Um Mecanismo de Defesa Contra Ataques Traffic Side-Channel Temporais na IoT

Nelson G. Prates Jr.[1], **Andressa Vergütz** [1], **Ricardo T. Macedo**[2], **Michele Nogueira**[1]

[1]Centro de Ciência de Segurança Computacional (CCSC) – UFPR
[2]Depto. de Tecnologia da Informação - Campus Frederico Westphalen – UFSM

{ngpjunior,michele,avergutz}@inf.ufpr.br, rmacedo@inf.ufsm.br

Abstract. *The Internet of Things (IoT) connects objects for delivering innovative services. However, the occurrence of temporal traffic-channel attacks threatens the IoT users privacy by revealing insider information about their behavior. This work presents a Temporal Traffic Side-Channel Attack Defense Mechanism for IoT. The mechanism follows two modules, vulnerability testing and privacy protection. The vulnerability testing module identifies temporal side-channel leakages and initiates the defense process. The privacy protection module implements three approaches that mask the behavior of networked devices to hide time leakages. The results of the performance evaluation conducted in an experimental scenario show that the best approach reduces device identification accuracy by up to 63 %.*

Resumo. *A Internet das Coisas (IoT) conecta objetos à Internet para prestar serviços inovadores. Entretanto, a ocorrência de ataques traffic side-channel temporais ameaçam ferir o princípio de privacidade dos usuários IoT ao revelar informações privilegiadas sobre o seu comportamento. Este trabalho apresenta um Mecanismo de Defesa Contra Ataques Traffic Side-Channel Temporais na IoT. O mecanismo segue dois módulos, o de teste de vulnerabilidade e o de proteção de privacidade. O módulo de teste de vulnerabilidade identifica os vazamentos temporais side-channel e inicia o processo de defesa, diferente dos trabalhos prévios que apenas identificam as vulnerabilidades. O módulo de proteção de privacidade implementa três abordagens para mascarar o comportamento dos dispositivos em rede e ocultar os vazamentos temporais, diferentemente dos trabalhos da literatura focam em outros vazamentos como eletromagnetismo ou consumo de energia. Os resultados da avaliação de desemprenho conduzida em um cenário experimental mostram que a melhor abordagem reduz a acurácia de identificação dos dispositivos em até 63%.*

1. Introdução

O rápido avanço das tecnologias de comunicação sem fio tem possibilitado a conexão de objetos à Internet, contribuindo para o surgimento da IoT (*Internet of Things*) [Al-Fuqaha et al. 2015, Cervantes et al. 2015]. Tais objetos (coisas) são embarcados com sensores a fim de proporcionar serviços inteligentes através da troca de uma grande quantidade de dados em tempo real [Prates et al. 2018]. Como exemplo didático, a IoT oferece serviços de socorro, que através de redes de comunicação seguras, realizam o monitoramento da saúde de pessoas em suas atividades do cotidiano, informam transeuntes próximos

sobre sua situação emergencial, ou ainda predizem se há a iminência de um ataque cardíaco [Vergütz et al. 2017]. A padronização de protocolos para a IoT, como o *Constrained Application Protocol* (CoAP) e *IPv6 over Low power Wireless Personal Area Networks* (6LoWPAN), motivaram o desenvolvimento de novos serviços, disponibilizando assim novos recursos e abrindo espaço para a criação de cidades mais inteligentes. Além disso, a IoT se torna atraente para os negócios devido ao volume de dados gerados que permite a melhor compreensão do comportamento de consumo, interesses pessoais, entre outros.

Todavia, devido ao volume e tipo de informação trafegados pela IoT, ela se torna alvo dos ataques *traffic side-channel* temporais. Este tipo de ataque se aproveita de todo ou qualquer dado gerado e transmitido, em especial, os dados referentes ao instante de transmissão e o tempo de resposta (vazamentos temporais *side-channel*), para inferir informações que ferem a privacidade dos usuários. Assim, os atacantes mediantes *sniffers* capturam o tráfego de rede e sondam os vazamentos temporais *side-channel* [Yan et al. 2017]. Os vazamentos temporais compreendem o tempo de execução de determinadas operações computacionais dos dispositivos, como o tempo de resposta, que é a diferença do instante de envio de uma requisição com o do recebimento da resposta. Com base nestes vazamentos, realizam-se análises estatísticas para encontrar padrões de comportamento entre os dispositivos. Estas informações, quando cruzadas com padrões de comportamento humano, ferem o direito de privacidade dos usuários. Por exemplo, em [Srinivasan et al. 2008] os autores inferiram informações sobre os usuários por meio de análises de tráfego em um cenário de casa inteligente (*smart-home*). Essas informações são adquiridas através da similaridade dos *intervalos de tempo* (vazamentos temporais) entre as transmissões realizadas pelos dispositivos IoT com o comportamento dos moradores. Isso possibilita a identificação de cômodos, quantidade de residentes e até mesmo possíveis visitantes. A fácil viabilidade desse ataque mostra a necessidade de defesas para preservar o direito de privacidade dos usuários da IoT.

