N° d'ordre NNT : $2024ISAL0075$

THESE de DOCTORAT DE L'INSA LYON, membre de l'Université de Lyon

Ecole Doctorale ED 512
(InfoMaths)

Spécialité/ discipline de doctorat : Informatique

Soutenue publiquement le 27/09/2024, par :

Samuel Pélissier

Privacy-preserving communications for the IoT

Devant le jury composé de :

RIVANO	**Hervé**	Professeur des Universités	INSA Lyon	Président
MITTON	**Nathalie**	Directeur de Recherche	Centre Inria de l'Univ. de Lille	Rapporteuse
RASMUSSEN	**Kasper**	Professeur des Universités	Univ. of Oxford	Rapporteur
CARNEIRO VIANA	**Aline**	Directeur de Recherche	Centre Inria Saclay	Examinatrice
FRANCILLON	**Aurélien**	Professeur des Universités	EURECOM	Examinateur
CUNCHE	**Mathieu**	Professeur des Universités	INSA Lyon	Directeur de thèse
ROCA	**Vincent**	Chargé de recherche	Inria	Co-directeur de thèse

INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Département de la Formation par la Recherche et des Études Doctorales (**FEDORA**)

Bâtiment INSA direction, 1er étage
37, av. J. Capelle
69621 Villeurbanne Cédex
fedora@insa-lyon.fr

Référence : TH1128_PÉLISSIER Samuel

L'INSA Lyon a mis en place une procédure de contrôle systématique via un outil de détection de similitudes (logiciel Compilatio). Après le dépôt du manuscrit de thèse, celui-ci est analysé par l'outil. Pour tout taux de similarité supérieur à 10%, le manuscrit est vérifié par l'équipe de FEDORA. Il s'agit notamment d'exclure les auto-citations, à condition qu'elles soient correctement référencées avec citation expresse dans le manuscrit.

Par ce document, il est attesté que ce manuscrit, dans la forme communiquée par la personne doctorante à l'INSA Lyon, satisfait aux exigences de l'Etablissement concernant le taux maximal de similitude admissible.

INSA LYON
Campus LyonTech La Doua
20, avenue Albert Einstein - 69621 Villeurbanne cedex - France
Tél. +33 (0)4 72 43 83 83 - Fax +33 (0)4 72 43 85 00
www.insa-lyon.fr

membre de UNIVERSITÉ DE LYON

Département FEDORA – INSA Lyon - Ecoles Doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
ED 206 CHIMIE	**CHIMIE DE LYON** **https ://www.edchimie-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage **secretariat@edchimie-lyon.fr**	**M. Stéphane DANIELE** C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 Boulevard du 11 novembre 1918 69616 Villeurbanne **directeur@edchimie-lyon.fr**
ED 341 E2M2	**ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION** **http ://e2m2.universite-lyon.fr** Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 **secretariat.e2m2@univ-lyon1.fr**	**Mme Sandrine CHARLES** Université Claude Bernard Lyon 1 UFR Biosciences Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69622 Villeurbanne CEDEX **e2m2.codir@listes.univ-lyon1.fr**
ED 205 EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTÉ** **http ://ediss.universite-lyon.fr** Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 **secretariat.ediss@univ-lyon1.fr**	**Mme Sylvie RICARD-BLUM** Laboratoire ICBMS - UMR 5246 CNRS - Université Lyon 1 Bâtiment Raulin - 2ème étage Nord 43 Boulevard du 11 novembre 1918 69622 Villeurbanne Cedex Tél : +33(0)4 72 44 82 32 **sylvie.ricard-blum@univ-lyon1.fr**
ED 34 EDML	**MATÉRIAUX DE LYON** **http ://ed34.universite-lyon.fr** Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 **yann.de-ordenana@ec-lyon.fr**	**M. Stéphane BENAYOUN** Ecole Centrale de Lyon Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully CEDEX Tél : 04.72.18.64.37 **stephane.benayoun@ec-lyon.fr**
ED 160 EEA	**ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE** **https ://edeea.universite-lyon.fr** Sec. : Philomène TRECOURT Bâtiment Direction INSA Lyon Tél : 04.72.43.71.70 **secretariat.edeea@insa-lyon.fr**	**M. Philippe DELACHARTRE** INSA LYON Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 **philippe.delachartre@insa-lyon.fr**
ED 512 INFOMATHS	**INFORMATIQUE ET MATHÉMATIQUES** **http ://edinfomaths.universite-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 **infomaths@univ-lyon1.fr**	**M. Hamamache KHEDDOUCI** Université Claude Bernard Lyon 1 Bât. Nautibus 43, Boulevard du 11 novembre 1918 69 622 Villeurbanne Cedex France Tél : 04.72.44.83.69 **direction.infomaths@listes.univ-lyon1.fr**
ED 162 MEGA	**MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE** **http ://edmega.universite-lyon.fr** Sec. : Philomène TRECOURT Tél : 04.72.43.71.70 Bâtiment Direction INSA Lyon **mega@insa-lyon.fr**	**M. Etienne PARIZET** INSA Lyon Laboratoire LVA Bâtiment St. Exupéry 25 bis av. Jean Capelle 69621 Villeurbanne CEDEX **etienne.parizet@insa-lyon.fr**
ED 483 ScSo	**ScSo**[1] **https ://edsciencessociales.universite-lyon.fr** Sec. : Mélina FAVETON Tél : 04.78.69.77.79 **melina.faveton@univ-lyon2.fr**	**M. Bruno MILLY** (INSA : J.Y. TOUSSAINT) Univ. Lyon 2 Campus Berges du Rhône 18, quai Claude Bernard 69365 LYON CEDEX 07 Bureau BEL 319 **bruno.milly@univ-lyon2.fr**

1. ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Contents

List of Figures

List of Tables

Remerciements

Trois ans auparavant, je n'aurais pu imaginer une fraction de ce qu'il m'a été possible de vivre. Au-delà de la richesse en apprentissages et en émotions, cette thèse n'aurait pas abouti sans l'aide, la patience et la bienveillance de nombreuses personnes.

Je souhaiterais tout d'abord remercier mes encadrants, Mathieu Cunche et Vincent Roca, pour leurs précieux conseils et leur soutien indéfectible ; merci d'avoir fait ce pari et offert la chance de découvrir la recherche dans un tel environnement. Merci à Didier Donsez pour ses innombrables idées et son éternel enthousiasme, ainsi qu'à Abhishek, qui m'a tant appris en si peu de temps. Merci à Jan et Thomas pour les pauses eau-café, les projets d'arrosage fonctionnels-mais-étranges, et le camping. Plus généralement, merci à toute l'équipe Privatics pour son accueil chaleureux et les discussions passionnantes.

Je n'aurais jamais pu débuter l'aventure, et encore moins la continuer, sans ma famille. Merci à mes parents, mes frères, mes grands-parents et M. pour votre amour au quotidien. Panou, j'aurais aimé pouvoir te montrer le manuscrit terminé ; tu aurais sûrement suggéré de le traduire en français ou en espagnol.

Enfin, merci à mes ami·e·s J., M. et A. pour m'avoir aidé à naviguer trois années étranges et rebondissantes. Merci au cercle des imbéciles anonymes et au chaudron, certainement piliers du bon déroulement de cette thèse, et sans qui j'aurais probablement pu tout faire en deux ans environ.

Résumé

Les dernières décennies ont été témoins de l'émergence et de la prolifération d'objets connectés, communément appelés *Internet des Objets (IdO)*. Cet écosystème divers correspond à une large gamme de dispositifs spécialisés, allant de la caméra IP aux capteurs détectant les fuites d'eau, chacun conçu pour répondre à des objectifs et des contraintes de consommation d'énergie, de puissance de calcul ou de coût. Le développement rapide de nombreuses technologies et leur connexion en réseau s'accompagne de la génération d'un important volume de données, soulevant des préoccupations en matière de vie privée, en particulier dans des domaines sensibles tels que la santé ou les maisons connectées.

Dans cette thèse, nous exploitons les techniques d'apprentissage automatique (*machine learning*) pour explorer les problèmes liés à la vie privée des objets connectés via leurs protocoles réseau. Tout d'abord, nous étudions les attaques possibles contre LoRaWAN, un protocole longue distance et à faible coût d'énergie. Nous explorons la relation entre deux identifiants du protocole et montrons que leur séparation théorique peut être contrecarrée en utilisant les métadonnées produites lors de la connexion au réseau. En nous appuyant sur une approche multi-domaines (contenu, temps, radio), nous démontrons que ces métadonnées permettent à un attaquant d'identifier les objets connectés de manière unique malgré le chiffrement du trafic, ouvrant la voie au traçage ou à la ré-identification.

Nous explorons ensuite les possibles contre-mesures, en analysant systématiquement les données utilisées lors de ces attaques et en proposant des techniques pour les obfusquer ou réduire leur pertinence. Nous démontrons que seule une approche combinée offre une réelle protection. Par ailleurs, nous proposons et évaluons diverses solutions de pseudonymes temporaires adaptées aux contraintes de LoRaWAN, en particulier la consommation énergétique.

Enfin, nous adaptons notre méthodologie d'apprentissage automatique à DNS, un protocole largement déployé dans l'IdO grand public. À nouveau basées sur les métadonnées, notre attaque permet d'identifier les objets connectés, malgré le chiffrement du flux DNS-over-HTTPS. Explorant les contre-mesures potentielles, nous observons un non-respect des standards liés au padding, entraînant la compromission partielle de la vie privée des utilisateurs.

Plus généralement, nos travaux mettent en évidence que les efforts déployés par les protocoles IdO tels que LoRaWAN pour protéger la vie privée sont insuffisants. Des évolutions potentiellement profondes sont nécessaires pour correctement répondre à ces enjeux.

Mots-clefs : Vie privée; Internet des Objets; Objets connectés; Sécurité; Protocoles; Radio; Réseaux sans fil; Métadonnées; Empreinte; Re-identification; Inférence d'activités; LoRaWAN; LPWAN; DNS; DNS-over-HTTPS.

List of publications

Accepted

International conferences

- [165] Samuel Pélissier, Jan Aalmoes, Abhishek Kumar Mishra, Mathieu Cunche, Vincent Roca, and Didier Donsez. Privacy-preserving pseudonyms for LoRaWAN. *Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2024.
- [166] Samuel Pélissier, Mathieu Cunche, Vincent Roca, and Didier Donsez. Device reidentification in LoRaWAN through messages linkage. *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2022. Code and data available: `https://gitlab.inria.fr/spelissi/wisec-2022-reproductibility`.

National conferences

- Samuel Pélissier, Jan Aalmoes, Abhishek Kumar Mishra, Mathieu Cunche, Vincent Roca, and Didier Donsez. Privacy-preserving pseudonyms for LoRaWAN. *14ème Atelier sur la Protection de la Vie Privée (APVP)*, 2024.
- Samuel Pélissier, Mathieu Cunche, Vincent Roca, and Didier Donsez. Introducing privacy-preserving identifiers in LoRaWAN. Poster *LPWAN Days*, 2023.
- Samuel Pélissier, Mathieu Cunche, Vincent Roca, and Didier Donsez. Linkage attacks and countermeasures in LoRaWAN. *13ème Atelier sur la Protection de la Vie Privée (APVP)*, 2023.
- Samuel Pélissier, Mathieu Cunche, Vincent Roca, and Didier Donsez. Device reidentification in LoRaWAN through messages linkage. *12ème Atelier sur la Protection de la Vie Privée (APVP)*, 2022.

Under review

- [167] Samuel Pélissier, Abhishek Kumar Mishra, Mathieu Cunche, Vincent Roca, and Didier Donsez. Introducing Multi-Domain Fingerprints for Lorawan Device Linkage. Submitted to the *International Journal for the Computer and Telecommunications Industry (Computer Communications)*.
- Samuel Pélissier, Gianluca Anselmi, Abhishek Kumar Mishra, Anna Maria Mandalari, Mathieu Cunche. Does Your Smart Home Tell All? Assessing the Impact of DNS Encryption on IoT Device Identification. Submitted to the *20th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2024.
- Abhishek Kumar Mishra, Samuel Pélissier, and Mathieu Cunche. Fingerprinting connected Wi-Fi devices using per-network MAC addresses. Submitted to the *Workshop on Security and Artificial Intelligence (SECAI)*, 2024.

Part I

Introduction and Context

Chapter 1

Introduction

Contents

During the past decades, we have witnessed the emergence of connected devices, commonly referred to as the Internet-of-Things (IoT). Unlike general-purpose, consumer-grade computers, IoT is not a monolithic concept with a singular, clear definition. Instead, it represents a diverse ecosystem corresponding to a wide range of specialized devices, each designed to meet specific objectives and constraints, whether it is energy efficiency, computational power, or affordability. For instance, IoT encompasses devices as varied as wired IP cameras deployed in smart home environments and wireless water leakage sensors installed in industrial settings. Despite their disparate functionalities and deployment scenarios, both examples fall under the umbrella of IoT devices.

By definition, these devices are connected to their surroundings through networks of various topologies and technologies. This connectivity leads to the generation of specialized data, much of which raises privacy concerns, especially in sensitive domains such as health monitoring or activity tracking in smart homes.

In parallel to the expanding capabilities and ubiquitous deployment of IoT, machine learning techniques have proved to be powerful tools for automating the analysis of large volumes of data and showcasing privacy attacks. By applying these methods to the network traffic generated by IoT devices, we study network protocols and uncover new insights into their vulnerabilities to various privacy attacks.

In this thesis, we leverage such tools to explore privacy issues in IoT devices through the lens of their communications. With a specific focus on the highly constrained LoRaWAN protocol, we study the trade-offs between privacy and energy consumption. We complete this analysis by adapting our machine learning methodology to encrypted DNS, a battle-tested and widely deployed protocol in consumer-grade IoT devices.

1.1 Challenges and contributions

The central theme of this thesis is organized around three main research questions, leading to various contributions. Figure 1.1 illustrates a summary of the overall process. To support it, we conduct a thorough state of the art on eavesdropping-enabled attacks and associated defenses in IoT networks.

Q_1 What are the privacy challenges of the LoRaWAN protocol?

LoRaWAN is a recent protocol, which saw most of its related research focus on either applicability to specific use-cases [34, 38, 96, 114, 137, 147, 175, 187, 192, 199, 204] or security [99, 100, 154, 159, 212]. On the other hand, privacy in this context have seldom been

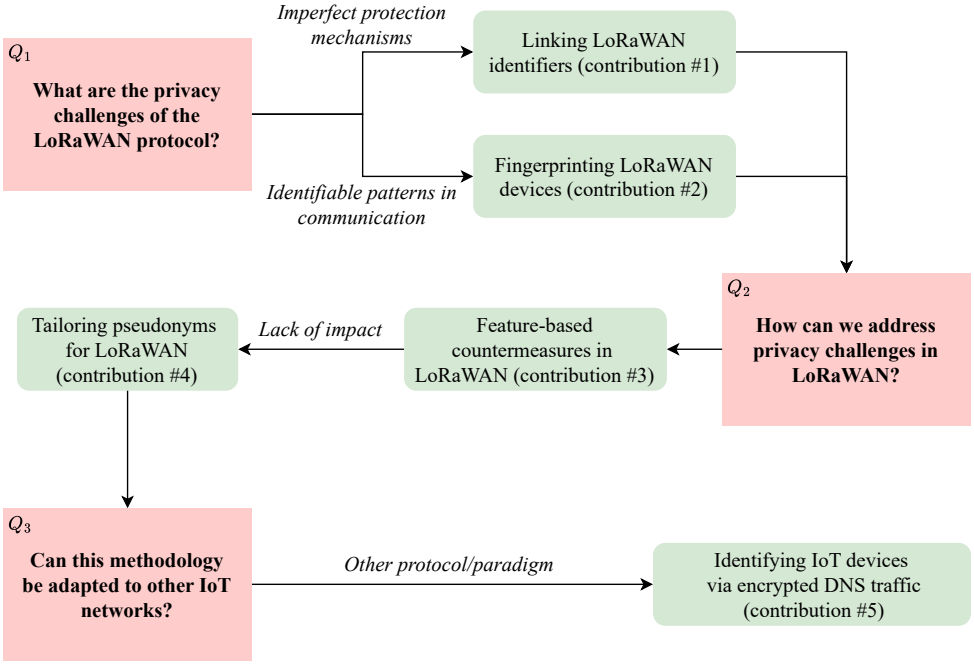


Figure 1.1: Thesis research questions and contributions.

studied [74, 131, 196], with major features lacking attention from the community. By focusing on identifiers leveraged by the protocol, we identify two possible privacy threats during the lifetime of a LoRaWAN device: imperfect protection mechanisms in the join process, and identifiable patterns in the actual transmission.

Contribution #1: Linking LoRaWAN identifiers (Chapter 4).

- We analyze the LoRaWAN join process mechanism and identify keys features leading to a potential linkage.
- We leverage a large-scale dataset of millions of LoRaWAN messages, captured over three years to test our hypothesis. To the best of our knowledge, this is the most comprehensive dataset studied in the literature.
- We utilize a robust machine learning method and reliably link two seemingly unrelated messages and their identifiers (`DevEUI` and `DevAddr`).

Contribution #2: Fingerprinting LoRaWAN devices (Chapter 5).

- We provide a high-level analysis of network fingerprinting related works and extensively compare their solutions.
- We propose a new fingerprinting representation and evaluate it on LoRaWAN traffic. Our machine learning method is able to reliably re-identify the origin of sequences of messages via a multi-domain approach leveraging content, time, and radio-based features.
- By exploiting a real-world, large-scale dataset of third-party operators, we explore various attack scenarios, including a limited number of received messages, the availability of listening stations, and access to radio information when targeting mobile devices.

Q_2 How can we address privacy challenges in LoRaWAN?

Following the discovery of two privacy threats in the LoRaWAN protocol, we study how they could be mitigated through easily deployable software-based solutions.

Contribution #3: Feature-based countermeasures in LoRaWAN (Chapter 6).

- We design a set of mitigations for each feature previously identified as posing a threat against privacy, and respecting the constraints of the protocol. This includes padding, delay, and radio power modulation.
- Through careful evaluation on the same datasets used for conducting attacks, we measure the individual and combined impact of such countermeasures.
- We study their inherent overhead and outline how both their limited effect on attacks performance and restrictions of the LoRaWAN protocol restrict their potential viability.

Based on the marginal influence of feature-based countermeasures and on the presence of stable identifiers in LoRaWAN, we propose an alternative, privacy-oriented, design of pseudonyms specifically for this protocol.

Contribution #4: Tailoring pseudonyms for LoRaWAN (Chapter 7).

- We study multiple state-of-the-art pseudonyms solutions in other wireless technologies, including Wi-Fi, BLE, and VANETs, and compare their existing design based on various targeted properties.
- Given the constraints of LoRaWAN, we tailor two solutions for the protocol and propose a renewal strategy to rotate pseudonyms.
- We thoroughly assess their performance and intrinsic limitations via a set of theoretical and simulation-based evaluations.
- Finally, we reflect on possible adaptations in other Low Power Wide Area Network protocols under similar constraints.

Q_3 Can this methodology be adapted to other IoT networks?

While our machine learning methodology demonstrates great efficiency for LoRaWAN, it is interesting to test in a different context. DNS is utilized by a high number of IoT devices, operating in less constrained environments, such as smart homes [203]. By following a similar methodology, we show the robustness of our approach to detect privacy vulnerabilities in a different network protocol.

Contribution #5: Identifying IoT devices via encrypted DNS traffic (Chapter 8).

- We analyze potential contenders for encrypted DNS in IoT devices and choose to focus on DNS-over-HTTPS (DoH).
- Only leveraging metadata from the network traffic, we evaluate machine learning methods and find that device identification is both possible and reliable despite encryption.
- We study possible countermeasures and analyze various padding strategies via simulations.
- Coincidentally, we find that multiple DNS resolvers do not respect relevant standard, failing to correctly protect the privacy of their users.

1.2 Outline

This thesis is structured as follows. Chapter 2 presents a state of the art regarding privacy in the IoT, both from an offensive and defensive perspective. Additional elements on the LoRaWAN protocol and machine learning are summarized in Chapter 3.

The first part of this thesis proposes some answers to Q_1 *What are the privacy challenges of the LoRaWAN protocol?*. Chapter 4 deals with linking theoretically unrelated identifiers during the join process, allowing an eavesdropper to associate a device identity with its

activity. We design a second attack in Chapter 5, and fingerprint devices based on their communication patterns.

The second part explores the second research question, Q_2 *How can we address privacy challenges in LoRaWAN?*, studying possible countermeasures. Chapter 6 applies various noises to machine learning features in order to reduce their susceptibility to linkage and identification attacks. A more radical and impactful solution is presented in Chapter 7, with the design of rotating pseudonyms for LoRaWAN.

The third - and last - part includes Chapter 8, answering Q_3 *Can this methodology be adapted to other IoT protocols?*, by adjusting the machine learning methodology to identify IoT devices via encrypted DNS, despite its encryption.

Finally, Chapter 9 concludes this thesis, summarizing our contributions and hinting at both short and long-term perspectives.

1.3 Funding

This thesis was carried out within the CITI laboratory (INSA Lyon & Inria), in the Privatics team (Inria). It was funded by the ANR-BMBF PIVOT project (ANR-20-CYAL-0002), and supported by the H2020 SPARTA project, as well as the IoT SPIE ICS - INSA Lyon chair.

Chapter 2

State of the art

Contents

Building upon the thesis outline and research questions presented in the previous section, we delve deeper into the multifaceted realm of the IoT privacy. Through this exploration, we aim to offer an overview of various techniques, both offensive and defensive, related to the specific context of IoT privacy, while setting reasonable boundaries to such a broad scope. We begin by highlighting the inherent challenges and security vulnerabilities of IoT ecosystems. Next, we explore the nuanced concept of privacy in the context of IoT, exploring its various dimensions and implications. Subsequently, we dissect the prevalent attacks targeting IoT privacy, outlining a generic eavesdropping threat model and the main methodologies employed by adversaries. Finally, we analyze the countermeasures and mitigation strategies deployed to combat these privacy attacks.

2.1 Security in wireless communications for the IoT

The open nature of wireless communication makes the IoT ecosystem particularly vulnerable to various security threats. In recent years, eavesdropping on largely deployed protocols such as Wi-Fi [198] and Bluetooth [20] has become increasingly easy thanks to affordable off-the-shelf hardware and Software-Defined Radios (SDRs). Both active and passive attacks can be conducted on open, user-facing networks with minimal investments. Previous vulnerabilities include denial-of-service [45], forging authentication messages [126], man-in-the-middle

(MitM) [41, 154], MAC spoofing attacks [41], and many others [159]. Emerging IoT protocols such as LoRaWAN suffer from similar weaknesses, from the process allowing devices to join a network, to data transmission [100, 154]. For instance, a bit flipping attack is possible by design as the payload authentication is not checked by the application server [100]. Although newer versions tackle most of these vulnerabilities, a few still remain, such as denial-of-service attacks [212, 99].

A significant portion of security research focuses on data integrity and confidentiality. Confidentiality, achieved through encryption and access controls, plays a crucial role in preventing unauthorized access to sensitive information. It is important to note that privacy cannot exist without encryption and the confidentiality measures it provides. However, privacy extends beyond mere confidentiality by encompassing metadata generation and individuals' rights to control their personal data, regardless of security mechanisms. As explored further in Section 2.2, it includes broader ethical and legal considerations regarding data collection, use, and sharing.

2.2 Privacy for the IoT

Our work is heavily immersed in technology and highly specialized, to the extent that discerning human involvement behind network traces can be challenging. In this section, we highlight its broader context by first proposing an IoT-centric definition of privacy, considering historical, sociological, and technological perspectives. Then, we present relevant texts of law in our jurisdiction, the European Union (EU), before highlighting associated guidelines and standards.

2.2.1 From human to data

According to Serge Gutwirth, privacy is a "*cornerstone of contemporary [...] society*", while "*[remaining] out of the grasp of every academic chasing it*" [94]. Its perception fluctuates through History and other cultural contexts, as each social group adapts the concept to its customs. For instance, it would be foolish to directly compare modern views on privacy with the ones held during Roman antiquity, when communal living with less distinct personal boundaries was the norm [52]. Likewise, the interpretation of privacy is highly contextual [158, 193]: teenagers over-sharing on the Internet generally do not want their parents to know about their *personal* information [51].

As they are social constructs, we argue privacy definitions follow the technologies developments of the authors' era. Seminal article "*The Right To Privacy*" published in 1890 defines privacy as "*the right to be left alone*" [221]. This somewhat limited vision is usually complemented by Westin's 1968 book, motivated by advances in surveillance technology: "*privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others*" [223, Chap. 1]. Inheriting from this information-centric definition, multiple high-level classifications have emerged during the past decades, adding more *types* of privacy. Clark highlights its multidimensional aspect by introducing privacy of the person (or bodily privacy), privacy of personal data, privacy of personal behavior and privacy of personal communication [60]. Finn et al. update these categories to account for new advancements, up to 7 types [82]. For instance, it includes privacy of thoughts and feeling, to protect them from emerging neurodata innovations.

As a new technology, the IoT borrows from existing definitions while enhancing them with its own requirements [139]. Inspired by Finn et al. [82] and Hahn et al. [95], we select three types of privacy relevant to data produced by IoT devices:

- *Activity*: what kind of actions is performed, and when.
- *Location*: where the action is performed, or more generally, where the device is used.
- *Identity*: who is using the device, or more generally, which device is used.

For instance, when capturing traffic from a device using the address A and known to be a glucose monitor [55], the *identity* corresponds to A and the *activity* to monitoring blood glucose levels. We can also infer the *location* based on radio signals (see Section 2.3.3). Of

course, such a baseline could be leveraged to infer additional information: a glucose monitor is probably used by someone with diabetes, and the attacker could know who uses the device with address A.

In a broader sense, we examine the privacy of IoT users through the lens of their *informational identities* [218]. Instead of a single identifier (e.g. name only), the identity is considered as a set of multidimensional information forming a profile (e.g. behavior, preferences) [83, 184].

2.2.2 Privacy regulations

While privacy may appear abstract, its modern foundation lies in various legal documents designed to ensure responsible handling of collected personal data. Privacy laws serve to safeguard personal data from misuse, breaches, and unauthorized surveillance [218].

Privacy protection is a concern for legislators worldwide, leading to the establishment of comprehensive frameworks in numerous countries [46], such as the *California Consumer Privacy Act* (CCPA) in the United States of America, or the more recent *Digital Personal Data Protection Act* (DPDPA-2023) in India. Likewise, more low-level, technical legal documents are pushed to regulate IoT devices: United Kingdom's *Product Security and Telecommunications Infrastructure (Security Requirements for Relevant Connectable Products) Regulations* enacted in 2023 describes various security requirements for manufacturers, including passwords complexity and support periods [163].

In this section, we outline EU-based documents, because 1) it is geographically relevant to our work, and 2) the legal framework is well-defined, comparatively advanced compared to the other countries. Its main component has been deployed for a few years and, following (and completed by) decades of national laws, such as the 1978's "*Informatique et Libertés*" (Information Technology and Liberty) French law [6]. We provide an overview of two major legal documents: the currently enacted General Data Protection Regulation (GDPR), and a future implementation of the ePrivacy Regulation (ePR).

The GDPR is a comprehensive data protection law enacted by the EU to safeguard individuals' privacy rights and regulate the processing of *personal data* [209]. GDPR imposes strict requirements on how data collected by these interconnected technologies is handled, emphasizing principles such as data minimization, purpose limitation, and transparency. IoT device manufacturers and service providers with customers from the EU are obligated to ensure that personal data processing activities comply with GDPR provisions, including obtaining informed consent from users, implementing appropriate security measures, and facilitating individuals' rights to access, rectify, and erase their personal data [209].

The ePR is a proposed EU legislation intended to complement the GDPR by specifically addressing privacy in electronic communications [62].[1] While still under interinstitutional negotiations, the ePR is expected to have significant implications for IoT devices. Contrary to the GDPR, it covers the large scope of electronic communications, independently whether it is personal or non-personal data. This also includes metadata, defined by Article 4 as "*data processed in an electronic communications network for the purposes of transmitting, distributing or exchanging electronic communications content; including data used to trace and identify the source and destination of a communication, data on the location of the device generated in the context of providing electronic communications services, and the date, time, duration and the type of communication*". Additionally, Article 8 is particularly relevant to our work on consumer IoT devices. Titled "*Protection of information stored in and related to end-users' terminal equipment*", it applies GDPR concepts, such as purpose and consent, to processing and storage capabilities of IoT devices, expanding the scope of data privacy regulations in the EU. The broad approach of the ePR offers compelling avenues for both legal and technical research.

Our work follows the trend of current and future regulations requiring companies to prioritize data protection, implement robust security measures, and uphold transparency standards to ensure compliance and safeguard individuals' privacy in the IoT ecosystem.

[1] The ePrivacy *Directive* has been active since 2002 and inspires the proposed *Regulation*. A directive is a goal that member states should achieve, without defined means to achieve said results. It is implemented nationally as laws and regulations. On the other hand, a European *regulation* is effective as law in all member states at once (e.g. GDPR).

2.2.3 Guidelines and standards

In addition to legal frameworks, multiples non-biding documents containing advices and good practices have been published in recent years, both as general guidelines and technical standards. Often providing a step-by-step translation of legal documents, they help manufacturers comply with regulations and developers implement specific privacy-enhancing technologies.

At a high level, guidelines provided by the European Data Protection Board (EDPB) clarify the legal framework established by the GDPR, providing concrete examples for better understanding. More specifically, many documents are driven by the Article 35 of the GDPR, requiring a Data Protection Impact Assessment (DPIA) when a processing is "*likely to result in a high risk to the rights and freedoms of natural persons*" [209]. For instance, the European Data Protection Authorities provide a unified interpretation on DPIA, to ensure compliance with GDPR requirements [164]. Additionally, the French administrative regulatory body (Commission Nationale de l'Informatique et des Libertés (CNIL)) published guidelines on how to conduct a Privacy Impact Assessment specifically for IoT devices. They detail the process, from defining the processed data to associated feared events, and provide advices to manufacturers on how to mitigate privacy risks for their products [61].

Other documents are less compliance-motivated yet remain impactful, including protocol standards such as Requests For Comments (RFCs) for the Internet or 802.11 for Wi-Fi. RFCs are published by Internet Engineering Task Force (IETF) and written by individuals or organizations, documenting methods, behaviors, research, or innovations applicable to the operation of the Internet and Internet-connected systems. Not all RFCs are standards; they are organized by various status: Standards Track, Best Current Practice, Informational, Experimental, or Historic. For instance, multiple RFCs detail privacy considerations [226] (Informational) and privacy-preserving mechanisms [144, 145] (Standard track and Experimental) for the Domain Name System (DNS) protocol, later implemented by developers.

The IEEE 802.11 is a set of technical standards specifying physical and MAC layers of wireless network protocols, such as Wi-Fi. It includes recommended practice for privacy [63], such as a threat model and a checklist for protocol designers. Additional work on privacy is ongoing with two specification amendments: "*Operation with Randomized and Changing MAC Addresses*" (P802.11bh) and "*Enhanced Service with Data Privacy Protection*" (P802.11bi), that should be unveiled in late 2024.[2]

These documents form an ever-expanding corpus, following both legal regulations and technology innovations. We hope our work acts as inspiration and support for new and improved privacy-preserving solutions, and even more accurate guidelines.

2.3 Attacks on IoT privacy

In this section, we first outline the threat model under which we operate, before discussing how each type of privacy presented in Section 2.2.1 is associated with a specific class of attacks. An overview of the attack hierarchies is available in Figure 2.1.

2.3.1 Threat model

Multiple threat models for both security and privacy are detailed in the IoT literature [81, 206, 222]. At a high level, they can be analyzed through two axes: the capabilities of the attacker, and which element of the network is targeted. For instance, an attacker can be passive (only listening to communications) or active (injecting new frames, altering existing or replaying previously transmitted ones); they can benefit from full physical access to IoT devices or be limited to eavesdropping. Likewise, Torres et al. [206] define 6 attack vectors against IoT networks, based on the target: the environment around the sensor, the device itself, the medium between the device and the gateway, the gateway itself, the link between the gateway and the server, and the server itself.

Analyzing a complete infrastructure at once is an overwhelming task, so we restrict their model and focus on a specific class of privacy attacks enabled by the very nature of open wireless communication. As seen in Figure 2.2, any attacker having access to the medium, for

[2] https://www.ieee802.org/11/PARs/P802.11bi.pdf

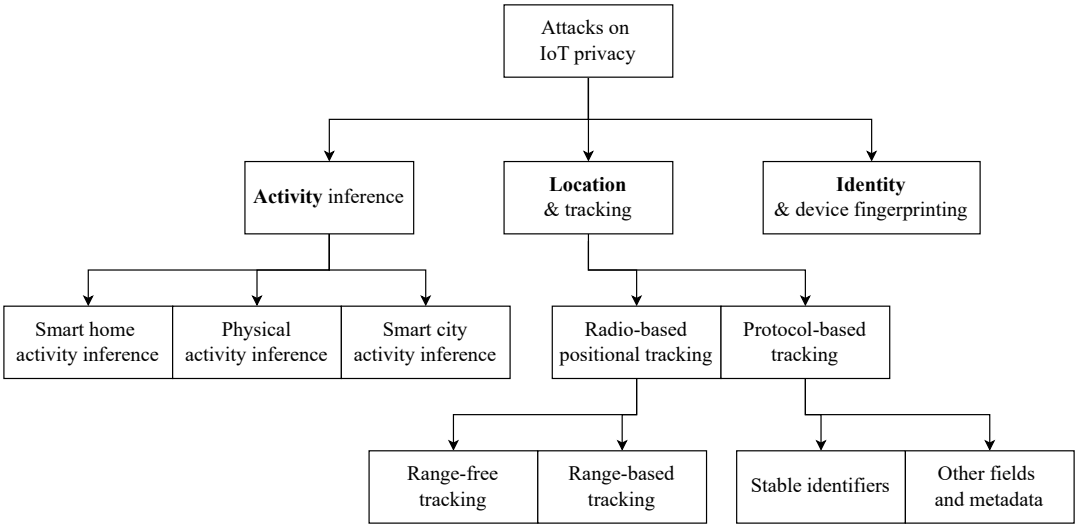


Figure 2.1: Overview of privacy attacks against the IoT.

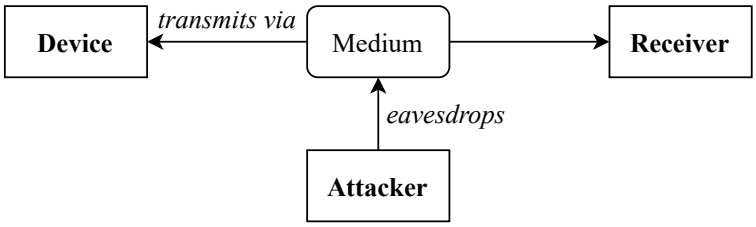


Figure 2.2: Threat model overview.

instance via their own gateway or off-the-shelf hardware, can eavesdrop on communications between devices and receivers.

Our threat model is thus defined as follows. First, every other part of the infrastructure is trusted [222], including device, gateways, and servers physical and software integrity. Second, contrary to scenarios detailed by Torres et al. [206], the attacker is only passive. For instance, they cannot alter existing frames, inject new ones in the communication, jam it, or replay previous exchanges. This is a reasonable assumption as messages generally benefit from integrity protection and counters are used to detect replays (see Section 3.1.3). Third, we assume the cryptographic layer is secured: it is impossible to access the content of encrypted payloads, and the integrity as well as authentication are robust. Following sections present various attacks under these assumptions.

2.3.2 Activity inference

As outlined in Section 2.2.1, actions executed via IoT devices may be directly tied with people's activities, and the data they produce can reveal user behaviors, preferences, and patterns [131]. Activity inference typically falls into three categories: smart home, human, and smart city. While these align with expected applications for official purposes, malicious actors can leverage network traffic patterns to extract this information as well.

2.3.2.1 Smart home activity

Analyzing sensor data from devices like motion detectors, door sensors, and smart thermostats enables the extraction of occupancy patterns within a home, which could provide valuable information to potential thieves [15, 27, 66, 131, 186]. For instance, Copos et al. show that application traffic (HTTP, DNS, NTP, TLS protocols) coming from a thermostat as well as a smoke and carbon dioxide detector can be used to infer home occupancy [66]. Their work is further expanded to other smart home appliances (e.g. camera switch, lightbulb, home assistants) via traffic rate analysis by Apthorpe et al. [27].

Such attacks also concern lower level and more IoT specific protocols such as ZigBee [186] or LoRaWAN [131]. Interesting information can be gleaned not only from the traffic content but also from its structure: Leu et al. demonstrate that if a real-life event is directly linked to network messages, a single frame is enough to infer a device state (e.g. a garage door opening) [131]. Although Copos et al. extract features from clear-text protocols such as DNS [66], following works only leverage metadata such as length and timing of messages [27, 131]. For instance, Acar et al. provide a detailed analysis on smart home activity inference via encrypted traffic, from device activation to movements in the house [15]. They distinguish between time-independent, non-sequential events (e.g. device power cycling) and time-dependent, sequential events (e.g. movement between locations), shedding light on the intricacies of activity inference in smart home environments.

2.3.2.2 Human activity

Wearable devices equipped with sensors like accelerometers and gyroscopes can track and infer various human activities such as walking, running, cycling, or even specific exercises [123]. Fitness trackers are generally connected via Bluetooth Low Energy (BLE) to smartphone applications, letting eavesdroppers monitor their encrypted traffic. Das et al. show that each activity has distinctive patterns (e.g. the number of messages increases with the physical activity intensity), making it possible to re-identify them [72]. External attacks are also possible via adversarial motion detection through wireless signals. For instance, existing Wi-Fi signal can be used to reliably detect users paths in specific rooms [198].

2.3.2.3 Smart city activity

Connected city infrastructure enables large-scale monitoring of various aspects such as traffic flow, air quality, and energy consumption [199]. However, attackers are also able to infer various information based on the generated traffic. For instance, frequent smart metering measurements can be leveraged to infer highly precise information, such as household occupancy, lifestyle, economic status [32], or even which TV program is currently watched [91]. Likewise, Dujicrodic et al. expose that signal power fluctuations of parking sensors used to monitor space occupancy in real-time are enough to discern patterns of user activity [74]. While we focus on users' privacy, smart city activity inference is also relevant in industrial sites.

2.3.3 Radio-based positional tracking

IoT devices embedded in everyday objects, like smartphones, wearables, or smart vehicles, maintain constant communication with their surroundings. Whether they are scanning to detect other devices and receivers, or transmitting data within established connections, they generate numerous messages that can be intercepted by eavesdroppers. Such information is easily associated with location data to enable positional tracking, for instance via known receivers.

Radio-based tracking of IoT devices involves using Radio Frequency (RF) signals emitted by the devices to determine their location. By nature, this approach is relevant to any wireless network protocol. Indoor and outdoor positioning are dynamic and intricate areas of study, with advanced techniques typically relying on two fundamental approaches: *range-free* and *range-based* methodologies. [135].

2.3.3.1 Range-free tracking

Range-free positioning consists of two phases. First, a collection of known positions is mapped with radio features of receivers, which are then converted into fingerprint positions. In Figure 2.3, they are denoted as FP. Such fingerprints can take on different formats, such as mean or distributions of Received Signal Strength Indicator (RSSI) values [25, 135, 174]. Second, when determining the location of a device, its fingerprint position is compared to the existing dataset to identify the closest match [135].

The accuracy of fingerprint-based approaches highly depends on the *discrete* dataset quality, both at its creation and in time [135]. For instance, this solution may not work if a device

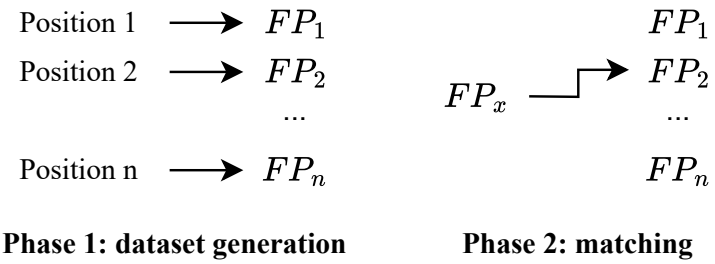
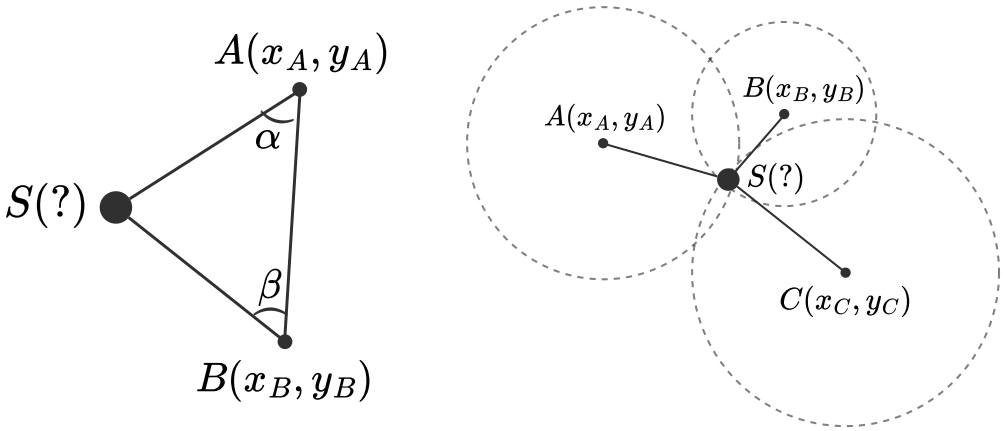


Figure 2.3: Range-free positioning process.

is located somewhere that was not previously surveyed. Hernández et al. estimate the location of non-mapped fingerprints for indoor Wi-Fi positioning through *continuous* dataset thanks to support vector machines [98]. From an attacker's point of view, this approach allows reducing by half the number of surveyed positions used to train the model while keeping the same mean distance error [98].

2.3.3.2 Range-based tracking

Range-based positioning is more robust to unknown settings as it does not exploit a pre-generated dataset. Figure 2.4 present the two main techniques: *triangulation* and *trilateration* [79, 124]. In both cases, the receivers positions (A, B, or C) are known, and we search for the location of a source device S.



(a) Triangulation. (b) Trilateration.

Figure 2.4: Range-based positioning.

As seen in Figure 2.4a, triangulation requires two receivers with directional antennas. The position of S is computed through trigonometry using the Angle of Arrival (AoA) at both A and B, and the distance $d(A, B)$. Triangulation based on AoA requires less hardware compared to other methods, but its effectiveness is optimized in scenarios with clear line-of-sight [172]. As the distance between the device and receivers increases, the accuracy of AoA-based triangulation tends to decrease [172].

Figure 2.4b presents trilateration based on 3 receivers with non-directional antennas using Time of Arrival (ToA). The distance between each receiver and S is obtained via subtraction of a timestamp included in the frame sent by S and the timestamp of arrival at A, B, and C. To precisely compute the position, time synchronization is required between device and receivers, which can be challenging [79, 135]. Such a technique falls outside the scope of our threat model, as it requires a close synchronization with the device, that should not be possible for a passive attacker.

Multilateration expands such a concept by including more than 3 receivers [124]. Another improvement also involves computing the distance based on relative time instead of absolute

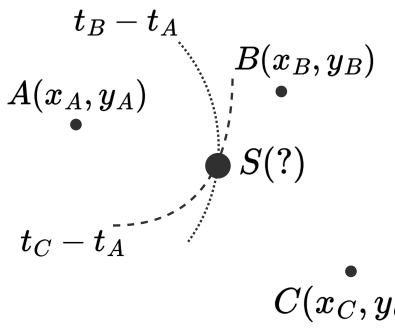


Figure 2.5: TDoA-based positioning, with t_r the time of arrival at a receiver r.

time. Figure 2.5 illustrates how the Time Difference of Arrival (TDoA) can be computed at each receiver to find the position of S, *without time synchronization* (i.e. without collaboration from S). Fargas and Petersen leverage such a method in LoRaWAN with 4 receivers, obtaining an outdoor precision of around 100 meters [79]. Their work is improved by Podevijn et al.: they evaluate various trajectories (walking, cycling, and driving) in a larger network, reaching a median accuracy of 75 meters when enriching TDoA with the underlying road map [172]. Although such figures are impressive, they may not be sufficient when tracking devices in dense environments where multiple devices are deployed within meters of each other. Instead, alternative solutions based on the protocol itself can be explored.

2.3.4 Protocol-based tracking

Protocol-based tracking is enabled by any piece of information that is both stable across packets and unique to the device. The most obvious field is the device identifier (e.g. MAC address), but other adjacent data can also uniquely identify the device.

2.3.4.1 Stable identifiers

Wireless link-layer network protocols are generally organized in two stages. First, a device goes through a setup phase, where it discovers and connects to access points, through *probe request* frames in Wi-Fi [213] or *advertising packets* in BLE [111]. Second, once the connection is initiated, the device communicates using a variety of data packets. During both steps, devices include identifiers in their messages, allowing them to be easily tracked across various locations and timeframes [89]. Real-world deployments of such solutions include the one described by Bonne et al., which tracks visitors at large social events including at concerts and in city campuses [47]. Such a straightforward privacy vulnerability is generally countered by address randomization and pseudonyms, presented in Section 2.4.5. However, this issue is still relevant in constrained protocols utilizing stable identifiers such as LoRaWAN.

2.3.4.2 Other fields and metadata

The address itself is not the only piece of data able to identify a device. Vanhoef et al. highlight that certain protocol components may still disclose identifying information. For instance, some devices include their Universally Unique Identifier (UUID) in probe requests, circumventing MAC address randomization [213]. Additionally, companies leverage BLE features, including custom fields in advertising packets, to develop their own advertising protocols. Becker et al. illustrate that certain implementations enable user tracking even when randomized addresses are employed. For instance, the payload remains unchanged when a pseudonym is updated [44]. Similarly, Martin et al. show that BLE-based protocols employed in Apple products may facilitate tracking through various fields [143]. For example, a sequence number is incremented only when specific actions are performed, allowing eavesdroppers to track the device via the value itself. This also defeats MAC address randomization as the value is carried over to the new pseudonym. Their work is completed by Celosia and Cunche,

highlighting that interlayer synchronization, i.e. changing *all* identifiers and counters at once, is required to correctly protect from protocol-based tracking [56].

2.3.5 Device fingerprinting

A device's identity is closely tied with multiple information, including the identity of its user, the type of device, and its related activities. For instance, knowledge of the device's identity may reveal vulnerabilities and enable targeted attacks [206]. Likewise, knowing which device is currently in use may be highly relevant for activity recognition [15, 131]. Contrary to unique device identifiers presented in Section 2.3.4, we consider fingerprints as snapshots of the identity of the device, possibly encompassing multiple elements, from the radio signal and/or protocol headers.

Similar concepts apply to device fingerprinting. Fundamentally, it is a tool gathering a set of distinguishing features describing an object and organizing them in an intelligible manner. First, raw features are extracted from network traces as a vector $v = (x_1, x_2, ..., x_n)$. Second, a fingerprinting function $f(.)$ produces a fingerprint $F = f(v)$, corresponding to a specific device. Finally, a distance function $d(.)$ is computed over two input fingerprints as $D = d(F_A, F_B)$. The distance value D should reflect the (dis)similarity of provided fingerprints and hint at their origin. In this section, we explore both aspects: data representations after the fingerprint function, and distances between fingerprints.

Table 2.1: Summary of fingerprint representations and associated distances.

Data representation	Distance
Vector of values	Euclidean [17], Jaccard index [162]
Histograms	Euclidean [78, 174], Cosine similarity [156], Kullback-Leibler divergence [217], ML-based [35, 208], Unknown [19]
Descriptive statistics	ML-based [117, 160]
Markov chain	Maximum likelihood Criterion [121, 189], Custom solution [161]

Although most related works ask "*do these two fingerprints come from the same device?*", the others choose a different path. Instead of computing a distance between two fingerprints, they let a machine learning model determine to which device (i.e. class) the fingerprint corresponds. This advantageously avoids computing a distance, but is less robust: adding a new device to the network requires to re-train the machine learning model. Such an approach is abusively denoted as "ML-based" distance in Table 2.1 for completeness.

2.3.5.1 Data representations

Distinguishing between a fingerprint's *representation* and its actual *content* is important. The term "content" encompasses features extracted from network traces, irrespective of their origin, while "representation" refers to the format used to refine, organize, and utilize these features as a fingerprint. For instance, when looking at the payload lengths of captured frames, the content corresponds to actual length values, while a possible representation is the mean of those values. In this section, we explore different fingerprint *representations* employed in prior research on network fingerprinting.

Vectors of values : Multiple works rely on vectors of raw values, without any modification of information extracted from network traces [17, 162], such as $F = v$. For instance, Aernouts et al. fingerprint the location of LoRaWAN devices based on their signal power, with values directly associated with GPS positions [17]. When dealing with more than one feature, the fingerprint is a concatenation of multiple vectors: Pang et al. identify devices via various features extracted from their wireless traces, including a vector of packet lengths *and* a vector of network identifiers (SSID) [162]. For n features, the fingerprint thus corresponds to: $\forall v, F = f(v_1, v_2, ..., v_n) = (v_1, v_2, ..., v_n)$.

Histograms : Binning, also known as discretization, is a data preprocessing technique used in statistics and machine learning to reduce the number of distinct values in a continuous variable by grouping them into a smaller number of bins or categories. The process involves dividing the range of a continuous variable into intervals or bins, and then assigning each data point to the appropriate bin based on its value.

Binning is often employed to simplify complex datasets, reduce noise, and improve the efficiency of algorithms by making them less sensitive to small variations in the data [118]. Additionally, using a vector of values directly may be impractical because of the high number of measurements, or asymmetries between devices. For example, if one device sent 100 messages and another one only 50, how should their values be paired during comparison?

After binning raw data, a fingerprint is a vector of n_b elements representing the number of values in each bin (i.e. heights of a discrete histogram H_{n_b}). This representation provides a compact yet interpretable format, and is one of the most prevalent in the literature. It has been used with a variety of network features, including timing [19, 35, 208] and packet length [156]. Additionally, it is often chosen for radio-based features, such as received signal strength [174, 78], or lower level physical characteristics (e.g. carrier frequency offset) [217].

Fingerprints, and hence bin sizes, should reflect reality while providing exploitable information. For instance, IAT of network packets is often measured in milliseconds, but such a precision is unnecessary when identifying low-throughput devices (e.g. a message every hour). Worse, it can be detrimental if slight variations put a value in the wrong bin, reducing the accuracy. Multiple formulas (or "*rules*") exist to determine the optimal width and number of bins (or "*class intervals*"), depending on data distribution and acceptable number of bins [185]. Weirdly, all works cited previously use 300 bins to classify their time values [19, 35, 208]. Uluagac et al. "*empirically determined*" this number [208], but to the best of our knowledge, other histogram-based fingerprinting research simply does not mention the way bins are created.

Descriptive statistics : Alternatively to histograms, some works leverage descriptive statistics as a way to fingerprint devices, e.g. $F = (\overline{v}, \sigma)$. This is particularly useful when the raw data is too large or too complex to analyze at once [117]. Additionally, it does not require optimizing for the right number of bins. For instance, Overdorf et al. re-identifies visited Tor websites based on mean and mode of various features [160]. Additionally, more complex statistics such as variance, skewness, and kurtosis are leveraged by Klein et al. to fingerprint wireless devices based on radio signals [117].

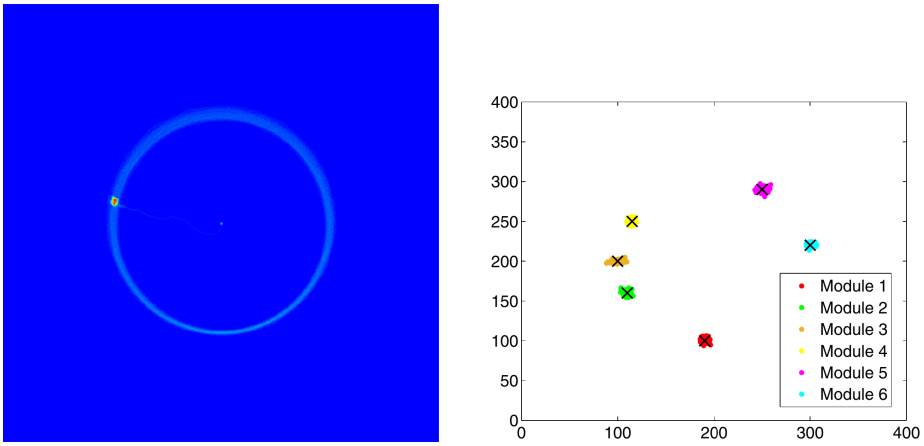
Markov chains : IoT devices have distinct communication patterns (cf. Section 2.3.2.1). For example, a sensor can be programmed to send a report every hour, or every day at midnight, with occasional alerts in between if a threshold is exceeded. Previous fingerprint representation do not fully exploit such a stateful behavior.

Although this aspect has seldom been investigated in wireless networks, a few works consider Web browsing as a stochastic process and model it as Markov chains [121, 161, 189]. They all model TLS communication as a series of states, constructing homogeneous[3] Markov chains from browsing traces, with the fingerprint corresponding to a stochastic matrix $P = \{p_{i,j}\}$, where $p_{i,j}$ is the probability of transition from state i to state j. These chains, built using known websites, are then compared to unknown visits to infer users' browsing habits despite encryption.

More precisely, initial work by Korczynski and Duda considers a single state based on the TLS message type (e.g. Application Data, Client Hello, etc.), with an *enter* state, various *transition* states, and an *exit* state [121]. Following research by Shen et al. and Pan et al. improves the solution in two ways: 1) a state is now a combination of TLS message type and message length, and 2) their Markov chains are not first order, but second order, meaning that the next state is not only influenced by the current state, but also the previous one [161, 189]. While more accurate, this approach generates complex chains with many states. Increasing the order and number of states in Markov chains results in high computational complexity due to the underlying representation as transition matrices.

[3]The transition probabilities do not vary across time.

Image-based : A subset of radio fingerprinting works focus on the physical layer using techniques previously developed for image recognition. This approach is based on the environment differences between devices, as well as the unique hardware generating identifying variations in the radio signal [197]. It involves converting radio frequency signals into visual representations, including spectrograms illustrating signal frequency over time [113, 188]. These representations are then analyzed using image recognition techniques, such as convolutional neural networks (CNNs). A highly visual example is provided by Jiang et al. [113], reproduced in Figure 2.6. By drawing the differential constellation trace figure[4] of various LoRa devices (Figure 2.6a), they are able to uniquely identify them (Figure 2.6b).



(a) Example of differential constellation trace figure.

(b) Clustering for 6 LoRa devices based on differential constellation traces.

Figure 2.6: Image-based radio fingerprinting, figures by Jiang et al. [113].

These works are currently limited by the medium quality, with their accuracy depending on good Signal-to-Noise Ratio (SNR) [188] while some physical layer protocols, such as Long Range (LoRa), are designed to operate effectively with low SNR. Although our work primarily focuses on higher-level network protocols and fundamental radio information such as signal power, it is essential to acknowledge physical layer fingerprinting as a potential threat to privacy.

2.3.5.2 Fingerprint distances

Fingerprinting is always deployed along some technical way to compare two fingerprints to determine their origin. Such a distance is highly depended on the fingerprint representation. Sadly, not all literature works cited in Section 2.3.5.1 explicitly select a distance function, let alone justify their choice. In this section, we present those who do provide information on their distance computations.

The simplest approach for vectors of raw values [17] or histograms [174, 78] is to compute the Euclidean distance. The Jaccard similarity index has also been used for vectors of raw values [70, 162]. Defined as the size of the intersection of the sets divided by the size of the union of the sets, it does not take into account the rarity of elements in each set. To address this issue, some weight can be added based on known data [162].

Histograms can be compared through the cosine similarity [156] or Kullback-Leibler (KL) divergence [217]. For two vectors A and B of length n corresponding to discrete probability distributions, they are computed as:

$$\text{cosine similarity} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$$

$$D_{\mathrm{KL}}(A \parallel B) = \sum_{i=1}^{n} A_i \log\left(\frac{A_i}{B_i}\right)$$

[4] A constellation trace is a graphical representation of the signal based on its amplitude and phase shift. The differential version studies their variations across time.

Works leveraging Markov chains depend on the Maximum Likelihood Criterion [121, 189]. It seeks to find the parameter values that make the observed data the most probable under the assumed statistical model. In the Web browsing fingerprinting scenario, it helps identify the website that is most likely to be the source of the captured encrypted traffic.

In text classification and document clustering, the KL divergence outperforms the cosine similarity [103], which in turn outclasses the Jaccard index [132] and Euclidean distance [103]. However, results highly depend on the studied dataset [103, 132]. To the best of our knowledge, no comparison of these various distance have been published in the context of network traffic fingerprinting.

2.4 Countermeasures against IoT privacy attacks

Although finding privacy vulnerabilities in IoT systems is a mentally stimulating first step, it is also crucial to provide mitigations protecting end-users. In this section, we highlight some privacy metrics used to measure the impact of countermeasures. Then, we present various approaches to reduce the efficiency of privacy attacks, such as dummy traffic, adding noise, modifying the radio medium, and pseudonym rotation strategies. A summary of countermeasures relative to the attack(s) they thwart is available in Figure 2.7.

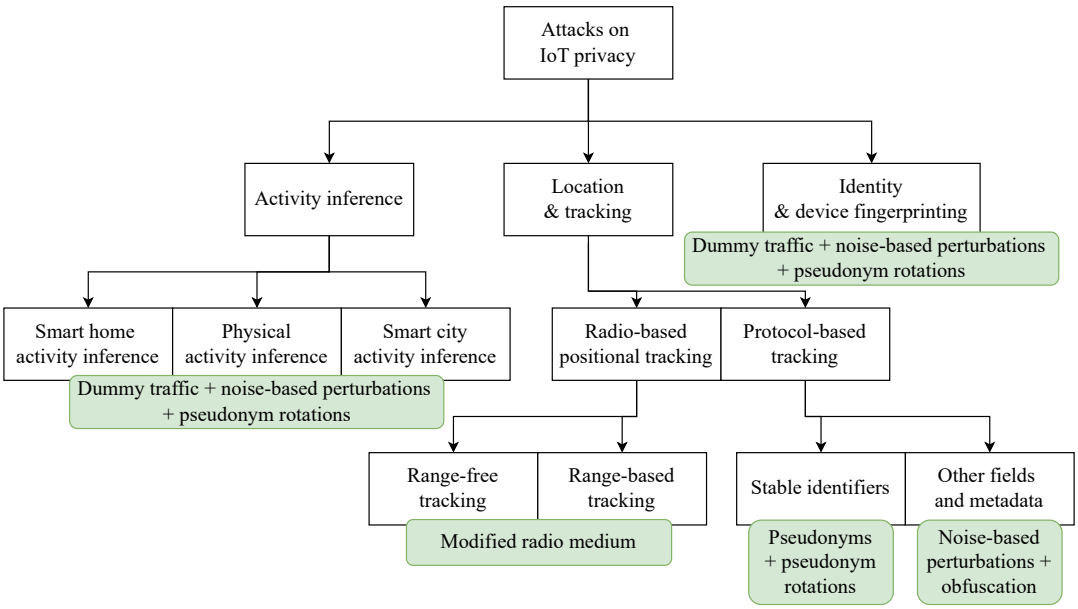


Figure 2.7: Countermeasures associated to each privacy attack.

Please note that each mitigation is presented in isolation, but combining multiple countermeasures is often required for higher impact. For example, padding alone (included in "noise-based perturbations" in Figure 2.7) may not be sufficient to counter DNS fingerprinting [53]. Furthermore, practicing data minimization whenever possible is mandatory: an attacker cannot exploit information they do not have [27]. Finally, for attacks based on cleartext fields leaking information (see Section 2.3.4.2), hiding their content through encryption remains the best countermeasure.

2.4.1 Privacy metrics

Privacy metrics are quantitative measures used to evaluate the level of privacy protection provided by a system, algorithm, or protocol. They play a crucial role in designing and evaluating privacy-preserving systems and ensuring compliance with privacy regulations and standards. They enable researchers, developers, and policymakers to assess the effectiveness of privacy-enhancing techniques and make informed decisions regarding data handling and protection. Wagner and Eckhoff provide a complete overview of such metrics and highlight that they are context-dependent [219]. First, they vary based on the specific type of privacy

being targeted. For instance, different metrics are employed for location privacy compared to protecting against activity inference. Second, privacy metrics are influenced by the threat model, which encompasses the goals and capabilities of potential attackers [219].

From the eighty privacy metrics Wagner and Eckhoff cite, authors usually pick a few depending on their use-case. For instance, Becker et al. work on pseudonyms in BLE and define an *anonymity set* (i.e. the set of users that an attacker cannot distinguish from the targeted user) and the *maximum tracking time* (ie. the cumulative time during which an user is uniquely identified) [44]. More generally, machine learning-based privacy research often focuses on the change of accuracy of models trained with or without countermeasures [22, 53].

2.4.2 Dummy traffic

To counter both activity inference and device fingerprinting, one can break existing patterns by inserting artificial or irrelevant data. Doing so obfuscates legitimate information, thus protecting user privacy, and can be done directly by the device [131, 153], or by an external anonymity provider [27]. For instance, some works have focused on protecting the location of nodes or sink in wireless sensor networks to conceal the actual origin of legitimate nodes message [112, 229]. Likewise, dummy traffic has been used in the context of anonymous Web browsing: Yu et al. leverage server-side predictions of adjacent pages to hide the actual visit [230].

More generally, constant rate transmission has been proposed to mitigate activity inference based on event detection. However, selecting the interval is a complex problem: too short, and it depletes device batteries, too long, and near real-time capabilities are lost [139]. Instead, Yang et al. design a system leveraging an exponential rate of transmission: if an event occurs, sensor nodes wait for the next interval following the existing distribution. Such an approach greatly reduces the event transmission delays compared to a constant rate scheme [228].

More closely related to our field of research, Acar et al. artificially obfuscate smart home traffic and find that: "*injecting false data equivalent to 10% of packets exchanged during the observation time resulted in a decrease by 13%*" of accuracy [15]. Leu et al. actually implement countermeasures and explore the difficulties of generating dummy packets for Low Power Wide Area Networks (LPWANs) in a statistically convincing way (i.e. taking into account generic traffic *and* rare anomalies) [131]. They find that not knowing the actual communication pattern greatly increases the risk of information leakage and that generating only a few fake anomalies has a limited impact, contrary to saturating communications with dummy traffic.

Instead, Möllers propose *naive exponential dummies*, which generate traffic only if no message is sent during a specified time window, effectively reducing spent energy. The author notes that this approach does not provide protection guarantees against actions generating different number of messages. For example, if switching on a light produces 3 frames, and opening a door leads to 4 messages, it is possible the system still outputs 3 frames, letting an attacker infer the performed action [153]. Apthorpe et al. follow a similar path by developing *stochastic traffic padding*, which hides existing burst patterns by randomly adding data during or in between user activity via a middlebox [27].

In any case, generating dummy traffic comes at a cost, while offering imperfect protections. Similarly to previous works [112, 229], Leu et al. outline that additional traffic generates significant energy consumption [131]. Following research by Möllers confirms that constant-rate dummy traffic is not suited for low throughput networks (e.g. home automation) [153].

2.4.3 Noise-based perturbations

Machine learning-based attacks against IoT privacy require large amount of data to train models. Based on a variety of features (e.g. length of payloads, timing), they are fundamentally statistical representations and can be countered by adding noise to the data. In this section, we present perturbation-based mitigations whose goal is to reduce identifiability of network traces, such as padding and delay.

2.4.3.1 Padding data

While most IoT network protocols are encrypted, few conceal the length of the underlying data. Hence, the encryption process typically results in an output length depending on the input. As seen in Sections 2.3.2.1 and 2.3.5.1, such an information can be exploited by attackers. *Padding* is a straightforward solution adding data at the end of the payload to hide its actual length once it is encrypted.

Dyer et al. define multiple padding strategies for encrypted HTTP traffic [76], such as:

1. *Linear padding*: Packets are padded to the nearest multiple of 128 (or to the Maximum Transmission Unit (MTU), usually 1500 bytes).
2. *Packet Random MTU padding*: Packets are randomly padded, up to the MTU.
3. *Pad to MTU*: Packets are padded to the MTU.
4. *Exponential padding*: Packets are padded to the nearest power of 2 (or to the MTU).
5. *Mice-Elephants padding*: Packets are padded to 128 if smaller, or to the MTU if bigger. This is relevant when packets are generally small, with a few large outliers.

The RFC 8467 [145] on DNS padding cites the first 3 strategy and advocates only for the first (named *Block-Length Padding*), while proposing *Random-Block-Length Padding*, which pads to a randomly selected multiple of 128.

At a high level, two main strategies emerge from the remaining literature: padding to the maximum length possible or randomly padding within a specified interval. These involve trade-offs between privacy and performance: padding to the MTU provides perfect protection, but requires significant resources. Pinheiro et al. observe that real-world packet lengths are mostly under 100 bytes long, suggesting that uniform padding up to the MTU is inefficient [171]. They propose a hybrid approach: padding packets to the nearest hundred up to 300 and randomly beyond 301 to the MTU. This method reduces resource consumption while maintaining privacy. Additionally, they explore adaptive padding based on network activity, padding more during idle periods to obscure information and less during active network usage. Engelberg and Wool evaluate the approach and find that while it effectively hides the specific device being used in the network, it does not prevent detecting the number of active devices [77].

Most of the literature focuses on high-throughput network protocols (e.g. HTTP [76] or DNS [145]) allowing to pad 128-bytes long blocks of data. However, Shafqat et al. outline that this approach may not be feasible in constrained protocols, such as ZigBee for which they propose a random padding of 0 to 3 bytes [186] (also see Section 3.1.7).

Finally, Alshehri et al. explore the choice of distribution for random padding selection, aiming to achieve Differential Privacy [22]. While Laplace noise seemed promising, its implementation requires reducing half of the packet lengths. Instead, they demonstrate that uniformly random noise satisfies (ϵ, δ) Approximate Differential Privacy, a more lenient form [75]. Parameters $\epsilon = 0$ and $\delta = \frac{\Delta s}{n}$ are selected, with n the maximum number of bytes, ensuring optimal privacy protection (minimal ϵ). Here, Δs denotes the maximum difference in packet size: higher potential padding correlates with stronger privacy guarantees.

2.4.3.2 Delaying messages

Intentionally adding delay to packet transmissions mitigates the risk of privacy breaches by making it more difficult for adversaries to infer sensitive information from network activities (see Section 2.3.2.1) [139]. Even if delay and dummy traffic are often used together, we purposefully separate them as only delay may be conceivable in constrained networks.

The main difference of various literature works is based on the protocol type and amount of delay. For instance, Zhang and Zang focus on mesh sensor networks, where each node on path adds a slight delay [231]. In smart home settings, Srinivasan et al. advocate for 15 to 20 minutes of random delay, demonstrating that rare events require more time to be hidden from eavesdroppers [197]. Ashur et al. go even further when designing a BLE & LoRa privacy-preserving device tracking system: messages are randomly sent in a one-hour interval [33]. Additionally, Prates et al. discover that the time between request and answer can also be

used to identify devices [173]. They address this issue by introducing a random 10ms delay before responding to a request. However, their analysis combines this countermeasure with dummy traffic, making it challenging to determine which measure has the most significant impact.

Unlike padding, delaying packets can significantly disrupt the application flow, potentially causing timeouts or incorrect arrival order if the delay is too high [139]. Srinivasan et al. emphasize the importance of adjusting delay values for each application, as some can handle extended delays of several minutes, while others cannot tolerate any delay at all (e.g. emergency applications) [197].

2.4.4 Modified radio medium

Radio-based countermeasures are in theory extremely powerful in mitigating privacy attacks within wireless networks at their source. Physical layer security techniques focus on reducing the eavesdropper ability to access the medium while others mitigate the identifiability of radio signals. For instance, selective jamming [23], signal reflection [198], and frequency hopping [20] have been developed to secure wireless communication channels and prevent eavesdropping.

Anajemba et al. utilize multiple antennas on both the transceiver and receiver sides, alongside a jamming station, to minimize the number of messages accessible to attackers while preserving link quality for legitimate receivers [23]. Likewise, Staat et al. develop a prototype of intelligent reflecting surface targeting adversarial motion detection. By introducing random variations in the wireless propagation, the surface perturbs the signal enough to counter sensing [198]. Finally, Albazrqaoe et al. propose software-based solutions to better obfuscate the channel hopping mechanism of Bluetooth, by flipping the status of randomly selected subchannels [20].

To counter hardware-based identification, Srinivasan et al. cite more effective yet complex and/or costly solutions, such as using a single radio transceiver for multiple sensors, or using potentiometers instead of resistors and varying these between communications [197]. As we highlight in Chapter 6, additional software-based countermeasures can be deployed to reduce signal power relevance during fingerprinting.

However, contrary to more generic approaches such as padding (see Sections 2.4.3.1), modifying the medium itself requires significant resource investments (e.g. adding new antennas, deploying reflective surfaces). Likewise, jamming techniques are generally used in space-limited environments. For instance, Anajemba et al. deploy their system with 3 meters between nodes, which is far from the multiple kilometers separating LPWAN devices (see Section 3.1) [20].

2.4.5 Pseudonyms

Instead of relying on stable identifiers, modern devices often opt for random and ephemeral link-layer addresses to enhance privacy, often named *pseudonyms*. Pseudonyms are intimately linked with the devices capabilities they are deployed on and the protocol requirements for address space (i.e. the number of bits available). As seen in Section 2.2.3, this is still an open question for the 802.11 family, and work is required to bridge the gap between resource-intensive pseudonym solutions and constrained IoT devices. We propose a tailored solution to this problem in Chapter 7.

In this section, we present various pseudonym schemes for 802.11/Wi-Fi [30, 90], sensor networks [152, 231], BLE [92], and vehicular networks [138, 169]. The solutions discussed here only support mutual authentication, where both the device and central authority can authenticate the message's source. Hence, we consider device anonymity without two-way verification as out of scope. For instance, we exclude randomized MAC addresses in Android Wi-Fi module [88] or group signatures in vehicular networks [169], for which the receiver can not authenticate the sender.

2.4.5.1 Encrypted link-layer pseudonyms

A first way to hide the value of the link-layer identifier field is to completely encrypt it. Armknecht et al. apply this context in a new link-layer protocol and discuss its adaptation to 802.11 [30]. An encryption key is pre-shared between each device and the Access Point

(AP) they are connected to. It is used to encrypt MAC addresses along a sequence number s_i by both the devices and AP. Such encrypted number s_i is also appended at the end of the frame. Then, the receiving device or AP matches the encrypted pseudonym with its own pre-generated values.[5]

If a frame is lost, the sender still updates s_i, meaning the device and AP are now desynchronized. Upon receiving a frame containing a pseudonym not matching any pre-generated value, the AP tries to decrypt s_i with each pre-shared key. If s_i is found, the pseudonym can be decrypted and both parties are synchronized again.

This approach has been updated by Greenstein et al. [90]. Their solution, named *Shroud*, is different in two aspects. First, they encrypt the sequence number via AES-128, selecting the whole block as identifier, instead of appending a sequence value at the end of the frame. Second, they generate multiple encrypted pseudonyms at once, contrary to one at a time. Thus, the receiver can look up a larger set of pre-generated pseudonyms when packet loss occurs. Although this approach requires slightly more initial computation power and memory storage, Greenstein et al. leverage hash tables instead of lists to reduce complexity and do not require a costly process to re-synchronize, as long as enough pseudonyms are pre-generated.

2.4.5.2 HASHA

Zhang and Zhang propose an alternative design to protect device location in sensor networks [231]. First, devices transmit their original address (i.e., MAC) through beacon frames, resulting in each device maintaining a list of all nearby identifiers. Second, each pseudonym is generated using a rolling key, starting with an arbitrary fixed value. The temporary pseudonym TP of the first data message is generated by both devices using a fixed-value key k and the original address OA of both source S and destination D as: $TP_s = \text{HMAC}(k, OA_s)$. Finally, devices update the key by XORing it with the payload of the previous message. All messages are acknowledged to make sure the keys are correctly synchronized between two devices.

2.4.5.3 Resolvable random private address

So far, all solutions require some sort of synchronization between sender and receiver, which is not the case for Resolvable random Private Addresses (RPAs), the alternative solution deployed in BLE. RPAs are created by generating a 24-bit hash h using a hash function H, an Identity Resolution Key (IRK) I_k, shared between the sender and receiver, and a 22-bit random value r: $h = H(I_k, r)$ [92, sec. 1.3.2.2]. Upon receiving a pseudonym, it can be resolved by comparing the transmitted hash value with the one computed locally using the corresponding IRK.

2.4.5.4 Pooling pseudonyms

Although they operate as mesh networks instead of star topologies, Vehicular Ad hoc NETworks (VANETs) are notorious pseudonym users [169]. Each vehicle can request a trusted third-party for *multiple* pseudonyms (i.e. certificates based on asymmetric cryptography) and sign their messages with private keys to prove their origin upon reception. VANET pseudonyms are delivered in bulk and valid for a specific period, meaning a vehicle randomly selects one of its certificate for authentication [138]. As connectivity can not be guaranteed, receivers must be able to verify the validity of incoming pseudonyms locally. Thus, the pseudonym (i.e. complete certificate) is appended to each message [169].

A similar approach is described by Misra and Xue for wireless sensor network. The base station allocates random subranges from a network-wide pool of pseudonyms to the nodes. Nodes subsequently select a pseudonym value randomly from their allocated subrange. Both the base station and nodes maintain tables correlating subranges with devices, facilitating encryption and path forwarding capabilities among neighboring nodes [152].

[5] We note a similar method leveraging an encrypted counter is used in *rolling codes*, deployed in Remote Keyless Entry systems [86]. In their case, the mechanism is used to prevent replays.

2.4.6 Pseudonym rotations

As discussed in Section 2.4.5, numerous pseudonym schemes are available. However, relying on a single pseudonym for an extended period of time is insufficient to effectively thwart tracking. Therefore, regular updates to pseudonyms are necessary to enhance privacy protection, either unsynchronized or with devices working together to protect their privacy. We also note that rotating pseudonyms *can* thwart any attack based on collecting large amounts of data corresponding to a device's identity. For instance, activity inference and fingerprinting are less effective if the information available for a single identifier is reduced by frequent pseudonym rotation.

Most rotation strategies are based on uncoordinated timings. For example, BLE RPAs are usually rotated every 15 minutes without external synchronization [92, 143]. Martin et al. note that such a naive approach offers limited privacy protection as 1) infrequent pseudonym rotation increases the risk of linking two pseudonyms when one disappears and a new one appears, and 2) updates every a precise number of minutes make it easier to monitor user activities. Instead, they advocate drawing a random time after which the pseudonym should be updated [143]. Such a concept is applied by Liu et al. to VANETs [138]. Vehicles randomly pick a new pseudonym from a pool (see Section 2.4.5.4), use it until its expiration date, and then wait for a random delay before communicating again. This approach requires a buffering period where the device does not generate any traffic [138].

Similar synchronized solutions were initially developed to protect location privacy. They rely on *mix zones*, where multiple devices operate together to *mix* their traces and mislead an attacker [84]. Huang et al. introduce a silent mix zone: pseudonyms are updated only after a random delay during which devices do not communicate [104]. However, situations with limited number of devices nearby are frequent. Vaas et al. address this issue in VANETs. By broadcasting dummy traffic of *chaff* vehicles via roadside units, they create safe zones where a vehicle can update its pseudonym [211].

Again, authors highlight that, no matter the complexity and intricacies of pseudonym rotation, it is mandatory to correctly take into account other metadata to avoid address carry-over [44, 143] (see Section 2.3.4.2).

2.5 Conclusion

Privacy, influenced by cultural, sociological, legal and technological factors, remains a complex concept, even within the reduced scope of IoT devices. Our work aligns with evolving regulations that emphasize data protection, security, and transparency to uphold privacy in the IoT ecosystem. As guidelines and standards continue to evolve alongside legal and technological advancements, our research hopes to inspire privacy-preserving solutions and contribute to more precise guidelines.

While this section sheds light on various eavesdropping-based attacks in the IoT landscape, there remains ample opportunity for further exploration, particularly in resource-constrained protocols like LPWANs. We also outline that existing tools such as fingerprinting are battle-tested but rarely systematically compared.

Countermeasures against IoT privacy attacks span a diverse array of strategies, reflecting the multifaceted nature of the threats they mitigate. However, implementing these countermeasures often entails navigating a delicate balance between enhancing privacy and minimizing the impact on network performance, ranging from increased bandwidth usage to potential message loss. Such trade-offs are even more apparent in highly constrained network protocols such as LoRaWAN.

Chapter 3

Background

Contents

This chapter provides additional elements required for the rest of the thesis. It details applications, architecture and various technical features of the LoRaWAN protocol, followed by an introduction to the fundamentals of machine learning and its pitfalls.

3.1 LoRaWAN

In this section, we introduce Long Range Wide Area Network (LoRaWAN), a Low Power Wide Area Network (LPWAN) protocol. Standardized for the first time in 2015 by the LoRa Alliance, its adoption grew rapidly and LoRaWAN is now deployed by 181 operators worldwide [67]. Although multiple versions evolve in parallel, we focus on LoRaWAN v1.1 as it provides the most mature solution [65, 140]. Likewise, the protocol is available in various frequency bands, but we are interested in the ones deployed in Europe: EU863-870MHz and EU433MHz [64].

Although we refrain from rewriting the entire LoRaWAN specification [65], we provide an overview of relevant concepts related to networks and privacy. Thus, we first present existing LoRaWAN application, before delving into its network architecture. We introduce the frame structure and relevant fields for our thesis, including identifiers. We then detail the join process along with keys generation and message types. Finally, we present the multiple constraints under which the protocol operates.

3.1.1 Applications

Anything connected to a low data rate sensor, sending at most a few messages per hour, can leverage LoRaWAN for data extraction or occasional notifications. Following in the steps

of large-scale IoT networks, LoRaWAN now connects 300 million devices to a wide range of applications [67].

From urban environments to rural landscapes, LoRaWAN finds utility in various sectors such as smart cities, agriculture, environmental monitoring, logistics, healthcare, and industrial automation [96, 199]. In smart cities, LoRaWAN facilitates intelligent infrastructure management, including waste management [175], street lighting, and parking [74]. In agriculture, it aids in precision farming [192], crop monitoring, and livestock tracking [114]. Environmental monitoring applications leverage LoRaWAN for air and water quality monitoring [137, 204], weather stations, and wildlife conservation [34, 38]. Furthermore, LoRaWAN plays a pivotal role in asset tracking, supply chain management, and fleet monitoring in logistics [187]. In healthcare, it enables remote patient monitoring, medication adherence tracking, and telemedicine initiatives [147].

3.1.2 Architecture

As seen in Figure 3.1, a classic LoRaWAN architecture includes various components [65].

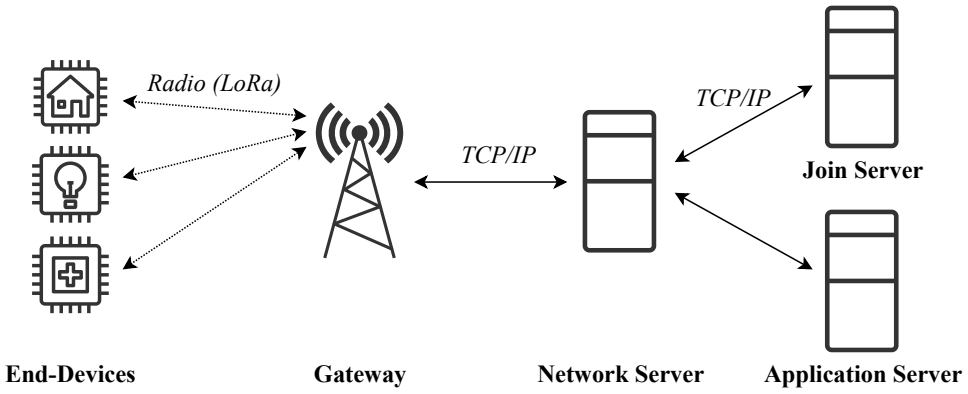


Figure 3.1: LoRaWAN architecture overview.

At the core are **End-Devices**, which include sensors or actuators deployed in the field to collect or act on data. These End-Devices transmit information to **gateways** acting as intermediaries between the End-Device and the network infrastructure. Gateways thus receive data from End-Devices via the LoRa radio modulation and forward it through a classic TCP/IP link. We note that *LoRa* is different from *LoRaWAN*: the radio modulation is only one of the physical layers hosting the MAC layer protocol. In our work, we focus on LoRaWAN.

Following the gateways, a set of **servers** manages the incoming messages (`uplink`) and, if necessary, responds to End-Devices (`downlink`). Many LoRaWAN implementations use a single node to host the servers for deployment simplicity. In any case, they can be functionally separated as three entities:

- The **Network Server** is responsible for managing the communication between End-Devices and gateways. It handles network-related tasks, including message deduplication, radio parameter optimization, and MAC layer commands. The NS also enforces security mechanisms, such as authentication and encryption to protect the integrity and confidentiality of the data transmitted over the network. Finally, it acts as a routing interface between End-Devices and the other servers. For instance, it routes uplink application payloads to the designated Application Server (AS) and facilitates the join process with the Join Server (JS) (see Section 3.1.6).
- The **Join Server** handles the onboarding process for new End-Devices joining the network. It manages the authentication and key generation process between End-Devices and the NS (see Section 3.1.6).
- The **Application Server** receives and processes data from End-Devices forwarded by the NS. It interfaces with external systems or applications, integrating LoRaWAN data

and triggering actions or alerts for real-time decision-making and automation.

3.1.3 Frame structure

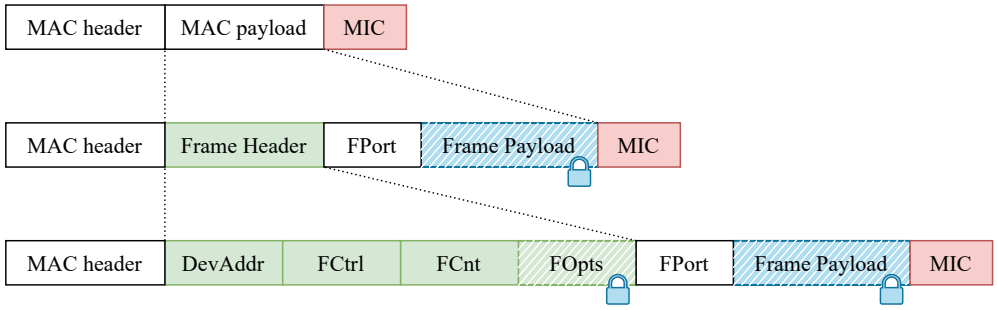


Figure 3.2: Frame format of `uplink` and `downlink` LoRaWAN messages.

Both `uplink` (End-Device to server) and `downlink` (server to End-Device) messages follow the same structure [65, Sec. 4]. They are encapsulated in a MAC payload, along with a MAC header, which includes the Message Type (`uplink`, `downlink`, etc.) and the protocol Major Version (v1.0, v1.1). The following fields are relevant to our work (from left to right in Figure 3.2):

- **DevAddr:** The *Device Address* is used to identify End-Devices in the network (see Section 3.1.4.2).
- **FCnt:** Each `uplink` and `downlink` message is ordered via a separate *Frame Counter*, respectively `FCntUp` and `FCntDown`. Starting at 0 after the join process (see Section 3.1.6) and sent unencrypted, the values are initially computed on 32 bits but only the 16 least-significant bits are actually transmitted.

  Although the `FCntUp` is a unique value, the `FCntDown` corresponds to two counters incremented independently. The `NFCntDown` is increased for each `downlink` messages on port 0 (or when the field is missing), while `AFCntDown` is incremented for each `downlink` messages with a port different from 0. Both End-Devices and NS track `FCnt` values to re-order messages or detect duplicate messages.
- **FOPts:** *Frame options* are used to send *MAC commands* for network management and configuration tasks, including device status request, data rate adjustments, and security parameters management.
- **FPort:** Similarly to the TCP/IP model, LoRaWAN leverages a *Frame Port* to direct traffic to specific applications. For instance, one port may serve maintenance functions while another is selected to transmit sensor data.
- **Frame Payload:** Containing application data or MAC commands, the payload is encrypted using AES-128 CCM* (see Section 3.1.6). Its maximum length depends on a radio parameter controlling the number of bits sent per second, the DataRate (DR). Table 3.1 details possible length and DR values [64, Sec. 2.1.6].
- **MIC:** The *Message Integrity Code* enables the verification of the frame integrity by computing an AES-128 CMAC over various fields (see Section 3.1.6). It may also be used for resolution of identifier conflicts in some cases (see Section 3.1.4.2).

The payload and `FOpts` are encrypted (see Section 3.1.6) but all other fields are sent in clear-text and thus available to any eavesdropper.

3.1.4 LoRaWAN identifiers

In LoRaWAN, two link-layer identifiers with distinct roles are used: the `DevEUI`, and `DevAddr`. As seen in Chapter 4, both of them are highly relevant to privacy.

Table 3.1: Maximum payload length based on the DataRate (EU863-870Hz band).

DataRate value(s)	Maximum payload length
0..2	51
3	115
4..7	222

3.1.4.1 DevEUI

Each LoRaWAN device is identified by a globally unique EUI-64 identifier named `DevEUI`. Allocated by the manufacturer or owner, the `DevEUI` remains unchanged throughout the device lifespan. Similarly to a BLE or Wi-FI MAC addresses, the first 3 bytes corresponds to the Organizationally Unique Identifier (OUI) of the End-Device, identifying the device's vendor, manufacturer or organization [5], and registered to the Institute of Electrical and Electronics Engineers (IEEE) [108].

3.1.4.2 DevAddr

Every time an End-Device joins the network, the NS allocates a new 32-bit temporary identifier called `DevAddr`. It is then used for the duration of the session. Similarly to IP addresses, this identifier conveys both routing and identity information. More precisely the `DevAddr` is divided into two subfields:

- The `AddrPrefix` is fixed and identifies the network an End-Device belongs to.
- The `NwkAddr` is randomly generated by the NS and identifies the device within the network it belongs to (designated by the `AddrPrefix`).

Table 3.2: `DevAddr` format based on the network type.

Network type	`AddrPrefix` (bits)	`NwkAddr` (bits)
0	7	25
1	8	24
2	12	20
3	15	17
4	17	15
5	19	13
6	22	10
7	25	7

Due to the diverse requirements of network operators, LoRaWAN supports 8 network types (type 0 to type 7), each offering an increasing range of available addresses. As indicated in Table 3.2, End-Devices are addressed over 7 to 25 bits, corresponding to 2^7 to 2^{25} unique identifiers per network.

Multiple End-Devices might share a single `DevAddr`. This happens due to configuration errors or address allocation optimization [97]. In such instances, End-Devices are differentiated through the Message Integrity Code (MIC). The receiver computes the MIC using various potential keys until one of them produces a value equal to the one received, thereby singling-out the device.[1] In case of an End-Device receiving the message, it only has to compute the MIC with its own key.

3.1.5 Message types

LoRaWAN leverages multiple types of messages for various purposes within the protocol, such as device registration, data transmission, and acknowledgment. Each LoRaWAN messages has

[1] An example of such an implementation can be found in the *The Things Network* repository: https://github.com/TheThingsNetwork/lorawan-stack/blob/301af52a0e64d67f90576ededa59836bca19a03b/pkg/networkserver/grpc_gsns.go#L931

a type defined through the Message Type (MType) field in the MAC header and summarized below:

- **Join Request:** an End-Device asks to join the network.
- **Rejoin Request:** an End-Device asks to re-join the network, by changing a part or all of the session context (inducing a `DevAddr` update).
- **Join Accept:** the NS validates a connection to the network (answering both Join and Rejoin requests).
- **Data up:** `uplink` data frame, can require acknowledgment (Confirmed) or (Unconfirmed) from the NS.
- **Data down:** `downlink` data frame, can require acknowledgment (Confirmed) or (Unconfirmed) from the End-Device.
- **Proprietary:** available for non-standard communications.

3.1.6 Join process and keys generation

An End-Device has two ways of joining a LoRaWAN network: Activation By Personalization (ABP) or Over-The-Air-Activation (OTAA). In both cases, two keys are assumed shared between the End-Device and the infrastructure: the NwkKey and AppKey [65, fig. 49]. The OTAA mechanism is further analyzed from a privacy perspective in Chapter 4.

3.1.6.1 Activation By Personalization

In ABP, End-Devices are pre-provisioned with various information by the network administrator. This includes the `DevAddr`, and all hard-coded session keys.

ABP is simpler and quicker to implement than OTAA. However, it is not possible to update the cryptographic materials without physically accessing the End-Device. This a) poses a threat to data security if the keys are compromised and b) is cumbersome for network management as an End-Device cannot change of network provider without manual intervention.

3.1.6.2 Over-The-Air-Activation

In OTAA, End-Devices dynamically obtain the `DevAddr` and cryptographic materials during the join process. Figure 3.3 illustrates this sequence of events.

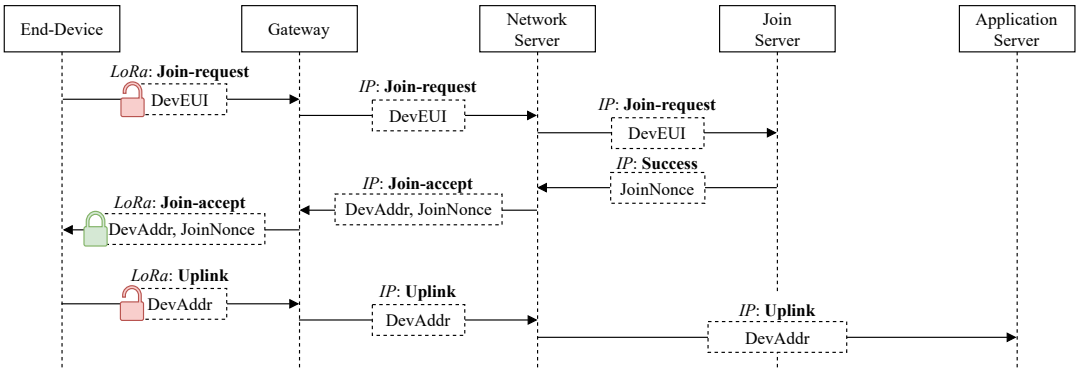


Figure 3.3: Join procedure using OTAA (red open lock: this specific field is unencrypted; green closed lock: these specific fields are encrypted).

First, an End-Device sends a `join-request` including its unique, clear-text `DevEUI`. The Network Server (NS) forwards this information to the Join Server (JS), which authenticates the device. Then, the JS generates the `JoinNonce` and sends it to the NS. The NS produces a new `DevAddr` and forwards it along with the nonce to the End-Device. At this point, both End-Device and JS possess the same `JoinNonce` value and can derive various session keys:

- The pre-shared `AppKey` derives the `AppSKey`, shared with the Application Server (AS), and responsible for payload confidentiality via AES-128 CCM* encryption [65, sec. 6.2.5].
- The pre-shared `NwkKey` derives the `NwkSEncKey`, `FNwkSIntKey`, and `SNwkSIntKey`, all shared with the NS.[2] Network-specific information such as MAC commands and `FOpts` are encrypted using the `NwkSEncKey` via AES-128 CCM*. `FNwkSIntKey` and `SNwkSIntKey` are leveraged to compute the MIC via AES-128 CMAC, guaranteeing integrity [65, sec. 6.2.5].

Once every key is derived, the End-Device can start the actual communication, by inserting its clear-text `DevAddr` in all following `uplink` messages. The same value is used by the NS in `downlink` messages.

3.1.7 Constraints

LoRaWAN operates under several constraints, making it suitable primarily for low-bandwidth applications, such as sensor networks, rather than high-throughput ones. As we propose a new design in Chapter 7, it is crucial to consider these constraints.

First, End-Devices usually operate on battery power during long period of time, spanning multiple years. Energy is a scarce resource and everything is made to optimize its usage. For instance, high transmission costs encourage developers to reduce the payload to a minimum [54].

Second, LoRaWAN devices transmit in unlicensed frequency bands, respecting the ETSI EN 300 220-2 standard for channel access restrictions [110]. Along with low data rates (250 to 11000 bit/s [64, Sec. 2.1.3]) and limited payload sizes inherent to LoRaWAN, the standard imposes strict duty cycles: an End-Device must not transmit more than 0.1% to 10% of the maximum time depending on the band [110, Tab. B.1].

Third, End-Devices are generally low-cost sensors with limited computational and memory capabilities, placing them between class 0 and class 1 of IETF classification of constrained devices [49]. They are often equipped with CPUs operating at a few dozen megahertz (contrary to the gigahertz of modern personal computers) [50]. Semtech, the leading manufacturer of LoRaWAN chipsets, states that RAM can be as low as 8 kB. [3]

Finally, contrary to low range wireless protocols such as Wi-Fi and BLE, LoRaWAN is expected to work in large and dense environments (e.g. cities). In this context, the Packet Loss Rate (PLR) is high; average values ranging from 1% to 40% have been reported in real-world deployments [128, 136, 170].

3.2 A machine learning primer

Machine learning is a common denominator across the contributions presented in this thesis. Rather than a fundamental research subject, it is used as a tool. In this section, we present the basics of the discipline while providing insights on the steps taken to avoid common pitfalls.

3.2.1 Step-by-step overview

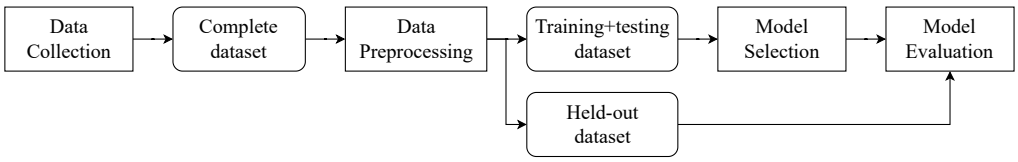


Figure 3.4: Typical research machine learning pipeline.

Machine learning is generally organized in sequential phases presented in Figure 3.4, including [36]:

[2] Here, F corresponds to "Forwarding" and S to "Serving", a terminology relative to roaming scenarios, which are outside the scope of this thesis.

[3] https://lora-developers.semtech.com/documentation/tech-papers-and-guides/mcu-memory-management/

1. *Data Collection*: Gathering relevant data that will be used to train and test the machine learning model. This data should include features (inputs) and corresponding labels (outputs).

2. *Data Preprocessing*: Cleaning the data to make it suitable for modeling, and preparing sub-datasets for training and testing.

3. *Model Selection*: Choosing the appropriate machine learning algorithm based on the nature of the problem, the type of data, and the desired outcomes.

4. *Model Evaluation*: Training the model and assessing its performance.

In real-world deployments, the initial steps involve selecting the most effective model to be used in production (e.g. make predictions on incoming data). However, our research goal is only to show that it is *possible* to build a robust model, not to actually exploit it in the wild. Thus, some steps are left out, including as *Hyperparameter Tuning* (when possible, see Section 8.6.3), *model deployment*, and continuously *monitoring and maintaining* the model.

3.2.2 Unsupervised vs supervised learning

In machine learning, two prominent approaches exist depending on the task and technical requirements: unsupervised and supervised learning [36].

Unsupervised learning involves training a model on unlabeled data, aiming to uncover hidden patterns or structures within the data without guidance from labeled examples. Unsupervised learning is valuable for exploratory data analysis and uncovering insights from large datasets where labeled data may be scarce or unavailable. Clustering and dimensionality reduction are common tasks in unsupervised learning [36]. For example, such techniques can be used to group devices exhibiting similar behavior together. However, the lack of *ground truth* makes it challenging to evaluate the performance of unsupervised learning models objectively. The interpretation of the discovered patterns or clusters may also vary depending on the context and domain knowledge [210].

Conversely, supervised learning involves training a model on labeled data, where the input data is paired with corresponding output labels [36]. The model learns the relationship between the input and output data, enabling it to make predictions or classifications when presented with new, unseen data. This approach is effective for tasks such as classification, regression, and prediction [36]. For instance, in a supervised learning scenario, a model can be trained to classify network traces and detect known attacks (e.g. DoS) [37]. In this thesis, we work with labeled data and therefore select supervised learning as the evaluation method.

3.2.3 Common pitfalls

Machine learning highly depends on the utilized dataset, which is why we make sure to conduct extensive, large-scale, and realistic studies involving a substantial number of devices (see Sections 4.3, 5.3, and 8.4). In this section, we cite some of its technical pitfalls [31] and how we avoid them.

3.2.3.1 Data snooping

Data snooping occurs when information from the *test* dataset, which would not be available to the model in reality, influences the model during *training*, leading to overly optimistic performance estimates. To prevent data snooping, it is mandatory to separate the dataset into distinct training and test datasets [31].

3.2.3.2 Training on imbalanced data

In many classification tasks, the dataset contains more samples corresponding to one class than the other, creating an *imbalance*. This can lead to biased models that favor the majority class, resulting in poor performance for detecting rare events [42]. For instance, this is particularly relevant in intrusion detection, where attacks constitute a small fraction of network traffic; a model that incorrectly labels intrusions as benign is ineffective [31].

To balance the training dataset, multiple techniques exist. *Under-sampling* reduces the number of majority class samples, while *over-sampling* increases the size of the minority class [36]. Under-sampling may lead to loss of information from the majority class, while over-sampling can result in overfitting (see Section 3.2.3.3). As loosing information can produce subpar results, we prefer over-sampling, along with other methods to reduce overfitting. More precisely, we select Synthetic Minority Oversampling Technique (SMOTE) [59, 129]. This method generates synthetic samples for the minority class without directly duplicating original data. It has been shown to correctly prevent overfitting and has been used in various network security-oriented works [74, 201].

In any case, only the *training* dataset is balanced while *testing* is kept intact. First, it avoids data leakage [115], since SMOTE selects certain data points during resampling, potentially including them in testing (see Section 3.2.3.1). Second, maintaining the natural imbalance in the testing dataset helps identify any biases present in the model by reflecting the original dataset's distribution [31].

3.2.3.3 Overfitting on training data

One common challenge of machine learning is overfitting, where a model learns to capture noise in the training data rather than the underlying patterns, resulting in poor performance on unseen data [31].

To avoid this, one can hold a part of the dataset from the training process, and use it to test the trained model. Such *cross-validation* is interesting but remains perilous: there is no guarantee that the selected test set is overall representative of the data. To improve this, one can leverage a *k-fold cross-validation* [119, 227], where the dataset is split into k smaller subsets. For each of the k subsets, a model is trained with $k-1$ subsets of data, and tested with the k_{th} remaining fold. As the test set changes every time, all data is used in a test at some point. Finally, performances are averaged over the k iterations.

Cross-validation is generally used to estimate performances on unseen data and to compare machine learning processes. Once it is done, one can train the best performing model on the whole dataset and deploy it in production for actual predictions.

3.2.3.4 Incorrect performance metrics

Performance metrics in machine learning are essential for assessing the effectiveness of models [31]. As said in Section 3.2.3.2, only the training set is virtually balanced: the testing set used to evaluate the performance of the model keeps the original distribution.

In classification tasks, the predictions are classified as:

- True Positive (TP): the model correctly predicts a positive outcome;
- True Negative (TN): the model correctly predicts a negative outcome;
- False Positive (FP): the model incorrectly predicts a positive outcome when the actual outcome is negative;
- False Negative (FN): the model incorrectly predicts a negative outcome when the actual outcome is positive.

Then, a classic evaluation metric such as the Accuracy (A) can be computed. It is the ratio between the number of correct predictions and the total number of predictions:

$$A = \frac{TP + TN}{TP + FN + FP + TN}$$

However, the accuracy may be skewed by the high number of correct predictions because of the class imbalance. For instance: in intrusion detection, a model biased toward predicting benign traffic would generate numerous TPs due to the rarity of attacks, resulting in a seemingly high accuracy despite its limitations.

Instead, we use the Balanced Accuracy (BA) [31]. It represents the mean of the sensitivity and specificity, correctly taking into account the class imbalance of the dataset:

$$BA = \frac{1}{2} \times \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right)$$

3.2.4 Summary

In this thesis, we leverage *supervised* machine learning based on labeled datasets extracted from network traffic. Additionally, we take into account the inherent *imbalance* of such datasets, by re-sampling the training dataset, using cross-validation, and measuring performance via the *balanced* accuracy.

Part II

Unveiling Vulnerabilities: Privacy Threats in LoRaWAN

Chapter 4

Linking LoRaWAN identifiers

Contents

This chapter explores how LoRaWAN separates the identity *from the* activity *of a device during the join process by using two identifiers in distinct, theoretically unlinkable messages. By leveraging a large-scale real-world dataset of more than 27 millions messages, we analyze various content, time, and radio-based domains features. We demonstrate that a machine learning process can reliably link back the two relevant messages, reaching a balanced accuracy of ∼0.999. Notably, we find that a single field of the header is particularly relevant during classification, and should be obfuscated.*

This chapter is partially based on our peer-reviewed and published work *Device Re-identification in LoRaWAN through Messages Linkage* [166]. The present chapter differs mainly by the following points:

- A larger dataset is leveraged and analyzed. Instead of roughly one year and a half, experiments are conducted on three years of LoRaWAN traffic, making it the most complete dataset to date in the literature [195, 196].
- We provide more details on the intuition behind the work and further justify the feature selection process.
- We streamline our approach to meet current methodology standards and conduct a thorough analysis of the `FCnt`'s impact on message linkage. Additionally, we explore the relevance of each feature using a more robust solution.

4.1 Introduction

LoRaWAN utilizes two identifiers: the `DevEUI`, and the `DevAddr`. Looking back at the privacy definition for IoT devices given in Section 2.2.1, this mechanism allows LoRaWAN to separate the *identity* (`DevEUI`) from the *activity* (`DevAddr`) of a device.[1] As illustrated by Figure 4.1, the `DevAddr` is sent to the device encrypted, preventing an eavesdropper from trivially associating it to a `DevEUI`. In that sense, both identifiers are by design *theoretically* unlikable.

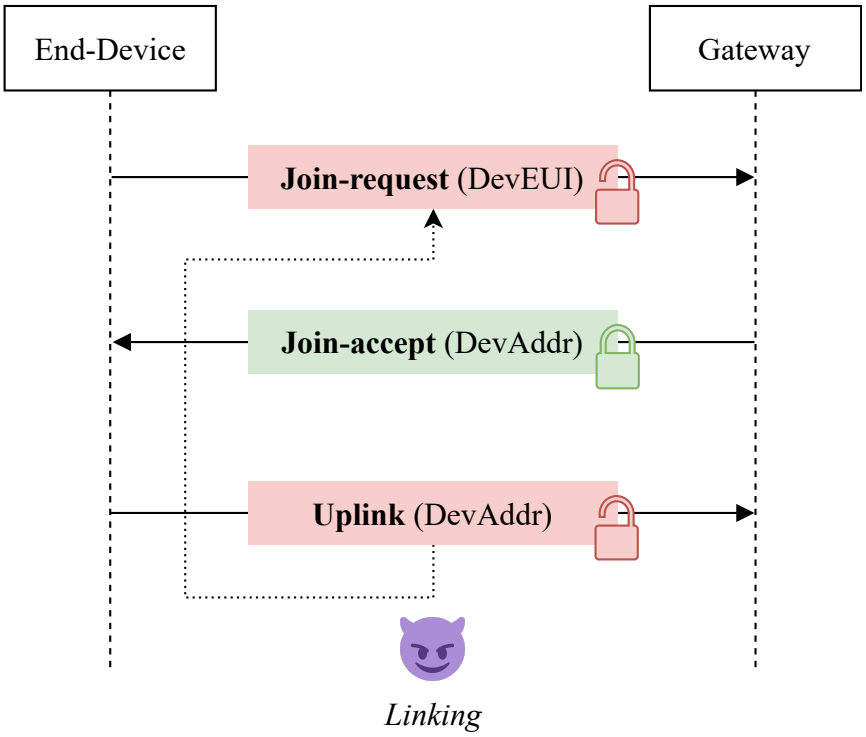


Figure 4.1: Linking LoRaWAN identifiers during the join process.

Linking back the `join-request` with the following `uplink` message opens up the path to two types of attacks. First, one can infer the identity corresponding to known activity traces. For instance, in the case of a smart parking lot [74], an attacker could precisely link network communications with a physical spot. Likewise, linking enriches the existing traces with additional context: the `DevEUI` contains an OUI related to the End-Device's manufacturer, with some makers focusing only on a certain type of devices (e.g. smart home sensors). Second, it enables tracking a known identity across its multiple communications if a disconnection or re-join occurs, by linking a set of `DevAddr` to a single `DevEUI`. Hence, an attacker could follow a specific mobile End-Device as it moves around between its (re)join process.

In this section, we propose a solution to link back the `DevEUI` sent in the `join-request` and the following `DevAddr`, contrary their initial design. First, we present the threat model under which we operate, before detailing the dataset used to carry out experiments. Then, we showcase the overall process and finish by discussing our experimental results.

4.2 Threat model

The generic threat model based on a passive eavesdropper presented in Section 2.3.1 can reasonably be adapted to LoRaWAN and the two types of attacks presented previously for multiple reasons:

- **Long range of transmission:** by design, LoRaWAN transmits over hundreds of meters in dense environments and kilometers in less crowded spaces [96]. Unlike eavesdrop-

[1] In practice, the `DevAddr` can also be considered a secondary yet temporary *identity*.

ping on Wi-Fi or BLE, where proximity is required (tens of meters at most), an attacker can easily deploy sniffing devices *outside* the immediate vicinity and thus stealthily capture traffic for longer periods of time.

- **Low financial investment:** an attacker can build a gateway using affordable hardware, such as a Raspberry Pi and a LoRa concentrator, for less than 200€ [106].
- **Multi-band access:** an attacker can listen to all bands simultaneously and thus have access to all `uplink` messages.
- **Publicly available demodulation:** while the LoRa radio modulation scheme is proprietary to Semtech, Robyns et al. have published an open-source SDR implementation [178]. Thus, an attacker are not limited to official commercial gateways and can build a listening station based on off-the-shelf hardware.

4.3 Dataset

To carry out our experiments, we utilize network traces captured via CampusIoT. This experimental platform is managed by the University of Grenoble[2] and focuses on teaching and research projects. Thanks to its private LoRaWAN network and ∼50 gateways deployed around the city of Grenoble, France, CampusIoT continuously listens to surrounding traffic on the EU 868MHz band. More specifically, gateways listen to 8 channels: 867.1, 867.3, 867.5, 867.7,867.9,868.1,868.3,868.5 MHz. While other additional channels can be used by private operators, they are not received by the current gateways and are irrelevant for our work.

We have access to these logs from the 23rd June 2020 to the 20th August 2023, representing a total of 419,688,857 LoRaWAN messages. Table 4.1 provides an overview of messages available to us. For more information on message types, please refer to Section 3.1.5.

Table 4.1: Distribution of message types in the CampusIoT dataset.

Message Type	Number of occurences
Unconfirmed Data Up	278,223,701
Join Request	81,205,339
Confirmed Data Up	34,491,710
Unconfirmed Data Down	11,133,851
Proprietary	6,617,186
Join Accept	3,101,259
Rejoin request	2,747,406
Confirmed Data Down	2,168,405

For the current experiment, only `join-request` and `uplink` messages for which we know the valid association are relevant. As this information is unavailable for third-party operators, we focus solely on CampusIoT's End-Devices, summing up to ∼1.2M `join-request`, and 26M `uplink` messages. This represents respectively ∼1.9k `DevEUI` and 36k `DevAddr`.

We note such a dataset is based on experimental data, where End-Devices may be affected by abnormal behaviors (e.g. high frequency of transmission). However, it is impossible to produce a reliable ground truth based on more realistic third-party traces because of the protocol's design.

4.4 Linking join requests and uplink messages

In this section, we detail a method leveraged to link a `DevEUI` with the corresponding `DevAddr`. To do so, we compare various features of the `join-request` with the ones from `uplink` messages following it during a 1-hour time window. If characteristics from both `join-request`

[2] https://campusiot.github.io/

and `uplink` messages are similar, there is a high chance both originate from the same End-Device. We begin by developing this intuition and further detail selected features, before presenting our machine learning method.

4.4.1 Intuitions

We motivate our process with two intuitions: based on the protocol design, and on the nature of End-Devices as well as their environment. First, the `FCnt` is initialized at zero after a join process, which should be a reliable indicator of the relevant `uplink` following a `join-request`. Additionally, LoRaWAN End-Devices should send their first `uplink` message soon after joining the network. The End-Device is not obligated to immediately utilize the newly acquired `DevAddr` upon reception. However, it is generally advantageous for a sensor to either request its configuration or start data transmission promptly. Figure 4.2 displays the distribution of times between a `join-request` (containing the `DevEUI`) with the first `uplink` (with the `DevAddr`). Generally, both messages are sent close to each other, with a majority following a ∼10 seconds delay.

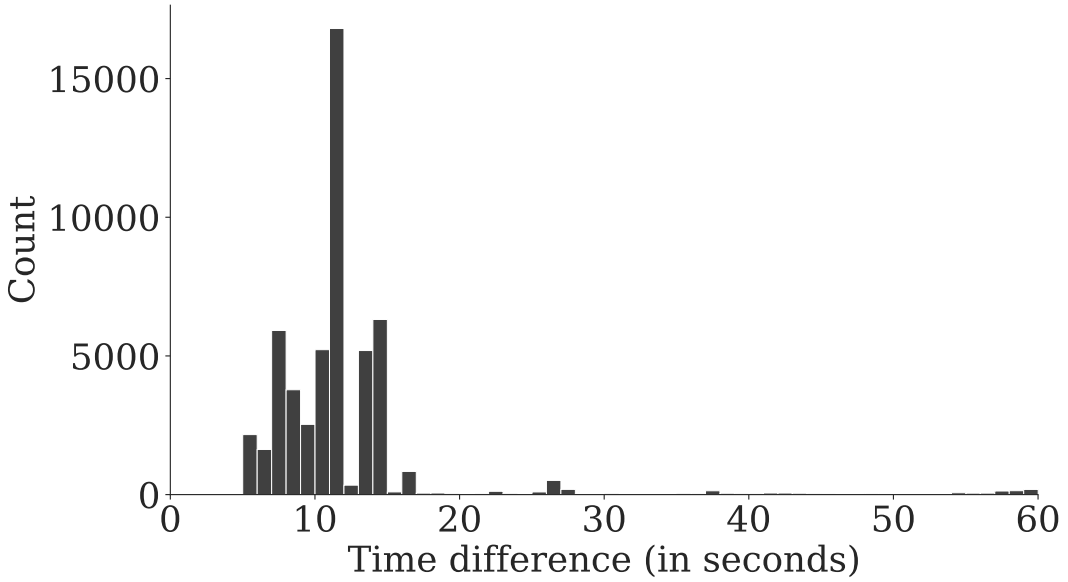


Figure 4.2: Time between a `join-request` and its associated first `uplink` message.

Second, End-Devices communication features should not vary significantly in between two messages. If static (e.g. metering sensors), devices should exhibit stable radio signals [25], as gateways distance and surroundings do not change. While mechanisms adapting transmission power exist and could influence radio-based features, they are generally used after 20 `uplink` messages, and not directly post join process [68]. If mobile (e.g. position trackers), other header fields should still help us link back the `DevEUI` and `DevAddr`.

4.4.2 Features

We extract content, time, and radio-based features from on our real-world dataset. To determine which ones to select, we study their distribution for the actual `uplink` directly following a `join-request` (valid pair), and for other unrelated `uplink` messages (invalid pair). This is possible because we control the network and possess the ground truth of associations between `DevAddr` and `DevEUI`. Some features correspond to a single value in the `uplink`, while others are distances compared with the `join-request` of reference or previously known information. A complete list is available below.

The feature is deemed important, i.e able to discriminate between valid and invalid pairs, if the variation from one category to the other is significant. Figure 4.3 shows the distribution discrepancy between normalized values from valid and invalid pairs, highlighting the difference between them. For instance, the `FCnt` is equal to 0 for the first `uplink` following

a `join-request`, with some outliers when packet loss occurs. While this does not guarantee relevance during the machine learning process, it hints at strong distinguishability.

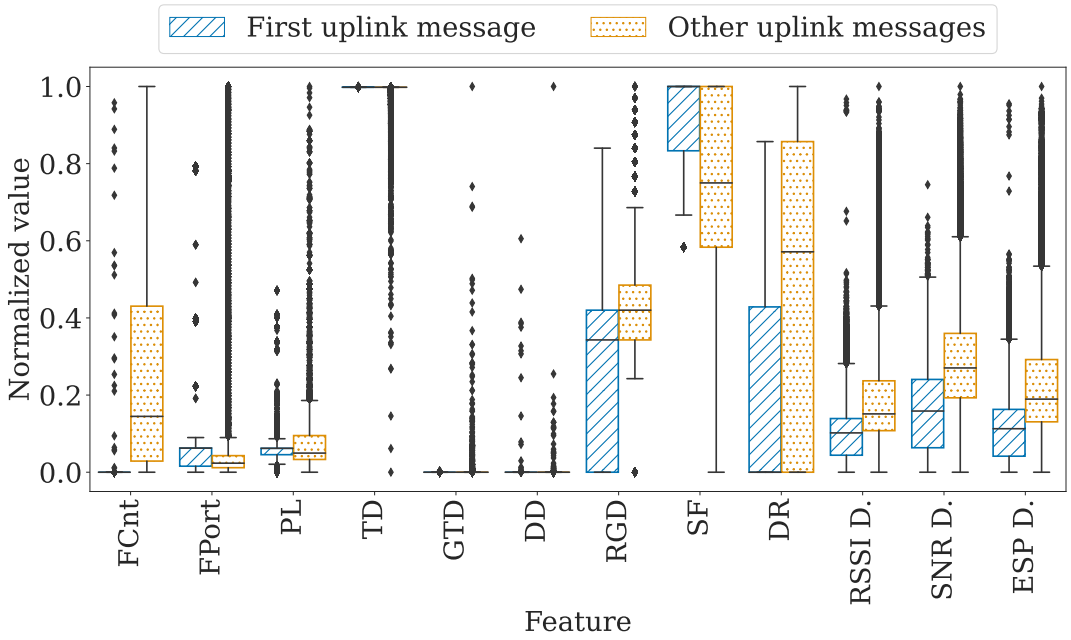


Figure 4.3: Distinguishing behavior of features based first `uplink` received after a `join-request`, or other `uplink` messages.

4.4.2.1 Content-based features

Content-based features are extracted either directly from the clear-text message header, or from metadata. We consider three of them: Frame counter, Frame port, and Payload length.

- *Frame counter* (`FCnt`): Initiated at zero after a join or re-join process and incremented by 1 with each subsequent message, up to 65535, the `FCnt` is included in each `uplink` message unencrypted. Used to detect packet loss, duplications, and re-order received messages, it also provides an easy way to spot the first `uplink`. Given the likelihood of packet losses [136], analyzing low frame counter values is essential.
- *Frame port* (`FPort`): Ranging from 0 to 255, the `FPort` allows developers to target specific applications. Its value may vary depending on the nature of the message; for instance, a port might be designated for management tasks such as configuration updates, while another might be assigned for sensor data transfer. Some values are reserved: 0 corresponds to MAC commands, 1 to 223 target applications, and 224 to 255 are utilized for the LoRaWAN MAC layer test protocol [65, sec 4.3.2]. Here, a specific value could be used during the first `uplink` to request configuration information, and another one for the rest of data transfer.
- *Payload Length* (PL): LoRaWAN payloads are small: from 0 to 242 bytes. Their length vary based on radio parameters and regions [21]. While payloads are encrypted, they are not padded, leaking some information about the underlying data length. Similarly to `FPort`, it could change depending on the `uplink` order.

4.4.2.2 Time-based features

Metadata can be extracted from traffic based on message timings. Three time-based features are selected: Time Difference, Gateway Time Distance, and DevAddr Difference.

- *Time Difference* (TD): The difference between timestamps of `join-request` and `uplink` messages is computed for the first gateway receiving them as an integer. Following previous works, we expect a low value for valid links [24].

- *Gateway Time Distance* (GTD): A message is sent once but can be received by multiple gateways with a slight delay. The difference of time of arrival at gateways is computed pair-wise and saved in vectors. A euclidean distance is computed between the vector produced by the `join-request`, and following `uplink` messages. Assuming numerous End-Devices are static, relative time differences and distances should be low.
- *DevAddr Difference* (DD): As seen in Section 3.1.6, the `DevAddr` is randomly generated during the join process. Thus, we compute the last time such an identifier has been seen, expecting the correct `uplink` to be its first occurrence.

4.4.2.3 Radio-based features

The radio layer provides multiple interesting raw features that should be similar for both the `join-request` and following `uplink` message. First, messages from static End-Devices are transmitted over a stable environment, and received by a consistent list of gateways. This produces similar values for classic quality-of-medium features depending on the radio power (RSSI, SNR, and ESP). Second, some radio parameters may change based on the device or the period of communication (Spreading Factor & DataRate).

- *Receiving Gateways Distance* (RGD): Static devices should be received by similar sets of gateways for both the `join-request` and subsequent `uplink` messages. Hence, we compute the Hamming distance using gateways as vector indexes, expecting low distance values for valid links.
- *Spreading Factor (SF) & DataRate (DR)*: Radio configuration utilizes default values for the first `uplink` message(s) and may change for later ones. Transmission information such as the data rate (number of bits transmitted per second, from 0 to 7) and SF (number of bits encoded per symbol, from 7 to 12) are not directly linked to the `uplink` order. However, they are determined dynamically based on the network's Adaptive Data Rate (ADR) mechanism, which re-evaluates the link quality after 20 `uplink` messages by default [68]. This optimization adjusts SF and DR accordingly, with potential variations expected between first and subsequent `uplink` messages.
- *SNR*: Defined as the ratio between received power signal and the noise floor power level, the Signal-to-Noise Ratio (SNR) can be positive or negative, depending on the transmission quality. In our dataset, the SNR varies from -26 to +30.5.
- *ESP*: Derived from both RSSI and SNR, the Estimated Signal Power (ESP) is used to compare the channel quality in radio communications. It is an integer ranging from 0 to -160dBm in our dataset, providing more precise values than the RSSI when its value drops around -120dBm. Experiments show that, contrary to the ESP, the RSSI distribution saturates when signal is way below the noise floor [14]. It is computed via the formula 4.1.

$$\begin{aligned} ESP_{dBm} = RSSI_{dBm} + SINR_{dB} \\ -10\log_{10}(1 + 10^{0.1SNR_{dB}}) \end{aligned} \tag{4.1}$$

RSSI, SNR, and thus ESP, are available for each receiving gateways. Hence, we compute the euclidean distance for each of these values between the `join-request` and `uplink` messages, and expect low values for valid links (RSSI D., SNR D. ESP D.).

4.4.3 Method

In order to link a `join-request` with the following `uplink` messages, we consider a machine learning-based method utilizing features detailed in Section 4.4.2. The problem can be designed as a binary classification: either a given `uplink` corresponds to a `join-request` (valid pair), or it does not (invalid pair). In this section, we present steps followed to prepare the data, considered classifiers, performance metrics, and overall process.

4.4.3.1 Data preparation

We extract messages from the original dataset (cf. Section 4.3), with a set $\mathcal{J}$ containing `join-request` messages, and $\mathcal{U}$ for `uplink` messages. We then leverage a set of known links $\mathcal{K}$ to create two classes. These are available thanks to our administrator access on the NS. Based on this ground truth, we select valid pairs $(j, u_v) \in \mathcal{K}$, with $j \in \mathcal{J}$ and $u_v \in \mathcal{U}$. Invalid pairs $(j, u_i) \notin \mathcal{K}$, with $u_i \in \mathcal{U}$, are generated by randomly selecting leftover `uplink` messages. This ensures the second class is heterogeneous and contains a vast variety of associations.

In both cases, features are extracted as vectors from `join-request` and `uplink` messages along their state (valid or invalid), and fed to the machine learning models. At the end of the process, the complete dataset contains 56 133 valid links, and 256 003 invalid ones.

4.4.3.2 Considered classifiers

Taking inspiration from the work of Acar et al. [15], who employed multiple machine learning methods to classify smart-home network traces effectively, we adopt a similar approach. We select the following classifiers: Decision Tree (DT), Naive Bayes (NB), Logistic regression (LR), K-Nearest Neighbours (kNN), Random Forest (LR), AdaBoost (AB), and LightBGM (LBGM). Contrary to previous works [15], we upgrade XGBoost with LightBGM, an equivalent but faster gradient boosting method [116].

Our goal is to find at least *one* machine learning method among those that can convincingly carry the attack, proving that it is indeed possible to link back a `join-request` with the following `uplink` messages.

4.4.3.3 Training and evaluation

As described in Section 3.2, we follow a classic machine learning approach. The dataset is split into training/testing (75%) and held-out (25%). Each pair of `join-request` and `uplink` is independent: the existence of one pair has no impact on the existence of another. Therefore, we can randomly sample pairs to create the training and testing datasets without the risk of data snooping (see Section 3.2.3.1). We use SMOTE on the *training* set to reduce the imbalance and stratified 5-fold cross validation to avoid overfitting on training data (see Section 3.2.3.2). Finally, we repeat the process 15 times with different random seeds to smooth results and select the balanced accuracy as a performance metric.

4.5 Experimental results

In this section, we present the various results obtained through the machine learning evaluation process.

4.5.1 Models comparison

We train each model indenpendently and compare their performance in Table 4.2. We note the balanced accuracy remains stable during cross validation, hinting at good generalization properties. All models are highly accurate when linking `join-request` with the corresponding `uplink` messages, with Naive Bayes (NB) lagging behind others. Random Forest (RF) reaches a nearly perfect $\sim$0.9985 balanced accuracy, showing that a) a linking attack is possible, and b) it is extremely reliable.

4.5.2 Features importance

To study the importance of each feature used by machine learning models, we carry out a *permutation feature importance* analysis. The method involves systematically permuting the values of a single feature while keeping the others constant and observing the impact on the balanced accuracy. By measuring the change in performance, one can assess how much each of them contributes to the predictive power of a given model. Features that lead to a significant drop in performance when shuffled are considered more important, as they contain valuable information to make predictions. We note this does not predict the generic future importance of a feature, but rather indicate how relevant it is for a particular model.

Table 4.2: Performance comparison of various classifiers via 5-fold cross validation.

Classifier	TP	TN	FP	FN	Balanced accuracy
RF	14030	63866	115	16	0.9985
LGBM	14027	63855	15	136	0.9985
AB	14023	63760	222	13	0.9979
kNN	14011	63713	267	25	0.9970
DT	13960	63877	115	79	0.9962
LR	13949	63004	993	67	0.9896
NB	13878	40817	23194	153	0.8114

The process is repeated 15 times for each random seed to smooth values. Following Section 4.4.2, we expect the `FCnt` to have significant influence on the results due to its straightforward exploitation. Additionally, it can be easily obfuscated via encryption (see Section 6.2.1.1) and we anticipate a future removal from available features.

Results are presented in Figure 4.4 (high value means high impact), from which we can draw multiple insights. First, both `FCnt` and Time Difference are strongly relevant. This is expected, as nearly all `uplink` messages have a `FCnt` equal to 0. Likewise, the time between `join-request` and the following valid `uplink` is distinctively low (cf. Figure 4.2). Second, when removing the `FCnt`, models tend to favor the Time Difference and Frame Port (as well as the Payload Length more marginally), while other features remain low importance.

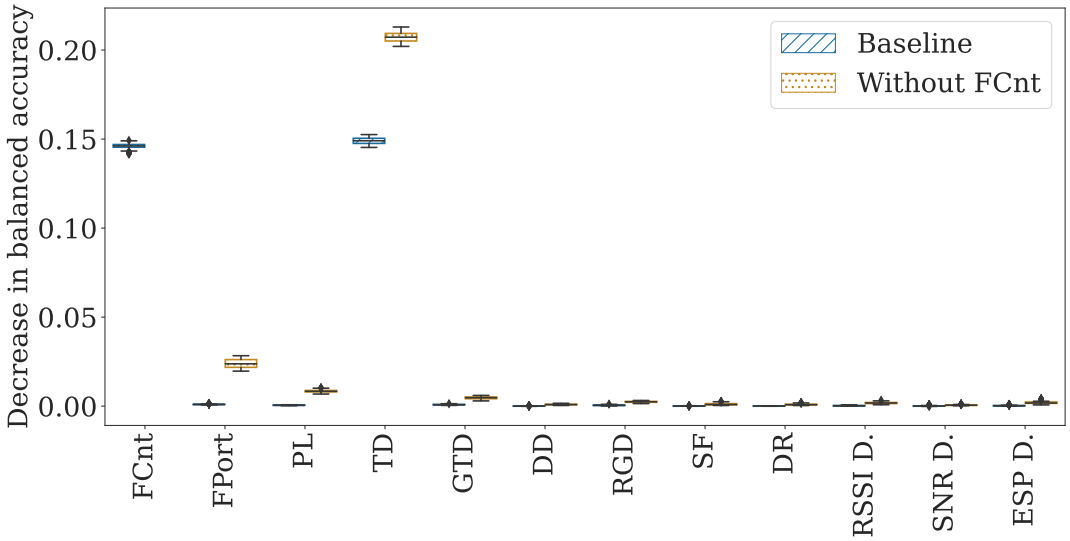


Figure 4.4: Permutation feature importance.

To complete this analysis by anticipating a possible obfuscation of the `FCnt`, we train and compare models with all features (*Baseline*), with all features but no `FCnt` (*Without `FCnt`*), and with *`FCnt` only*.

Figure 4.5 shows that all machine learning models (with Naive Bayes (NB) as a notable exception) follow a similar pattern: removing the `FCnt` slightly decreases performance, further reduced when using `FCnt` only. For instance, Random Forest decreases from ∼0.999 to 0.997 when removing the `FCnt`, and as low as ∼0.986 with `FCnt` only.

First, this confirms that other features are able to effectively compensate a lack of `FCnt`: the attack would work even if it was not available. Second, the `FCnt` is in itself highly relevant for an eavesdropper, achieving high accuracy using only that piece of information.

4.6 Conclusion

We demonstrate the robust association of two theoretically unlinkable messages: the `join-request` and the corresponding first `uplink` message. Through this linkage, we establish a connection

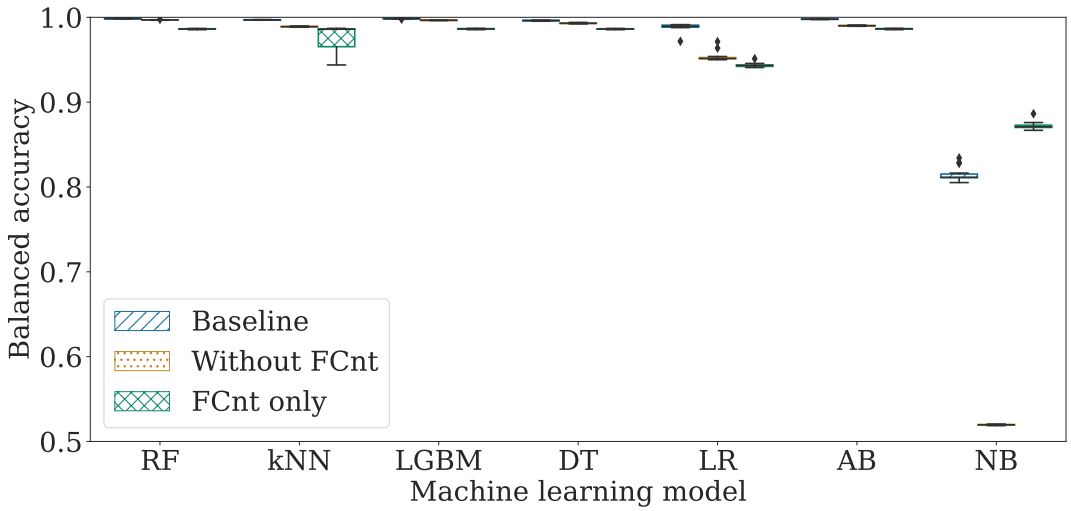


Figure 4.5: Comparison of machine learning models performance.

between the *identity* (`DevEUI`) and *activity* (`DevAddr`) of a LoRaWAN End-Device. Our machine learning model matches both identifiers with a ∼0.999 balanced accuracy, revealing that additional efforts on the protocol design are required. Notably, we underscore the significance of the `FCnt`, which enables accurate linking of both messages, even when used alone. Our analysis of each selected feature underscores the critical role of timing between `join-request` and `uplink`, calling for additional countermeasures.

While linking identifiers poses a privacy risk, other potential attacks on communication data warrant attention. Given our ability to reliably link identity and activity, in the next chapter, we investigate privacy threats associated with the activity itself.

Chapter 5

Fingerprinting LoRaWAN devices

Contents

This chapter studies how communication patterns of LoRaWAN devices can be leveraged to accurately identify them. After analyzing relevant features and proposing a new, holistic approach to fingerprinting, we utilize a large and realistic dataset consisting of 41 million ***`uplink`*** *messages from third-party professional operators. Our machine learning approach successfully re-identifies the origin of message sequences with a ∼0.98 balanced accuracy. Supported by various scenarios, such as identifying mobile devices and utilizing a limited number of listening stations, we show that fingerprinting LoRaWAN devices is both largely applicable and robust.*

This chapter is partially based on *Introducing Multi-Domain Fingerprints for Lorawan Device Linkage* [167] (currently under review). The present work differs mainly by the following points:

- We expand motivations and further introduce fingerprinting as an additional security measure to monitor wireless networks.
- We offer additional insights into the dataset used in our research, providing a more comprehensive overview of its characteristics and composition.
- We enhance the justifications for feature selection, offering more technical explanations to support our choices.
- We outline the limitations inherent in our approach, acknowledging potential constraints and challenges that may affect the interpretation and application of our findings.

5.1 Introduction

Linking the `join-request` with the following `uplink` messages is only one of the steps in tracking LoRaWAN End-Devices. The flow of messages is equally interesting for eavesdroppers. It contains all the actual communication data and metadata, revealing information about the *identity*, *activity* and *location* of a device.

End-Devices are distinguished by their `DevAddr`, which may change throughout their lifespan, after each rejoin process or disconnection event. This process can be frequent (cf. Table 4.1), and we find that around 50% of devices in our year-long dataset use an identifier lasting less than a week. While such high numbers can be explained by short activity periods, experiments and packet losses, other mechanisms including pseudonyms rotation, already utilized in BLE [92, 143], may be further increase them in future versions of LoRaWAN if they are deployed. In any case, a changing pseudonym breaks the flow of data for an eavesdropper, leaving them with only part of the whole communication.

Fingerprinting is a generic tool for network traffic analysis, enabling tracking devices based solely on their characteristics and bypassing pseudonym-based protections (see Section 2.3.5). While already utilized across various protocols such as BLE [19], Wi-Fi [174, 213], and Web browsing [121, 161, 189], it has seldom been studied in LoRaWAN networks.

In this chapter, we explore fingerprinting LoRaWAN End-Devices based on their distinct communication patterns. We begin by outlining our motivations and the threat model under which we operate. Then, we present the dataset leveraged during experiments and its relevant characteristics. After proposing a new fingerprint representation, we illustrate its usage via a method to link back sequences of `uplink` messages generated before and after a change of `DevAddr`. We end by showcasing the experimental results of fingerprinting End-Devices, and by exploring some limitations of our approach.

5.2 Motivation & threat model

A comprehensive study of fingerprinting in the specific context of LoRaWAN (and more broadly, on the LPWAN family) has never been done. As such protocols show fundamentally different communication patterns than other wireless technologies (see Section 3.1.7), it is valuable to provide an overview of their fingerprintability. While Section 2.3.5 presents fingerprinting as a tool leveraged against privacy, it can be also be used defensively (e.g. in intrusion detection scenarios). In this section, we explore both visions, and refine the threat model accordingly.

5.2.1 Fingerprinting as a privacy threat

Existing attacks against privacy in wireless networks are enabled mainly by stable identifiers. For instance, grouping the traffic based on the `DevAddr` allows to 1) identify the sending

device, and 2) link the trace to an activity (e.g. occupying a parking space) [74]. As explored in Section 2.4.5, some protocols support identifier rotations through pseudonyms (e.g. in Bluetooth) [92]. While LoRaWAN does not support rotating addresses yet (see Chapter 7), a disconnection or rejoin process induce the NS to generate a new `DevAddr`. An End-Device can trigger this process independently of its application traffic by sending a "*Rejoin-Request message*" [65, sec. 6.2.4]. Additionally, the server can initiate a rejoin via a "*ForceRejoinReq*" command [65, sec. 5.13]. As seen in Table 4.1, this feature is already employed in real-world deployments, thwarting tracking and profiling solutions.

Fingerprinting enables attackers to establish a consistent device identity by leveraging diverse (meta)data. Contrary to others consumer-grade IoT devices such as smartwatches, LoRaWAN devices are generally tied to a specific activity (e.g. water metering). Hence, identifying the devices is usually enough to gain information on the underlying activity. However, fingerprinting can also serve as a tool for more general activity inference, by comparing the targeted traffic with known, generic communication patterns. In this study, we focus on device identification as a proof-of-concept.

5.2.2 Fingerprinting as security protection

LoRaWAN faces a multitude of security threats, as outlined in existing literature [159]. Such attacks range from denial-of-service [99] to MitM [154], jamming [100] and replaying packets [159], all of which have the potential to severely disrupt communications and compromise network integrity. Additionally, devices can be taken over, through physical access or software, leading to cryptographic material leakage. An adversary could then impersonate a fake device, send spurious information, and escalate their attacks on the network.

To counter those threats, and more generally to detect intrusion in wireless networks, device fingerprinting has been presented as a promising approach [28, 71, 117, 217]. Building fingerprints for known devices can be used by two security features.

First, it enables another soft authentication mechanism by identifying signals coming from external devices. For instance, Arackaparambil et al. leverage this technique to identify trusted 802.11 access points and detect fake ones [28]. In LoRaWAN, Danish et al. demonstrate that legitimate devices can be detected during the join process, thus reducing the impact of specific jamming attacks [71]. Such an approach rarely utilizes fingerprinting alone, as it can not provide sufficient security guarantees. It is viewed as support of other cryptographic-based authentication mechanisms [28].

Second, a network administrator could detect changes in behavior in previously genuine End-Devices over long periods of time and thus dynamically identify compromise of elements in the network. Fingerprinting thus provides a way to reinforce the security of LoRaWAN networks by detecting malicious devices [190].

5.2.3 Updated threat and security model

We combine both offensive (against privacy) and defensive (for network security) fingerprinting under a single use-case. Instead of referring to an "*attacker*", we outline the goal and capabilities of a "*fingerprinting actor*". As illustrated by Figure 5.1, this actor eavesdrops LoRaWAN communications during an initial period and later attempts to link new `uplink` messages to previously recorded traffic. Such a setup induces a privacy breach in the offensive case, and a detection of malicious device in the defensive one.

In offensive fingerprinting, the actor strategically places LoRaWAN listening stations within the target area. As already demonstrated in Section 4.2, this setup is realistic based on both protocol's design and easy access to LoRaWAN hardware. In contrast, the defensive fingerprinting actor controls the infrastructure and has direct access to this data.

In both scenarios, data collection is passive, with no injection or modification of wireless messages by the actor. We assume encryption remains unchanged, and the content of encrypted payloads is inaccessible to an eavesdropper and to the *network* administrator (on the NS). While an *application* administrator (on the AS) could potentially access clear-text payloads for defensive fingerprinting, such a scenario falls outside the scope of our current study. Regardless, we obtain a high linkage accuracy without clear-text payloads data.

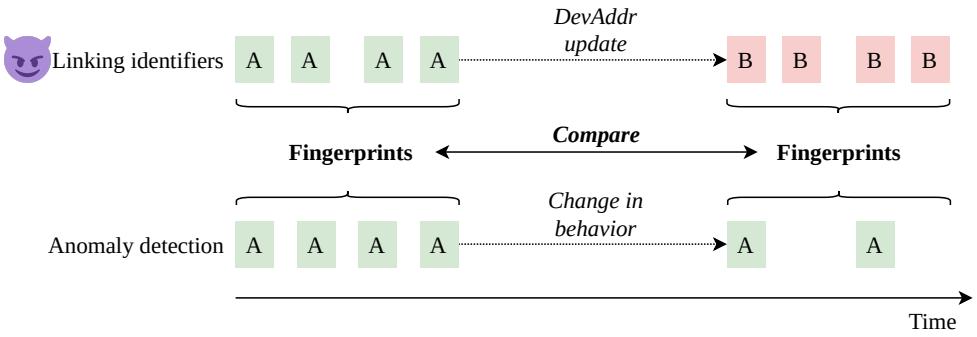


Figure 5.1: Offensive and defensive fingerprinting of LoRaWAN devices.

5.3 Dataset

For the current work, we utilize a subset of the dataset previously employed for linking, as discussed in Section 4.3. We focus on December 2021 to September 2022, because this period shows regular activity, corresponding to 41 million `uplink` messages, a significant enough number to draw insights from. Instead of relying on traffic generated directly by CampusIoT to establish ground truth, we study the data of third-party operators. This approach enables us to analyze realistic traces from professional deployments, aiming for high accuracy in fingerprinting real-world traffic. The counterpart is a lack of knowledge of the actual deployment nature: it is impossible to know precisely which End-Devices are communicating, and from where. As shown in Section 5.7.3, we find alternative ways to overcome these limitations.

5.3.1 Selecting relevant data

We clean up the dataset to select only relevant data for our evaluation using multiple methods. First, we exclusively study `uplink` messages. LoRaWAN is primarily designed for device-to-server communications and we find that `uplink` messages provide enough information to reliably fingerprint End-Devices. Additionally, we lack access to third-party downlink messages, as they are not monitored by the gateways of CampusIoT.

Second, we exclude data generated by known experimental devices [107], notably every End-Device using a `DevAddr` below 0x03FFFFFF, as they are officially reserved for experiments. Tests often exaggerates End-Device behavior due to controlled and artificial network conditions. This can lead to communication patterns not representative of real-world scenarios, such as devices transmitting as frequently as every minute, which is uncommon in actual deployments to preserve battery.

Similarly, The Things Network is an operator known for its openness towards hobbyists, allowing anyone to register an account for free and start experimenting.[1] While this approach is valuable for the community, it may generate significant unwanted noise in the dataset. For the same reason, we exclude traffic from both The Things Network and CampusIoT. In the initial dataset, ∼10% of `uplink` messages belong to experimental addresses, 35% to CampusIoT, and 16% to The Things Network. While we take steps to remove irrelevant data sources from the dataset, we acknowledge the possibility that professional third-party operators may occasionally use their address space for experimental purposes. However, we believe that such instances are infrequent enough that the majority of their traffic remains relevant to our experiments.

Finally, our experiments exclusively deal with `DevAddr` that appear at least five times in the complete dataset (see Section 5.6.2), and we discard the rest. In the end, we keep 41 million messages originating from more than 50 third-party operators. They correspond to ∼25,000 unique `DevAddr`, with 140,000 daily messages on average.

[1] `https://www.thethingsnetwork.org/`

5.3.2 Characterizing `uplink` messages

Our fingerprinting solution groups messages based on their `DevAddr`. Figure 5.2 displays the percentage of `DevAddr` sending at least a given number of `uplink` messages. A significant portion (67%) only appears once in the dataset, attributable to two potential factors. First, our dataset encompasses third-party End-Devices that may be situated at the edge of the gateways' range, resulting in only one message being received. Second, occasional errors or experimental activity may have caused some addresses to appear less often.

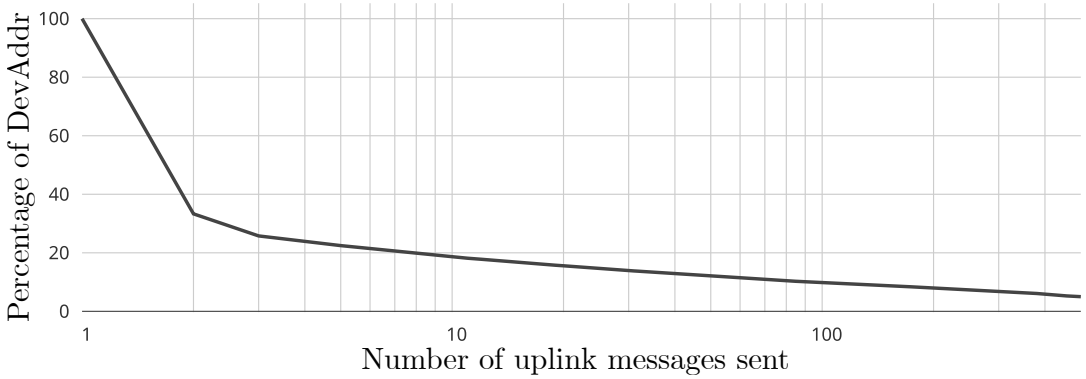


Figure 5.2: Reverse cumulative distribution of the number of `uplink` messages received by `DevAddr`, after cleaning the dataset.

As highlighted in Section 4.5.2, time-based features play a significant role in linking the `join-request` with following `uplink` messages. To further study `uplink` messages, Figure 5.3 displays the distribution of time window between each message, named IAT. IAT measurements are concentrated around recognizable values, such as 10, 15, 20 minutes. Albeit with fewer occurrences, this periodicity is also observed for full hours, including 24 and 48 hours.

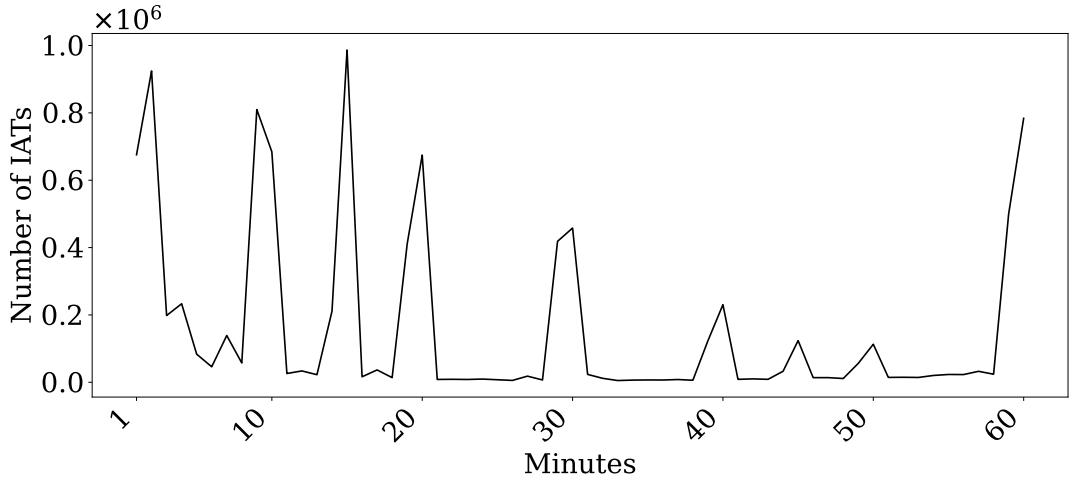


Figure 5.3: Distribution of IAT occurrences (limited to 60 minutes for clarity).

5.4 A new fingerprinting representation

As presented in Section 2.3.5.1, fingerprints can be formatted in multiple data representations. On top of studying the various state-of-the-art solutions, we propose our own approach: the *holistic fingerprint*. Instead of selecting one of the representations, we leverage all of them at once (vectors of values v, histograms h_b, descriptive statistics including mean $\overline{v}$, variance σ^2, standard deviation σ, skewness γ_1, and kurtosis $Kurt$, as well as Markov chains P):

$$F = (v, h_b, \overline{v}, \sigma^2, \sigma, \gamma_1, Kurt, P)$$

The machine learning approach alleviates concerns regarding the high dimensionality of the fingerprint. Through this method, models can effectively identify patterns and prioritize the most pertinent features, thus minimizing the impact of the representation's complexity.

In contrast to simpler formats like vectors of values, combining multiple representations simultaneously, including Markov chains, requires computing distances between matrices. We consider this feasible for two reasons: 1) fingerprinting occurs outside constrained nodes, and 2) LoRaWAN is designed as a low-throughput protocol.[2] This ensures sufficient power and time between each message to compute fingerprints, even dynamically.

5.5 Fingerprint-based linkage

Our 3-step methodology aims to ascertain whether two *sequences of* `uplink` *messages* originate from the same End-Device; it is illustrated in Figure 5.4. Following previous work on linking `join-request` and `uplink`, we extract features from three domains: content, time, and radio. These features are utilized to create fingerprints based on the representations outlined in Section 2.3.5.1. Subsequently, a distance computation is performed between the two fingerprints. The resulting vector of distances is then inputted into a supervised machine learning model, which finally classifies it as either *linked* (indicating both sequences originate from the same End-Device), or *not linked* (suggesting different origins).

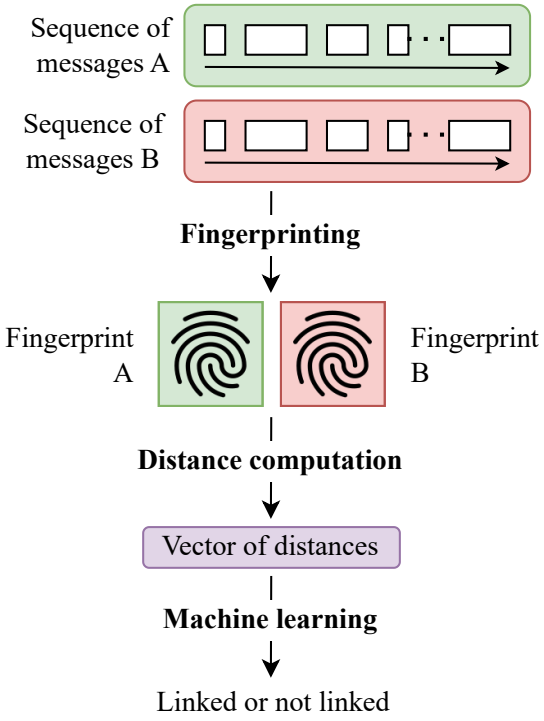


Figure 5.4: Overview of the 3-step fingerprint and linking process.

We select similar features as the ones presented in Section 4.4.2, focusing on both stable and possibly *stateful* features. We remove low impact ones based on our previous results in Chapter 4 and empirical study. Notably, we observe that parameters derived from the Adaptive Data Rate mechanism, such as Spreading Factor and DataRate, do not contribute significantly to the fingerprint. Additionally, the *Receiving Gateways Distance* is excluded because similar information is already available via other radio-based features saved in gateway-based vectors (RSSI, SNR, and ESP). A summary of the 8 selected raw features corresponding to the same three domains (content, time, and radio) is available in Table 5.1.

[2] We also note that, in our case, matrices are sparse due to the limited number of states an End-Device occupies.

Table 5.1: Summary of features leveraged for linking.

Domain	**Raw feature**	**Considered stateful** (Markov chains)
Content	`FPort`	YES
	Payload length	YES
Time	Arrival time	YES
	(refined as: Hour of day, Day of week)	
	Inter-Arrival Time	YES
Radio	RSSI	NO
	SNR	NO
	ESP	NO

5.5.1 Content-based features

We select two content-based features: the `FPort`, and payload length. Both display significant values on their own, along potentially interesting variations across a sequence of messages. Following works on Web fingerprinting by Shen et al. and Pan et al. [161, 189], we combine the `FPort` and payload length as a single Markov chain state and empirically confirm that combining them yields better results. This is possible thanks to the low number of available values for both raw features: the resulting matrix remains small.

The `FCnt` is only relevant to re-order messages and lacks meaning during the fingerprinting process. We thus exclude it from raw features, but use it throughout to correctly generate states of Markov chains for on other features. Additional header-based features *could* be added, such as `FOpts`. However, as noted in Chapter 7.6.1, remaining header fields lack sufficient entropy in our dataset and do not actually improve fingerprinting accuracy.

5.5.2 Time-based features

We select two new time-related features: Arrival time, and IAT.

- *Arrival time*: Defined as a Unix timestamp precise to the millisecond, it denotes the moment a message is received by a listening station.
- *IAT*: The time interval between two consecutive `uplink` messages is computed based on the first listening stations receiving them. Contrary to others features, the IAT requires two messages.

Both features target End-Devices programmed to send recurring reports, for instance communicating sensor data every hour with an occasional management frame or alert. They are refined in two ways.

First, the arrival time serves as the basis for deriving additional time-based features, including the hour of reception (in ranges of 10 minutes, e.g. 2:30 PM) and the day of the week (e.g. Thursday).

Second, the millisecond precision measured by the listening station is too high to create significant states, due to various latency and treatment delays, long propagation over the air, and clock drift between the NS and End-Devices. For example, a message every 3600042 or 3599068 milliseconds roughly corresponds to 1 hour in both cases. To reduce the precision, we bin values and generate multiple sizes of bins, letting the machine learning model pick the most relevant one.[3] Based on the apparent periodicity shown in Figure 5.3, we select peaks corresponding to more than 1% of the total number of IAT values, translating them into bins of 1, 5, 10, 15, 20, 30, 40 minutes, as well as 1, 2, and 4 hours.

Finally, as seen in Section 5.3, a majority of IAT values clusters *around* rounded increments (e.g. a message every hour). Figure 5.5 illustrates how values theoretically sent at exactly t_j and t_{j+1} can be misclassified into the wrong bin due to various time perturbations. By shifting the bins b_i and b_{i+1}, we are able to group clustered values in the correct context. As

[3] We empirically find that manually selecting only a subset of bins is detrimental to final results.

depicted in Figure 5.3, the time drift never exceeds 3 minutes around significant values. To err on the side of caution, we shift bins by 5 minutes.

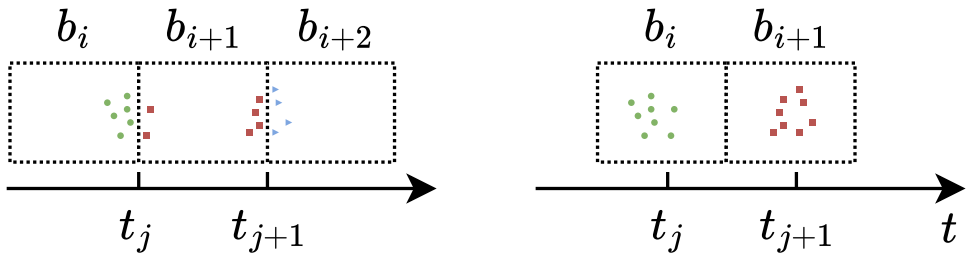


Figure 5.5: Shifting bins to correctly classify clustered measurements.

5.5.3 Radio-based features

We select three radio features: RSSI, SNR, ESP, based on concepts already explored in Chapter 4 for `join-request` and following `uplink` messages. The environment surrounding an End-Device typically experiences minimal changes, except in cases involving mobile objects equipped with tracking sensors. Consequently, sequences of messages originating from the same End-Device are expected to demonstrate similar radio-based features. This is confirmed by related works showing that RSSI exhibit stability in scenarios involving geographically static End-Devices [25].

Finally, we do not represent radio-based features as Markov chains. Radio power does not depend on the internal application state of a static End-Device, contrary to the length, port, or IAT. For instance, there should be no difference between the radio power corresponding to an application message compared to an update message.

5.6 Evaluation methodology

In this section, we outline the process of dataset creation and the generation of its groundtruth. Additionally, we detail the specifics of the machine learning techniques employed and discuss the performance metrics utilized in our analysis.

5.6.1 Dataset generation

To produce a fingerprint dataset, we first extract *sequences* of `uplink` messages labeled by their `DevAddr`. We then process each sequence to extract relevant features and format them following the fingerprint representations detailed in Section 2.3.5.1. As seen in Section 2.3.5.2, there are numerous distance functions to compare two fingerprints. We select the most straightforward ones for ease of implementation: statistical aggregates features (e.g. average or distributions) are compared via Euclidean distance, and we utilize the Euclidean norm (or l^2-norm) for matrices representing Markov chains.

Dealing with third-party operator traffic raises two issues: there is no guarantee that a specific `DevAddr` a) corresponds to the same End-Device for the whole duration of our capture as it can be re-assigned, or b) belongs to one and only one End-Device as multiple End-Devices could share the same identifier.

To address a), we introduce a period of inactivity after which the `DevAddr` is considered associated with a new End-Device. We choose $T = 1$ week under the assumption that the majority of End-Devices transmit at least one message per week. If a device communicates less often than T, it produces too few messages per sequence; the rest of our process excludes it from evaluation (see Section 5.6.2).

To address b), we identify multiple End-Devices using the same `DevAddr` via their uncoordinated `FCnt`. We detect any sequence of messages with unordered `FCnt` values and discard them. We note this also filter out malfunctioning End-Devices with ordering issues.

5.6.2 Groundtruth: labeling sequences

Section 3.2 emphasizes the necessity of a dataset with *labels* (i.e. ground truth) for supervised machine learning. Hence, vectors of distances between fingerprints are labeled based on the origin of sequences: *linked* pairs of fingerprints originate from the same End-Device, and *not linked* are emitted by distinct End-Devices. Once generated, these pairs can be leveraged to train our machine learning model.

To produce a ground truth from third-party operators' traffic, we segment sequences of `uplink` messages based on already publicly known origins, i.e. `DevAddr`. For each set of `uplink` messages S identified via its `DevAddr` and corresponding to a single End-Device, S is divided into chronologically ordered sequences $S_1, S_2, ..., S_m$, each containing c messages. As seen in Figure 5.6, a `DevAddr` corresponding to M messages generates $\lfloor \frac{M}{c} \rfloor$ sequences.

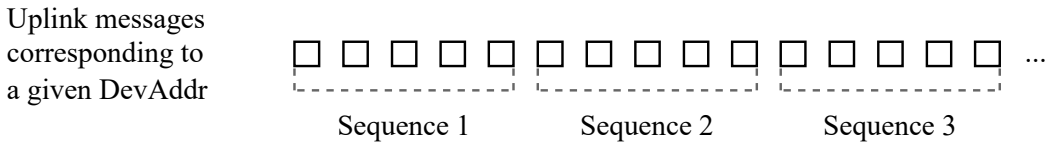


Figure 5.6: Splitting a set of `uplink` messages coming from the same End-Device into sequences of length 5.

In practice, we select $c \in \{5, 10, 20, 50, 100, 200\}$ based on the distribution of `uplink` messages received by a `DevAddr` (see Figure 5.2). For instance, if a set contains 233 messages, a total of 46, 23, 11, 4, 2, and 1 sequences of lengths 5, 10, 20, 50, 100, and 200 are respectively created. End-Devices associated with a significant number of messages in the dataset generate a large number of sequences, resulting in over-representation, particularly for short-length sequences. To address this issue, we limit the maximum number of sequences per End-Device for each length to 3.

Finally, we virtually match sequences of *correct* and *incorrect* origins to generate both *linked* and *not linked* vectors of distances. With the `DevAddr` identifying the origin of each sequence, we can precisely label the vectors, thereby establishing a reliable ground truth.

5.6.3 Linkage via machine learning

The objective of the linkage process is to determine whether two distinct sequences of messages originate from the same End-Device. This process entails applying the previous method to a pair of sequences: fingerprints are generated for each sequence, distances are computed accordingly, resulting in a vector that is then given to the pre-trained machine learning model. The model finally generates a binary prediction: linked or not linked.

For this task, we follow a classic machine learning approach already presented in Section 3.2. As the dataset is imbalanced (75% incorrect to 25% correct pairs), we equalize the training set using SMOTE [59] to reduce biases (see Section 3.2.3.2). After going through 5-fold cross validation, we select the Random Forest algorithm, following previous results on identifier linkage (see Section 4.5). To reduce data snooping [31] (see Section 3.2.3.1), we make sure fingerprints generated by the same End-Device (i.e. using the same `DevAddr`) are *either* in the training or testing dataset.[4]

Finally, the evaluation process is repeated 15 times with different seeds initializing Pseudorandom Number Generators (PRNGs). Results are presented using the Balanced Accuracy (BA), with graph axes limited from 0.7 to 1 whenever possible for clarity.

[4]This method effectively avoid fingerprints coming from the same `DevAddr` to be used in both training and testing. However, it does not protect from End-Devices updating their `DevAddr` and randomly being present in both datasets. Given the relative rarity of re-join processes (as indicated in table 4.1) and the inability to detect such behavior third-party traffic, we believe this unlikely event does not significantly impact the overall process.

5.7 Experimental results

This section presents the evaluation of the fingerprinting and linkage process, using the 41 million messages real-world dataset presented in Section 5.3. Unless stated otherwise, all available listening stations are utilized.

We begin by assessing the effectiveness of our approach and investigate the influence of different feature domains. We then present various scenarios, including mobile End-Devices and limited number of controlled listening stations.

5.7.1 Fingerprint representations analysis

The quality of a fingerprint is based on two key attributes: *consistency* and *discriminatory power*. A robust fingerprinting process yields similar values for data originating from the same device (consistency) and produces notably distinct values for data from different entities (discriminatory power) [151]. In our case, linked pairs of fingerprints should exhibit high consistency, resulting in minimal distances, while non-linked pairs should demonstrate significantly larger distances.

Figure 5.7 illustrates the distribution of *distances between two pairs of fingerprints* across all sequence lengths for each representation (vectors, distributions, descriptive statistics, Markov chains, and a combination of all of them). Since two fingerprints generate a vector of distances spanning various orders of magnitude, we normalize the values. A preliminary examination reveals a stark difference in distributions between linked and non-linked pairs: linked pairs generally yield lower distances. Furthermore, distribution-based fingerprints exhibit considerable heterogeneity in distances, with non-linked pairs consistently surpassing their linked counterparts.

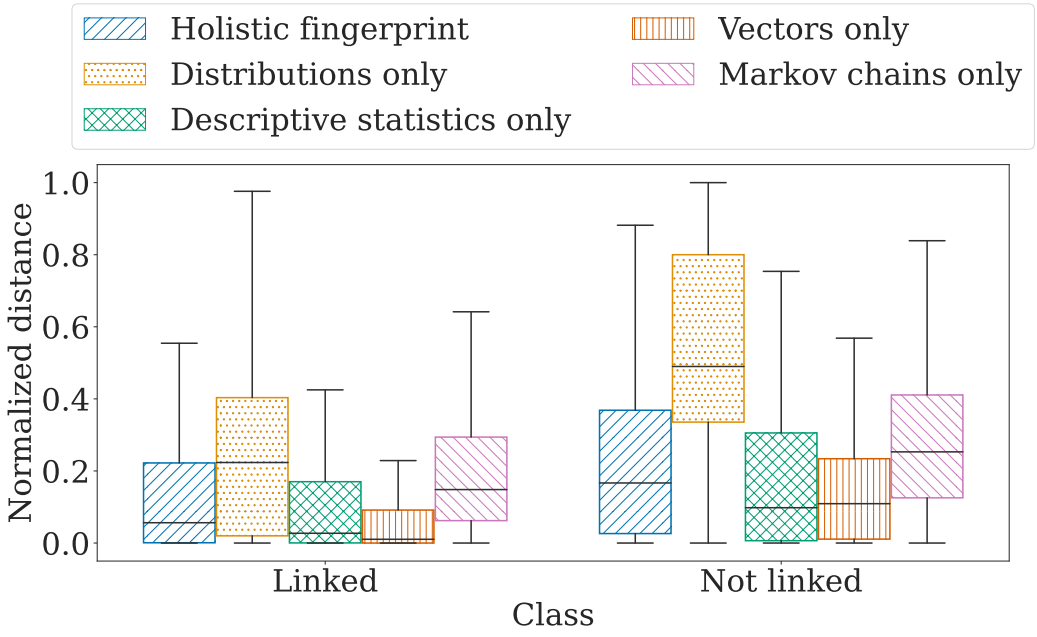


Figure 5.7: Normalized distances between linked and not linked fingerprints pairs w.r.t. fingerprint representation.

Similar patterns are observed when analyzing individual features. As depicted in Figure 5.8, certain features demonstrate superior consistency and discriminatory power compared to others. For instance, the length distribution exhibits discernible differences between linked and non-linked pairs, contrary to the port variance which remains consistent across both categories, hinting at lower relevance during classification.

While these findings suggest decent consistency and discriminatory power for fingerprints, they do not reveal the actual performance variations for each representation during classification. Therefore, we train models using only one fingerprint representation and compare their balanced accuracy.

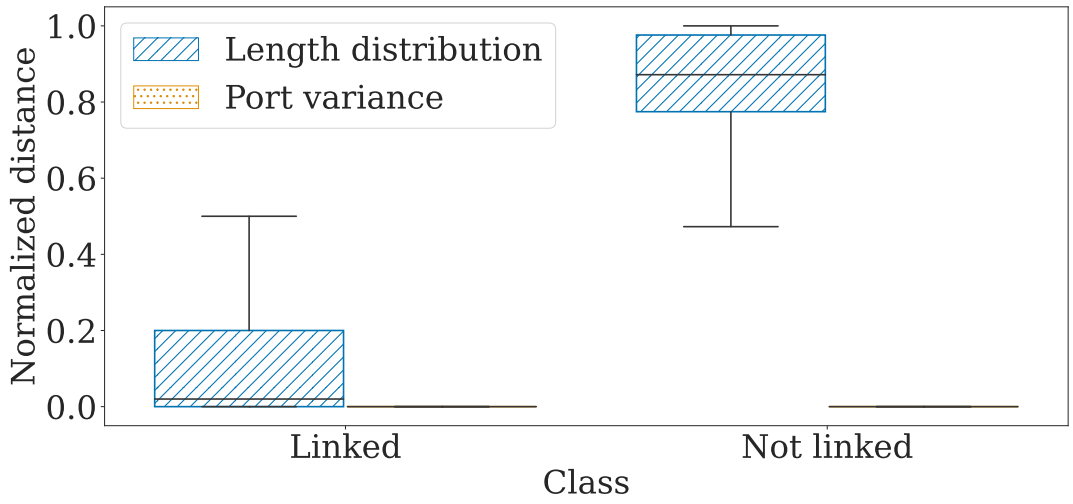


Figure 5.8: Two examples of fingerprint distances between linked and not linked pairs.

Figure 5.9 illustrates that the holistic fingerprint expectedly outperforms other representations from the literature. Remarkably, distributions and descriptive statistics yield similar results for sequences longer than 100 uplink messages, indicating that these simpler representations remain relevant when sufficient data is available.

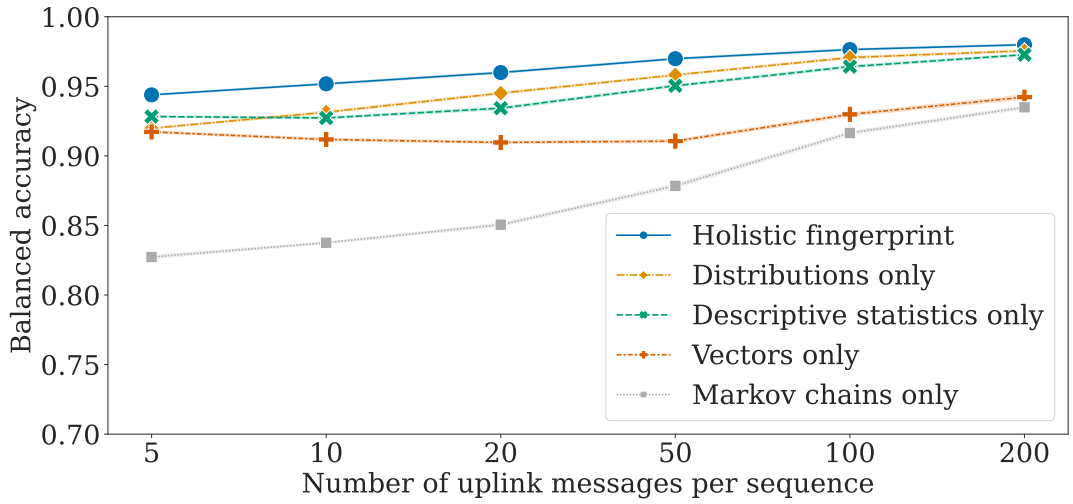


Figure 5.9: Performance w.r.t. fingerprint representations.

Radio-based features are considered stateless and thus not represented as Markov chain. This can explain why the Markov chains representation initially produces a comparatively low balanced accuracy compared to other representations. However, upon removal of Markov chains from the holistic fingerprint, we observe a slight reduction in balanced accuracy. Therefore, we retain it as part of the holistic fingerprint representation despite its lower performance. All other results presented in the following sections are derived from the holistic fingerprint representation.

5.7.2 Measuring the impact of sequence length and feature domains

Once the fingerprint representation is selected, we can consider other parameters, such as the sequence lengths. We consider sequences containing 5, 10, 20, 50, 100, and 200 `uplink` messages (see Section 5.6.2). Figure 5.10 displays the balanced accuracy obtained by models trained using the various lengths, utilizing fingerprints based on *all features*, or limited to one of the three feature domains.

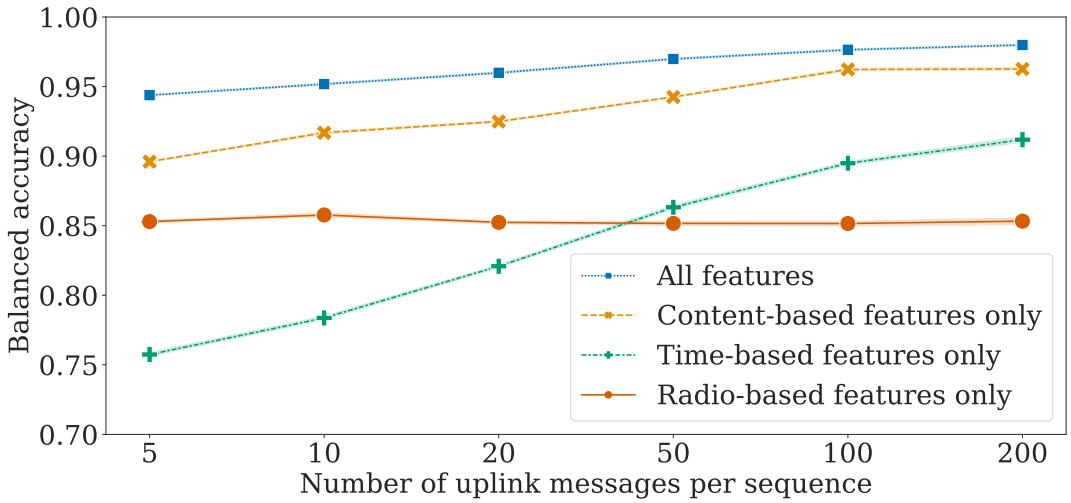


Figure 5.10: Performance w.r.t. feature domains.

In the *all features* setting, the performance remains consistently high across all sequence lengths, with a balanced accuracy ranging from ∼0.95 for sequences of 5 messages to 0.98 for sequences of 200 messages. In other words, even a few messages are sufficient to produce reliable fingerprints. The improvement in performance with longer sequences is expected, as more data points contribute to more accurate fingerprints.

Additionally, figure 5.10 illustrates the performance when using only one domain of features. Content-based features alone achieve the highest performance, starting at a balanced accuracy of 0.90 for sequence length of 5 and increasing to nearly match the performance of the *all features* configuration for longer sequences. On the other hand, performances with time-based features alone start lower at 0.76 and gradually increase but remain below the *all features* setting.

When using only radio-based features, the balanced accuracy starts at 0.8532 for 5-message sequences and is somewhat stable regardless of sequence length, ending at 0.8537 for 200-message sequences. This could be attributed to the selection bias of End-Devices with longer message sequences, where their radio features may be less reliable than other devices.

In defense-oriented scenarios such as network security monitoring, rapid fingerprinting and linkage are imperative. Our sequences consist of a minimum of 5 `uplink` messages. Despite this brief sequence length, LoRaWAN networks generally maintain a low transmission rate. In our dataset, it takes approximately 6 hours, as per the median time, to collect 5 messages from the same End-Device. While this duration might appear significant, the detection process remains fast when considering the numerical aspect.

Moreover, attackers aiming to remain under the radar would likely avoid disrupting existing communication patterns and thus be limited by the current low-throughput constraints. On the other hand, sending rapid successions of messages during the exploitation would result in both a) an easily identifiable disruption of reference patterns, and b) a reduction of the overall time of detection.

5.7.3 Fingerprinting mobile End-Devices

Our fingerprinting approach incorporates features that capture the attributes of the radio channel between the sender and the listening station. However, radio-based fingerprinting may be less effective for mobile end devices, as factors like distance, environment, and obstacles can significantly alter the channel. Determining the mobility of an end device solely based on fluctuations in radio features is impractical. Therefore, we explore a scenario where radio-based features are unavailable.

Figure 5.11 illustrates that for sequences comprising 5 messages, the balanced accuracy decreases from approximately 0.94 to 0.92 (-2.87%), while for sequences of 200 messages, the decrease is minimal (around -0.30%). Despite the slight reduction in performance, our scheme demonstrates applicability in scenarios where radio features are either unavailable or

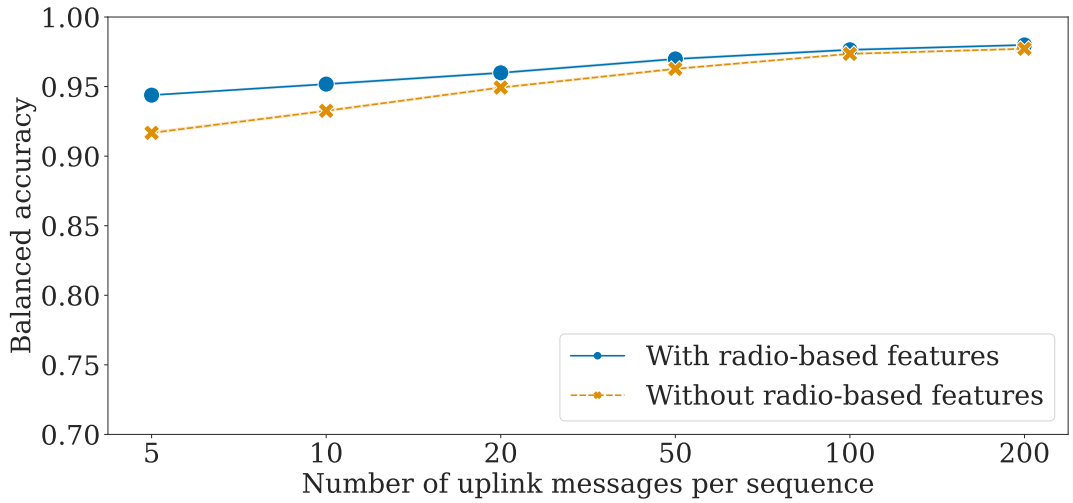


Figure 5.11: Performance w.r.t. radio-based features.

unreliable, particularly relevant for mobile End-Devices.

5.7.4 Impact of the number of controlled listening stations

Radio-based features enhance classification accuracy, particularly for short sequences of messages. Presently, our findings are based on utilizing all available listening stations, approximately 50 in total. Each of these stations contributes a segment of radio-based information, collectively enriching the dataset. Studying scenarios with fewer listening stations offers insights into situations with limited eavesdropper capabilities and coverage, deepening our understanding of how reduced monitoring impacts our approach's effectiveness.

To do so, we analyze nested datasets corresponding to varying numbers of listening stations. The process involves the following steps:

- Filtering the dataset to retain only the information received by n listening stations.
- Training machine learning models using the filtered dataset.
- Reusing the subset containing information from the n listening stations and filtering messages from $n-1$ listening stations.
- Repeating the process iteratively, progressively reducing the number of listening stations until reaching a single receiving station.

Thus, for a set containing all messages received by n listening stations S_n, we obtain $S_n \supseteq S_{n-1} \ldots \supseteq S_{n-(n-1)}$. Hence, fingerprints used for each classification are identical, mimicking the control - or absence thereof - of each receiver.[5]

We repeat the process for 10 unique groups of the top receiving listening stations and opt for $n = 5$ to manage computational complexity. Figure 5.12 illustrates the evolution of performance based on the number of available listening stations. Moving from 1 to 5 listening stations, the balanced accuracy for sequences of 5 messages increases by 9.43%, while for sequences of 200 messages, the improvement is slightly lower at 7.96%.

The number of listening stations directly influences performance, with balanced accuracy increasing as the number of stations rises. However, as sequence lengths increase, the significance of the number of listening stations diminishes. This decline is due to the increasing amount of information from other domains, such as time and content, as the number of messages in a sequence grows.

Nevertheless, results obtained with fewer listening stations remain noteworthy, despite yielding lower performance compared to scenarios with all stations available. Given that we

[5]We note that this property cannot be established by directly segmenting the complete dataset to filter messages received by n, $n-1$, ..., $n-(n-1)$ listening stations. For instance, if a message is received by $n-1$ listening stations, it does not appear in the S_n superset.

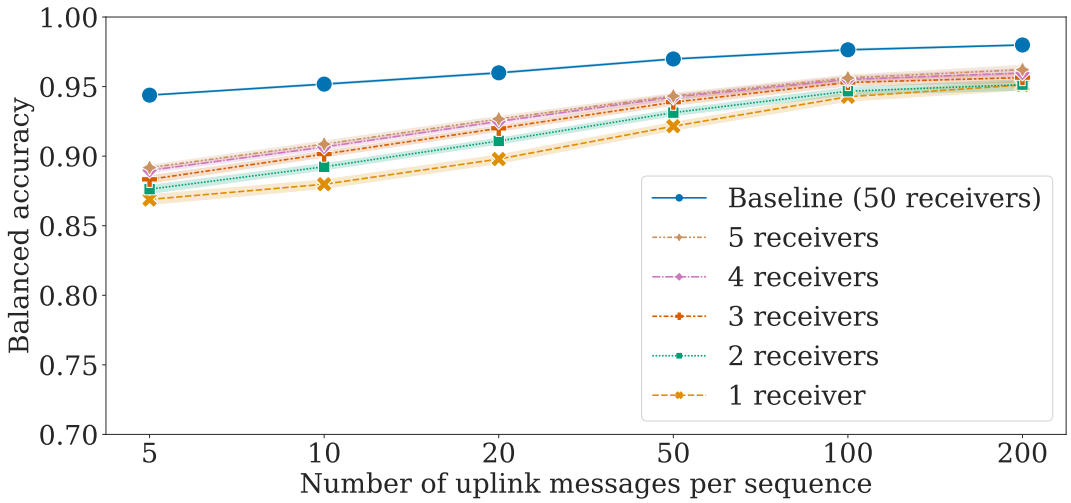


Figure 5.12: Performance w.r.t. the number of controlled listening stations.

did not consider the positions of listening stations, it is conceivable that results could be further enhanced with strategically deployed stations. In practical terms, even with only a handful of listening stations, fingerprinting would remain effective in real-world scenarios

5.8 Limitations and future works

An overview of the study's limitations reveals some considerations to be taken into account.

Dataset Imperfections: Despite our best efforts to curate the dataset, it is not flawless. We may have failed to filter out experimental devices deployed by third-party operators without external notice. However, it remains the most extensive and comprehensive dataset available compared to related works [195, 196].

Ground truth generation: By considering short sequences of messages, we assume that End-Devices have a method for generating new `DevAddr`. This is currently supported by the protocol through a costly rejoin process, which generates exploitable messages and metadata. We do not take these into account and assume the existence of a transparent mechanism, such as the one proposed in Chapter 7. On the other hand, if End-Devices use short-lived `DevAddr`, it becomes difficult to generate a ground truth. We assume the attacker can build this in a closed-world setting before testing on an open-world dataset. Further experiments could be conducted in this regard.

Selection of Inactivity Period: Setting the maximum period of inactivity to $T = 1$ week may not capture *all* End-Devices accurately. Some of them may communicate even less frequently. However, the choice of one week is justified by the rarity of such occurrences and the abundance of available data.

Holistic Fingerprint Applicability: While the holistic fingerprinting approach yields excellent results, its applicability to specific defensive use cases in other wireless protocols (necessitating rapid recognition) remains uncertain. This aspect warrants further investigation beyond the scope of this thesis.

Minimum Sequence Length Constraint: Due to complexity considerations, the study limits sequences to a minimum of five messages. It could be possible to reliably fingerprint End-Devices using even less `uplink` messages, but additional research is required.

Device Similarities and Differentiation: Fingerprinting relies on the premise that different devices produce distinguishable data. However, End-Devices with identical software and configurations may exhibit similarities in content and time-based features. While radio-based features can aid in differentiation, distinguishing between End-Devices deployed in the same location may pose challenges. Further experiments, beyond the scope of the current dataset, are necessary to investigate this highly specific scenario.

5.9 Conclusion

In this study, we demonstrate the feasibility of reliably fingerprinting LoRaWAN End-Devices, effectively linking sequences of messages even if they use different identifiers.

We first introduce the holistic fingerprint, a novel agnostic representation applied to LoRaWAN End-Device communication. Through rigorous comparisons, we establish its superiority over existing formats, demonstrating remarkable classification accuracy. Leveraging this representation, we effectively fingerprint End-Devices using content, time, and radio-based features.

Our experiments reveal the robustness of our approach, reliably identifying origins of sequences as short as 5 messages with a balanced accuracy of 0.95, up to 0.98 based on sequences of 200 messages. Additionally, we explore various simulated scenarios, including mobile End-Devices and eavesdroppers with limited resources. Despite these challenges, our fingerprinting technique remains effective across all scenarios, demonstrating its versatility and resilience.

Beyond offensive applications, our work has broader implications for network security, offering a reliable framework for device identification and classification. By studying LoRaWAN fingerprinting, we pave the way for enhanced network monitoring and security measures.

Part III

Safeguarding Privacy: Countermeasures Against Attacks on LoRaWAN Privacy

Chapter 6

Feature-based countermeasures in LoRaWAN

Contents

In this chapter, we explore ways to counter linkage and fingerprinting attacks (Chapters 4 and 5) based on machine learning models. By studying each feature and selecting or designing specific mitigations w.r.t. LoRaWAN's constraints, we apply countermeasures to the dataset and observe the protection gained via the loss in attack efficiency. When combining all mitigations at once, we reduce linking attacks by 12.5% and device fingerprinting by 7.32% in realistic settings. We complete our analysis by estimating the inherent overhead of such countermeasures, for instance the additional bytes transmitted due to padding.

This chapter is partially based on our work *Device Re-identification in LoRaWAN through Messages Linkage* [166] (peer-reviewed and published) and *Introducing Multi-Domain Fingerprints for Lorawan Device Linkage* [167] (currently under review). The present chapter differs mainly by the following points:

- We enhance justifications and descriptions of possible countermeasure deployments in LoRaWAN.
- We introduce a novel evaluation of mitigations, measuring its impact on the linking attack for the first time, and complete previously reported results on fingerprinting.
- We discuss the effects of countermeasures on End-Devices and offer comprehensive assessments of the associated overhead. Additionally, we outline potential research directions for addressing challenges that remain beyond our current scope.

6.1 Introduction

The privacy attacks discussed in previous chapters rely on robust machine learning models that utilize diverse features, many of which overlap (e.g. radio-based features are all derived from the radio power). Given the effectiveness of these attacks, there is a pressing need for further investigation into potential countermeasures to safeguard End-Devices and their users effectively. As explored in Section 2.4, numerous mitigations for wireless networks have been proposed for various contexts, including dummy traffic [15, 27, 112, 131, 153, 229], padding [22, 76, 145, 171, 186], random delay between messages [33, 139, 173, 197, 231], or modifying the radio medium [20, 23, 198].

However, such countermeasures have seldom been studied in the specific context of LoRaWAN, nor in a similar scale. In this chapter, our primary focus lies in formulating viable mitigations for each of these features, and assessing their feasibility within the framework of LoRaWAN. Through a methodical approach, we first study the potential deployment scenarios of these mitigations and their efficacy in thwarting attacks. Then, we evaluate their accuracy in isolation, before examining their combined effects under diverse settings. Finally, we measure how these mitigations impact End-Devices by inherently producing overhead, both in computation and communication.

6.2 Feature-based countermeasures

Inspired by related works on mitigations presented in Section 2.4, we explore how *each feature* previously exploited can be mitigated through tailored countermeasure. A summary of each of them and their corresponding mitigations is available in Table 6.1.

An alternative approach to introducing noise to break communication patterns is generating dummy traffic. As outlined in Section 2.4.2, developing a convincing solution is complex and induces significant overhead [131]. In LoRaWAN, transmission consumes more energy than other operations and should be minimized [43]. Hence, we exclude dummy traffic from possible countermeasures, focusing on noise-based perturbations.

Table 6.1: Summary of studied countermeasures.

Features	Countermeasure	Specified by	Supported by LoRaWAN standard
`FCnt` & `FPort`	Encryption	Standard	No
Payload length	Padding	Application	Yes
Time-based	Delay	Application	Yes
Radio-based	Transmit power modulation	Application	Yes

Additionally, we focus on *software-based countermeasures* due to easier large-scale implementation: they do not require additional hardware, and could be deployed via a firmware update. However, this approach inherently comes with a drawback: some features are by design nearly impossible to influence. For instance, the time difference of arrival of the same message at each gateway used in Chapter 4 is inherently tied to the physics of wave propagations. Similarly, the last time a `DevAddr` has been observed is difficult to hide in isolation. As we will see in Chapter 7, more global alternatives exist.

6.2.1 Content-based features

Attacks presented in Chapters 4 and 5 extensively leverage content-based features. We start by studying the header fields, including `FCnt` and `FPort`, and then discuss how noise can be added to the payload length.

6.2.1.1 FCnt & Fport

Both `FCnt` and `FPort` are relevant in privacy attacks against LoRaWAN. As seen in Section 4.5.2, the `FCnt` is notably utilized to easily link the `join-request` with following `uplink` messages. While the `FPort` is less prevalent, its patterns are also utilized for fingerprinting

(see Section 5.5.1). In practice, the server relies on precise values from these fields, rendering noise-based countermeasures ineffective. Therefore, we opt to encrypt both `FCnt` and `FPort` using existing cryptographic primitives.

Encrypting the `FCnt`: The `FCnt` is encrypted using the NwkSEncKey, shared with the NS, along other already obfuscated network-specific information including MAC commands and `FOpts`.

Encrypting the `FPort`: The `FPort` is encrypted using the NwkSEncKey and decrypted by the NS, before relaying the clear-text value to the AS. The `FPort` is leveraged by both the NS and AS to correctly route traffic to relevant applications. For instance, when the `FPort` is set to 0, it indicates that the payload contains encrypted MAC commands destined for the NS, while values ranging below 223 are application-specific. This solution necessitates trust in the NS, but it represents an improvement over the current situation.

With both fields encrypted, their corresponding information are now unavailable to a passive eavesdropper. In practice, they are removed from datasets used in the machine learning process.

6.2.1.2 Payload length

In both attacks, we heavily rely on payload length as a distinguishing feature, making padding an intuitive method for obfuscation. Such an approach necessitates two building blocks: a) a way to distinguish between actual data and padding, b) enough available space to pad.

First, this can be achieved by employing a designated byte as a separator or by prefixing the payload with the length of the useful portion [144] (see Figure 6.1). As padding is introduced along the payload, this process can be left to the application and does not affect the LoRaWAN protocol itself.

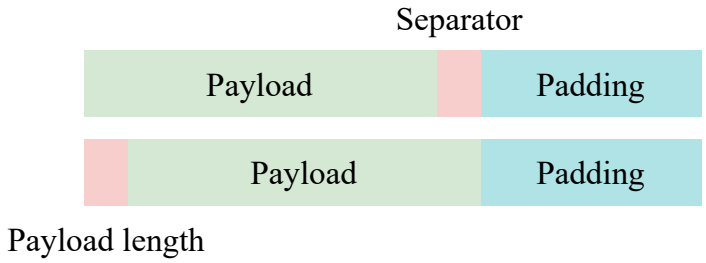


Figure 6.1: Padding implementation strategies.

Second, the amount of padding is limited by the maximum payload length. In LoRaWAN, it depends on the DataRate (DR) (from 51 to 222 bytes, see Table 3.1). Figure 6.2 presents the observed payload sizes for `uplink` messages in our real-world dataset already presented in Sections 4.3 and 5.3. We note that 99.82% of the 71 million `uplink` messages captured from June 2020 to August 2023 contain a payload shorter than 51 bytes, allowing for padding even in the most conservative scenarios where DR 0 to 2 are used (max. 51 bytes).

Given a message of payload size ℓ and a maximum size L, the padding amplitude p_A is calculated as $L - \ell$. Based on these parameters, we explore two padding strategies, illustrated by Figure 6.3.

Max-padding: We fill the payload to its maximum size, ensuring that all `uplink` messages have a consistent length of L after padding. Messages that are already at the maximum size remain unaffected by this process.

Uniform padding: We randomly select a length of padding from a uniform distribution over the range $[0, p_A]$. This method ensures that messages with a length $\ell < L$ are padded to a length greater than or equal to their initial size, while messages already at the maximum size

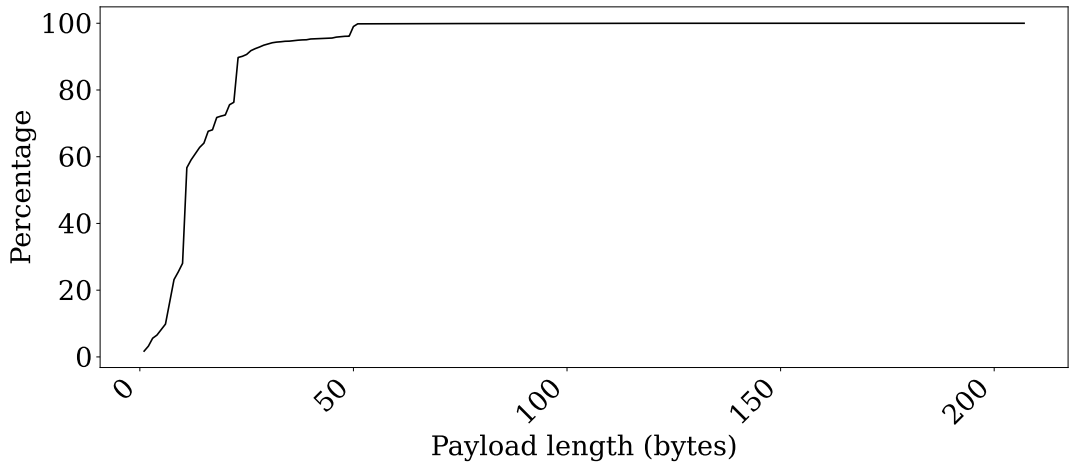


Figure 6.2: Cumulative distribution function of payload lengths for `uplink` messages.

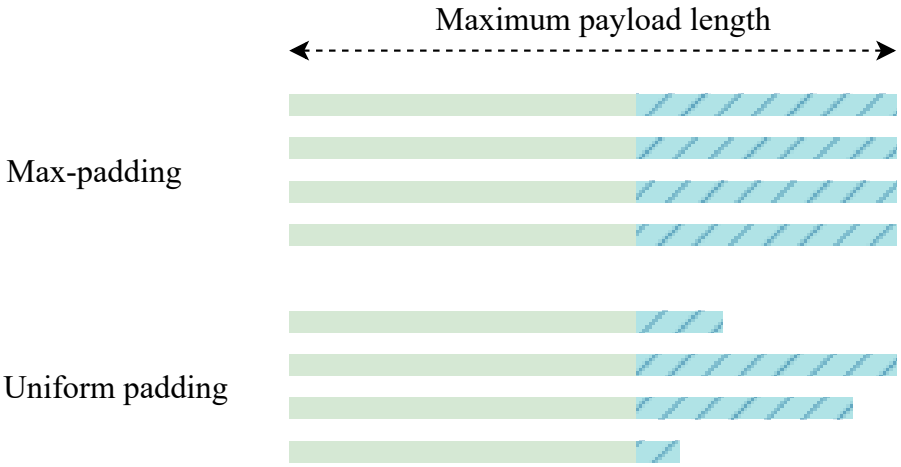


Figure 6.3: Padding strategies, with original payload on the left and padding (hatched) on the right.

remain unchanged. As discussed in Section 2.4.3.1, applying this padding method provides guarantees of Approximate Differential Privacy, a relaxed version of Differential privacy [75]. In practice, we arbitrarily select $p_A \in 5, 10, 40, 80, 200$, corresponding to an Approximate Differential Privacy of parameters $(0, \frac{L}{p_A})$ [75].

6.2.2 Time-based features

Time-based features have proven to be highly valuable in previous attacks. A viable strategy illustrated in Figure 6.4 involves introducing random delays in message transmission [173, 197, 231]. Such measures disrupt the regularity of message timing, making it more challenging for attackers to discern patterns and derive meaningful information. Both the time difference between `join-request` and `uplink` (Chapter 4) and Inter-Arrival Time between `uplink` messages (Chapter 5) are impacted by random delays. They can be deployed at the application level, directly implemented by the End-Device.

Disrupting communications by adding delay may not be possible with time-sensitive applications. For instance, alerts when temperature exceeds a specific threshold should be sent as soon as possible. We choose to ignore this issue and present best-case scenarios where all messages can be delayed, while proposing realistic values of said delay.

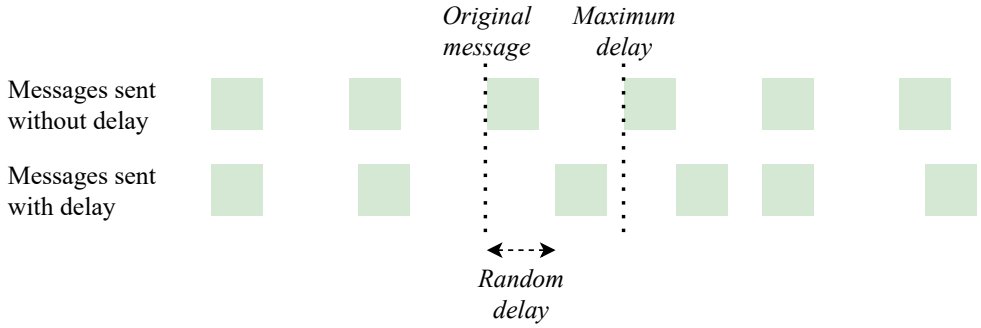


Figure 6.4: Introducing random delay between messages.

In practice, we draw random values from an interval $[0..\Delta]$.[1] To counter linking `join-request` and following `uplink` messages, we select $\Delta \in \{30, 60, 300, 600, 1800, 3600\}$ seconds (from 30 seconds to 1 hours). As the time between consecutive `uplink` messages is larger than between `join-request` and following `uplink` messages, we increase the possible delay for fingerprinting: $\Delta \in \{60, 300, 600, 900, 1200, 1800, 2400, 3600, 7200, 14400\}$ seconds (from 1 minute to 4 hours).

6.2.3 Radio-based features

Radio-based features provide a robust foundation for constructing our attacks, as demonstrated in Section 5.7.2. Even if other features are protected, they inherently contain sufficient information to fingerprint devices. In our approach, we concentrate on passive countermeasures, refraining from jamming eavesdroppers and deploying directional antennas [23], or physical shielding [198]. Instead, we choose a strategy introducing noise through dynamic adjustments in transmission power to minimize the distinguishability of signals between messages.

Radio-based indicators (RSSI, SNR, and ESP) are all influenced by the transmit power. According to the Friis transmission equation [85], the power at the transmitting and receiving antennas (resp. P_t and P_r) are linked to the respective gain of those antennas (resp. G_t and G_r), their distance R, and the constant λ:

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi R}\right)^2$$

Thus, modifying the transmit power P_t results in a proportional adjustment of P_r, upon which both the RSSI and SNR linearly depend (and consequently, the ESP).

In LoRa modules, the transmit power can be adjusted from -4dBm to 20dBm, with 1dBm steps [48]. We implement this countermeasure by introducing a random and uniformly distributed offset to the power signal for each record, and we adjust the RSSI, SNR, and ESP accordingly. It is assumed that End-Devices operate at their base power level, enabling adjustments up to 20dBm or down to -4dBm. This approach is considered more representative of typical deployment scenarios. Hence, we vary the size of the intervals from which the power offsets are drawn: $[0; 5]$, $[0; 10]$, $[0; 15]$, $[0; 20]$, and $[-4; 20]$.

Because of the lack of information on actual deployments, we do not account for the potential effects on frame transmission success or failure, such as simulating message loss or reception by gateways that were previously out of range.

[1] We do not consider negative delay (i.e. sending a report *before* it should have been transmitted), as it requires knowing exactly which type of application is running.

6.3 Evaluation methodology

We follow the same methodology presented in Sections 4.4.3 and 5.6. However, we apply countermeasures during the process, training machine learning models on noisy data.

As shown in Figure 6.5, we first implement countermeasures in isolation, meaning only one feature is impacted. As other features may compensate and smooth results when updating mitigations parameters (e.g. the amount of padding), we train and evaluate models with and without having access to features left intact. Finally, we *combine* multiple countermeasures to create two scenarios, one of moderate and another of aggressive impact, based on their expected disruption of LoRaWAN applications and energy consumption. In doing so, we try to offer more realistic settings to appreciate the impact of all mitigations if they were actually deployed.

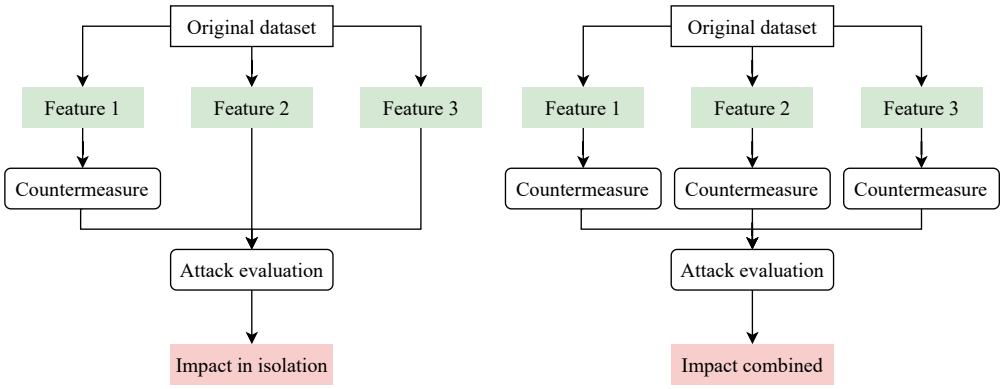


Figure 6.5: Countermeasures applied in isolation (left) and combined (right).

For the linkage attack (Chapter 4), we intentionally keep the Gateway Time Distance, the DevAddr Difference, Receiving Gateways Distance, Spreading Factor, and DataRate, as no countermeasure presented here influence them. When targeting the fingerprinting attack (Chapter 5), we select a sequence length equal to 50 `uplink` messages. This offers a good compromise between attack performance, realistic number of available sequences to an eavesdropper, and comparatively quick results generation. For combined countermeasures, we study the impact evolution on all sequence lengths.

In both cases, evaluations are done using all listening stations, and are repeated 15 times with different random seeds to smooth results, showcased using the Balanced Accuracy (BA).

6.4 Experimental results

In this section, we present results obtained when implementing countermeasures various scenarios: in isolation with or without other features, and combined. We conclude by examining the inherent overheads associated with these solutions.

6.4.1 Applying countermeasures in isolation

By first applying mitigations in isolation, we are able to detect trends and confirm that increasing the amplitude of noise parameters (e.g. the amount of bytes used in padding) decreases the performance of attacks. To demonstrate this, we focus on two examples, one for each attack: padding against fingerprinting and delay against linking. Finally, we show that isolated features do not actually protect LoRaWAN from our attacks and should be applied in combination to thwart attacks. In all cases, we compare models against a baseline, obtained without any countermeasure applied.

6.4.1.1 Reducing fingerprintability through padding

Figure 6.6 illustrates the impact of introducing padding on End-Devices fingerprinting, while keeping other features impact. The max-padding strategy yields the best results, decreasing

the balanced accuracy of the attack from ∼0.97 to 0.92 (-4.7%) in all cases. Uniform-padding provides less protection, producing a -2.92% decrease for a padding amplitude of 40 bytes. In both cases, the decrease in balanced accuracy is ultimately limited, with a small progression for uniform-padding when increasing the padding amplitude.

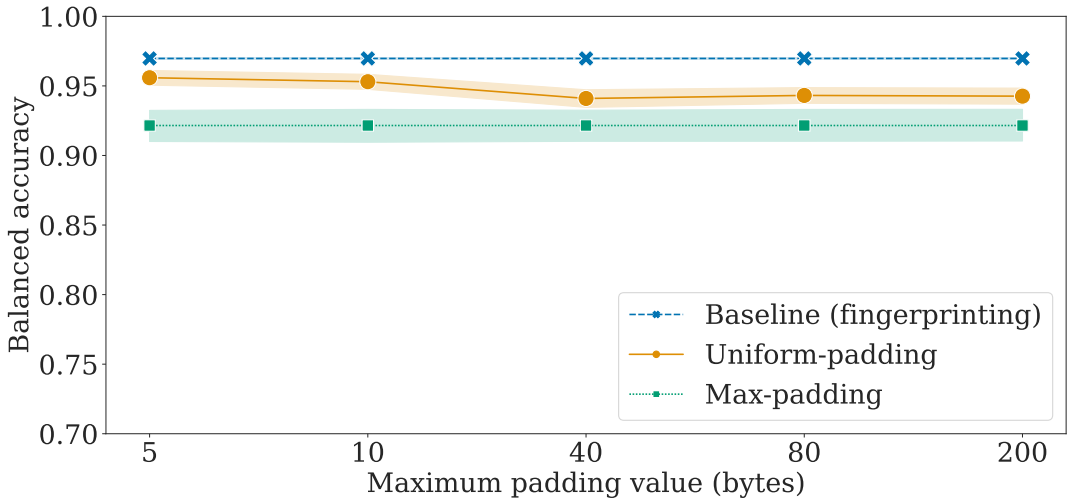


Figure 6.6: Impact of the padding on fingerprinting performances.

As seen in figure 6.2, many `uplink` messages are 11 bytes long, with 78% of the dataset transmitted using DR 0 - 2, allowing for 51-byte payloads at most. Such configuration can explain the plateau of performance from 40 bytes and onwards, as all available bytes are already padded.

Based on these results, padding alone does not significantly reduce the fingerprintability of LoRaWAN devices, with max-padding showing a slightly higher impact.

6.4.1.2 Thwarting linkage attack via random delay

Figure 6.7 shows the impact of adding random delay between `join-request` and `uplink`, both with and without keeping other features. For instance, the balanced accuracy when using only the delay-impacted features (i.e time-based features) starts at ∼0.67 with 30 seconds delay, decreasing to ∼0.51 with 1-hour delays. Such a trend is not visible when other features remain available to the machine learning model.

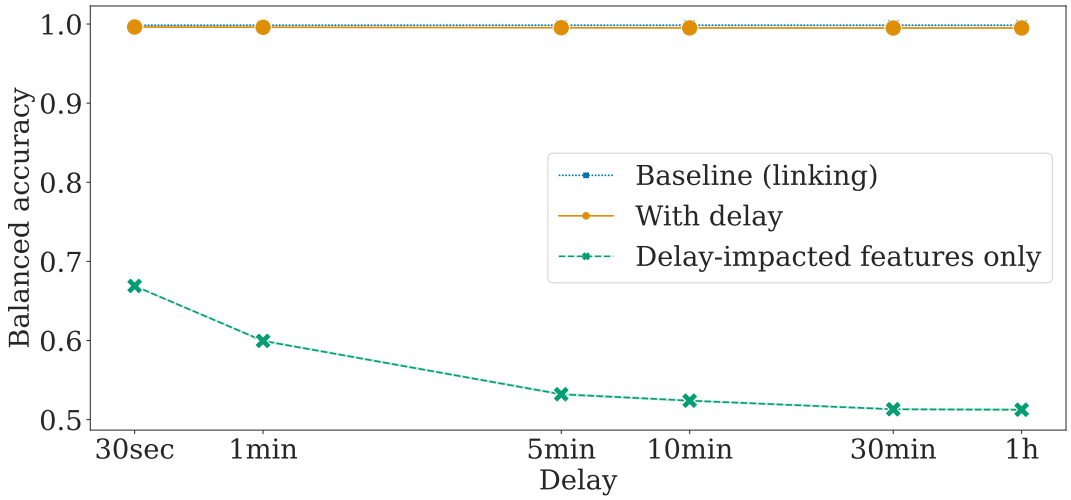


Figure 6.7: Impact of the delay on linking performances.

To summarize, adding a small delay effectively reduces the accuracy of the attack when

looking at time-based features only. However, it is useless when taking into account other features.

6.4.1.3 Limits of isolated countermeasures

Table 6.2 showcases the impact of mitigations used in isolation, i.e. when all other unaffected features are still available to machine learning models. As hinted at by previous results, applying countermeasures individually shows limited impact on attacks' performance. At best, uniformly padding messages decreases the efficiency of the fingerprinting attack by 2.92%.

Table 6.2: Impact of isolated countermeasures compared to the baseline of both attacks.

Countermeasure	**Values**	**Linking (%)**	**Fingerprinting (%)**
Uniform-padding	10 bytes	0.01	-1.61
	40 bytes	0.01	-2.92
Delay	10 minutes	-0.26	-0.37
	60 minutes	-0.34	-0.28
Power modulation	$[0; 10]$	0.0	-0.57
	$[-4; 20]$	0.0	-0.74
Encrypted `FPort`		-0.0	-1.78
Encrypted `FCnt`		-0.15	

These small values can be attributed to models adjusting for the presence of countermeasures by selecting alternative features. For example, as previously demonstrated in Section 4.5.2, the linking attack heavily rely on the `FCnt` alone to achieve a high balanced accuracy, which is still available in all but the "Encrypted `FCnt` " case.

Additionally, as noise amplifies within certain features, machine learning models can more easily discard outliers and concentrate on meaningful data. This pattern is observed in fingerprinting, where increased perturbation of time-based features via random delay actually improves the attack accuracy instead of decreasing it.

6.4.2 Combining countermeasures

To enhance our assessment, we measure the collective impact of all countermeasures by observing the reduction in balanced accuracy compared to the baseline, where no mitigations are applied. We study two perturbation settings based on varying levels of perturbations. In the *moderate* perturbation scenario, countermeasures exert a moderate influence on communication, including 10 bytes of uniform-padding amplitude, a transmit power modulation ranging from 0 to 10 dBm, and a random delay spanning 10 minutes.

In the *aggressive* perturbation scenario, countermeasures could significantly impair communication performance, with 40 bytes of uniform padding amplitude, a transmit power modulation spanning -4 to 20 dBm, and a random delay extending to 1 hour. Although longer delays are theoretically feasible, we limit the maximum duration to realistic values.

In both scenarios, we encrypt both `FCnt` and `FPort`, and select uniform-padding to avoid excessive overheads (see Section 6.4.3).

6.4.2.1 Impact of combined countermeasures on linking attacks

Table 6.3 showcases the impact of combining all mitigations against the linking attack. The balanced accuracy is decreased by $\sim$12.5% with *moderate* settings, and 15.7% with *aggressive* countermeasures.

While those values may seem underwhelming, we highlight that it is currently impossible to completely thwart the attack while preserving energy. First, some information remains in features disrupted via noise, such as payload length, time difference between `join-request` and `uplink` message, or radio characteristics. Second, several features are not influenced by

Table 6.3: Impact of combined countermeasures on linking performance.

Settings	Balanced accuracy
Baseline	0.9984
Moderate	0.8740 (-12.5%)
Aggressive	0.8414 (-15.7%)

our approach, including the last time a `DevAddr` has been seen, or the list of gateways that received both `join-request` and `uplink` messages.

6.4.2.2 Impact of combined countermeasures on End-Devices fingerprinting

Figure 6.8 illustrates the impact of combining all countermeasures against the fingerprinting attack. For 50 messages per sequence, the balanced accuracy decreases by roughly -2.24% with moderate mitigations, and by -5.10% with aggressive settings. When decreasing the number of messages per sequence to 5, the balanced accuracy also decreases: -7.32% with moderate settings, and -10.22% with aggressive countermeasures.

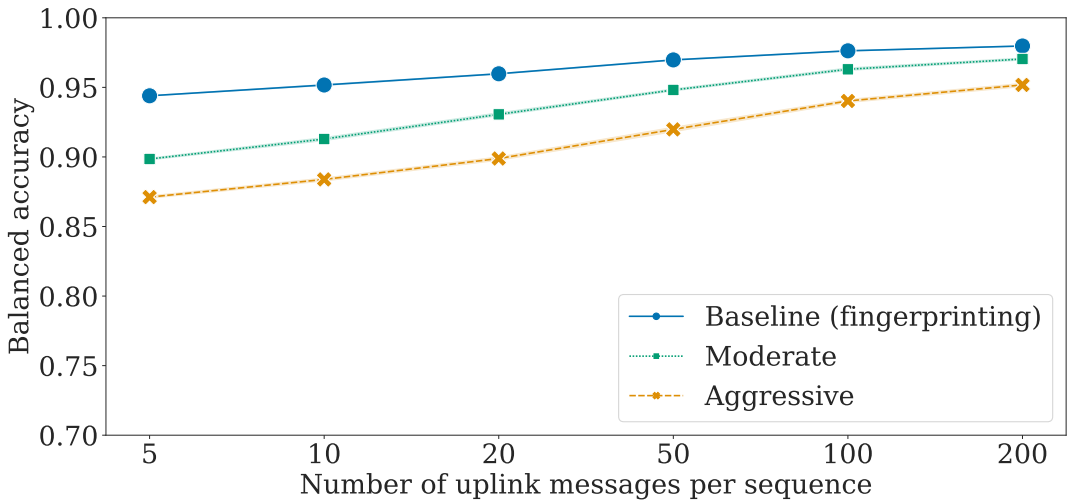


Figure 6.8: Impact of combined countermeasures on fingerprinting performance.

Additionally, we train models using a limited number of listening stations with a similar methodology as in Section 5.7.4. We observe that diminishing the number of listening stations amplifies the effectiveness of countermeasures. For example, implementing transmit power modulation across sequences of 50 messages results in a performance reduction of -0.03% across all listening stations, yet it escalates to -2.50% when only three listening stations are present under moderate settings. This suggests that countermeasures could potentially be more impactful in real-world scenarios, where eavesdroppers are constrained to a limited number of listening stations.

In summary, aggressive mitigations expectedly show greater impact on the attack's performance compared to moderate settings. However, all things considered, fingerprinting is seldom affected, even when all countermeasures are combined. Moreover, their impact diminishes as the number of `uplink` messages per sequence grows: models are able to efficiently classify End-Devices despite aggressive settings, thanks to data aggregations corresponding to a single identity. Even more aggressive methods would be required to further thwart the attack, which may not be acceptable from the point of view of LoRaWAN applications.

6.4.3 Countermeasures overhead and impact on communications

In this section, we explore overheads and possible impacts on communication generated by the various mitigations presented previously. However, not all countermeasures induce an easily measurable impact. For instance, while delay potentially disrupt traffic, it is hard to measure

its exact impact without knowing the requirements of underlying applications. Hence, we focus on encryption of `FCnt` and `FPort`, padding, and transmit power modulation.

6.4.3.1 The limited impact of encrypting `FCnt` and `FPort`

Encrypting the `FCnt` and `FPort` could induce both communication and computation overheads.

First, by encrypting these fields via AES-128 CCM*, we preserve their length.[2] Hence, this process does not introduce additional bytes on air: there is no communication overhead.

Second, AES-128 deals with 16-byte data blocks, which means that any input shorter than this threshold requires 1 encryption *block*. For instance, a 4-byte MAC command requires 1 block, and adding a 2-byte `FCnt` and 1-byte `FPort`, summing up to 7 bytes, would *still* require 1 encryption operation.

To determine if additional AES-128 blocks are required by the End-Device, we monitor the length of MAC commands encrypted by the NS, either in `FOpts` or in the actual payload with `FPort` equal to 0.

We observe $\sim$ 14 million `uplink` messages containing a command over the whole dataset. While encrypting `FCnt` and `FPort` increases the number of encrypted *bytes* by $\sim$64%, it only corresponds to a 4% increase in encrypted AES *blocks*. Hence, the minimal overhead of encrypting both fields makes it a compelling solution, especially to thwart fingerprinting.

6.4.3.2 Increasing on-air bytes via padding

Introducing additional data and expanding the payload size inevitably implies an overhead, as lengthier messages consume more energy and increase the occupation of the shared medium. In LoRaWAN, those overheads are characterized by the Time On Air (TOA), the time required for an End-Device to send data to the receiver. The TOA does not linearly depend on the length but can be derived from parameters related to the DataRate [127]. As communications are particularly energy-consuming [43], the TOA is directly linked to battery depletion.

Figure 6.9 displays the evolution of the TOA for each padding strategy, compared to the baseline. Unsurprisingly, the max-padding strategy produces the highest overhead with a $\sim$96% increase in TOA. On the other hand, uniform-padding varies between $\sim$5% and 46% overhead for padding amplitudes of 5 and 200 bytes respectively.

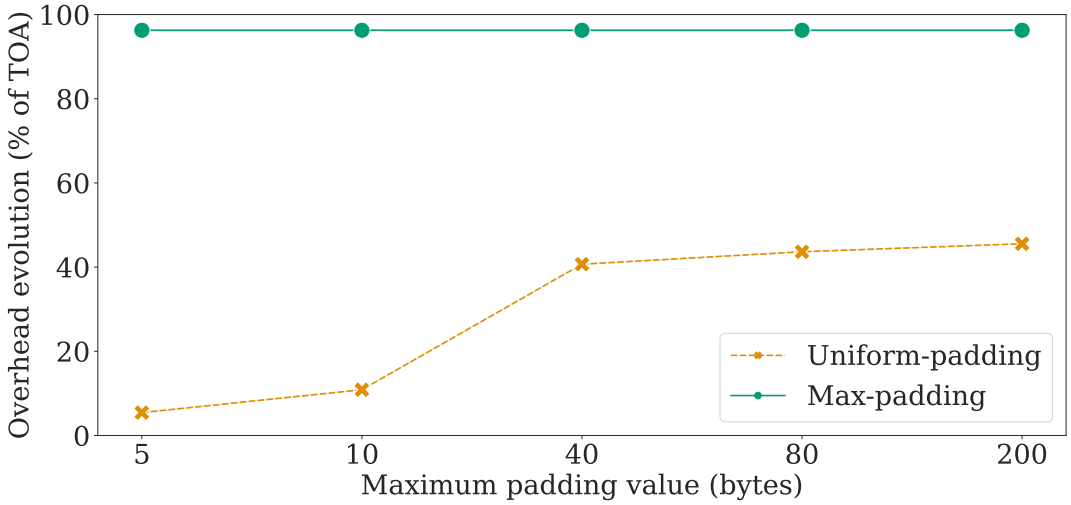


Figure 6.9: Impact of padding on Time On Air.

Unless a privacy threat is deemed severe by a network administrator, the adoption of maximum padding seems impractical due to its substantial energy consumption. Although less demanding, the utilization of uniform padding still results in a noteworthy increase in TOA. Selecting a low number of bytes for padding is mandatory.

[2] CCM is derived from CTR (counter mode), an AES implementation allowing for encrypted outputs of equal size as clear-text inputs.

6.4.3.3 Complex transmit power modulation overheads and impacts

Estimating the impact of power modulation poses several challenges. First, employing power modulation with negative values introduces increased risks of loosing messages. As actual conditions during communication are unknown, logs can not be leveraged to accurately estimate if an `uplink` message would be lost when reducing the transmit power.

Second, increasing transmit power increases energy consumption, depleting the battery of End-Devices. Despite some works on LoRaWAN radio parameters and corresponding energy consumption [48], further investigation and real-world deployments are necessary to accurately assess both potential losses and increased energy consumption.

6.5 Conclusion

Our exploration into implementing feature-based countermeasures to mitigate privacy attacks has revealed both opportunities and challenges. While it is feasible to introduce mitigations, our findings highlight the importance of combined approaches. Merely applying isolated countermeasures without considering the broader context proves ineffective, as demonstrated by the adaptability of machine learning models. Nevertheless, our experiments demonstrate that combining multiple mitigations can yield improved results. Despite this, even aggressive countermeasures induce at most $\sim$ 15% of decrease in balanced accuracy. Moreover, the adoption of these mitigations inherently introduces overheads, which poses challenges for resource-constrained end-devices.

In light of these limitations, alternative strategies warrant investigation, such as reducing the number of `uplink` messages associated with a single identity. Pseudonym-based technologies offer promising avenues for further research and development in enhancing the security and privacy of LoRaWAN networks.

Chapter 7

Privacy-preserving pseudonyms for LoRaWAN

Contents

In this chapter, we design rotating pseudonyms tailored to the specific constraints of LoRaWAN. We first analyze the limits of existing schemes, including alternatives already supported by the protocol's standard and certificate-based approaches implemented in VANETs. We then study how resolvable pseudonyms implemented in BLE, and sequential pseudonyms designed for Wi-Fi, can be adapted to LoRaWAN, evaluating them based on various desired properties. We reinforce our proposition with theoretical analyses and simulations on a large-scale dataset, discussing limitations of each solution, and we find that sequential pseudonyms match the requirements. Finally, we explore potential generalization to other LPWAN protocols.

This chapter is partially based on our peer-reviewed and published work *Privacy-Preserving Pseudonyms for LoRaWAN* [165]. The present chapter differs mainly by the following points:

- We complete the initial analysis with additional pseudonyms solutions, such as HASHA [231], certificate-based [138, 169], range-based [152], and encrypted link-layer pseudonyms [30].
- We thoroughly discuss possible renewal strategies for pseudonyms in the specific context of LoRaWAN.
- We enhance the evaluation of resolvable and sequential schemes, providing additional insights on observed behaviors.
- We extend our analysis to other LPWAN protocols, demonstrating that it is possible to adapt pseudonyms to enhance privacy across similar ecosystems.
- We highlight the limitations of the selected solution and discuss corresponding leads for future works.

The content of the theoretical analysis in Section 7.6.4.1 has been jointly authored with Jan Aalmoes and is further developed in [13].

7.1 Introduction

Clear-text and stable identifiers allow eavesdroppers to track devices across time and space [89]. For instance, fingerprinting attacks discussed in Chapter 5, along with other activity inference threats [131], rely on aggregating numerous messages behind a single address. As seen in Section 2.4.5, a viable countermeasure is to deploy temporary, unlinkable, and regularly updated values called *pseudonyms*. For example, introducing such a mechanism would reduce the number of messages per sequence available as ground truth for the machine learning model, effectively thwarting the fingerprinting attack presented in Chapter 5.

Existing solutions deployed in technologies such as BLE [92] or VANETs [138, 169] were not specifically designed with strict energy usage optimization in mind. As outlined in Section 3.1.7, LoRaWAN operates under distinct constraints, including the need for optimized energy consumption, low communication throughput, and limited computational power. Given these differences, further research is needed to assess the feasibility of implementing pseudonyms in the context of LoRaWAN. In this section, we explore how the `DevAddr`, crucial for network access and communication, can be replaced by a *rotating pseudonym* to minimize the number of messages associated with a single identity, while preserving its functionality.

According to Petit et al., *a pseudonym [...] should be useable for authentication, but must not contain any personal identifiable information that could link to the pseudonym holder's real identity* [169]. First, we extend their definition by proposing multiple properties tailored for the constraints of the LoRaWAN protocol. Then, we explore existing solutions in light of those constraints, and select two contenders to evaluate, through both theoretical analysis and simulations. Finally, we generalize our work to other LPWAN technologies and discuss security-related issues.

7.2 Properties

We define several properties, stemming from our threat model based on a passive eavesdropper lacking knowledge of the cryptographic keys shared between devices and servers (see Section 2.3.1). More precisely, the attacker aims to compile sets of `uplink` messages corresponding to a specific device to track and/or fingerprint it. Currently, linking messages is trivially achieved through the `DevAddr`. Our goal is to thwart such an approach while respecting the constraints of the LoRaWAN protocol (see Section 3.1.7)

A suitable privacy-preserving pseudonym scheme for LoRaWAN should have the following properties:

- $\mathcal{P}_1$**: Unlinkability:** Two messages originating from the same device should not be linked using the pseudonym (previously `DevAddr`) and other frame fields available in clear (e.g. `FCnt`).
- $\mathcal{P}_2$**: Minimal communication overhead:** Because of the strict duty cycle and energy constraints of End-Devices, the communication overhead has to be kept minimal [54].
- $\mathcal{P}_3$**: Low computation/memory overhead:** The computation and memory overhead has to be minimal, in particular for constrained End-Devices [54].
- $\mathcal{P}_4$**: Reliability:** Receivers should be able to recover the identity of End-Devices without the scheme introducing additional message losses.
- $\mathcal{P}_5$**: Legacy support:** To facilitate its co-existence with current specifications and progressive adoption, the scheme should require only limited modifications of the protocol. In particular, the general structure of the frame should be preserved to facilitate adapting parsers and to ensure a progressive adoption. The scheme should only use cryptographic primitives available in the specifications of LoRaWAN (e.g. AES).

In practice, $\mathcal{P}_2$ (Minimal communication overhead) is more important than $\mathcal{P}_3$ (Low computation/memory overhead), as LoRaWAN communications incur significant energy costs. Indeed, a complete transmission (a few millijoules) consumes $\sim 10^9$ times more than encryption (only picojoules), for the same payload length [43, 207].

Additionally, $\mathcal{P}_3$ (Low computation/memory overhead) should take both End-Devices and servers into account: the later can currently handle thousands of concurrent devices. While it is easier to scale up the deployment of grid-connected machines than constrained IoT devices, taking into account complexities while receiving `uplink` messages is important to avoid resource exhaustion.

7.3 Limits of existing pseudonym schemes

In this section, we explore existing pseudonym schemes in light of LoRaWAN constraints and expected properties defined in Section 7.2. On top of two legacy solutions, already available in the LoRaWAN standard, we discuss approaches inspired by related works, detailed in Section 2.4.5.

7.3.1 Legacy schemes

The LoRaWAN standard currently supports two ways of implementing rotating pseudonyms: the rejoin process, and multiple End-Devices sharing the same `DevAddr`.

7.3.1.1 Rejoin process

In LoRaWAN, the `DevAddr` is assigned by the NS to an End-Device during the join process (see Section 3.1.6.2). At any time, a rejoin process can be triggered, thereby renewing the `DevAddr` to another randomized value, effectively acting as rotating pseudonym. A rejoin process can be initiated by both the End-Device via a ReJoin-request message, or the NS via a ForceRejoinReq downlink MAC command. This approach implies a costly communication overhead of at least one ReJoin request and one Join Accept. Moreover, as demonstrated in Chapter 4, an eavesdropper can easily leverage the generated traffic of the join process to link back the `rejoin-request` and following `uplink` messages.

As this solution is weak w.r.t. $\mathcal{P}_1$ (Unlinkability) and $\mathcal{P}_2$ (Minimal communication overhead), *we exclude it from further comparisons.*

7.3.1.2 Shared `DevAddr`

In LoRaWAN, multiple End-Devices can be assigned the same `DevAddr` by the NS (see Section 3.1.4.2). Upon receiving an `uplink` message, the NS leverages the MIC to identify devices: it iterates through all keys corresponding to the `DevAddr` until the computed MIC value matches the one present in the message header. While straightforward and already supported, such an approach has several shortcomings.

First, it does not offer a built-in way of rotating pseudonym, other than leaning on the rejoin process. As stated in Section 7.3.1.1, it induces significant energy consumption and generates metadata dangerous for privacy.

Second, this solution produces significant computation overhead on the NS. For D devices sharing the same `DevAddr`, receiving an `uplink` message generates $1+\frac{D-1}{2}$ MIC computations on average, contrary to 1 without shared `DevAddr`.

Third, this scheme would provide only a k-anonymity type of privacy protection [200], by hiding the End-Device in a set of indistinguishable devices. Generating such sets is a complex problem: heterogeneous groups of End-Devices makes it easy to differentiate between devices, and homogeneous groups introduce the risk of forming clusters containing only the same class of devices. Finally, it incites to increase the number of devices per group, further intensifying the computation overhead.

As shared `DevAddr` do not satisfy properties $\mathcal{P}_1$ (Unlinkability), $\mathcal{P}_2$ (Minimal communication overhead), and $\mathcal{P}_3$ (Low computation/memory overhead), *we exclude it from further comparisons.*

7.3.2 HASHA

Another potential avenue for pseudonyms, called HASHA, has been proposed by Zhang and Zhang to protect the location privacy of wireless sensor networks [231]. Here, the pseudonym is encrypted via a rolling key derived based on the hash of the previous payload. Unfortunately, this approach has several shortcomings. First, an eavesdropper can circumvent address randomization: 1) pseudonyms are generated based on original addresses and a fixed, publicly known value, and 2) the original address is sent in clear-text via sniffable beacon frames.

Second, the unlinkability property hinges on the attacker missing a single data packet, leading to a loss of synchronization with the targets. As long as the attacker is synchronized with senders, they can predict following identifier values. One could argue that LoRaWAN's reported 40% packet loss in urban environments [136] would help in that regard, but it does not offer a strong guarantee.

Third, devices must send acknowledgment frames to maintain synchronization. It would require LoRaWAN End-Devices to receive and parse `downlink` for all of their `uplink`, incurring higher energy consumption. Additionally, high numbers of `downlink` messages has been shown to deteriorate the overall performance of the network [150].

Because of these various issues regarding properties $\mathcal{P}_1$ (Unlinkability) and $\mathcal{P}_2$ (Minimal communication overhead), *we exclude the HASHA scheme from further comparisons.*

7.3.3 Pool-based pseudonyms

In this section, we explore the possible implementation of two solutions based on randomly sampling pseudonyms from a pool assigned to End-Devices.

7.3.3.1 Certificate-based pseudonyms

Due to unreliable network conditions, some Vehicular Ad hoc NETworks (VANETs) generate their pseudonyms in bulk, with vehicles randomly pooling one of them for a given time period [138]. As outlined in Section 2.4.5.4, this approach requires a cryptographic certificate to be sent along each data message [169], which poses multiple problems, both for their transfer, and storage.

First, it supposes the presence of the following lengthy fields in the header of messages:

- *Public Key*: generated by the vehicle for secure communication.
- *Actual pseudonym*: unique identifier assigned to the vehicle for a specific period.

- *Signature from the Sender (Vehicle)*: a mean for the receiver to verify the authenticity of the sender. When the vehicle sends a message, it signs the message using its private key, corresponding to the public key included in the certificate.
- *Signature from the Trusted Authority*: a mean for the receiver to trust the sender. The vehicle's certificate is signed by a trusted authority, ensuring authenticity and integrity.

The security of asymmetric cryptography depends particularly on key lengths. As of January 2019, the National Institute of Standards and Technology, USA (NIST) advise selecting RSA keys of at least 2048 bits. When using Elliptic Curves, keys must be over 224 bits for the same level of security [39]. These recommendations are relevant for years prior to 2030; longer keys will be required later. Similar values are given by other state-funded organization, such as the French National Agency for the Security of Information Systems (ANSSI) [26]. A 224-bit cryptographic key corresponds to 55% of the maximum LoRaWAN payload size, in the most used DR case (see Section 6.2.1.2). Its usage is unrealistic in a protocol as constrained as LoRaWAN. Multiple projects propose lightweight alternatives to existing public key cryptography schemes [202], both in key size and computations. However, they require (at best) 80 bits for keys alone, not accounting for signatures and pseudonym.

Second, one vehicle stores multiple pseudonyms, valid for a specific period of time. Constrained LoRaWAN End-Devices may not be able to manage a pool of lengthy pseudonyms, requiring additional costly communications to rotate them.

Thus, the implementation of pseudonyms in VANETs goes against properties $\mathcal{P}_2$ (Minimal communication overhead), $\mathcal{P}_3$ (Low computation/memory overhead), and $\mathcal{P}_5$ (Legacy support), and can not be transferred to the highly constrained LoRaWAN. *We exclude certificate-based pseudonyms from further comparisons.*

7.3.3.2 Range-based pseudonyms

Pseudonym ranges are akin to computers controlling multiple subnets instead of a single IP address, and randomly sampling an identifier when sending messages. Misra and Xue advise to split the original pseudonym space to protect the unlinkability of pseudonyms; "*the pseudonym space [is] divided into at least N^2 subranges*", with N as the number of active devices [152]. As the available address space in LoRaWAN is limited, further dividing it into subranges would create two major issues.

First, following the definition of subranges given by Misra and Xue, a `NwkAddr` of b bits would allow $\lfloor\sqrt{2^b}\rfloor$ unique active devices without permanent collisions. For instance, type 7 networks can currently be addressed using only $b = 7$ bits, corresponding $2^7 = 128$ unique End-Devices. Using subranges, this number would drop to 11. Increasing the number of active devices over that threshold produces collisions, generating extra MIC computations, as exposed for shared addresses in Section 7.3.1.2.

Second, when maximizing the number of active devices in a network, which happens quickly, ranges correspond to 1 or 2 pseudonyms. Once an End-Device has used all of its possible values, it is forced to re-use pseudonyms. Thus, an attacker can link back communications to previously observed values. As subranges should be communicated during a join process, requiring extra bytes on the air, updating them regularly would deplete battery life.

As this approach applied to LoRaWAN contradicts properties $\mathcal{P}_1$ (Unlinkability), $\mathcal{P}_2$ (Minimal communication overhead), and possibly $\mathcal{P}_3$ (Low computation/memory overhead), *we exclude it from further comparisons.*

7.3.4 Resolvable pseudonyms

Resolvable random Private Addresses (RPAs) include a random value and a hash computed using the random value and a private key, shared between sender and receiver. This straightforward mechanism is already deployed in BLE and represent a well-tested pseudonym scheme [92, sec. 1.3.2.2].

By design, resolvable pseudonyms respect most desired properties: they are unlinkable, reliable, and do not generate additional communication. Hence, *we propose to study them further in LoRaWAN.*

7.3.5 Encrypted link-layer pseudonyms

The approach by Armknecht et al [30], proposes to encrypt the MAC address via a pre-shared key along a sequence number to keep sender and receiver synchronized. However, this solution has several limitations. First, it is a completely new link-layer protocol, with no possible legacy support. Second, it introduces some delay and jitter after 7 nodes connected to an AP [30], but this may not be relevant for low-throughput LoRaWAN applications. Third, the implementation suffers from a significant computational complexity: each packet with an unknown address (i.e. no pre-generated value matches) requires to decrypt the s_i block for *all* keys. Ignoring apparent energy consumption issues, this can lead to denial-of-service attacks. Thus, *we exclude it from further comparisons.*

Some of these issues are resolved by *Shroud*, proposed by Greenstein et al. [90]. By pre-generating sequential pseudonyms on both the sender and receiver, they avoid having to communicate the counter itself, embedding it into the pseudonyms. However, pre-generating addresses still has a cost (encrypting the sequence number, and managing memory). In their paper, Greenstein et al. advocate for 50 pre-generated addresses, which corresponds to 1MB for 256 devices connected to an AP. Additionally, they propose a complete new link-layer protocol and extend address size without consideration for existing implementation and legacy support.

As the length of pseudonyms and the number of pre-generations can be adapted to LoRaWAN, *we propose to further study the concept as a possible adaptation respecting the protocol's constraints.*

7.4 Adapting pseudonym schemes to LoRaWAN

In this section, we present two pseudonym schemes adapted to LoRaWAN's constraints, inspired by resolvable private addresses (Section 2.4.5.3) and encrypted link-layer pseudonyms (Section 2.4.5.1).

7.4.1 Resolvable pseudonyms

In this section, we explore how BLE's Resolvable random Private Addresses (RPAs) can be adapted to LoRaWAN, by listing their requirements, equivalences, and associated challenges. Figure 7.1 illustrates both pseudonym generation and resolution.

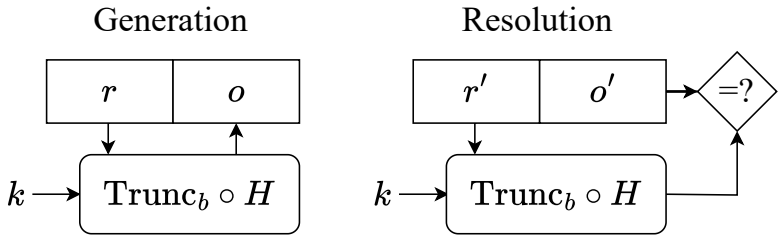


Figure 7.1: Generation and resolution of resolvable pseudonyms in LoRaWAN.

7.4.1.1 Requirements and adaptation

RPAs require devices to possess a resolution key k shared with the server, a random value r, and a hash function H. LoRaWAN includes a key derivation process during OTAA (see Section 3.1.6.2); k can thus be generated at the same time, and shared between the NS and End-Device. Generating a random value in constrained hardware can be a difficult task, as entropy sources are scarce and may lack robustness. For instance, the SX1272 transceiver utilizes the RSSI signal to produce random values, and can be influenced by jamming [205]. However, such an active attack aiming at predicting random values falls outside the scope of our threat model, and we consider randomness sources strong enough for this task. Finally, we select AES-128 CMAC as the hash function of the resolvable scheme, based on its deployment by default in the LoRaWAN standard (initially for MIC computations).

In practice, a LoRaWAN resolvable pseudonym leverages the `NwkAddr` field. We propose to subdivide it into 2: r (3 to 12 bits, based on the network type), and a truncated hash value $o = \text{Trunc}_b(\text{H}(r, k))$, where Trunc_b is the function that keeps only the first b bits, and b is between 4 and 13 bits.

To resolve a pseudonym $\phi = (r', o')$, the NS needs the same shared resolution key k used during generation. If both $\text{Trunc}_b(H(r', k))$ and o' are equal, the sender's identity is recovered; otherwise, the message may belong to another device and the remaining shared keys must be tested.

7.4.1.2 Challenges

The limited available space poses two significant challenges. First, 24-bit hashes in BLE RPAs are highly likely to be resolved by a single key. In contrast, hashes in LoRaWAN can be as short as 4 bits, leading to collisions, i.e. multiple keys generating the same hash, and a pseudonym corresponding to more than one End-Device. To avoid identity conflicts during the resolution, we propose to utilize the MIC (see Section 3.1.3), as illustrated by Figure 7.2. In case of collisions, the NS can compare the received MIC with the re-computed value based on the shared key, reliably identifying the actual sender.

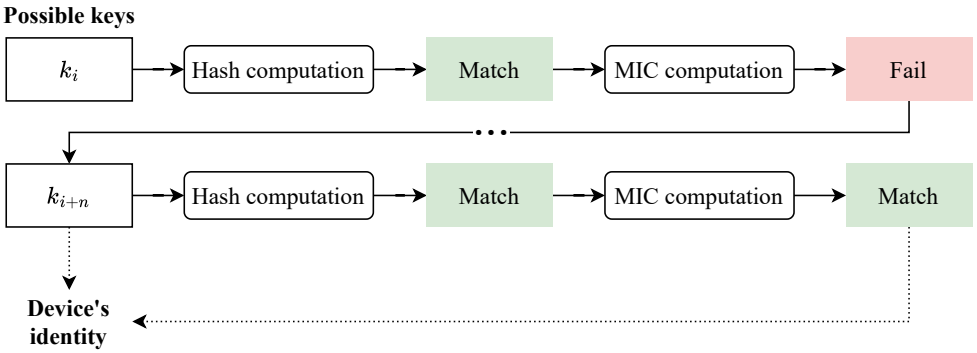


Figure 7.2: Device identification via MIC computation when using resolvable pseudonyms.

Second, LoRaWAN networks are designed to host a higher number of devices than BLE, leading to complexity issues when resolving pseudonyms. For instance, Android supports only 7 associated BLE devices simultaneously[1], in stark contrast with the thousands of concurrent End-Devices in some LoRaWAN deployments. When the NS receives an `uplink` message, it has to try *all* the keys of N active devices until it successfully resolves the received pseudonym ($1 + \frac{N-1}{2}$ tries on average). This open challenge for the `uplink` message resolution leads us to explore sequential pseudonyms in Section 7.4.2.

We note that this issue does not arise with `downlink` messages because an End-Device simply needs to resolve the pseudonym using its own key. Additionally, End-Devices listen to `downlink` channels during time slots closely following their `uplink` message, mostly receiving messages intended for them, and therefore requiring only a few resolution attempts.

Finally, to protect the `FCnt` from being used for pseudonym linkage purposes (see Chapter 4), we propose to encrypt this field like other network-specific information, as explored in Section 6.2.1.1, using the NwkSEncKey shared with the NS.

7.4.2 Sequential pseudonyms

In this section, we present how *Shroud*, the solution by Greenstein et al. [90] based on encrypted link-layer identifiers, can be adapted to LoRaWAN as *sequential pseudonyms*. Contrary to the initial proposition leveraging 128-bit identifiers, we design a tailored scheme respecting the space constraint already presented for RPAs in Section 7.4.1. We highlight technical requirements, steps taken to adapt the scheme to LoRaWAN, and accompanying challenges.

[1] https://android.googlesource.com/platform/external/bluetooth/bluedroid/+/master/include/bt_target.h#1428

7.4.2.1 Requirements and adaptation

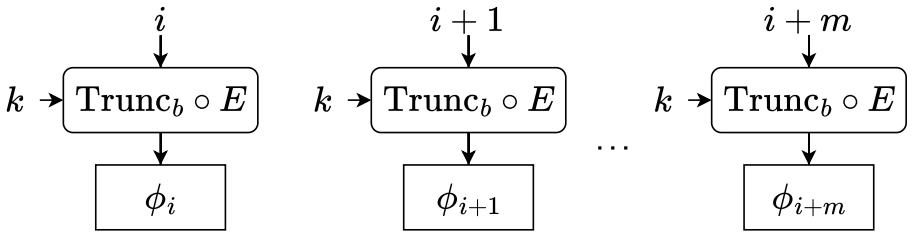


Figure 7.3: Sequential pseudonyms in LoRaWAN.

As illustrated in Figure 7.3, sequential pseudonyms require a counter i, a key k shared with the server, and an encryption function E. The LoRaWAN standard provides a counter by default, the `FCnt`, and a key derivation process between the NS and End-Devices (see Section 3.1.6.2). Likewise, we encrypt the counter using AES-CTR, which is already supported in the LoRaWAN specification. To match the size requirements of the `NwkAddr`, the encrypted output is truncated to b bits via the Trunc_b function, resulting in $\phi_i = \text{Trunc}_b(E(i, k))$.

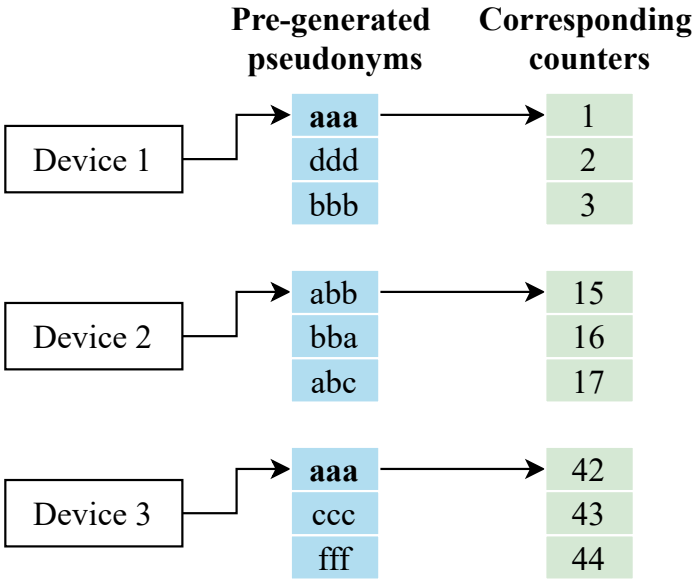


Figure 7.4: Server-side storage of sequential pseudonyms, for $m = 3$ and an example of collision in bold.

In practice, both the NS and End-Device generate a sequence of encrypted pseudonyms based on a synchronized counter, up to m values. Upon receiving a pseudonym ϕ', it is matched against pre-generated values, retrieving the underlying counter value and the sender's identity. As seen in Figure 7.4, the NS can potentially receive `uplink` messages from N active End-Devices, thus it maintains N lists of m pseudonyms. End-Devices only have to update their own.

7.4.2.2 Challenges

As highlighted previously, LoRaWAN identifiers are tightly constrained in space: only 7 to 25 bits are available. Hence, the probability of collision, i.e. multiple devices using the same

sequential pseudonym, is high. For instance, both Device 1 and 2 in Figure 7.4 use the pseudonym *aaa*, corresponding to different counters. Following propositions for resolvable pseudonyms, we leverage the MIC to detect the actual origin when two or more devices generate the same pseudonym.

The `FCnt` field can not be kept in clear to avoid linkability attacks (see Chapter 4). It is redundant with sequential pseudonyms, already functioning as counters themselves. We propose to extend the `NwkAddr` field by the 2 vacated bytes of the `FCnt` to reduce the number of potential collisions. The sequential scheme can thus utilize pseudonyms of length ranging from 23 (i.e. 16 + 7) to 41 (i.e. 16 + 25) bits, depending on the network type

7.5 Renewal strategies

We differentiate the pseudonym itself from its renewal strategy, i.e. how (often) it is updated, which applies to both approaches presented previously.[2] Inspired by existing solutions presented in Section 2.4.6, we discuss various methods to renew a pseudonym with regard to LoRaWAN specificities.

To counter a stable `DevAddr` (see Section 3.1.4.2), renewal strategies are organized along two axes: the renewal origin, and the renewal frequency. First, a pseudonym can be updated either at the demand of the server, or by the device independently. For both privacy and energy efficiency considerations, it is preferable for the End-Device to autonomously change its pseudonym (respectively: no metadata generation, and no additional costly communication).

Second, the renewal frequency is set following solutions detailed in related works: either on a time window (every M minutes) [92, 84, 138], or a number of messages (every N message). For instance, addresses in BLE are renewed every 15 minutes [92]. Both strategies can be enriched with random offsets (e.g.: a pseudonym is updated every $M + s_r$ minutes with s_r a random number of seconds) to break patterns [138].

The renewal frequency faces additional challenges due to synchronization issues: a) uncoordinated timing-based solutions can be abused [143] or require a buffer period [138], which may not be available for event-centric systems, b) synchronization between devices is complex to deploy [169], requiring numerous devices to correctly mix the signals [211].

Solutions based on precise coordination and mix zones are not conceivable in LoRaWAN, with its low traffic, uncoordinated End-Devices, and loosely synchronized servers/devices. On the other hand, generating pseudonyms is based on low-cost encryption operations and communications are sparse, enabling energy-efficient pseudonym renewals (see Section 7.6.3). Thus, we consider an approach similar to Shroud [90], where *pseudonyms are renewed for each* ***`uplink`*** *message.*

7.6 Evaluation

We evaluate both resolvable and sequential pseudonym schemes w.r.t. the properties presented in Section 7.2. When possible, we complement theoretical analyses by conducting simulations on a real-world dataset. Already presented in Sections 4.3 and 5.3, this dataset corresponds to 71 million of `uplink` messages from third-parties captured from June 2020 to August 2023.

7.6.1 $\mathcal{P}_1$: Unlinkability

The unlinkability property is related to pseudonyms themselves as well as surrounding metadata. First, both resolvable and sequential schemes generate their pseudonyms for each message via AES as a Cryptographically Secure PRNG (CSPRNG). This guarantees a high entropy output, meaning the pseudonyms themselves are unlinkable.

Second, both the payload encryption and MIC computation require *unique* clear-text `FCnt` values. Even if we encrypt the `FCnt` with resolvable pseudonyms or hide it in sequential pseudonyms, its underlying value is still available to both the End-Device and NS. Thus, for two payloads with equal inputs (e.g. temperature sensors reporting the same value), the encrypted output is guaranteed to remain different [65, sec. 4.3.3].

[2] In theory, the counter i of sequential pseudonyms could be incremented only every 10 messages.

Additionally, the rest of the header (`FCtrl` and `FPort`) contains flags and options amounting for a total of 16 bits. In practice, only the `FPort` exhibits *some* contextual variation, with ∼3.5 bits of entropy. As outlined in Section 6.4.3.1, it can be encrypted with limited overhead.

Hence, after encrypting the `FCnt` and `FPort`, *we consider that both resolvable and sequential pseudonyms are unlinkable.*

7.6.2 $\mathcal{P}_2$: Minimal communication overhead

In terms of communication, both resolvable and sequential pseudonyms are indistinguishable. There is no alteration to the message size, and under the assumption of no desynchronization (see Section 7.6.4.2), no additional communication overhead is induced. If an unlikely desynchronization occurs, a rejoin process is required and previously lost data must be sent again.

7.6.3 $\mathcal{P}_3$: Low computation/memory overhead

We analyze computation and memory overheads, for both End-Devices and the NS in worst case scenarios. We start by considering N active End-Devices managed via a single NS. For both schemes, we assume c collisions at the NS for an `uplink` message, i.e. c End-Devices using the same pseudonym. A summary of the comparison between both schemes is available in Table 7.1. We choose to estimate the overhead produced by the generation and handling of pseudonyms in AES-128 encryption block operations, as well as the amount of memory in bits. AES-128 encryption is a one of the most expensive operations done by the device (excepting communication itself), which means that other operations are comparatively negligible. As a fundamental tool already supported by the standard, it is utilized throughout both pseudonym schemes, making it a perfect candidate to estimate their overhead on common grounds.

Table 7.1: Computation and memory overhead of resolvable and pseudonym schemes.

Scheme	**Transmission** (# AES)	**Reception** (# AES)	**Memory** (bits)
Resolvable (Network Server)	2	$4c + 2N + 2$	-
Sequential (Network Server)	2	$4c + 2$	$N(3mb - \ell)$
Resolvable (End-Device)	5	4	-
Sequential (End-Device)	2	2	$3mb - \ell$

We note values correspond to *overhead*, meaning they do not represent existing requirements of the current version of the standard.

7.6.3.1 Computation overhead

We do not take into account existing encrypted data such as MAC commands (see Section 6.4.3.1) and consider each encryption operation in isolation. For instance, the resolvable scheme encrypts the `FCnt`, inducing an overhead of one AES block encryption, for both transmission and reception. While suboptimal, we select the worst case scenario, regardless of possible optimizations during implementation.

Transmission: Sequential pseudonyms require only 1 AES encryption to encrypt the counter and generate the pseudonym on both End-Device and NS, and 1 to encrypt the `FPort`. On the other hand, for resolvable pseudonyms, the End-Device has to do 5 encryption operations: it generates a random value (1 operation), produces the hash via CMAC (2 operations) [194], as well as encrypts of the `FCnt` (1 operation) and `FPort` (1 operation). Both `FCnt` and `FPort` are also encrypted by the NS.

Reception: On the server side, the computation overhead depends on the number of collisions c. By default, a collision is handled by computing the MIC, requiring 4 AES encryption (2 CMAC operations), totaling to $4c$ AES encryption operations [65, sec. 4.4]. For resolvable

pseudonyms, the NS needs to *resolve* pseudonyms by computing the hash via CMAC using the shared keys of all N active devices, so $2N$ AES encryption operations.[3] Finally, it also has to decrypt the `FCnt` and `FPort`, summing up to $4c + 2N + 2$ encryption operations. For sequential pseudonyms, the NS only has to maintain the list of pre-generated pseudonyms by producing a new one once a message is received (1 operation) as well as decrypting the `FPort` (1 operation), resulting in $4c + 2$ encryption operations.

For End-Devices, resolvable pseudonyms require 4 encryption operations: 2 for hash computation via CMAC to resolve the pseudonym, and 2 for `FCnt` and `FPort` decryption. Sequential pseudonyms only need 2 encryption operations, to produce a new pseudonym while maintaining the pre-generated list length, and to decrypt the `FPort`.[4]

In summary, sequential pseudonyms show a significantely lower computation overhead than resolvable pseudonyms, notably during reception server-side, as well as in both transmission and reception device-side.

7.6.3.2 Memory overhead

Sequential pseudonyms require $3mb - \ell$ bits of storage, for m pre-generated pseudonyms, with the NS needing it for all N active devices. More precisely, each End-Device utilizes 3 lists of m pre-generated pseudonyms, one for each `FCnt` (see Section 3.1.3). A pseudonym requires b bits, and we make sure to subtract the existing length (ℓ) of the `NwkAddr` from the overhead.

We confirm in Section 7.6.4.2 that m can be kept as low as 15 and pseudonyms are at most 41 bits. In a worst case scenario (high number of lengthy pre-generated pseudonyms), it would correspond to $3mb - \ell = 3 \times 15 \times 41 - 41 = 1804$ bits. Although much higher than current implementations, this is still manageable, corresponding to ∼3% of RAM of lower-end End-Devices.[5]

Thus, the lower computation overhead of sequential pseudonyms comes with a higher, yet low, memory overhead than their resolvable counterpart.

7.6.4 $\mathcal{P}_4$: Reliability

We investigate two dimensions of pseudonym robustness. First, as outlined in Section 7.4, both resolvable and sequential schemes can result in pseudonym collisions, necessitating additional operations. Second, packet loss might lead to desynchronization between End-Devices and the NS, requiring a costly rejoin process upon detection.

7.6.4.1 Pseudonym collisions

We analyze collisions, first through analytical evaluation and then via simulations on the dataset presented in Sections 4.3 and 5.3.

Analytical evaluation: Our objective is to theoretically determine the number of pseudonym collisions (Y) for both schemes. We note that this problem concerns solely the NS, which has to track multiple End-Devices and associated shared keys. We start by analyzing sequential pseudonyms.

Let ϕ be a pseudonym and X the number of devices using this pseudonym at a given time. For N devices, sequential pseudonyms require the NS to maintain N lists of m pseudonyms. Let p be the probability that ϕ belongs to a generated pseudonym list. Counting the number of devices producing ϕ in their pseudonym list is the same as counting the number of successes of a Bernoulli experiment of probability p. Formally, this counting follows a binomial probability law with the number of lists tested and p as parameters. Since ϕ necessarily belongs to at least one of the lists, we can write $X = 1 + Y$, where Y is the number of pseudonym collisions and X the number of matching lists. Y follows the binomial law of parameter $(N - 1, p)$.

[3] We ignore the cost of looking up keys, assuming it is negligible compared to the actual CMAC computations.

[4] We note an encryption operation is required when receiving a sequential pseudonym on the End-Device because the NS do not answer with the last sequential pseudonym used in an `uplink`. Rather, it communicates a pseudonym generated specifically for `downlink` communications, following the existing concept of direction-based counters (`FCntUp` and `FCntDown`).

[5] https://lora-developers.semtech.com/documentation/tech-papers-and-guides/mcu-memory-management/

Let T be the size of the address space; for instance, a 32-bit pseudonym has $T = 2^{32}$ available addresses. Then $p = 1 - (1 - \frac{1}{T})^m$. In conclusion, the probability law of X is the same as $1 + Y$ where Y follows a binomial law of parameter $\left(N - 1, 1 - (1 - \frac{1}{T})^m\right)$.

Considering resolvable pseudonyms, half of the available bits are utilized to store a random value. Thus, for a b-bit pseudonym, there are $T = 2^{\frac{b}{2}}$ available addresses. Using the expression deduced above for Y, we notice that decreasing T increases the probability of collisions. In conclusion, because resolvable pseudonyms have a smaller address space than sequential pseudonyms, they have a higher probability of collisions.

Simulation: In addition to the analytical evaluation, we simulate the two pseudonym schemes on the dataset detailed in Sections 4.3 and 5.3. To do so, we need to know which End-Devices are active at a given time. Third-party data does not include such an information: an End-Device could disconnect and its `DevAddr` be re-assigned later to a new End-Device without external notice.

Hence, we chronologically analyze incoming `uplink` messages in sliding time windows of one month, considering all observed `DevAddr` as unique active End-Devices during the whole duration. We replicate the process for each month in the dataset, average the results, and corresponding to a median of 4789 active End-Devices monthly. Selecting one month as a time window insures enough End-Devices are considered active, simulating a well-populated network.

For each received `uplink` message, we simulate both resolvable and sequential pseudonyms, and enumerate the number of collisions. When a pseudonym matches n devices, we count $n-1$ collisions (because one of them is the real receiver). We configure sequential pseudonyms with $m \in \{5, 10, 15, 30\}$ pre-generated pseudonyms, with and without utilizing the `FCnt` for comparison purposes.

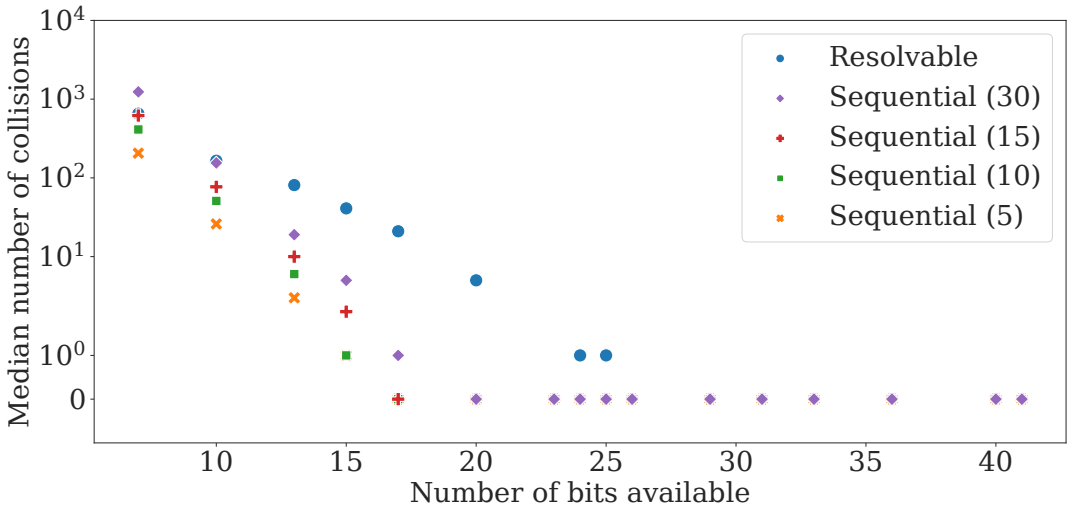


Figure 7.5: Median number of collisions for an `uplink` message received by the NS (data points for sequential pseudonyms overlapping for 0 collisions with $b \geq 20$).

Figure 7.5 illustrates the median number of collisions for *a single* ***`uplink`*** *message* based on the required number of bits per pseudonym. The sequential scheme generates fewer collisions than the resolvable scheme across nearly all numbers of available bits, even in scenarios where the `FCnt` is not utilized (less than 25 bits). As the number of collisions is computed for a single incoming `uplink` message, resolvable pseudonyms producing a median of 1 collision with 25-bits addresses could increase quickly as networks grow to thousands of active devices.

In summary, *sequential pseudonyms outperform resolvable ones*, demonstrating significantly fewer collisions on the server-side, both analytically and through simulations.

7.6.4.2 Message losses and desynchronization

Desynchronization occurs between End-Devices and the NS when consecutive losses surpass the number of pre-generated sequential pseudonym. Any subsequent message is unrecognized

by the receiving end, both potentially loosing data and necessitating an energy-consuming rejoin process upon detection.

One option to detect desynchronization requires some sort of periodic acknowledgment in `downlink` messages and has to be done on the End-Device side, as it is impossible for the NS to directly contact a device. An End-Device would still send multiple `uplink` messages before detecting the lack of periodic `downlink` acknowledgment and then proceed with a rejoin process. Thus, it is significantly better to avoid desynchronization completely.

As seen in Section 3.1.7, the Packet Loss Rate (PLR) of LoRa networks highly depends on the environment and deployment conditions, reaching as high as 40% PLR [136], suggesting that desynchronization could be a recurring problem for sequential pseudonyms. To alleviate such concerns, we analyze the probability of desynchronization both analytically and through simulation.

Analytical evaluation: For a packet loss rate PLR, the average number of `uplink` messages sent before m consecutive losses is $\frac{PLR^{-m}-1}{1-PLR}$ [87], where m is the pre-generated list length.

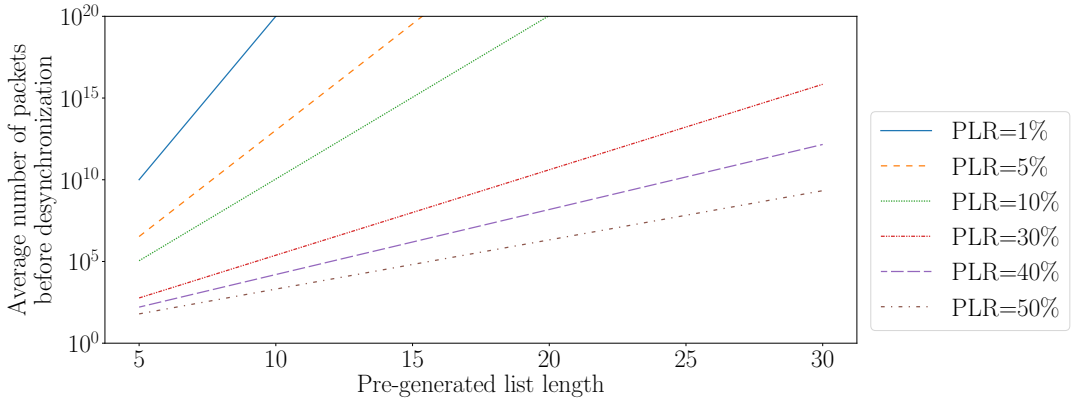


Figure 7.6: Average number of packets before desynchronization (cropped for clarity).

As seen in Figure 7.6, lists of 15 pre-generated pseudonyms require $\sim 10^5$ packets on average before a desynchronization under an extreme 50% PLR. To put it into perspective, it would take roughly 11 years for an End-Device communicating every hour to lose synchronization with the NS. This duration usually exceeds the battery lifetime.

Simulation: We compute the percentage of End-Devices desynchronized at least once during the observation period based on the third-party real-world dataset. We leverage the `FCnt` to identify losses (i.e. gaps in sequences) and select End-Devices with an estimated PLR below 50% to exclude devices with significantly erratic behavior, probably due to experiments or End-Devices at the limit of our coverage area.

Table 7.2: Comparison between simulations and theoretically expected fractions of devices desynchronizing at least once.

Number of pre-generated pseudonyms (m)	5	10	15	30
Measured fraction of devices	30.5%	9.6%	4.8%	0.0%
Expected probability	31.5%	1.6%	0.0%	0.0%

Table 7.2 compares the measurements based on the simulation versus the expected probability computed following the theoretical approach. 30.5% of devices experience desynchronization with only 5 pre-generated pseudonyms, but this number falls to 0.0% with a longer list of 30 pseudonyms, following expectation. For $m \in \{10, 15\}$, we measure a higher fraction of desynchronized devices. This difference can be explained by End-Devices at the edge of our coverage area, entering and leaving reception range. With smartly deployed gateways, we believe effective desynchronizations would fall to numbers closer to theoretical values.

In conclusion, such results demonstrate that *a long enough list of pre-generated pseudonyms prevents desynchronization* for most to all devices. Based on the theoretical framework and

simulations, at least 15 pseudonyms are enough to avoid this feared event, while leading to acceptable memory overhead (see Section 7.6.3.2).

7.6.5 $\mathcal{P}_5$: Legacy support

Both schemes re-purpose the `NwkAddr` field of the `DevAddr`. Preserving the length of said field allows existing parsers to be quickly updated. While multiple addressing schemes could cohabit, the `FCnt` encryption and re-purposing requires updating the standard version. A new version can be defined through the Major Version field, currently leveraged to differentiate messages using LoRaWAN 1.0.X or 1.1.Y.

7.6.6 Summary and artifacts

According to the results presented in this section, sequential pseudonyms are the best solution for privacy-preserving pseudonyms in LoRaWAN. They outperform their resolvable alternative w.r.t. to nearly all properties, with two slight drawbacks. First, they require marginally more memory to store all the pre-generated pseudonyms. Second, desynchronization can induce additional communication costs when dealing with high packet loss rate and low number of pre-generated pseudonyms. However, this last issue can be resolved by adopting lists of at least 15 pseudonyms.

We provide a proof-of-concept of sequential addresses implemented in Python to demonstrate the main logic[6] as well as the source code used to estimate collisions and desynchronizations[7].

7.7 Complementary security considerations

We examine the security implications of both resolvable and sequential pseudonyms, delving into possible side-channel attacks.

First, jamming techniques are a threat against synchronized sequential pseudonyms. However, it would require to 1) block *all* gateways potentially receiving `uplink` messages, and 2) effectively jam a significant number of *consecutive* messages, i.e. surpassing the number of pre-generated pseudonyms m, which could span multiple days. Failing to jam a single message would allow the End-Device and NS to maintain synchronization.

Second, an eavesdropper could capture `uplink` messages and replay them later to trigger specific server behavior [100, 232]. For example, Zang and Lin demonstrate that BLE central devices respond differently depending on whether a pseudonym is included in their allowlist [232]. By replaying packets, they are able to effectively track devices regardless of MAC address randomization. In practice, such an attack is possible due to the lack of sequence number or time-based randomized address generation. In LoRaWAN, both resolvable and sequential pseudonym scheme specifically preserve the `FCnt`, allowing the NS to ignore any replayed message. To further protect pseudonyms from *jamming and replays* attacks (where the original message is never received by the NS) [100], it is possible to include a timestamp in the header.[8] However, it induces significant energy consumption due to extra bytes for each transmitted message.

Third, `uplink` message length has been used for linking and fingerprinting (see Part II), as well as activity inference [15]. Although the same payload encrypted twice produces different outputs, its length is stable and could be used to link pseudonyms. At the cost of communication overhead (see Section 6.4.3.2), the payload could be padded. Lastly, LoRaWAN already provides a robust mechanism to detect message tampering via the MIC [100], effectively protecting the pseudonyms.

[6] https://gitlab.inria.fr/spelissi/lorawan-pseudonyms/-/tree/pseudonyms/simulator

[7] https://gitlab.inria.fr/jaalmoes/spistats

[8] Introducing a timestamp in the payload is ill-advised, because the encryption is done by the AS and the NS should have access to this information. Additionally, the timestamp has to be included into the MIC computation to avoid tampering.

7.8 Applicability to other LPWAN protocols

Although widely deployed, LoRaWAN is not the only LPWAN protocol, nor will it be the sole technical solution forever. In this section, we discuss how the constraints and properties applied to (sequential) pseudonyms in LoRaWAN can be extended to other existing or future LPWAN protocols.

According to surveys [16, 58, 206], there are around 3 prevalent LPWAN standards: Sigfox (11 millions devices deployed) [134], NB-IoT (based on the 3GPP LTE standard) [11], and Dash7 [225].[9] While their exact technical implementation, scope, and business model may differ, their constraints are similar. They are all designed with long range and reasonably low energy consumption in mind. Table 7.3 summarizes the current state of identifiers in LPWANs.

Table 7.3: Overview of the identifiers in various LPWAN protocols.

Protocol	Identifier	Rotation	Field length
LoRaWAN	DevAddr	Every join/rejoin	7 to 25 bits [65, Sec. 6.1.2.1]
Sigfox	Identifier (ID)	Never	32 bits [134, Sec. 3.7]
NB-IoT	Temporary Mobile Subscriber Identity (TMSI)	Every join/rejoin	32 bits [11, Sec. 2.4]
Dash7	Virtual ID (VID)	At the discretion of the administrator	16 bits [225]

LPWAN protocols can be classified into two groups based on their identifier policy: a) a lifetime static identifier (e.g. Sigfox), and b) a session pseudonym (e.g. LoRaWAN). In the second case, the pseudonym is updated only if the device is disconnected from the network or by explicitly sending a command (e.g. the `ForceRejoinReq` in LoRaWAN). Neither option fully satisfies the unlinkability property $\mathcal{P}_1$, demanding regular pseudonym rotations.

At a high level, pseudonyms technically require two building blocks that are already available to LPWANs: enough bits to transmit them, and cryptographic primitives for their generation and resolution. First, while LPWAN protocols leverage identifiers of limited size to reduce time-on-air and energy consumption, we have demonstrated that this number is manageable for sequential pseudonyms. Second, all LPWAN protocols presented here support the cryptographic primitives required to produce encrypted counters for sequential pseudonyms. Sigfox [134, Sec. 5.3] and Dash7 [224] operate using AES-128, and NB-IoT is based on the LTE standard utilizing AES-128 for various tasks [12, Sec. 5.1.3.2].

Adapting sequential pseudonyms to other LPWAN protocols would require additional research to correctly implement them while taking into account all the specificities. For example, a similar update to the `FCnt` in LoRaWAN can be done in Sigfox to hide its 12-bits Message Counter [134, sec. 3.7]. However, we believe regular pseudonym rotations is both possible and highly beneficial for the privacy of LPWAN protocols.

7.9 Limitations and future works

Despite our best efforts to conduct a thorough analysis of privacy-preserving pseudonyms in LoRaWAN, some limitations remain.

Additional identification techniques: While we make sure to encrypt and/or hide metadata such as the `FCnt` and `FPort`, other information can be utilized to link `uplink` messages,

[9] Other protocols are often cited, such as Ingenu [109] or Weightless-P. However, they did not achieve widespread adoption. In any case, they show similar design trends, including identifiers of limited length, respectively of 32 and 18 bits.

notably the radio signal itself (see RSSI fingerprinting in Chapter 5). In this case, a lack of stable identifiers to group messages during training would warrant additional work in real-world scenarios.

Lack of implementation: Actually implementing sequential pseudonyms in open source libraries such as Chirpstack or The Things Network would encourage the standardization body to consider our approach. Although it requires significantly more investment to deploy a complete solution, we provide a proof-of-concept in Python, demonstrating the feasibility of sequential pseudonyms.[10] After presenting our work to the LoRa Alliance, responsible for specifying the LoRaWAN protocol, we received interest and positive feedback.

Major Version required: Multiple versions of the protocol can cohabit thanks to the Major Version field. Hence, backward compatibility is guaranteed with introducing a new major version. Previous implementations of NS would not be able to handle pseudonyms and could dynamically discard them based on the protocol's version number, while up-to-date servers could support both solutions.

Limited to `uplink` messages: As we focus on the `DevAddr`, we did not explore possible pseudonyms replacing the 64-bit `DevEUI` used in linkage attacks (Chapter 4). Instead of sending this unique identifier in clear to any eavesdropper, End-Devices could generate a temporary and unlinkable value via their stored NwkKey, resolved by the NS akin to Resolvable random Private Address in BLE. Then, linking the pseudonym to following `uplink` message would be meaningless, as it would correctly separate the *identity* of a device from its *activity*. Further research is required to study the feasibility of such a solution.

7.10 Conclusion

While pseudonyms have been widely studied in the literature and are already deployed in some wireless technologies, we present the first solution tailored to LoRaWAN. After defining key properties motivated by both privacy and resource consumption, we analyze various existing approaches, from VANETs to LoRaWAN's legacy features. Then, we propose two solutions thwarting identifier-based tracking: resolvable and sequential pseudonyms. We evaluate them both theoretically and via simulations on a real-world dataset, based on computation and memory overheads, and probability of adverse events. We show that the sequential scheme outperforms its resolvable counterpart, with a reduced memory overhead and limited risks of collisions and desynchronization when configured to work with 15 pre-generated pseudonyms.

While our work focuses on LoRaWAN, based on our preliminary analyses, such a solution could also be adapted to other LPWAN protocols, including Sigfox and NB-IoT. Moving forward, this proposal still needs to be validated with an integration and evaluation in a real LoRaWAN network.

[10] https://gitlab.inria.fr/spelissi/lorawan-pseudonyms/-/tree/pseudonyms/simulator

Part IV

An Alternative IoT Context: Machine Learning-based DNS Traffic Identification

Chapter 8

Identifying IoT devices via encrypted DNS traffic

Contents

In order to broaden the scope of our research and adapt our methodology, we now study on a cornerstone of the Internet (of Things): DNS. In order to protect end-users' privacy, multiple recent solutions, such as DNS-over-HTTPS (DoH), encrypt DNS.

In this chapter, we explore IoT device identification based solely on encrypted DNS traffic, and assess if protections are sufficient to protect privacy. Focusing on the use-case of a smart home, we leverage a extensive IoT dataset and show that consumer-grade devices can be reliably identified, with up to 0.98 balanced accuracy. We study possible countermeasures such as padding and show that they effectively thwart our attack. Finally, we discover that half of the evaluated DNS resolvers do not respect the corresponding standard, compromising users' privacy.

This chapter is partially based on our work *Does Your Smart Home Tell All? IoT Device Identification through DNS-over-HTTPS* (currently under review). The present chapter differs mainly by the following points:

- We provide comprehensive motivation and contextualization w.r.t. previous works, as well as background information, including a comparison of various alternatives for encrypted DNS.
- We expand the threat model to include both defensive IoT identification, and IPv6-based vulnerabilities.
- We compare in greater details various machine learning methods with the state of the art, both in terms of accuracy and prediction time. In addition, we further analyze the cause of misbehaving DNS resolvers w.r.t. padding responses.

8.1 Introduction

Contrary to LoRaWAN devices studied in previous chapters, consumer-grade IoT devices generally have access to greater amounts of computation power and battery. Equally ubiquitous in everyday environment, there are often deployed in smart homes, including speakers, baby monitors, and automation devices [203].

In this ecosystem, DNS plays a crucial role, allowing communication over the Internet between devices and their respective application servers. Efforts have been made in recent years to protect this previously clear-text protocol: multiple alternatives encrypting its content are now available and widely deployed, including DNS-over-HTTPS (DoH). Its ties with human *activity* as well as large-scale deployment make it a prime candidate for studies to better understand associated privacy implications. Indeed, as seen in Section 2.3.2.1, determining which device is currently deployed in a house represents a threat to users, allowing outsiders to profile and *identify* them [15, 27, 66, 131, 186].

Inferring activity or identifying devices through their DNS traffic have been studied in various contexts. Table 8.1 outlines the current state of research. While extensive work have been conducted on clear-text DNS for both Web browsing and IoT devices, as well as DoH for Web browsing, a notable gap remains in the analysis of IoT-based DoH traffic. Additionally, it has been shown that *IoT traffic exhibits distinct characteristics compared to Web browsing*, such as length of queried domain names and timing of communications [168, 220]. This prompts for further research to confirm whether existing identification techniques remain effective in encrypted DNS within IoT networks.

Table 8.1: Summary of related works on DNS-based identification.

Traffic type \ Protocol	Clear-text DNS	DNS-over-HTTPS (DoH)
Web browsing	[93, 122]	[53, 69, 191]
IoT	[18, 168, 203]	Our work

In this chapter, we address this gap by analyzing a large dataset of DoH traffic generated by 34 consumer-grade IoT devices representative of a smart home environment. First, we summarize necessary background on DNS and its encrypted versions, before adapting the threat model to a smart home setup. Then, we detail our testbed and our DoH-based IoT

device identification process. After presenting the evaluation methodology, we highlight key results and discuss their implication for privacy as well as inherent limitations of our approach.

8.2 Background

In this section, we provide essential information on DNS in IoT devices and its encryption.

8.2.1 DNS in consumer-grade IoT

Consumer-grade IoT devices regularly communicate over IP with remote hosts, retrieving software updates or transmitting data. Instead of identifying remote connections with hard-coded IP addresses, which can change overtime, devices address servers via a *domain name*, dynamically *resolved* by a DNS server/resolver.

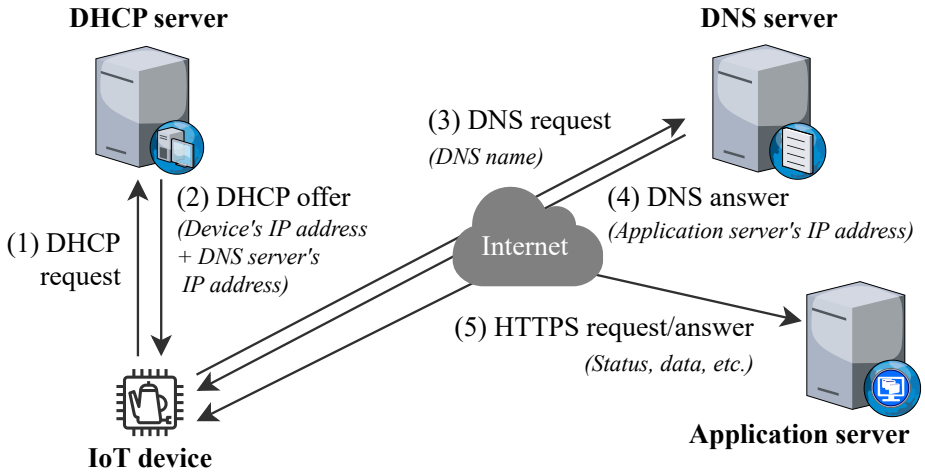


Figure 8.1: Example architecture of a consumer-grade IoT setup.

Figure 8.1 illustrates a typical domestic IoT environment. As soon as devices are switched on, they obtain their own IP address from a DHCP server within the local network, as well as the IP address of a DNS server. A DNS resolver may be preconfigured within devices [220]; however, such occurrences are rare to prevent malfunctioning when the IP of the DNS resolver is updated. Alternatively, DNS servers can be deployed locally, which does not significantly change the setup nor the threat model and attack. Then, devices obtain IP addresses of remote application servers by sending their corresponding domain names to the DNS resolver. Following communications *usually* occur over TLS-encrypted HTTP sessions.[1]

While IoT devices continuously report to application servers [220], the majority of their DNS requests are generated within the first few minutes of establishing a connection to fetch updates and transmit status reports. Hence, analyzing communication patterns in the initial traffic offers an effective and easily automated method for device identification [203] (see Section 8.6.1.1).

8.2.2 Encrypted DNS protocols

DNS is historically a clear-text protocol: requests and responses are transmitted unencrypted, exposing resources (e.g. domain names) and corresponding IP addresses to any eavesdropper on the path between a device and the DNS resolver. In order to provide confidentiality and privacy for users, multiple approaches implementing an encryption layer have recently been developed.

As presented in Table 8.2, most privacy enhancing solutions encapsulate DNS in another protocol providing encryption. For instance, multiple IETF standards select this approach by transmitting DNS in a Transport Layer Security (TLS) session, either directly or over

[1] Mischievous readers will note that some HTTP sessions may be unencrypted, but we hope they are.

another protocol: DNS-over-TLS (DoT) [102], DNS-over-QUIC (DoQ) [105], or DNS-over-HTTPS (DoH) [101]. Additionally, experimental IETF standards such as DNS-over-DTLS (DoDTLS) [176], ongoing works on DNS-over-CoAP (DoC) [130], or open source alternative including DNSCrypt[2] exist, but lack large-scale adoption [141].

Table 8.2: Encrypted DNS solutions.

Name	Encryption	IETF standard	Port
DNS-over-HTTPS (DoH)	TLS	Yes	443 (TCP)
DNS-over-TLS (DoT)	TLS	Yes	853 (TCP)
DNS-over-QUIC (DoQ)	TLS	Yes	853 (UDP)
DNS-over-DTLS (DoDTLS)	TLS	Experimental	853 (UDP)
DNS-over-CoAP (DoC)	TLS or OSCORE	Draft	5683
DNSCrypt	Custom	No	443 (TCP & UDP)

We focus our study on DoH for two main reasons.[3] First, it benefits from a large-scale support in several major DNS resolvers, such as Cloudflare [2] and Google [4], and in Linux systems via multiple solutions (e.g. BIND [1] or Unbound [9]). Second, routing DNS traffic in port 443 along other existing HTTPS communications (e.g. Web browsing) [3] provides another layer of privacy protection, as DNS needs to be successfully extracted, and then analyzed. On the other hand, port 853 is utilized exclusively for DoT or DoQ (respectively over TPC and UDP), facilitating the attack.

8.2.3 DNS-over-HTTPS

DoH is a simple encapsulation: a DNS request is sent as an HTTP POST or GET request to a DNS server, encoded in the same binary format it would have directly over UDP:

```
GET /dns-query?dns=q80BAAABAAAAAAAAA3d3dwdleGFtcGxlA2NvbQAAAQAB HTTP/1.1
Host: dns.google
Accept: application/dns-message
```

Here, the `dns` URL query parameter contains a DNS request for `www.example.com`'s A record, in binary format (and base64 encoded to be used as URL parameter). Alternatively, some resolvers such as Google and Cloudflare support a non-standardized JSON format for easier human parsing and development [2, 4]. In this work, we focus on the official and widely deployed standard using DNS in wire format.

From an eavesdropper's perspective, there are two main distinctions between DoH and traditional DNS traffic. First, DoH includes extra messages for TCP and TLS session initiation and termination, in addition to packets carrying requests and responses, whereas DNS traffic comprises only the latter messages. Second, the size of packets containing queries and responses varies: while encapsulation in HTTPS adds a constant amount of data to each message, the message size still depends on the content of the request or response.

8.3 Threat model

In this section, we present the threat model relative to device identification via DoH traffic. Previous threat models are targeting wireless protocols with a passive eavesdropper possessing some sort of receiver. We adapt the privacy threat model presented in Section 2.3.1 to a smart home setting and define two scenarios where stable patterns of DNS traffic allow an attacker to reliably track IoT devices, illustrated in Figure 8.2.

[2] https://dnscrypt.info/

[3] While we only present results obtained using DoH, we also test DoT observe similar values and behaviors. All TLS-based DNS encryption (excluding DoDTLS because of possible fragmentation) should follow the same trends.

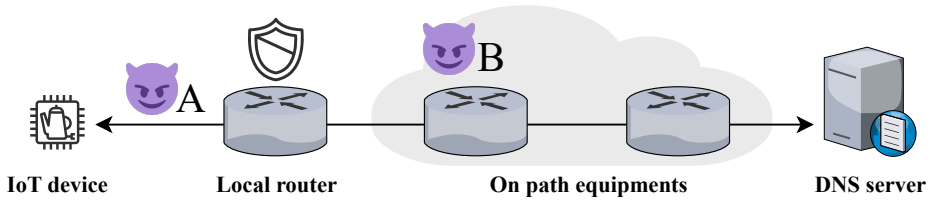


Figure 8.2: Threat model, with an attacker A in the local network, and an external attacker B on path.

8.3.1 Position in the network

We consider two possible locations for an attacker. First, they can monitor communications on the *local network*. For instance, eavesdroppers could have access to a malicious IoT device sniffing the surrounding traffic [149], or a compromised local router.

Second, we discuss an attacker outside the local network, *between the local router and DNS resolver*, tracking devices via uniquely attributed IPv6 addresses. We consider this scenario relevant and possible for multiple reasons. The ongoing large-scale adoption of IPv6 enables directly addressing devices behind a router, offering a one-to-one mapping without relying on NAT [181]. To avoid tracking, both parts of the IPv6 address can be dynamically updated: 1) Internet Service Providers (ISPs) offer Network Prefix rotation, a way to change the prefix corresponding to a given router (e.g. all devices in a household), and 2) devices can update their unique Interface Identifier accordingly. However, this process occurs by default every 24 hours which still enables an eavesdropper to track devices during this time period [40]. Additionally, multiple works have shown that correctly implementing IPv6 address rotation is complex, and some vulnerabilities may persist, leading to robust tracking via this protocol [180, 181].

8.3.2 Data access

Following the works from Thompson et al. on rapid IoT device identification via clear-text DNS traffic [203], we focus on the first minutes of communication, assuming an attacker captures network traces during the device's boot. Additionally, we consider DoH traffic in isolation for identification, excluding underlying protocols such as IP or Ethernet, for two reasons: 1) we demonstrate the high reliability of DoH alone for device identification, and 2) other information such as MAC or IP addresses may vary over time [80] and/or may not be available to an eavesdropper based on their position in the network.

8.3.3 Alternative scenarios

Finally, we note that device identification can also be leveraged in defensive scenarios, as studied in Chapter 5 and illustrated by a shield on the local router in Figure 8.2. Previous works have indeed shown that a network administrator can detect misbehaving and/or compromised IoT devices through continuous monitoring [149].

8.4 IoT testbed

To conduct our experiments, we set up a testbest collecting DNS traffic produced by IoT devices. As seen in Figure 8.3, they are plugged in a set of smart plugs. Both devices and smart plugs are connected via Wi-Fi to a server providing IP connectivity towards the Internet as an Access Point. This server fulfills two roles: 1) it records all network activity going through its access point, and 2) it controls smart plugs via automation scripts to switch them on and off, effectively restarting IoT devices (see Section 8.6.1.1).

8.4.1 IoT Devices

The complete list of the 34 selected consumer IoT devices is available in Table 8.3. We consider a significant number of devices commonly used in smart homes and offering high

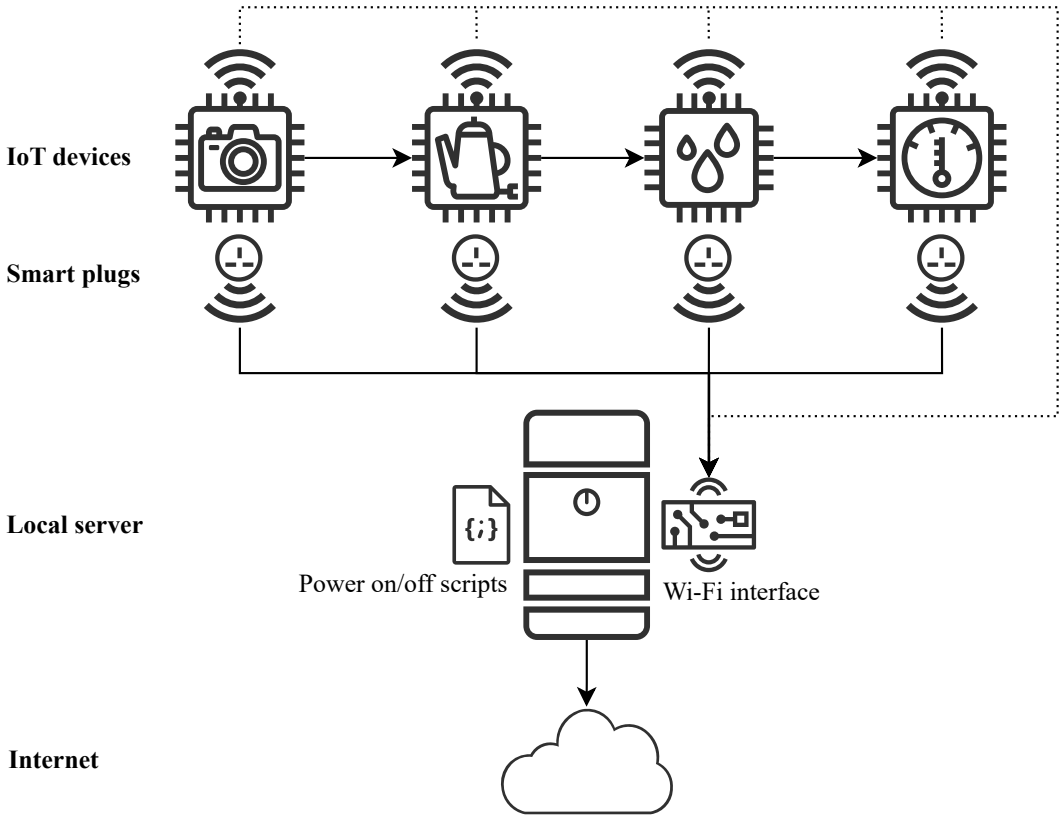


Figure 8.3: Testbed overview, with smart plugs and IoT devices all connected via Wi-Fi to a local orchestration server.

variety across several categories: Appliance (4), Baby Monitor (2), Camera (5), Doorbell (4), Hub (2), Light (6), Pet (2), Plug (1), Medical (1), Sensor (2) and Speaker (5). While they do not support DoH yet, we design a way to realistically simulate this protocol in Section 8.6.1.2.

Table 8.3: IoT devices present in the testbed.

Category	Device name
Appliance	Alexa Swan Kettle, Coffee Maker Lavazza, Cosori Air Fryer, Meross Garage Door
Baby Monitor	Boifun Baby, VTech Baby Camera
Camera	Arlo Camera Pro4, Blink Mini Camera, Google Nest Camera, SimpliCam, Wyze Cam Pan v2
Doorbell	Eufy Chime, Google Nest Doorbell, Reolink Doorbell, Ring Chime Pro
Hub	Aqara HubM2, Google Nest Hub
Light	Govee Strip Light, Lepro Bulb, Lifx Mini, NanoLeaf Triangles, Wiz Bulb, Yeelight Bulb
Medical	Withings Sleep Analyser
Pet	Furbo Dog Camera, Petsafe Feeder
Plug	Tapo Plug
Sensor	Netatmo Weather Station, Sensibo Sky Sensor
Speaker	Bose Speaker, Echodot4, Echodot5, Homepod, Sonos Speaker

This selection is interesting for two reasons. First, some categories of devices are relevant to eavesdropper. For instance, in case of burglary, knowing whether cameras are installed inside a house or if a dog is present could be valuable information. Second, we want to test if

similar devices (same manufacturer) exhibit similar behaviors and can still be distinguished. To study this question further, we also include two versions of the same device: Echodot4 and Echodot5.

8.4.2 DNS resolvers

While DoH is an official IETF standard, slight variations can be detected in the transmitted data depending on the resolver, for instance due to the HTTP header [157]. To broaden our approach and mitigate dependence on a single resolver implementation, we select 6 public DNS resolvers based on their widespread usage (default integration in Web browsers like Firefox and Chromium) and longevity [69]. These resolvers include Google, Cloudflare, Quad9, CleanBrowsing, NextDNS, and AdGuard.

8.5 Identifying IoT devices via DoH traffic

Building upon previous chapters, we describe our process for IoT device identification using DoH traffic. We first characterize and select relevant features, before explaining how they are extracted. Then, we employ machine learning to classify network traces based on said features, effectively identifying devices.

8.5.1 Features selection

In this section, we discuss which features can be used to train our models. Previous works utilize clear-text DNS messages and access the full domain name directly [203]. While this is impossible due to traffic encryption in DoH, metadata remain: packets `length` and Inter-Arrival Time (`IAT`) are still available to eavesdroppers. The intuition is that each device requests different domain names, both in terms of length and times, producing discriminative enough patterns to identify them.

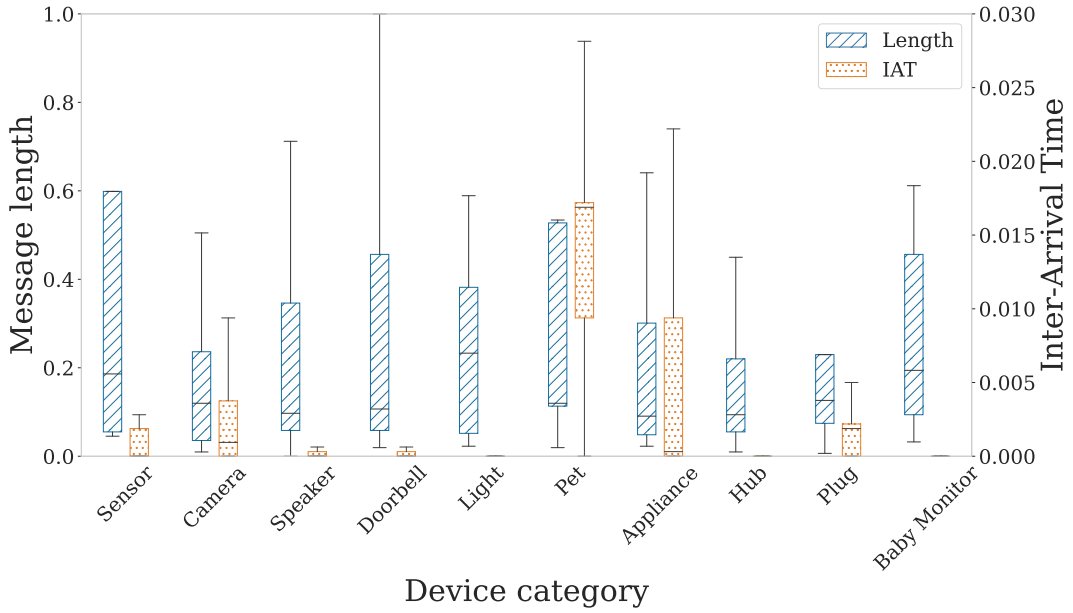


Figure 8.4: Discriminative behavior of `length` and `IAT`.

To detect possible discrepancies between devices, we normalize both message `length` and `IAT` of DNS messages for all devices in our dataset, and examine their respective distributions. Figure 8.4 illustrates these distributions by category for clarity, but later results are reported per device. Analysis of the message size reveals significant variability across device categories, hinting at possible length-based identification. Notably, plugs and hubs exhibit smaller message `length` compared to sensors. `IAT` values remain consistently low across most device categories, with pet-oriented devices and appliance showing higher values than speakers and

doorbells. Therefore, we opt to include both message `length` and `IAT` of DNS messages as features.

8.5.2 Features extraction

To extract both features, we collect the DNS traffic for each power cycle (switching on and off smart plugs). For each DNS request, we ignore irrelevant network data, including the generic TCP handshake and TLS negotiation, and extract the `length` from the TLS Application Data packets containing the DNS domain name and corresponding IP. Additionally, we compute the `IAT` in between DNS requests.

After extracting raw `length` and `IAT` values from the DoH traffic, we format them to a format exploitable by a machine learning model. Figure 8.5 illustrates the process. First, we save raw features as vectors: based on the maximum number of DNS requests observed during a power cycle, their number is low enough to be utilized directly (up to 30). We complete them with descriptive statistics already proved to be effective [117, 174, 208, 217, 19]: mean, variance, standard deviation, skewness, and kurtosis.

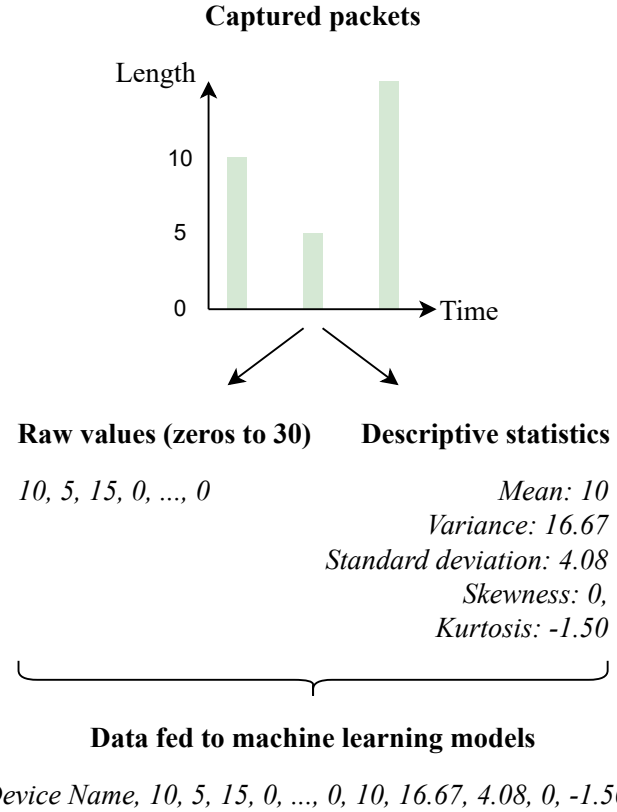


Figure 8.5: Extracting features from DNS traffic (here: length).

Contrary to our previous holistic approach on LoRaWAN fingerprinting (see Chapter 5), we utilize less complex data representations, excluding distributions and Markov chains. First, we empirically find that such complex implementations are not required to reach high accuracy. Second, an alternative use-case of IoT device identification is security, with related works focusing on *fast* identification on edge device (e.g. routers) [203]. In the case of Markov chains, we observe a median of 5 DNS requests per power cycle. Based on previous chapters, this is far fewer than the required number of messages to witness significant improvements (>50, see Figure 5.7).

8.6 Experimental methodology

In this section, we present how we collect DoH to build our comprehensive dataset and handle its inherent imbalance, as well as how the machine learning model is selected.

8.6.1 Dataset collection

To collect our dataset, we first generate DNS traffic by powering on and off devices and then convert clear-text DNS to DoH by replaying requests, effectively simulating DoH adoption for all IoT devices.

8.6.1.1 Power experiments

Previous studies have shown that IoT devices tend to generate a significant volume of DNS requests immediately after being powered on [203]. To optimize the capture of DNS requests, we perform a power cycle (switching on and off a device) for each experiment. Hence, each test run comprises 5 minutes of network traffic captured using the Mon(IoT)r tool [177], followed by a 2-minute timeout period to guarantee the device shuts down properly.

To ensure thorough coverage of different time periods for a longitudinal analysis, we repeat the entire process 50 times per day for 15 consecutive days. This extended duration guarantees a robust and comprehensive dataset, yielding a total of 25,500 on-off experiments.

8.6.1.2 DNS traffic encryption

IoT devices in our testbed currently lack support for DoH, requiring encrypting existing clear-text DNS requests. The number and variety of IoT devices in the testbed, as well as the unavailability of the source code, prevent us from updating all of their firmware to deploy encrypted DNS in a timely fashion.

Additionally, studying multiple mitigations techniques (cf. Section 8.7.6) increases the number of DoH requests corresponding to one original clear-text DNS query. The high number of requests may lead a resolver to block them, especially if they are not slightly rate limited. These limitations present challenges in implementing a proxy that can seamlessly convert clear-text requests to DoH on-the-fly.

Instead, we propose a two-steps process: 1) capture the clear-text traffic generated by IoT devices, and 2) replay DNS requests using DoH from the same vantage point. To ensure consistency, the DNS requests are replayed with the same timings they were originally sent, thereby minimizing discrepancies such as differences in DNS cache states. Unlike other components of our pipeline, replaying each DNS request is executed only once due to the high number of requests (i.e. the 15 days with 50 on-off periods are each replayed consecutively once).

8.6.2 Handling dataset imbalance

While some devices consistently generate dozens of DNS queries upon powering on, others send only a few requests. This behavior produces a dataset where some devices are more represented than others. As suggested in Section 3.2.3.2, we address the imbalance in the dataset by employing oversampling on the training set and keeping the evaluating set imbalanced to reflect real-world conditions. In our previous research, we employed SMOTE due to its high efficiency; however, to match the implementation of Thompson et al. [203], we have opted for random oversampling in this study. Following earlier chapters and to address dataset imbalance [31], we utilize *balanced accuracy*, calculating the average recall for each class.

8.6.3 Machine learning method selection

To benchmark against prior state-of-the-art IoT device identification using DNS traffic [203], we adopt a methodology that deviates slightly from the approaches outlined in previous chapters. This process allows us to 1) compare the various machine learning methods in the same environment as Thompson et al. [203], 2) extract the best possible results, while

3) avoiding test snooping and demonstrating that the method correctly generalizes against unseen data [31].

First, as presented in Chapter 4, we select various well-known multi-class machine learning methods: Random Forest, K-Nearest Neighbors, Complement Naive Bayes, Logistic Regression, Support Vector Classification (linear, one-vs-one, one-vs-the-rest). We complete them by re-implementing the solution deployed by Thompson et al. using a Neural Network [203], with the same layers.[4]

Second, in order to evaluate each method to its full potential and correctly compare with the state of the art where it is used, we select best performing hyperparameters via Halving Random Search Cross-Validation [133].

Following best practices highlighted in Section 3.2, we split the dataset into training and held-out data following a 80:20 ratio. Then, cross validation is used by Halving Random Search, further splitting the training dataset into 5 subsets of 80:20 training:testing to avoid overfitting on training data. After selecting the hyperparameters producing the highest balanced accuracy for each machine learning method and comparing them, we pick the best performing method itself. Finally, we train the model using relevant hyperparameters on the original 80% of the dataset, and validate it against the held-out 20%, producing the final results.

The machine learning pipeline is run 15 times, each initialized with a different seed for the PRNGs. Unless otherwise specified, we aggregate model performances across all DNS resolvers and a single day of replay. We ensure performance consistency across multiple days through manual confirmation and designate a random day as an illustrative baseline.

8.7 Results

In this section, we present the performance analysis of identification over the 34 IoT devices deployed in our testbed. We start by selecting the best performing machine learning method, and then proceed with actual identification.

8.7.1 Comparison of machine learning methods

Table 8.4 showcases the performance of all machine learning methods initially tested. More specifically, it reports the median value of averaged balanced accuracy over all cross-validations. Random Forest is the best performing method, reaching ∼0.97 balanced accuracy, well above results obtained via the setup of Thompson et al. based on a Neural Network [203] (∼0.89 only).

Table 8.4: Performance of machine learning methods during Halving Random Search Cross-Validation.

Machine learning method	Balanced accuracy
Random Forest	0.9684
Neural Network [203]	0.8931
SVC (linear)	0.8861
Logistic Regression	0.8485
K-Nearest Neighbors	0.8462
SVC (one-vs-the-rest)	0.8433
SVC (one-vs-one)	0.8392
Complement Naive Bayes	0.5816

For security purposes, rapid identification is crucial to promptly detect any unidentified or malfunctioning devices. Contrary to LoRaWAN, DNS traffic is generally more frequent, possibly generating multiple requests per seconds when an IoT device is switched on, and requiring a fast responding model. To estimate the time required for a single prediction when receiving a new message, we utilize the mean duration needed to compute balanced accuracy during the cross-validation process.

[4] We simply adapt the last layer to match the number of output classes (34).

In practice, the Random Forest method is faster than many other machine learning approaches, with a mean scoring time of ∼35 milliseconds.[5] Given that this time frame corresponds to *thousands* of predictions for the balanced accuracy, we infer that the prediction time is sufficiently brief for quick identification.

8.7.2 Device identification

Once the machine learning method is selected, we study its performance w.r.t. to actual identification in more details. Figure 8.6 displays the confusion matrix for all devices, where predictions are represented on the y-axis and actual devices on the x-axis. The matrix is normalized to accurately account for the varying occurrences of each device. Any values outside the diagonal indicate misclassifications.

Notably, there are very few false positives, resulting in a balanced accuracy of approximately 0.97 for all devices. We note that three devices distributed by the same company (Google's Nest Camera, Nest Doorbell, and Nest Hub) are correctly classified. Similarly, despite only differing in version (4 and 5), the two Echo Dot devices exhibit sufficiently distinct traffic patterns to enable accurate classification.

Figure 8.6: Devices identification confusion matrix.

While other results are presented as an average across all DNS resolvers, we also study the identification performance for each of them in Table 8.5. There is minimal variation observed among DNS resolvers, with stable values observed across all results. Said differently, changing DNS resolver has no impact on the performance of the attack.

[5]For comparison purposes with the state of the art, Neural Network require ∼520 milliseconds.

Table 8.5: Impact of the resolvers on performance.

Resolver	Balanced accuracy
AdGuard	0.9820
Google	0.9788
Quad9	0.9781
Cloudflare	0.9778
NextDNS	0.9768
CleanBrowsing	0.9765

8.7.3 Rapid identification

In this section, we virtually restrict the number of consecutive messages available to machine learning models to simulate a shorter traffic capture, either by absolute number (*only the first n DNS requests*), or time (*only the first s seconds*).

As depicted in Figure 8.7, even a single message is adequate for identifying devices with a balanced accuracy of approximately 0.92. Similar results are achieved when analyzing the time window of activity instead of focusing solely on the absolute number of DNS requests. We notice that only 1 second after switching a device on is sufficient to achieve the same balanced accuracy, enabling eavesdroppers to quickly assess the content of the targeted network. While such results may degrade in denser networks where DoH is more widely adopted, they hint at the possibility of rapid detection of misbehaving devices in defensive scenarios.

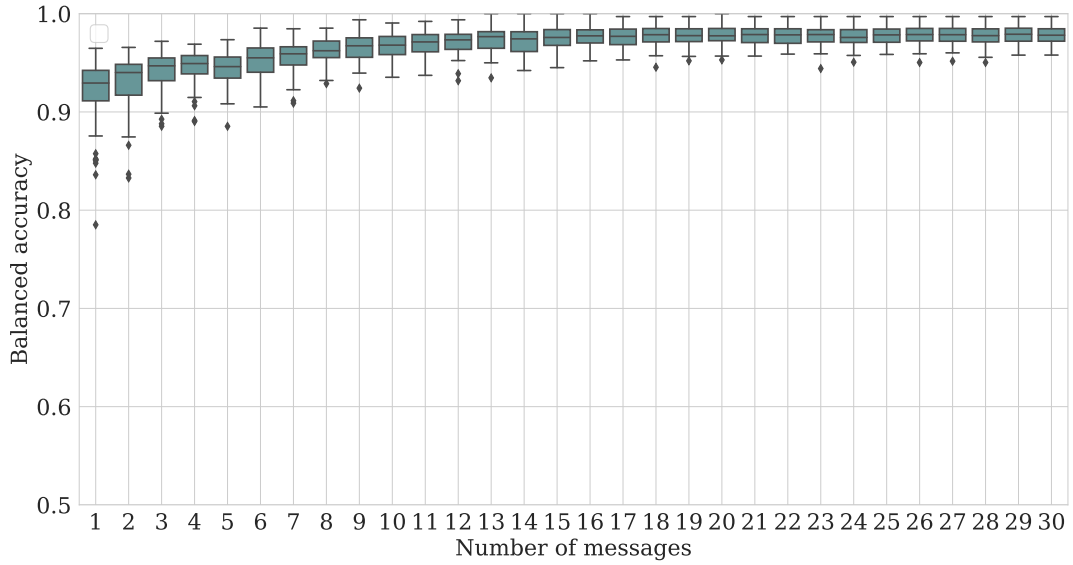


Figure 8.7: Evolution of the balanced accuracy based on the number DNS requests.

8.7.4 Importance of length vs IAT

Reaching a high balanced accuracy using only *one DNS request* raises doubts regarding the utility of the `IAT`, necessitating at least 2 messages. We find that length-based features alone yield a balanced accuracy of ∼0.98, compared to 0.79 for `IAT`-based features across all resolvers.

Such a steep decrease can be explained by multiple factors. First, as demonstrated in Section 8.5.2, `IAT` values tend to exhibit greater concentration compared to `length`. Second, the overall number of DNS requests is low (with a median of 5 per power cycle), resulting in a lower number of `IAT` values available. Third, `IAT` is susceptible to random variations, influenced by factors including DNS resolver response time and cache, as well as time required by a device to process incoming information.

8.7.5 Model degradation over time

In machine learning, a well-known phenomenon is the degradation of model performance over time. As features gradually drift away from their values at the time of training (e.g. due to software updates), the initial model produces increasingly worse results [120].

To assess this decline in performance, we evaluate the model using new, unseen data collected in the days following the reference training. Figure 8.8 illustrates that our model maintains a mean balanced accuracy of over ∼0.91 across all resolvers over a span of 15 days, with day 0 serving as the reference point.

This stability can be attributed to DoH traffic exhibiting consistent patterns over time. Optionally, it may indicate a robust model able to generalize against an evolving dataset.

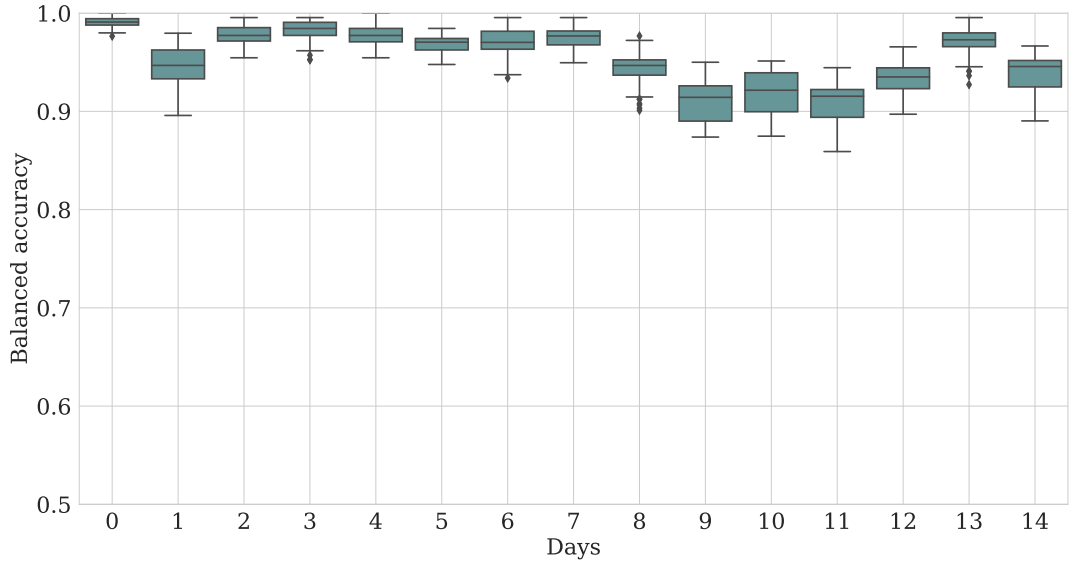


Figure 8.8: Performance evolution over 15 days, with day 0 used as reference.

8.7.6 Countermeasures

Due to the limited number of raw features in the machine learning process (`IAT` and `length`), options for mitigating our attack are restricted. In this section, we explore potential delays and grouping DNS requests, as well as padded queries.

8.7.6.1 Delay and reducing the number of DNS requests

It is difficult to introduce meaningful delays in consumer-grade IoT devices. First, the `IAT` is deeply tied with the application logic. For instance, a device may first contact an update server and then request the IP address of an application server to upload its data. Second, contrary to LoRaWAN devices characterized by their low throughput and limited reliance on time-sensitive functionalities, consumer-grade IoT devices generally rely on real-time applications and do not tolerate delays.

Hence, it would require a device-by-device analysis of the underlying application, which is not feasible. Additionally, deploying a network middleware to introduce delays could lead to certain requests timing out, possibly causing denial-of-service for devices.

Alternatively, DNS queries correspond to a single DNS resource, meaning there is a one-to-one relationship between domain names and observed packets on the wire. One would want to merge all requests into a single packet to obfuscate exactly which domains are queried. Although the original DNS RFC hints at multiple resources in one message via `QDCOUNT` [10], its under-specification (e.g. how to deal with partial resource availability) led to lack of support by resolvers. Multiple attempts to standardize this approach have been made [215, 216], but they failed gaining adoption. Other solutions include forwarding queries to neighbors in a mesh network [29], yet they lack implementation and real-world usage. Thus, we consider merging queries currently out of scope but interesting for future works.

8.7.6.2 Padded DNS

The DNS request `length` can be partially concealed through the addition of padding before encryption via the EDNS(0) Padding Option [144]. RFC 8467 advises DNS clients to select the "*closest multiple of 128 octets*" and "*multiple of 468 octets*" for servers [145]. Additionally, authors propose the "Random-Block-Length Padding" method, which randomly selects a block length (128, 256, etc.) to pad with. Alternative solutions such as padding up to the maximal length (MTU) or drawing from a known distribution (see Section 2.4.3.1) generate excessive overhead or are impractical in diverse IoT networks [53].

Evaluating padding strategies: We utilize two padding strategies by configuring the EDNS(0) option in the DNS query. The first strategy involves padding to the nearest 128-byte block, while the second involves randomly selecting the closest block size from the range of $[128, 256, 384, 512]$ bytes. While expanding this range is possible, we refrain from increasing the number of padding bytes to prevent reaching the MTU and generating unrealistic overhead.

Then, we replicate the replay methodology with padded queries and extract features accordingly. Finally, we train separate models for each resolver using *only the* ***`length`*** of messages with each padding strategy and compare their performance against models trained without any padding. Additionally, we explore a hypothetical scenario of perfect padding protection where the `length` does not leak any information. In such a case, potential attackers are limited to leveraging *only the* ***`IAT`*** for identification purposes.

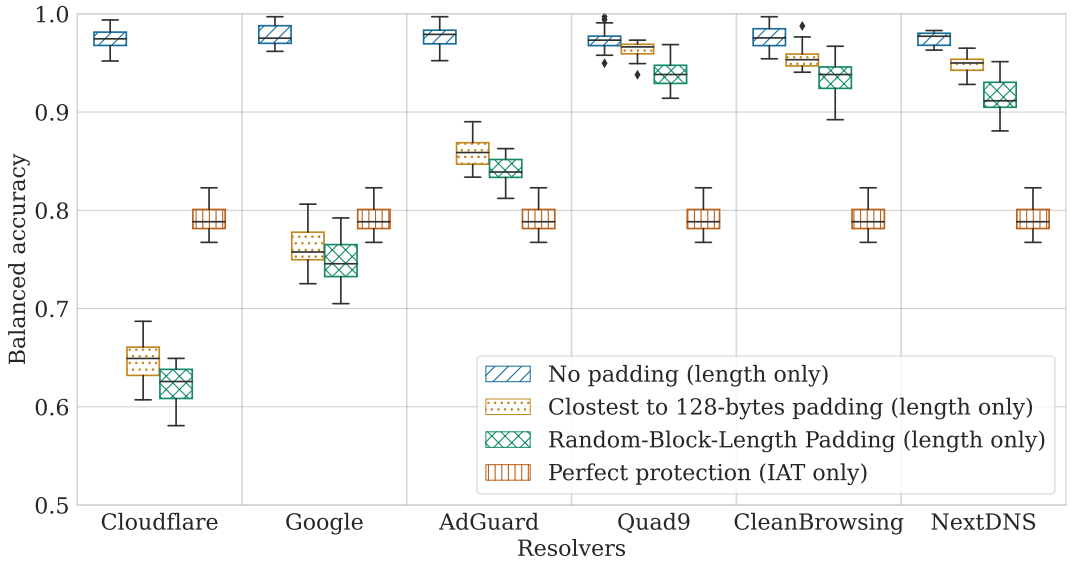


Figure 8.9: Impact of padding strategies on the performance, compared to no padding and perfect protection.

Figure 8.9 shows that balanced accuracy is significantly reduced for Cloudflare, Google, and AdGuard with both strategies. For instance, using 128-byte padding with Cloudflare results in a ∼33% decrease. Random-Block-Length Padding yields similar results while generating considerably more overhead (respectively 125% and 67% additional bytes on the wire).

Detecting misconfigurations: Through manual analysis of network traces, we find that the differences of behaviors between DNS resolvers stems from the failure of Quad9, CleanBrowsing, and NextDNS to respect padding standards, i.e. padded requests are not answered with padding. More precisely, we observe that each DNS resolver follow a different padding strategy upon receiving a padded query:

- Cloudflare and Google pad up to a fixed value the response to any padded request (respectively: exactly 707 encrypted bytes and around 850 encrypted bytes).
- AdGuard roughly matches the padding in the request.

- CleanBrowsing nearly never pads the answer, based on unknown heuristics.[6]
- NextDNS and Quad9 send an empty ENDS(0) option containing no padding, effectively adding a few bytes but maintaining a similar length to the original.

RFC 7830 states that "*responders MUST pad DNS responses when the respective DNS query included the Padding option*" [144], and RFC 8467 adds that resolvers "*SHOULD pad the corresponding response to a multiple of 468 octets*". Hence, multiple widely used public DNS servers (Quad9, CleanBrowsing, and NextDNS) do not respect the standard and put the privacy of their users at risk.

To confirm this failure as the cause of previous results, we train new models using only the `length`, separated based on the message direction: either from a) request messages (IoT devices to DNS resolvers), or b) answer messages (DNS resolvers to IoT devices). As expected, models trained solely on padded requests show a significant drop in balanced accuracy across all servers (from ∼0.97 to 0.64 on average). On the other hand, models trained on answers only are not affected when Quad9, CleanBrowsing, and NextDNS are used.

8.8 Discussion

In this section, we explore how padding could be deployed based on our threat model and technical requirements. We analyze the impact of such an attack on privacy and discuss the limitations of our approach.

8.8.1 Implementing padding mitigation

If an adversary gains direct access to the local network (e.g. via a compromised device or the router itself), the only viable course of action is deploying padding directly on the IoT devices. However, implementing countermeasures directly within IoT devices poses significant scale challenges, and waiting for widespread deployment or updates by manufacturers is impractical.[7]

When considering an eavesdropper outside the local network, we advocate for the deployment of a middlebox at the router level, acting as a local DNS resolver and effectively relaying clear-text DNS requests as padded DoH. Similarly to our experimental setup, a default DNS server is attributed to IoT devices via the *Domain Name Server* DHCP option [73]. For this approach to work, devices must accept DHCP-assigned DNS resolvers. In our experiments, only 3-12% of devices do not support it[8], with similar values reported in previous works [220]. If devices ignore DHCP-assigned DNS resolvers, we can resort to a man-in-the-middle solution intercepting clear-text DNS and forwarding them as padded DoH. In practice, such an approach could be deployed directly in routers, for example via an extension of open source software OpenWRT [7].

8.8.2 Implications for privacy and broader scope

We reliably identify IoT devices via DoH traffic and show that corresponding mitigations are conceivable for manufacturers and end-users. However, they rely on both DNS clients and resolvers to respect the standard, which does not seem to be currently the case.

Furthermore, this issue affects well-behaved clients, that correctly sends padded DoH requests and thus expect the best level of protection. We again highlight that DNS answers do not show any form of warning when padding is asked for but ultimately missing. Without detailed network trace analysis (e.g. via wireshark), or an automated check of DNS options client-side, it is impossible to detect this misconfiguration.

As DoH continues to be deployed in large-scale and critical user-facing applications like Web browsers, it becomes increasingly important to provide the best privacy protection possible, with clear mechanisms to inform users about any missing features.

[6] We believe it is related to the cache of the resolver. We could not confirm this theory, as CleanBrowsing did not reply to our emails.

[7] The difficulty in updating existing IoT devices does not diminish the global impact of implementing mitigations directly at the IoT manufacturer level, anticipating new deployments, or upgrading existing devices.

[8] We can not be certain of the DNS used by Google devices because our DHCP server advertises 8.8.8.8 and 8.8.4.4 as DNS resolvers.

8.8.3 Limitations and future work

An overview of the study's limitations reveals some considerations to be taken into account:

Dataset Completeness: Our dataset is limited to traffic originating from 34 devices, which may appear to be a small sample size. While previous works [203, 220] operate using fewer devices, hopping to reach internet scale in a lab is not feasible. However, we make sure to match the diversity of IoT setups, considering similar devices: same usage, or same manufacturer and different version. In both cases, our model is able to reliably identify each device, hinting at its robustness in broader deployments.

Traffic capture: We specifically focus on DNS traffic directly following a device power on, as it generates a high number of requests [203]. This supposes either that a) an attacker can monitor such an event, or b) the DNS traffic generated throughout the remainder of a device's lifespan is equally identifiable. While previous works have shown that long-term monitoring via DNS queries is possible [168], its possible adaptation to DoH require additional investigation.

Identifying DoH itself: Based on our threat model, we consider that DoH is already extracted from other encrypted traffic on port 443. This assumption is supported by prior research, which has demonstrated that distinguishing DoH from HTTPS is trivial via known IP addresses (e.g. 1.1.1.1 for Cloudflare, or 8.8.8.8 for Google) [53, 157]. Additionally, it can be achieved thanks to specific features, including shorter packet length and communication timing [69, 157, 155, 214].

Static IP addresses: Some IoT devices may not be affected by our attack as they do not rely on DNS and rather use hard-coded IP addresses. However, due to the maintenance challenges associated with static IP addresses, very few IoT devices adopt this approach.[9]

8.8.4 Responsible disclosure

The padding misconfiguration has been disclosed to the relevant DNS resolvers. Only Quad9 provided clarifications, stating there is currently no support for padding via the front-end they use, `dnsdist` [8]. 5 months after disclosure, we have yet to receive a comment from others; they still do not respect RFC 7830 when receiving a padded request.

8.9 Conclusion

While DoH brings significant privacy improvements in consumer-grade IoT devices, it does not guarantee a perfect protection. In this study, we show that the remaining metadata, including length and IAT, is enough to yield a 0.98 balanced accuracy when identifying 34 devices, across all studied DNS resolvers. Likewise, devices distributed by the same manufacturer, or even only differentiated by their version, are reliably classified.

Although other mitigations are too complex to implement, we find that padding is a straightforward and appropriate countermeasure. However, we discover that some DNS resolvers do not respect the corresponding standard and threaten their user's privacy.

More generally, we demonstrate that the trends observed in protocol operating under completely different constraints, LoRaWAN, also apply to DoH. We are able to re-adapt our methodology and successfully build an attack against privacy, while proposing similar countermeasures.

[9] We have yet to see a modern device behave like this in our dataset.

Part V

Conclusion

Chapter 9

Contributions and Future Directions

Contents

This chapter concludes the thesis by first presenting a summary of our contributions, before exploring both short and long term perspectives.

9.1 Contributions summary

In this section, we summarize how we explored each research question introduced in Chapter 1.

Q_1 **What are the privacy challenges of the LoRaWAN protocol?** We discovered that metadata inherently produced by the LoRaWAN protocol, including the size of encrypted messages and various header fields, timing between messages, as well as radio characteristics, poses significant privacy risks.

First, we showed that metadata generated during the join process allows an eavesdropper to effectively link the identity of a device with its activity. By leveraging machine learning techniques and various features from content, time, and radio domains, we were able to associate two theoretically unlinkable identifiers, the `DevEUI` and `DevAddr`. We also identified key features, such as the `FCnt`, that are crucial for reliable linking.

Second, we explored device identification through fingerprinting based on metadata and traffic patterns observed in sequences of messages. We compared the performance of various fingerprint representations by formatting features from the content, time, and radio domains, and demonstrated that combining all of them into a holistic representation yields the best results. Additionally, we studied multiple scenarios, including mobile devices and situations where the attacker has access to limited resources, such as a reduced number of listening stations. In doing so, we showed the robustness of our approach and its reliability in fingerprinting LoRaWAN devices. While this contribution is initially thought as a privacy assessment, it can also be leveraged in defensive scenarios such as network monitoring.

Q_2 **How can we address privacy challenges in LoRaWAN?** We found that previous privacy attacks based on metadata and traffic patterns can be thwarted via two approaches, with varying overhead costs.

First, we systematically analyzed the features exploited in previous attacks and designed methods to either completely obfuscate them or introduce noise to reduce their relevance for machine learning models. Our findings indicated that countermeasures applied in isolation rarely diminish the efficiency of attacks; therefore, mitigations need to be combined. Unfortunately, implementing these countermeasures leads to additional resource consumption,

such as increased transmitted bytes due to padding. Even with significant network impacts, feature-based mitigations alone proved insufficient to effectively thwart the attacks.

Second, due to the fact that the attacks were not significantly impacted and the overhead was considerable, we investigated the root cause. Using a stable identifier throughout the entire session allows an eavesdropper to group sequences of messages. Replacing the stable `DevAddr`, we designed a privacy-preserving pseudonym scheme for LoRaWAN. We first established desired properties for the scheme and excluded multiple existing solutions due to the LoRaWAN's resource constraints. We then further evaluated resolvable pseudonyms, inspired by Resolvable random Private Addresses in BLE, and sequential pseudonyms, based on Shroud by Greenstein et al. [90]. Our findings showed that sequential pseudonyms offer a compelling solution, generating limited extra energy consumption, ensuring a reduced number of collisions, and proving to be reliable. We concluded our study by expanding our analysis to other LPWAN protocols, stating that privacy-preserving pseudonyms could be both valuable and implementable.

Q_3 **Can this methodology be adapted to other IoT networks?** We found that our machine learning approach can be applied to different IoT contexts to assess privacy, with similarly high accuracy.

To explore this, we focused on DNS, a protocol widely deployed in consumer-grade IoT devices, specifically its encrypted version, DNS-over-HTTPS (DoH). Despite the differences in resource consumption and communication patterns compared to LoRaWAN, we demonstrated that devices can be identified based solely on packet lengths and timings. Our process shows high reliability regardless of the DNS resolver used or whether the devices were from the same manufacturer. Furthermore, we demonstrated rapid identification and model stability over time. Finally, we explored countermeasures and found that padding has a significant impact on mitigating our attack. Additionally, we discovered that some DNS resolvers do not respect padding specifications, compromising users' privacy.

To conclude, we showed that privacy attacks are possible against different IoT networks such as LoRaWAN and encrypted DNS, leveraging the same machine learning approach. While we explore some conceivable countermeasures, various protocol constraints, as well as the inherent difficulty to update both the standard and devices, suggest that further efforts are required for truly privacy-preserving communication in IoT networks.

As IoT devices become increasingly intertwined with human activities, privacy should be a default consideration during protocol design. This assessment should go beyond basic payload encryption and take metadata into account to fully protect the activity, location, and identity of end-users.

9.2 Perspectives

We first presented how our contributions are built upon years of prior research in the state of the art (see Chapter 2). In this section, we highlight how they could pave the way for new explorations, whether in the near or distant future.

9.2.1 Short-term

While we outlined some limitations and associated future works in previous sections, we now select five generic, short-term possible extensions of our work.

Fingerprinting LoRaWAN activity: Fingerprinting LoRaWAN devices has proven to be highly efficient in Chapter 5, but it can be further developed. Instead of focusing solely on *device identification*, we could perform *device activity inference* in a more generic and large-scale manner than what is currently presented in the literature [74, 131]. For instance, we could infer that a given network trace corresponds to a health monitoring device. This requires a correctly labeled dataset with known devices and their associated activities, which was virtually impossible with our deployment.

Fingerprinting distance evaluation: Although we compare various fingerprint *representations* in Chapter 5, we do not study the impact of *distances* between fingerprints, solely leveraging the Euclidean distance (see Section 2.3.5.2). A meta-analysis on the choice of distance metrics could provide valuable insights into their influence on identification accuracy (e.g. Jaccard similarity index [70, 162], cosine similarity [156], Kullback-Leibler divergence [217]).

Implementing data transmission pseudonyms for LoRaWAN: Our work on pseudonyms (Chapter 7) could be expanded. So far, we only have proposed a theoretical framework for privacy-preserving pseudonyms, with a prototypical proof-of-concept. An actual implementation in open source Network Servers (e.g. Chirpstack[1] or The Things Network's[2]) would greatly help the community embrace pseudonyms, eventually leading to its standardization and large-scale adoption.

Designing and implementing join pseudonyms for LoRaWAN: In Chapter 7, we focused solely on the `DevAddr`, ignoring the `DevEUI`. This completely static identifier is exposed to eavesdroppers and allows them to track End-Devices during their (re)join process. As outlined at in Section 7.9, the required tools are already available to design and implement another privacy-preserving pseudonym for this use-case. Doing so would also thwart the linking attack presented in Chapter 4.

Large-scale fingerprinting attack on encrypted DNS: The encrypted DNS identification attack presented in Chapter 8 is tied to the devices used to produce the dataset. By working conjointly with Internet Service Providers, it could be possible to collect a large dataset of consumer IoT-based IPv6 traffic[3]. This would enable an analysis of the complete attack pipeline: first extracting DoH from the HTTPS traffic, and then identifying devices, thereby enabling effective measurements of the actual impacts on privacy.

[1] https://github.com/chirpstack/chirpstack

[2] https://github.com/TheThingsNetwork/lorawan-stack

[3] This would require serious discussions with the relevant ethics committee and possible safeguards to respect user privacy.

9.2.2 Long-term

In addition to low-hanging fruits mainly targeting LoRaWAN, we propose four long-term perspectives related to machine learning, protocol design, and energy consumption.

Towards expert-agnostic machine learning for privacy: While we strive to extract all relevant features from analyzed network protocols, this process ultimately depends on our technical expertise. Like the previous engineers who designed the protocol and overlooked privacy aspects, we might miss significant features. Alternative machine learning-based approaches, such as those relying on genetic algorithms [18], can automatically select a subset of relevant features without needing expert knowledge. Taking it a step further, deep learning solutions can directly use raw network bytes to classify traffic [142].

With the increasing complexity of network protocols as well as feature extraction and associated machine learning models, it is crucial to prioritize *explainability* [179]. Allowing users to interpret model results is essential for drawing meaningful insights when analyzing technologies through the lens of privacy. Instead of relying on a set of pre-determined rules [57], a solution could be developed to automatically extract relevant features from raw network traces, regardless of the underlying protocol, and provide clear privacy insights to engineers and users (e.g., linkable identifiers). Contrary to our contributions, which focus on a single protocol, such an approach could lead to the design of a generic framework to analyze the privacy of network protocols through all their layers.

Guidance for privacy-preserving LPWAN protocols design: Our contributions are among many technical research projects that highlight the lack of consideration for privacy in the design of network protocols. For instance, other members of the LPWAN family, such as Sigfox and NB-IoT, are designed with similar constraints as LoRaWAN and similarly lack a foundational approach to privacy beyond basic encryption (see Section 7.8).

Efforts by legal bodies to protect privacy, notably via the GDPR or ePR, provide general principles, with interpretations and high-level examples detailed by the European Data Protection Board. They are completed by national-level guidelines, including the ones published by the CNIL, proposing recommendations for a given technology (e.g. how to conduct a Privacy Impact Assessment for IoT devices [61]). Finally, standardization bodies precisely describe network protocols, such as the IEEE 802.11 specifying physical and MAC Layers of wireless protocols (e.g. Wi-Fi), including their privacy aspects [63].

This structure can be impacted in multiple places to improve the state of network protocols' privacy. First, there is a lack of privacy guidelines specifically thought for LPWAN and other communication-constrained protocols. Producing such a document would both shed a light on these issue, and help the industry navigate them.

Second, while general legal requirements are usually taken into account when designing technical standards, concerns regarding metadata and stable identifiers remain out of scope. Collaborating with legal scholars on this specific privacy issue could be beneficial, potentially leading to updates in the legal framework and reinforcing the importance of privacy in protocol design. This approach can draw inspiration from research community feedback on legal texts, including the upcoming ePR [183].

Improved popularization: In addition to the significant volume of privacy research published over the past decades, already existing guidelines, and researchers generally willing to help[4], protocol designers have access to tools enabling formal proofs of security, such as Tamarin [148]. In parallel, the concept of *privacy by design* dates back to the early 2000s [125] and is now well known, yet it is evidently not widely implemented in practice.

We believe that the absence of privacy considerations during the design phase is not due to insufficient external resources, but rather to an overall lack of consideration, motivation, and incentives. Therefore, broader popularization of privacy research, including its motivations and implications, is a first crucial step for achieving a significant and lasting impact. To reach new audiences, we could participate in more industry-focused conferences (e.g. international

[4]For instance, TLS is specified partly with researchers validating its security proposals: `https://datatracker.ietf.org/group/tls/about/`. A similar approach could be implemented for privacy in other protocols.

ones regrouping numerous major companies or more local alternatives such as SIDO[5]) or even organize interventions with the public (e.g. *Fête de la Science*.[6]) To complete this bottom-up approach, large-scale awareness campaigns could be organized in collaboration with official structures, such as the CNIL[7], to inform citizens about privacy implications of the IoT and latest measures taken to protect their privacy.

Finally, successful software such as Signal shows that there is a demand for privacy-preserving solutions. Although private partnerships were seldom investigated during this thesis, joint work with companies on IoT devices with privacy as a selling point could be an interesting avenue to further popularize the concept.

Data minimization and energy consumption in network protocols: Working on constrained protocols and their privacy often involves complex and difficult choices between enhanced privacy properties and increased energy consumption. Finding the right balance is essential, as robust security measures can significantly drain the limited power resources of devices, particularly in IoT applications. Countermeasures to attacks presented in previous chapters generally involve trade-offs between privacy and energy consumption, given that LoRaWAN is already highly optimized for a low footprint. Finding the right trade-off requires additional research, selecting representative privacy metrics (see Section 2.4.1) and acceptability values for energy consumption.

However, this does not imply that alternative solutions for both privacy and resource-constrained network protocols are unattainable; future projects should strive to design such solutions. Since communication is usually the energy bottleneck, these approaches should be implemented over physical layers further optimized for energy consumption [182]. Moreover, lightweight cryptography [146, 202] would enable low-cost header encryption for metadata minimization and privacy-preserving pseudonyms.

Ultimately, only cross-disciplinary approaches involving standardization bodies, legal entities, sustainability experts, and privacy specialists can effectively address these challenges.

[5] https://www.sido-lyon.com/
[6] https://www.fetedelascience.fr/
[7] The CNIL already works with the press: https://cnil.fr/fr/espace-presse.

Bibliography

[1] Bind: Versatile, classic, complete name server software. https://www.isc.org/bind/. Accessed: 2024-05-22.

[2] Cloudflare: Dns over https. https://developers.cloudflare.com/1.1.1.1/encryption/dns-over-https/. Accessed: 2023-11-30.

[3] Dns over tls vs. dns over https. https://www.cloudflare.com/learning/dns/dns-over-tls/. Accessed: 2024-04-29.

[4] Google: Dns over https. https://developers.google.com/speed/public-dns/docs/doh/. Accessed: 2023-11-30.

[5] IEEE Registration Authority. https://standards.ieee.org/products-programs/regauth/. Accessed: 2024-02-13.

[6] Loi n° 78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés.

[7] Openwrt. https://openwrt.org/docs/guide-user/services/dns/doh_dnsmasq_https-dns-proxy. Accessed: 2024-05-08.

[8] Quad9 support edns(0) padding. https://github.com/PowerDNS/pdns/issues/10018. Accessed: 2023-11-30.

[9] Unbound. https://www.nlnetlabs.nl/projects/unbound/about/. Accessed: 2024-05-22.

[10] Domain names - implementation and specification. Request for Comments RFC 1035, Internet Engineering Task Force, November 1987.

[11] 3GPP. Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; Numbering, addressing and identification (3GPP TS 23.003 version 16.12.0 Release 16), 2023.

[12] 3GPP. Technical Specification Group Services and System Aspects; 3GPP System Architecture Evolution (SAE); Security architecture (Release 18), March 2024.

[13] Jan Aalmoes. Mass function of the number of coin flips to get m consecutive heads. 2024.

[14] Ahmed Abdelghany, Bernard Uguen, Christophe Moy, and Dominique Lemur. On Superior Reliability of Effective Signal Power versus RSSI in LoRaWAN. In *2021 28th International Conference on Telecommunications (ICT)*, pages 1–5, London, United Kingdom, June 2021. IEEE.

[15] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and Selcuk Uluagac. Peek-a-boo: i see your smart home activities, even encrypted! In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 207–218, Linz Austria, July 2020. ACM.

[16] Kuburat Oyeranti Adefemi Alimi, Khmaies Ouahada, Adnan M. Abu-Mahfouz, and Suvendi Rimer. A Survey on the Security of Low Power Wide Area Networks: Threats, Challenges, and Potential Solutions. *Sensors*, 20(20):5800, January 2020.

[17] Michiel Aernouts, Rafael Berkvens, Koen Van Vlaenderen, and Maarten Weyn. Sigfox and LoRaWAN Datasets for Fingerprint Localization in Large Urban and Rural Areas. *Data*, 3(2):13, June 2018.

[18] Ahmet Aksoy and Mehmet Hadi Gunes. Automated IoT Device Identification using Network Traffic. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2019.

[19] Hidayet Aksu, A. Selcuk Uluagac, and Elizabeth S. Bentley. Identification of Wearable Devices with Bluetooth. *IEEE Transactions on Sustainable Computing*, 6(2):221–230, April 2021.

[20] Wahhab Albazrqaoe, Jun Huang, and Guoliang Xing. A practical Bluetooth traffic sniffing system: Design, implementation, and countermeasure. *IEEE/ACM Transactions on Networking*, 27(1):71–84, 2018.

[21] LoRa Alliance. RP2-1.0.3 LoRaWAN® Regional Parameters, 2021.

[22] Ahmed Alshehri, Jacob Granley, and Chuan Yue. Attacking and protecting tunneled traffic of smart home devices. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, pages 259–270, 2020.

[23] Joseph Henry Anajemba, Tang Yue, Celestine Iwendi, Pushpita Chatterjee, Desire Ngabo, and Waleed S. Alnumay. A secure multiuser privacy technique for wireless IoT networks using stochastic privacy optimization. *IEEE Internet of Things Journal*, 9(4):2566–2577, 2021.

[24] Lucas Ancian and Mathieu Cunche. Re-identifying addresses in LoRaWAN networks. page 28, 2020.

[25] Mahnoor Anjum, Muhammad Abdullah Khan, Syed Ali Hassan, Aamir Mahmood, and Mikael Gidlund. Analysis of RSSI Fingerprinting in LoRa Networks. page 7, 2019.

[26] ANSSI. Mécanismes cryptographiques | ANSSI. https://cyber.gouv.fr/publications/mecanismes-cryptographiques, 2021.

[27] Noah Apthorpe, Danny Yuxing Huang, Dillon Reisman, Arvind Narayanan, and Nick Feamster. Keeping the smart home private with smart (er) iot traffic shaping. *arXiv preprint arXiv:1812.00955*, 2018.

[28] Chrisil Arackaparambil, Sergey Bratus, Anna Shubina, and David Kotz. On the reliability of wireless fingerprinting using clock skews. In *Proceedings of the Third ACM Conference on Wireless Network Security*, pages 169–174, Hoboken New Jersey USA, March 2010. ACM.

[29] Oscar Arana, Hector Benítez-Pérez, Javier Gomez, and Miguel Lopez-Guerrero. Never Query Alone: A distributed strategy to protect Internet users from DNS fingerprinting attacks. *Computer Networks*, 199:108445, November 2021.

[30] F. Armknecht, J. Girao, A. Matos, and R. L. Aguiar. Who Said That? Privacy at Link Layer. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pages 2521–2525, Anchorage, AK, USA, 2007. IEEE.

[31] Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. Dos and Don'ts of Machine Learning in Computer Security. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3971–3988, 2022.

[32] Muhammad Rizwan Asghar, György Dán, Daniele Miorandi, and Imrich Chlamtac. Smart meter data privacy: A survey. *IEEE Communications Surveys & Tutorials*, 19(4):2820–2835, 2017.

[33] Tomer Ashur, Jeroen Delvaux, Sanghan Lee, Pieter Maene, Eduard Marin, Svetla Nikova, Oscar Reparaz, Vladimir Rožić, Dave Singelée, Bohan Yang, and Bart Preneel. A privacy-preserving device tracking system using a low-power wide-area network: CANS 2017. *Cryptology and Network Security - 16th International Conference, CANS 2017, Revised Selected Papers*, pages 347–369, January 2018.

[34] Eyuel D. Ayele, Nirvana Meratnia, and Paul JM Havinga. Towards a new opportunistic IoT network architecture for wildlife monitoring system. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE, 2018.

[35] Leonardo Babun, Hidayet Aksu, Lucas Ryan, Kemal Akkaya, Elizabeth S. Bentley, and A. Selcuk Uluagac. Z-IoT: Passive Device-class Fingerprinting of ZigBee and Z-Wave IoT Devices. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–7, June 2020.

[36] Solveig Badillo, Balazs Banfai, Fabian Birzele, Iakov I. Davydov, Lucy Hutchinson, Tony Kam-Thong, Juliane Siebourg-Polster, Bernhard Steiert, and Jitao David Zhang. An Introduction to Machine Learning. *Clinical Pharmacology & Therapeutics*, 107(4):871–885, April 2020.

[37] Miloud Bagaa, Tarik Taleb, Jorge Bernal Bernabe, and Antonio Skarmeta. A Machine Learning Security Framework for Iot Systems. *IEEE Access*, 8:114066–114077, 2020.

[38] Tariq Al Balushi, Ayoub Al Hosni, Hashim Al Theeb Ba Omar, and Dawood Al Abri. A LoRaWAN-based Camel Crossing Alert and Tracking System. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, volume 1, pages 1035–1040, July 2019.

[39] Elaine Barker. Recommendation for Key Management: Part 1 – General. Technical Report NIST Special Publication (SP) 800-57 Part 1 Rev. 5, National Institute of Standards and Technology, May 2020.

[40] David Barrera, Glenn Wurster, and P C van Oorschot. Back to the Future: Revisiting IPv6 Privacy Extensions. 2011.

[41] Arup Barua, Md Abdullah Al Alamin, Md Shohrab Hossain, and Ekram Hossain. Security and privacy threats for bluetooth low energy in iot and wearable devices: A comprehensive survey. *IEEE Open Journal of the Communications Society*, 3:251–281, 2022.

[42] Gustavo EAPA Batista, Ana LC Bazzan, and Maria Carolina Monard. Balancing training data for automated annotation of keywords: A case study. *Wob*, 3:10–8, 2003.

[43] E Bäumker, A Miguel Garcia, and P Woias. Minimizing power consumption of LoRa ® and LoRaWAN for low-power wireless sensor nodes. *Journal of Physics: Conference Series*, 1407(1):012092, November 2019.

[44] Johannes K Becker, David Li, and David Starobinski. Tracking Anonymized Bluetooth Devices. *Proceedings on Privacy Enhancing Technologies*, 2019(3):50–65, July 2019.

[45] John Bellardo and Stefan Savage. 802.11 {Denial-of-Service} attacks: Real vulnerabilities and practical solutions. In *12th USENIX Security Symposium (USENIX Security 03)*, 2003.

[46] Eleanor Birrell, Jay Rodolitz, Angel Ding, Jenna Lee, Emily McReynolds, Jevan Hutson, and Ada Lerner. SoK: Technical Implementation and Human Impact of Internet Privacy Regulations. In *45th IEEE Symposium on Security and Privacy*, San Francisco, CA, USA, May 2024. IEEE.

[47] Bram Bonne, Arno Barzan, Peter Quax, and Wim Lamotte. WiFiPi: Involuntary tracking of visitors at mass events. In *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pages 1–6, Madrid, June 2013. IEEE.

[48] Martin Bor and Utz Roedig. LoRa transmission parameter selection. In *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 27–34. IEEE, 2017.

[49] Carsten Bormann, Mehmet Ersue, and Ari Keränen. Terminology for Constrained-Node Networks. Request for Comments RFC 7228, Internet Engineering Task Force, May 2014.

[50] Taoufik Bouguera, Jean-François Diouris, Jean-Jacques Chaillout, Randa Jaouadi, and Guillaume Andrieux. Energy consumption model for sensor nodes based on LoRa and LoRaWAN. *Sensors*, 18(7):2104, 2018.

[51] Danah Boyd. *It's Complicated: The Social Lives of Networked Teens*. Yale University Press, 2014.

[52] Peter Brown, Collectif, Georges Duby, and Philippe Ariès. *Histoire de la vie privée, tome 1 : De L'Empire romain à l'an mil*. Seuil, October 1999.

[53] Jonas Bushart and Christian Rossow. Padding ain't enough: Assessing the privacy guarantees of encrypted DNS. 2020.

[54] Lluís Casals, Bernat Mir, Rafael Vidal, and Carles Gomez. Modeling the energy performance of LoRaWAN. *Sensors*, 17(10):2364, 2017.

[55] Guillaume Celosia and Mathieu Cunche. Saving private addresses: An analysis of privacy issues in the bluetooth-low-energy advertising mechanism. In *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 444–453, Houston Texas USA, November 2019. ACM.

[56] Guillaume Celosia and Mathieu Cunche. Discontinued privacy: Personal data leaks in apple bluetooth-low-energy continuity protocols. *Proceedings on Privacy Enhancing Technologies*, 2020:26–46, 2020.

[57] Guillaume Celosia and Mathieu Cunche. Valkyrie: A generic framework for verifying privacy provisions in wireless networks. In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 278–283, Linz Austria, July 2020. ACM.

[58] Bharat S. Chaudhari, Marco Zennaro, and Suresh Borkar. LPWAN Technologies: Emerging Application Characteristics, Requirements, and Design Considerations. *Future Internet*, 12(3):46, March 2020.

[59] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[60] Roger Clarke. Introduction to dataveillance and information privacy and definitions of terms. *www. anu. edu. au/people/Roger. Clarke/DV/Intro. html*, 1997.

[61] CNIL. PIA, application to connected objects. 2018.

[62] European Commission. Proposal for a Regulation on Privacy and Electronic Communications. https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-privacy-and-electronic-communications, January 2017.

[63] LAN/MAN Standards Committee. IEEE Recommended Practice for Privacy Considerations for IEEE 802(R) Technologies, 2020.

[64] LoRa Alliance Technical Committee. LoRaWAN® Regional Parameters v1.1rA. https://resources.lora-alliance.org/technical-specifications/lorawan-regional-parameters-v1-1ra, October 2017.

[65] LoRa Alliance Technical Committee. LoRaWAN® Specification v1.1. https://lora-alliance.org/resource_hub/lorawan-specification-v1-1/, 2017.

[66] Bogdan Copos, Karl Levitt, Matt Bishop, and Jeff Rowe. Is Anybody Home? Inferring Activity From Smart Home Network Traffic. In *2016 IEEE Security and Privacy Workshops (SPW)*, pages 245–251, San Jose, CA, May 2016. IEEE.

[67] Semtech Corporation. Semtech LoRa Technology Overview | Semtech. https://www.semtech.com/lora.

[68] Semtech Corporation. LoRaWAN – simple rate adaptation recommended algorithm, 2016.

[69] Levente Csikor, Himanshu Singh, Min Suk Kang, and Dinil Mon Divakaran. Privacy of DNS-over-HTTPS: Requiem for a Dream? In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 252–271, Vienna, Austria, September 2021. IEEE.

[70] Mathieu Cunche, Mohamed Ali Kaafar, and Roksana Boreli. I know who you will meet this evening! Linking wireless devices using Wi-Fi probe requests. In *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–9, San Francisco, CA, USA, June 2012. IEEE.

[71] Syed Muhammad Danish, Arfa Nasir, Hassaan Khaliq Qureshi, Ayesha Binte Ashfaq, Shahid Mumtaz, and Jonathan Rodriguez. Network intrusion detection system for jamming attack in LoRaWAN join procedure. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[72] Aveek K. Das, Parth H. Pathak, Chen-Nee Chuah, and Prasant Mohapatra. Uncovering Privacy Leakage in BLE Network Traffic of Wearable Fitness Trackers. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*, pages 99–104, St. Augustine Florida USA, February 2016. ACM.

[73] Ralph Droms and Steve Alexander. DHCP Options and BOOTP Vendor Extensions. Request for Comments RFC 2132, Internet Engineering Task Force, March 1997.

[74] Lea Dujić Rodić, Toni Perkovic, Maja Skiljo, and Petar Solic. Privacy Leakage of Lorawan Smart Parking Occupancy Sensors, March 2022.

[75] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25*, pages 486–503. Springer, 2006.

[76] Kevin P. Dyer, Scott E. Coull, Thomas Ristenpart, and Thomas Shrimpton. Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail. In *2012 IEEE Symposium on Security and Privacy*, pages 332–346. IEEE, 2012.

[77] Aviv Engelberg and Avishai Wool. Classification of Encrypted IoT Traffic Despite Padding and Shaping, October 2021.

[78] Sinem Coleri Ergen, Huseyin Serhat Tetikol, Mehmet Kontik, Raffi Sevlian, Ram Rajagopal, and Pravin Varaiya. RSSI-fingerprinting-based mobile phone localization with route constraints. *IEEE Transactions on Vehicular Technology*, 63(1):423–428, 2013.

[79] Bernat Carbonés Fargas and Martin Nordal Petersen. GPS-free geolocation using LoRa in low-power WANs. In *2017 Global Internet of Things Summit (Giots)*, pages 1–6. IEEE, 2017.

[80] Ellis Fenske, Dane Brown, Jeremy Martin, Travis Mayberry, Peter Ryan, and Erik Rye. Three years later: A study of mac address randomization in mobile devices and when it succeeds. *Proceedings on Privacy Enhancing Technologies*, 2021.

[81] Mohamed Amine Ferrag, Lei Shu, Xing Yang, Abdelouahid Derhab, and Leandros Maglaras. Security and Privacy for Green IoT-Based Agriculture: Review, Blockchain Solutions, and Challenges. *IEEE Access*, 8:32031–32053, 2020.

[82] Rachel L. Finn, David Wright, and Michael Friedewald. Seven Types of Privacy. In Serge Gutwirth, Ronald Leenes, Paul de Hert, and Yves Poullet, editors, *European Data Protection: Coming of Age*, pages 3–32. Springer Netherlands, Dordrecht, 2013.

[83] Luciano Floridi. The Informational Nature of Personal Identity. *Minds and Machines*, 21(4):549–566, November 2011.

[84] Julien Freudiger, Mohammad Hossein Manshaei, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. On the Age of Pseudonyms in Mobile Ad Hoc Networks. In *2010 Proceedings IEEE INFOCOM*, pages 1–9, March 2010.

[85] Harald T. Friis. A note on a simple transmission formula. *Proceedings of the IRE*, 34(5):254–256, 1946.

[86] Flavio D. Garcia, David Oswald, Timo Kasper, and Pierre Pavlidès. Lock it and still lose it—on the ({In) Security} of automotive remote keyless entry systems. In *25th USENIX Security Symposium (USENIX Security 16)*, 2016.

[87] Paul Ginsparg. How many coin flips on average does it take to get n consecutive heads. *https://courses. cit. cornell. edu/info2950_2012sp/mh. pdf*, 2005.

[88] Google. MAC Randomization Behavior, Android documentation. https://source.android.com/docs/core/connect/wifi-mac-randomization-behavior, 2024.

[89] Ben Greenstein, Ramakrishna Gummadi, Jeffrey Pang, Mike Y Chen, Tadayoshi Kohno, Srinivasan Seshan, and David Wetherall. Can Ferris Bueller Still Have His Day Off? Protecting Privacy in the Wireless Era. page 6, 2007.

[90] Ben Greenstein, Damon McCoy, Jeffrey Pang, Tadayoshi Kohno, Srinivasan Seshan, and David Wetherall. Improving wireless privacy with an identifier-free link layer protocol. In *Proceeding of the 6th International Conference on Mobile Systems, Applications, and Services - MobiSys '08*, page 40, Breckenridge, CO, USA, 2008. ACM Press.

[91] Ulrich Greveler, Peter Glösekötterz, Benjamin Justusy, and Dennis Loehr. Multimedia content identification through smart meter power usage profiles. In *Proceedings of the International Conference on Information and Knowledge Engineering (IKE)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2012.

[92] Core Specification Working Group. Bluetooth® Core Specification 5.4. https://www.bluetooth.com/specifications/specs/core-specification-5-4/, February 2023.

[93] Saikat Guha and Paul Francis. Identity Trail: Covert Surveillance Using DNS. In Nikita Borisov and Philippe Golle, editors, *Privacy Enhancing Technologies*, volume 4776, pages 153–166. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[94] Serge Gutwirth. *Privacy and the Information Age*. Rowman & Littlefield, 2002.

[95] Dalton A. Hahn, Arslan Munir, and Vahid Behzadan. Security and Privacy Issues in Intelligent Transportation Systems: Classification and Challenges. *IEEE Intell. Transp. Syst. Mag.*, 13(1):181–196, 2021.

[96] Jetmir Haxhibeqiri, Eli De Poorter, Ingrid Moerman, and Jeroen Hoebeke. A Survey of LoRaWAN for IoT: From Technology to Application. *Sensors*, 18(11):3995, November 2018.

[97] Inc Helium Systems. Buy An Organizationally Unique Identifier | Helium Documentation. https://docs.helium.com/iot/run-an-lns/buy-an-oui/#multiplexing-with-devaddrs, 2024.

[98] Noelia Hernández, Manuel Ocaña, Jose M. Alonso, and Euntai Kim. Continuous space estimation: Increasing WiFi-based indoor localization resolution without increasing the site-survey effort. *Sensors*, 17(1):147, 2017.

[99] Frank Hessel, Lars Almon, and Flor Álvarez. ChirpOTLE: A Framework for Practical LoRaWAN Security Evaluation. In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 306–316, July 2020.

[100] Frank Hessel, Lars Almon, and Matthias Hollick. LoRaWAN Security: An Evolvable Survey on Vulnerabilities, Attacks and their Systematic Mitigation. *ACM Transactions on Sensor Networks*, 18(4):1–55, November 2022.

[101] Paul E. Hoffman and Patrick McManus. DNS Queries over HTTPS (DoH). Request for Comments RFC 8484, Internet Engineering Task Force, October 2018.

[102] Zi Hu, Liang Zhu, John Heidemann, Allison Mankin, Duane Wessels, and Paul E. Hoffman. Specification for DNS over Transport Layer Security (TLS). Request for Comments RFC 7858, Internet Engineering Task Force, May 2016.

[103] Anna Huang. Similarity measures for text document clustering. In *Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand*, volume 4, pages 9–56, 2008.

[104] Leping Huang, Kanta Matsuura, Hiroshi Yamane, and Kaoru Sezaki. Enhancing wireless location privacy using silent period. In *IEEE Wireless Communications and Networking Conference, 2005*, volume 2, pages 1187–1192. IEEE, 2005.

[105] Christian Huitema, Sara Dickinson, and Allison Mankin. DNS over Dedicated QUIC Connections. Request for Comments RFC 9250, Internet Engineering Task Force, May 2022.

[106] The Things Industries. Building a gateway with Raspberry Pi and IC880A. https://www.thethingsindustries.com/docs/gateways/models/raspberry-pi/, 2022.

[107] The Things Industries. NetID and DevAddr Prefix Assignments. https://www.thethingsnetwork.org/docs/lorawan/prefix-assignments/, 2024.

[108] Information Sciences Institue University of Southern California. Internet Protocol. Request for Comments RFC 791, Internet Engineering Task Force, September 1981.

[109] Development Ingenu. An Idiot's Guide to RPMA Development – Part 2, August 2016.

[110] European Telecommunications Standards Institute. ETSI EN 300 220-2, 2018.

[111] Taher Issoufaly and Pierre Ugo Tournoux. BLEB: Bluetooth Low Energy Botnet for large scale individual tracking. In *2017 1st International Conference on Next Generation Computing Applications (NextComp)*, pages 115–120, Mauritius, July 2017. IEEE.

[112] Arshad Jhumka, Matthew Leeke, and Sambid Shrestha. On the use of fake sources for source location privacy: Trade-offs between energy and privacy. *The Computer Journal*, 54(6):860–874, 2011.

[113] Yu Jiang, Linning Peng, Aiqun Hu, Sheng Wang, Yi Huang, and Lu Zhang. Physical layer identification of LoRa devices using constellation trace figure. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):223, December 2019.

[114] C Joshitha, P Kanakaraja, Mallela Divya Bhavani, Yerramsetti Naga Venkata Raman, and Tadepalli Sravani. LoRaWAN based Cattle Monitoring Smart System. In *2021 7th International Conference on Electrical Energy Systems (ICEES)*, pages 548–552, February 2021.

[115] Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):1–21, 2012.

[116] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154, 2017.

[117] Randall W. Klein, Michael A. Temple, and Michael J. Mendenhall. Application of wavelet-based RF fingerprinting to enhance wireless network security. *Journal of Communications and Networks*, 11(6):544–555, December 2009.

[118] Kevin H. Knuth. Optimal Data-Based Binning for Histograms, September 2013.

[119] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. page 8, 1995.

[120] Roman Kolcun, Diana Andreea Popescu, Vadim Safronov, Poonam Yadav, Anna Maria Mandalari, Richard Mortier, and Hamed Haddadi. Revisiting IoT Device Identification, July 2021.

[121] Maciej Korczyński and Andrzej Duda. Markov chain fingerprinting to classify encrypted traffic. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 781–789, April 2014.

[122] Srinivas Krishnan and Fabian Monrose. DNS prefetching and its privacy implications: When good things go bad. In *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More*, pages 10–10, 2010.

[123] Jacob Leon Kröger, Philip Raschke, and Towhidur Rahman Bhuiyan. Privacy implications of accelerometer data: A review of possible inferences. In *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy - ICCSP '19*, pages 81–87, Kuala Lumpur, Malaysia, 2019. ACM Press.

[124] A. R. Kulaib, R. M. Shubair, M. A. Al-Qutayri, and Jason W. P. Ng. An overview of localization techniques for Wireless Sensor Networks. In *2011 International Conference on Innovations in Information Technology*, pages 167–172, Abu Dhabi, United Arab Emirates, April 2011. IEEE.

[125] Marc Langheinrich. Privacy by Design — Principles of Privacy-Aware Ubiquitous Systems. In Gerhard Goos, Juris Hartmanis, Jan Van Leeuwen, Gregory D. Abowd, Barry Brumitt, and Steven Shafer, editors, *Ubicomp 2001: Ubiquitous Computing*, volume 2201, pages 273–291. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[126] Arash Habibi Lashkari, Mir Mohammad Seyed Danesh, and Behrang Samadi. A survey on wireless security protocols (WEP, WPA and WPA2/802.11 i). In *2009 2nd IEEE International Conference on Computer Science and Information Technology*, pages 48–52. IEEE, 2009.

[127] Alexandru Lavric and Valentin Popa. A LoRaWAN: Long range wide area networks study. In *2017 International Conference on Electromechanical and Power Systems (SIELMEN)*, pages 417–420, October 2017.

[128] Huang-Chen Lee and Kai-Hsiang Ke. Monitoring of Large-Area IoT Sensors Using a LoRa Wireless Mesh Network System: Design and Evaluation. *IEEE Transactions on Instrumentation and Measurement*, 67(9):2177–2187, September 2018.

[129] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

[130] Martine S. Lenders, Christian Amsüss, Cenk Gündogan, Marcin Nawrocki, Thomas C. Schmidt, and Matthias Wählisch. Securing Name Resolution in the IoT: DNS over CoAP. *Proceedings of the ACM on Networking*, 1(CoNEXT2):1–25, September 2023.

[131] Patrick Leu, Ivan Puddu, Aanjhan Ranganathan, and Srdjan Čapkun. I Send, Therefore I Leak: Information Leakage in Low-Power Wide Area Networks. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pages 23–33, Stockholm Sweden, June 2018. ACM.

[132] Baoli Li and Liping Han. Distance Weighted Cosine Similarity Measure for Text Classification. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Hujun Yin, Ke Tang, Yang Gao, Frank Klawonn, Minho Lee, Thomas Weise, Bin Li, and Xin Yao, editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2013*, volume 8206, pages 611–618. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[133] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization, June 2018.

[134] Alexandre Liaut. Sigfox Radio Specifications v1.7. 2023.

[135] Fen Liu, Jing Liu, Yuqing Yin, Wenhan Wang, Donghai Hu, Pengpeng Chen, and Qiang Niu. Survey on WiFi-based indoor positioning techniques. *IET Communications*, 14(9):1372–1383, June 2020.

[136] Qian Liu, Yanyan Mu, Jin Zhao, Jingxia Feng, and Bin Wang. Characterizing packet loss in city-scale LoRaWAN deployment: Analysis and implications. In *2020 IFIP Networking Conference (Networking)*, pages 704–712. IEEE, 2020.

[137] Yan-Ting Liu, Bo-Yi Lin, Xiao-Feng Yue, Zong-Xuan Cai, Zi-Xian Yang, Wei-Hong Liu, Song-Yi Huang, Jun-Lin Lu, Jing-Wen Peng, and Jen-Yeu Chen. A solar powered long range real-time water quality monitoring system by LoRaWAN. In *2018 27th Wireless and Optical Communication Conference (WOCC)*, pages 1–2, April 2018.

[138] Zishan Liu, Lin Zhang, Wei Ni, and Iain B. Collings. Uncoordinated Pseudonym Changes for Privacy Preserving in Distributed Networks. *IEEE Transactions on Mobile Computing*, 19(6):1465–1477, June 2020.

[139] Javier Lopez, Ruben Rios, Feng Bao, and Guilin Wang. Evolving privacy: From sensors to the Internet of Things. *Future Generation Computer Systems*, 75:46–57, October 2017.

[140] Slim Loukil, Lamia Chaari Fourati, Anand Nayyar, and K.-W.-A. Chee. Analysis of LoRaWAN 1.0 and 1.1 protocols security mechanisms. *Sensors*, 22(10):3717, 2022.

[141] Chaoyi Lu, Baojun Liu, Zhou Li, Shuang Hao, Haixin Duan, Mingming Zhang, Chunying Leng, Ying Liu, Zaifeng Zhang, and Jianping Wu. An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come? In *Proceedings of the Internet Measurement Conference*, pages 22–35, Amsterdam Netherlands, October 2019. ACM.

[142] Gonzalo Marín, Pedro Casas, and Germán Capdehourat. RawPower: Deep Learning based Anomaly Detection from Raw Network Traffic Measurements. In *Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos*, pages 75–77, Budapest Hungary, August 2018. ACM.

[143] Jeremy Martin, Douglas Alpuche, Kristina Bodeman, Lamont Brown, Ellis Fenske, Lucas Foppe, Travis Mayberry, Erik C. Rye, Brandon Sipes, and Sam Teplov. Handoff All Your Privacy: A Review of Apple's Bluetooth Low Energy Continuity Protocol. *arXiv preprint arXiv:1904.10600*, 2019.

[144] Alexander Mayrhofer. The EDNS(0) Padding Option. Request for Comments RFC 7830, Internet Engineering Task Force, May 2016.

[145] Alexander Mayrhofer. Padding Policies for Extension Mechanisms for DNS (EDNS(0)). Request for Comments RFC 8467, Internet Engineering Task Force, October 2018.

[146] Kerry McKay, Lawrence Bassham, Meltem Sönmez Turan, and Nicky Mouha. Report on lightweight cryptography. Technical report, National Institute of Standards and Technology, 2016.

[147] Afef Mdhaffar, Tarak Chaari, Kaouthar Larbi, Mohamed Jmaiel, and Bernd Freisleben. IoT-based health monitoring via LoRaWAN. In *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*, pages 519–524, July 2017.

[148] Simon Meier, Benedikt Schmidt, Cas Cremers, and David Basin. The TAMARIN Prover for the Symbolic Analysis of Security Protocols. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Natasha Sharygina, and Helmut Veith, editors, *Computer Aided Verification*, volume 8044, pages 696–701. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[149] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N. Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 2177–2184, June 2017.

[150] Konstantin Mikhaylov, Juha Petäjäjärvi, and Ari Pouttu. Effect of downlink traffic on performance of LoRaWAN LPWA networks: Empirical study. In *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–6. IEEE, 2018.

[151] Abhishek Kumar Mishra, Aline Carneiro Viana, and Nadjib Achir. Introducing benchmarks for evaluating user-privacy vulnerability in WiFi. In *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, pages 1–7. IEEE, 2023.

[152] Satyajayant Misra and Guoliang Xue. Efficient anonymity schemes for clustered wireless sensor networks. *International Journal of Sensor Networks*, 1(1/2):50, 2006.

[153] Frederik Möllers. Energy-Efficient Dummy Traffic Generation for Home Automation Systems. *Proceedings on Privacy Enhancing Technologies*, 2020(4):376–393, October 2020.

[154] Mohammad Mezanur Rahman Monjur, Joseph Heacock, Rui Sun, and Qiaoyan Yu. An Attack Analysis Framework for LoRaWAN applied Advanced Manufacturing. In *2021 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–7, Boston, MA, USA, November 2021. IEEE.

[155] MohammadrezaMontazeriShatoori, Logan Davidson, Gurdip Kaur, and Arash Habibi Lashkari. Detection of doh tunnels using time-series classification of encrypted traffic. In *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 63–70. IEEE, 2020.

[156] Christoph Neumann, Olivier Heen, and Stéphane Onno. An empirical study of passive 802.11 Device Fingerprinting. *arXiv:1404.6457 [cs]*, April 2014.

[157] Frank Nijeboer. Detection of HTTPS Encrypted DNS Traffic. 2020.

[158] Helen Nissenbaum. Privacy as contextual integrity. *Wash. L. Rev.*, 79:119, 2004.

[159] Hassan Noura, Tarif Hatoum, Ola Salman, Jean-Paul Yaacoub, and Ali Chehab. LoRaWAN security survey: Issues, threats and possible mitigation techniques. *Internet of Things*, 12:100303, December 2020.

[160] Rebekah Overdorf, Mark Juarez, Gunes Acar, Rachel Greenstadt, and Claudia Diaz. How Unique is Your .onion?: An Analysis of the Fingerprintability of Tor Onion Services. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 2021–2036, Dallas Texas USA, October 2017. ACM.

[161] Wubin Pan, Guang Cheng, and Yongning Tang. Wenc: Https encrypted traffic classification using weighted ensemble learning and markov chain. In *2017 IEEE Trustcom/BigDataSE/ICESS*, pages 50–57. IEEE, 2017.

[162] Jeffrey Pang, Ben Greenstein, Ramakrishna Gummadi, Srinivasan Seshan, and David Wetherall. 802.11 user fingerprinting. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking - MobiCom '07*, page 99, Montr&233;al, Québec, Canada, 2007. ACM Press.

[163] UK Parliament. The Product Security and Telecommunications Infrastructure (Security Requirements for Relevant Connectable Products). https://www.legislation.gov.uk/uksi/2023/1007/contents/made, 2023.

[164] Article 29 Working Party. Guidelines on Data Protection Impact Assessment (DPIA), 2017.

[165] Samuel Pélissier, Jan Aalmoes, Abhishek Kumar Mishra, Mathieu Cunche, Vincent Roca, and Didier Donsez. Privacy-preserving pseudonyms for LoRaWAN. In *17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec 2024)*, 2024.

[166] Samuel Pélissier, Mathieu Cunche, Vincent Roca, and Didier Donsez. Device reidentification in LoRaWAN through messages linkage. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 98–103, 2022.

[167] Samuel Pélissier, Abhishek Kumar Mishra, Mathieu Cunche, Vincent Roca, and Didier Donsez. Introducing Multi-Domain Fingerprints for Lorawan Device Linkage. *Available at SSRN 4728641*.

[168] Roberto Perdisci, Thomas Papastergiou, Omar Alrawi, and Manos Antonakakis. IoTFinder: Efficient Large-Scale Identification of IoT Devices via Passive DNS Traffic Analysis. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 474–489, Genoa, Italy, September 2020. IEEE.

[169] Jonathan Petit, Florian Schaub, Michael Feiri, and Frank Kargl. Pseudonym schemes in vehicular networks: A survey. *IEEE communications surveys & tutorials*, 17(1):228–255, 2014.

[170] Tara Petric, Mathieu Goessens, Loutfi Nuaymi, Laurent Toutain, and Alexander Pelov. Measurements, performance and analysis of LoRa FABIAN, a real-world implementation of LPWAN. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–7, Valencia, Spain, September 2016. IEEE.

[171] Antônio J. Pinheiro, Paulo Freitas de Araujo-Filho, Jeandro de M. Bezerra, and Divanilson R. Campelo. Adaptive packet padding approach for smart home networks: A tradeoff between privacy and performance. *IEEE Internet of Things Journal*, 8(5):3930–3938, 2020.

[172] Nico Podevijn, David Plets, Jens Trogh, Luc Martens, Pieter Suanet, Kim Hendrikse, and Wout Joseph. TDoA-Based Outdoor Positioning with Tracking Algorithm in a Public LoRa Network. *Wireless Communications and Mobile Computing*, 2018:1–9, May 2018.

[173] Nelson Prates, Andressa Vergütz, Ricardo T. Macedo, Aldri Santos, and Michele Nogueira. A defense mechanism for timing-based side-channel attacks on IoT traffic. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE, 2020.

[174] Michael Quan, Eduardo Navarro, and Benjamin Peuker. Wi-fi localization using rssi fingerprinting. 2010.

[175] S. R. Jino Ramson, S. Vishnu, A. Alfred Kirubaraj, Theodoros Anagnostopoulos, and Adnan M. Abu-Mahfouz. A LoRaWAN IoT-Enabled Trash Bin Level Monitoring System. *IEEE Transactions on Industrial Informatics*, 18(2):786–795, February 2022.

[176] Tirumaleswar Reddy.K, Dan Wing, and Prashanth Patil. DNS over Datagram Transport Layer Security (DTLS). Request for Comments RFC 8094, Internet Engineering Task Force, February 2017.

[177] Jingjing Ren, Daniel J Dubois, David Choffnes, Anna Maria Mandalari, Roman Kolcun, and Hamed Haddadi. