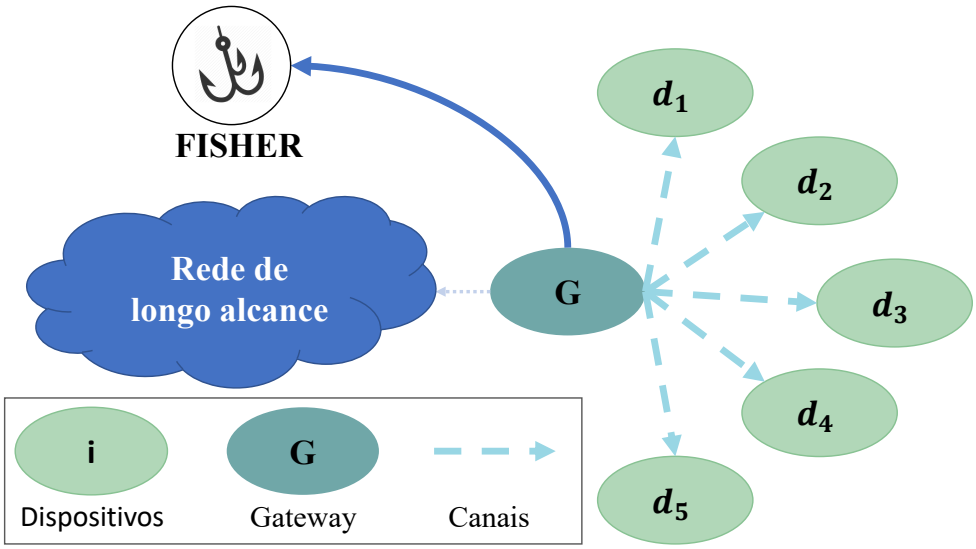


Figura 4.1: Rede IoT

Símbolo	Valor
N	Rede IoT
T	Instante de envio
G	Gateway de rede
d	Um dispositivo
D	Subconjunto de conjunto de dispositivos
C	Subconjunto de canais
c	Um canal
Tr	Transmissões
V	Conjunto de características estatísticas
m	Mensagem
req	Requisição
$resp$	Resposta
$Traf$	Tráfego de rede
Tr	Transmissão
$\mathcal{P}$	Família de distribuições
p_v	Distribuição
P	Probabilidade
A	Conjunto de amostras de tempo de resposta
η	Limiar de requisições capturadas
τ	Tempo de resposta
w	Precisão
X	Variáveis de tempo de resposta
χ	Conjunto de possíveis tempos de resposta

Tabela 4.1: Terminologia

um *gateway* G e um determinado dispositivo d_i. Nestes canais são trocadas mensagens m de requisições $m.req$ e respostas $m.resp$, ou seja, todas as requisições são transmitidas do *gateway* para um dispositivo final. Cada dispositivo d_i ao receber uma mensagem de requisição $m.req_i$ gera uma resposta $m.resp_i$. As trocas de mensagens geram um tráfego de rede por

dispositivo $Traf^{d_i} = \{(req_1, resp_1), (req_2, resp_2), (req_3, resp_3), ..., (req_{|Traf^{d_i}|}, resp_{|Traf^{d_i}|})\}$, visto que todas as req_i e $resp_i$ são mensagens m enviadas para e recebidas de um dispositivo d_i.

4.1.2 Modelo do Ataque

Este trabalho considera os ataques *traffic side-channel* temporais não-invasivos, ou seja, o atacante apenas observa o tráfego de rede visando classificar as distribuições de tempo de resposta p_ν em $Traf^d$ (definido na subseção 4.1.1) com o objetivo de identificar cada dispositivo $d_i \in D$. Desta forma, os atacantes organizam os vazamentos em amostras $V = \{Tr_1, Tr_2, ..., Tr_x\}$ de x transmissões e procuram classificar as diferenças entre as distribuições entre cada p_ν através da identificação de padrões. A distribuição p_ν vem de uma possível família de funções de densidade $\mathcal{P} = \{p_\nu\}_{\nu=1}^{s}$, onde denotamos a probabilidade *a priori* de um dispositivo d_i gerar pacotes conforme uma determinada p_ν como $P_{p_\nu}(V) \triangleq P(a_k \sim p_\nu)$. Portanto, cada $\mathcal{P}$ pode representar os tipos de dispositivo IoT e ν um determinado dispositivo d_i ou, $\mathcal{P}$ um conjunto de estados de operação, e ν o estado de operação atual. Assim, os atacantes podem utilizar classificadores não-supervisionados e supervisionados para aprender p_ν e $P_{p_\nu}(Tr_x)$. Os classificadores não-supervisionados identificam os possíveis conjuntos de $Tr_x \in V$ e a probabilidade de pertencerem a um determinado d_i. Um classificador supervisionado exige que as amostras estejam rotuladas. Assim, os atacantes podem utilizar alguns atributos através de um ν qualquer, previamente conhecido, possivelmente ligado a algum d_i que transmitiu mensagens em V e treinar um classificador supervisionado para identificar a probabilidade $P_{p_\nu}(Tr_x)$ de cada transmissão Tr_x vir de uma distribuição p_ν.

4.2 DETALHAMENTO DO MECANISMO

O mecanismo FISHER melhora a privacidade dos usuários e dos dispositivos evitando que os ataques *traffic side-channel* sejam eficientes. Ele atua sobre a rede como um serviço virtual a fim de mascarar os vazamentos *side-channel*. Desta forma, a arquitetura do mecanismo segue o módulo de teste de vulnerabilidades e o módulo de proteção de privacidade, como mostra a Figura 4.2. O primeiro módulo monitora as mensagens de requisição e resposta, calcula as características estatísticas dos instantes de envio e recebimento para a identificação dos vazamentos *side-channel* entre os dispositivos da rede de área pessoal. O segundo módulo envia mensagens falsas e insere atrasos na rede a fim de mascarar os seus instantes de envio e recebimento com base nas análises realizadas para mascarar os vazamentos temporais *side-channel*. Os módulos trabalham em sinergia e atuam em ciclos de identificação e mascaramento dos vazamentos, ou seja, o módulo de proteção de privacidade depende das análises realizadas pelo módulo de identificação de vulnerabilidades. Desta forma, o mecanismo identifica e máscara os vazamentos *side-channel* controlando o risco de privacidade.

4.2.1 Módulo de Teste de Vulnerabilidade

O módulo de teste de vulnerabilidade coleta o tráfego de rede, extrai as amostras de tempo de resposta, treina e testa um classificador supervisionado para identificar os vazamentos *side-channel* por dispositivo. Para isso, ele segue as funções: $F1 : Traf \rightarrow A$, $F2 : A \rightarrow V$ e a $F3 : V \rightarrow D'$. A Figura 4.3 demonstra, com valores ilustrativos, a ordem cronológica e os cabeçalhos das amostras de saída. A primeira é uma função sobrejetora $F1 : Traf \rightarrow A$ que recebe como entrada o tráfego de rede $Traf$ e para cada conjunto de amostras $Traf^d \in Traf$, coleta o endereço, os instantes das trocas de mensagens (*timestamps*= T) e calcula os tempos de resposta τ. Desta forma, cada amostra em $a \in A^d \in A$ é composta pelos atributos:

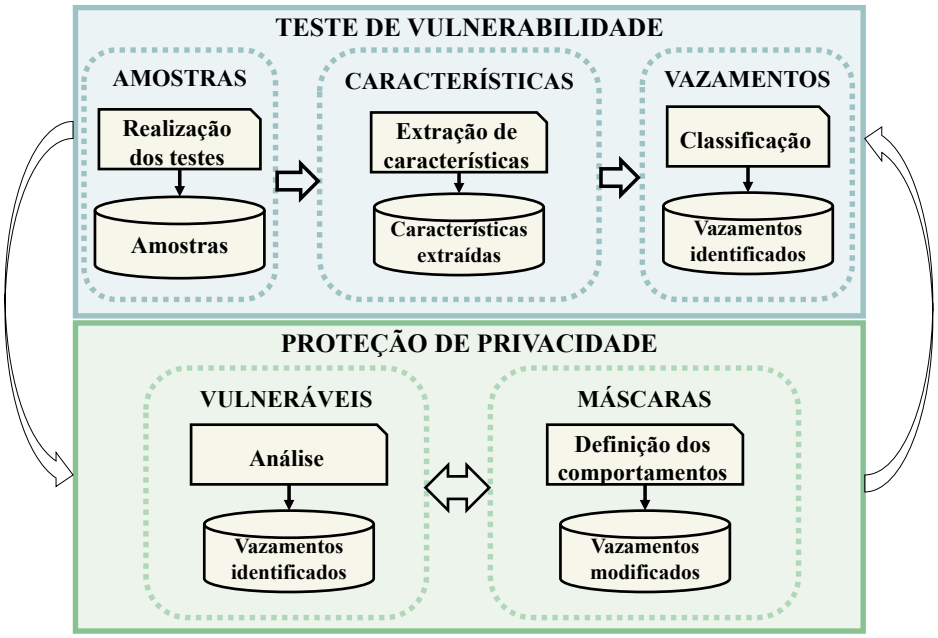


Figura 4.2: Arquitetura do Mecanismo

endereço do dispositivo $a_i.id$, o instante de envio $a_i.T$ e o tempo de resposta $a_i.\tau$. Nesta função, o módulo calcula o tempo de resposta (τ_i^d) dos dispositivos a partir da diferença entre os instantes de envio de uma requisição e de recebimento de uma resposta, ou seja, $\tau_j^d \in A^d = resp_j - req_,(\forall d \in D \ \ \& \ \ \forall j \in Traf^d)$. Esta função ($F1$) retorna um conjunto de amostras $A^d = \{a_1, a_2, a_3, ..., a_{|Tr^d|}\}$ por $d \in D$, $a_1.T < a_2.T < a_3.T$, ordenados em uma sequência discreta de tempo. A segunda função sobrejetora $F2 : A^d \rightarrow V^d$ recebe como entrada as amostras A^d e, respeitando a ordem do conjunto, calcula as medidas estatísticas (apresentadas na Tabela 4.2), elas ajudam na otimização dos algoritmos empregados na próxima função. Esta função $F2$ divide as amostras $a \in A^d$ em subconjuntos A'^d de tamanho t para calcular as medidas estatísticas, então, para cada $A_t'^d$ a função calcula uma amostra de características $v_{(|A^d|/t)}$. Assim, esta função retorna um conjunto de amostras $V^d = \{v_1, v_2, v_3, ..., v_{(|A^d|/t)}\}$ por dispositivo, onde cada amostra é composta pelo endereço do dispositivo e as medidas estatísticas. A última função $F3 : V^d \rightarrow D'$ recebe como entrada as amostras V^d e emprega um classificador supervisionado a fim de identificar o os dispositivos vulneráveis $D' \in D$. Desta forma, cada dispositivo $d \in V$ representa uma classe de entrada para o classificador.

As amostras A^d são independentes e identicamente distribuídas (iid) pela função de distribuição $p_\nu(A^d)$. Esta distribuição representa o comportamento dos dispositivos IoT, ao trocarem mensagens, com base nos tempos de resposta. Desta forma, p_ν vem de uma possível família de funções densidade de probabilidade $\mathcal{P} = \{p_\nu\}_{\nu=1}^{s}$, onde denotamos a probabilidade *a priori* de um dispositivo $d \in D$ gerar pacotes conforme uma determinada p_ν como $P_{p_\nu}(A^d) \triangleq P(a_k \sim p_\nu)$. Os algoritmos classificadores procuram identificar os padrões de cada V^d que representam uma determinada distribuição p_ν. Então, o módulo divide o conjunto V^d em dois conjuntos por dispositivo, treino V_{treino}^d e teste V_{teste}^d.

Os classificadores supervisionados recebem como entrada o conjunto de amostras V_{treino}^d e treinam um modelo para cada classe d da distribuição p_ν. Assim, o modelo identifica a probabilidade de cada v_i pertencer a uma determinada classe de dispositivos d a partir das entradas de teste V_{teste}^d. Além disso, este módulo é flexível quanto a especificidade dos classificadores supervisionados. Estes classificadores são avaliados através da métrica de desempenho precisão.

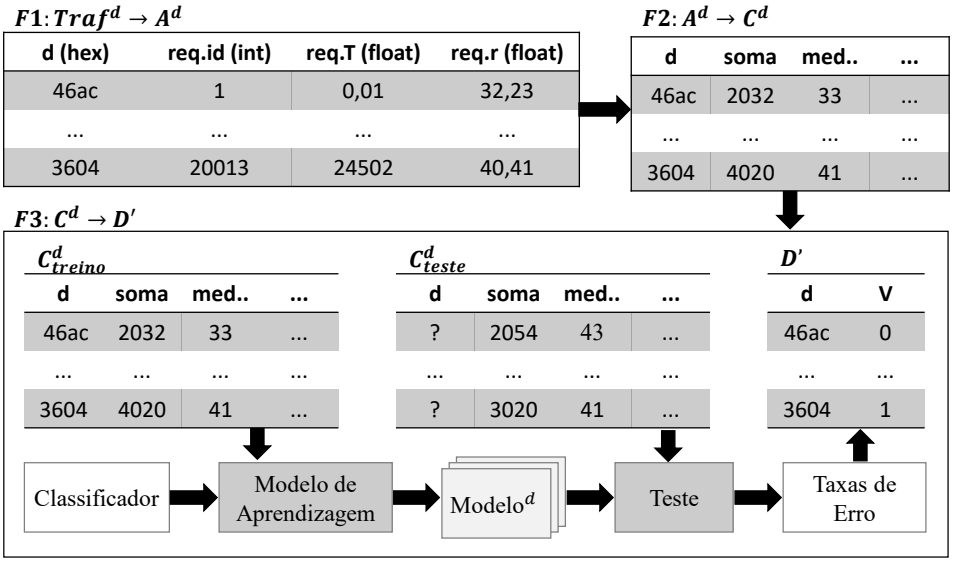


Figura 4.3: Funções do Módulo de Teste de Vulnerabilidade

Medida Estatística	Equação
Mínima (τ)	$min = min(x_i)$
Soma (τ e T)	$sum = \sum_{i=1}^{N} x_i$
Média (τ)	$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$
Limiar Superior (τ)	$LS = \mu + 1.96\sigma$
Moda (τ e T)	$mode = freq(X)$
Mediana (τ e T)	$Md = \frac{1}{2}(X_{\frac{N}{2}} + X_{\frac{N}{2}+1})$ de $sort(X)$
Coef. Correlação de Pearson (τ e T)	$cp = \frac{\sum_{i=1}^{N}(x_i-\mu_x)(y_i-\mu_y)}{\sqrt{\sum_{i=1}^{N}(x_i-\mu_x)^2 \sum_{i=1}^{N}(y_i-\mu_y)^2}}$

X = conjunto de dados; x_i = amostra i dos dados; N = quantidade de amostras; σ = variância; $freq$ = valor mais frequente; $sort$ = valores ordenados.