Os trabalhos relacionados identificam e apresentam defesas contra a variedade de ataques *side-channel*. [Yan et al. 2017] identificaram que informações como o tamanho dos pacotes e *tempos de respostas* apresentam determinada fragilidade nos protocolos desenvolvidos para IoT. Além disso, os autores propuseram um conjunto de recomendações sobre a ocultação dos vazamentos *side-channel*, porém eles não apresentaram avaliações para as recomendações de defesa. [Selis and Marshall 2017] utilizaram um método estatístico eficiente de identificar características únicas entre os dispositivos através de variáveis relacionadas ao tempo. No entanto, apesar de apresentarem um método eficiente ao considerar variáveis temporais, ele aborda outro tipo de ataque. [Prates et al. 2019] identificaram e apontaram a gravidade dos vazamentos temporais, onde classificaram dispositivos IoT idênticos (com o mesmo *hardware*, executando os mesmos protocolos e aplicações). No geral, a maioria das defesas consideram mascarar ou ocultar os vazamentos relacionados ao consumo de energia ou emanação de eletromagnetismo [Li et al. 2017, Yu and Köse 2017]. No melhor do nosso conhecimento, apenas [Xiong et al. 2018] apresentaram um método de defesa para os ataques *traffic side-channel*, que oculta o tamanho dos pacotes por meio da teoria da privacidade local diferencial. Esta teoria esconde as informações de um indivíduo através de análises estatísticas realizadas sobre os dados de um grupo de indivíduos. Entretanto, esses trabalhos não abordam os ataques *traffic side-channel* temporais considerando os protocolos padronizados para dispositivos IoT.

Este trabalho apresenta um Mecanismo de Defesa Contra Ataques *Traffic Side-Channel* Temporais no contexto da IoT. O mecanismo evita a ocorrência destes ataques através da identificação e ocultação dos vazamentos temporais *side-channel*, melhorando a privacidade dos usuários e dos dispositivos IoT. 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 o tráfego em amostras para a caracterização e identificação dos vazamentos temporais [Prates et al. 2019]. O módulo de proteção de privacidade define regras individuais para os dispositivos a fim de mascarar o comportamento e ocultar os vazamentos temporais das análises estatísticas realizadas pelos ataques *traffic side-channel*. Este módulo manipula as variáveis relacionadas ao tempo por meio de três abordagens. A primeira abordagem (**A1**) controla os instantes de envio através da duplicação das requisições. A segunda (**A2**) insere os atrasos de forma aleatória considerando o ciclo da fila de processos do sistema operacional. A terceira (**A3**) insere operações de atraso na função de envio das mensagens aproximando os tempos médio de resposta dos dispositivos. Dessa forma, o mecanismo oculta os vazamentos temporais, e assim dificulta a aquisição dos dados relacionados a informações cruciais sobre a privacidade dos usuários como nos ataques *traffic side-channel* temporais.

A avaliação de desempenho do mecanismo compara a eficiência de três abordagens executadas pelo módulo de proteção de privacidade, ao ocultar os vazamentos temporais *side-channel*. Avaliamos estas abordagens em um cenário experimental de rede composto por dispositivos IoT Memsic Iris. Esta rede executa os principais protocolos padronizados para a IoT e simula uma aplicação de monitoramento de luminosidade. Para cada abordagem, o módulo de teste de vulnerabilidade gera uma carga de trabalho com requisições CoAP através do *framework* Californium[1] e coleta o tráfego na ferramenta Wireshark [2]. Em seguida, realiza operações de extração de características estatísticas e amostragem dos dados na ferramenta R[3]. Os algoritmos de classificação empregados, *Random Forest* e *Multilayer Perceptron* (implementados pelo *software* WEKA[4]), classificam o tráfego a partir das amostras geradas. O módulo de teste de vulnerabilidade foi avaliado previamente, alcançando 100% de acurácia e precisão na identificação dos dispositivos pelos vazamentos temporais *side-channel* [Prates et al. 2019]. Com isso, neste trabalho consideramos eficiente a abordagem que consegue ocultar os vazamentos de forma que reduza as métricas supracitadas. Os resultados mostram que a abordagem **A1** é eficiente, reduzindo as métricas em até 63%, **A3** é marginalmente eficiente e **A2** é ineficaz, pois os classificadores não apresentaram dificuldade ao identificar os dispositivos.

O restante do artigo está organizado como segue. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 detalha o mecanismo proposto. A Seção 4 descreve a metodologia de avaliação. A Seção 5 apresenta e discute os resultados obtidos. Por fim, a Seção 6 conclui o trabalho e apresenta as direções futuras.

[1]Californium (Cf) https://www.eclipse.org/californium/. Último acesso em Jun/2019

[2]Wireshark, https://www.wireshark.org/. Último acesso em Jun /2019.

