UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ISRAEL DE CASTRO VIDAL

PROTECTING: GARANTINDO A PRIVACIDADE DE DADOS GERADOS EM CASAS INTELIGENTES LOCALMENTE NA BORDA DA REDE

FORTALEZA

2020

ISRAEL DE CASTRO VIDAL

PROTECTING: GARANTINDO A PRIVACIDADE DE DADOS GERADOS EM CASAS INTELIGENTES LOCALMENTE NA BORDA DA REDE

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Ciência da Computação.

Orientador: Prof. Dr. Javam de Castro Machado.

FORTALEZA

2020

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

V691p Vidal, Israel de Castro.
ProTECting : Garantindo a privacidade de dados gerados em casas inteligentes localmente na borda da rede / Israel de Castro Vidal. – 2020.
85 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2020.
Orientação: Prof. Dr. Javam de Castro Machado.

1. Privacidade diferencial. 2. Casas inteligentes. 3. Streaming de dados. I. Título.

CDD 005

ISRAEL DE CASTRO VIDAL

PROTECTING: GARANTINDO A PRIVACIDADE DE DADOS GERADOS EM CASAS INTELIGENTES LOCALMENTE NA BORDA DA REDE

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Ciência da Computação.

Aprovada em: 09/10/2020.

BANCA EXAMINADORA

Prof. Dr. Javam de Castro Machado (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. José Maria da Silva Monteiro Filho
Universidade Federal do Ceará (UFC)

Prof. Dr. Leonardo Oliveira Moreira
Universidade Federal do Ceará (UFC)

Prof. Dr. Carlos Eduardo Santos Pires
Universidade Federal de Campina Grande (UFCG)

Aos meus pais, família e amigos que contribuíram para que eu fosse capaz de trilhar o caminho que me levou até aqui.

AGRADECIMENTOS

Aos meus pais, Ofélia e Neto, por todo o apoio que deram para que eu me dedicasse aos estudos. Aos meus irmãos, Nathielly e Eduardo, que apesar de todas as brigas e implicâncias ao longo da vida, sempre estiveram lá para me apoiar. À minha namorada, Fernanda, por ter estado ao meu lado durante essa trajetória de mestrado, e além dela, ajudando a lidar com os estresses e me apoiando.

Ao meu orientador, Prof. Dr. Javam Machado, por me dar o suporte necessário para concluir este trabalho, tanto no que diz respeito a orientação científica, quanto disponibilizando um ambiente propício para que este trabalho fosse desenvolvido.

Aos meus amigos e companheiros de pesquisa, seja de maneira direta ou indireta: Daniel Praciano. Davi Torres, Diogo Martins, Eduardo Rodrigues, Edvar Bento, Felipe Timbó, Hélio Sales, Iago Chaves, Isabel Lima, José Serafim, Malú Maia, Paulo Amora, Sérgio Marinho e Victor Aguiar. Em particular, ao André Luís, que contribuiu diretamente neste trabalho, me ajudando com discussões intermináveis e ajudando na revisão deste texto, assim como ao Bruno Leal, cujas discussões científicas e maratonas de escrita de artigos me ajudaram a desenvolver a perseverança e paciência que são necessárias para desenvolver um trabalho científico.

À todos os integrantes do LSBD, cujas interações durante os cafés ajudavam a tornar o dia-a-dia mais leve, assim como as conversas de corredor que certamente contribuíram, de maneira direta ou indireta, para o desenvolvimento deste trabalho.

À todos os professores do curso de Ciência da Computação da UFC, que me ensinaram toda a base prática e teórica que culminou no desenvolvimento deste trabalho, além de todo o aprendizado não acadêmico que obtive com as interações com eles.

À Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP), pelo suporte financeiro durante o desenvolvimento desde trabalho.

À todos, meus mais sinceros agradecimentos por todo o apoio. Vocês foram de suma importância para que este trabalho viesse a ser concluído.

RESUMO

Com o crescimento da Internet das Coisas (IoT) e casas inteligentes, há uma quantidade crescente de dados provenientes das casas das pessoas. Esses dados são valiosos para análise e descoberta de padrões, a fim de melhorar serviços e produzir recursos com mais eficiência, por exemplo, usando dados de medidores inteligentes para gerar energia com menos desperdício. Apesar de seu alto valor para análise, esses dados são intrinsecamente privados e devem ser tratados com cuidado. Os dados de IoT são fundamentalmente infinitos, e essa propriedade torna ainda mais desafiador aplicar modelos convencionais para obter privacidade. Neste trabalho, propomos uma solução baseada em Privacidade Diferencial Local para estimar frequências de valores no contexto de dados de casas inteligentes. Nossa solução é dividida em duas estratégias diferentes, sendo a primeira baseada no conceito de janela deslizante, garantindo privacidade diferencial para um número definido de respostas. A segunda estratégia utiliza duas rodadas de randomização e funciona mesmo para um número infinito de respostas, ao mesmo tempo em que foca em obter uma melhor utilidade do que o *baseline*. No que diz respeito à resultados específicos de comparação com o *baseline*, a estratégia de dupla randomização foi capaz de obter uma redução de pelo menos 35% no Erro Quadrático Médio (EQM). Quanto à *Jensen-Shannon Distance* / Distância de Jensen-Shannon (JSD), a outra métrica utilizada para quantificar a utilidade neste trabalho, a redução média mínima foi de, aproximadamente, 17%.

Palavras-chave: Privacidade diferencial. Casas inteligentes. Streaming de dados.

ABSTRACT

With the growth of the Internet of Things (IoT) and Smart Homes, there is an ever-growing amount of data coming from within people's houses. These data are valuable for analysis and to discover patterns in order to improve services and produce resources more efficiently, e.g., using smart meter data to generate energy with less waste. Despite their high value for analysis, these data are intrinsically private and should be treated carefully. IoT data are fundamentally infinite, and this property makes it even more challenging to apply conventional models to achieve privacy. In this work, we propose a locally differentially private solution to estimate frequencies of values in the context of Smart Home data. Our solution is divided into two different strategies, the first one being based on the concept of sliding window, and guaranteeing differential privacy for a defined number of reports. The second strategy uses two rounds of randomization and works even for an infinite number of reports while focusing on getting better utility than the baseline. Concerning specific results of comparison with baseline, the double randomization strategy was able to achieve a reduction of at least 35% in Erro Quadrático Médio (EQM). As for *Jensen-Shannon Distance* / Distância de Jensen-Shannon (JSD), the other metric used to quantify the utility in this work, the minimum average reduction was approximately 17%.

Keywords: Differential privacy. Smart homes. Streaming data.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

AP	*Access Point* Wifi / Ponto de Acesso Wifi
CD	Codificação Direta
CR	Conjunto de Respostas
DUE	*Discrete Unary Encoding* / Codificação Unária Discreta
EG	*Edge Gateway* / *Gateway* de Borda
EQM	Erro Quadrático Médio
IoT	*Internet of Things* / Internet das Coisas
JSD	*Jensen-Shannon Distance* / Distância de Jensen-Shannon
OF	Oráculo de Frequência
OPT-DR	*Optimized Double Randomization* / Dupla Randomização Otimizada
OPT-WB	*Optimized Window Based* / Otimizada Baseada em Janela
PD	Privacidade Diferencial
PDL	Privacidade Diferencial Local
PeGaSus	*Perturb-Group-Smooth* / Perturbar-Agrupar-Suavizar
ProTECting	***Pr**ivacy for **IoT** and **E**dge **C**omput**ing***
PS	Provedor de Serviços
RR	Resposta Randômica
UE	*Unary Encoding* / Codificação Unária
WB	*Window Based* / Baseada em Janela

LISTA DE SÍMBOLOS

N	Número de casas inteligentes que enviará dados para o Provedor de Serviços
ε	Limite (ou *Budget*) de privacidade
w	Número de respostas aleatórias a serem enviadas utilizando um dado ε, utilizando a estratégia baseada em janela
H	Parâmetros enviados do Provedor de Serviço para os participantes usados na codificação e decodificação de valores
d	Número de barras utilizadas na representação de histogramas
valor_min	Valor mínimo do domínio de valores
valor_max	Valor máximo do domínio de valores
δ	Intervalo de tempo entre o envio de duas respostas
B	Vetor de bits representando um valor v após a execução da Codificação Unária
H_{janela}	Parâmetros enviados do Provedor de Serviços (PS) para os participantes na estratégia baseada em janela deslizante
M	Tamanho do conjunto de respostas utilizado pelo Provedor de Serviços (PS) para estimar as frequências de valores

SUMÁRIO

1 INTRODUÇÃO

1.1 Motivação

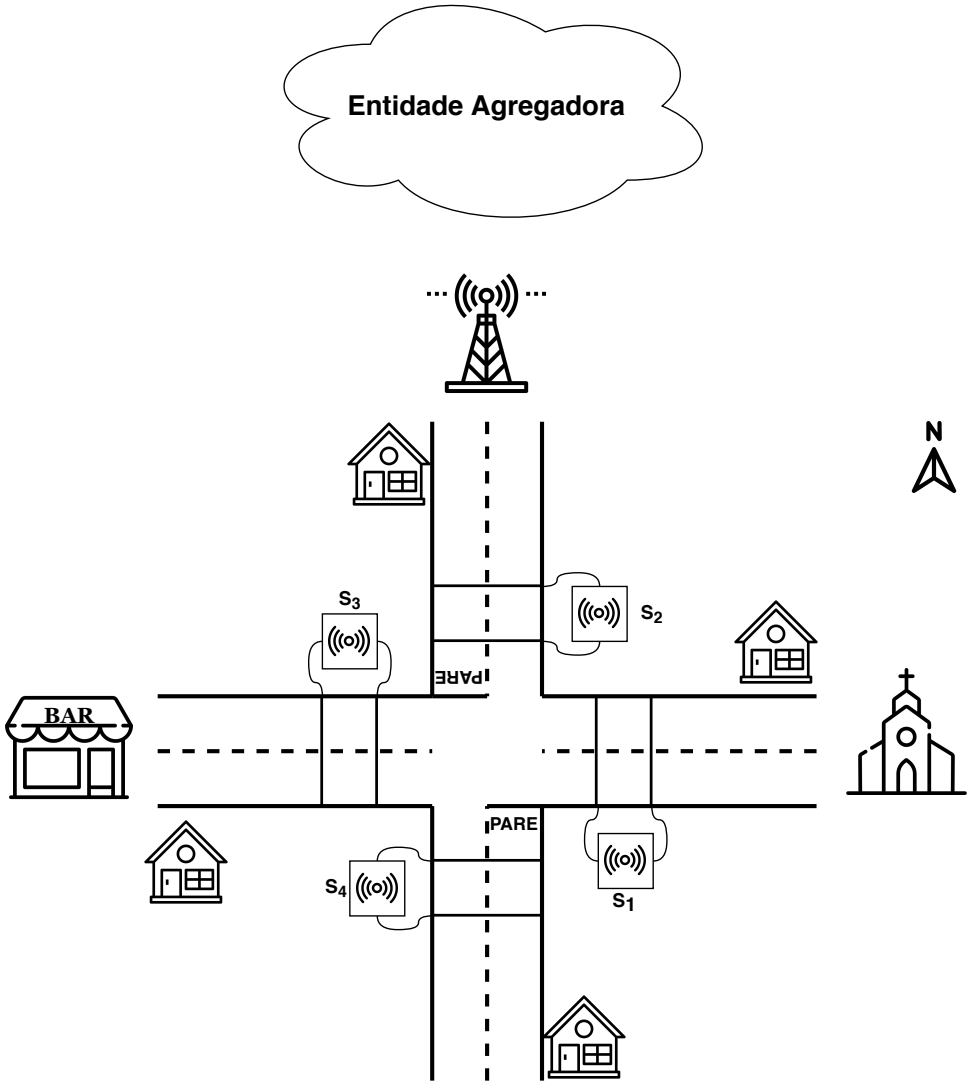
Com a popularização da Internet das Coisas (IoT) e a maior disponibilidade de uma grande variedade de sensores no mercado, há uma quantidade crescente de dados sendo gerados. Espera-se que até o ano de 2021, a quantidade de dados gerados por dispositivos, pessoas e máquinas de IoT atinja a magnitude de *zettabytes* (NETWORKING, 2016). Esses dados podem ser benéficos para prover melhoria de serviços, por exemplo, ao usar dados de medidores inteligentes para obter uma melhor compreensão do uso de energia em uma cidade.

Costumeiramente, dados de *Internet of Things* / Internet das Coisas (IoT) apresentam-se como *streaming* de dados, o que traz alguns desafios devido as suas características intrínsecas, como, por exemplo, o fato de serem potencialmente infinitos e surgirem em uma ordem não previsível. Além disso, soluções no contexto de IoT muitas vezes exigem resposta rápida, o que pode inviabilizar a estratégia tradicional de enviar os dados para serem processados em um servidor de nuvem. Uma alternativa para a computação em nuvem é processar dados na borda da rede, o que motiva o surgimento da computação em borda (Shi *et al.*, 2016).

Embora esses dados possam ser benéficos para a melhoria dos serviços, é de extrema importância que haja um cuidado minucioso no que diz respeito à privacidade dos indivíduos produtores desses dados. A falta de cuidado com a privacidade pode levar a graves problemas, por exemplo, um atacante mal intencionado coletando informações sensíveis dos usuários. Igualmente grave seria o uso desses dados em processamento não autorizado pelo indivíduo dono do dado. Como mostrado em (MOLINA-MARKHAM *et al.*, 2010), utilizando métodos estatísticos simples, um atacante é capaz de inferir uma grande quantidade de informações privadas a partir de dados pessoais coletados. Por exemplo, dados de consumo de energia podem revelar se as pessoas de uma determinada casa tomam café da manhã quente ou frio, ou ainda se uma criança foi deixada sozinha em casa.

Para melhor apresentarmos alguns conceitos importantes sobre dados de sensores, *streaming* de dados, assim como para expormos um exemplo de como esses dados podem ser utilizados e um vazamento de privacidade proveniente do uso desses dados, apresentamos o exemplo bastante simplificado de uma Cidade Inteligente, ilustrada na Figura 1, juntamente com uma história fictícia dos moradores dessa cidade. Esse exemplo apresenta uma cidade pacata onde vivem apenas quatro pessoas: André, o padre da cidade, Eduardo, o dono da empresa de

Figura 1 – Exemplo de Cidade Inteligente



tecnologia, responsável pela instalação de um sensor em cada segmento de rua da cidade, Iago, o dono do bar, e Daniel, o professor. Cada um dos sensores instalados na cidade por Eduardo tem a capacidade de identificar quando uma pessoa passa sobre si e, em seguida, emitir um sinal para a torre da cidade que, então, envia esse dado para a Entidade Agregadora, que é responsável por processar esses dados com o intuito de entender melhor o fluxo de pessoas na cidade e, com isso, propor projetos que melhorem a vida das pessoas, como por exemplo desligando a iluminação pública de regiões que estão inabitadas em um dado instante de tempo e, com isso, gerando uma redução dos gastos da cidade. Os dados gerados pelos sensores são compostos por tuplas $t_j = (s_i, t_j, c)$, onde s_i é o identificador do sensor, t_j é o instante de tempo em que o dado foi gerado, e c é a quantidade de pessoas detectadas pelo sensor s_i no instante de tempo t_j. Perceba que os dados que passam pela antena da cidade podem ser vistos como uma *stream* infinita de tuplas $S = \{t_0, t_1, t_2, t_3, \dots\}$. Um prefixo dessa *stream* trata-se de um conjunto finito dessas tuplas.

Após muita insistência do professor Daniel, o dono da empresa de tecnologia, Eduardo, decide disponibilizar os dados gerados através dos sensores da cidade. Ao analisar os dados provenientes de uma noite em que Daniel possuía o conhecimento prévio de que todas as pessoas da cidade estavam na missa, Daniel foi capaz de chegar à conclusão que, depois da missa, o Padre André saiu para beber no bar do Iago, portanto, violando a privacidade dos moradores dessa cidade fictícia. Para chegar a essa conclusão, o seguinte prefixo da *stream* de dados foi analisado:

$$\begin{aligned} S_p = \{&(S_2, t_1, 1), (S_3, t_1, 1), (S_4, t_1, 1), \\ &(S_1, t_2, 3), \\ &(S_1, t_{60}, 3), \\ &(S_2, t_{61}, 1), (S_3, t_{61}, 1), (S_4, t_{61}, 1), \\ &(S_1, t_{70}, 1), (S_3, t_{71}, 1)\} \end{aligned}$$

Perceba que ao analisar esses dados, podemos ver claramente o fluxo de cada um dos moradores da cidade se deslocando até a igreja, nos instantes de tempo t_1 e t_2. Em seguida, nos tempos t_{60} e t_{61} podemos ver o fluxo a partir da igreja para cada uma das casas e, finalmente, nos tempos t_{70} e t_{71}, observa-se um fluxo da igreja em direção ao Bar. Apesar de se tratar de uma simplificação extrema, essa história é capaz de ilustrar um pouco dos problemas que podem ocorrer ao se disponibilizar dados de sensores, mesmo que à primeira vista esses dados pareçam não conter informações sensíveis dos usuários associados à esses sensores.

Muitos trabalhos na literatura propõem uma estratégia de garantia de privacidade por meio de uma entidade externa confiável que tem acesso aos dados brutos de um conjunto de usuários. Por exemplo, se na história da Cidade Inteligente apresentada e ilustrada na Figura 1, fosse decidido resolver o problema de privacidade que foi apresentado ao implementar um mecanismo de privacidade sobre a Entidade Agregadora. No entanto, em cenários do mundo real, geralmente não é razoável depender dessa entidade por motivos de privacidade. A entidade externa pode ser maliciosa, ou ser hackeada, e acabar expondo as informações pessoais dos usuários, por exemplo. Outros trabalhos focam na perspectiva local da privacidade, onde o processo de privacidade é realizado mais próximo do usuário, sem depender de uma entidade externa confiável.

A Privacidade Diferencial (PD) é o modelo de privacidade que vem se tornando o padrão tanto na academia, quanto na indústria, através de sua versão local, chamada de Privacidade Diferencial Local (PDL). De maneira informal, a PD é uma definição que se

aplica a mecanismos aleatórios que respondem a consultas com base em uma distribuição de probabilidade, tendo como objetivo ofuscar a resposta real dessa consulta. Esse modelo de privacidade é interessante em cenários onde se deseja obter informações agregadas dos dados. Por exemplo, num cenário de uma cidade que possui medidores inteligentes de energia nas casas, a PD poderia ser útil para aprender de maneira privada sobre o padrão de consumo de energia na cidade. Por outro lado, a aplicação da PD para tentar obter informações sobre o padrão de consumo de uma casa não seria possível, pois a quantidade de ruído necessário para tornar essa resposta privada seria tão grande que resultaria em valores que não representariam em nada os reais. Um mecanismo aleatório $\mathcal{M}$ é considerado diferencialmente privado se a probabilidade de qualquer saída de $\mathcal{M}$ não variar significativamente, por um limiar de privacidade de ε, independente da entrada. A PD foi proposta, inicialmente, para funcionar como um modelo interativo (DWORK *et al.*, 2006), respondendo de maneira privada à consultas estatísticas feitas sobre uma base de dados. Nesse cenário interativo, é necessário que exista uma entidade confiável que tem acesso aos dados brutos. Entretanto, em trabalhos mais recentes, como (ERLINGSSON *et al.*, 2014), por exemplo, vem aumentando o interesse pela versão local deste modelo, a PDL.

1.2 Objetivos

Este trabalho estuda o problema da privacidade de indivíduos residentes em casas inteligentes diante da coleta contínua de dados sobre consumo de energia. Nos interessamos particularmente por envios de *streams* de dados realizados por meio da internet a partir de objetos instalados em uma casa para prestadores de serviço que agregam esses dados. Tais prestadores podem aprender sobre o serviço associado ao objeto, mas também tem a chance de descobrir muito mais sobre os indivíduos residentes. Procuramos propor soluções científicas capazes de permitir o aprendizado do serviço, portanto facilitar a sua evolução pelo prestador, e ainda assim garantir a privacidade dos moradores dessas residências inteligentes. Aplicamos os nossos estudos à coleta de dados de medidores inteligentes de consumo de energia residencial.

1.2.1 Objetivo Geral

Diante do cenário apresentado na motivação, o objetivo geral deste trabalho consiste em desenvolver uma solução para o problema de privacidade no contexto de casas inteligentes, onde os dados de IoT se apresentam como *streaming* de dados. A solução tem como objetivo

garantir a privacidade dos indivíduos com o auxílio da PDL, ao mesmo tempo em que busca manter a utilidade das informações extraídas dos dados privados.

1.2.2 Objetivo Específicos

Para alcançar o objetivo geral deste trabalho, os seguintes objetivos específicos são definidos:

- Apresentar uma arquitetura de ataque ao problema, baseando-se no conceito de computação em borda;
- Propor uma solução que atenda a PDL para o contexto de casas inteligentes, com o objetivo de minimizar a perda de informação decorrente do processo de privacidade;
- Garantir que um nível definido de privacidade é válido para o envio de múltiplas respostas, para atender a cenários realistas de *streaming* de dados;
- Avaliar a utilidade da solução proposta utilizando dados reais de IoT.

1.3 Contribuições

Como resultado deste trabalho, propomos o ProTECting, ***Pr**ivacy for **IoT** and **E**dge **C**omput**ing***, que divide-se em duas diferentes estratégias, ambas garantindo a definição da PDL para múltiplas respostas. A primeira estratégia baseia-se no conceito de janela deslizante e garante a PDL para um número limitado de respostas. A segunda estratégia tem como objetivo eliminar a limitação imposta pela estratégia anterior, garantindo a definição da PDL mesmo quando um número infinito de respostas são enviadas. Ambas estratégias do ***Pr**ivacy for **IoT** and **E**dge **C**omput**ing*** (ProTECting) foram avaliadas em termo de utilidade usando duas diferentes métricas: o Erro Quadrático Médio (EQM) e a *Jensen-Shannon Distance* / Distância de Jensen-Shannon (JSD). Em especial para a segunda estratégia, fomos capazes de mostrar que a solução proposta alcança melhor utilidade do que o *baseline* para os mesmos níveis de privacidade.

1.3.1 Produção Científica

O trabalho desta dissertação deu origem a duas publicações científicas, a primeira descreve a estratégia baseada em janela deslizante, e a segunda apresenta a solução que possibilita o envio de infinitas respostas sem a restrição do tamanho de janela. As duas publicações são apresentadas a seguir:

- Israel C. Vidal, Franck Rousseau, and Javam C. Machado, Achieving Differential Privacy in Smart Home Scenarios, Anais do XXXIV Simpósio Brasileiro de Banco de Dados, SBC, 2019.
- Israel C. Vidal, André L. C. Mendonça, Franck Rousseau, and Javam C. Machado, ProTECting: An Application of Local Differential Privacy for IoT at the Edge in Smart Home Scenarios, Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, SBC, 2020.

Além disso, como contribuição indireta deste trabalho, foram publicados artigos que também abordam o problema de privacidade no contexto de *streaming* de dados utilizando a PD centralizada, diferentemente deste trabalho, que foca na sua configuração local. Esses trabalhos são:

- Bruno C. Leal, Israel C. Vidal, Javam C. Machado, Anonimização de Streaming de Dados em DOCA, Anais do XXXIII Simpósio Brasileiro de Banco de Dados, SBC, 2019
- Bruno C. Leal, Israel C. Vidal, Felipe T. Brito, Juvêncio S. Nobre, Javam C. Machado, δ-DOCA: Achieving Privacy in Data Streams, Data Privacy Management, Cryptocurrencies and Blockchain Technology. Springer, Cham, 2018.

1.4 Estrutura da Dissertação

Esta dissertação organiza-se da seguinte forma: No Capítulo 2 são apresentados os principais conceitos relacionados a IoT, computação em borda e PD. Além disso, é apresentada a técnica de Resposta Randômica (RR), que é a base de muitos protocolos diferencialmente privados e, por último, a PDL é definida. O Capítulo 3 apresenta e discute trabalhos relevantes que abordam o problema de privacidade no contexto de casas inteligentes e também no contexto de *streaming* de dados. Posteriormente, no Capítulo 4, a solução proposta, ProTECting, é apresentada e suas duas diferentes estratégias são detalhadas. Em seguida, o Capítulo 5 apresenta a avaliação experimental, definindo as métricas que são utilizadas para medir a utilidade das respostas e, em seguida, apresentando os resultados. Por último, o Capítulo 6 conclui o trabalho e mostra algumas direções futuras de pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados alguns conceitos importantes para o entendimento do método proposto, o ProTECting, bem como o ambiente em que este está inserido. Inicialmente, a Seção 2.1 apresenta os conceitos relacionados a *Internet of Things* / Internet das Coisas (IoT) e a computação em borda. Em seguida, na Seção 2.2, a Privacidade Diferencial (PD) é discutida e suas propriedades apresentadas. Na Seção 2.3 é apresentada a técnica de Resposta Randômica (RR) como uma técnica proposta anteriormente à definição da PD, mas que também satisfaz sua definição. Por último, a Seção 2.4 apresenta a definição de Privacidade Diferencial Local (PDL), explica o que são Oráculos de Frequência (OF) e apresenta seu protocolo mais simples, o Codificação Direta (CD).