Tabela 4.2: Medidas Estatísticas para Caracterização do Tráfego (Selis e Marshall, 2017)

Tal métrica considera as taxas de verdadeiro positivo (VP) e verdadeiro negativo (VN). Portanto, o mecanismo considera um dispositivo vulnerável quando um classificador identifica de forma eficiente um dispositivo. A precisão w estima a porcentagem de verdadeiros positivos dentre todos as amostras de tempos de resposta classificadas como positivos ($w = VP/(VP + FP)$) e é a principal métrica para avaliação de um classificador supervisionado. Contudo, o mecanismo se embasa na métrica de precisão, pois representa a taxa de dispositivos corretamente que foram identificados, o que comprova a existência de vazamentos temporais nas distribuições referentes a um dispositivo quando comparado aos outros. Portanto, definimos um limiar l para classificar os dispositivos como vulneráveis ou mascarados. Assim, a função sobrejetora $F3 : V^d \rightarrow D'$, para cada $d \in D$, $D' = \{d'_1, d'_2, ...d_{|D|}\}$. Cada d' é um valor binário por dispositivo; 0 representa um dispositivo seguro, ou seja, não apresenta vazamentos quando comparado com as outras distribuições e 1 um dispositivo vulnerável, as regras são aplicadas conforme a equação a seguir:

$$F3(V^{d_{i+1}}) = \begin{cases} 1 & se \quad \frac{VP^{d_i}}{(VP^{d_j}+FP^{d_j})} \geq l \\ 0 & se\ não \end{cases}, \left(\forall d_j \in D\right) \tag{4.1}$$

4.2.2 Proteção de Privacidade

O módulo de proteção de privacidade recebe os vazamentos identificados a fim de mascará-los para diminuir efetividade de um ataque *traffic side-channel* temporal. O módulo parte do princípio de que o atacante tem um classificador treinado previamente com uma distribuição da família $\mathcal{P}$ o que possibilita o pré-conhecimento da probabilidade P_{pv}. Portanto, um atacante observa uma troca de mensagens, calcula o tempo de resposta x e descobre uma nova probabilidade sobre o comportamento da distribuição $p_v \in \mathcal{P}$. Isto pode revelar qual dispositivo ou por qual objetivo ele está trocando mensagens na rede almejada, para prejudicar a privacidade dos usuários. O módulo manipula as possíveis capturas de tráfego através da modificação do comportamento padrão dos dispositivos e dificulta as análises realizadas pelos ataques. Assim, o módulo identifica as diferenças médias entre os tempos de resposta e insere atrasos ou gera pacotes falsos. Então, o módulo recebe os conjuntos A^d, V^d e D' e identifica os dispositivos que apresentaram vazamentos em D'. Por fim, para cada nova requisição, o mecanismo define novos instantes de envio ou tempos de resposta. Isto gera uma distribuição $p_{v'}$ a partir das distribuições de tempo de resposta $p_v(A^d)$ para manipular o comportamento futuro de tal forma que não representem uma característica única por dispositivo na rede $d \in D$.

O módulo tem o objetivo de preservar a privacidade dos dispositivos com base nos conjuntos herdados do módulo anterior. Assim, denotamos o módulo de proteção de privacidade a partir da observação do fluxo de pacotes de um dispositivo IoT, observados através das variáveis $X_1, X_2, ..., X_i$ como uma sequência discreta de tempo de tempos de resposta por requisição, onde cada X_i pode assumir um valor em intervalos de $x \in \chi = \{a_1, a_2, ..., a_{|\chi|}\}$, onde χ é um conjunto pré-determinado de todos os possíveis tempos de resposta médios a_i de cada dispositivo. Contudo, o mecanismo precisa modificar x de tal forma que ele mascare a verdadeira distribuição $p_v(\chi)$ na perspectiva de um observador central G.

Conforme a literatura, este módulo considera dois métodos principais que implementam de forma prática a modificação do comportamento dos dispositivos com base nos tempos de resposta observados. A modificação do comportamento manipula as capturas de tráfego de qual forma que dificulta a identificação dos dispositivos através de análises realizadas sobre o tempo. Assim, o mecanismo considera dois métodos que compreendem a inserção de atrasos $\tau' = \tau + \gamma$ no núcleo do sistema operacional dos dispositivos d ou o envio de uma requisição falsa m_f. Desta forma, o método de inserção de atrasos procura aproximar as diferenças médias entre todas as distribuições de tempos de resposta por dispositivo e forma centralizada. No método de pacotes falsos, o módulo cria mensagens de requisição falsas visando diminuir a diferença entre os instantes de envio dos pacotes, dificultando os métodos utilizados para calcular do tempo de resposta. Elas são direcionadas para um dispositivo real e recebe os mesmos tratamentos de uma mensagem real. Este método visa aproximar os instantes de envio e recebimento capturados.

O *gateway* da rede de área pessoal executa o mecanismo. Então ele realiza atividades como a captura do tráfego da rede IoT e o compartilhamento de dados entre os módulos. O módulo de teste de vulnerabilidade monitora o tráfego de rede através de um *sniffer*. O tráfego capturado e amostrado é armazenado em um espaço de memória compartilhado entre os módulos, o que possibilita acesso livre aos dados. O módulo de teste de vulnerabilidade é executado constantemente, diferentemente do módulo de proteção de privacidade. Em um primeiro momento do ciclo de execução, o módulo de proteção de privacidade precisa ser iniciado pelo módulo de teste de vulnerabilidade. Uma vez que iniciado, o módulo de proteção de privacidade se ajusta a cada novo vazamento detectado. Portanto, o módulo de teste de vulnerabilidade define uma quantidade η de requisições por dispositivo. Desta forma, este módulo só inicia as funções de amostragem quando todos os dispositivos receberem η requisições. Assim, conforme a Figura 4.3, ao coletar $|Traf^d| = \eta$ requisições por dispositivos o mecanismo aplica as funções

de amostragem e classificação. Caso um vazamento temporal for identificado, o módulo de proteção de privacidade, baseado nas amostragens realizadas, define um novo comportamento para cada dispositivo e solicita uma nova avaliação pelo módulo de teste de vulnerabilidades. Esta relação entre os módulos é executada sempre que um novo comportamento vulnerável for identificado pelo módulo de teste de vulnerabilidade.

4.3 RESUMO

Este capítulo apresentou o mecanismo FISHER de defesa contra os ataques *traffic side-channel* temporais na IoT. O mecanismo protege a privacidade dos dispositivos que compõe uma rede IoT padronizada conforme os protocolos IEEE 802.15.4, 6LoWPAN, RPL, UDP e CoAP. Ele segue dois módulos, o primeiro testa as vulnerabilidades dos vazamentos temporais, o segundo máscara os vazamentos vulneráveis. Desta forma, um ciclo de atividades define os momentos em que cada módulo atua, com o objetivo de proteger a confidencialidade dos dispositivos em rede e a privacidade dos usuários. O capítulo apresentou também um exemplo do funcionamento geral do mecanismo proposto e como ele é executado na rede.

5 AVALIAÇÃO

Este capítulo avalia o mecanismo FISHER e responde as seguintes questões de pesquisa: (i) Qual o impacto das análises realizadas sobre os vazamentos *side-channel* na privacidade da rede? (ii) Qual a eficácia ao implementar os métodos de geração de pacotes falsos e inclusão de atrasos no contexto da IoT? Como descrito na Seção 3.1, os ataques *traffic side-channel* temporais são pouco explorados no contexto da IoT. Portanto, este trabalho implementa um estudo de caso que avalia os efeitos deste tipo de ataque contra os vazamentos temporais. Assim, dois experimentos foram conduzidos considerando diferentes cenários de rede, compostos por dispositivos idênticos, executando o conjunto de protocolos padronizados para IoT. Estes experimentos procuram demonstrar que, mesmo com duas arquiteturas de hardware diferentes, os ataques *traffic side-channel* temporais podem revelar informações privadas sobre o uso dos dispositivos. Além disso, os experimentos e as avaliações auxiliaram na análise da eficiência do mecanismo ao identificar e mascarar os vazamentos explorados por este tipo de ataque. Portanto, este capítulo apresenta as avaliações e os resultados do mecanismo FISHER e responde as questões de pesquisa.

5.1 METODOLOGIA DE AVALIAÇÃO

Esta seção apresenta as características dos cenários experimentais, as métricas de avaliação e como o tráfego foi caracterizado para todas as avaliações realizadas neste capítulo. Esta avaliação utilizou dois cenários experimentais (CCSC e IoTLaB) para comprovar a existência de vazamentos *side-channel* nas trocas de mensagens em diferentes arquiteturas de *hardware*. O primeiro cenário de experimentação é composto por dispositivos e sistemas operacionais mais simples. O segundo cenário compreende dispositivos mais robustos, com processador e sensores mais modernos. As métricas de avaliação são relacionadas aos classificadores supervisionados empregados para a identificação dos diferentes dispositivos e suas propriedades. Assim, os classificadores recebem como entrada as características extraídas a partir dos tempos de resposta e dos instantes de envio das mensagens trafegadas na rede.

5.1.1 Cenários Experimentais

As avaliações foram desenvolvidas em dois cenários experimentais CCSC e IoTLab. O primeiro cenário CCSC é composto por quatro dispositivos IoT Memsic Iris. Os dispositivos realizam diferentes operações, três dos dispositivos agem como servidores e o quarto como *gateway*. Os dispositivos são equipados com *chips* Atmel's AT86RF230 compatíveis com as especificações IEEE 802.15.4 (Montenegro et al., 2007a). Além disso, placas proprietárias de sensores podem ser acopladas aos dispositivos. As placas utilizadas neste cenário são as do modelo *MTS300CB* equipadas com sensores de iluminação e temperatura. Uma vez que todos os dispositivos IoT executam o Contiki SO, foi necessário desenvolver uma adaptação para o *driver* da placa de sensores que originalmente foi programada para outro sistema operacional. A Figura 5.1 apresenta o cenário CCSC. O segundo cenário IoTLab foi implementado na plataforma de *hardware* do FIT/IoT-Lab (Adjih et al., 2015). O FIT-IoT Lab é uma das maiores plataformas de IoT abertas, ela oferece uma interface de experimentação com acesso a todas as camadas de rede e a instalação de *drivers* e sistemas operacionais customizados. Neste experimento, usamos 9 dispositivos ARM Cortex M3 equipados com *chips* AT86RF231 compatíveis com as

especificações IEEE 802.15.4 (Montenegro et al., 2007a). Estes dispositivos são embarcados com sensores de iluminação, pressão, acelerômetro, magnetômetro e giroscópio. Os dispositivos executavam o Contiki-NG OS, uma versão mais atualizada do sistema operacional utilizada no cenário CCSC. A Figura 5.2 apresenta o cenário IoTLab. Devido ao mecanismo FISHER atuar no contexto IoT, ambos os cenários experimentais utilizados nas avaliações possuem dispositivos IoT que englobam diferentes gerações e possuem baixa capacidade computacional. A principal diferença entre os cenários consiste que no cenário CCSC os dispositivos são mais simples e antigos, enquanto no cenário IoTLab os dispositivos são mais modernos e robustos.

Figura 5.1: Cenário Experimental CCSC

A experimentação compreende o tráfego de dados empíricos, gerados e capturados por meio dos cenários experimentais. Nestes cenários, os dispositivos só realizam a captura do tráfego dos sensores a partir do recebimento de uma requisição. Os protocolos de rede utilizados são: 6LoWPAN (Montenegro et al., 2007b), RPL (Thubert et al., 2017), UDP e CoAP (Shelby et al., 2014). Um dos dispositivos atua como *gateway* e roteador de borda, ou seja, recebe e encaminha as requisições externas. Estas requisições são geradas pelo computador conectado por meio de uma interface serial/USB com o *gateway*. Ele simula um cliente CoAP que gera requisições e captura o tráfego gerado. As capturas de rede são realizadas através da ferramenta Wireshark [1]. Neste contexto, um cliente CoAP é compreendido como uma aplicação de monitoramento do ambiente, medindo rotineiramente o status do ambiente.

5.1.2 Métricas de Avaliação

Para avaliar o mecanismo proposto e os vazamentos temporais, foram utilizadas diferentes métricas de desempenho. Em particular, essas métricas avaliam o comportamento dos algoritmos de classificação. Elas envolvem a acurácia, precisão, *recall* e *F-score* (também conhecida como *F-measure*). Tais métricas consideram a taxa de verdadeiro positivo (VP), verdadeiro negativo (VN), falso positivo (FP) e falso negativo (FN). Dessa forma, estatisticamente a acurácia se refere à proporção de tráfego de dados classificados corretamente em relação a todas as amostras de tráfego. A precisão (p) estima a porcentagem de verdadeiros positivos dentre todos os exemplos de tráfego classificados como positivos ($VP/(VP+FP)$). O *recall* (r) ou revocação

[1] Wireshark, https://www.wireshark.org/. Último acesso em Fev/2020.

Figura 5.2: Cenário Experimental IoTLab

consiste da porcentagem de verdadeiros positivos dentre todos os exemplos cuja classe esperada é a positiva ($VP/(VP + FN)$). Por fim, o *F-Score* faz uma relação entre as medidas de precisão e *recall* através da estimação da média harmônica ($2rp/(r + p)$). Portanto, os resultados dos classificadores se embasam nessas métricas.

5.1.3 Caracterização do Tráfego

Na caracterização do tráfego de cada dispositivo foram considerados os dados referentes ao instante de envio (T) e tempo de resposta (τ) dos pacotes. Outras características dos dados de tráfego não foram consideradas, como o tamanho do pacote, pois a literatura é pobre em relação a informação tempo para os protocolos da IoT. Além do mais, a possibilidade de caracterizar o comportamento do tráfego de um dispositivo IoT apenas usando a informação tempo aponta ser um vazamento de informação *side-channel* crucial para a garantia da privacidade dos usuários. A caracterização dos dados segue a Tabela 4.2, onde as medidas estatísticas consideradas englobam a mínima (min), soma (sum), média (μ), limiar inferior (LI), limiar superior (LS), moda ($mode$), mediana (Md) e coeficiente de correlação de Pearson (r).

Medida Estatística	**Equação**
Mínima (τ)	$min = min(x_i)$
Soma (τ e T)	$sum = \sum_{i=1}^{N} x_i$
Média (τ)	$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$
Limiar Superior (τ)	$LS = \mu + 1.96\sigma$
Moda (τ e T)	$mode = freq(X)$
Mediana (τ e T)	$Md = \frac{1}{2}(X_{\frac{N}{2}} + X_{\frac{N}{2}+1})$ de $sort(X)$
Coef. Correlação de Pearson (τ e T)	$r = \frac{\sum_{i=1}^{N}(x_i-\mu_x)(y_i-\mu_y)}{\sqrt{\sum_{i=1}^{N}(x_i-\mu_x)^2 \sum_{i=1}^{N}(y_i-\mu_y)^2}}$

X = conjunto de dados; x_i = amostra i dos dados; N = quantidade de amostras; σ = variância; $freq$ = valor mais frequente; $sort$ = valores ordenados.

Tabela 5.1: Medidas Estatísticas para Caracterização do Tráfego

Para algumas medidas estatísticas, os valores de entrada consistem no instante de envio, enquanto outras medidas seguem o tempo de resposta. Essas definições se embasaram no estudo de Selis e Marshall (2017). Para a medida do coeficiente de correlação de Pearson, foi estimado

o grau de correlação entre o instante de envio e o tempo de resposta. Com base nessas medidas, cada subconjunto possui um conjunto específico de informações que contém o tempo de resposta, o instante de envio, as medidas estatísticas e o rótulo. O rótulo se refere à classe alvo necessária para identificar os dispositivos por meio de classificadores supervisionados. A informação verdadeira (*ground-truth information*) sobre qual tráfego pertence a cada dispositivo foi extraída durante a coleta do tráfego no cenário experimental, o que auxiliou na rotulação da base.