[3]R, https://www.r-project.org/. Último acesso em Jun/2019.

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

2. Trabalhos Relacionados

Na literatura existem trabalhos que identificam ou apresentam defesas dos vazamentos que levam à variedade de ataques *side-channel*. No entanto, poucos estudos consideram os vazamentos *side-channel* presentes nas redes compostas pelos dispositivos e protocolos padronizados para a IoT [Yan et al. 2017, Prates et al. 2019]. [Chen et al. 2010, Srinivasan et al. 2008] apontaram a possibilidade de revelar informações confidenciais detalhadas dos usuários através de dados vazados (ex., tamanho do pacote e tempo de resposta) de aplicativos e/ou protocolos de redes, intitulados vazamentos *side-channel*. Estes vazamentos consistem em dados não criptografados identificados em capturas de rede de larga escala e submetidos a análises estatísticas a fim de correlacionar as informações da rede com o comportamento humano. Assim, realiza-se a inferência de informações privadas dos usuários, ou seja, os ataques *traffic side-channel*. No entanto, ambos os estudos são no contexto de redes tradicionais, se abstendo das redes IoT.

Existem trabalhos que identificam e analisam o impacto causado pelos vazamentos *side-channel* no contexto da IoT. [Yan et al. 2017] são os precursores neste contexto, pois provaram a existência dos vazamentos *side-channel* e quais características eles podem revelar em redes baseadas no protocolo 6LoWPAN. Para isso, os autores capturaram o tráfego de uma rede composta por dois dispositivos diferentes que executam os protocolos padronizados para IoT. [Prates et al. 2019] realizaram uma análise sobre os vazamentos temporais *side-channel* e revelaram a possibilidade de encontrar características únicas mesmo em dispositivos idênticos, *i.e.*, dispositivos rodando os mesmos protocolos e aplicações, fomentando novas possibilidades para a quebra de privacidade dos usuários. Através de um cenário experimental composto por três dispositivos IoT, os autores realizaram uma série de requisições e capturas do tráfego da rede, onde extraíram os dados relacionados ao tempo e submeterem para análises estatísticas. As características estatísticas extraídas do instante de envio e do tempo de resposta foram submetidas a algoritmos de classificação, nos quais a rede neural *Multilayer Perceptron* e o classificador *Random Forest* alcançaram precisão de 100% na identificação dos dispositivos. A eficiência na classificação de dispositivos é devido às soluções apresentadas em [Selis and Marshall 2017], onde apesar de implementar uma forma de identificação para outro tipo de ataque, a metodologia utilizada melhora a exploração das variáveis de tempo. No entanto, apesar das evoluções apresentadas, nenhum dos trabalhos apresentaram ou avaliaram soluções para os vazamentos *side-channel*.

A maioria dos trabalhos que propuseram soluções de defesa para os ataques *side-channel* na IoT utilizam métodos para mascarar ou ocultar os vazamentos [Patranabis et al. 2018]. Os autores em [Li et al. 2017] propuseram uma técnica de ocultação que mascara operações de criptografia dos ataques SPA, através da equalização por compensação do consumo de energia de um circuito. [Yu and Köse 2017] implementaram o algoritmo de criptografia AES baseado em chave falsa, o qual evita que a chave secreta seja armazenada e vaze da caixa de substituição sob ataques CPA. Apesar destes trabalhos dificultarem a identificação dos vazamentos *side-channel*, eles não consideraram o mesmo tipo de ataque e os vazamentos capturados a partir do tráfego de rede, como nos ataques *traffic side-channel*. No melhor do nosso conhecimento, apenas um trabalho apresentou uma solução para os ataques focados neste trabalho. [Xiong et al. 2018] implementaram uma técnica baseada na privacidade local diferencial, a fim de ocultar o tamanho dos pacotes e impedir que informações estatísticas sejam extraídas deste dado encontrado em capturas

de tráfego. No entanto, a informação tempo não foi considerada. Além do mais, as avaliações realizadas não foram testadas em um cenário experimental, sendo avaliadas somente a partir de testes empíricos, não especificando os protocolos utilizados.

Neste sentido este trabalho propõe um mecanismo que identifica e oculta os vazamentos temporais como forma de defesa contra os ataques *traffic side-channel* em um cenário composto pelos principais protocolos padronizados para a IoT. Ele segue dois módulos, teste de vulnerabilidade e proteção de privacidade. O módulo de teste de vulnerabilidade extrai características estatísticas dos vazamentos temporais para melhorar a capacidade de identificação como em [Selis and Marshall 2017]. O módulo de proteção de privacidade, mascara o comportamento dos dispositivos a fim de ocultar os vazamentos *side-channel*, como em [Xiong et al. 2018, Yu and Köse 2017, Li et al. 2017], apesar de considerarem outros tipos de vazamentos. Entretanto, o principal diferencial deste trabalho compreende a arquitetura do mecanismo que utiliza em conjunto ferramentas de identificação e defesa contra os ataques *traffic side-channel* temporais.