2.1 Internet das Coisas e Computação em Borda

IoT é um paradigma de comunicação relacionado à interconexão de objetos do dia-a-dia com a Internet. Esse paradigma tem ganhado força com a evolução das tecnologias sem fio. A ideia por trás da IoT consiste em uma variedade de objetos físicos, também chamados de dispositivos, equipados com sistemas embarcados, sensores e conexões de redes, que são capazes de se comunicar entre si e com os usuários, tornando a Internet ainda mais imersiva e universalizada (KHAN; SALAH, 2018) .

O uso e as interações entre esses dispositivos interconectados tem o potencial de gerar uma grande quantidade de dados e, consequentemente, de aplicações que fazem uso desses dados. Essas aplicações podem ter que lidar com domínios heterogêneos como, por exemplo, automação industrial e doméstica, e gerenciamento de tráfego e de energia.

Nesse cenário complexo, a grande quantidade de dados gerados e transportados por redes de comunicações, juntamente com a característica heterogênea das aplicações, leva a problemas desafiadores. Os desafios dizem respeito não somente a dificuldades envolvendo redes de comunicações e o transporte dos dados por elas, mas também envolve desafios no que se refere à privacidade dos usuários, que, por exemplo, poderiam ter sua intimidade violada por um servidor de nuvem malicioso que coleta e analisa os dados gerados por dispositivos de uma casa para descobrir quando essa casa está ou não ocupada. Os problemas de rede acarretados por cenários de IoT não são o foco deste trabalho e não serão discutidos a diante. No que concerne à privacidade, os problemas, em geral, são acarretados pela falta de cuidado e atenção

com as informações coletadas relacionadas a atividades humanas diárias que, juntamente com métodos estatísticos relativamente simples, podem revelar informações privadas cruciais, como será melhor detalhado na Seção 3.1.1.

Casas Inteligentes, ou *Smart Homes*, são um cenário específico de IoT, onde dispositivos interconectados à Internet localizam-se dentro de uma casa, por exemplo com o objetivo de automatizar uma atividade do dia-a-dia. Um medidor inteligente, ou *smart meter*, é um medidor de energia avançado que, além de medir o consumo de energia elétrica, tem a capacidade de prover informações adicionais quando comparado a um medidor convencional (DEPURU *et al.*, 2011). Uma aplicação conhecida no cenário de Casas Inteligentes consiste no uso de *smart meters* para medir e coletar dados de consumo de energia nas casas de uma cidade. Entretanto, como as informações geradas pelos dispositivos são altamente sensíveis, esses dados podem revelar informações sobre as atividades diárias e o comportamento dos moradores das casas. Portanto, é desejável que não seja revelada informações sobre como os dados de energia de uma casa são utilizados sem que antes sejam tomados os devidos cuidados para prover a privacidade (MOLINA-MARKHAM *et al.*, 2010).

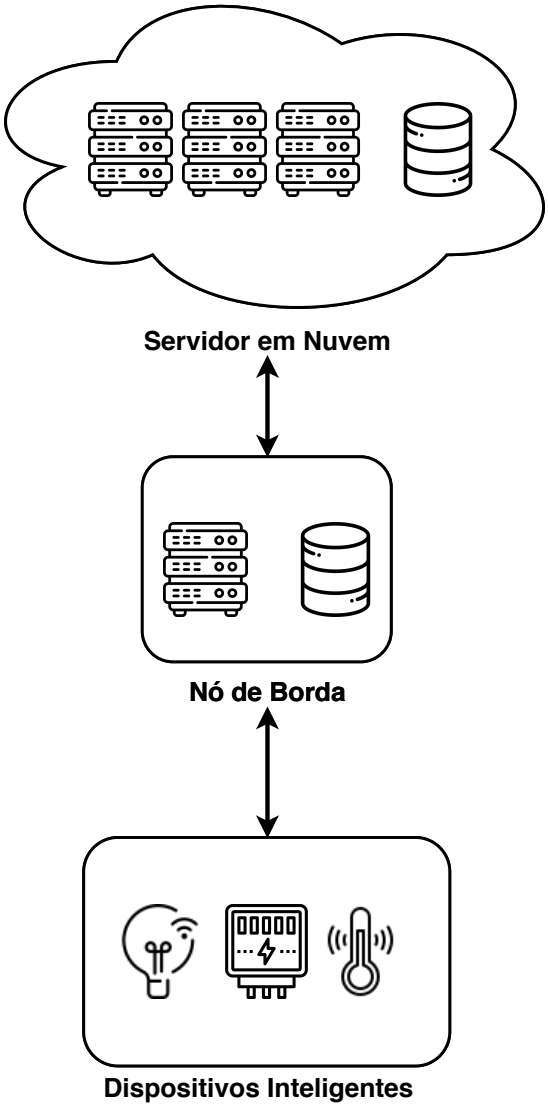
Computação em borda é um paradigma de computação que aproxima o processamento dos dados para onde eles são necessários, ou seja, na borda da rede. Em outras palavras, ao invés de processar os dados em um servidor de nuvem, todo processamento, ou parte dele, é feito localmente, na borda (Shi *et al.*, 2016). Esse paradigma surgiu com o potencial de lidar com problemas relacionados ao tempo de resposta, tempo de bateria dos dispositivos embarcados, largura de banda de rede, segurança e privacidade (Shi *et al.*, 2016). Observe que quase todos esses conceitos estão intrinsecamente relacionados à infraestrutura de rede, exceto a privacidade, que está diretamente relacionada aos usuário e constitui o foco desta dissertação. Em cenários de Casas Inteligentes, a computação em borda pode ser vista como um utensílio eletrônico que coleta e processa dados gerados por todos os dispositivos inteligentes conectados dentro de uma casa.

Um exemplo de como a computação em borda pode ser utilizada em um cenário de casa inteligente é apresentado na Figura 2, onde uma casa possui três dispositivos inteligentes instalados: uma lâmpada, um termômetro e um medidor. O utensílio eletrônico responsável pela computação em borda aparece na figura como Nó de Borda. Quando em funcionamento, um dispositivo inteligente pode ter a capacidade de produzir informação, como por exemplo o termômetro é responsável por medir a temperatura de um cômodo e pode enviar esses dados para

o Nó de Borda, que por sua vez pode processar esses dados como, por exemplo, para medir a temperatura média da casa e, com base em regras definidas pelos usuários, enviar de volta um sinal para o termômetro para que ele aqueça ou esfrie o cômodo. Além disso, o Nó de Borda pode se comunicar com um servidor em nuvem para, por exemplo, fazer *backup* dos dados.

Com relação ao medidor inteligente, também apresentado na Figura 2, o Nó de Borda pode ser responsável tanto por calcular o valor final da conta de energia da casa com base em uma política de preços gerada pelo fornecedor de energia, que também pode ser visto como o servidor em nuvem, quanto também pode ser responsável por gerar gráficos de consumo de energia com base nos dados gerados pelo medidor. Esses gráficos, por sua vez, podem ser consumidos pelos usuários da casa através de um *smartphone* que se comunica diretamente ao Nó de Borda.

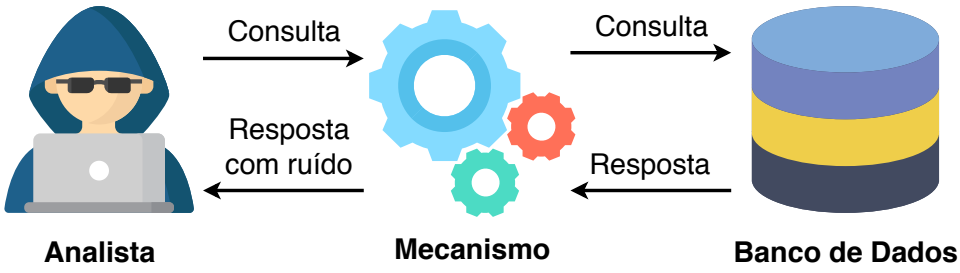
Figura 2 – Exemplo de computação em borda em uma casa inteligente.



2.2 Privacidade Diferencial

A Privacidade Diferencial (PD) é um modelo matemático proposto por Dwork (DWORK, 2006) que resultou em grande interesse acadêmico devido a suas fortes garantias de privacidade. Inicialmente, a PD foi pensada para funcionar como um *middleware* entre analistas e uma base de dados, respondendo de maneira privada a consultas executadas sobre essa base de dados (DWORK *et al.*, 2006). Neste trabalho, essa configuração será chamada de Privacidade Diferencial Centralizada, por depender de uma entidade centralizadora confiável com acesso a uma base de dados, sendo essa entidade responsável por prover a privacidade. O fluxo de funcionamento da Privacidade Diferencial Centralizada é ilustrado na Figura 3.

Figura 3 – Fluxo da Privacidade Diferencial Centralizada.



Um mecanismo aleatório $\mathcal{M}$ é dito diferencialmente privado se a probabilidade de qualquer saída desse mecanismo não variar muito, dado um limite de privacidade ε, independente da presença ou ausência de qualquer indivíduo na base de dados (DWORK; ROTH, 2014). Em outras palavras, a adição ou remoção de qualquer indivíduo de uma base de dados consultada utilizando um mecanismo diferencialmente privado, não deve afetar substancialmente o resultado de qualquer análise estatística executada sobre esse base. A PD é definida a seguir.

Definição 1. Um algoritmo aleatório, ou mecanismo, $\mathcal{M}$, garante a ε-Privacidade Diferencial (ε-PD) se, para quaisquer conjuntos de dados D_1 e D_2, diferindo em, no máximo, um indivíduo, e para todo $S \subseteq \text{Imagem}(\mathcal{M})$, ou seja, S está contido no conjunto de todos os resultados possíveis de $\mathcal{M}$, tem-se que:

$$\frac{Pr[\mathcal{M}(D_1) \in S]}{Pr[\mathcal{M}(D_2) \in S]} \leq e^{\varepsilon},$$

onde a probabilidade é dada sobre a aleatoriedade de $\mathcal{M}$.

Um importante aspecto da Privacidade Diferencial, tanto em sua configuração centralizada, apresentada anteriormente, quanto na configuração local, que será apresentada a seguir, na Seção 2.4, é que ela tem como objetivo principal garantir a privacidade de indivíduos. Entretanto, existem propostas baseadas em privacidade diferencial que procuram garantir a privacidade de grupos de indivíduos. Outro ponto importante, é que a quantidade de ruído adicionado por um mecanismo diferencialmente privado é dependente da consulta. Além disso, esse ruído é adicionado segundo uma distribuição de probabilidade de forma a buscar um resultado não tão distante da resposta real, ao mesmo tempo em que garante a privacidade. A perturbação por meio de ruído pode ser aplicada ao resultado da consulta, como mostrado na Figura 3, ou ainda pode ser aplicada sobre todo o banco de dados quando se quer publicar todo o conjunto de dados ao invés de responder a perguntas ou consultas específicas. A estratégia de publicação do banco de dados ruidoso, cujo ruído é adicionado por meio de um mecanismo diferencialmente privado, tende a gerar altos índices de perturbação no dado original, a fim de garantir a privacidade dos indivíduos presentes no conjunto de dados.

Existem várias formas de atender a definição da PD (DWORK, 2008), ou seja, garantir que as saídas de um mecanismo são computacionalmente indistinguíveis para conjuntos de dados vizinhos, portanto, que diferem em no máximo um indivíduo, como apresentado na Definição 1. Frequentemente, a forma de garantir a PD é baseada na adição de ruído às respostas de consultas, como ilustrado na Figura 3. Uma característica de mecanismos diferencialmente privados é que a garantia proporcionada não é dependente do poder computacional de um possível adversário nem da quantidade de informação externa que esse adversário pode ter, o que faz com que seja uma definição bastante sólida quando comparada com outros modelos de privacidade.

Um dos mecanismos mais conhecidos na literatura, o mecanismo de Laplace (DWORK, 2008) , é geralmente utilizado na resposta de consultas numéricas e baseia-se na adição de um ruído, obtido através de uma amostra da distribuição de Laplace, à resposta de uma consulta. Para que esse ruído seja suficiente para garantir a Definição 1, é necessário que a distribuição de Laplace tenha algumas características específicas, sendo elas: possuir média zero e escala $\frac{\Delta f}{\varepsilon}$, onde ε é o limite de privacidade e Δf consiste na sensibilidade da consulta. A sensibilidade, por sua vez, pode ser vista como o maior impacto que um indivíduo é capaz de ter sobre o resultado de uma consulta f. A seguir, são apresentadas as definições formais do mecanismo de Laplace, da sensibilidade e também da distribuição de Laplace.

Definição 2. (Sensibilidade) Seja χ o universo de todos os possíveis registros da base de dados. Seja $f : \mathbb{N}^{|\chi|} \rightarrow \mathbb{R}^k$ uma função que mapeia conjuntos de dados em uma resposta real de dimensão k, a sensibilidade de f é dada por:

$$\Delta f = max_{x,y \in \mathbb{N}^{|\chi|}} \parallel f(x) - f(y) \parallel_1$$

para quaisquer conjuntos de dados vizinhos x, y, ou seja, que diferem em no máximo um elemento (DWORK, 2008). Portanto, a sensibilidade é máxima distância, dada pela Norma $L1$, entre dois conjuntos de dados vizinhos quaisquer. Lembre-se que a Norma $L1$ é dada pela soma da diferença dos componentes de um vetor, logo, essa definição abrange funções de consultas multidimensionais.

A sensibilidade é necessária para que seja adicionado ruído suficiente para proteger o impacto de qualquer possível indivíduo no resultado de uma consulta e, quanto maior a sensibilidade Δf da consulta, mais ruído é necessário ser adicionado para que o resultado atenda a definição da PD (Domingo-Ferrer *et al.*, 2016). Note que o conceito de um indivíduo está presente na definição de sensibilidade ao definir conjuntos de dados vizinhos.

Definição 3. (Mecanismo de Laplace) Seja χ o universo de todos os possíveis registros da base de dados. Seja $f : \mathbb{N}^{|\chi|} \rightarrow \mathbb{R}^k$ uma função que mapeia conjuntos de dados em uma resposta real de dimensão k. Seja, ainda, $x \in \mathbb{N}^{|\chi|}$ um conjunto de dados, o mecanismo de Laplace é definido como:

$$\mathcal{M}_{\mathcal{L}}(x, f(\cdot), \varepsilon) = f(x) + (Y_1, \ldots, Y_k)$$

onde Y_i são amostras aleatórias i.i.d obtidas da distribuição de Laplace com média zero e escala $\frac{\Delta f}{\varepsilon}$.

Definição 4. (Distribuição de Laplace) A distribuição de Laplace com média zero e escala b é dada pela função densidade de probabilidade a seguir:

$$Lap(x|b) = \frac{1}{2b} exp(-\frac{x}{b})$$

Um outro mecanismo bastante utilizado na literatura é o mecanismo exponencial (DWORK; ROTH, 2014). Diferentemente do mecanismo de Laplace, que responde de maneira diferencialmente privada a consultas com respostas reais, o mecanismo exponencial é um método de dar respostas diferencialmente privadas para consultas que selecionam valores de um conjunto

discreto de saídas possíveis. Um exemplo desse tipo de consulta é: "Qual o nome mais popular na cidade de Fortaleza no ano de 2020?". É importante notar que o mecanismo de Laplace pode ser utilizado para responder consultas desse tipo ao adicionar ruído sobre as quantidades computadas. No exemplo da consulta anteriormente apresentada, o mecanismo de Laplace poderia ser utilizado para adicionar ruído às contagens dos nomes e, em seguida, o nome com o maior valor ruidoso seria retornado. Entretanto, essa estratégia gera uma quantidade de ruído proibitivamente grande.

Com o objetivo de responder consultas do tipo "melhor" resposta de maneira a preservar a utilidade do resultado das consultas, surge o mecanismo exponencial, cuja intuição é retornar qualquer saída possível $r \in \mathcal{R}$ com uma probabilidade que leve em consideração um *score* de utilidade de r ao mesmo tempo em que garante a definição da PD. A definição de função de utilidade que é utilizada pelo mecanismo exponencial é apresentada a seguir.

Definição 5. (Função de Utilidade) Dado um intervalo arbitrário $\mathcal{R}$, a função de utilidade $u : \mathbb{N}^{|\chi|} \times \mathcal{R} \to \mathbb{R}$ mapeia pares de conjuntos de dados ($\mathbb{N}^{|\chi|}$) e saídas($\mathcal{R}$) a um *score* de utilidade.

A sensibilidade da função de utilidade, por sua vez, é definida de maneira ligeiramente diferente da sensibilidade apresentada anteriormente na definição 2, e sua definição é dada por:

$$\Delta u = \max_{r \in \mathcal{R}} \max_{x,y:\|x-y\|_1 \leq 1} |u(x,r) - u(y,r)|$$

Ou seja, a sensibilidade da função de utilidade Δu mensura a máxima diferença de utilidade de respostas em conjuntos de dados vizinhos. A definição formal do mecanismo exponencial é apresentada a seguir.

Definição 6. (Mecanismo Exponencial) Seja $x \in \mathbb{N}^{|\chi|}$ um conjunto de dados, u uma função de utilidade e $\mathcal{R}$ um universo de saídas possíveis, o mecanismo exponencial, dado por $\mathcal{M}_{\mathcal{E}}(x, u, \mathcal{R})$, seleciona e retorna um elemento $r \in \mathcal{R}$ com uma probabilidade proporcional a $\exp(\frac{\varepsilon u(x,r)}{2\Delta u})$.

A PD possui três propriedades muito importantes, sendo elas: a composição sequencial, composição paralela e imunidade ao pós-processamento. As três propriedades são apresentadas a seguir.

- **Composição sequencial:** quando aplica-se uma sequência de mecanismos ε_i-diferencialmente privados, o limite de privacidade sofre uma degradação de maneira controlada, resultando em um limite de $\sum_i \varepsilon_i$ (DWORK; ROTH, 2014).

- **Composição paralela:** um caso especial da composição de mecanismos ocorre quando a sequência de mecanismos é aplicada em conjuntos de dados disjuntos entre si. Neste caso, não há a degradação do limite de privacidade, resultando em um limite de privacidade de $\max_i \varepsilon_i$ para essa sequência de aplicações de mecanismos (MCSHERRY, 2010).
- **Imunidade ao pós-processamento:** a saída de um mecanismo diferencialmente privado é imune a pós-processamento, ou seja, um adversário sem conhecimento adicional sobre a base de dados não é capaz de tornar a saída de um mecanismo menos privada, independente de seu poder computacional ou de qualquer informação auxiliar externa que esse atacante possa vir a ter (DWORK; ROTH, 2014).

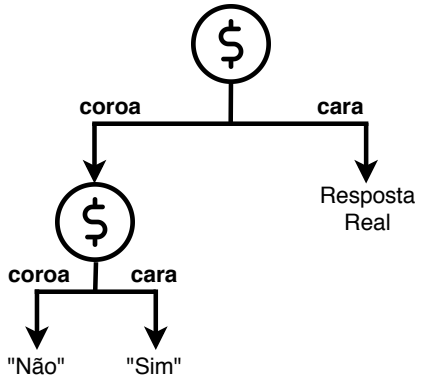
2.3 Resposta Randômica

A técnica de Resposta Randômica (RR) surgiu no contexto de pesquisas estatísticas, ou *survey*, em inglês, desenvolvidas nas ciências sociais, com intuito de fazer com que as pessoas sintam-se inclinadas a responder às pesquisas mesmo quando existem perguntas constrangedoras, ou até mesmo relacionadas a comportamentos ilegais. A ideia da RR é adicionar certa aleatoriedade às respostas, de modo que os participantes da pesquisa tenham a segurança da negação plausível, do inglês *plausible deniability*, no caso, por exemplo, de uma resposta incriminatória (WARNER, 1965). Como exemplo, imagine o cenário em que uma empresa, ou órgão do governo, deseja fazer uma pesquisa para estimar quantas pessoas em uma determinada região fizeram uso de uma droga ilegal no último mês. Essa, possivelmente, seria uma pesquisa com baixa adesão e, para as pessoas que se propusessem a responder, é provável que respondessem incorretamente, com o intuito de se proteger, o que faria com que o resultado da pesquisa fosse arbitrariamente tendencioso. Entretanto, ao utilizar um mecanismo aleatório para responder tal pergunta, tanto é mais provável que essa pesquisa tenha mais adesão, quanto os participantes não precisem mentir para se proteger, uma vez que os participantes têm a negação plausível sobre ter feito uso de uma droga ilegal, devido ao mecanismo utilizado. Além disso, ao utilizar esse mecanismo, é possível remover o viés (*bias*) das respostas, o que não seria possível caso um número não previsível de participantes respondesse incorretamente à pergunta, como aconteceria caso a RR não fosse utilizada.

Suponha que a pesquisa tenha o objetivo de estimar quantas pessoas possuem uma certa propriedade P. Os participantes respondem à pergunta sobre possuir, ou não, a propriedade P da seguinte forma:

1. Jogue uma moeda;
2. Caso o resultado seja **cara**, responda a verdade sobre possuir, ou não, P;
3. Caso o resultado seja **coroa**, então jogue uma nova moeda e responta “Sim”, caso obtenha cara, e “Não”, caso obtenha coroa.

Figura 4 – Procedimento executado na técnica de RR.



O procedimento executado na RR é ilustrado na Figura 4. As probabilidades de obter cara ou coroa nas moedas podem ser definidas arbitrariamente, mas, para facilitar o entendimento, vamos considerar uma moeda justa, onde cada opção ocorre com probabilidade $\frac{1}{2}$. Perceba que, caso a propriedade P seja equivalente a uma atividade criminosa, como exemplificado anteriormente, até mesmo uma resposta “Sim” não é incriminatória, uma vez que esse mecanismo resultará em uma resposta “Sim” com probabilidade $\frac{1}{4}$, independente de o participante possuir, ou não, a propriedade P.

Como dito anteriormente, ao utilizar esse mecanismo aleatório, é possível remover o viés das respostas, uma vez que as probabilidades desse mecanismo são conhecidas. Seja p a fração real de pessoas que possuem a propriedade P. Perceba que o número esperado de respostas do tipo “Sim” é igual a $(1-p)\cdot(\frac{1}{4}) + p\cdot(\frac{1}{2}) + p\cdot(\frac{1}{4})$, onde o primeiro termo da soma contabiliza as pessoas que possuem P mas responderam “Não”, o segundo termo refere-se às pessoas que possuem a propriedade e responderam “Sim”, enquanto a terceira considera os participantes que possuem P mas responderam “Não”. Simplificando os três termos tem-se que o número esperado de respostas “Sim” é igual a $\frac{1}{4} + \frac{p}{2}$. Portanto, pode-se estimar p como duas vezes o número de respostas do tipo “Sim”, menos $\frac{1}{2}$, ou seja, $\hat{p} = 2\cdot(\frac{1}{4} + \frac{p}{2}) - \frac{1}{2}$ (DWORK; ROTH, 2014).

Com isso, pode-se perceber que a estratégia de RR pode ser utilizada para prover a negação plausível para resposta à perguntas constrangedoras ou a respeito de comportamentos ilegais, enquanto permite que seja obtida uma estimativa das respostas reais à essas perguntas. A

prova que essa estimativa é não tendenciosa pode ser encontrada em (WARNER, 1965).

Apesar da técnica de RR ter sido proposta em um contexto diferente, posteriormente foi mostrado que essa técnica atende à definição da PD para um limite de privacidade $\varepsilon = ln(3)$ (DWORK; ROTH, 2014). Caso as probabilidades da moeda sejam diferentes de $\frac{1}{2}$, esse limite de privacidade será alterado, mas ainda garantirá a PD para algum *budget* definido em função dessas probabilidades.

2.4 Privacidade Diferencial Local

Uma definição semelhante ao que posteriormente viria a ser a Privacidade Diferencial Local (PDL) foi proposta com o nome de amplificação (EVFIMIEVSKI *et al.*, 2003), tratando-se, também, de um processo de randomização executado no cliente com o objetivo de limitar a violação de privacidade da saída randomizada. Subsequentemente, surge a menção direta à PDL em (DUCHI *et al.*, 2013), como uma configuração especial da PD onde não é necessário que exista uma entidade externa confiável responsável por prover a privacidade para os dados de um conjunto de usuários, ao mesmo tempo em que provê as fortes garantias da privacidade diferencial.

Apesar de a PD ter despertado grande interesse acadêmico, foi a PDL que provocou o interesse também da indústria. Grandes empresas fizeram proposições e implementaram protocolos para garantir a privacidade para seus usuários através da PDL como, por exemplo, a Google, que propôs uma forma de coletar informações de seu navegador de maneira privada (ERLINGSSON *et al.*, 2014). Outro exemplo é a empresa Apple, que utiliza a PDL para aprender sobre o uso de *emojis* e palavras digitadas em seus dispositivos (GREENBERG, 2016). A definição de PDL é apresentada a seguir.