5.1.4 Métodos de Defesa

O módulo de proteção de privacidade emprega dois métodos de pacotes falsos e inserção de atrasos. O método de pacotes falsos harmoniza as capturas, forçando que dois ou mais dispositivos atuem de forma semelhante no mesmo instante de tempo. O instante de envio e recebimento dos dispositivos tem um papel importante na identificação dos dispositivos através dos ataques *traffic side-channel*, pois é a única informação que o atacante tem acesso na íntegra. Ele significa o instante de atividade do dispositivo, além de servir como base de cálculo para o tempo de resposta. Assim, a partir dos instantes definimos uma rotina de atividades e um padrão de comportamento. Além disso, é uma variável que serve como base para qualquer cálculo relacionado ao tempo, diante da carência de dados, que representa a realidade de um atacante. Por exemplo, aplicações que exigem poucas requisições diárias, para um dispositivo específico em um horário específico, serão facilmente identificadas diante de capturas simples sobre a atividade dos dispositivos. A abordagem empregada pelo método de pacotes falsos dificulta esta capacidade de identificação definindo que o *gateway*, dispositivo que encaminha as requisições externas à rede, gera uma mensagem de requisição falsa, idêntica, às recebidas e a encaminha para outro destinatário. Copiando as características do dispositivo almejado originalmente pelo remetente da mensagem.

O método de *inserção de atrasos* controla o tempo de resposta acrescentando pequenos intervalos de tempo entre os processos dos dispositivos finais. Assim, modifica-se o *kernel* do sistema operacional dos dispositivos inserindo atrasos por meio de duas abordagens: a primeira considera o relógio da fila de processos e a segunda insere uma operação de espera na função de envio. Na primeira abordagem, o intervalo de tempo definido pelo sistema operacional é maior, oferecendo menor precisão na proteção dos atrasos, sendo cada ciclo do relógio de aproximadamente 10ms. Na segunda abordagem, os atrasos inseridos agregam uma operação de espera na função de envio. Dessa forma, o mecanismo consegue um intervalo de tempo menor, logo, melhor controle sobre o atraso inserido. Ambas as abordagens inserem atrasos pequenos e não interferem no relógio dos dispositivos, evitando os conflitos que podem ser causados com protocolos, aplicações e serviços que dependem do tempo.

No método de *inserção de atrasos*, assume-se que as capturas são representadas por um conjunto $X_i = \{x_1, x_2, ..., x_n\}$ de i dispositivos, onde cada $x \in X$ representa o tempo de médio resposta de n amostras de tamanho k, ou seja, $x_n = (\sum_{j=1}^{k} amostra_j)/k$. Desta forma, a distribuição $P_i(X)$ representa o comportamento de cada dispositivo X_i. Os atacantes se beneficiam ao conhecer previamente uma distribuição, ou ao identificar distribuições diferentes. Para mascarar os dados referentes ao tempo de resposta precisa-se definir uma nova distribuição $\rho_i(X) = \hat{X}$ para que as distribuições se aproximem ao máximo, ou não ofereçam um padrão característico único de cada dispositivo. Assim, aplicamos as duas abordagens referentes ao método de *inserção de atrasos*. Na primeira o mecanismo define que ρ é um valor aleatório, onde, considerando a fila de processos do dispositivo, assume ciclos de aproximadamente 10ms. Desta forma, definimos que para cada nova requisição, os dispositivos acrescentam aleatoriamente um atraso entre zero a dois ciclos em ΔT_1. A segunda calcula um valor médio para cada X_i (conforme a equação 5.1) e define um novo conjunto ordenado $D = \{M_1, M_2, ..., M_i\}$ de valores

médios. Este novo conjunto possibilita o cálculo de um atraso r_i ideal para somar com ΔT_1 de cada dispositivo final para aproximar as diferenças entre as distribuições.

$$r_{i+1} = \max(D) - M_{i+1} \tag{5.1}$$

Quatro abordagens que mascaram os vazamentos foram avaliadas: **A1**, **A2**, **A3** e **A4**. A primeira abordagem **A1** segue o método de *pacotes falsos*, por isso ele duplica as requisições recebidas pelo *gateway* de rede, forçando com que mais de um dispositivo opere ao mesmo tempo. As próximas duas abordagens implementam o método de *inserção de atrasos*. **A2** considera os ciclos do relógio de controle de processos, inserindo de forma aleatória os atrasos. **A3** inclui uma operação de espera, que oferece um controle mais preciso sobre os atrasos. Assim, nesta abordagem o dispositivo com menor tempo médio de resposta incrementa o atraso de processamento com um valor equivalente à diferença entre os tempos médios de resposta dos outros dispositivos. A abordagem **A4** implementa uma versão mista dos dois métodos apresentados, ou seja, gera um pacote falso com um atraso aleatório entre 0ms e 10ms. As trés primeiras abordagens foram implementadas e avaliadas no cenário CCSC, as abordagens **A1** e **A4** no cenário de experimentação IoTLab. Estas abordagens implementam os diferentes métodos de manipulação das variáveis de tempo apresentados no Capítulo 4, com o objetivo de evitar que as análises estatísticas realizadas pelos ataques *traffic side-channel* identifiquem os dispositivos. Tais abordagens foram escolhidas por respeitarem o fluxo de operações computacionais, evitando assim o mau funcionamento de outros protocolos ou serviços. Para testar o mascaramento, utilizamos dois algoritmos de aprendizagem de máquina amplamente utilizados na literatura (Pacheco et al., 2018) e previamente avaliados na Seção 5.2 e em Prates et al. (2019a): o algoritmo baseado em árvores de decisão *Random Forest* e a rede neural *Multilayer Perceptron*. Para validar os algoritmos de classificação, a avaliação utiliza uma abordagem tradicional que emprega um conjunto de dados de treino (70% dos dados) e teste (30% dos dados), a fim de treinar os modelos de classificação e computar as métricas de desempenho (Pacheco et al., 2018).

5.2 ANÁLISE DOS VAZAMENTOS TEMPORAIS

Esta seção apresenta uma análise sobre os vazamentos temporais *side-channel*, a fim de caracterizar e identificar dispositivos idênticos em uma rede IoT e responder à questão de pesquisa (i). Esta análise segue as etapas de experimentação, caracterização e identificação, conforme mostrado na Figura 5.3 e fundamentado na Subseção 2.2.1.2. A caracterização emprega a ferramenta de análises estatísticas RStudio[2] para selecionar e extrair as características sobre as capturas de rede, como o instante de envio e o tempo de resposta. Embasado nessas características, a etapa de identificação analisa o comportamento único de cada dispositivo IoT através de classificadores. Na ferramenta de mineração de dados WEKA[3], aplicam-se os classificadores de aprendizagem de máquina para identificar os dispositivos. Além disso, analisa-se a eficiência de cada classificador através de métricas como acurácia e *F-Score*. As próximas subseções detalham a avaliação de desempenho e discutem os resultados.

5.2.1 Avaliação de Desempenho - Cenário CCSC

Este estudo analisa o vazamento de informações *side-channel* e o comportamento do tráfego no cenário experimental CCSC. Neste cenário, um cliente realiza requisições para três dispositivos IoT Memsic Iris que agem como servidores CoAP. A partir da captura, na ferramenta

[2]RStudio, https://www.rstudio.com/. Último acesso em Mar/2019.

[3]WEKA, https://www.cs.waikato.ac.nz/ml/weka/. Último acesso em Mar/2019.

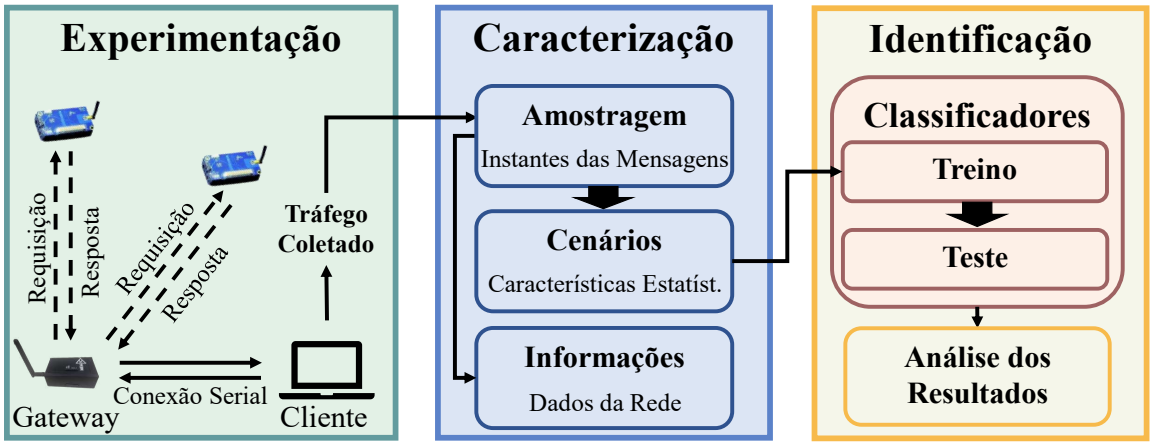


Figura 5.3: Etapas da Análise dos Vazamentos

de análises estatísticas RStudio foram extraídas as características relevantes em relação ao tráfego, como tempo de resposta e tempo de envio dos dispositivos. Além disso, foram computadas as medidas estatísticas como média, mínima e moda. Tais características serviram como entrada para a caracterização do comportamento específico de cada dispositivo e para a identificação do tráfego deles. Na ferramenta de mineração de dados WEKA, foram identificados os dispositivos IoT por meio de classificadores, como *Naive Bayes* e KNN. As próximas subseções apresentam a seleção e extração de características, os cenários de avaliação e uma discussão dos resultados.

O conjunto de dados empregado nessa análise foi coletado a partir de três dispositivos embarcados com sensores de monitoramento de iluminação e de temperatura ambiente. Os dados estão distribuídos em seis subconjuntos, intitulados de *Nó 1-l*, *Nó 2-l* e *Nó3-l* para os sensores de iluminação e *Nó 1-t*, *Nó 2-t* e *Nó 3-t* para os sensores de temperatura. Cada subconjunto de dados contém uma coleta de 100.000 requisições, as quais foram divididas em 1.000 amostras de 100 requisições. A captura do tráfego foi realizada na perspectiva do **cliente**, que registrou um total de 1.226.428 pacotes transmitidos divididos em 626.428 requisições e 600.000 respostas. Além disso, como o objetivo deste estudo consiste em analisar o vazamento de informações *side-channel* de dispositivos IoT, os dados coletados consistem em gravações normais, sem possuir nenhum tipo de ataque ou variação significante.

Dois cenários de avaliação (**C1** e **C2**) foram considerados nas análises dos vazamentos temporais dos dispositivos IoT. Nestes cenários variou-se a quantidade de medidas estatísticas do tráfego de rede utilizadas nas análises. Para isso, foi criado um subconjunto das características estatísticas apresentadas na Tabela 5.1, que engloba apenas a média e a moda do tempo de resposta. Assim, no primeiro cenário (**C1**) foi submetido o tempo de resposta e o subconjunto de características estatísticas criado. Enquanto, no segundo cenário (**C2**) foram utilizadas todas as características, tanto do instante de envio, quanto do tempo de resposta. Dessa forma, é possível analisar o impacto das medidas estatísticas na caracterização do tráfego de dispositivos idênticos.

A identificação dos dispositivos IoT engloba cinco algoritmos de aprendizagem de máquina amplamente utilizados e bem conhecidos na literatura (Pacheco et al., 2018). Tais algoritmos compreendem o algoritmo clássico não paramétrico *K Nearest Neighbors* (KNN), os algoritmos baseados em árvores de decisão *Random Forrest* e *J48* (também conhecido como C4.5), a rede neural *Multilayer Perceptron*, e por fim, o algoritmo embasado na classificação bayesiana *Naive Bayes*. Esses algoritmos são aplicados em problemas de multi-classificação, ou seja, em situações que várias classes de dados precisam ser identificadas. Esta análise segue

um problema de multi-classificação, pois objetiva-se identificar o tráfego de dados dos três dispositivos IoT considerados, o que justifica a escolha de tais algoritmos de classificação.

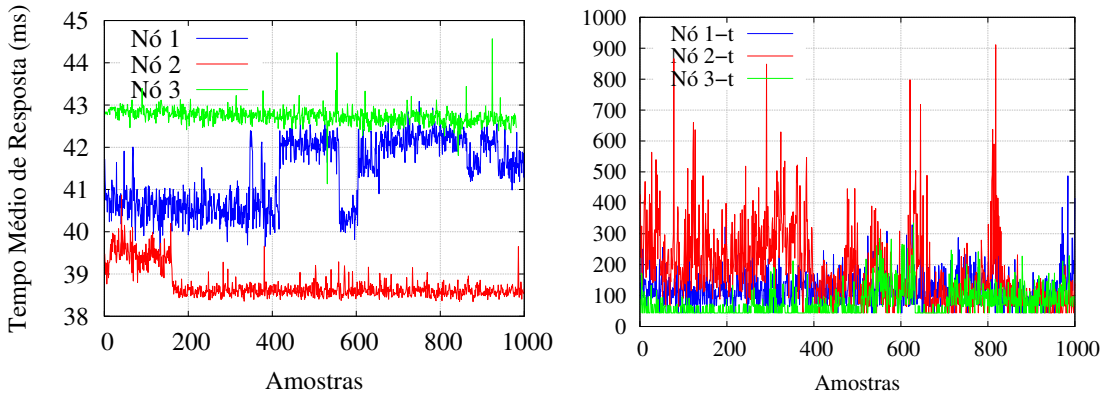
Para validar os algoritmos de classificação seguiu-se a abordagem tradicional que emprega um conjunto de dados de treino (70% dos dados) e teste (30% dos dados), a fim de treinar os modelos de classificação e computar as métricas de desempenho (Pacheco et al., 2018). O conjunto de dados de treino possui uma captura de 100.000 requisições de dados para cada um dos seis subconjuntos, totalizando 600.000 requisições. As capturas de cada dispositivo com seu respectivo sensor foram divididas em conjuntos de 1.000 amostras, a fim de calcular as características estatísticas da Tabela 5.1. Assim, após pré-processamento, foi criado um conjunto de treino para ambos os sensores (temperatura e luz) com 3.000 exemplos de dados em cada, incluindo o rótulo dos três dispositivos IoT. O conjunto de dados de teste possui uma captura de 10.000 requisições de dados para cada subconjunto dos dispositivos IoT. Cada captura foi dividida em 100 amostras para computação das características estatísticas. Dessa forma, o conjunto de teste totalizou 300 exemplos de dados para cada sensor.

5.2.1.1 Resultados

Esta seção apresenta os resultados de caracterização e identificação do tráfego de dispositivos IoT por meio do vazamento temporal *side-channel*. Os resultados seguem dois cenários de avaliação **C1** e **C2**. Além disso, os resultados se embasam no tráfego de três dispositivos IoT idênticos coletados em um cenário experimental. Tais dispositivos possuem dois tipos de sensores: um sensor que monitora a iluminação e outro que monitora a temperatura ambiente. Assim, os resultados são apresentados e discutidos seguindo uma análise crítica para cada dispositivo com seu respectivo sensor.

Em relação a caracterização do tráfego, as Figuras 5.4(a) e 5.4(b) apresentam os comportamentos no tráfego de dados dos três dispositivos IoT Memsic Irirs considerados em nossas análises. A Figura 5.4(a) mostra o tempo médio de resposta dos três dispositivos com sensores que monitoram a iluminação do ambiente. Enquanto, a Figura 5.4(b) apresenta o comportamento do tempo médio de resposta dos dispositivos com sensores de temperatura. Para cada dispositivo o tráfego de dados foi dividido em conjuntos de 1.000 amostras a fim de obter o tempo médio de resposta das requisições. Dessa forma, é possível observar um comportamento característico de cada dispositivo com seu respectivo sensor em relação ao seu tempo de resposta. Entretanto, vale observar que os dispositivos com sensores de temperatura possuem um tempo médio de resposta maior do que os de iluminação.