3. Mecanismo de Defesa Contra Ataques *Traffic Side-Channel*

Esta seção apresenta o Mecanismo de Defesa Contra Ataques *Traffic Side-Channel*. Este mecanismo melhora a privacidade dos usuários e dos dispositivos e evita a ocorrência de ataques *traffic side-channel*. Este trabalho assume uma rede de sensores executando os protocolos 6LoWPAN e CoAP, conectados à Internet, simulando uma aplicação de coleta de luminosidade. O mecanismo atua nesta rede como um serviço virtual a fim de ocultar os vazamentos *side-channel*. Ele é composto pelos módulos de teste de vulnerabilidades e proteção de privacidade, como mostra a Figura 1. O primeiro módulo realiza as operações básicas para descoberta de recursos que compõem a estrutura de rede e desempenha uma rotina de requisições para a caracterização dos dispositivos, com o intuito de identificar os vazamentos *side-channel*. O segundo módulo mascara o comportamento dos dispositivos a fim de ocultar os vazamentos temporais *side-channel*. Nas próximas subseções descrevemos cada um desses módulos.

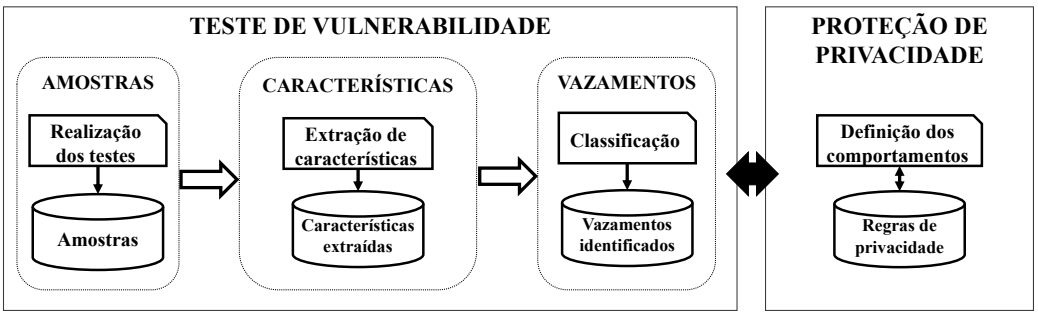


Figura 1. Arquitetura do Mecanismo

3.1. Teste de Vulnerabilidade

O módulo de teste de vulnerabilidade coleta o tráfego de rede e identifica os vazamentos *side-channel*. Para isso, ele segue três funções principais (Figura 1): *(i)* amostragem dos dados coletados, *(ii)* extração de características estatísticas e a *(iii)* identificação dos vazamentos temporais. A partir de coletas de tráfego da rede, a amostragem desses dados

extrai os dados necessários para o cálculo do tempo de resposta. Assim, cada amostra criada contém o instante de envio da requisição, o instante de recebimento da resposta e o tempo de resposta. Em seguida, o mecanismo extrai as medidas estatísticas, sendo elas a média, moda, mediana, limite superior e inferior, e a correlação de *Pearson*. A entrada para a computação dessas medidas estatísticas compreende os dados extraídos na amostragem, ou seja, os instantes e o tempo de resposta. Assim, todas essas informações em conjunto auxiliam na caracterização dos dispositivos. Por conseguinte, submetem-se as amostras para a fase de identificação dos dispositivos através do classificador *Random Forest* e da rede neural *Multilayer Perceptron*. Para isso, o módulo organiza as amostras em conjuntos de dados de treino e teste. Com base na capacidade dos classificadores, identificam-se os vazamentos temporais *side-channel*.

3.2. Proteção de Privacidade

Este módulo reduz a probabilidade dos atacantes inferirem informações que violam a privacidade dos usuários. A privacidade consiste do 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]. Os ataques *traffic side-channel* ameaçam esse direito explorando a informação tempo, pois ela revela características importantes sobre os dispositivos [Prates et al. 2019, Yan et al. 2017]. Através disso, estas características podem ser cruzadas com informações relacionadas ao comportamento humano, possibilitando a inferência de informações pessoais críticas, como os cômodos de uma casa, a quantidade de visitantes, entre outros [Srinivasan et al. 2008]. A fim de ocultar os vazamentos temporais *side-channel*, o módulo de privacidade recebe os vazamentos identificados e realiza uma base de cálculo sobre as amostragens coletadas pelo módulo anterior. Com base no resultado dos cálculos, define-se um comportamento para cada dispositivo, visto que a modificação do comportamento manipula as capturas de tráfego realizadas, dificultando a identificação dos dispositivos através de análises realizadas sobre o tempo. Assim, empregam-se dois métodos: de *aproximação* e *inserção*. No método de *aproximação*, o *gateway* duplica as requisições direcionadas para um dispositivo e recebe as respostas de mais de um dispositivo, aproximando assim os instantes de envio e recebimento capturados. Enquanto, o método de *inserção* adiciona atrasos nas operações de rede no *kernel* do sistema operacional dos dispositivos finais IoT, influenciando no tempo de resposta dos dispositivos.