Definição 7. Dado um domínio de valores $\mathcal{D}$, um algoritmo aleatório, ou mecanismo, $\mathcal{M}$, garante ε-Privacidade Diferencial Local (ε-PDL) se, para todos os pares de valores $v_1, v_2 \in \mathcal{D}$, e para todo $S \subseteq \text{Imagem}(\mathcal{M})$, tem-se que:

$$\frac{Pr[\mathcal{M}(v_1) \in S]}{Pr[\mathcal{M}(v_2) \in S]} \leq e^{\varepsilon},$$

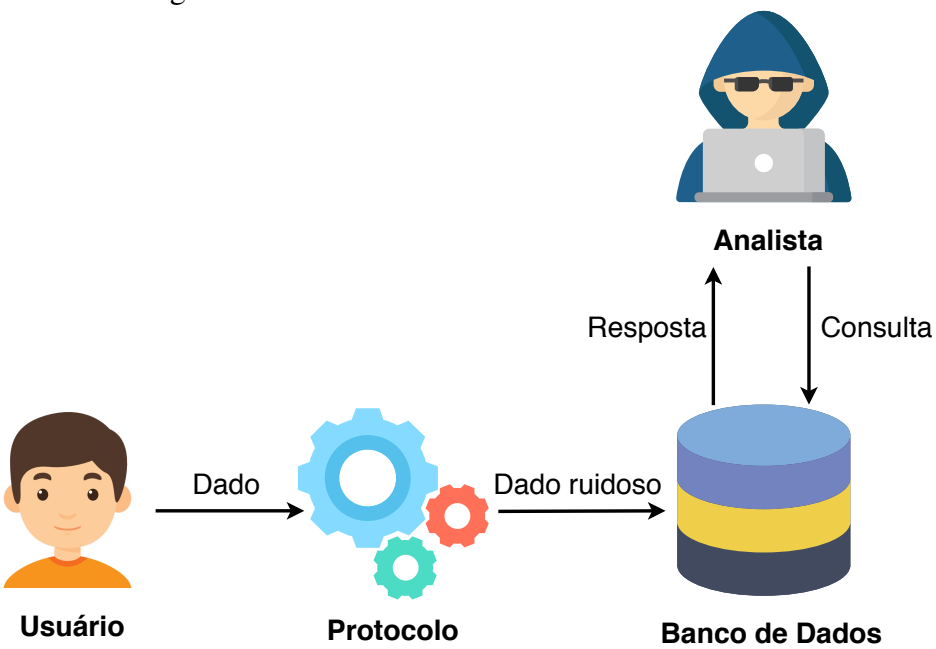
onde a probabilidade é dada sobre a aleatoriedade de $\mathcal{M}$.

Note que a PDL é um caso particular da PD em que as entradas de $\mathcal{M}$ são valores, ao

invés de conjuntos de dados. Logo, entradas vizinhas são definidas como dois valores quaisquer do domínio $\mathcal{D}$, diferentemente de conjuntos de dados diferindo em no máximo um indivíduo, como ocorre na PD. Uma característica da PDL é que essa fornece uma garantia de privacidade ainda mais forte que a PD, uma vez que não depende de uma entidade confiável externa para prover a privacidade. Em contrapartida, na PD, uma menor adição de ruído se faz necessária para garantir o mesmo limite de privacidade ε.

Um dos problemas principais na PDL trata-se de protocolos para estimar valores de um domínio $\mathcal{D}$ de maneira privada (CORMODE *et al.*, 2018). Esses protocolos são chamados de Oráculo de Frequência (OF) e o procedimento executado nesses codifica inicialmente a informação privada, conhecida como resposta, em um formato específico e, em seguida randomiza esse valor codificado para obter uma saída que garante a definição da PDL. Em seguida, essa saída é enviada para uma entidade agregadora, que performa uma ação para agregar e decodificar as respostas obtidas, com o objetivo de inferir estatísticas sobre a população. Para essa inferência, a entidade agregadora estima quantos usuários possuem cada um dos valores de resposta que são de interesse. O processo de randomização executado por esses protocolos pode ter como base a técnica de RR, apresentada previamente na Seção 2.3. A Figura 5 ilustra o fluxo de um protocolo que atende a PDL. Perceba que o Analista apresentado nessa figura faz o papel da entidade agregadora descrita anteriormente.

Figura 5 – Fluxo da Privacidade Diferencial Local.



Como supracitado, muitos dos protocolos baseiam-se na ideia da RR para prover a

privacidade. Entretanto, note que a RR, conforme apresentada na Seção 2.3, aplica-se diretamente apenas para domínios binários, sendo necessário modificar levemente esse protocolo para estimar frequência de valores de um domínio $\mathcal{D}$, quando $|\mathcal{D}| > 2$. Um OF básico, chamado de Codificação Direta (CD), estende essa definição, como especificado a seguir:

Definição 8. Na Codificação Direta (CD), uma saída aleatória y de uma resposta v é gerada com as seguintes probabilidades:

$$\forall_{y \in \mathcal{D}} Pr[\text{CD}(v) = y] = \begin{cases} \frac{e^{\varepsilon}}{e^{\varepsilon}+|\mathcal{D}|-1}, \text{ se } y = v \\ \frac{1}{e^{\varepsilon}+|\mathcal{D}|-1}, \text{ se } y \neq v \end{cases}$$

Ou seja, o valor real da resposta v, tem uma probabilidade maior de ser retornado, enquanto todos os outros elementos ocorrem com uma probabilidade uniforme e menor. Note que quanto maior a probabilidade de v ser retornado, melhor será a acurácia. Entretanto, essa probabilidade depende do tamanho do domínio, $|\mathcal{D}|$. Buscando propor um protocolo cuja acurácia não se degrada com o aumento do tamanho do domínio, os autores de (ERLINGSSON *et al.*, 2014) propuseram a *Unary Encoding* / Codificação Unária (UE), que será definida e discutida no Capítulo 4.

2.5 Conclusão

Neste capítulo foram apresentados os principais conceitos relacionados a *Internet of Things* / Internet das Coisas (IoT), a computação em borda e como esses conceitos se relacionam com ambientes inteligentes, como casas e cidades. Foi discutida, ainda, a enorme geração de dados proveniente desses ambientes inteligentes e como esses dados podem vazar informações íntimas de seus usuários se o devido cuidado com a privacidade não por tomado. Posteriormente, o modelo da Privacidade Diferencial (PD) foi apresentado, tanto na sua configuração centralizada quanto na local, além de serem mencionadas as suas principais propriedades, ou seja, a composição sequencial, a composição paralela e a imunidade a pós-processamento. Além disso, foram expostos os principais mecanismos da PD centralizada, sendo eles o mecanismo de Laplace e o mecanismo exponencial. Finalmente, foram apresentadas a técnica de Resposta Randômica (RR) que, apesar de ter proposta em um contexto diferente, mostrou-se que garante a PDL, e o mais básico OF, chamado de Codificação Direta (CD).

3 TRABALHOS RELACIONADOS

Este capítulo apresenta alguns trabalhos relevantes para a área de privacidade de dados, que focam tanto no contexto específico de casas inteligentes, quanto no problema mais genérico de garantir a privacidade no contexto de dados que são gerados em *streams*. As soluções de privacidade apresentadas a seguir consistem, em sua maioria, em aplicações do conceito da PD e da PDL.

O restante do capítulo apresenta os trabalhos agrupados em três principais categorias, com base nos problemas que propõem resolver. As categorias são: privacidade em casas inteligentes, privacidade em *streaming* de dados e Privacidade Diferencial Local. Perceba que as três categorias não são disjuntas entre si, visto que dados de Casas Inteligentes, por exemplo, podem ser gerados em *streaming* de dados. Assim como pode-se propor soluções baseadas em PDL para problemas em diversos contextos, incluindo *streaming* de dados de casas inteligentes.

3.1 Privacidade em Casas Inteligentes

Dados gerados no contexto de casas inteligentes têm como importante característica o teor altamente sensível. Isso ocorre pois esses dados são gerados por dispositivos situados dentro das casas das pessoas, podendo carregar muitas informações sensíveis que podem passar despercebidas, como será apresentado a seguir, na Seção 3.1.1.

Um importante exemplo real de dados gerados nesse contexto são dados de consumo de energia, gerados por medidores inteligentes (*smart meters*). Por conta da relevância desse tipo de dados, gerados com base em consumo de um recurso instalado no interior de uma residência, existem trabalhos que propõem soluções para duas classes de problemas. A primeira classe refere-se ao problema de tarifação e cobrança, de maneira privada, por esse consumo. Por sua vez, a segunda classe aborda o problema de extrair informação desses dados, também de maneira privada, sem necessariamente se preocupar com a possível cobrança acarretada pelo consumo. A seguir, apresentamos uma solução proposta para cada um desses problemas.

3.1.1 Private Memoirs of a Smart Meter

Proposto em (MOLINA-MARKHAM *et al.*, 2010), o trabalho apresenta uma solução para o primeiro problema apresentado, isto é, como cobrar de maneira privada pelo consumo de um recurso. Primeiramente, para justificar a importância e a necessidade da privacidade na

coleta de dados de consumo de energia, os autores fazem uma análise estatística sobre dados coletados nesse contexto. Para que essa análise fosse possível, foram registrados dados de consumo de energia de três diferentes casas por um período de dois meses. Cada casa registrava o consumo de energia agregado a cada segundo. O fato de o registro do consumo ser agregado, significa dizer que não há a diferenciação, por exemplo, de qual eletrodoméstico foi responsável por determinado consumo de energia. Um dado de consumo trata-se de uma tupla de energia (t, c), consistindo de um *timestamp* t e o consumo médio de energia c, em kilowatt, do último segundo. Esses dados são, em um primeiro momento, armazenados localmente em cada uma das residências. Em seguida, a cada hora, esses dados são enviados para um repositório central, onde será efetuada a análise.

A análise divide-se em quatro partes. Primeiramente, os dados de energia são pré-processados usando um algoritmo de agrupamento, com o intuito de identificar e rotular eventos de energia similares entre si. Um conjunto de tuplas de energia pertencentes a um mesmo grupo é chamado de segmento de energia, ou *power trace*, em inglês. Em seguida, esses segmentos de energia são adicionados de alguns atributos identificadores, como por exemplo a duração do segmento. O resultado dessa fase é uma 6-tupla representando um segmento, que inclui o rótulo, o tempo de início, o consumo médio, a duração, o início da variação de energia e um rótulo para a forma do segmento de energia. A variação de energia refere-se ao aumento ou diminuição de energia que ocorre no início de um segmento. O passo seguinte de análise consiste em filtrar segmentos de energia que correspondem a eletrodomésticos automatizados, como geladeiras, aquecedores e ar-condicionados, uma vez que o objetivo final é identificar atividade humana através dos dados. O último passo da análise consiste em mapear eventos da vida real nos segmentos de energia, utilizando conhecimento externo.

Com a aplicação dessas simples técnicas estatísticas, os autores mostraram que, de posse dos dados de consumo de granulosidade fina, é possível responder a perguntas que ferem a privacidade dos indivíduos dessas casas. Diferentes informações sensíveis podem ser reveladas com acesso a dados de diferentes granulosidades, que podem variar de uma frequência de segundos a horas. A Tabela 1 apresenta algumas das perguntas privadas, os padrões de energia que podem revelar suas respostas e os níveis de granularidade necessários para identificar esses padrões de maneira precisa.

Tendo em vista a quantidade de informações privadas que podem ser reveladas com acesso a dados de granulosidade fina de consumo de energia, e com a necessidade de cobrar por

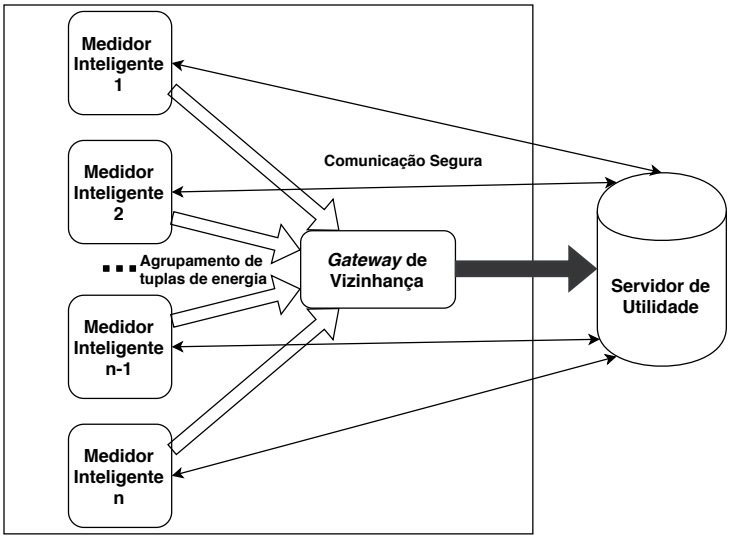
Tabela 1 – Perguntas privadas que podem ser reveladas com dados de consumo de energia de granulosidade fina.

Pergunta	Padrão	Granulosidade
Você estava em casa durante sua licença médica?	Sim: Atividades de energia durante o dia Não: Baixo uso de energia durante o dia	Hora / Minuto
Você teve uma boa noite de sono?	Sim: Sem eventos de energia durante a noite, por pelo menos 6 horas Não: Eventos aleatórios de energia durante a noite	Hora / Minuto
Você assistiu ao jogo na noite passada?	Sim: Evento de energia correspondente ao programa na TV Não: Sem eventos de energia correspondente ao horário do jogo	Minuto / Segundo
Você saiu atrasado para o trabalho?	Sim: Último evento de energia ocorrendo após o tempo estimado de deslocamento Não: Último evento de energia com tempo suficiente para o transporte	Minuto
Você deixou seu filho sozinho em casa?	Sim: Padrão de atividade de uma única pessoa Não: Eventos simultâneos de energia em áreas distintas da casa	Minuto / Segundo
Você toma café da manhã quente ou frio?	Sim: Picos de energia durante a manhã (micro-ondas, máquina de café) Não: Sem eventos correspondentes a utensílios utilizados para um café da manhã quente	Segundo

Fonte: (MOLINA-MARKHAM *et al.*, 2010)

esse consumo, os autores propõem uma arquitetura para cenários onde o consumo de energia ocorre através de medidores inteligentes (*smart meters*), com o objetivo de prover a privacidade. Essa arquitetura baseia-se em três componentes: Medidores Inteligentes, *Gateway* de Vizinhança e Servidor Remoto de Utilidade (MOLINA-MARKHAM *et al.*, 2010). A arquitetura pode ser visualizada na Figura 6.

Figura 6 – Arquitetura proposta.



Nessa arquitetura, cada Medidor Inteligente instalado em um residência tem a função de armazenar localmente tuplas de consumo de energia (r,t,c), que consistem em um identificador aleatório r, um *timestamp* t, e o consumo de energia c correspondente ao período de tempo desde a última tupla. Em seguida, o Medidor Inteligente envia essas tuplas para o *Gateway* de Vizinhança, que, por sua vez, tem a responsabilidade de remover o identificador r e redirecionar as tuplas de um conjunto de usuários para Servidores de Utilidade. As tuplas (t,c), sem o campo de identificação, que são redirecionadas para o Servidor de Utilidade, são chamadas de *blinded tuples*. As comunicações entre os componentes dessa arquitetura ocorrem através de

canais seguros. Neste cenário, a privacidade se dá pelo Servidor de Utilidade não ter acesso à identificação real das tuplas de energia.

Para cobrar pelo consumo de energia, o Servidor de Utilidade periodicamente comunica-se diretamente com cada um dos Medidores Inteligentes através de um protocolo de conhecimento-nulo. Um protocolo de conhecimento-nulo, definido em (FEIGE *et al.*, 1988), é um método de criptografia que consiste em permitir que um **provador** (Medidor Inteligente) demonstre conhecimento a respeito de um **segredo** (Tuplas de Energia) para um **verificador** (Servidor de Utilidade) sem a necessidade de revelar nenhuma informação que ajudaria o **verificador** a inferir o segredo. Através da utilização desse protocolo de conhecimento-nulo, o Servidor de Utilidade tem a garantia de que os valores coletados para pagamento por cada Medidor Inteligente correspondem ao consumo ocorrido em sua respectiva residência por um período de tempo, sem a necessidade de ter conhecimento de quais tuplas de energia estão associadas a cada Medidor Inteligente.

Vale lembrar que constantemente o *Gateway* de Vizinhança agrupa tuplas de um conjunto de usuários e redireciona as *blinded tuples* para o Servidor de Utilidade. Por outro lado, a cobrança pelo consumo de energia acontece em ciclos, através do protocolo de conhecimento-nulo proposto que é composto por três fases em cada ciclo de cobrança: registro, coleta de tuplas e reconciliação. Na fase de registro, os Medidores Inteligentes se comprometem criptograficamente a um conjunto de N *tags* pseudoaleatórias $\{r_i\}$ e um conjunto de chaves $\{k_1, \ldots, k_m\}$, onde N é o número de tuplas de energia necessárias para computar o consumo no período de cobrança estipulado, e m é a quantidade de rodadas do protocolo de verificação. Se comprometer criptograficamente significa que uma entidade externa, como o Servidor de Utilidade, não é capaz de identificar o conjunto de *tags* escolhidos por um de seus clientes. Uma quantidade maior de rodadas do protocolo, definido por m, provê uma maior segurança contra uma possível tentativa de fraudar o sistema por parte do usuário, por exemplo ao tentar utilizar tuplas falsas para um período de cobrança tentando diminuir o valor a pagar. Entretanto, os autores alegam que na prática, um valor pequeno de m, por exemplo um $m = 10$ é suficiente para prover uma garantia adequada. Na segunda fase do protocolo, a coleta de tuplas, os Medidores Inteligentes criam tuplas (r_i, t_i, c_i), onde r_i é uma *tag* pseudoaleatória do conjunto comprometido anteriormente na fase de registro, t_i é o *timestamp* da tupla, e c_i é consumo reportado pelo Medidor Inteligente no *timestamp* t_i. Um ponto importante é que o *Gateway* de Vizinhança não revela ao Servidor de Utilidade a origem das tuplas com um dado identificador pseudoaleatório r_i. Na terceira e última

fase, o cliente computa o valor total de sua conta utilizando uma política de preços dinâmica como $E = \sum_{i=1}^{N} Cost(t_i, c_i)$, usando as tuplas (r_i, t_i, c_i) e, então, envia o total E para o Servidor de Utilidade. Para provar que E está correto, o cliente e o servidor executam, então, m rodadas de verificação. O Medidor Inteligente consegue passar por cada rodada se o valor E tiver sido calculado utilizando as $tags$ e chaves comprometidas durante a fase de registro.

3.1.2 I have a DREAM! (DiffeRentially privatE smArt Metering)

O trabalho (ÁCS; CASTELLUCCIA, 2011), por sua vez, utiliza-se da Privacidade Diferencial (PD) para propor uma solução para o problema de aprender, de maneira privada, com os dados gerados pelo consumo de energia de um recurso. Esse trabalho, assim como o apresentado anteriormente, foca em dados de energia gerados através de *smart meters*. A informação de interesse para os autores se trata de soma de consumo de energia de uma quantidade N de casas (nós) em um dado tempo. A solução proposta baseia-se na adição distribuída de ruído, de modo a garantir a definição da PD.

O modelo proposto possui duas principais entidades, os usuários (ou nós) e o agregador. Cada usuário possui um medidor inteligente, que mede o consumo de energia a cada período de tempo T_p e envia uma versão privada desse consumo para o agregador. Seja X_t^i o consumo do usuário i no tempo t, o agregador está interessado em obter, em cada tempo t, a soma dos consumos X_t de um número N de usuários, isto é: $X_t = \sum_{i=1}^{N} X_t^i$. A estratégia proposta baseia-se em dois passos principais: geração distribuída do ruído de Laplace e criptografia de dados de medições de consumo individuais.

O primeiro passo, a geração distribuída do ruído de Laplace, utiliza-se do conceito de uma distribuição de probabilidade infinitamente divisível, que significa que é possível representar tal distribuição como a distribuição de probabilidade da soma de um número arbitrário de independentes e identicamente distribuídas (i.i.d) variáveis aleatórias (FINETTI, 1929). Esse passo baseia-se, então, no fato de que a distribuição de Laplace, aqui representada por $\mathcal{L}$, uma distribuição que deu origem a um dos mecanismos mais conhecidos por garantir a PD, é infinitamente divisível e pode ser construída através da soma de distribuições Gama (Γ) i.i.d. O procedimento de construção distribuída do ruído de Laplace, portanto, consiste em cada usuário i calcular o valor $\hat{X}_t^i = X_t^i + \Gamma_1(N, \lambda) - \Gamma_2(N, \lambda)$ em um dado tempo t, onde $\Gamma_1(N, \lambda)$ e $\Gamma_2(N, \lambda)$ são dois valores aleatórios selecionados independentemente da mesma distribuição Gama. Lembre-se que N é o número de usuários, X_t^i é valor de consumo do usuário i no tempo t, e λ refere-se à

escala da distribuição. Com isso, quando o agregador somar todas os consumos recebidos em um tempo t, ele obterá $\sum_{i=1}^{N} \hat{X}_t^i = \sum_{i=1}^{N} X_t^i + \sum_{i=1}^{N} [\Gamma_1(N,\lambda) - \Gamma_2(N,\lambda)] = X_t + \mathcal{L}(\lambda)$. A prova formal da equivalência da soma da diferença entre as distribuições Gama e a distribuição de Laplace pode ser encontrada em (ÁCS; CASTELLUCCIA, 2011). Atente que, para que a soma garanta a propriedade da PD, a distribuição de Laplace deve ter escala $\frac{\Delta f}{\varepsilon}$, onde Δf é a sensibilidade e ε é o *budget*, ou limite de privacidade. Portanto, $\lambda = \frac{\Delta f}{\varepsilon}$ garante que a soma de consumo de um tempo t atende a definição da PD.

Note que o procedimento apresentado anteriormente garante que a soma dos valores obtidos pelo agregador em um tempo t atende a PD. Lembre-se que os valores recebidos correspondem ao consumo somado de um ruído gerado conforme definido anteriormente. Entretanto, como dito anteriormente, os consumos individuais $\hat{X}_t^i$, enviados pelo *smart meter* i no tempo t, possuem um ruído que não é suficiente para garantir essa propriedade. É necessário, portanto, garantir que o agregador não tenha acesso individual aos valores de $\hat{X}_t^i$, mas apenas à soma desses valores. Para que isso ocorra, cada usuário criptografa o valor $\hat{X}_t^i$ de uma maneira que o agregador não é capaz de descriptografar uma medida individual, mas somente a soma das medidas, que, por sua vez, garante a PD.

Note que, conforme apresentado, o procedimento depende da participação de N usuários para garantir a privacidade. Para aumentar a robustez do esquema proposto, os autores propõem, ainda, uma forma de permitir que uma quantidade de até M nós possam falhar sem comprometer a privacidade. Para isso, cada usuário adiciona um quantidade de ruído maior que o que seria estritamente necessário.

3.2 Privacidade em *Streaming* de Dados

Uma característica dos dados de IoT, onde também se encaixam os dados gerados no contexto de casas inteligentes, é que frequentemente esses dados ocorrem como *streams*. Trabalhos que lidam com o problema de garantir a privacidade no contexto de *streaming* de dados têm que lidar com complexidades adicionais. Isso ocorre porque esses dados são potencialmente infinitos e gerados continuamente a velocidades rápidas. A propriedade de dados potencialmente infinitos é, provavelmente, a característica que mais impõe dificuldades para a aplicação da privacidade, pois muitas das soluções propostas são limitadas a um conjunto imutável de dados. Outras soluções, baseadas em PD, costumeiramente utilizam o *budget* ε para um número limitado de consultas ou de dados, o que pode tornar inviável sua aplicação direta nesse contexto.

Trabalhos que focam na privacidade de *streaming* de dados muitas vezes propõem simplificações ao problema para conseguir chegar a uma solução. A seguir, serão apresentadas três diferentes trabalhos que propõem soluções para dados de *stream*. O primeiro deles, (CAO; YOSHIKAWA, 2015), foca, especificamente, em *streaming* de dados de trajetória, enquanto o segundo, (CHEN *et al.*, 2017), propõe uma solução para responder à consultas de contagem em *streaming* de dados de estados, enquanto o trabalho (LEAL *et al.*, 2018), apresenta uma solução para *streaming* de dados numéricos. As três soluções utilizam a PD centralizada como solução e cada uma impõe certas limitações ao problema para propor suas soluções. Cada trabalho será detalhado nas seções a seguir.

3.2.1 Differentially Private Real-time Data Release over Infinite Trajectory Streams

O problema abordado em (CAO; YOSHIKAWA, 2015) consiste em publicar estatísticas de *streaming* de trajetórias em tempo real. Os dados brutos consistem em tuplas com o formato (u_i, t_j, l_k), que indicam que o usuário u_i encontra-se na localização l_k no tempo t_j. Um exemplo de um conjunto de dados pode ser visto na Tabela 2. Uma sequência de tuplas constituem uma trajetória. A informação desejável para publicação é a quantidade de pessoas em uma determinada localização em um instante de tempo t_j. A Tabela 3 mostra a representação em trajetórias dos dados extraídos da Tabela 2, enquanto a Tabela 4 apresenta as estatísticas que este trabalho visa publicar sobre os dados, também obtidas através dos dados da Tabela 2.

Tabela 2 – Dados brutos.

uID	**tempo**	**localização**
u_1	t_1	cantina
u_3	t_1	cantina
u_2	t_2	bar
u_3	t_2	escritório
u_3	t_3	academia
u_1	t_5	bar
u_2	t_5	cantina
u_3	t_5	bar
...	...	...

Tabela 3 – Dados brutos representados como trajetórias.

	t_1	t_2	t_3	t_4	t_5	...
u_1	cantina				bar	...
u_2		bar			cantina	...
u_3	cantina	escritório	academia		bar	...