Mais especificamente, na Figura 5.4(a) o dispositivo *Nó 1* alcançou uma variação maior em seu tempo médio de resposta, atingindo valores em torno de 39 a 43ms. Em contrapartida, o dispositivo *Nó 2* apresentou uma leve queda no seu tempo médio de resposta em torno da amostra de número 200, diminuindo de 40ms para 37ms. Após essa queda, o dispositivo estabilizou com valores próximo a 37ms. Por fim, o dispositivo *Nó 3* apresentou um comportamento estável ao longo de todas as amostras de tráfego, onde seu tempo médio de resposta ficou em torno de 42 e 43ms. Desse modo, pode-se observar uma diferença clara no comportamento de cada dispositivo. Por outro lado, os dispositivos com sensores de temperatura da Figura 5.4(b) apresentaram um comportamento mais semelhante alcançando picos de valores de tempo médio de até 900ms para o *Nó 5*. Por se tratar de dispositivos idênticos da mesma marca, modelo e com os mesmos protocolos e aplicações em execução, esperava-se um comportamento semelhante para ambos os sensores. Entretanto, através dos resultados observados é possível notar um comportamento específico para cada sensor. A justificativa para o sensor de temperatura alcançar valores maiores consiste no tipo de fenômeno monitorado, pois além das características dos dispositivos serem iguais, a geração e a coleta do tráfego seguiram os mesmos passos.



(a) Sensores de Iluminação

(b) Sensores de Temperatura

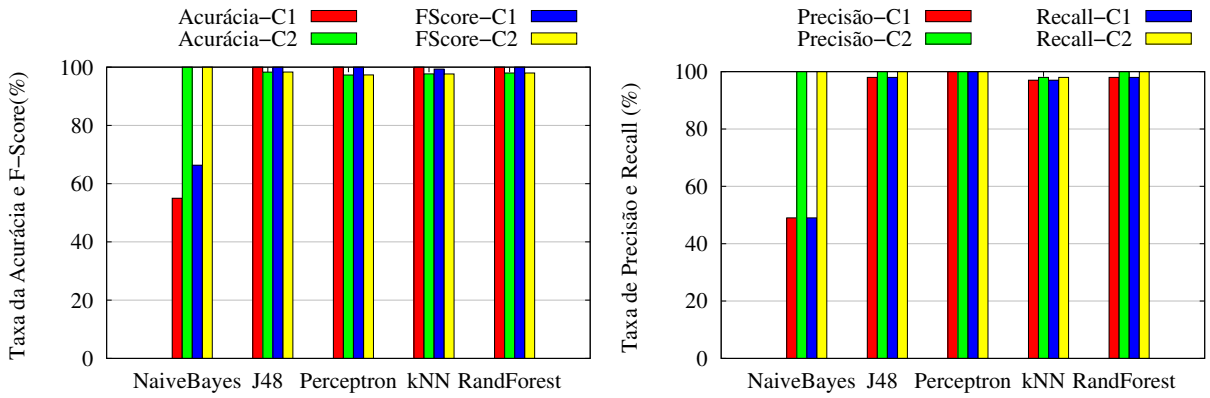
Figura 5.4: Comportamento do Tempo de Resposta dos Dispositivos

A Tabela 5.2 apresenta maiores detalhes sobre cada dispositivo com seu respectivo sensor. Como durante a geração e a coleta de tráfego de dados foi aplicado o mesmo tamanho de pacote de dados, bem como o mesmo número de requisições, a taxa de retransmissão e a média de pacotes por segundo apresentaram comportamentos similares para os três dispositivos com seus sensores. Contudo, a quantidade de requisições aumentou devido às taxas de retransmissões dos dispositivos, como por exemplo o *Nó 1-l* atingiu uma taxa de retransmissão de 7.84%. Assim, através dos resultados sobre o comportamento do tráfego de cada dispositivo observou-se que mesmo dispositivos IoT idênticos possuem um comportamento único que os diferencia no que tange o tempo médio de resposta. Tal comportamento pode ser analisado apenas considerando a característica sobre o tempo de resposta dos dispositivos. Todavia, essa informação não possui nenhuma proteção ou técnica de criptografia, o que facilita a execução de ataques *traffic side-channel*. A maioria dos administradores de rede utilizam técnicas de gerenciamento de segurança que criptografam apenas o conteúdo do pacote e outras informações do cabeçalho do pacote, se abstendo de informações consideradas não tão relevantes como o tempo de reposta. Entretanto, por meio dos resultados apresentados é possível observar que o vazamento *side-channel* possui informações básicas que permitem caracterizar os dispositivos conectados na rede.

	Sensor de iluminação			**Sensor de temperatura**		
	Nó 1-l	*Nó 2-l*	*Nó 3-l*	*Nó 1-t*	*Nó 2-t*	*Nó 3-t*
Taxa de Retransmissão	7.84%	6.19%	3.18%	2.91%	4.98%	1.3%
Média de Pacotes/s	4.5	5.1	6.1	6.33	5.33	7.2
Tempo de Resposta Médio	41.36	38.74	46.11	119.47	182.87	77.21

Tabela 5.2: Detalhes dos Dispositivos IoT

Em relação a identificação do tráfego de dados dos dispositivos, os gráficos da Figura 5.5 apresentam os resultados referentes ao desempenho dos cinco classificadores para os sensores de iluminação. Tais resultados seguiram os cenários de avaliação **C1** e **C2**. Conforme o gráfico da Figura 5.5(a), a maioria dos classificadores alcançaram taxas de acurácia e *F-Score* próximo a 97%. Apenas o classificador *Naive Bayes* apresentou um resultado pobre para o cenário **C2**, pois, neste cenário foram utilizados apenas os valores sobre a moda e a média do tempo de resposta. Porém, no cenário **C1** foram utilizadas todas as medidas estatísticas, melhorando assim o desempenho do *Naive Bayes* e dos demais classificadores. O que comprova que quanto maior o nível de detalhamento dos dados, melhor será o resultado de identificação. Um desempenho semelhante dos classificadores pode ser visto na Figura 5.5(b). Conforme as taxas de precisão e



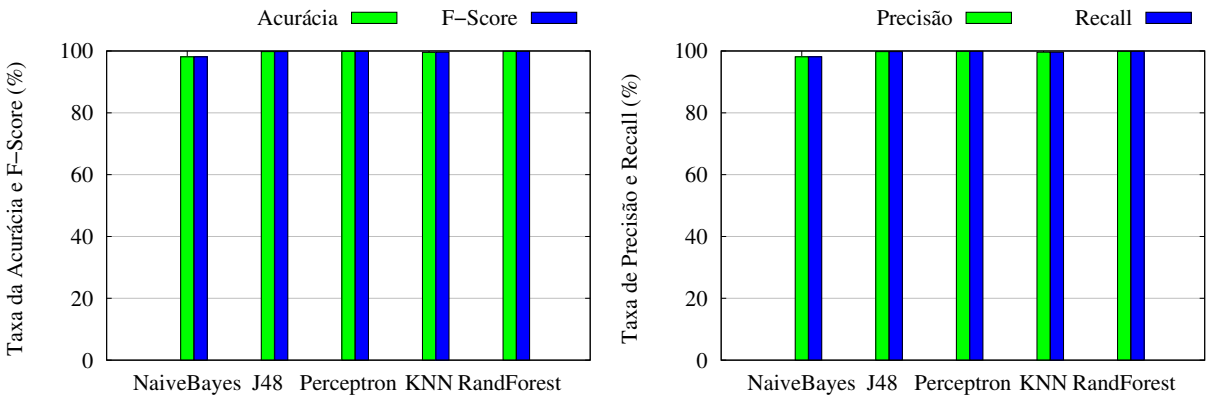
(a) Acurácia e *F-Score*

(b) Precisão e Recall

Figura 5.5: Desempenho dos Classificadores com Sensores de Iluminação

recall, a proporção de dispositivos classificados corretamente é alta. Dessa forma, com base no desempenho obtido por meio dos classificadores comprova-se a possibilidade de identificar dispositivos IoT idênticos apenas utilizando informações relacionadas ao tempo de resposta e o instante de envio dos pacotes de dados.

Para os dispositivos IoT com sensores de temperatura os resultados dos cinco classificadores também atingiram um desempenho altamente satisfatório, como pode ser observado nas Figura 5.6(a) e Figura 5.6(b). Devido aos resultados anteriores alcançarem valores ótimos para o cenário **C1**, que emprega todas as medidas estatísticas, esta segunda análise considerou apenas este cenário. Nesta análise todos os classificadores alcançaram valores em torno de 90% e 99% de acurácia, *F-Score*, precisão e *recall*. Esses resultados observados reforçam as análises anteriores, o que comprova a identificação de dispositivos idênticos por meio de vazamento *side-channel*. No sentido de ataques *traffic side-channel*, os atacantes podem identificar os dispositivos e assim, se passar por eles e roubar informações privadas dos usuários. Dessa forma, os resultados obtidos apontam a importância da informação tempo e a necessidade do desenvolvimento de técnicas de segurança que considerem tais informações.



(a) Acurácia e *F-Score*

(b) Precisão e Recall

Figura 5.6: Desempenho dos Classificadores com Sensores de Temperatura

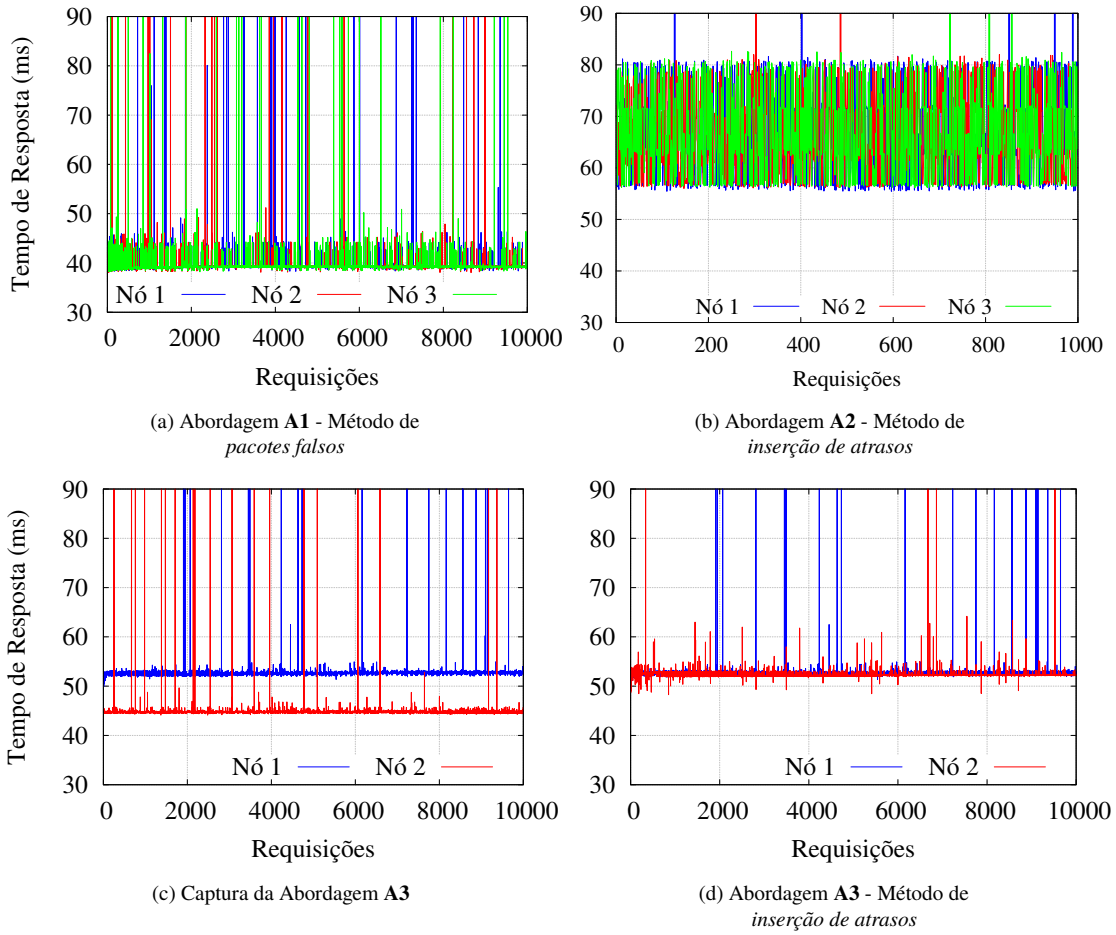
Os resultados das análises realizadas sobre o mecanismo proposto. O módulo de teste de vulnerabilidade de vazamentos *side-channel* procura classificar os dispositivos que apresentam maior dispersão sobre os conjuntos de dados, essas dispersões representam o comportamento dos dispositivos. O módulo de proteção de privacidade máscara as informações relacionadas ao

tempo, a fim de melhorar a privacidade dos usuários e dos dispositivos em relação aos ataques *traffic side-channel*. Para isso, os resultados são referentes a avaliação de três abordagens **A1**, **A2** e **A3**, onde a abordagem **A1** implementa o método de *pacotes falsos* do mecanismo proposto e as abordagens **A2** e **A3** implementam o método de *inserção de atrasos* do mecanismo. Além disso, os resultados se embasam no tráfego de três dispositivos IoT idênticos com sensores de iluminação coletado em um cenário experimental. Assim, os resultados são apresentados e discutidos seguindo uma análise crítica para cada abordagem e dispositivo considerado.

Com o intuito de mascarar os vazamentos *side-channel*, os gráficos da Figura 5.7 apresentam o tempo de resposta *versus* o número de requisições de cada dispositivo para cada abordagem considerada. Inicialmente, a Figura 5.7(a) mostra os resultados da abordagem **A1**, onde a variável tempo de cada dispositivo é controlada conforme o módulo de *pacotes falsos*. Ou seja, duplicam-se as requisições recebidas pelo *gateway* para que os dispositivos operem simultaneamente. Em consequência, o tempo de resposta dos dispositivos alcança valores muito semelhantes, o que dificulta a ocorrência dos ataques temporais *traffic side-channel*. Na abordagem **A1**, todos os dispositivos atingiram tempo de médio de resposta por volta de 44ms, sendo clara a máscara aplicada sobre tempo de resposta. Ao ser avaliado pelo módulo de teste de vulnerabilidade, os classificadores tiveram uma queda considerável nas taxas das métricas de desempenho, tendo uma queda de aproximadamente 63% na acurácia (Figura 5.8(a)). Além disso, esta abordagem possui algumas vantagens, por exemplo, para controlar o tempo não foi necessário alterar o atraso de processamento (tempo que o dispositivo leva desde o recebimento até o envio das respostas). Também, por apresentar um tempo de resposta menor em relação as outras abordagens da Figura 5.7, há uma menor chance de prejudicar outros protocolos e/ou aplicações (por exemplo, não prejudica mecanismos de garantia de entrega dos pacotes de dados).