O tempo de resposta pode ser explorado considerando os atrasos de rede, como o atraso de processamento e propagação. O atraso de processamento representa o tempo que um dispositivo leva para receber, interpretar, montar um pacote e responder determinadas requisições. Ou seja, representa a soma de todas as micropartículas de tempo das inúmeras operações computacionais executadas por cada função. O atraso de propagação representa o tempo que o pacote demora para ser transmitido pelo meio físico. Com base nisso, o tempo de resposta de uma requisição representa a soma do atraso de propagação do envio (ΔT_0), do atraso de processamento (ΔT_1) e do atraso de propagação da resposta (ΔT_2). A Figura 2 apresenta uma visão geral dos atrasos que compõem o tempo de resposta. Através de uma captura de tráfego, estas variações de tempo são embutidas no tempo de resposta, pois a única informação acessível é o instante de tempo em que o pacote foi enviado ou recebido por um dispositivo.

O método de *aproximação* harmoniza as capturas, forçando que dois ou mais dis-

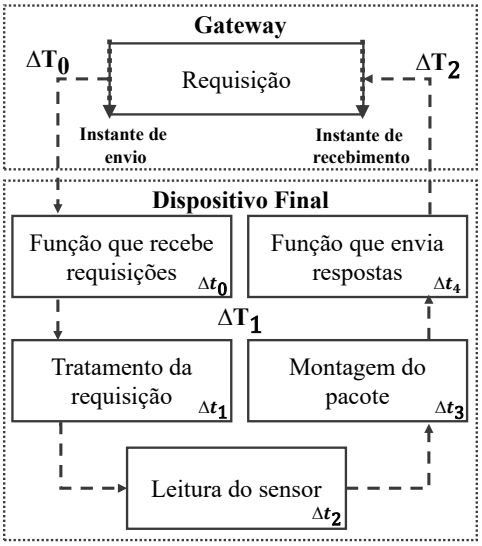


Figura 2. Visão Geral dos Atrasos

positivos 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 (tempo de resposta). Estes dados podem ser cruzados com dados de comportamento humano, revelando com detalhes informações pessoais críticas, como os cômodos de uma casa inteligente, a rotina dos usuários, entre outros [Srinivasan et al. 2008]. 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 *aproximação* dificulta esta capacidade de identificação definindo que o *gateway*, dispositivo que encaminha as requisições externas à rede, envia a requisição recebida por um dispositivo para outros dois dispositivos, imitando as características do dispositivo almejado originalmente.

O método de *inserção* controla o tempo de resposta através da inserção de um atraso em ΔT_1 nos 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*, assume-se que as capturas são representadas por um con-

junto $X_i = \{x_1, x_2, ..., x_n\}$ 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 ocultar as informações 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*. 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, definindo um novo conjunto ordenado $D = \{M_1, M_2, ..., M_i\}$, a partir deste conjunto, é possível calcular o atraso r_i para acrescentar em ΔT_1 de cada dispositivo final. O atraso é definido pela Equação 1.

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

4. Avaliação de Desempenho

Esta seção apresenta uma avaliação de desempenho do mecanismo proposto conduzida sobre um cenário experimental de rede. Neste cenário experimental, um cliente realiza requisições para três dispositivos IoT Memsic Iris, que atuam como servidores CoAP. Durante essas requisições, o mecanismo captura o tráfego da rede através da ferramenta de monitoramento de redes Wireshark. A partir da captura, na ferramenta de análise estatística R, ele extrai características relacionadas ao tráfego, como o tempo de resposta e o instante de envio. Tais características servem como entrada para a caracterização e identificação do tráfego de cada dispositivo. Na ferramenta de mineração de dados WEKA, foi realizada a identificação dos dispositivos IoT por meio dos classificadores *Random Forest* e *Multilayer Perceptron*. Além disso, conforme os módulos do mecanismo proposto, ao longo das avaliações foram comparadas três abordagens de ocultação dos vazamentos temporais *side-channel* a fim de eleger a melhor abordagem. Dessa forma, a abordagem de ocultação é considerada eficaz quando os classificadores apresentarem uma redução no desempenho da identificação dos dispositivos através dos vazamentos.

4.1. Cenário Experimental de Rede

O cenário experimental é composto por quatro dispositivos IoT Memsic Iris e um computador. Os dispositivos IoT realizam diferentes operações, três deles agem como servidores e o último como estação base. Estes dispositivos IoT são equipados com *chips* Atmel's AT86RF230 compatíveis com as especificações IEEE 802.15.4 [Montenegro et al. 2007a]. Além disso, nos dispositivos podem ser acopladas placas proprietárias de sensores. As utilizadas neste cenário são as placas *MTS300CB* equipadas com sensores de iluminação. A estação base, por sua vez, atua como um *gateway* e roteador de borda, ou seja, recebe e encaminha as requisições externas. Todos os dispositivos IoT executam o Sistema Operacional Contiki, com os protocolos de rede 6LoWPAN [Montenegro et al. 2007b], RPL [Thubert et al. 2017] e UDP. Para a aplicação, o protocolo CoAP [Shelby et al. 2014]. A Tabela 4.1 organiza os protocolos utilizados no modelo de cinco camadas. Por fim, o computador simula um cliente CoAP gerando requisições e capturando

o tráfego gerado. A simulação do cliente é feita através do *framework* Californium e as capturas são realizadas através da ferramenta Wireshark compreendendo as camadas física, enlace e rede. O computador e o roteador de borda são interconectados através de uma interface serial/USB. Além disso, neste cenário os servidores só realizam a captura dos sensores quando recebem uma requisição.

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

Tabela 1. Protocolos Padronizados para IoT

4.2. Abordagens Avaliadas

Três abordagens de ocultação foram avaliadas: **A1**, **A2** e **A3**. A primeira abordagem **A1** segue o método de *aproximação* do mecanismo, 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*. **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. Estas abordagens implementam os diferentes métodos de manipulação das variáveis de tempo apresentados na Seção 3, 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.

4.3. Detalhes da Identificação dos Dispositivos

Para testes de ocultação, utilizamos dois algoritmos de aprendizagem de máquina amplamente utilizados na literatura [Pacheco et al. 2018] e previamente avaliados em [Prates et al. 2019]: o algoritmo baseado em árvores de decisão *Random Forest* e a rede neural *Multilayer Perceptron*. Esses algoritmos são aplicados em problemas de multi-classificação, ou seja, em situações que várias classes de dados precisam ser identificadas. Esta avaliação segue um problema de multi-classificação, pois o objetivo é identificar o tráfego de dados dos dispositivos IoT considerados, o que justifica a escolha de tais algoritmos de classificação. 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]. Os conjuntos de dados de treino possuem uma captura de 100.000 requisições de dados para cada uma das avaliações, totalizando 300.000 requisições. As capturas de cada dispositivo foram divididas em conjuntos de 1.000 amostras, a fim de calcular as características estatísticas conforme apresentadas pelo módulo de teste de vulnerabilidade.

As métricas de desempenho consideradas 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 a 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 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 quatro métricas.

5. Resultados

Esta seção apresenta 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 realiza a ocultação das 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 *aproximação* do mecanismo proposto e as abordagens **A2** e **A3** implementam o método de *inserção* do mecanismo (as abordagens são descritas na Seção 4). Além disso, os resultados se embasam no tráfego de três dispositivos idênticos IoT 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.

Em relação à captura do tráfego da rede, a Figura 3 apresenta o comportamento do tráfego de dados dos três dispositivos IoT Memsic Irirs (*Nó 1, Nó 2* e *Nó 3*), sem nenhum tipo de atraso considerados em nossas análises. 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. Mais especificamente, o *Nó 1* alcançou um tempo médio de resposta entre 41ms e 43ms, o *Nó 2* atingiu valores em torno de 30ms, enquanto o *Nó 3* obteve até 45ms. Apesar da pequena diferença, nota-se um comportamento característico de cada dispositivo, o que facilita a identificação os vazamentos *side-channel*.

Com o intuito de ocultar os vazamentos *side-channel*, os gráficos da Figura 4 apresentam o tempo de resposta *versus* o número de requisições de cada dispositivo para cada abordagem considerada. Inicialmente, a Figura 4(a) mostra os resultados da abordagem **A1**, onde a variável tempo de cada dispositivo é controlada conforme o módulo de *aproximação*. 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 **A1**, todos os dispositivos atingiram tempo de médio de resposta em torno de 44ms, sendo clara a ocultação da variável tempo quando comparado ao comportamento do tráfego na Figura 3. Ao ser avaliado pelo módulo de teste de vulnerabilidade, os classificadores tiveram uma queda considerável nas taxas das métri-

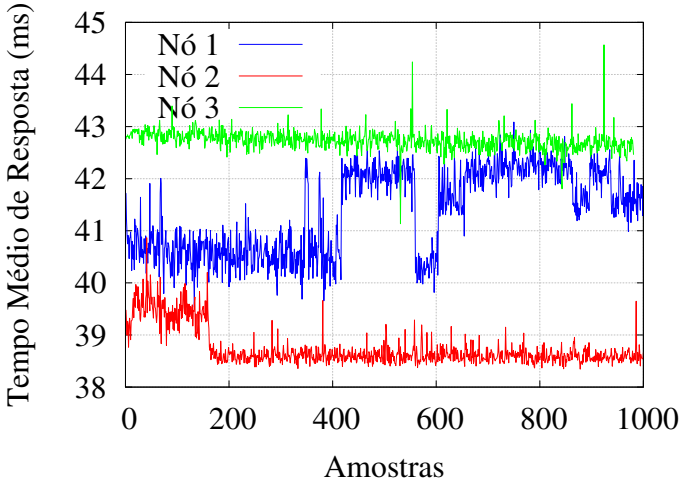
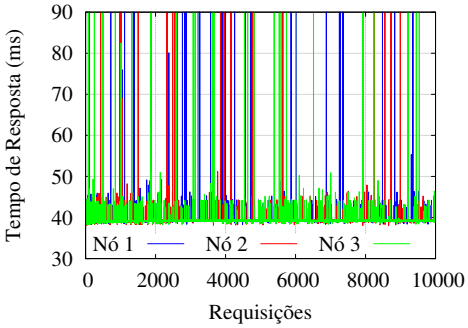
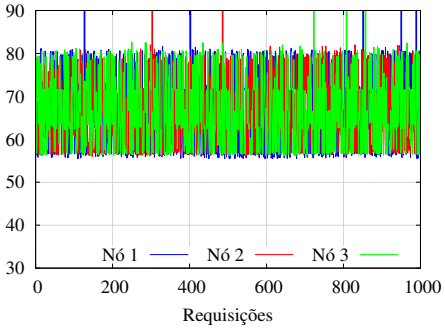


Figura 3. Comportamento do Tráfego dos Dispositivos

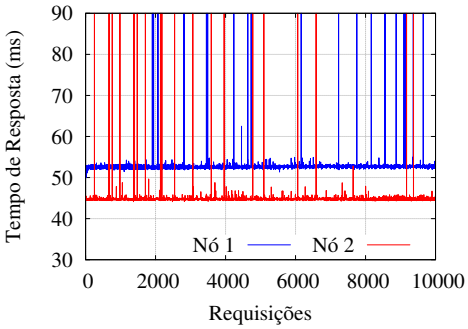
cas de desempenho, apresentando uma diminuição de aproximadamente 63% na acurácia (Figura 5(a)). Além disso, esta abordagem possui algumas vantagens, por exemplo, para controlar o tempo não foi necessário alterar o atraso de processamento (tempo que o dispositivo leva desde o recebimento até o envio das respostas). Também, por apresentar um tempo de resposta menor em relação as outras abordagens da Figura 4, há uma menor chance de prejudicar outros protocolos e/ou aplicações (*ex.*, não prejudica mecanismos de garantia de entrega dos pacotes de dados).



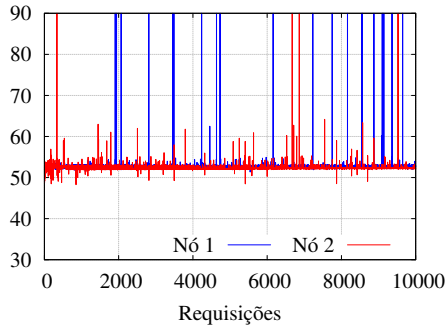
(a) Abordagem **A1** - Método de *aproximação*



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



(c) Captura do Abordagem **A3**

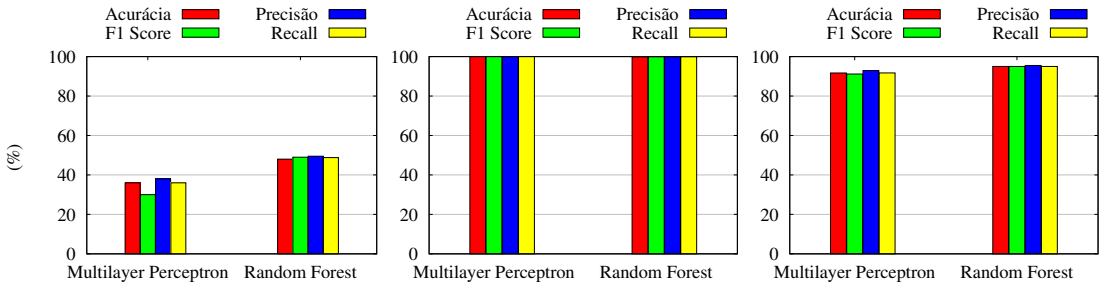


(d) Abordagem **A3** - Método de *inserção*

Figura 4. Resultados sobre a Ocultação dos Vazamentos *Side-Channel*

Em contrapartida, na abordagem **A2** os atrasos foram inseridos pelo módulo de

inserção de forma aleatória conforme os ciclos do relógio de controle de processos. A Figura 4(b) apresenta os resultados obtidos em **A2**, onde os valores para o tempo de resposta dos três dispositivos se sobrepõem, alcançando parcialmente o objetivo de ocultação da variável tempo. No entanto, os classificadores não apresentaram dificuldade ao identificar os dispositivos, mantendo as métricas em 100%, como mostra a Figura 5(b). Para a abordagem **A3**, explorou-se a capacidade de controle sobre os tempos médios de resposta, conforme pode ser observado nas Figuras 4(c) e 4(d). Na Figura 4(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 4(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(c), na qual o classificador *Random Forest* ainda conseguiu classificar com sucesso de 95% da base de teste.



(a) Identificação em **A1** (b) Identificação em **A2** (c) Identificação em **A3**

Figura 5. Desempenho dos Classificadores na Identificação dos Dispositivos

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*. Em um trabalho prévio, os classificadores *Multilayer Perceptron* e *Random Forest* identificaram os dispositivos IoT através dos vazamentos temporais *side-channel* com precisão de 100% [Prates et al. 2019]. Neste trabalho, observa-se que 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. Isto é mostrado nas Figuras 4(c) e 4(d), onde foi possível inserir atrasos, aproximando os tempos médios de resposta com uma determinada precisão. Além disso, oculta-se estes vazamentos através das variáveis que influenciam as capturas realizadas sobre o tempo. No método de *aproximação* 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(a)) se comparados aos resultados das abordagens **A2** e **A3** (Figuras 5(b) e 5(c)), implementando o método de *inserção* (inserem atrasos no tempo de resposta).

6. Conclusão

Este artigo propôs um mecanismo de defesa contra ataques *traffic side-channel* com o intuito de melhorar a privacidade dos usuários e dos dispositivos das redes IoT. Considerou-se uma rede de sensores executando os protocolos 6LoWPAN e CoAP, conectados à Internet, simulando uma aplicação de coleta de luminosidade. Assim, o mecanismo atua nesta rede como um serviço virtual que oculta os vazamentos *side-channel*. Na avaliação de desempenho do mecanismo, realizaram-se análises sobre três abordagens compostas por diferentes formas de manusear as variáveis 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 na identificação dos dispositivos e que mesmo com os tempos de resposta manipulados, os classificadores são capazes de identificá-los. Porém, o mecanismo implementando o método de *aproximação* reduziu a capacidade de identificação dos dispositivos em 63%. Como direções futuras, pretende-se avaliar o impacto que os métodos explorados geram nos atributos da rede como largura de banda e latência. Também, avaliar a eficiência do mecanismo proposto *online*.

Agradecimentos

Os autores agradecem o apoio da UFPR, CAPES e CNPq. Este trabalho contou com auxílio financeiro do projeto PROA CNPq/Universal #432204/2018-0 e bolsas de pesquisa do processo CNPq #309129/2017-6 e processos CAPES #1806338 e #1758423.

Referências

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Commun. Surveys & Tuts.*, 17(4):2347–2376.

Cervantes, C., Poplade, D., Nogueira, M., and Santos, A. (2015). Detection of sinkhole attacks for supporting secure routing on 6lowpan for internet of things. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 606–611.

Chen, S., Wang, R., Wang, X., and Zhang, K. (2010). Side-channel leaks in web applications: A reality today, a challenge tomorrow. In *Symposium on Security and Privacy*, pages 191–206. IEEE.

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.

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

Montenegro, G., Kushalnagar, N., Hui, J., and Culler, D. (2007a). Transmission of IPv6 packets over IEEE 802.15.4 networks. Technical report, IETF.

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

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

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

Prates, N., Pelloso, M., Macedo, R., and Nogueira, M. (2018). Ameaças de segurança, defesas e análise de dados em IoT baseada em SDN. In *Minicursos SBSeg 2018*, chapter 1, pages 1–50. SBC.

Prates, N., Vergütz, A., Macedo, R., and Nogueira, M. (2019). Análise de vazamentos temporais side-channel no contexto da internet das coisas. *Anais do Workshop de Gerência e Operação de Redes e Serviços (WGRS - SBRC)*, 24:157–170.

Selis, V. and Marshall, A. (2017). A fake timing attack against behavioural tests used in embedded IoT M2M communications. In *Cyber Security in Netw. Conference*, pages 1–6. IEEE.

Shelby, Z., Hartke, K., and Bormann, C. (2014). The Constrained Application Protocol (CoAP). Technical Report 7252, IETF.

Srinivasan, V., Stankovic, J., and Whitehouse, K. (2008). Protecting your daily in-home activity information from a wireless snooping attack. In *International Conference on Ubiquitous Comput.*, pages 202–211. ACM.

Thubert, P., Bormann, C., Toutain, L., and Cragie, R. (2017). IPv6 over low-power wireless personal area network (6LoWPAN) routing header. Technical report, IETF.

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

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

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

Yu, W. and Köse, S. (2017). A lightweight masked AES implementation for securing IoT against CPA attacks. *IEEE Trans. Circuits Syst. I, Reg. Papers*, 64(11):2934–2944.