Um cenário que exemplifica o problema de privacidade que ocorre ao publicar as informações sobre as localizações é: um atacante possui o conhecimento de que o usuário

Tabela 4 – Estatísticas desejáveis sobre as localizações.

	t_1	t_2	t_3	t_4	t_5	...
cantina	2	0	0	0	1	...
escritório	0	1	0	0	0	...
bar	0	1	0	0	2	...
academia	0	0	1	0	0	...

u_3 encontrava-se em algumas localizações nos tempos t_1, t_3 e t_5. Esse atacante não possuía o conhecimento de quais eram essas localizações. Entretanto, ao observar a Tabela 4, esse atacante seria capaz de inferir a trajetória do usuário u_3 como *cantina* $\rightarrow$ *academia* $\rightarrow$ *bar* com uma probabilidade de $\frac{2}{3}$. Dessa forma, para evitar os riscos para a privacidade acarretados pela publicação desse tipo de estatística, os autores apresentam um modelo baseado na PD.

Como afirmado anteriormente, o fato de os dados serem gerados infinitamente traz dificuldades adicionais para proteção da privacidade. Os autores afirmam que prover a privacidade a nível de usuário no contexto de trajetórias é praticamente impossível, pois suas trajetórias são infinitas, o que inviabilizaria a publicação de informações sobre as mesmas. Para contornar esse problema, a solução sugerida propõe o conceito de privacidade a nível de l-trajetórias, que consistem em uma sequência de l tuplas (u_i, t_j, l_k) de um usuário u_i. Note que uma trajetória de tamanho l não necessariamente ocorre em l *timestamps* consecutivos.

Para garantir a PD, os autores utilizam o mecanismo de Laplace. As soluções propostas baseiam-se em dividir o *budget* ε para garantir a privacidade de uma l-trajetória. Lembre-se que o objetivo deste trabalho é publicar estatísticas sobre localizações de interesse ao longo do tempo, como exemplificado da Tabela 4, ou seja, em cada instante de tempo t, deseja-se publicar um vetor r_t contendo as contagens das localizações. De maneira formal, define-se como uma consulta de contagem $Q^c : \mathcal{D}_i \rightarrow \mathbb{R}^{|locs|}$.

A ideia é que, ao invés de publicar a quantidade real de pessoas em uma dada localização em um tempo t_j, publica-se esse valor somado de um ruído obtido de uma amostra da distribuição de Laplace com escala $\frac{\Delta f}{\varepsilon_j}$, onde o valor do *budget* ε é dividido ao longo do tempo, formando ε_j, de modo a garantir a privacidade de l-trajetórias. Para definirmos a sensibilidade Δf nesse contexto, é necessário primeiramente definir o que são *datasets* vizinhos em um *timestamp* t (Definição 9). Lembre-se, ainda, que um *dataset* consiste em um conjunto de tuplas (u_i, t_j, l_k), onde u_i refere-se ao identificador do usuário.

Definição 9. (*Datasets* Vizinhos em um *timestamp* t): Se dois *datasets* $\mathcal{D}_i$ e $\mathcal{D}'_i$ são coletados em um tempo $i \in [1, t]$, e diferem em uma única localização l_k de um usuário u_i, então $\mathcal{D}_i$ e $\mathcal{D}'_i$

são vizinhos com respeito a u_i.

Como um usuário aparece em no máximo uma localização em cada instante de tempo t_j, modificar uma localização l_k impactará em, no máximo 2 no resultado de Q^c, resultando em uma sensibilidade de 2.

Para entender a relação de uma l-trajetória e a PD, é necessário definir prefixos vizinhos de *streaming* de trajetórias (Definição 10).

Definição 10. (Prefixos Vizinhos de *Streaming* de Trajetórias): Um prefixo de uma *streaming* de trajetória corresponde a todos os dados de uma *streaming* de trajetória infinita até o *timestamp* corrente t. Sejam dois prefixos $S_t = \{\mathcal{D}_1, \ldots, \mathcal{D}_t\}$ e $S'_t = \{\mathcal{D}'_1, \ldots, \mathcal{D}'_t\}$. S_t e S'_t são ditos vizinhos se um é obtido a partir do outro ao modificar-se todas as localizações (locs) de uma l-trajetória $l_{u,k}$. É dito, então, que S_t e S'_t são vizinhos com relação a $l_{u,k}$.

Com isso, define-se uma l-trajetória ε-PD como a seguir (Definição 11).

Definição 11. Seja $\mathcal{M}$ um algoritmo que tem como entrada um prefixo de *streaming* de trajetória $S_t = \{\mathcal{D}_1, \ldots, \mathcal{D}_t\}$ e seja $N_t = \{n_1, \ldots, n_t\}$ uma saída qualquer de $\mathcal{M}$. É dito que $\mathcal{M}$ garante a l-trajetória ε-PD se para quaisquer prefixos vizinhos S_t e S'_t:

$$Pr[\mathcal{M}(S_t) = N_t] \leq \varepsilon * Pr[\mathcal{M}(S'_t) = N_t]$$

Uma das estratégias apresentadas neste trabalho para garantir a l-trajetória ε-PD consiste em uniformemente alocar o *budget* $\varepsilon_j = \frac{\varepsilon}{l}$ em cada instante de tempo t_j. Portanto, para a contagem de cada localização, é adicionado um ruído n gerado através de uma amostra da distribuição de Laplace com escala $\frac{\Delta f}{\varepsilon_i}$. Como definimos anteriormente, tem-se que $\Delta f = 2$, portanto $n = Lap(\frac{2l}{\varepsilon})$. Os autores propõem, ainda, uma diferente estratégia que visa melhorar a utilidade, onde ao invés de dividir o *budget* de maneira uniforme, utiliza-se um decaimento exponencial, ou seja, aloca-se para um tempo t um *budget* $\varepsilon_t = \frac{\varepsilon_{t-1}}{2}$.

Além disso, os autores apresentam duas estratégias de reutilização de valores previamente publicados, também tendo como objetivo estatísticas mais próximas das reais, ou seja, com mais utilidade. A primeira estratégia parte do pressuposto que é provável que, em uma *streaming*, as contagens em um dado instante de tempo sejam semelhantes às contagens publicadas em um instante de tempo adjacente. Portanto, essa estratégia divide o *budget* ε_t alocado para um instante de tempo t entre $\varepsilon_{t,1}$ e $\varepsilon_{t,2}$, onde $\varepsilon_{t,1}$ será utilizado para decidir entre gerar um novo ruído (usando $\varepsilon_{t,2}$) sobre as contagens reais desse tempo, ou reutilizar os valores

publicados no *timestamp* anterior. Para isso, é calculado Erro Médio Absoluto (EMA), e, entre o vetor de contagens, r_t, e o vetor publicado no tempo anterior n_{t-1}, ou seja, $e = EMA(r_t, n_{t-1})$, e adicionado a esse valor um ruído r obtido através da distribuição de Laplace, utilizando o *budget* ε_1, ou seja, $r = Lap(\frac{2}{|locs|*\varepsilon_{t,1}})$. Esse ruído é necessário pois, para que essa decisão seja tomada, é preciso ter acesso ao valor real de r_t, portanto, isso deve ser feito de maneira privada. A decisão é tomada como a seguir: se $e + r \leq \frac{2}{|locs|*\varepsilon_{t,2}}$, então publica-se novamente o vetor r_{t-1}. Caso contrário, publica-se $r_t + Lap(\frac{2}{|locs|*\varepsilon_{t,2}})$.

A segunda estratégia de reutilização de estatísticas publicadas tem como objetivo atacar cenários onde as estatísticas de trajetórias possuem um certo padrão de repetição. Por exemplo, o fluxo máximo de trajetórias deve ocorrer às 8 horas da manhã e, posteriormente, às 17 horas. Nesse caso, é mais interessante uma estratégia que permita a reutilização de estatísticas publicadas em um diferente *timestamp*, que não o imediatamente adjacente. Para o desenvolvimento dessa estratégia os autores utilizam o mecanismo exponencial para obter o vetor n_i de menor distância para o vetor r_t de maneira privada e, posteriormente, é tomada uma decisão semelhante à apresentada para a estratégia anterior, para escolher entre reutilizar n_i ou gerar novos ruídos para publicar as contagens de r_t.

Este trabalho apresenta, ainda, uma análise comparativa entre os diferentes algoritmos propostos, isto é, utilizando alocação uniforme ou com decaimento exponencial, e as diferentes estratégias de reutilização, sendo elas a adjacente e a que considera a menor distância. Para os diferentes conjuntos de dados utilizados, a estratégia de alocação uniforme possui sempre um erro EMA superior às outras, enquanto a estratégia de alocação de *budget* com decaimento exponencial, juntamente com a realocação utilizando a menor distância (Mecanismo Exponencial) obteve sempre menor erro.

3.2.2 PeGaSus: Data-Adaptive Differentially Private Stream Processing

O trabalho proposto em (CHEN *et al.*, 2017), por sua vez, trata do problema de publicar, em tempo real, resultados privados de consultas de contagem em dados de *stream*. Para resolver esse problema, os autores propõem o *framework Perturb-Group-Smooth* / Perturbar-Agrupar-Suavizar (PeGaSus).

As consultas de contagem abordadas nesse trabalho se resumem em duas categorias: alvo único e alvos múltiplos. Além disso, as consultas possuem, também, uma das subcategorias a seguir: Unitária, Janela Deslizante e Monitoramento de Eventos. Para melhor explicar cada um

desses termos, utilizaremos o mesmo cenário desenvolvido em (CHEN *et al.*, 2017) para ilustrar a aplicação do PeGaSus, que consiste em uma análise de dados de *Access Point* Wifi / Ponto de Acesso Wifi (AP) onde um analista está interessado em obter a informação, por exemplo, de quantas pessoas estão conectadas a um AP em um dado instante de tempo.

O termo alvo único, refere-se ao fato de que uma consulta analisa um único alvo, no exemplo acima o alvo trata-se do AP. Enquanto consultas de alvos múltiplos analisam um conjunto de alvos ao mesmo tempo, como por exemplo, os AP de uma casa ou do auditório de um campus. Consultas desse tipo podem ser feitas sobre uma agregação das consultas de alvos únicos ou, ainda, como um conjunto de múltiplas consultas unitárias.

Contagem unitária, por sua vez, trata-se de a consulta ser direcionada à um único instante de tempo, enquanto janela deslizante, por sua vez, indica que a consulta é efetuada sobre um intervalo de tempo. Pode-se responder consultas de uma janela deslizante através da soma de consultas unitárias. Já uma consulta de monitoramento de evento é aquela que busca identificar mudanças de comportamento em uma *streaming* de dados, por exemplo ao observar um aumento ou declínio na contagem de uma sequência de janelas deslizantes.

Perceba que, apesar de cada uma dessas consultas ser capaz de efetuar uma análise ligeiramente diferente da outra, todas são baseadas em consultas de contagem. Para que essas consultas possam ser respondidas de maneira privada, o PeGaSus utiliza-se do conceito da PD. Vale ressaltar que a abordagem deste trabalho é a da PD não local, significando que existe uma entidade centralizadora com acesso aos dados brutos dos usuários e que é responsável por garantir sua privacidade. Além disso, a garantia de privacidade oferecida é correspondente a *w-event*, o que significa que a privacidade dos usuários é garantida para uma janela de tempo de tamanho w. As três fases do *framework* são detalhadas a seguir.

3.2.2.1 Perturb

Essa fase tem como entrada a resposta real à consulta de contagem no tempo t, dado por c_t, e o limite de privacidade ε_p, e tem como saída a resposta diferencialmente privada, dada por $\tilde{c}_t$. Esta fase utiliza o Mecanismo de Laplace para garantir a PD e consiste em adicionar ruído à resposta real da consulta de contagem para um dado tempo t. Por se tratar de uma aplicação direta do Mecanismo de Laplace sobre o resultado de uma consulta de contagem, a sensibilidade Δf é 1. O *budget* de privacidade consumido nessa fase é de ε_p.

3.2.2.2 *Group*

A fase de agrupamento também tem como entrada a resposta real à consulta de contagem, assim como a fase anterior, além de receber também o *budget* ε_g e um limiar θ, que será explicado a seguir. O particionamento, ou agrupamento, é feito tendo como objetivo minimizar o desvio médio das partições de maneira aproximada. Vale ressaltar que esse desvio, dado por $dev(G)$, mede a diferença absoluta entre um grupo G de contagens e a média entre elas. O motivo dessa minimização ser aproximada, como pode-se supor, é para que esse processo atenda à PD. A saída dessa fase em um dado tempo t é o particionamento P_t.

O procedimento de agrupamento consiste em, durante a primeira execução do algoritmo, no tempo t_1, criar um novo grupo contendo apenas a primeira resposta real c_1, ou seja $P_1 = \{\{c_1\}\}$. Em cada tempo seguinte, atualiza-se o valor do limiar para $\tilde{\theta} = \theta + Lap(\frac{4}{\varepsilon_g})$. Em seguida, calcula-se $desvio = dev(G \cup c_t) + Lap(\frac{8}{\varepsilon_g})$ e compara-se com esse limiar $\tilde{\theta}$. Caso $desvio < \tilde{\theta}$, adiciona-se a contagem c_t ao grupo G, caso contrário, cria-se um novo grupo para c_t. Esse processo baseia-se na Técnica do Vetor Esparso (DWORK; YEKHANIN, 2008), cuja prova mostra que, quando utilizada com os parâmetros conforme apresentados anteriormente, a adição de ruído tanto sobre o limiar $\tilde{\theta}$ quanto sobre o desvio do grupo garantem a PD para um limite ε_g.

3.2.2.3 *Smooth*

O processo de suavização, por sua vez, recebe como entrada as saídas tanto do *Perturb* quanto do *Group* e, então, combina seus resultados com o objetivo de obter uma resposta para a consulta de contagem que possui um menor erro. Note que, uma vez que este processo utiliza apenas dados que já atendem à PD, isso se trata de um pós-processamento e, portanto, não ocasiona em um consumo adicional de *budget*. O limite de privacidade do PeGaSus, em um dado tempo t é, então, $\varepsilon = \varepsilon_p + \varepsilon_g$.

Para o processo de suavização, os autores propõem três diferentes algoritmos, onde cada um deles processa de maneira diferente as consultas adicionadas de ruído, dada por $\tilde{c}$, presentes no grupo G, onde G é o último grupo gerado na partição P_t. As alternativas para o *Smoother* são:

1. *AverageSmoother*: Utiliza a média das contagens adicionadas de ruído presentes no grupo G para estimar a contagem do dado atual.

$$\hat{c_t} = \frac{\sum_{i \in G} \tilde{c}_i}{|G|}$$

2. *MedianSmoother*: Utiliza a mediana das contagens adicionadas de ruído presentes no grupo G para estimar a contagem do dado atual.

$$\hat{c_t} = mediana\{\tilde{c_i} | i \in G\}$$

3. *JSSmoother*: Aplica o estimador de James-Stein (STEIN, 1956) para atualizar a estimativa da contagem do dado atual utilizando os valores de $\tilde{c_i} \in G$.

$$\hat{c_t} = \frac{\tilde{c_t} - avg}{|G|} + avg$$

É importante notar que tanto o *Perturb* quanto o *Group* são sempre executados da mesma forma, enquanto o *Smooth* deve ser modificado para que seja especializado para cada uma das consultas apresentadas anteriormente. Um ponto importante com relação ao PeGaSus é sua capacidade de responder simultaneamente a cada uma das diferentes consultas sem que haja um consumo adicional de *budget*, uma vez que somente o *Smooth* é executado múltiplas vezes para que isso ocorra.

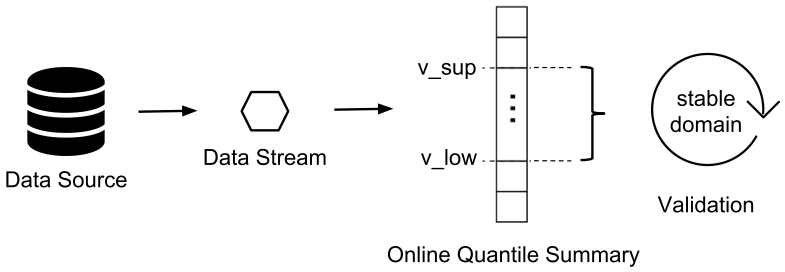
Para avaliar o PeGaSus, os autores utilizaram dados reais de APs e chegaram à conclusão de que além de apresentarem um *framework* capaz de responder múltiplas consultas de contagem, foram capazes de obter resultados com um erro menor do que o estado da arte. Além disso, concluíram ainda que, entre as três alternativas de *Smoother*, o *MedianSmoother* foi o que obteve melhores resultados sob quaisquer configurações de seus experimentos.

3.2.3 δ-DOCA: Achieving Privacy in Data Streams

O trabalho (LEAL *et al.*, 2018), por sua vez, trata do problema de prover a privacidade no contexto de *streaming* de dados de números reais. A estratégia proposta também se utiliza do mecanismo de Laplace para prover a garantia da PD. Um dos pontos de foco principal desse trabalho é com relação à estimativa do valor da sensibilidade Δf da *streaming* de dados. Diferentemente de outros trabalhos, que assumem que Δf é previamente conhecido, os autores apresentam uma estratégia baseada no cálculo *online* dos quantis de uma *streaming* de dados para obter uma aproximação da sensibilidade. Lembre que a sensibilidade é um parâmetro de alguns mecanismos diferencialmente privados que, em alto nível, serve para mensurar o máximo impacto de um indivíduo numa resposta de uma consulta. Note, ainda, que a solução proposta nesse trabalho tem como objetivo a publicação diferencialmente privada de uma *streaming* de dados numéricos, diferentemente do cenário iterativo, onde adiciona-se ruído ao resultado de

uma consulta para que a mesma garanta a PD. Portanto, uma vez que a solução proposta baseia-se em adicionar ruído diretamente sobre o dado, calcular Δf é equivalente a conhecer o domínio desses dados.

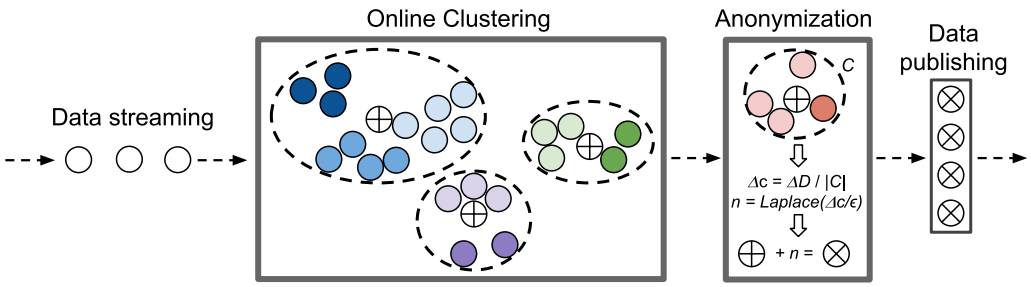
Figura 7 – δ-DOCA estágio 1 – Limitação de Domínio por δ.



Fonte: (LEAL *et al.*, 2018)

O passo de cálculo aproximado do domínio da *streaming* de dados é ilustrado na Figura 7. Note que esse passo trata-se de uma fase de pré-processamento onde, por algum tempo, no início da execução, não se publica nenhuma informação sobre os dados recebidos, até que tenha-se um conhecimento aproximado do domínio dos dados.

Figura 8 – δ-DOCA estágio 2 – Agrupamento *online* e adição de ruído.



Fonte: (LEAL *et al.*, 2018)

Soluções que baseiam-se em adicionar o ruído diretamente sobre os dados, como é o caso desse trabalho, possuem o problema de obter uma pior utilidade, devido a grande quantidade de ruído adicionado. Para contornar esse problema, os autores propõem uma estratégia de micro-agregação, que consiste em um agrupamento *online* seguido da substituição de todos os elementos de um grupo C pelo valor do seu centroide, conforme ilustrado na Figura 8. Em seguida, é adicionado um ruído a cada elemento de C, obtido através de uma distribuição de Laplace com média zero e escala $\frac{\Delta c}{\varepsilon}$, onde $\Delta c = \frac{\Delta f}{|C|}$. Veja que a utilização de Δc no lugar de Δf faz com que a sensibilidade seja reduzida proporcionalmente ao tamanho do grupo C. Portanto, quanto maior o tamanho de C, menor será o ruído adicionado.

Perceba que essa solução deve levar em consideração dois tipos de erros que serão adicionados pelo processo. Por erro, entenda como uma medida da distância do valor gerado por um processo com relação ao seu valor original. O primeiro erro ocorre devido à substituição dos elementos de um grupo pelo seu centroide, fazendo com que a escolha errada desses grupos impacte negativamente na utilidade. Já o segundo erro provém do ruído adicionado que, por sua vez, depende do tamanho dos grupos formados. Existe, portanto, um claro *trade-off* entre a obtenção de grupos mais representativos e a formação de grupos maiores, para obter um resultado com melhor utilidade.

3.3 Privacidade Diferencial Local

Como os dados gerados em casas inteligentes contém informações altamente confidenciais sobre as pessoas que vivem nessas casas, nem sempre são aceitáveis soluções que dependem de uma entidade externa confiável para prover a privacidade desses dados, como as apresentadas anteriormente. Com isso, surge o interesse na Privacidade Diferencial Local (PDL), que além da garantia formal dada pela PD, tem a vantagem de ser independente da confiança em uma entidade externa. Isso ocorre pois, na PDL, o processo de randomização deve ser executado localmente, nos equipamentos de coleta de dados associados diretamente aos usuários.

A seguir, serão apresentados dois trabalhos que propõem soluções baseadas na PDL. O primeiro, (ERLINGSSON *et al.*, 2014), propõe uma estratégia que faz uso de dois processos de randomização para prover a privacidade mesmo quando infinitas respostas privadas são enviadas. O segundo trabalho, faz uma comparação entre diferentes protocolos da PDL e, por fim, apresenta parâmetros que, para um mesmo limite de privacidade ε, minimizam o ruído adicionado.

3.3.1 RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response

Proposto em (ERLINGSSON *et al.*, 2014), o trabalho foca no problema de coletar estatísticas sobre *strings* geradas por clientes, como numa arquitetura cliente/servidor. A solução proposta utiliza dois processos de randomização que possuem objetivos distintos.

O processo consiste em um cliente que deseja enviar uma resposta para o servidor como, por exemplo, o nome da página inicial configurada no navegador do cliente. Suponha que a resposta real do cliente seja r = "*www.mdcc.ufc.br*". Primeiramente, o cliente codifica a resposta em uma estrutura de *Bloom Filter*, gerando uma representação da resposta como um

vetor de bits $B = [1,0,1,\ldots,0,1,0]$, onde $|B| = k$, sendo B gerado utilizando um número h de funções *hash*. É gerado um vetor de bits B para cada resposta distinta enviada. O uso da estrutura de *Bloom Filter*, embora seja de uso opcional no modelo proposto, é justificado pela capacidade de obter respostas mais compactas.

No passo seguinte, B é randomizado por meio de um processo chamado de Randomização Permanente, que gera o vetor de bits B' da seguinte forma:

$$B'[i] = \begin{cases} 1, & \text{com probabilidade } \frac{f}{2} \\ 0, & \text{com probabilidade } \frac{f}{2} \\ B[i], & \text{com probabilidade } 1-f \end{cases}$$

Onde f é um parâmetro de entrada que define o nível de privacidade obtido por esse procedimento. Os autores apresentam a prova formal de que esse passo de Randomização Permanente garante a PDL para um limite de privacidade $\varepsilon_\infty = 2h \cdot ln\left(\frac{1-\frac{1}{2}f}{\frac{1}{2}f}\right)$, onde h é o número de funções *hash* utilizadas na criação do *Bloom filter*, como apresentado anteriormente.

Essa representação de bits randomizada B' é, então, armazenada pelo cliente e reutilizada sempre que a mesma resposta r for reenviada. Em seguida, antes de enviar a resposta para o servidor, B' passa por outro processo de randomização, chamado de Randomização Instantânea. Esse processo recebe como entrada B' e tem como saída o vetor de bits S, que é gerado com as seguintes probabilidades:

$$Pr[S[i] = 1] = \begin{cases} q, & \text{se } B'[i] = 1 \\ p, & \text{se } B'[i] = 0 \end{cases}$$

A resposta S é, então, enviada para o servidor. Esse procedimento também garante a PDL para um limite de privacidade ε_1, que leva em consideração tanto a Randomização Permanente, quanto a Randomização Instantânea. Note que, uma vez que a Randomização Instantânea tem como entrada o resultado da Randomização Permanente, o limite de privacidade ε_1 é limitado por ε_∞.

A garantia dada por esse procedimento é de que, mesmo quando um grande número de respostas instantâneas são enviadas para uma mesma resposta real r, um atacante é capaz de identificar, no máximo, o vetor B' que deu origem às respostas instantâneas, não sendo capaz de identificar o *Bloom Filter* B de origem e, consequentemente, a resposta real r. O principal objetivo da Randomização Permanente é, portanto, garantir a privacidade longitudinal, i.e. ao

longo do tempo, enquanto a Randomização Instantânea garante que um vetor B' não se tornará identificador de um usuário.

3.3.2 Locally Differentially Private Protocols for Frequency Estimation

O trabalho (WANG *et al.*, 2017) apresenta um *framework* que generaliza muitos dos protocolos de PDL propostos na literatura, apresentando, também, uma análise desses protocolos que permite a escolha de parâmetros otimizados, resultando na proposta de dois novos protocolos. O trabalho foca no problema em que um usuário possui um valor v e reporta uma única vez para uma entidade agregadora. Portanto, este trabalho não foca diretamente no contexto de *streaming* de dados.