Em contrapartida, na abordagem **A2** os atrasos foram inseridos pelo módulo de *inserção de atrasos* de forma aleatória conforme os ciclos do relógio de controle de processos. Os resultados podem ser observados na Figura 5.7(b), onde os valores para o tempo de resposta dos três dispositivos se sobrepõem, alcançando parcialmente o objetivo de mascarar a o vazamento temporal. No entanto, os classificadores não apresentaram dificuldade ao identificar os dispositivos, mantendo as métricas em 100%, como pode ser observado na Figura 5.8(b). Para a abordagem **A3**, explorou-se a capacidade de controle sobre os tempos médios de resposta, conforme pode ser observado nas Figuras 5.7(c) e 5.7(d). Na Figura 5.7(c) os dispositivos não possuem nenhum tipo de inserção de atraso, então *Nó 1* e *Nó 2* apresentaram tempos médios de resposta de aproximadamente 57ms e 44ms, respectivamente. Na Figura 5.7(d), o *Nó 2* soma a cada requisição o valor que representa a diferença entre os tempos médios de resposta, ou seja, 13ms. Dessa forma, é possível notar que o tempo de resposta dos dispositivos se aproximam após a inserção do atraso. Estes resultados podem ser considerados positivos, pois mostraram que podemos ter determinado controle sobre a distribuição dos tempos médios de resposta diante da aleatoriedade das variáveis. Apesar disso, o impacto na capacidade de identificação dos dispositivos não foi muito considerável, conforme mostra a Figura 5.8(c), na qual o classificador *Random Forest* ainda conseguiu classificar com sucesso de 95% da base de teste.

Através dos resultados obtidos, observa-se que é possível diminuir a dispersão dos dados relacionados ao tempo e com isso dificultar a realização dos ataques *traffic side-channel*. Na seção anterior os classificadores *Multilayer Perceptron* e *Random Forest* identificaram os dispositivos através dos vazamentos temporais *side-channel* com precisão de 100%. Nesta seção, observa-se que o módulo de teste de vulnerabilidade oferece informações detalhadas que possibilitam o processamento das variáveis temporais, e consequentemente oferece informações precisas para o módulo de proteção de privacidade atuar de forma eficiente. Isto é mostrado nas Figuras 5.7(c) e 5.7(d), onde foi possível inserir atrasos, aproximando os tempos médios



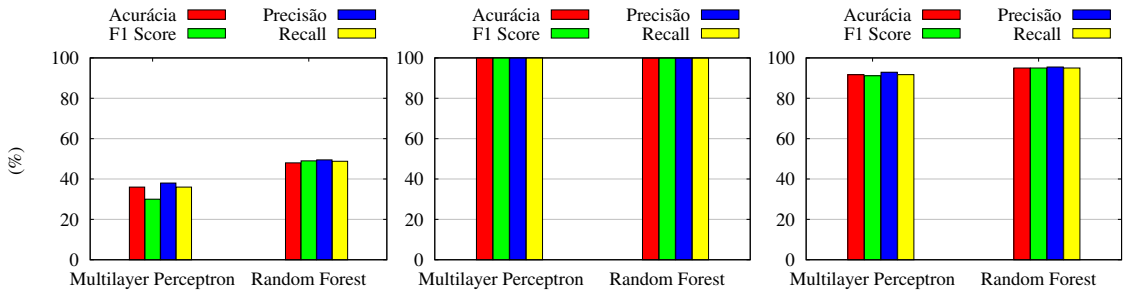
(a) Abordagem **A1** - Método de *pacotes falsos*

(b) Abordagem **A2** - Método de *inserção de atrasos*

(c) Captura da Abordagem **A3**

(d) Abordagem **A3** - Método de *inserção de atrasos*

Figura 5.7: Resultados sobre as Abordagens



(a) Identificação em **A1**

(b) Identificação em **A2**

(c) Identificação em **A3**

Figura 5.8: Desempenho dos Classificadores na Identificação dos Dispositivos

de resposta com uma determinada precisão. Além disso, nota-se que é possível mascarar estes vazamentos através das variáveis que influenciam as capturas realizadas sobre o tempo. No método de *pacotes falsos* observa-se que os instantes de envio e recebimento são os parâmetros mais influentes na identificação dos dispositivos, pois quando controlados, as métricas de desempenho caíram significativamente (Figura 5.8(a)) se comparados aos resultados das abordagens **A2** e **A3** (Figuras 5.8(b) e 5.8(c)), que implementam o método de *inserção de atrasos*. Estes resultados respondem a questão de pesquisa (ii), provando que os métodos de pacotes falsos e inserção de atrasos são eficientes, no entanto, depende de como eles são aplicados.

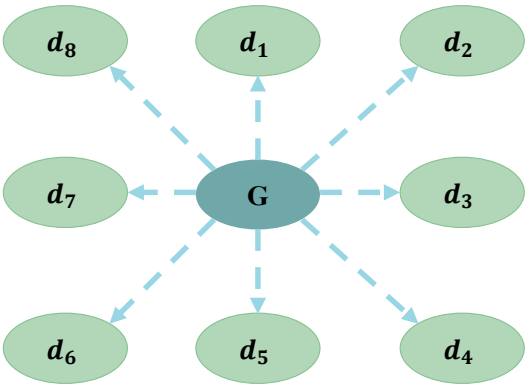


Figura 5.9: Cenário de Rede IoT Lab

5.2.2 Avaliação de Desempenho - Cenário IoTLab

Esta avaliação compreende identificar os vazamentos *side-channel* temporais, quais características eles revelam sobre a estrutura de rede e qual a eficiência da identificação diante de uma rede com dispositivos mais modernos. Uma vez que no cenário de avaliação anterior (CCSC) foram utilizados dispositivos mais antigos, neste cenário IoTLab foram empregados dispositivos mais recentes para avaliar os vazamentos temporais sob diferentes configurações. Para este fim, este cenário utiliza a plataforma FIT-IoT Lab (Adjih et al., 2015) que oferece uma interface de controle e experimentação de dispositivos IoT sem fio embarcados com sensores de iluminação, pressão, acelerômetro, magnetômetro e giroscópio. A Figura 5.9 mostra a posição dos dispositivos em rede, onde o *gateway* gera tráfego para os oito dispositivos IoT. Com base nos resultados alcançados no cenário de avaliação anterior CCSC, o cenário IoTLAb considerou apenas os algoritmos KNN, *Randon Forest* e *Multilayer Perceptron*.

Neste cenário, cada dispositivo IoT possui cinco sensores embarcados. Logo, cada dispositivo *Nó j* contém cinco atributos: iluminação *Nó j-i*, pressão *Nó j-p*, acelerômetro *Nó j-a*, magnetômetro *Nó j-m* e giroscópio *Nó j-g*. Esta análise compreendeu uma captura de tráfego realizada na perspectiva do *gateway* e registrou 560.701 pacotes transmitidos. Os dados compreendem 40 subconjuntos (1 subconjunto para cada sensor de um determinado dispositivo), onde cada subconjunto contém 10.000 requisições. Para analisar a identificação dos dispositivos e sensores em diferentes circunstâncias, a organização dos dados dos subconjuntos segue dois diferentes cenários de avaliação: **C3** e **C4**. O cenário de avaliação **C3** tem o objetivo de identificar os dispositivos conforme os detalhes e características estatísticas utilizados no cenário **C2** (Seção 5.2.1). Desse modo, neste cenário os classificadores receberam como entrada 40 subconjuntos (1 subconjunto para cada sensor de um determinado dispositivo), divididos em 1000 amostras com as características estatísticas (Tabela 5.1) a cada 10 requisições.

O cenário **C4** tem o objetivo de identificar padrões entre os tempos de resposta e os sensores iguais embarcados em dispositivos diferentes. Assim, neste cenário as amostras foram divididas em 5 subconjuntos, onde das 80.000 requisições por tipo de sensor, cada subconjunto é composto por 8.000 amostras de características estatísticas extraídas de cada 10 requisições e rotuladas pelos nomes dos sensores (*i, p, a, m*), independente dos dispositivos. Este cenário considera apenas as características estatísticas mínima, soma, média, limiar superior e moda extraídos dos tempos de resposta. Em ambos os cenários (**C3** e **C4**), cada subconjunto de entrada foi dividido em 70% de amostras de treino e 30% de teste.

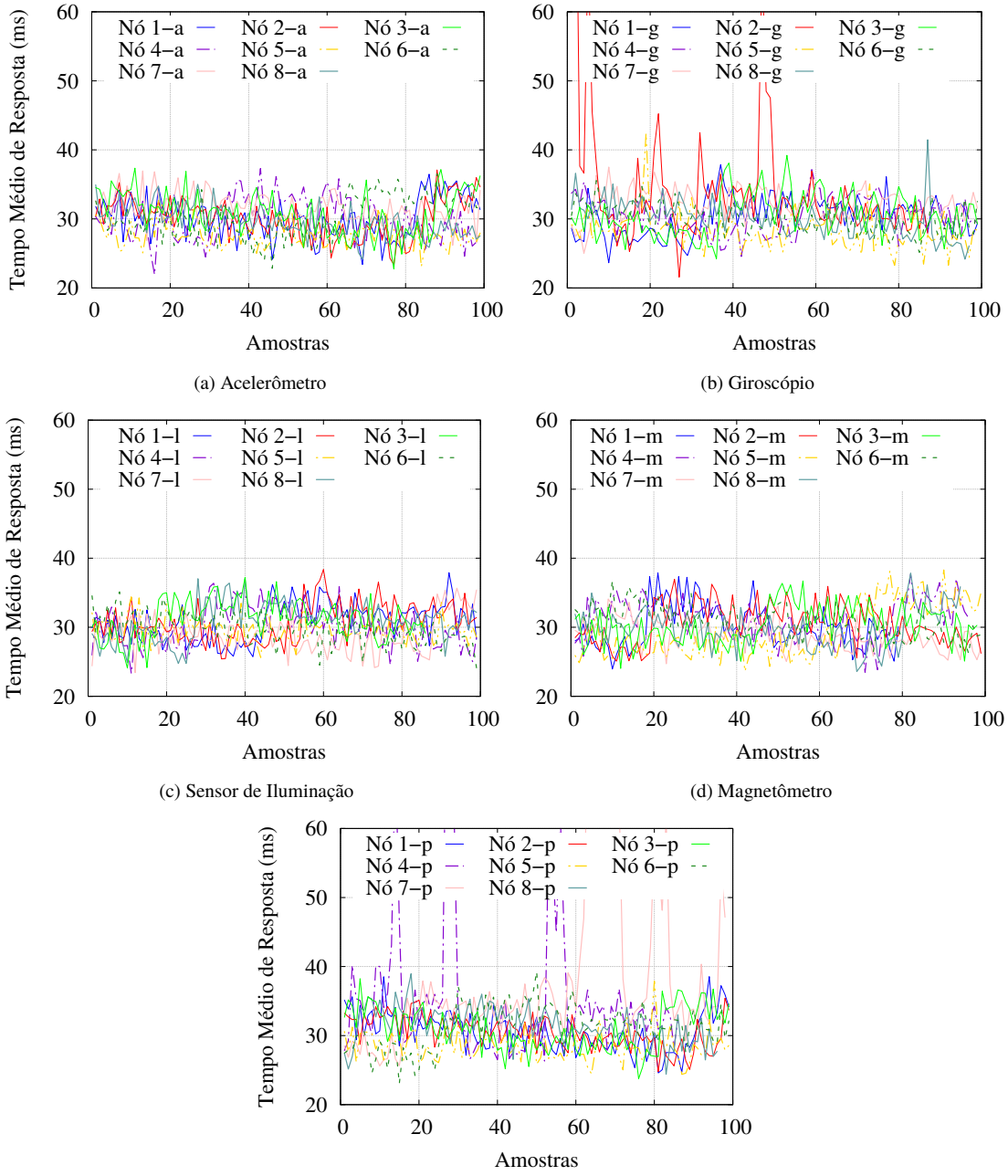
5.2.2.1 Resultados

Esta seção apresenta os resultados das avaliações realizadas conforme os cenários **C3** e **C4**. Os cenários organizam e avaliam as amostras de diferentes formas com o objetivo de inferir informações sobre os dispositivos por meio da caracterização e classificação das variáveis referentes aos instantes de envio e tempo de resposta. Os resultados apontam os vazamentos *side-channel* temporais encontrados em uma rede composta por 8 dispositivos, cada um equipado com 5 sensores: iluminação, pressão, acelerômetro, magnetômetro e giroscópio. Dessa forma, esta seção apresenta, analisa e discute os resultados alcançados.

Os gráficos da Figura 5.10 apresentam os resultados referentes a caracterização dos comportamentos dos dispositivos e seus respectivos sensores no cenário de avaliação **C3**. Os comportamentos observados neste cenário apresentam uma impressão diferente dos comportamentos avaliados no cenário **C2** (Subseção 5.2.1.1). Neste cenário de experimentação, os tempos de resposta médios se sobrepõem. Isto dificulta a identificação visual de um comportamento característico por sensor como no cenário CCSC. No entanto, nota-se que os sensores de pressão (Figura 5.10(e)) e giroscópio (Figura 5.10(b)) apresentam uma maior variabilidade nos dados, o que representa uma característica destes sensores ao realizarem o monitoramento de diferentes fenômenos. No geral, os tempos médios de resposta por sensor variam entre 27,73 ms (*Nó 5-a*) e 38 ms (*Nó 7-p*). Estes comportamentos, apesar de visualmente sobrepostos, quando observados detalhadamente demonstram características únicas entre os dispositivos e os sensores.

Em relação a identificação dos dispositivos através dos classificadores, os gráficos da Figura 5.11 apresentam os resultados alcançados. Estes resultados seguiram o cenário de avaliação **C3**. As métricas consideradas foram a acurácia, precisão, *recall* e *F-score*. Elas apontam uma grande diferença entre o desempenho dos classificadores, principalmente quando comparados com os resultados alcançados no cenário **C2**. Nesta avaliação, os classificadores KNN e *Multilayer Perceptron* atingiram os piores desempenhos ao identificar o tráfego dos dispositivos. Este efeito se explica pela forma que os classificadores atuam ao identificar os padrões entre os dados de entrada. O KNN identifica a proximidade entre os dados de entrada e o *Multilayer Perceptron* separa as entradas baseado em uma classificação linear. Como observado na Figura 5.10, mesmo em diferentes dispositivos, os valores dos dados são muito próximos, o que justifica a queda de desempenho. O classificador *Random Forest* apresentou um melhor desempenho, pois ele encontra a melhor característica para construir uma sequência de árvores de decisão, isto permite analisar as particularidades entre as entradas mais detalhadamente. No entanto, mesmo com a queda nas métricas, os classificadores apresentaram taxas de precisão superiores a 77%. Desta forma, com base no desempenho obtido por meio dos classificadores, comprova-se a possibilidade de identificar os dispositivos IoT através dos vazamentos temporais, mesmo com dispositivos mais robustos e sistemas operacionais mais modernos que os dispositivos encontrados no cenário experimental CCSC.

A Tabela 5.3 apresenta uma parte da matriz de confusão do classificador KNN ao realizar as avaliações conforme o cenário **C3**. Nesta matriz é possível observar que quando uma entrada é classificada erroneamente, ela é atribuída a uma classe que representa um sensor igual, porém de um dispositivo diferente (exemplo destes resultados estão destacados em negrito na tabela). Este padrão se manteve na avaliação do classificador *Multilayer Perceptron*. Isto demonstra que o tempo que um sensor leva para capturar os dados do ambiente pode revelar informações sobre qual dado está sendo monitorado. Observando isso, as avaliações a seguir procuram confirmar a hipótese de que existe uma relação entre o tempo de resposta e os tipos de sensores embarcados em cada dispositivo. Desta forma, estes dados motivaram as avaliações referentes ao cenário **C4** avaliado a seguir.



(a) Acelerômetro

(b) Giroscópio

(c) Sensor de Iluminação

(d) Magnetômetro

(e) Sensor de Pressão

Figura 5.10: Comportamento do Tempo de Resposta dos Dispositivos por Sensor

O cenário de avaliação **C4** compreende avaliar as características de sensores idênticos embarcados em dispositivos diferentes. A Figura 5.12 mostra os resultados referentes ao desempenho dos classificadores ao identificarem os padrões nos tempos de resposta entre os sensores. Nesta avaliação, os classificadores apresentaram uma determinada eficiência ao identificar os sensores, onde o classificador *Random Forest* se destacou com 72,2% de precisão. O KNN e o *Multilayer Perceptron* apresentaram 66,1% e 63,7%, respectivamente. Esta avaliação confirma a hipótese de que existe uma relação entre os tempos de resposta e os tipos sensores embarcados em cada dispositivo. Este fato agrava os impactos contra a privacidade de seus usuários, pois a possibilidade de identificar qual o sensor embarcado no dispositivo permite o atacante inferir informações sobre o que está sendo monitorado pelos sensores.

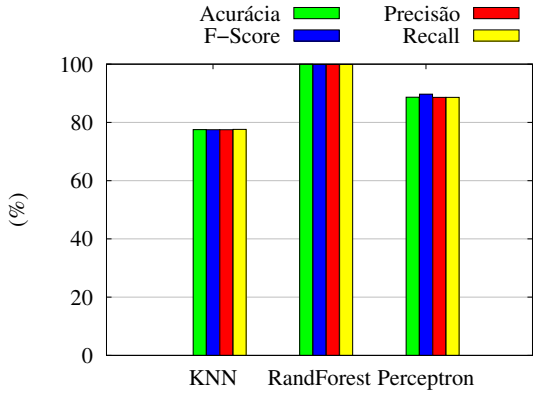


Figura 5.11: Desempenho dos Classificadores - **C3**

1-a	1-g	**1-l**	1-m	1-p	2-a	2-g	**2-l**	2-m	2-p	
261	0	**0**	0	0	39	0	**0**	0	0	Nó 1-a
0	248	**0**	0	0	0	48	**0**	0	0	Nó 1-g
0	0	**232**	0	0	0	0	**44**	**0**	**0**	**Nó 1-l**
0	0	**0**	188	0	0	0	0	56	0	Nó 1-m
0	0	**0**	0	269	0	0	0	0	47	Nó 1-p
39	0	**0**	0	0	239	0	0	0	0	Nó 2-a
0	35	**0**	0	0	0	210	0	0	0	Nó 2-g
0	0	**37**	**0**	**0**	**0**	**0**	**270**	**0**	**0**	**Nó 2-l**
0	0	0	28	0	0	0	0	264	0	Nó 2-m
0	0	0	0	48	0	0	0	0	245	Nó 2-p

Tabela 5.3: Matriz de Confusão Referente ao Classificador KNN - **C3**

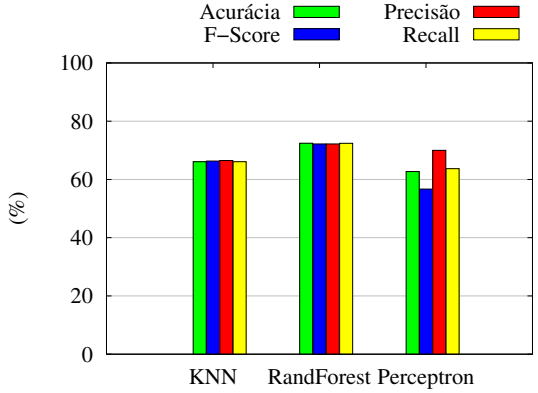


Figura 5.12: Desempenho dos Classificadores - **C4**

As matrizes de confusão apresentadas nas Tabelas 5.4, 5.5 e 5.6 revelam características semelhantes entre os tempos de resposta por sensor. Ao observar as matrizes de confusão nota-se a relação entre os sensores acelerômetro, giroscópio e magnetômetro. Esta relação pode ser explicada pela forma em que os componentes são programados para a realização da coleta. Desta forma, naturalmente entende-se que os sensores acelerômetro e giroscópio atuam de forma semelhante. Entretanto, a relação mais aparente está entre o acelerômetro e o magnetômetro. Na Tabela 5.4, o classificador *Random Forest* sugeriu que 94 entradas referentes ao acelerômetro foram classificadas erroneamente como magnetômetro e 103 entradas referentes ao magnetômetro

foram classificadas como acelerômetro, este padrão se repete nas Tabelas 5.5 e 5.6. Esta relação é explicada pelo fato de que os dados são coletados pelo mesmo sensor, logo, mesmo que a coleta seja programada de uma forma diferente, o componente responsável pelo sensoriamento apresenta vazamentos *side-channel* temporais identificáveis através do tempo de resposta.

a	g	i	m	p	
177	8	0	94	0	Acelerômetro (a)
29	130	0	55	0	Giroscópio (g)
0	0	251	0	0	Iluminação (i)
103	55	0	100	0	Magnetômetro (m)
0	0	0	0	246	Pressão (p)

Tabela 5.4: Matriz de Confusão *Random Forest* - **C4**

a	g	i	m	p	
159	26	0	94	0	Acelerômetro (a)
49	107	0	58	0	Giroscópio (g)
0	0	251	0	0	Iluminação (i)
126	47	0	85	0	Magnetômetro (m)
0	0	0	0	246	Pressão (p)

Tabela 5.5: Matriz de Confusão KNN - **C4**

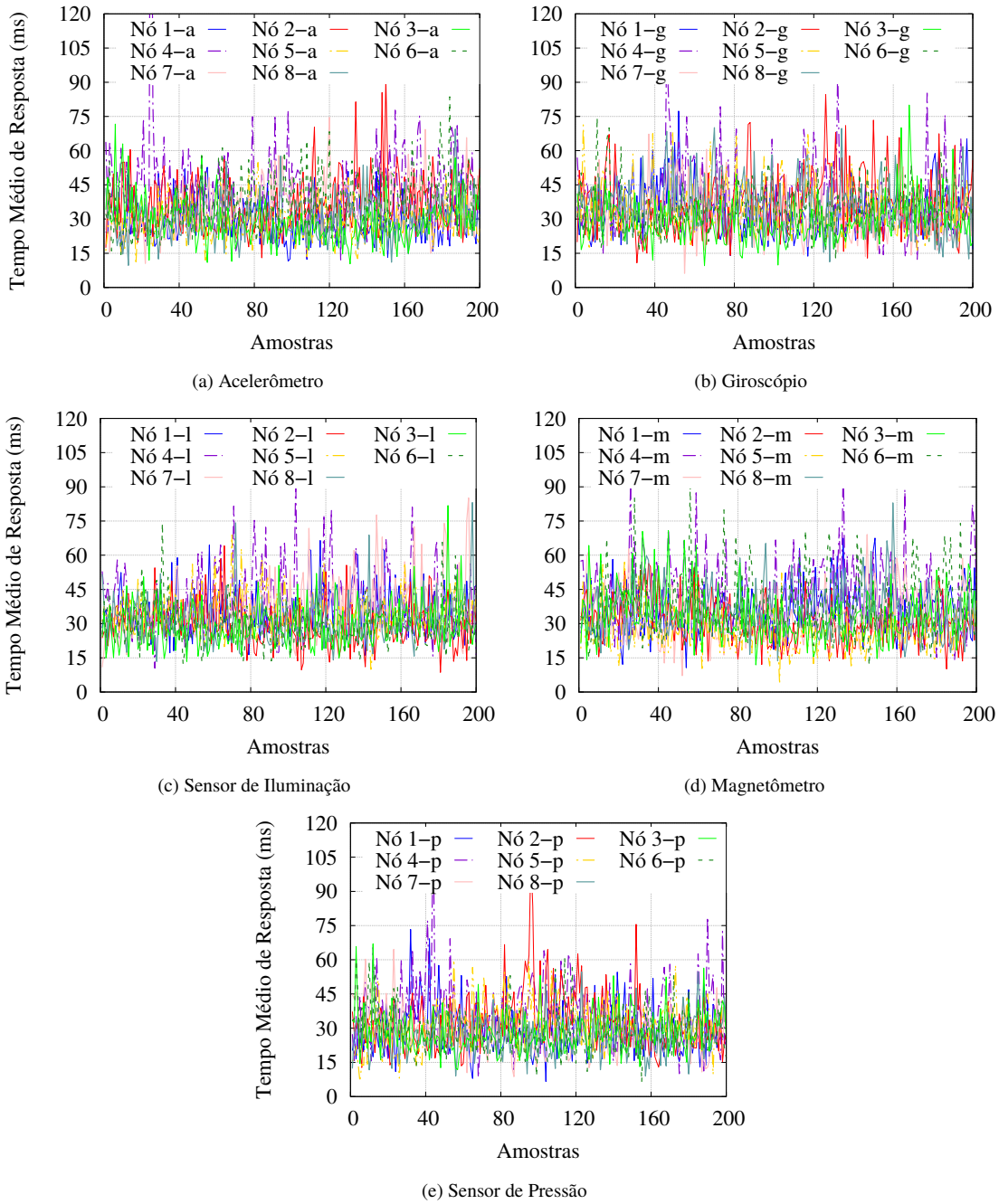
a	g	i	m	p	
265	8	0	6	0	Acelerômetro (a)
192	19	0	3	0	Giroscópio (g)
0	0	251	0	0	Iluminação (i)
235	9	0	14	0	Magnetômetro (m)
0	0	0	0	246	Pressão (p)

Tabela 5.6: Matriz de Confusão *Multilayer Perceptron*- **C4**

Esta seção apresenta a avaliação e os resultados atingidos no cenário IoTLab, com dispositivos, sistemas operacionais e protocolos mais atualizados sob proteção do mecanismo proposto. Assim, apresentamos os resultados conforme uma versão do mecanismo que implementa as abordagens **A1** e **A4** apresentadas na introdução desta seção. A abordagem **A1** se mostrou a mais eficiente no cenário de experimentação CCSC, pois apresentou o melhor desempenho ao enganar os algoritmos classificadores e a menor sobrecarga nos tempos de resposta médios. A abordagem **A4** implementa uma técnica que mistura os métodos de pacotes falsos e inserção de atrasos. Esta abordagem procura mascarar os padrões encontrados entre os instantes de tempo com os dispositivos e os tempos de resposta com os sensores embarcados, conforme apresentados nas análises implementadas na Seção 5.2. Estas abordagens são implementadas pelo módulo de proteção de privacidade. As avaliações conduzidas compreenderam a captura de 2000 requisições por sensor sem a proteção do mecanismo e 2000 requisições sob a proteção do mecanismo. O módulo de proteção de privacidade extraiu as amostras de características estatísticas a cada 10 requisições, totalizando 200 amostras por sensor.

As Figuras 5.13 e 5.14 apresentam os resultados dos dispositivos sob a proteção da abordagem **A1**. Os gráficos na Figura 5.13 apresentam os resultados referentes a caracterização

dos comportamentos dos dispositivos e seus respectivos sensores. Os comportamentos observados neste cenário apresentam uma maior variabilidade entre os tempos de resposta médios por amostra, quando comparados com as caracterizações sem a proteção do mecanismo. Isto representa que a geração de pacotes falsos implica no tempo de resposta dos dispositivos. No entanto, o tempo de resposta médio se manteve por volta dos 30 ms com e sem a proteção do mecanismo. A Figura 5.14 mostra os desempenhos dos classificadores *Random Forest* e *Multilayer Perceptron* ao procurarem identificar os padrões no tráfego protegido pelo mecanismo. Este gráfico comprova a eficiência da abordagem **A4** ao mascarar os instantes de envio, reduzindo a precisão de classificação do *Random Forest* em até 41% e do *Multilayer Perceptron* 46%. No entanto, o padrão, entre os sensores iguais em dispositivos diferentes, encontrado na matriz de confusão se manteve. Revelando que ainda existem características únicas entre os sensores.



(a) Acelerômetro

(b) Giroscópio

(c) Sensor de Iluminação

(d) Magnetômetro

(e) Sensor de Pressão

Figura 5.13: Comportamento do Tempo de Resposta dos Dispositivos por Sensor - Abordagem **A1**

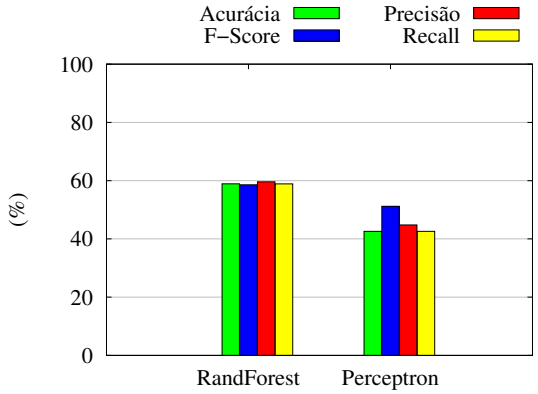


Figura 5.14: Desempenho dos Classificadores - Abordagem A1

A abordagem **A4** elimina padrões entre os tempos de resposta e os sensores embarcados, então, ela implementa a geração de pacotes falsos e a inserção de atrasos aleatórios. As Figuras 5.15 e 5.16(a) demonstram as implicações no comportamento dos dispositivos e as avaliações dos classificadores ao receberem como entrada o tráfego protegido pelo mecanismo. Os gráficos na Figura 5.15 apontam uma diferença razoável nos tempos médios de resposta por amostra, quando comparado com a avaliação acima. No entanto, o tempo médio de resposta geral aumentou em torno de 7ms. Na Figura 5.16(a) os classificadores apontaram uma melhora de até 3% na precisão, quando comparado com as avaliações dos classificadores sobre o tráfego protegido pela abordagem **A3**. Apesar destes resultados, o gráfico na Figura 5.16(b) e as matrizes de confusão nas Tabelas 5.7 e 5.8 demonstram os resultados quando os dados são avaliados por sensor. A Figura 5.16(b) mostra que os classificadores apresentam uma precisão de 38.8% no classificador *Random Forest* e 27.3% no classificador *Multilayer Perceptron*. Além disso, as matrizes de confusão nas Tabelas 5.7 e 5.8 apontam a indistinguibilidade entre as características extraídas dos tempos de resposta por sensor. Estes fatos comprovam a eficiência da abordagem **A4** ao mascarar os vazamentos temporais nas duas perspectivas exploradas pela Subseção 5.2.2.

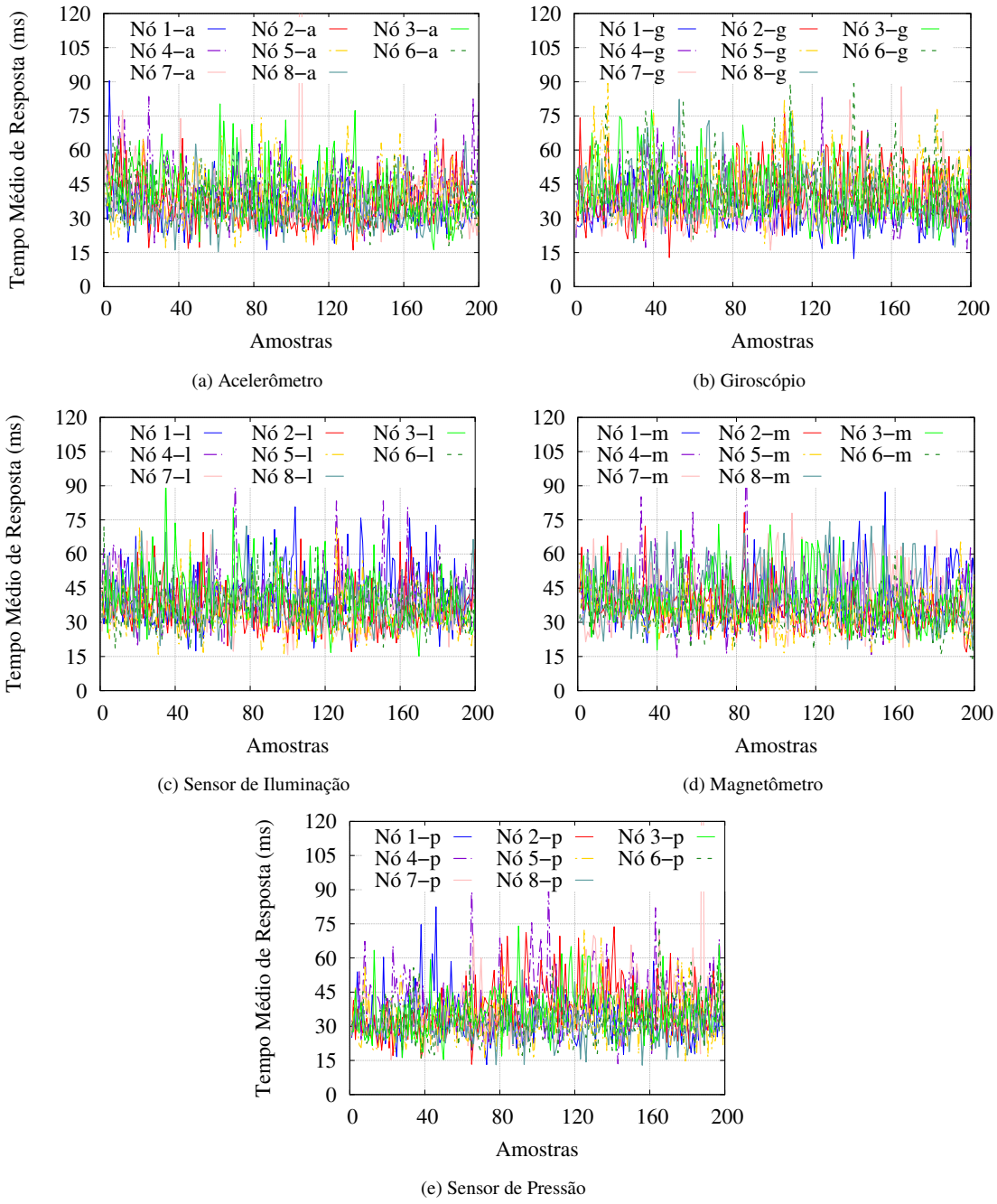
a	g	i	m	p	
137	122	116	133	15	Acelerômetro (a)
145	130	137	115	4	Giroscópio (g)
110	107	154	100	11	Iluminação (i)
129	109	123	119	12	Magnetômetro (m)
29	24	19	24	408	Pressão (p)

Tabela 5.7: Matriz de Confusão do Classificador *Random Forest*- Abordagem **A4**

a	g	i	m	p	
0	9	0	514	0	Acelerômetro (a)
0	15	0	516	0	Giroscópio (g)
0	6	0	476	0	Iluminação (i)
0	8	0	484	0	Magnetômetro (m)
0	8	0	496	0	Pressão (p)

Tabela 5.8: Matriz de Confusão do Classificador *Multilayer Perceptron*- Abordagem **A4**

Esta seção analisou o vazamento temporal *side-channel* no contexto de redes IoT e o mecanismo de defesa contra ataques *traffic side-channel*. O qual apontou os efeitos dos



(a) Acelerômetro

(b) Giroscópio

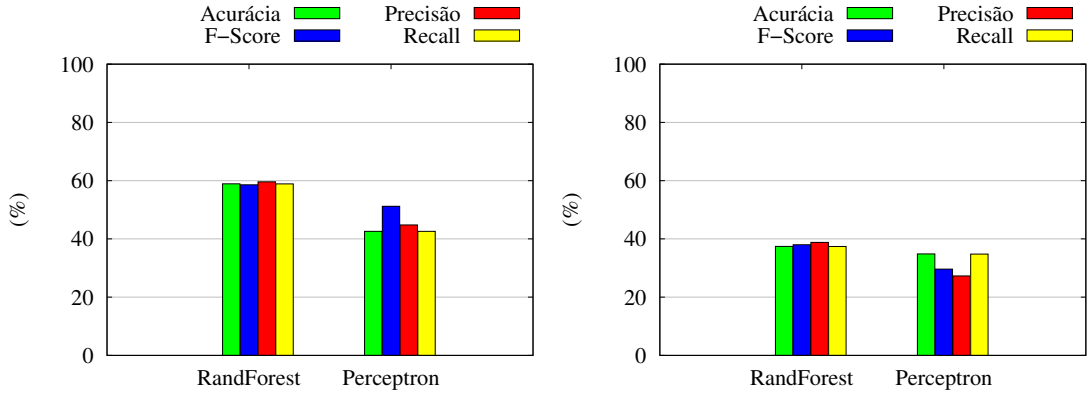
(c) Sensor de Iluminação

(d) Magnetômetro

(e) Sensor de Pressão

Figura 5.15: Comportamento do Tempo de Resposta dos Dispositivos por Sensor - Abordagem **A4**

ataques que exploram este vazamento e a melhora na privacidade dos usuários relacionadas aos dispositivos das redes IoT. Esta análise considerou dois cenários experimentais (CCSC e IoTLab) e observou o comportamento de dispositivos idênticos em questões de modelo, marca, protocolos e programas executados. Além disso, as análises consideraram apenas informações relacionadas ao instante de envio e ao tempo de resposta dos dispositivos. Através de análises do comportamento do tráfego de dados de cada dispositivo, confirma-se a existência de características únicas e específicas mesmo de dispositivos idênticos. também, este estudo mediu o desempenho de cinco classificadores na identificação do tráfego de cada dispositivo. Os resultados observados nessas análises apresentaram altas taxas de acurácia e precisão, comprovando assim uma grande eficiência em distinguir os dispositivos. Portanto, respondendo a questão de pesquisa (i), os



(a) Abordagem **A4**

(b) Cenário C4 - Abordagem **A4**

Figura 5.16: Desempenho dos Classificadores

vazamentos temporais revelam informações cruciais sobre a privacidade da rede, podendo revelar inclusive informações sobre os componentes embarcados em cada dispositivo.

As avaliações do mecanismo implementam diferentes versões de uma rede de sensores executando os protocolos 6LoWPAN e CoAP, conectados à uma rede de longo alcance. Assim, o mecanismo atua nesta rede como um serviço virtual que identifica e máscara os vazamentos *side-channel*. Na avaliação de desempenho do mecanismo, realizaram-se análises sobre quatro abordagens compostas por diferentes formas de manusear as variáveis de tempo. Cada uma das abordagens foi avaliada diante dos classificadores *Random Forest* e *Multilayer Perceptron*, onde os resultados apontam que o instante de envio tem um papel importante ao classificar os dispositivos e o tempo de resposta ao classificar os sensores embarcados em cada dispositivo. A abordagem **A4** implementa os métodos de geração de pacotes falsos e a inserção de atrasos, estes métodos almejam manipular os instantes de envio e os tempos de resposta respectivamente. Desta forma, a abordagem **A4** apresentou os melhores resultados quando comparado às outras três abordagens. Este fato responde à pergunta de pesquisa (ii), pois os métodos apresentados na literatura, em outros cenários de rede, são eficientes ao mascarar os vazamentos temporais na IoT.

5.3 RESUMO

Este capítulo apresentou duas avaliações conduzidas pelas perguntas de pesquisa (i) e (ii), a primeira compreende analisar os efeitos de um ataque *traffic side-channel* temporal na IoT e a segunda avalia o mecanismo FISHER ao mascarar os vazamentos temporais. As avaliações foram conduzidas em dois cenários experimentais CCSC e IoTLab. A primeira avaliação confirma a possibilidade de identificação de dispositivos idênticos através dos vazamentos temporais. Os resultados observados nas análises apontam que é possível caracterizar e classificar dispositivos idênticos em ambos os cenários de experimentação. Além disso, no cenário IoTLab, provou-se ser possível classificar os sensores iguais embarcados em dispositivos diferentes observando apenas os vazamentos temporais nos tempos de resposta. Diante dos comportamentos observados, a segunda avaliação compreendeu uma implementação parcial do mecanismo, a fim de avaliar a eficiência das quatro abordagens que visam mascarar os vazamentos temporais das análises realizadas pelos ataques *traffic side-cannel*. Assim, o mecanismo segue dois módulos de teste de vulnerabilidade e de proteção de privacidade. O modulo de teste de vulnerabilidade implementa o ataque avaliado anteriormente. O módulo de proteção de privacidade implementa as três abordagens visando reduzir a precisão do ataque implementado pelo módulo anterior. Os resultados mostram que a abordagem **A1** é eficiente, reduzindo a precisão em até 63% no cenário

CCSC, **A3** é marginalmente eficiente, **A2** é ineficaz e a abordagem **A4** reduzindo a precisão em até 46% no cenário IoILab. Portanto, este capítulo demonstrou que o mecanismo FISHER é eficiente ao mascarar os vazamentos temporais *side-channel* e responde as duas perguntas de pesquisa, confirmando que existem de vazamentos que representam um risco à privacidade dos usuários e que os métodos de pacotes falsos e inserção de atrasos mascaram estes vazamentos.

6 CONCLUSÃO

Este trabalho apresentou o mecanismo FISHER de defesa contra ataques *traffic side-channel* temporais no contexto da Internet das Coisas. Para isso, este trabalho realiza um estudo que avaliou empiricamente os efeitos destes ataques nas redes IoT. Este estudo foi guiado por duas perguntas de pesquisa: (i) Qual o impacto das análises realizadas sobre os vazamentos *side-channel* na privacidade das redes IoT? (ii) Qual a eficácia ao implementar os métodos de geração de pacotes falsos e inclusão de atrasos no contexto da IoT? Estas perguntas foram respondidas através de avaliações realizadas em dois cenários experimentais (CCSC e IoTLAB). Estas avaliações compreenderam a implementação das análises realizadas pelos ataques *traffic side-channel* temporais e do mecanismo FISHER. A implementação das análises seguiu três fases fundamentais: experimentação, caracterização e identificação. A fase de experimentação compreende a captura do tráfego de rede. A caracterização seleciona e extrai as características sobre as capturas de rede, como o instante de envio e o tempo de resposta. A fase de identificação analisa o comportamento único de cada dispositivo IoT por meio de classificadores. Os resultados sobre os ataques apontam a existência de vulnerabilidades que revelam informações detalhadas sobre os dispositivos e os componentes embarcados em cada um, o que responde a questão de pesquisa (i). O mecanismo FISHER foi implementado seguindo dois módulos, teste de vulnerabilidade e proteção de privacidade. O módulo de teste de vulnerabilidade segue fases semelhantes as implementadas na identificação dos vazamentos *side-channel*. O módulo de proteção de privacidade foi desenvolvido baseado nos métodos de geração de pacotes falsos e inserção de atrasos. Os resultados referentes ao mecanismo FISHER apontam que as abordagens de geração de pacotes falsos e a inserção de atrasos melhoram a privacidade diante destes ataques, o que responde a questão de pesquisa (ii).

6.1 TRABALHOS FUTUROS

Os resultados alcançados e as conclusões acima apresentadas demonstram que este trabalho atingiu os objetivos propostos. Contudo, segundo Liu et al. (2014), as operações de rádio frequência são responsáveis por 80% do consumo de energia dos dispositivos IoT. O módulo de proteção de privacidade gera pacotes falsos, aumentando a quantidade de transmissões, logo o consumo de energia. Diante disso, como trabalhos futuros pretende-se realizar uma modelagem para mensurar o consumo de recursos e a privacidade proporcionada pelo mecanismo, a fim de equilibrá-los. Ou seja, proporcionar máxima eficiência consumindo o mínimo de recursos. Esta modelagem ajudará no estabelecimento de limiares para o módulo de proteção de privacidade (por exemplo, tempo máximo de resposta e geração de pacotes falsos). Então, para esta modelagem o risco de privacidade precisa ser quantificado. Portanto, pretende-se seguir a teoria da privacidade diferencial conforme os trabalhos encontrados na literatura (Liu et al., 2018; Yan et al., 2017). Contudo, pretende-se avaliar também os impactos gerados pelos atrasos inseridos, as taxas de perdas de pacotes e o consumo de energia do mecanismo proposto.

6.2 PUBLICAÇÕES

Esta seção apresenta a lista de publicações realizadas até então.

- Minicurso: *Ameaças de Segurança, Defesas e Análise de Dados na IoT baseada em SDN*. Nelson G. Prates Jr., Mateus Pelloso, Ricardo T. Macedo, e Michele Nogueira.

Em **Minicursos do XVIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)**. Ano 2018.

- *Análise de vazamentos temporais side-channel no contexto da internet das coisas*. Nelson G. Prates Jr., Andressa Vergütz, Ricardo T. Macedo, e Michele Nogueira. Em **Anais do XXIV Workshop de Gerência e Operação de Redes e Serviços (WGRS) no XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)**. Ano 2019.

- *Um Mecanismo de Defesa Contra Ataques Traffic Side-Channel Temporais na IoT*. Nelson G. Prates Jr., Andressa Vergütz, Ricardo T. Macedo, e Michele Nogueira. Em **XIX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais**. Ano 2019.

- *A Defense Mechanism for Timing-based Side-Channel Attacks on IoT Traffic*. Nelson G. Prates Jr., Andressa Vergütz, Ricardo T. Macedo, Aldri Santos e Michele Nogueira. Em **IEEE Global Communications Conference**. Ano 2020.

- *A Method Aware of Concept Drift for Online Botnet Detection*. Bruno Henrique Schwengber, Andressa Vergütz, Nelson G. Prates Jr., e Michele Nogueira. Em **IEEE Global Communications Conference**. Ano 2020.

- *An Architecture for the Performance Management of Smart Healthcare Applications*. Andressa Vergutz, Nelson G. Prates Jr., Bruno Henrrique Schwengber, Aldri dos Santos, e Michele Nogueira. Em **MDPI - Sensors**. Ano 2020.

REFERÊNCIAS

802.15.4, I. S. (2016). Ieee standard for low-rate wireless networks. *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, páginas 1–709.

Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noel, T., Pissard-Gibollet, R., Saint-Marcel, F., Schreiner, G., Vandaele, J. et al. (2015). Fit iot-lab: A large scale open experimental iot testbed. Em *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, páginas 459–464. IEEE.

Alduais, N., Abdullah, J., Jamil, A. e Audah, L. (2016). An efficient data collection and dissemination for iot based wsn. Em *IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, páginas 1–6. IEEE.

Alexander, R., Brandt, A., Vasseur, J., Hui, J., Pister, K., Thubert, P., Levis, P., Struik, R., Kelsey, R. e Winter, T. (2012). RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. Relatório Técnico 6550, IETF.

Alrawais, A., Alhothaily, A., Hu, C. e Cheng, X. (2017). Fog computing for the Internet of Things: Security and privacy issues. *IEEE Internet Computing*, 21(2):34–42.

Apthorpe, N., Huang, D. Y., Reisman, D., Narayanan, A. e Feamster, N. (2019). Keeping the smart home private with smart (er) IoT traffic shaping. *Proc. Privacy Enhancing Technol.*, 2019(3):128–148.

Apthorpe, N., Reisman, D., Sundaresan, S., Narayanan, A. e Feamster, N. (2017). Spying on the smart home: Privacy attacks and defenses on encrypted iot traffic.

Arp, D., Yamaguchi, F. e Rieck, K. (2015). Torben: A practical side-channel attack for deanonymizing tor communication. Em *10th ACM Symposium on Information, Computer and Communications Security*, páginas 597–602. ACM.

ASCOM (2019a). Decreto que institui o plano nacional de internet das coisas é publicado. `http://www.mctic.gov.br/mctic/opencms/salaImprensa/noticias/arquivos/2019/06/Decreto_que_institui_o_Plano_Nacional_de_Internet_das_Coisas_e_publicado.html`. Último Acesso: Julho de 2019.

ASCOM (2019b). Mctic, bndes e qualcomm lançam primeiro fundo de investimentos para iot. `https://www.mctic.gov.br/mctic/opencms/salaImprensa/noticias/arquivos/2019/12/MCTIC_BNDES_e_Qualcomm_lancam_primeiro_fundo_de_investimentos_para_IoT.html`. Último Acesso: Fevereiro de 2020.

Atkinson, J. S., Mitchell, J. E., Rio, M. e Matich, G. (2018). Your wifi is leaking: What do your mobile apps gossip about you? *Future Generation Computer Systems*, 80:546 – 557.

Atzori, L., Iera, A. e Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15):2787 – 2805.

Avizienis, A., Laprie, J.-C., Randell, B. e Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing*, 1(1):11–33.

Bandyopadhyay, D. e Sen, J. (2011). Internet of Things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58(1):49–69.

Banks, A. e Gupta, R. (2014). MQTT version 3.1. 1. Relatório técnico, OASIS standard.

Bates, A., Mood, B., Pletcher, J., Pruse, H., Valafar, M. e Butler, K. (2012). Detecting co-residency with active traffic analysis techniques. Em *ACM Workshop on Cloud computing security workshop*, páginas 1–12. ACM.

Botta, A., De Donato, W., Persico, V. e Pescapé, A. (2016). Integration of cloud computing and Internet of Things: a survey. *Future Generation Computer Systems*, 56:684–700.

Cervantes, C., Nogueira, M. e Santos, A. (2014). Um sistema de detecção de ataques sinkhole sobre 6LoWPAN para internet das coisas. Dissertação de Mestrado, Universidade Federal do Paraná.

Chaddad, L., Chehab, A., Elhajj, I. H. e Kayssi, A. (2018). App traffic mutation: Toward defending against mobile statistical traffic analysis. Em *INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, páginas 27–32. IEEE.

Chen, F., Duan, H., Zheng, X., Jiang, J. e Chen, J. (2018a). Path leaks of https side-channel by cookie injection. Em *International Workshop on Constructive Side-Channel Analysis and Secure Design*, páginas 189–203. Springer.

Chen, J., Tian, Z., Cui, X., Yin, L. e Wang, X. (2018b). Trust architecture and reputation evaluation for internet of things. *Journal of Ambient Intelligence and Humanized Computing*, páginas 1–9.

Comissão Européia (2018). Reforma de 2018 das regras de proteção de dados da UE. `https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_pt`. Online; acessado dia 12 de Junho 2019.

Conti, M., Li, Q. Q., Maragno, A. e Spolaor, R. (2018). The dark side (-channel) of mobile devices: A survey on network traffic analysis. *IEEE Communications Surveys & Tutorials.*, 20(4):2658–2713.

Copos, B., Levitt, K., Bishop, M. e Rowe, J. (2016). Is anybody home? inferring activity from smart home network traffic. Em *2016 IEEE Security and Privacy Workshops (SPW)*, páginas 245–251. IEEE.

Feghhi, S. e Leith, D. J. (2016). A web traffic analysis attack using only timing information. *IEEE Transactions on Information Forensics and Security*, 11(8):1747–1759.

Feghhi, S. e Leith, D. J. (2019). An efficient web traffic defence against timing-analysis attacks. *IEEE Transactions on Information Forensics and Security*, 14(2):525–540.

Ferraz Júnior, T. S. (1993). Sigilo de dados: o direito à privacidade e os limites à função fiscalizadora do estado. *Revista da Faculdade de Direito, Universidade de São Paulo*, 88:439–459.

Grupo Globo (2013). Entenda o caso de Edward Snowden, que revelou espionagem dos EUA. http://g1.globo.com/mundo/noticia/2013/07/entenda-o-caso-de-edward-snowden-que-revelou-espionagem-dos-eua.html. Online; acessado dia 12 de Junho 2019.

Gu, J., Wang, J., Yu, Z. e Shen, K. (2019). Traffic-based side-channel attack in video streaming. *IEEE/ACM Transactions on Networking*, 27(3):972–985.

He, J., Xiao, Q., He, P. e Pathan, M. S. (2017). An adaptive privacy protection method for smart home environments using supervised learning. *Future Internet*, 9(1):7.

He, J., Xiao, Q. e Pathan, M. S. (2016). A method for countering snooping-based side channel attacks in smart home applications. Em *International Conference on Commun. and Netw.*, páginas 200–207. Springer.

Iannacci, J. (2018). Internet of things (iot); internet of everything (ioe); tactile internet; 5g – a (not so evanescent) unifying vision empowered by eh-mems (energy harvesting mems) and rf-mems (radio frequency mems). *Sensors and Actuators A: Physical*, 272:187 – 198.

Kausar, F., Aljumah, S., Alzaydi, S. e Alroba, R. (2019). Traffic analysis attack for identifying users' online activities. *IT Professional*, 21(2):50–57.

Khan, R., Khan, S. U., Zaheer, R. e Khan, S. (2012). Future Internet: the Internet of Things architecture, possible applications and key challenges. Em *Frontiers of Information Technology*, páginas 257–260. IEEE.

Lamaazi, H., Benamar, N., Jara, A. J., Ladid, L. e El Ouadghiri, D. (2014). Challenges of the internet of things: Ipv6 and network management. Em *Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, páginas 328–333.

Li, X., Yang, C., Ma, J., Liu, Y. e Yin, S. (2017). Energy-efficient side-channel attack countermeasure with awareness and hybrid configuration based on it. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(12):3355–3368.

Lima, M. N., Dos Santos, A. L. e Pujolle, G. (2009). A survey of survivability in mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 11(1):66–77.

Lin, H. e Bergmann, N. (2016). IoT privacy and security challenges for smart home environments. *Information*, 7(3):44.

Liu, J., Zhang, C. e Fang, Y. (2018). Epic: A differential privacy framework to defend smart homes against internet traffic analysis. *IEEE Internet of Things Journal*, 5(2):1206–1217.

Liu, L., Men, S., Liu, M. e Zhou, B. (2014). An energy saving solution for wireless communication equipment. Em *IEEE 36th International Telecommunications Energy Conference (INTELEC)*, páginas 1–3. IEEE.

Malik, N., Chandramouli, J., Suresh, P., Fairbanks, K., Watkins, L. e Robinson, W. H. (2017). Using network traffic to verify mobile device forensic artifacts. Em *2017 14th IEEE Annual Consum. Commun. & Netw. Conference (CCNC)*, páginas 114–119. IEEE.

Marília Marques, G. D. (2018). Mp do df aponta suposto esquema de venda de dados pessoais de brasileiros pelo serpro. https://g1.globo.com/df/distrito-federal/noticia/mp-do-df-aponta-suposto-esquema-de-venda-de-dados-pessoais-de-brasileiros-pelo-serpro.ghtml. Último Acesso: Julho de 2019.

Miao Wu, Ting-Jie Lu, Fei-Yang Ling, Jing Sun e Hui-Ying Du (2010). Research on the architecture of internet of things. Em *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, volume 5, páginas V5–484–V5–487.

Ministério de Relações Exteriores (2013). ONU aprova resolução sobre o Direito à Privacidade na Era Digital. http://www.itamaraty.gov.br/pt-BR/notas-a-imprensa/3436-resolucao-sobre-o-direito-a-privacidade-na-era-digital. Online; acessado dia 12 de Junho 2019.

Miraz, M. H., Ali, M., Excell, P. S. e Picking, R. (2015). A review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT). Em *Internet Technologies and Applications (ITA)*, páginas 219–224.

Montenegro, G., Kushalnagar, N., Hui, J. e Culler, D. (2007a). Transmission of ipv6 packets over ieee 802.15. 4 networks. Relatório técnico, IETF.

Montenegro, G., Schumacher, C. e Kushalnagar, N. (2007b). IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. Relatório Técnico 4919, IETF.

M.R., S. F.-X. V. I. (2010). *Introduction to Side-Channel Attacks*, páginas 27–42. Springer US, Boston, MA.

Márcio Padrão, Do UOL, e. S. P. (2018). Dados pessoais de 2,4 milhões de usuários do sus são vazados na internet. https://noticias.uol.com.br/tecnologia/noticias/redacao/2019/04/11/dados-pessoais-de-24-milhoes-de-usuarios-do-sus-sao-vazados-na-internet.html. Último Acesso: Julho de 2019.

Pacheco, F., Exposito, E., Gineste, M., Baudoin, C. e Aguilar, J. (2018). Towards the deployment of machine learning solutions in network traffic classification: A systematic survey. *IEEE Communications Surveys & Tutorials*.

Park, S. D., Kim, K.-H., Haddad, W., Chakrabarti, S. e Laganier, J. (2011). IPv6 over Low Power WPAN Security Analysis. Internet-Draft draft-daniel-6lowpan-security-analysis-05, Internet Engineering Task Force. Work in Progress.

Patranabis, S., Roy, D. B., Chakraborty, A., Nagar, N., Singh, A., Mukhopadhyay, D. e Ghosh, S. (2018). Lightweight design-for-security strategies for combined countermeasures against side channel and fault analysis in iot applications. *Journal of Hardware and Systems Security*, páginas 1–29.

Ponemon, I. (2018). Estudo ibm: Gastos com violações de dados caem no brasil, mas país é o mais provável a ter ataques de hackers entre os pesquisados. https://www.ibm.com/blogs/ibm-comunica/estudo-ibm-gastos-com-violacoes-de-dados-caem-no-brasil/. Último Acesso: Julho de 2019.

Postel, J. (1980). User Datagram Protocol. RFC 768.

Postel, J. (1981). Transmission Control Protocol. RFC 793.

Prates, N., Pelloso, M., Macedo, R. e Nogueira, M. (2018). Ameaças de segurança, defesas e análise de dados em iot baseada em sdn. Em *Minicursos SBSeg 2018*, capítulo: 1, páginas 1–50. SBC.

Prates, N., Vergûtz, A., Macedo, R. e Lima, M. N. (2019a). Análise de vazamentos temporais side-channel no contexto da internet das coisas. *Anais do XXIV Workshop de Gerência e Operação de Redes e Serviços (WGRS) no XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, 24(3):157–170.

Prates, N., Vergûtz, A., Macedo, R. e Lima, M. N. (2019b). Um mecanismo de defesa contra ataques traffic side-channel temporais na iot (a ser publicado). *XIX Simpósio Brasileiro de Seguranda da Informação e de Sistemas Computacionais*.

Saltaformaggio, B., Choi, H., Johnson, K., Kwon, Y., Zhang, Q., Zhang, X., Xu, D. e Qian, J. (2016). Eavesdropping on fine-grained user activities within smartphone apps over encrypted network traffic. Em *10th USENIX Workshop on Offensive Technologies (WOOT 16)*, Austin, TX. USENIX Association.

Sayakkara, A., Le-Khac, N.-A. e Scanlon, M. (2019). A survey of electromagnetic side-channel attacks and discussion on their case-progressing potential for digital forensics. *Digital Investigation*, 29:43 – 54.

Schatten, M., Ševa, J. e Tomičić, I. (2016). A roadmap for scalable agent organizations in the Internet of Everything. *Journal of Systems and Software*, 115:31 – 41.

Selis, V. e Marshall, A. (2017). A fake timing attack against behavioural tests used in embedded iot m2m communications. Em *1st Cyber Security in Networking Conference (CSNet)*, páginas 1–6.

Shelby, Z., Hartke, K. e Bormann, C. (2014). The Constrained Application Protocol (CoAP). Relatório Técnico 7252, IETF.

Shmatikov, V. e Wang, M.-H. (2006). Timing analysis in low-latency mix networks: Attacks and defenses. Em *European Symposium on Research in Computer Security*, páginas 18–33. Springer.

Sivanathan, A., Gharakheili, H. H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A. e Sivaraman, V. (2018). Classifying iot devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing*.

Sonar, K. e Upadhyay, H. (2014). A survey: DDoS attack on Internet of Things. *International Journal of Engineering Research and Development*, 10(11):58–63.

Srinivasan, V., Stankovic, J. e Whitehouse, K. (2008). Protecting your daily in-home activity information from a wireless snooping attack. Em *10th international conference on Ubiquitous computing*, páginas 202–211. ACM.

Sung, K., Biswas, J., Learned-Miller, E., Levine, B. N. e Liberatore, M. (2018). Server-side traffic analysis reveals mobile location information over the internet. *IEEE Transactions on Mobile Computing*, 18(6):1407–1418.

Symatec (2018). 2018 Internet Security Threat Report. `https://www.symantec.com/security-center/threat-report`. Último Acesso: Agosto de 2018.

Taylor, V. F., Spolaor, R., Conti, M. e Martinovic, I. (2018). Robust smartphone app identification via encrypted network traffic analysis. *IEEE Transactions on Information Forensics and Security*, 13(1):63–78.

Thubert, P., Bormann, C., Toutain, L. e Cragie, R. (2017). IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header. Relatório Técnico 8138, IETF.

Vergütz, A., da Silva, R., Nacif, J. A. M., Vieira, A. B. e Nogueira, M. (2017). Mapping critical illness early signs to priority alert transmission on wireless networks. Em *IEEE Latin-American Conference on Commun.*, páginas 1–6. IEEE.

Veysset, F., Courtay, O., Heen, O., Team, I. et al. (2002). New tool and technique for remote operating system fingerprinting. *Intranode Software Technologies*, 4.

Wang, Q., Yahyavi, A., Kemme, B. e He, W. (2015). I know what you did on your smartphone: Inferring app usage over encrypted data traffic. Em *IEEE Conference on Communications and Network Security (CNS)*, páginas 433–441. IEEE.

Wang, W., Motani, M. e Srinivasan, V. (2008). Dependent link padding algorithms for low latency anonymity systems. Em *15th ACM conference on Computer and communications security*, páginas 323–332. ACM.

Wu, K. (2018). Detecting Streaming Wireless Cameras with Timing Analysis. Dissertação de Mestrado, University of Washington, Seattle, Washington, Estados Unidos.

Xiong, S., Sarwate, A. D. e Mandayam, N. B. (2018). Defending against packet-size side-channel attacks in IoT networks. Em *IEEE Acoustics, Speech and Signal Processing (ICASSP)*, páginas 2027–2031. IEEE.

Yan, B. e Huang, G. (2009). Supply chain information transmission based on rfid and internet of things. Em *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*, volume 4, páginas 166–169. IEEE.

Yan, Y., Oswald, E. e Tryfonas, T. (2017). Exploring potential 6LoWPAN traffic side channels. *IACR Cryptology ePrint Archive*, 2017:316.

Yang, Z., Yue, Y., Yang, Y., Peng, Y., Wang, X. e Liu, W. (2011). Study and application on the architecture and key technologies for iot. Em *2011 International Conference on Multimedia Technology*, páginas 747–751. IEEE.

Yu, W. e Köse, S. (2017). A lightweight masked AES implementation for securing iot against CPA attacks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64(11):2934–2944.

Zhang, F., He, W., Liu, X. e Bridges, P. G. (2011). Inferring users' online activities through traffic analysis. Em *fourth ACM conference on Wireless network security*, páginas 59–70. ACM.

Zhao, K. e Ge, L. (2013). A survey on the internet of things security. Em *2013 Ninth international conference on computational intelligence and security*, páginas 663–667. IEEE.

Zhao, M., Kumar, A., Chong, P. H. J. e Lu, R. (2017). A comprehensive study of RPL and P2P-RPL routing protocols: Implementation, challenges and opportunities. *Peer-to-Peer Networking and Applications*, 10(5):1232–1256.