O *framework* proposto abrange protocolos que são "puros". Para definir um protocolo puro, é apresentada a definição de uma função de Suporte, que faz o mapeamento de toda possível saída y com o conjunto de valores de entrada que "suportam" y. Suponha que um valor v seja codificado em um vetor de bits B que possui o bit $B[v] = 1$ e todos os demais bits têm valor zero. Suponha um protocolo que randomiza esse vetor B como, por exemplo, o protocolo utilizado no passo de Randomização Instantânea do Rappor, apresentado na Seção 3.3.1, gerando como saída o vetor de bits randomizado B'. O vetor de bits de saída B' suporta os valores de entrada cujo bit de B' é definido como 1, ou seja, $Suporte(B') = \{i | B'[i] = 1\}$. Um protocolo puro é apresentado na Definição 12.

Definição 12. Um protocolo PE é puro se, e somente se, existem duas probabilidades $p^* > q^*$ tal que, para todo valor v_1,

$$Pr[PE(v_1) \in \{y | v_1 \in Suporte(y)\}] = p^*,$$
$$\forall_{v_2 \neq v_1} Pr[PE(v_2) \in \{y | v_1 \in Suporte(y)\}] = q^*$$

Para cada valor v_1, o conjunto $\{y | v_1 \in Suporte(y)\}$ representa todas as saídas y que suportam v_1 e pode ser definido como o conjunto suporte de v_1. Um protocolo puro requer que a probabilidade de um valor v_1 ser mapeado para seu próprio conjunto suporte seja a mesma para todos os valores (p^*). É necessário, ainda, que seja possível um valor $v_2 \neq v_1$ ser mapeado para o conjunto suporte de v_1 com uma probabilidade q^*.

Os autores seguem mostrando que existe uma fórmula fechada para calcular a variância de protocolos puros e, com isso, são capazes de analisar diferentes protocolos e comparar suas acurácias para um mesmo limite de privacidade ε, através de suas variâncias.

Por fim, são propostos dois novos protocolos, obtidos através da otimização dos parâmetros p e q de protocolos conhecidos na literatura. Para obter a versão otimizada de um protocolo, os autores propõem calcular sua variância, seguindo com o cálculo da derivada parcial com relação a probabilidade q e, então, igualar a zero. Ao resolver essa equação, a estratégia obtém os valores de probabilidade p e q do protocolo que minimizam a variância, ou seja, que resulta em menor erro para um dado limite de privacidade ε.

Os dois protocolos propostos são as versões otimizadas do *Unary Encoding* e do *Local Hashing*. A variância de ambos os protocolos é igual, o que significa que resultam na mesma utilidade. Entretanto, o *Optimized Unary Encoding (OUE)* possui a vantagem de ser mais rápido e simples de implementar, já que não se utiliza de funções *hash*. Por outro lado, o *Optimized Local Hashing (OLH)* possui menor custo de comunicação, $O(log(d))$, contra $O(d)$, que é o custo do OUE, onde d é o tamanho do vetor de saída (resposta) do protocolo.

3.4 Discussão

Os trabalhos apresentados neste capítulo propõem estratégias para solucionar o problema de privacidade tanto no contexto geral de *streaming* de dados, onde deve-se tratar a característica de geração infinita dos dados, quanto no contexto mais específico de casas inteligentes, que é caracterizado por dados altamente sensíveis. Os trabalhos apresentados que abordam o contexto de casas inteligentes focam em dois principais aspectos referentes à geração de dados de consumo de um recurso, sendo o primeiro deles a cobrança por esse consumo sem a necessidade do acesso aos dados com identificações de suas origens. Já o segundo aspecto aborda o problema de obter informações a partir de dados coletados mantendo a privacidade dos usuários.

O trabalho (MOLINA-MARKHAM *et al.*, 2010) apresenta uma solução baseada em criptografia, que permite a cobrança pelo consumo de um recurso através de dados não identificados. O trabalho foca na cobrança pelo consumo de energia, através de medidores inteligentes (*smart meters*). Uma importante limitação desse trabalho é que a entidade agregadora, chamada de Servidor de Utilidade, apesar de não ter acesso aos dados brutos de consumo, ainda tem acesso ao que os autores chamam de *blinded data*, que consiste no conjunto de tuplas de consumo de energia com os identificadores removidos. Entretanto, a remoção dos identificadores não é suficiente para prover a privacidade dos indivíduos em um conjunto de dados (SWEENEY, 2002), sendo, inclusive, um dos problemas que deram origem ao interesse pelo estudo da

privacidade de dados.

Já o trabalho (ÁCS; CASTELLUCCIA, 2011) apresenta uma solução diferencialmente privada capaz de obter a soma do consumo de energia de um conjunto de N casas. Os autores baseiam sua solução na adição distribuída do ruído de Laplace, que garante uma boa utilidade na obtenção da soma do consumo. Este trabalho não trata uma possível quebra de privacidade decorrente do envio de múltiplas respostas por um usuário ao longo do tempo. O ruído é adicionado de forma tal que, quando o agregador calcula a soma de consumo dos usuários em um tempo t, ou seja, $\sum_{i=1}^{N} \hat{X}_t^i$, obtém-se a soma dos consumos desse *timestamp*, adicionado de um ruído que segue a distribuição de Laplace, como a seguir: $\sum_{i=1}^{N} X_t^i + \mathcal{L}(\lambda)$, onde $\lambda = \frac{\Delta f}{\varepsilon}$. Acontece que a definição da sensibilidade (Δf) utilizada no trabalho é configurada como o consumo máximo no tempo t, ou seja $max_{1 \leq i \leq N} X_t^i$. Devido a isso, juntamente com o fato de que todo o *budget* é consumido a cada tempo t, a privacidade garantida por esse procedimento é a nível de medida de consumo por *timestamp*. Portanto, esse trabalho não considera que quando múltiplas medidas são enviadas ao longo do tempo, a privacidade dos indivíduos é degradada devido à propriedade da Composição Sequencial da PD.

A solução proposta em (CAO; YOSHIKAWA, 2015) para a privacidade na publicação em tempo real de estatísticas sobre *streaming* de trajetórias é, também, diferencialmente privada. O trabalho apresenta soluções que baseiam-se na simplificação do problema para proteger trajetórias de um tamanho definido l. Uma limitação desse trabalho é que a privacidade é garantida através de uma entidade confiável que tem acesso aos dados brutos de localização dos indivíduos, o que pode não ser aceitável em um contexto em que os dados são tão sensíveis como o que está inserido esse trabalho.

O trabalho (CHEN *et al.*, 2017) propõe uma solução para responder consultas de contagem em *streaming* de dados dependendo de uma entidade externa confiável para prover a privacidade. Os autores argumentam que sua solução enquadra-se no conceito de w-evento, onde a privacidade é definida para uma janela de tamanho w e que, caso essa janela tenha tamanho infinito, a privacidade pode ser limitada por $l * \varepsilon$, onde l define o número de tuplas de um usuário na *stream*, entretanto, para cenários onde l é infinito ou onde não se pode prever quão grande l pode ser, essa solução não é aceitável. Portanto, trata-se de uma abordagem interessante quanto à utilização de grupos para aumento da utilidade dos resultados, mas que não pode ser aplicada para cenários realistas, onde pode ser necessário que um usuário responda infinitas vezes às consultas.

A solução proposta em (LEAL *et al.*, 2018) também é dependente da confiança em uma entidade externa. Além disso, assim como o trabalho (CHEN *et al.*, 2017) discutido anteriormente, a solução proposta tem a limitação de não permitir o envio de múltiplos dados de um mesmo usuário, o que pode tornar a solução impraticável para cenários reais. Portanto, o trabalho insere-se no contexto de *streaming* de dados apenas no que diz respeito à complexidade dos algoritmos propostos, não abordando o problema da aplicação da PD ao longo do tempo para cenários realistas, ou seja, considerando que um mesmo usuário pode enviar respostas múltiplas vezes.

O trabalho (ERLINGSSON *et al.*, 2014) destaca-se como um dos poucos trabalhos na literatura que abordam o problema do envio de múltiplos dados privados por parte de um cliente para um servidor. A solução utiliza-se do conceito de *memoization* para garantir que um cliente seja capaz de enviar infinitas respostas sob a garantia da Privacidade Diferencial Local (PDL). A solução, portanto, não depende de uma entidade confiável, já que utiliza o conceito de PDL. A utilização de dois protocolos que garantem a PDL permitem a definição de diferentes limites de privacidade ε para as privacidades a curto e longo prazos. Um ponto fraco desse trabalho é que os autores não justificam a escolha dos protocolos utilizados, limitando-se a provar que os protocolos utilizados são diferencialmente privados.

O trabalho (WANG *et al.*, 2017) apresenta um *framework* para protocolos de PDL que são definidos como puros. O *framework* proposto permite a análise de protocolos existentes através de suas variâncias e, com isso, a proposição de dois novos protocolos que utilizam parâmetros otimizados com o objetivo de obter respostas que são mais próximas das reais. Uma limitação desse trabalho é que não aborda o problema de envio de múltiplas respostas por parte de um usuário.

A nossa contribuição, denominada ProTECting, apresenta duas estratégias para a privacidade de dados gerados no contexto de casas inteligentes. Devido à alta sensibilidade desses dados, a solução proposta utiliza-se da PDL para prover a privacidade com foco nos usuários, sem depender de uma entidade externa confiável. Além disso, ProTECting considera o aspecto infinito da geração de dados, apresentando, primeiramente, uma solução baseada em janela deslizante, que aborda o problema de uma forma simplificada. Em seguida, propomos uma diferente abordagem para o ProTECting, que utiliza-se da ideia de *memoization* e de dupla randomização proposta em (ERLINGSSON *et al.*, 2014) para prover a privacidade para infinitas respostas, por meio, ainda, de protocolos otimizados, visando obter uma melhor utilidade. Nós

propomos ProTECting com o objetivo de apresentar uma solução aplicável em cenários reais de casas inteligentes.

A Tabela 5 apresenta um resumo comparativo entre os trabalhos apresentados neste capítulo e as duas abordagens que compõem nossa contribuição, o ProTECting.

Tabela 5 – Análise comparativa entre os trabalhos relacionados.

Trabalho	**Modelo de Privacidade**	**Mecanismo**	**Independente de Entidade Confiável**	**Múltiplas Respostas**	**Aplicações de IoT**	**Protocolos Otimizados**
(MOLINA-MARKHAM *et al.*, 2010)	Criptografia	-	Sim	-	Sim	-
(ÁCS; CASTELLUCCIA, 2011)	PD	Laplace Distribuído	Sim	Não	Sim	-
(ERLINGSSON *et al.*, 2014)	PDL	Protocolos Locais	Sim	Sim	Não	Não
(CAO; YOSHIKAWA, 2015)	PD	Laplace	Não	Sim (limitado por l)	Não	-
(CHEN *et al.*, 2017)	PD	Laplace	Não	Sim (limitado por w)	Não	-
(WANG *et al.*, 2017)	PDL	Protocolos Locais	Sim	Não	Não	Sim
(LEAL *et al.*, 2018)	PD	Laplace	Não	Não	Não	-
ProTECting Janela	PDL	Protocolos Locais	Sim	Sim (limitado por w)	Sim	Sim
ProTECting Dupla Randomização	PDL	Protocolos Locais	Sim	Sim	Sim	Sim

3.5 Conclusão

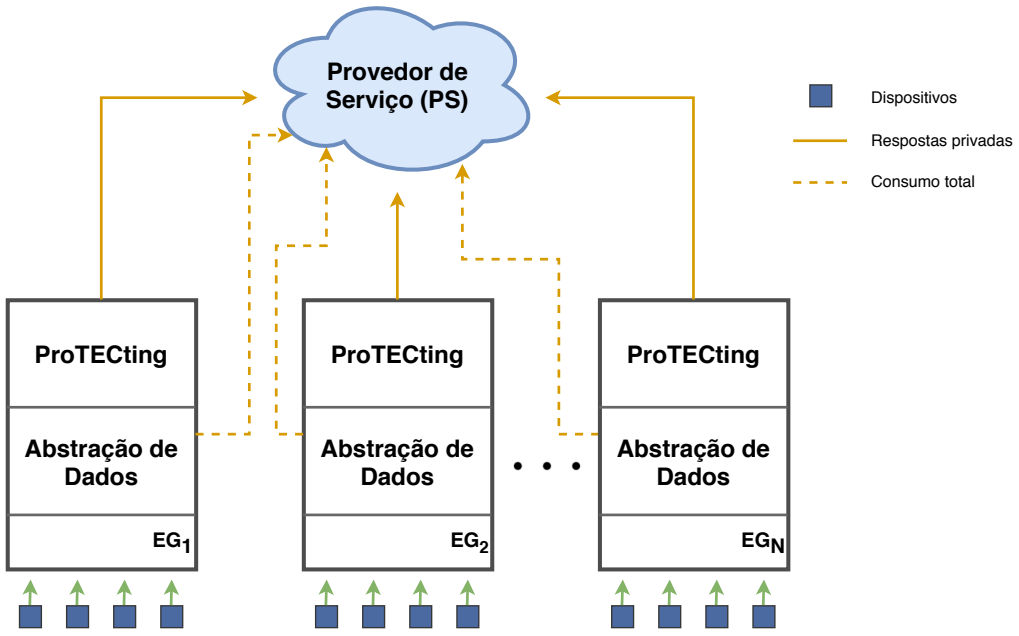
Neste capítulo foram apresentados os principais trabalhos relacionados ao tema desta dissertação. Os trabalhos foram classificados em três principais categorias não independentes entre si, sendo elas: privacidade em casas inteligentes, privacidade em *streaming* de dados e Privacidade Diferencial Local. Foram apresentadas as principais características de cada um dos trabalhos relacionados e, posteriormente, feita uma discussão sobre os mesmos. Por último, comparamos as soluções propostas nesta dissertação, que compõem o ProTECting, com esses trabalhos relacionados. A seguir propomos o ProTECting como uma solução para cenários reais de *streaming* de dados de casas inteligentes, tendo como objetivo a minimização do ruído adicionado no processo de garantir a PDL.

4 PROTECTING

O objetivo deste trabalho é garantir que uma entidade agregadora seja capaz de obter informações sobre dados gerados por dispositivos dentro de casas inteligentes de uma maneira privada. De agora em diante, considera-se que essa entidade agregadora é chamada de Provedor de Serviços (PS), tendo em vista que ela coleta dados com a intenção de melhor prover serviços. Num cenário real, a entidade agregadora e o provedor de serviços podem ser entidades distintas, sendo o primeiro simplesmente uma entidade que coleta dados mediante alguma recompensa para os moradores de casas inteligentes. Neste trabalho, entretanto, considera-se que o Provedor de Serviços (PS) é a entidade responsável tanto por prover serviços quanto por aprender com os dados gerados a partir das casas inteligentes.

Nossa estratégia, ProTECting, foi projetada para ser executada sobre um *edgeOS*, um sistema operacional especializado que roda sobre um *Edge Gateway* / *Gateway* de Borda (EG), na borda da rede, no ambiente de uma casa inteligente. No contexto de casas inteligentes, o EG consiste em uma unidade especializada que gerencia dispositivos inteligentes. Sobre a arquitetura do *edgeOS*, é importante ressaltar que existe uma camada de abstração de dados, responsável por reunir os dados gerados pelos dispositivos inteligentes de uma casa (Shi *et al.*, 2016).

Figura 9 – Comunicação entre o PS e as casas, através de seus EG.



ProTECting trabalha entre a camada de abstração de dados do *edgeOS* e toda comu-

nicação externa à casa, tendo como objetivo garantir que toda informação sobre dados gerados na casa, e enviados para fora dela, são privados. Uma possível exceção a isso é que, para cobrar por um serviço, pode ser necessário para o PS coletar medidas de grossa granularidade. Por grossa granularidade, entende-se como coletas feitas com grandes espaçamentos temporais entre si, como por exemplo o envio mensal do total de energia consumido em uma residencia. Neste trabalho, considera-se que a coleta de medidas de grossa granularidade não representa perigo para a privacidade. No entanto, como apresentado na Seção 3.1.1, existem formas de cobrar, de maneira privada, pelo consumo de um recurso. Vale ressaltar que o problema de cobrança privada pelo consumo de um recurso está fora do escopo deste trabalho. As interações entre o PS e o EG de cada casa são apresentadas na Figura 9.

Os dados, no contexto de *Internet of Things* / Internet das Coisas (IoT), apresentam-se como *streaming* de dados, em oposição a apresentar-se como *data sets* (TöNJES *et al.*, 2014). *Streaming* de dados possuem características intrínsecas que impõem desafios à execução de algoritmos e, em especial, à aplicação de privacidade. Uma dessas características que possui grande impacto na aplicação de privacidade é o fato de *streaming* de dados serem potencialmente infinitas. A aplicação da PD se torna especialmente difícil para *streaming* de dados porque, a princípio, o *budget* é consumido a cada consulta, na PD, e a cada resposta, na PDL.

Uma das estratégias utilizadas ao propor soluções para algoritmos em *streaming* de dados é o conceito de janelas deslizantes (TANBEER *et al.*, 2009), onde uma quantidade fixa de dados mais recentes são utilizados durante o processamento. Estratégias de janela deslizante conseguem simplificar o problema de *streaming* infinitas de dados por limitar o escopo dos dados que são considerados a cada instante de tempo.

A seguir, serão apresentadas três soluções propostas, todas utilizando a arquitetura do ProTECting e garantindo a privacidade através da Privacidade Diferencial Local (PDL). Como apresentado no Capítulo 3, os dados gerados por dispositivos dentro de casas inteligentes são altamente sensíveis e podem vazar informações importantes sobre os residentes da casa. Um exemplo de informação privada que pode ser aprendida com esses dados é se as pessoas de uma casa saíram atrasadas para o trabalho (MOLINA-MARKHAM *et al.*, 2010). A utilização da PDL é justificada pela natureza altamente sensível dos dados, que faz com que possa não ser aceitável depender de uma entidade centralizadora confiável para reunir os dados brutos dos usuários, como é necessário na Privacidade Diferencial (PD) centralizada.

As soluções propostas dividem-se em duas estratégias diferentes. A primeira, utiliza-

se de uma simplificação dos problemas em *streaming* de dados para garantir a privacidade através de janelas deslizantes. A segunda estratégia, elimina essa simplificação do problema utilizando-se de *memoization* e duas rodadas de randomização para garantir a PDL mesmo quando infinitas respostas precisam ser enviadas. Ambas estratégias serão detalhadas a seguir.

4.1 Definições Básicas

O procedimento inicial do ProTECting, descrito a seguir, é comum tanto para a estratégia baseada em janela, quanto para a estratégia de dupla randomização. Primeiramente, o PS entra em contato com os responsáveis de N casas oferecendo algo como, por exemplo, um desconto ou um pagamento, em troca do envio de dados a serem enviados usando um ε definido. Note que, quanto menor o valor de ε, mais privada é a saída (DWORK, 2008).

Inicialmente o dado é representado na forma de um vetor de *bits* utilizando o *Unary Encoding* / Codificação Unária (UE). Esse passo ocorre anteriormente ao processo de randomização, que é feita de tal forma a garantir a definição de PDL. O UE é definido a seguir.

Definição 13. Seja v o valor da resposta e d o tamanho do vetor de **bits** desejado:

$UE(v,d) = B = [0,\ldots,1,\ldots,0]$, onde $b_i = 1$, se $i = v$ e $b_i = 0$, caso contrário, para $1 \leq i \leq d$.

Como apresentado na Definição 13, o UE pode ser diretamente aplicado para valores inteiros, o que pode não ser verdadeiro para dados gerados no contexto de Internet das Coisas, que podem consistir de valores reais. Para a aplicação do UE, se faz necessário, portanto, utilizar uma representação discreta dos valores, que pode ser vista como uma representação por histograma desses valores, onde somente o *bin*[1] cujo intervalo contém o valor a ser enviado será definida como 1, enquanto os outros *bins* conterão o valor zero. Essa estratégia de discretização de valor e aplicação do UE será chamada de *Discrete Unary Encoding* / Codificação Unária Discreta (DUE) (Definição 14). Para que seja feita essa discretização dos valores, é necessário que sejam definidos os seguintes parâmetros: o número de *bins* (d) e o intervalo de valores ($valor\_min, valor\_max$) que contém os valores do domínio de v.

[1] utilizou-se o termo *bin*, em inglês, pois o termo classe, que é geralmente utilizado em português, poderia vir a confundir o leitor a pensar em categorias, ao invés de intervalos de valores, como pretendido.

Definição 14. Seja v o valor da resposta, d o tamanho do vetor de *bits* desejado e min_value e max_value os valores que compõem o domínio das respostas. Considere ainda que $range(b_i)$ indica o intervalo do *bin* b_i do histograma.

$DUE(v, min\_value, max\_value, d) = B = [0, \ldots, 1, \ldots, 0]$, onde $b_i = 1$, se $v \in range(b_i)$ e $b_i = 0$, caso contrário, para $1 \leq i \leq d$. O histograma possui d *bins* que dividem igualmente o domínio $(valor\_min, valor\_max)$. Adotaremos $v \in B_i$ para denotar que um valor v é codificado no vetor de *bits* B_i ao utilizar o DUE.

Os valores d, *valor_min* e *valor_max* farão parte de um conjunto H de parâmetros que deve ser enviado pelo PS para os participantes, onde serão utilizados tanto na codificação de cada valor v a ser enviado de cada participante para o PS quanto, posteriormente, na estimativa de cada valor, executada pelo PS. Além dos parâmetros definidos anteriormente, H possui também o valor de δ, que consiste no intervalo de tempo entre o envio de duas respostas. Também está contido em H o parâmetro de limite de privacidade ε. Portanto, temos que $H = \{\varepsilon, d, \delta, valor\_min, valor\_max\}$.

Com o conhecimento dos valores dos parâmetros de H, os participantes são capazes de utilizar o DUE para codificar um valor v, que deve ser enviado após um intervalo de tempo δ. Após a codificação de um valor, é necessário que seja feita a randomização Bit-a-Bit do vetor de *bits* B. Essa randomização é realizada de forma a garantir a definição da PDL. O procedimento de randomização, denominado *Bitwise_Randomization*, é apresentado na Definição 15.

Definição 15. $Bitwise\_Randomization(B, p, q) = B'$, onde B' consiste em um vetor de *bits* obtido a partir de uma randomização Bit-a-Bit de B com as seguintes probabilidades:

$$Pr[B'[i] = 1 | B[i] = 1] = p \tag{4.1}$$

$$Pr[B'[i] = 1 | B[i] = 0] = q \tag{4.2}$$

$$Pr[B'[i] = 0 | B[i] = 0] = 1 - q \tag{4.3}$$

$$Pr[B'[i] = 0 | B[i] = 1] = 1 - p \tag{4.4}$$

Perceba que, para que a randomização executada pelo protocolo *Bitwise_Randomization* atenda à propriedade da PDL, é necessário que a diferença, em logaritmo, entre a probabilidade de um vetor de resposta B' ter sido gerado por uma resposta B_1 e a probabilidade de ter sido gerado por um uma resposta B_2 seja limitada por um fator ε, para quaisquer B_1 e B_2.

Ou seja:

$$\begin{aligned} & ln(Pr[B'|B_1]) - ln(Pr[B'|B_2]) \leq \varepsilon \\ \equiv\ & ln\left(\frac{Pr[B'|B_1]}{Pr[B'|B_2]}\right) \leq \varepsilon \\ \equiv\ & \frac{Pr[B'|B_1]}{Pr[B'|B_2]} \leq e^{\varepsilon} \end{aligned} \tag{4.5}$$

Alguns trabalhos, como (ERLINGSSON *et al.*, 2014; WANG *et al.*, 2017), mostram que o procedimento executado por *Bitwise_Randomization*, que baseia-se na definição de Resposta Randômica proposta em (WARNER, 1965), garante essa propriedade. O Teorema 1 mostra a nossa versão dessa prova.

Teorema 1. O protocolo *Bitwise_Randomization* garante a propriedade da PDL para

$$\varepsilon = ln\left(\frac{p \cdot (1-q)}{q \cdot (1-p)}\right)$$

Demonstração. Para quaisquer valores de entrada, B_1 e B_2, e saída B', temos que

$$\frac{Pr[B'|B_1]}{Pr[B'|B_2]} = \frac{\prod_{i=1}^{d} Pr[B'[i]|B_1[i]]}{\prod_{i=1}^{d} Pr[B'[i]|B_2[i]]} \tag{4.6}$$

$$\leq \frac{Pr[B'[v_1] = 1|B_1[v_1] = 1] \cdot Pr[B'[v_2] = 0|B_1[v_2] = 0]}{Pr[B'[v_1] = 1|B_2[v_1] = 0] \cdot Pr[B'[v_2] = 0|B_2[v_2] = 1]} \tag{4.7}$$

$$= \frac{p \cdot (1-q)}{q \cdot (1-p)} = e^{\varepsilon} \tag{4.8}$$

A Equação (4.6) vem do fato de cada *bit* de B' ser randomizado de maneira independente dos demais, ou seja, a probabilidade de cada posição de B' depende apenas da respectiva posição de B. A probabilidade condicional de B' dado B_1 (ou B_2), portanto, é igual ao produto das probabilidades condicionais de cada posição.

Perceba que B_1 e B_2 vem de valores v_1 e v_2, respectivamente, ao aplicar o procedimento do DUE sobre esses valores e, portanto, os vetores de *bits* B_1 e B_2 diferem em apenas duas posições, v_1 e v_2. As probabilidades condicionais das posições $i \neq \{v_1, v_2\}$ se cancelam pois são iguais tanto no denominador (dado B_1), quanto no numerador (dado B_2). A Inequação (4.7) ocorre porque manter valores 1 e zero nas posições v_1 e v_2 do vetor B', respectivamente, maximizam a razão.

A igualdade (4.8) vem da definição da probabilidade em (4.1), (4.2), (4.3) e (4.4). Pode-se concluir, portanto, que a diferença em logaritmo das probabilidades condicionais de B' com relação a quaisquer duas entradas do protocolo *Bitwise_Radomization* é limitada pelo parâmetro ε. □

Com isso, mostramos que uma resposta obtida através da utilização do protocolo *Bitwise_Radomization* atende à definição da ε-PDL. Intuitivamente, existe a garantia de que, de posse de uma resposta desse protocolo, não se pode inferir, com grande certeza (limitado por ε), o valor original que gerou a resposta. Esse processo de randomização é utilizado nas duas estratégias a serem apresentadas a seguir, e os detalhes de cada uma serão apresentados em suas respectivas seções.

4.2 Estratégia Baseada em Janela

Esta estratégia tem como objetivo garantir a ε-PDL para uma janela deslizante de tamanho definido. A justificativa para a utilização do conceito de janela deslizante baseia-se na complexidade do problema de garantir a propriedade da Privacidade Diferencial, tanto a Local quanto a centralizada, no contexto de *streaming* de dados. A dificuldade de aplicação da PD nesse contexto ocorre devido ao limite de privacidade (ou *budget*), dado pelo parâmetro ε, que, a priori, é consumido a cada consulta, na PD centralizada, e a cada resposta, na PDL.

A ideia da estratégia é, então, dividir o *budget* entre um número limitado de respostas w a serem enviadas dentro de uma janela de tempo. Ou seja, para cada resposta, deve-se consumir um *budget* de $\varepsilon_i = \frac{\varepsilon}{w}$. Para entender porque é necessário essa divisão do *budget*, suponha que um determinado participante enviou, um número w de vezes, versões diferencialmente privadas de um valor v, onde em cada envio foi-se utilizado um *budget* de ε. Um atacante, ao analisar esse conjunto de respostas, poderia concluir, com uma precisão de $e^{w \cdot \varepsilon}$, que essas respostas provinham de v, não respeitando a definição de PDL, que exige que essa certeza não deve ser superior a e^{ε}, como apresentado em (4.5).

Nessa estratégia, os parâmetros enviados do PS para os participantes consistem nos definidos anteriormente em H, adicionado do parâmetro de número de respostas, w. Logo, o conjunto de parâmetros são: $H_{janela} = H \cup \{w\}$. Como temos os parâmetros de tempo entre duas respostas, δ, e de número w de respostas para um dado ε, podemos definir o tamanho de uma janela tanto em função de tempo $\Delta = \delta \cdot w$, quanto em função do número de respostas, w.

Como apresentado na seção anterior, após utilizar o DUE para obter uma representação de vetor de *bits B* de um valor v, o participante realiza uma randomização Bit-a-Bit deste vetor de modo a garantir a Privacidade Diferencial Local (PDL), cujo procedimento foi apresentado na Definição 15 e sua respectiva prova demonstrada no Teorema 1.

Note que, conforme apresentado no Teorema 1, temos uma relação direta entre o

valor de ε e os valores de p e q. Entretanto, a relação inversa, ou seja, de p e q com relação à ε, não se dá de maneira tão direta e simples. Como consideramos importante a definição das probabilidades com relação ao *budget*, optamos por utilizar a versão simétrica do protocolo *Bitwise_Randomization*, ou seja, onde tem-se que $p+q=1$. Com isso, juntamente com o *budget* definido em (4.8) para o protocolo *Bitwise_Randomization*, temos o sistema de equações a seguir:

$$\begin{cases} p+q = 1 \\ \frac{p\cdot(1-q)}{q\cdot(1-p)} = e^{\varepsilon} \end{cases}$$

Resolvendo esse sistema de equações, temos:

$$\begin{aligned} ln\left(\frac{p\cdot(1-q)}{q\cdot(1-p)}\right) = \varepsilon \quad & \equiv ln\left(\frac{p\cdot p}{q\cdot q}\right) = \varepsilon \\ \equiv ln\left(\frac{p^2}{q^2}\right) = \varepsilon \quad & \equiv 2\cdot ln\left(\frac{p}{q}\right) = \varepsilon \\ \equiv ln\left(\frac{p}{q}\right) = \frac{\varepsilon}{2} \quad & \equiv \frac{p}{q} = e^{\frac{\varepsilon}{2}} \\ \equiv \frac{p}{1-p} = e^{\frac{\varepsilon}{2}} \quad & \equiv p = e^{\frac{\varepsilon}{2}} - p\cdot e^{\frac{\varepsilon}{2}} \\ \equiv p\cdot(1+e^{\frac{\varepsilon}{2}}) = e^{\frac{\varepsilon}{2}} \quad & \equiv p = \frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}}+1} \end{aligned}$$

Com $p = \frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}}+1}$ e $q = 1-p$, temos também que $q = \frac{1}{e^{\frac{\varepsilon}{2}}+1}$. Como justificado anteriormente, apesar de não ser necessário para garantir a definição de PDL, a utilização de um protocolo simétrico facilita a definição das probabilidades do protocolo *Bitwise_Randomization* em função de ε. A primeira solução, utilizando a estratégia baseada em janela, com probabilidades simétricas, será chamada de *Window Based* / Baseada em Janela (WB).

Perceba que, com os valores de p e q simétricos, temos que seus significados são: a probabilidade de manter-se o valor de um *bit* e a probabilidade de invertê-lo, respectivamente. É importante ressaltar, ainda, que a utilidade obtida dessas respostas é inversamente proporcional ao valor de ε. Isso ocorre pois quanto menor o valor de ε, menor é a probabilidade do valor real de um *bit* ser passado adiante para o PS.

Note que a garantia pretendida pela estratégia desta Seção é com relação a um conjunto de w respostas a serem enviadas em um período de tempo Δ e, como definido anteriormente, o *budget* de privacidade utilizado para uma resposta do protocolo é $\varepsilon_i = \frac{\varepsilon}{w}$. Portanto, em uma janela de tempo Δ, temos a aplicação do protocolo *Bitwise_Radomization*, com probabilidades p e q definidas em função de ε_i, w vezes. Isso significa, segundo a propriedade da Composição Sequencial (MCSHERRY, 2010) da Privacidade Diferencial, que as repostas dessa janela garantem a definição da PDL para um *budget* $\varepsilon = w\cdot\varepsilon_i$. É importante ressaltar que essa garantia é

válida para qualquer sequência de w respostas, se enquadrando, com isso, no conceito de janela deslizante.

O Algoritmo 1 apresenta o procedimento executado no envio de uma resposta, que ocorre em uma frequência de tempo δ. Esse processo é executado por cada participante, e sua resposta diferencialmente privada B', é enviada para o PS.

Algoritmo 1: ProTECting - Estratégia WB

Entrada: $v, H_{janela} = \{\varepsilon, valor\_min, valor\_max, d, w\}$
Saída: B'

1 $\varepsilon_i \leftarrow \frac{\varepsilon}{w}$
2 $p \leftarrow \frac{e^{\frac{\varepsilon_i}{2}}}{e^{\frac{\varepsilon_i}{2}}+1}$
3 $q \leftarrow \frac{1}{e^{\frac{\varepsilon_i}{2}}+1}$
4 $B \leftarrow DUE(v, min\_value, max\_value, d)$
5 $B' \leftarrow Bitwise\_Randomization(B, p, q)$
6 **retorna** B'

Note que a garantia da propriedade de PDL para um janela de tamanho w ocorre devido à divisão do *budget* ε para uma quantidade limitada de respostas. Os parâmetros p e q, definidos em função de ε_i, como apresentado no Algoritmo 1, linhas 2 e 3, poderiam ser definidos de diferentes maneiras, desde que a diferença das probabilidades respeitassem a inequação (4.5). Dito isto, será apresentado a seguir uma diferente solução, também baseada em janela deslizante, que tem como objetivo minimizar o erro inserido pelo processo de randomização. Essa segunda estratégia será chamada de *Optimized Window Based* / Otimizada Baseada em Janela (OPT-WB).

A diferença da solução OPT-WB com relação à *Window Based* / Baseada em Janela (WB) se dá quanto à definição das probabilidades p e q. Será utilizada uma estratégia não simétrica, ou seja, onde não é necessário que $p + q = 1$. A utilização de probabilidades não simétricas é necessária para que seja possível implementar um protocolo otimizado, ou seja, que minimiza a variância dos resultados, tendo como objetivo obter respostas mais próximas das reais.

O trabalho (WANG *et al.*, 2017) propôs parâmetros otimizados para o procedimento executado pelo protocolo *Bitwise_Randomization*. Para fazê-lo, os autores calcularam a variância do protocolo, em seguida calcularam sua derivada parcial com relação a q e, então, igualaram o

resultado a zero, obtendo os parâmetros que minimizam a variância, que são:

$$\begin{aligned} p &= 0.5 \\ q &= \frac{1}{e^{\varepsilon}+1} \end{aligned} \tag{4.9}$$

Perceba que, apesar de essas probabilidades não resultarem em um protocolo simétrico, também é possível definir os valores de p e q em função de ε.

A solução OPT-WB consiste, portanto, no mesmo procedimento executado pelo Algoritmo 1, com a diferença de que as probabilidades p e q são definidas conforme (4.9).

É importante ressaltar que uma limitação das soluções WB e OPT-WB, que baseiam-se em janela deslizante, é que um atacante, em poder de uma quantidade de dados superior a w referente a um participante, pode ser capaz, em teoria, de inferir as respostas reais deste participante, com probabilidade superior a e^{ε}. A estratégia a ser apresentada na próxima seção tem o objetivo de eliminar essa limitação.

4.3 Estratégia Otimizada de Dupla Randomização

A estratégia baseada em janela deslizante, apresentada na Seção 4.2, surgiu com o objetivo de simplificar o problema complexo de coleta privada de dados no contexto de Casas Inteligentes, onde a geração de dados ocorre infinitamente. Entretanto, a solução apresentada tem como limitação o fato de a garantia de privacidade ser definida para uma janela de tamanho previamente definido, onde o nível de utilidade é inversamente proporcional ao tamanho de janela. Essa limitação pode fazer com que, em um contexto real, as pessoas não se sintam inclinadas a enviar seus dados para um PS utilizando essa estratégia.

Com o objetivo de eliminar a simplificação imposta pela estratégia de janela deslizantes, foi proposta uma nova abordagem, que baseia-se na ideia de utilizar duas rodadas de randomização localmente diferencialmente privada, proposta em (ERLINGSSON *et al.*, 2014), onde o resultado da primeira randomização de um valor v é armazenado e reutilizado sempre que v for enviado (*memoization*). A proposta utiliza-se de protocolos otimizados (WANG *et al.*, 2017), juntamente com a ideia de dupla randomização, com o objetivo de garantir que os participantes possam enviar infinitas respostas para o PS, ao passo que busca minimizar o ruído adicionado. Essa estratégia será chamada de *Optimized Double Randomization* / Dupla Randomização Otimizada (OPT-DR).

A primeira fase de randomização executada na solução *Optimized Double Randomi-*

zation / Dupla Randomização Otimizada (OPT-DR), chamada de Randomização Permanente, tem como objetivo garantir a privacidade a longo prazo. Um valor v a ser enviado de um participante para o PS é primeiramente codificado em um vetor de *bits* de tamanho d, utilizando a função DUE e, em seguida, é feita uma randomização Bit-a-Bit, utilizando o protocolo *Bitwise_Randomization*, obtendo somo saída o vetor de *bits* B'. Perceba que esse procedimento é o mesmo executado na estratégia baseada em janela deslizante, tendo como diferença o parâmetro de limite de privacidade ε a ser utilizado na randomização.

Uma outra diferença importante é que esta estratégia utiliza-se de *memoization* para impedir um possível ataque de cálculo de média (ERLINGSSON *et al.*, 2014). Como cenário exemplo, imagine um participante que reporta, um grande número de vezes, diferentes vetores de *bits* B' gerados pela função *Bitwise_Randomization* sobre um mesmo valor v. Um atacante seria capaz de inferir, através das médias dos *bits* de cada valor reportado, com uma certeza superior a ε, o valor v que originou as respostas. Perceba que esse problema não ocorre na estratégia baseada em janela, mesmo que ela não utilize o processo de *memoization*, pois o *budget* é dividido entre um determinado número de respostas.

Já a segunda fase de randomização, chamada de Randomização Instantânea, tem como principal objetivo dificultar que um cliente torne-se rastreável com base em suas respostas, além de tornar mais forte o nível de privacidade a curto prazo, uma vez que os dados possuem mais ruído. Para exemplificar um possível ataque, suponha o cenário onde um participante reporta a saída da fase de Randomização Permanente, B', um grande número de vezes. Caso B' seja grande o suficiente, é possível que B' torne-se infrequente entre os participantes a ponto de tornar-se um identificador do participante que o gerou.

As duas fases de randomização desta estratégia, ou seja, a Randomização Permanente e a Randomização Instantânea, diferenciam-se da utilizada anteriormente, na estratégia baseada em janela, pois não é necessário haver a divisão do *budget* entre um número w de respostas. Além disso, a solução OPT-DR, assim como a OPT-WB, apresentada anteriormente, também utiliza-se de parâmetros de probabilidade otimizados para obter resultados mais próximos dos reais, reduzindo o erro, conforme definidos em (4.9).

O Algoritmo 2 apresenta o procedimento executado por um participante para enviar uma versão privada de v para o PS. Primeiramente, o valor é codificado em um vetor de *bits* B utilizando a função DUE, representado na linha 3. Em seguida, nas linhas 4 a 7, é checado se o vetor de *bits* B' referente a v existe, usando a função *Get_Permanent_Randomization*.

Algoritmo 2: ProTECting - Estratégia OPT-DR

Entrada: $v, H = \{\varepsilon_1, valor\_min, valor\_max, d\}$
Saída: S

```
1  p ← 0.5
2  q ← 1/(e^ε1 + 1)
3  B ← DUE(v, valor_min, valor_max, d)
   // Randomização Permanente
4  B' ← Get_Permanent_Randomization(v)
5  se B' == ∅ então
6  |  B' ← Bitwise_Randomization(B, p, q)
7  |  Set_Permanent_Randomization(v)
8  fim
   // Randomização Instantânea
9  S ← Bitwise_Randomization(B', p, q)
10 retorna S
```

Caso não exista, a função retorna o valor $\emptyset$. O vetor de *bits* B' é criado utilizando a função *Bitwise_Randomization* e os parâmetros otimizados apresentados anteriormente, para um *budget* ε_1. Por fim, no passo de Randomização Instantânea, que acontece na linha 9, executa-se novamente o *Bitwise_Randomization* para gerar a resposta S que será enviada para o PS.

Perceba que, para gerar B', o protocolo *Bitwise_Randomization* recebe como entrada o vetor de *bits* B e os parâmetros $p = 0.5$ e $q = \frac{1}{e^{\varepsilon_1}+1}$ o que, pelo Teorema 1, mostra que o nível de privacidade garantido por esse processo é de ε_1. É importante notar, no entanto, que o processo de Randomização Instantânea recebe como entrada o vetor de *bits* B' que, por sua vez, já garante um limite ε_1 de privacidade. Uma implicação disso, é que o limite de privacidade oferecido pelo passo de Randomização Instantânea, ε_2, é limitado por ε_1. Para mensurarmos ε_2, é necessário avaliarmos uma versão do Teorema 1 considerando as probabilidades do vetor S dado B. Para fazê-lo, é necessário considerar, também, as probabilidades de B' dado B. Estas considerações e respectiva prova são apresentados no Teorema 2.

Teorema 2. O passo de Randomização Instantânea satisfaz ε_2-PDL.

Demonstração. Para provar que este passo satisfaz a Privacidade Diferencial Local (PDL) com um limite de privacidade de ε_2, é necessário mostrar que, para quaisquer entradas B_1 e B_2, que respectivamente codificam os valores v_1 e v_2:

$$\frac{Pr[S|B_1]}{Pr[S|B_2]} \leq e_2^{\varepsilon}$$

Para mostrarmos que essa desigualdade ocorre, como vimos no Teorema 1, é necessário analisar as probabilidades Bit-a-Bit. Precisa-se considerar, também, a randomização do

passo de Randomização Permanente, através das probabilidades condicionais de $B'[i]$ dado $B[i]$, que são apresentadas em (4.1), (4.2), (4.3) e (4.4). Lembre-se que o procedimento executado no passo de Randomização Instantânea é o mesmo do passo de Randomização Permanente, i.e. *Bitwise_Randomization*, portanto, as definições das probabilidades condicionais de $S[i]$ dado $B'[i]$ são similares, como apresentadas a seguir:

$$Pr[S[i]|B'[i]] = \begin{cases} Pr[S[i] = 1|B'[i] = 1] = p \\ Pr[S[i] = 1|B'[i] = 0] = q \\ Pr[S[i] = 0|B'[i] = 1] = 1 - Pr[S[i] = 1|B'[i] = 1] = 1 - p \\ Pr[S[i] = 0|B'[i] = 0] = 1 - Pr[S[i] = 1|B'[i] = 0] = 1 - q \end{cases}$$

Portanto, as probabilidades condicionais $Pr[S[i]|B[i]]$ são calculadas como apresentado a seguir:

$$Pr[S[i] = 1|B[i] = 1] = Pr[B'[i] = 1|B[i] = 1] \cdot Pr[S[i] = 1|B'[i] = 1] + Pr[B'[i] = 0|B[i] = 1] \cdot Pr[S[i] = 1|B'[i] = 0] \quad (4.10)$$

$$Pr[S[i] = 1|B[i] = 0] = Pr[B'[i] = 0|B[i] = 0] \cdot Pr[S[i] = 1|B'[i] = 0] + Pr[B'[i] = 1|B[i] = 0] \cdot Pr[S[i] = 1|B'[i] = 1] \quad (4.11)$$

$$Pr[S[i] = 0|B[i] = 1] = 1 - Pr[S[i] = 1|B[i] = 1] \quad (4.12)$$

$$Pr[S[i] = 0|B[i] = 0] = 1 - Pr[S[i] = 1|B[i] = 0] \quad (4.13)$$

Seja $p^* = Pr[S[i] = 1|B[i] = 1]$ e $q^* = Pr[S[i] = 1|B[i] = 0]$, podemos concluir, através de passos análogos à prova do Teorema 1, que:

$$e^{\varepsilon_2} = \frac{p^* \cdot (1 - q^*)}{q^* \cdot (1 - p^*)}$$

□

Com isso, mostramos que, tanto o passo de Randomização Permanente, quanto o de Randomização Instantânea, atendem a propriedade da Privacidade Diferencial Local (PDL) para diferentes limites de privacidade.

Lembre-se que o procedimento de Randomização Permanente é executado apenas uma vez para cada B_i, onde $1 \leq i \leq d$. Já a Randomização Instantânea ocorre sempre que um

valor $v \in B_i$ precisa ser enviado do participante para o PS, através da randomização do vetor B_i, obtendo como resultado o vetor B'_i. Após um grande número de respostas instantâneas serem enviadas, um possível atacante, em teoria, seria capaz de identificar o vetor B'_i que deu origem à essas respostas instantânea. Entretanto, mesmo com total certeza sobre um vetor B'_i, a capacidade de um atacante de descobrir o vetor de *bits* original B_i e, consequentemente, o valor v que o deu origem, é limitada por um fator ε_1.

Um leitor mais atento pode vir a questionar o motivo pelo qual, ao se utilizar o passo de Randomização Permanente para d vetores diferentes B_i, o limite de privacidade ε_1 não se deteriora, chegando a $d \cdot \varepsilon_1$, como ocorreria se considerássemos a propriedade da Composição Sequencial. A justificativa para não haver o acúmulo aditivo do custo de privacidade, se dá pela possibilidade de considerarmos os subconjuntos $B_i \in \{B_1, \dots B_d\}$ como disjuntos entre si, o que permite a aplicação da propriedade da Composição Paralela (MCSHERRY, 2010) que, por sua vez, é mais relaxada que a anterior, não sendo necessário fazer a composição aditiva dos *budgets*. Perceba que, para que possamos considerar os vetores de *bits* B_i como disjuntos entre si, é necessário considerar que o PS não possui a capacidade de identificar a origem de cada resposta recebida, o que é uma restrição aceitável para o problema.

4.4 Estimativa de Valores

Nas seções anteriores, apresentamos duas estratégias diferentes para resolver, sem a necessidade de um Provedor de Serviços (PS) confiável, o problema da coleta de dados privada em cenários de casas inteligentes. As duas estratégias baseiam-se na arquitetura proposta, ***Pr****ivacy for* ***IoT*** *and* ***E****dge* ***C****ompu****ting*** (ProTECting), e utilizam a Privacidade Diferencial Local (PDL) para prover a privacidade dos dados dos indivíduos ao longo do tempo.

A primeira estratégia baseia-se no conceito de janelas deslizantes para prover PDL para um conjunto de respostas de tamanho definido. Foram apresentadas duas soluções utilizando essa estratégia, a WB, que utiliza um protocolo simétrico, e a OPT-WB, que utiliza um protocolo otimizado para obter melhor utilidade. Já a segunda estratégia utiliza duas rodadas de randomização, com o objetivo de prover a PDL mesmo quando infinitas respostas são enviadas, ao mesmo tempo em que usa protocolos otimizados para obter respostas privadas mais próximas das reais.

Perceba que a randomização do ProTECting acontece na borda da rede, em um dispositivo com *edgeOS*, e cujas respostas privadas são enviadas para o PS ao longo do tempo,

com intervalos definidos de δ entre cada resposta. É responsabilidade do PS, portanto, agregar as respostas enviadas pelos participantes a fim de obter uma aproximação da distribuição dos valores reais. A seguir, descreveremos esse processo, executado pelo PS.

É de conhecimento do PS os parâmetros H (ou H_{janela}) utilizados na codificação e randomização dos valores enviados pelos participantes. É de seu interesse, portanto, obter estimativas para esses valores e, com isso, poder extrair algum nível de utilidade das respostas privadas recebidas. Lembre-se que os dados dos participantes são codificados em vetores de *bits* de tamanho d, através da função *Discrete Unary Encoding* / Codificação Unária Discreta (DUE), conforme apresentada na Definição 14. O processo executado pelo PS consiste, então, em estimar a frequência de ocorrência de cada um desses *bits*, obtendo uma espécie de histograma dos valores enviados pelos participantes. A estimativa de um *bin* j do histograma é obtida através da Equação apresentada na Definição 16.

Definição 16. A estimativa de um *bin* j do histograma é obtido por:

$$Estimativa[j] = \frac{\sum_{i=1}^{M} CR[i][j] - M \cdot q^*}{p^* - q^*}$$

Onde $CR[i][j]$ refere-se ao *bit* j da resposta i presente no Conjunto de Respostas (CR). A variável M refere-se ao tamanho do conjunto de respostas CR. Já p^* e q^* são as probabilidades definidas em função de ε nos protocolos de randomização. Considere, para a estratégia baseada em janela deslizante, apresentada na Seção 4.2, que $p^* = p$ e $q^* = q$. Note, ainda, que p^* e q^* são conhecidos pelo PS, uma vez que o valor de ε deve ser definido juntamente com os participantes no início do processo, como apresentado anteriormente.

A utilização da equação mostrada na Definição 16 para estimar frequências de respostas privadas através de protocolos que atendem a definição de PDL é conhecida na literatura, podendo ser encontrada, por exemplo, em (ERLINGSSON *et al.*, 2014). O trabalho (WANG *et al.*, 2017) apresenta a prova de que essa estimativa é não enviesada.

Perceba que o PS tem o interesse de estimar a frequência de valores para cada um dos *bins* do histograma, sendo necessário, portanto, executar alguns passos de pós-processamento no resultado obtido através da equação da Definição 16. Isso ocorre pois essa estimativa obtida será uma estimativa de contagem e, além disso, o resultado dessa estimativa pode produzir um número negativo, caso a quantidade de respostas em um *bin* não seja grande o suficiente para eliminar o ruído inserido pelo(s) protocolo(s) utilizados. Portanto, utilizaremos uma modificação

da estimativa apresentada anteriormente, a qual chamaremos de Est_Freq, para lidar com esses pontos, conforme mostrado na Definição 17.

Definição 17. A estimativa de frequência de um *bin* j do histograma é obtido por:

$$Est\_Freq[j] = \frac{max(0, Estimativa[j])}{\sum_i^d Estimativa[i]}$$

Onde $Estimativa$ refere-se ao conjunto de valores estimados através da equação apresentada na Definição 16

Ao utilizarmos a estimativa de frequência apresentada na Equação 17, eliminamos os resultados negativos e, como cada valor de contagem estimada no passo anterior é dividido pela soma das contagens, obtemos um vetor de distribuição de probabilidades, onde cada posição i do vetor Est_Freq apresenta a frequência estimada para o *bin* i do histograma.

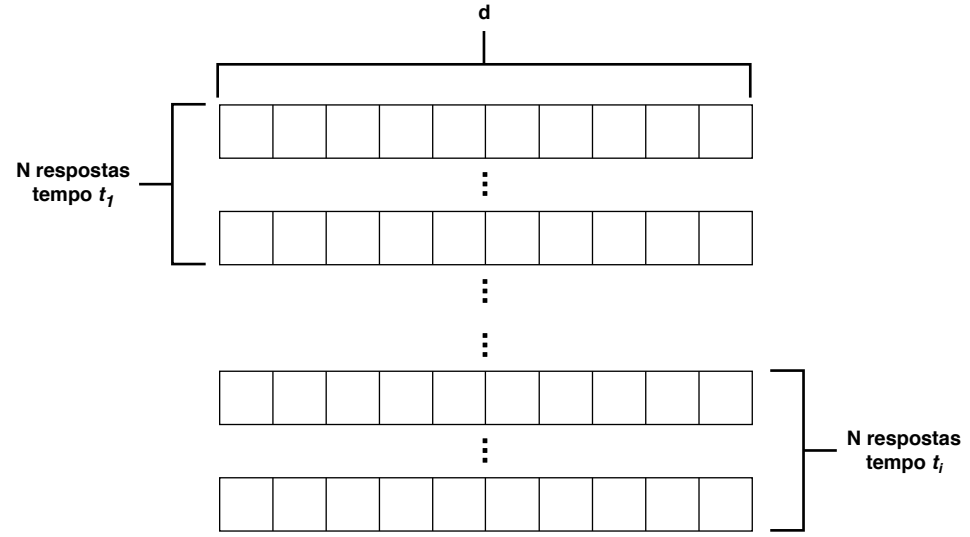
Vale ressaltar que a acurácia das estimativas depende diretamente de dois fatores. O primeiro deles é o valor de limite de privacidade, ε, utilizado na randomização das respostas. Este valor, acordado entre o PS e os participantes, deve ser o mesmo para todos os participantes para que essa estimativa seja possível. Quanto maior o valor de ε, mais relaxada é a privacidade e, consequentemente, mais próxima será a estimativa obtida. O segundo fator que impacta na acurácia é o número de respostas em um *bin* do histograma. Isso significa que, caso um *bin* seja muito infrequente, a estimativa obtida para aquele *bin* será menos próxima da real.

A representação do conjunto de respostas obtido até um tempo t_i pode ser visualizado na Figura 10. Note que, ao estimar as frequências de valores para cada *bin* $b_i, 1 \leq i \leq d$, o PS pode optar por usar o Conjunto de Respostas (CR) de diferentes maneiras. O CR pode referir-se a todo o conjunto de respostas até o momento, o que, possivelmente, resultaria em uma estimativa com maior acurácia. Outra alternativa seria analisar cada CR de um dado tempo t_i separadamente, o que permitiria uma análise que leva em consideração o fator tempo. Perceba que esta última estratégia é mais sensível ao número de participantes N, uma vez que não se faz possível o acúmulo de dados de diferentes instantes de tempo para que a análise seja executada.

4.5 Conclusão

Este capítulo apresentou ProTECting, uma solução para prover a Privacidade Diferencial Local (PDL) em cenários de casas inteligentes. O ProTECting divide-se em duas

Figura 10 – Representação do conjunto de respostas obtidas pelo PS até o tempo t_i.



principais estratégias, a baseada em janela e a baseada em dupla randomização. Além disso, para a estratégia baseada em janela, apresentamos também a sua versão otimizada, ou *Optimized Window Based* / Otimizada Baseada em Janela (OPT-WB), que consiste no uso de parâmetros otimizados. Os dois protocolos otimizados, isto é, o OPT-WB e o OPT-DR, utilizam parâmetros que minimizam a variância com o objetivo de diminuir o erro das estimativas obtidas. Por último, apresentamos o procedimento que deve ser executado pelo PS para obter estimativas de frequência não enviesadas.

5 AVALIAÇÃO EXPERIMENTAL

No capítulo anterior, mostramos a garantia de privacidade do ProTECting ao provar que a diferença das probabilidades de respostas geradas pelas consultas atendem à definição de PDL. Neste capítulo, serão apresentados os resultados experimentais das diferentes estratégias do ProTECting.

Primeiramente, na Seção 5.1, o ambiente de execução é apresentado. Em seguida, na Seção 5.2, descrevemos o conjunto de dados utilizado nos experimentos. Posteriormente, explicamos como é feita a análise de utilidade, detalhando as métricas que são utilizadas na Seção 5.3. Por último, apresentamos os resultados obtidos em termos das métricas utilizadas. Para a estratégia baseada em janela, comparamos os algoritmos WB e sua versão otimizada, o OPT-WB. Com relação à estratégia de dupla randomização, o algoritmo OPT-DR do ProTECting é comparado com o RAPPOR (ERLINGSSON *et al.*, 2014).

5.1 Ambiente de Execução

A máquina utilizada na execução dos experimentos, disponibilizada pelo Laboratório de Sistemas e Banco de Dados (LSBD/UFC), tem as seguintes configurações: processador Intel Core i7-7800X 3.50GHz, 6 núcleos, sistema operacional Ubuntu, com versão do *kernel* 5.3.0 e 125GB de RAM. As duas estratégias propostas do algoritmo ProTECting, baseada em janela deslizante e utilizando dupla randomização, foram implementadas utilizando a linguagem de programação Python, versão 3.7.4. Foram utilizadas, ainda, as bibliotecas Pandas (MCKINNEY, 2010), Numpy (van der Walt *et al.*, 2011) e Scipy (VIRTANEN *et al.*, 2020) para dar suporte a manipulação dos dados.

5.2 Conjunto de Dados

Nos experimentos, utilizamos um conjunto de dados real gerados por *smart meters* (UK Power Networks, 2015). Os dados consistem em leituras de consumo de energia de uma amostra de 5.567 casas de Londres geradas entre Novembro de 2011 e Fevereiro de 2014. Os dados contém o consumo de energia, em kWh (por meia hora), um identificador único da residência, data e hora, e o grupo a que a casa pertence. O grupo a que uma casa pertence indica se a mesma foi submetida ou não a uma tarifação dinâmica baseada no horário de uso. O conjunto de dados contém, aproximadamente, 167 milhões de registros e 10GB. Nos experimentos, foi

utilizado apenas o atributo de consumo de energia.

5.3 Análise de Utilidade

Utilizamos duas métricas para analisar os resultados obtidos com o objetivo de avaliar a utilidade, o Erro Quadrático Médio (EQM) (LEHMANN; CASELLA, 2006) e a *Jensen-Shannon Distance* / Distância de Jensen-Shannon (JSD) (WONG; YOU, 1985). As duas métricas utilizadas foram aplicadas para comparar as estimativas obtidas (Definição 17) com as frequências reais dos valores de respostas. O EQM foi escolhido para avaliar a proximidade dos valores de frequência, enquanto a JSD consiste em uma métrica mais precisa para avaliar os resultados como distribuições de probabilidades, calculando a distância entre duas distribuições. É esperado que os resultados possuam um comportamento semelhante em ambas as métricas, visto que um estimador que possui um erro menor entre as frequências geradas e as frequências reais, resultará em uma distribuição que também se assemelha mais à real. Análises realizadas sobre dados de frequência perturbados mais próximos dos dados reais tendem a apresentar mais utilidade para os provedores de serviço.

5.3.1 Métricas

5.3.1.1 Erro Quadrático Médio

O EQM é apresentado na Definição 18 e tem como objetivo mensurar a distância média entre as frequências estimadas pelo PS e as frequências reais. Note que, por estarmos tratando de frequências, o valor desse erro resultará em um número entre zero e 1, onde um resultado mais próximo de zero significa que as frequências estimadas são mais próximas das reais, e mais próximo de 1 que as frequências estimadas estão mais distantes.

Definição 18. Seja d o número de *bins* de um histograma. O Erro Quadrático Médio (EQM) é dado por:

$$EQM(Est\_Freq, Freq) = \frac{\sum_{i=1}^{d}(Est\_Freq[i] - Freq[i])^2}{d}$$

Onde Est_Freq é o vetor de frequências estimadas, obtido através da equação apresentada na Definição 17, e $Freq$ refere-se ao vetor real de frequências.

5.3.1.2 Distância de Jensen–Shannon

A divergência de Jensen-Shannon é um método para medir a similaridade entre duas distribuições de probabilidade e baseia-se na divergência de Kullback-Leibler (KULLBACK; LEIBLER, 1951), tendo como principal diferença o fato de tratar-se de uma distância simétrica. Além disso, essa divergência sempre resulta em um número finito, diferentemente da divergência de Kullback-Leibler, que pode resultar em um valor infinito. Já a JSD consiste em uma métrica (OSTERREICHER; VAJDA, 2003) e é calculada como a raiz quadrada da divergência, como apresentada na Definição 19.

Definição 19. A *Jensen-Shannon Distance* / Distância de Jensen-Shannon (JSD) é dada por:

$$\text{JSD}(Est\_Freq \parallel Freq) = \sqrt{\frac{\mathcal{D}(Est\_Freq \parallel M) + \mathcal{D}(Freq \parallel M)}{2}}$$

Onde Est_Freq é o vetor de frequências estimadas, $Freq$ é o vetor de frequências real, M é a média ponto-a-ponto de Est_Freq e $Freq$, e $\mathcal{D}$ é a divergência de Kullback-Leibler. Note que Est_Freq, $Freq$ e M são distribuições de frequências de valores e que essas se aproximam de distribuições de probabilidade à medida que aumenta-se o tamanho da amostra. A seguir, as distribuições de frequência são utilizadas como distribuições de probabilidade. A divergência de Kullback-Leibler, dado por $\mathcal{D}(P \parallel Q)$, por sua vez, é dada pela esperança da diferença, em logaritmo, entre as probabilidades P e Q, onde a esperança é dada usando as probabilidades de P.

A JSD tem como propriedade que $0 \leq \text{JSD}(P \parallel Q) \leq 1$. Assim como a métrica do EQM, apresentada anteriormente, um valor mais próximo de zero indica distribuições mais próximas entre si, enquanto que um valor mais próximo de 1 indica distribuições mais distantes.

O EQM foi escolhido por se tratar de uma métrica simples e bastante conhecida, sendo capaz de mensurar a diferença entre os valores de frequência estimados e reais, ao passo que penaliza valores muito diferentes entre si. Já a JSD tem a vantagem de se basear na divergência de Kullback-Leibler, que por sua vez mensura a entropia relativa entre duas distribuições, que trata-se de uma boa forma de medir a distância entre duas distribuições. Além disso, a JSD é definida mesmo quando a probabilidade de um valor é zero em alguma das distribuições, o que não é verdade na divergência de Kullback-Leibler. Essa característica é importante no contexto deste trabalho porque não se tem garantia que um determinado *bin* terá

uma frequência diferente de zero tanto na distribuição real quanto na estimada. Ademais, a JSD é uma métrica, atendendo a desigualdade dos triângulos e sendo simétrica.

5.3.2 Resultados

Para executar a análise experimental dos algoritmos, foram implementadas as estratégias baseadas em janela e baseadas em dupla randomização. Os resultados são apresentados a seguir agrupados conforme a estratégia utilizada.

Para avaliar a estratégia baseada em janela deslizante, implementamos tanto o algoritmo WB quanto sua versão otimizada, OPT-WB. Já para avaliar a estratégia de dupla randomização, implementamos o algoritmo proposto, OPT-DR, e o seu concorrente direto, o RAPPOR, apresentado na Seção 3.3.1.

Para uma simulação dos experimentos, foram geradas amostras aleatórias dos dados de tamanho N, onde N representa o número de casas participantes. Os valores de N utilizados foram: $N \in \{10^3, 10^4, 10^5, 10^6\}$. A variação dos valores de N tem como objetivo mostrar o impacto da quantidade de usuários na precisão da estimativa obtida pelo PS. Além disso, variamos o limite de privacidade com os seguintes valores: $\varepsilon \in \{1, 2, 3, 5, 10\}$. A escolha dos valores para esse parâmetro foi baseada em trabalhos da literatura (ERLINGSSON *et al.*, 2014; WANG *et al.*, 2017). Além disso, um valor menor implica em uma maior privacidade, enquanto que um valor maior significa maior utilidade.

O número de *bins* do histograma d foi fixado em 100. Também avaliamos o cenário onde cada usuário participante envia dados para o PS 10 vezes. Na estratégia baseada em janela, fixamos o tamanho da janela w em 10, garantindo que todas as respostas enviadas na avaliação são diferencialmente privadas. Os resultados apresentados a seguir são a média das 10 execuções dos algoritmos. Note que isso é equivalente ao PS utilizar o Conjunto de Respostas (CR) individualmente para cada tempo t_i, como apresentado na Seção 4.4.

5.3.2.1 Estratégia Baseada em Janela

A Figura 11 apresenta os resultados médios para a métrica EQM para os diferentes limites de privacidade, variando o número de participantes N. Cada gráfico dessa figura apresenta os resultados para um valor de ε. O eixo horizontal de cada gráfico corresponde ao número de participantes N, enquanto o eixo vertical apresenta o valor médio do EQM. Além disso, em cada gráfico a linha laranja com círculos representa a estratégia WB, enquanto a azul com quadrados

indica os resultados para o OPT-WB. Pode-se observar, de maneira geral, que à medida que o número de participantes aumenta, o EQM diminui. Esse comportamento é esperado e ocorre porque, com o aumento de N, a frequência estimada de cada um dos *bins* torna-se mais próxima da real. Para um limite de privacidade $\varepsilon = 2$ e $N = 10^5$, por exemplo, pode-se constatar um EQM de aproximadamente 0.001.

A Figura 12, por sua vez, apresenta cinco gráficos, cada um para um valor diferente de ε, onde as linhas das estratégias seguem as mesmas cores dos gráficos da Figura 11. O eixo horizontal desses gráfico exibem o número de participantes, enquanto o eixo vertical apresenta a JSD.

Note que para um $\varepsilon = 2$ e $N = 10^5$, diferentemente do que foi observado para o EQM, temos que as distribuições de frequência obtidas nas duas estratégias tem uma distância de, aproximadamente, 0.5 para a distribuição real de frequências, significando que as distribuições não são tão próximas entre si. Já para um $\varepsilon = 10$, temos tanto um EQM, quanto uma JSD, baixas, significando que foram obtidas distribuições de frequências muito mais próximas da real, mesmo quando o número de participantes não é tão grande. Assim como ocorre no EQM, à medida que N aumenta, a JSD diminui, seguindo a mesma justificativa. Também é possível observar, como esperado, que um maior valor de ε implica em um menor EQM e, também, uma menor JSD, uma vez que um maior *budget* implica em menor privacidade e, consequentemente, melhor utilidade das estimativas.

Esses resultados nos mostram que ambas as estratégias baseadas em janela deslizante são bastante sensíveis ao limite de privacidade utilizado, sendo necessário valores altos de ε para obter resultados próximos dos reais. Isso ocorre pois essas estratégias utilizam-se da fragmentação do limite de privacidade ε, fazendo com que não se tenha disponível um valor de *budget* razoável para uma resposta quando valores pequenos são utilizados.

Além disso, ao analisarmos os resultados de ambas as métricas, pode-se concluir que a utilização de parâmetros otimizados não foi suficiente para obter uma melhor utilidade para o OPT-WB. Antes de entendermos o motivo de isso ocorrer, observe a Tabela 6 que exibe, para os diferentes valores de ε_i utilizados, a variância dos protocolos WB e OPT-WB, assim como a variação percentual entre esses valores. Lembre-se que o ε_i corresponde ao *budget* utilizado para uma resposta, que, por sua vez, é igual a $\frac{\varepsilon}{w}$, onde w é o tamanho da janela. Note ainda que a variação percentual é dada por (5.1) e nos dá uma perspectiva de quão menor é o valor da variância do OPT-WB (valor_novo) com relação à variância do WB (valor_antigo). Ao

Figura 11 – EQM para as estratégias baseadas em janela ao variar o parâmetro N.

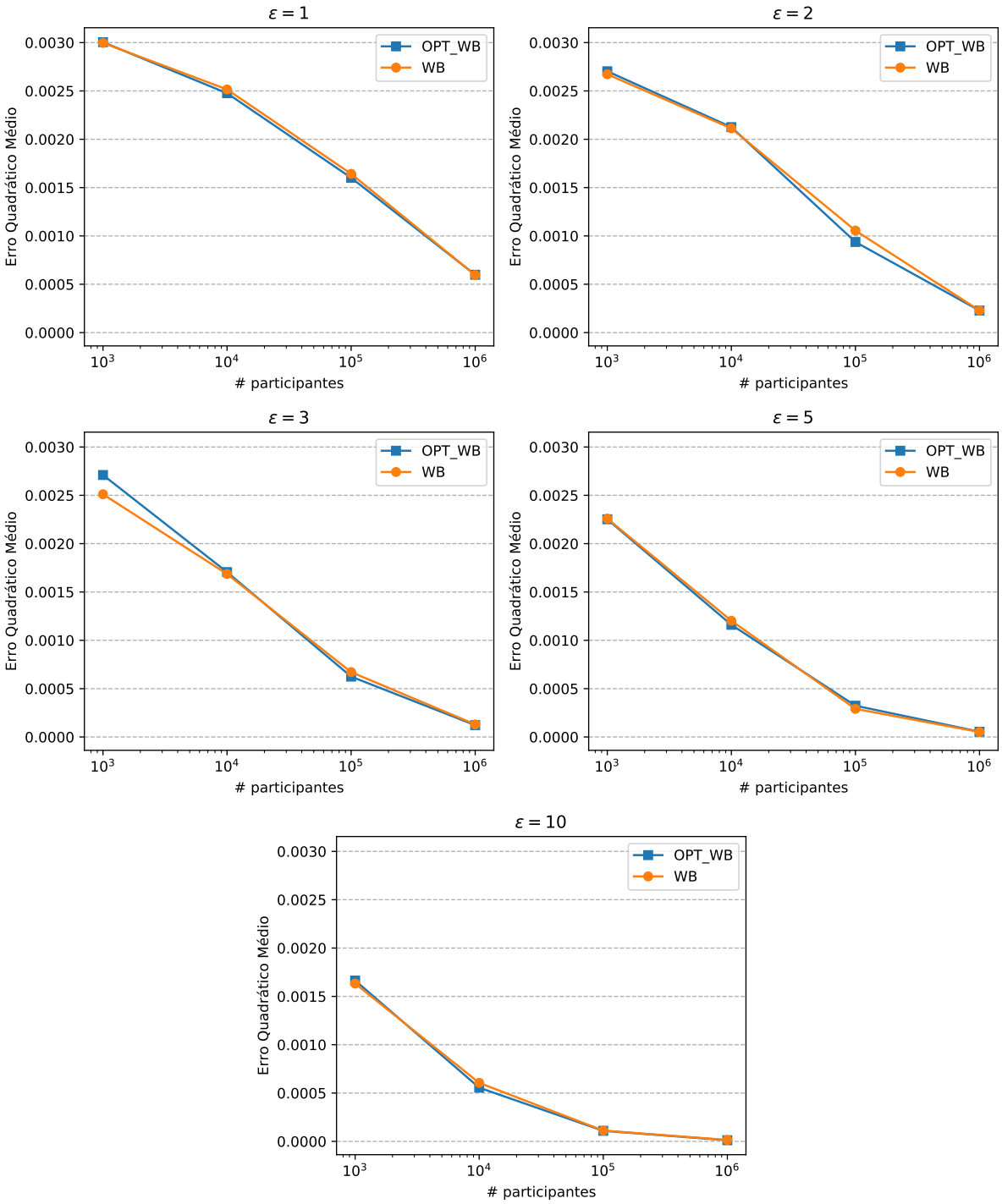
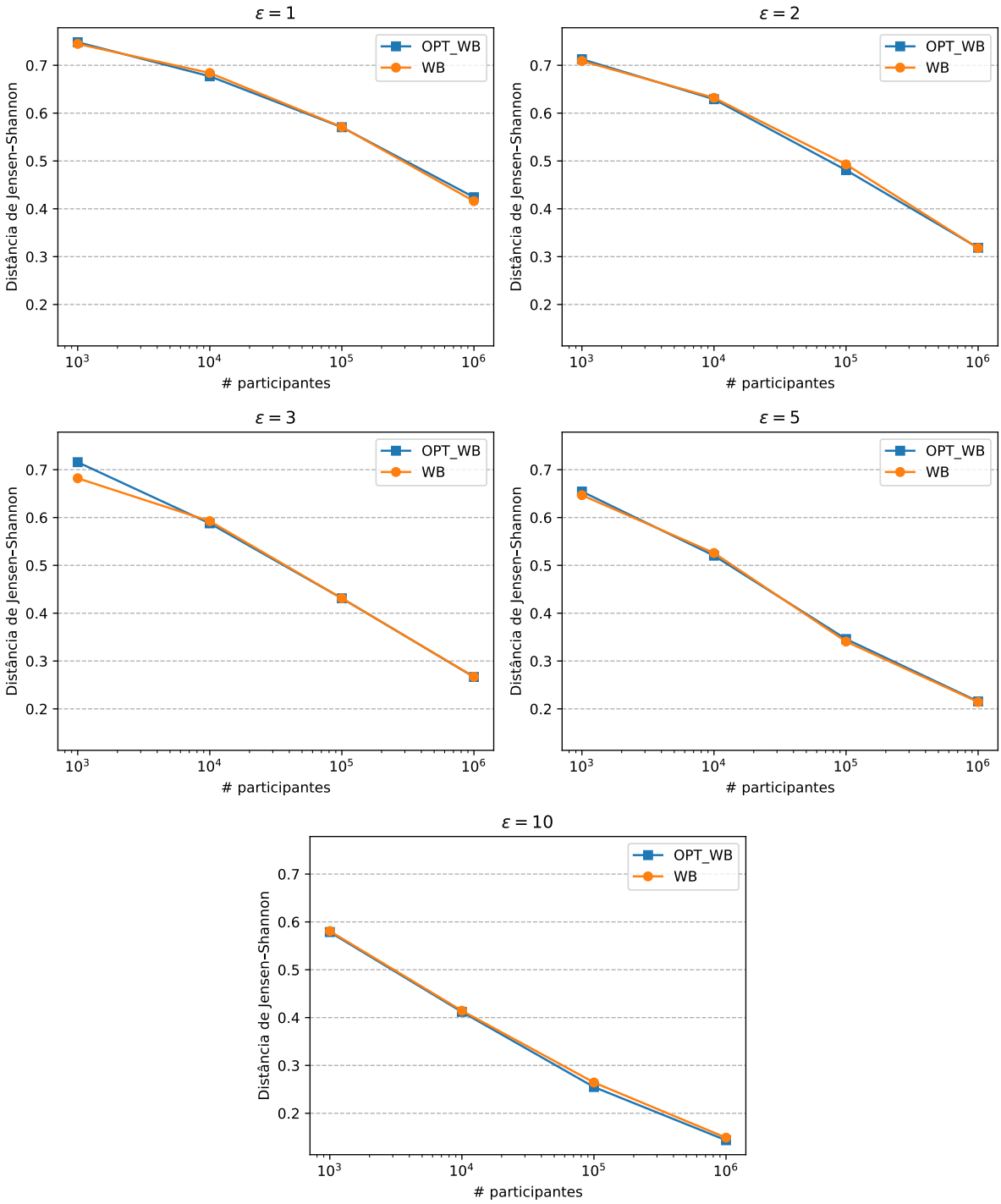


Figura 12 – JSD para as estratégias baseadas em janela ao variar o parâmetro N.



analisarmos essa tabela, podemos entender o motivo de não haver melhoria ao utilizar parâmetros otimizados. Isso ocorre, pois, quando o valor do limite de privacidade ε_i para uma resposta é muito baixo, as variâncias de ambos os protocolos são muito próximas entre si.

$$(\frac{valor\_novo}{valor\_antigo}) - 1 \tag{5.1}$$

Tabela 6 – Comparação entre as variâncias do WB e OPT-WB para os diferentes valores de ε_i.

ε_i	**Variância do WB**	**Variância do OPT-WB**	**Variação Percentual**
0.1	399.91667708229886	399.66683326721824	-0.06247396755330037%
0.2	99.91670831680456	99.66733227661182	-0.2495839228430663%
0.3	44.361204777471656	44.11260577079411	-0.560397328982809%
0.5	15.916926438868662	15.670792356131054	-1.5463669049607276%
1	3.9176980890327635	3.6826943768311695	-5.998515119362191%

A variância do WB pode ser calculada por $\frac{e^{\frac{\varepsilon_i}{2}}}{(e^{\frac{\varepsilon_i}{2}}-1)^2}$, enquanto que a variância do OPT-WB é dada por $\frac{4 \cdot e^{\varepsilon}_i}{(e^{\varepsilon}_i-1)^2}$ (WANG *et al.*, 2017) e, conforme o valor de ε_i aumenta, maior é também a diferença entre as variâncias.

5.3.2.2 Estratégia de Dupla Randomização

As Figuras 13 e 14 apresentam, respectivamente, os valores médios do EQM e da JSD para os algoritmos OPT-DR e RAPPOR. Assim como nos resultados da estratégia baseada em janela, pode-se constatar que um maior número de participantes implica em estimativas de frequências mais próximas das reais. Pode-se observar, ainda, que quando o número de participantes N é bastante elevado, a estratégia OPT-DR é capaz de proporcionar uma distribuição de frequências próxima da real, mesmo para valores de ε_1 relativamente baixos, como pode ser visualizado, por exemplo, quando tem-se $N = 10^6$ e $\varepsilon_1 = 2$, que resulta em um EQM próximo de zero e uma JSD de, aproximadamente, 0.19.

Diferentemente do observado na estratégia baseada em janela deslizante, ao comparar os resultados do OPT-DR e do RAPPOR, pode-se constatar que o uso de parâmetros otimizados é capaz de prover resultados consideravelmente melhores para ambas as métricas. As Figuras 15 e 16 apresentam a diferença percentual, dada por (5.1), entre os resultados obtidos com os dois algoritmos para as métricas de EQM e JSD, respectivamente. Note que, para todos os valores de N, houve uma redução percentual entre os resultados de ambas as métricas quando comparando o OPT-DR e o RAPPOR, comprovando que a utilização de protocolos otimizados resultou em

Figura 13 – EQM para as estratégias de dupla randomização ao variar o parâmetro N.

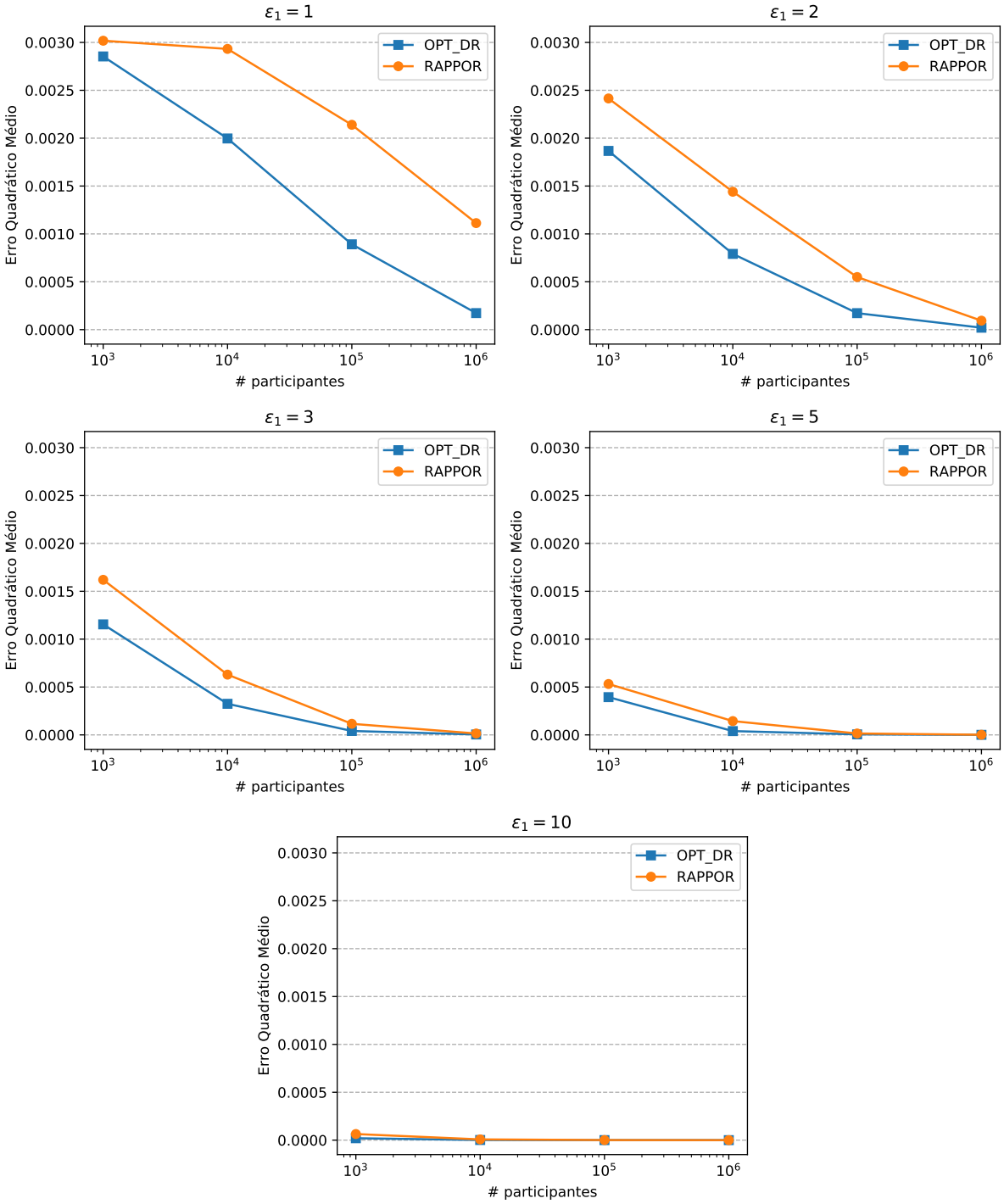
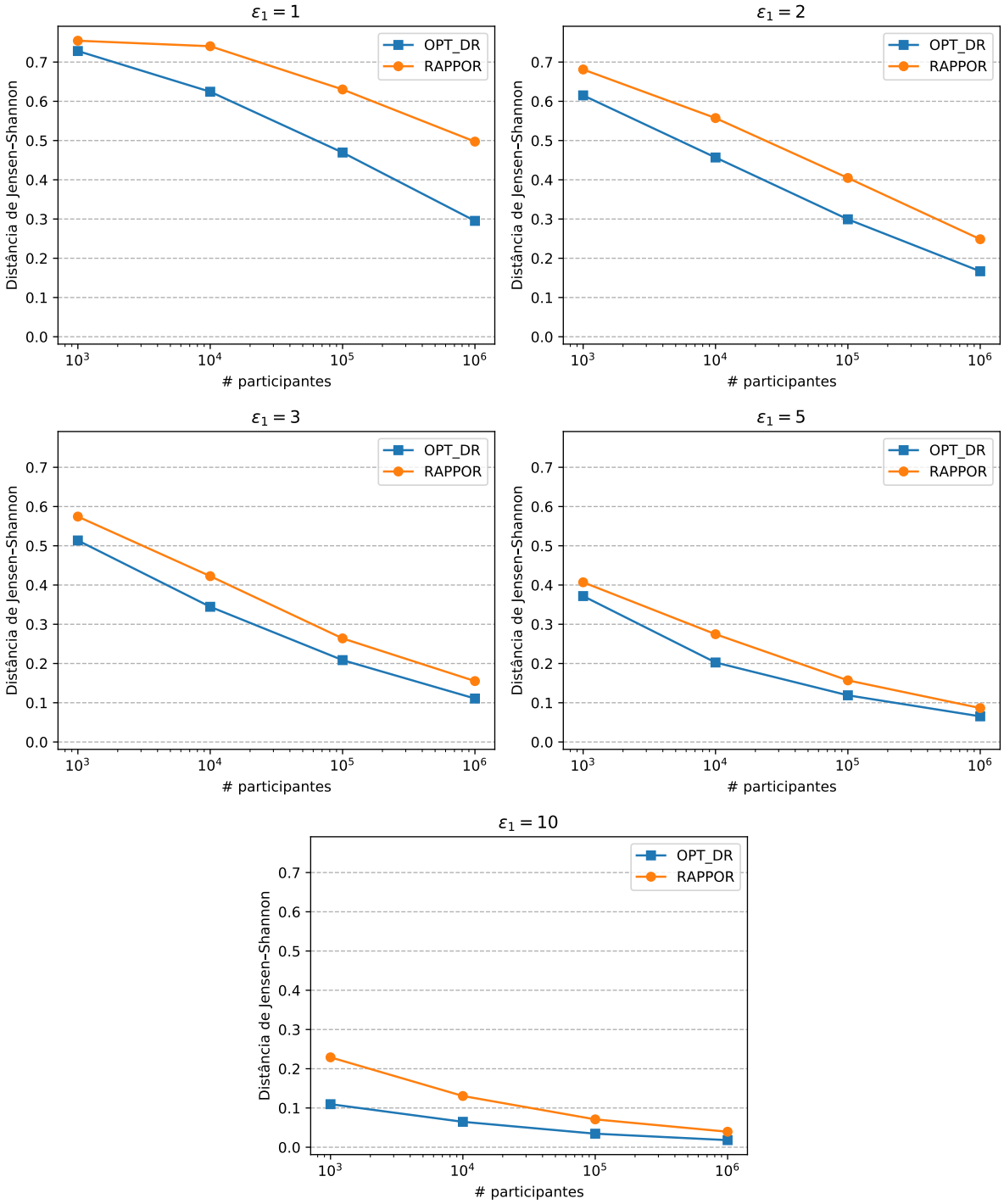
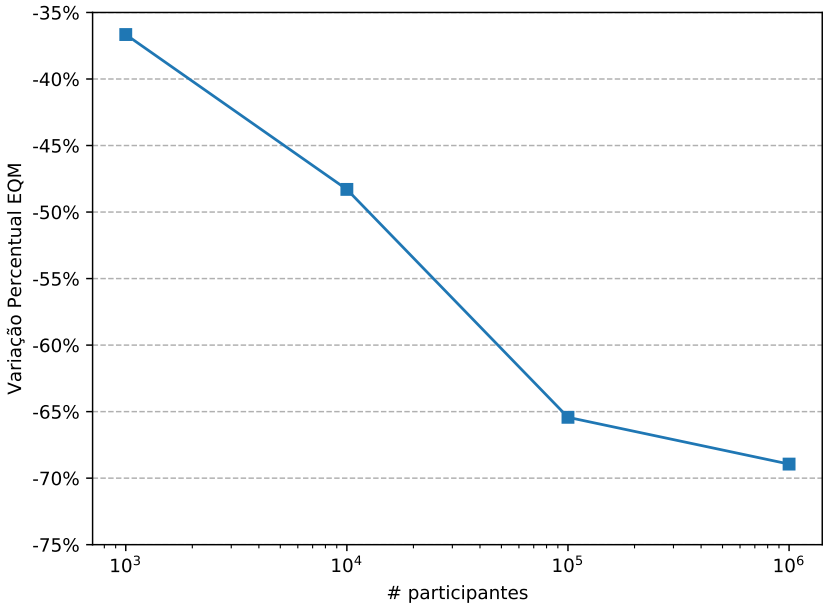


Figura 14 – JSD para as estratégias de dupla randomização ao variar o parâmetro N.



estimativas de frequência com menores erros. Pode-se observar ainda, uma tendência de uma maior diminuição percentual à medida que a quantidade de participantes aumenta. Vale ressaltar que as variações percentuais exibidas nas Figuras 15 e 16 são valores médios de todos os ε_1 analisados. A utilização desse valor médio tem como objetivo demonstrar essa relação entre a variação percentual e o número de participantes. Por outro lado, não pudemos observar a existência de uma relação entre essa variação percentual e o valor de ε_1 utilizado.

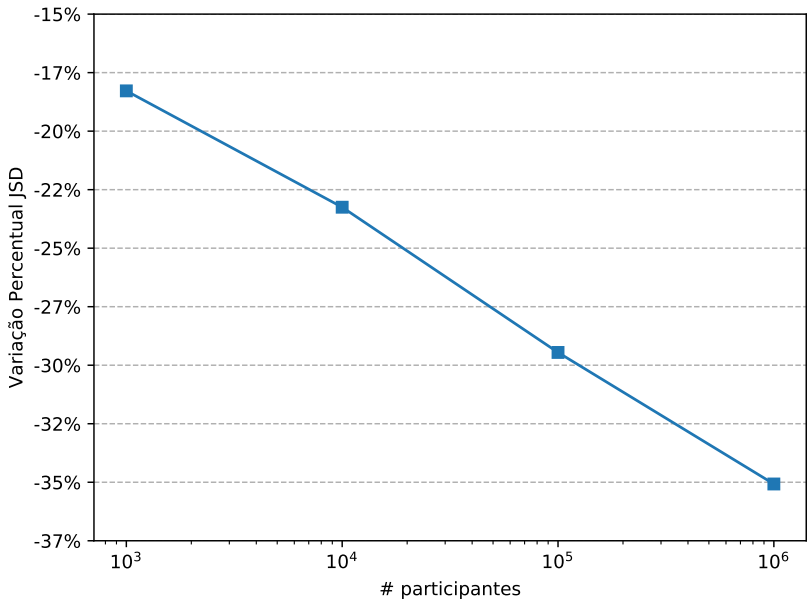
Figura 15 – Variação percentual média do EQM entre os resultados obtidos com o RAPPOR e o OPT-DR ao variar o parâmetro N.



5.4 Conclusão

Este capítulo apresentou uma análise comparativa entre protocolos otimizados e não otimizados, com relação as diferentes métricas de utilidade. A comparação foi feita internamente entre os protocolos de cada uma das duas abordagens do ProTECting, ou seja, baseada em janela deslizante e em dupla randomização. A análise permite concluir que mesmo utilizando parâmetros otimizados, o OPT-WB não consegue obter melhores resultados que o WB. Isso ocorre devido aos seus valores muito pequenos de ε disponíveis por resposta. Por outro lado, para a estratégia de dupla randomização, pode-se observar que o OPT-DR obteve resultados expressivamente melhores que seu concorrente direto, o RAPPOR, para os mesmos níveis de privacidade. Além disso, mostramos o impacto do número de participantes N e dos diferentes limites de privacidade ε na utilidade obtida.

Figura 16 – Variação percentual média da JSD entre os resultados obtidos com o RAPPOR e o OPT-DR ao variar o parâmetro N.



6 CONSIDERAÇÕES FINAIS

6.1 Principais Contribuições

Este trabalho investigou e propôs uma solução chamada de ProTECting, que garante a privacidade dos indivíduos com o auxílio da PDL, ao mesmo tempo em que mantém um bom nível de utilidade para as informações extraídas dos dados privados. A solução é apropriada para o contexto de *streaming* de dados de IoT pois garante que múltiplas respostas privadas podem ser enviadas ao longo do tempo, ou seja, considera a propriedade potencialmente infinita das *streaming* de dados.

O ProTECting é um algoritmo que deve ser executado em um *gateway* na borda da rede, na casa dos indivíduos participantes. A solução pode ser instanciada por meio de duas estratégias diferentes, ambas provendo a garantia da PDL para múltiplas respostas. A primeira estratégia, baseada em janela, garante a propriedade da PDL para um número limitado de respostas. A segunda estratégia, por sua vez, implementa dois processos de randomização, ambos utilizando parâmetros otimizados, para garantir que infinitas respostas possam ser enviadas sob a garantia da PDL, ao mesmo tempo em que consegue obter resultados com maior utilidade, ou seja, mais próximos dos reais. Além disso, foi realizada, ainda, uma avaliação experimental que mostra que os resultados do ProTECting com acurácia proporcional ao número de participantes e ao limite de privacidade ε definido. Em especial, para a estratégia baseada em dupla randomização, mostramos que o ProTECting consegue obter um resultado com maior utilidade que seu concorrente direto, o RAPPOR, para um mesmo limite de privacidade ε.

Ao analisar os resultados obtidos para diferentes métricas de utilidade, podemos concluir que o ProTECting constitui uma solução viável para prover a privacidade em contextos de casas inteligentes, dando fortes garantias formais quanto ao nível de privacidade obtido, conforme a definição da PDL. Além disso, a solução faz uso do conceito de computação em borda, o que faz com que suas contribuições sejam aplicáveis em cenários reais. No ProTECting, a estratégia baseada em janela não obteve melhoras nos resultados ao utilizar parâmetros otimizados. Além disso, embora seja possível obter estimativas com boa acurácia quando se tem um grande número de participantes, a estratégia baseada em janela tem a limitação intrínseca à solução de que a privacidade obtida é definida para um número fixo de respostas de um participante. A estratégia OPT-DR vem para suprir essa limitação, possibilitando o envio de infinitas respostas sem comprometer a privacidade dos usuários, ao mesmo tempo em que

consegue obter resultados com ainda maior acurácia que o *baseline*.

6.2 Trabalhos Futuros

Como trabalho futuro, podemos citar a adaptação da estratégia do ProTECting para a coleta de informações de redes de computadores, fazendo as modificações necessárias de acordo com as informações que são interessantes de se coletar nesse contexto. Para isso, possivelmente será necessário expandir a estratégia para funcionar para outros protocolos, além do protocolo Oráculo de Frequência (OF).

Ademais, pretendemos experimentar a estratégia utilizando outros conjuntos de dados com diferentes características. Embora o ProTECting não utilize em suas estratégias nenhuma informação referente aos dados e suas distribuições, fazendo com que isso não impacte sobre o nível de privacidade garantido, pode ser interessante avaliar como diferentes distribuições de dados impactam no ganho de utilidade obtido com relação ao *baseline*. Pretendemos igualmente avaliar o impacto da variação do parâmetro de tamanho da resposta d na utilidade. Note que este parâmetro não afeta diretamente a privacidade dos usuários, mas, como comentado anteriormente, um d muito grande pode fazer com que a frequência de cada uma das respostas obtidas seja pequena e, com isso, diminua a acurácia obtida. Por outro lado, um d muito pequeno implica numa maior discretização dos dados, o que também pode comprometer as informações obtidas. Portanto, é importante que seja feito um estudo quanto à escolha do parâmetro d para diferentes números de usuários n.

REFERÊNCIAS

ÁCS, G.; CASTELLUCCIA, C. I have a dream! (differentially private smart metering). *In*: FILLER, T.; PEVNÝ, T.; CRAVER, S.; KER, A. (ed.). **Information Hiding**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 118–132. ISBN 978-3-642-24178-9.

CAO, Y.; YOSHIKAWA, M. Differentially private real-time data release over infinite trajectory streams. *In*: **International Conference on Mobile Data Management**. [*S. l.*: *s. n.*], 2015. v. 2, p. 68–73.

CHEN, Y.; MACHANAVAJJHALA, A.; HAY, M.; MIKLAU, G. Pegasus: Data-adaptive differentially private stream processing. *In*: **Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security**. New York, NY, USA: Association for Computing Machinery, 2017. (CCS '17), p. 1375–1388. ISBN 9781450349468.

CORMODE, G.; JHA, S.; KULKARNI, T.; LI, N.; SRIVASTAVA, D.; WANG, T. Privacy at scale: Local differential privacy in practice. *In*: **Proceedings of the 2018 International Conference on Management of Data**. New York, NY, USA: Association for Computing Machinery, 2018. (SIGMOD '18), p. 1655–1658. ISBN 9781450347037.

DEPURU, S. S. S. R.; WANG, L.; DEVABHAKTUNI, V.; GUDI, N. Smart meters for power grid — challenges, issues, advantages and status. *In*: **2011 IEEE/PES Power Systems Conference and Exposition**. [*S. l.*]: IEEE, 2011. p. 1–7.

Domingo-Ferrer, J.; Sanchez, D.; Soria-Comas, J. **Database Anonymization: Privacy Models, Data Utility, and Microaggregation-based Inter-model Connections**. [*S. l.*: *s. n.*], 2016.

DUCHI, J. C.; JORDAN, M. I.; WAINWRIGHT, M. J. Local privacy and statistical minimax rates. *In*: **2013 IEEE 54th Annual Symposium on Foundations of Computer Science**. [*S. l.*]: IEEE, 2013. p. 429–438.

DWORK, C. Differential privacy. *In*: **33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)**. [*S. l.*]: Springer Verlag, 2006. (Lecture Notes in Computer Science, v. 4052), p. 1–12. ISBN 3-540-35907-9.

DWORK, C. Differential privacy: A survey of results. *In*: AGRAWAL, M.; DU, D.; DUAN, Z.; LI, A. (ed.). **Theory and Applications of Models of Computation**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 1–19. ISBN 978-3-540-79228-4.

DWORK, C.; MCSHERRY, F.; NISSIM, K.; SMITH, A. Calibrating noise to sensitivity in private data analysis. *In*: HALEVI, S.; RABIN, T. (ed.). **Theory of Cryptography**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 265–284. ISBN 978-3-540-32732-5.

DWORK, C.; ROTH, A. The algorithmic foundations of differential privacy. **Found. Trends Theor. Comput. Sci.**, Now Publishers Inc., Hanover, MA, USA, v. 9, n. 3–4, p. 211–407, ago. 2014. ISSN 1551-305X.

DWORK, C.; YEKHANIN, S. New efficient attacks on statistical disclosure control mechanisms. *In*: WAGNER, D. (ed.). **Advances in Cryptology – CRYPTO 2008**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 469–480. ISBN 978-3-540-85174-5.

ERLINGSSON, U.; PIHUR, V.; KOROLOVA, A. Rappor: Randomized aggregatable privacy-preserving ordinal response. *In*: **Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security**. New York, NY, USA: Association for Computing Machinery, 2014. (CCS '14), p. 1054–1067. ISBN 9781450329576.

EVFIMIEVSKI, A.; GEHRKE, J.; SRIKANT, R. Limiting privacy breaches in privacy preserving data mining. *In*: **Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems**. New York, NY, USA: Association for Computing Machinery, 2003. (PODS '03), p. 211–222. ISBN 1581136706.

FEIGE, U.; FIAT, A.; SHAMIR, A. Zero-knowledge proofs of identity. **Journal of cryptology**, Springer, v. 1, n. 2, p. 77–94, 1988.

FINETTI, B. D. Funzione caratteristica di un fenomeno aleatorio. *In*: **Atti del Congresso Internazionale dei Matematici: Bologna del 3 al 10 de settembre di 1928**. Bologna: [*S. n.*], 1929. p. 179–190.

GREENBERG, A. Apple's 'differential privacy'is about collecting your data—but not your data. **Wired, June**, v. 13, 2016.

KHAN, M. A.; SALAH, K. Iot security: Review, blockchain solutions, and open challenges. **Future Generation Computer Systems**, Elsevier, v. 82, p. 395–411, 2018. ISSN 0167-739X.

KULLBACK, S.; LEIBLER, R. A. On information and sufficiency. **The annals of mathematical statistics**, JSTOR, v. 22, n. 1, p. 79–86, 1951.

LEAL, B. C.; VIDAL, I. C.; BRITO, F. T.; NOBRE, J. S.; MACHADO, J. C. δ-doca: achieving privacy in data streams. *In*: GARCIA-ALFARO, J.; HERRERA-JOANCOMARTÍ, J.; LIVRAGA, G.; RIOS, R. (ed.). **Data Privacy Management, Cryptocurrencies and Blockchain Technology**. Cham: Springer International Publishing, 2018. p. 279–295. ISBN 978-3-030-00305-0.

LEHMANN, E. L.; CASELLA, G. **Theory of point estimation**. [*S. l.*]: Springer Science & Business Media, 2006.

MCKINNEY Wes. Data Structures for Statistical Computing in Python. *In*: **Proceedings of the 9th Python in Science Conference**. [*S. l.*: *s. n.*], 2010. p. 56 – 61.

MCSHERRY, F. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 53, n. 9, p. 89–97, set. 2010. ISSN 0001-0782.

MOLINA-MARKHAM, A.; SHENOY, P.; FU, K.; CECCHET, E.; IRWIN, D. Private memoirs of a smart meter. *In*: **Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building**. New York: Association for Computing Machinery, 2010. p. 61–66. ISBN 9781450304580.

NETWORKING, C. V. Cisco global cloud index: forecast and methodology, 2016–2021. **White paper. Cisco Public, San Jose**, 2016.

OSTERREICHER, F.; VAJDA, I. A new class of metric divergences on probability spaces and its applicability in statistics. **Annals of the Institute of Statistical Mathematics**, Kluwer Academic Publishers, v. 55, n. 3, p. 639–653, 2003.

Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. **IEEE Internet of Things Journal**, v. 3, n. 5, p. 637–646, 2016.

STEIN, C. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. *In*: **Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics**. Berkeley, Calif.: University of California Press, 1956. p. 197–206.

SWEENEY, L. K-anonymity: A model for protecting privacy. **Int. J. Uncertain. Fuzziness Knowl.-Based Syst.**, World Scientific Publishing Co., Inc., USA, v. 10, n. 5, p. 557–570, out. 2002. ISSN 0218-4885.

TANBEER, S. K.; AHMED, C. F.; JEONG, B.-S.; LEE, Y.-K. Sliding window-based frequent pattern mining over data streams. **Information Sciences**, Elsevier, v. 179, n. 22, p. 3843–3865, 2009. ISSN 0020-0255.

TöNJES, R.; BARNAGHI, P.; ALI, M.; MILEO, A.; HAUSWIRTH, M.; GANZ, F.; GANEA, S.; KJæRGAARD, B.; KUEMPER, D.; NECHIFOR, S.; PUIU, D.; SHETH, A.; TSIATSIS, V.; VESTERGAARD, L. **Real Time IoT Stream Processing and Large-scale Data Analytics for Smart City Applications**. 2014. Poster presented at European Conference on Networks and Communications.

UK Power Networks. **SmartMeter Energy Consumption Data in London Households**. 2015. https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households. Accessed: 2019-06-28.

van der Walt, S.; Colbert, S. C.; Varoquaux, G. The numpy array: A structure for efficient numerical computation. **Computing in Science & Engineering**, IEEE Computer Society, v. 13, n. 2, p. 22–30, 2011.

VIRTANEN, P.; GOMMERS, R.; OLIPHANT, T. E.; HABERLAND, M.; REDDY, T.; COURNAPEAU, D.; BUROVSKI, E.; PETERSON, P.; WECKESSER, W.; BRIGHT, J. *et al.* Scipy 1.0: fundamental algorithms for scientific computing in python. **Nature Methods**, Nature Publishing Group, p. 1–12, 2020.

WANG, T.; BLOCKI, J.; LI, N.; JHA, S. Locally differentially private protocols for frequency estimation. *In*: **Proceedings of the 26th USENIX Conference on Security Symposium**. USA: USENIX Association, 2017. (SEC'17), p. 729–745. ISBN 9781931971409.

WARNER, S. L. Randomized response: A survey technique for eliminating evasive answer bias. **Journal of the American Statistical Association**, [American Statistical Association, Taylor & Francis, Ltd.], v. 60, n. 309, p. 63–69, 1965. ISSN 01621459.

WONG, A. K.; YOU, M. Entropy and distance of random graphs with application to structural pattern recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, PAMI-7, n. 5, p. 599–609, 1985.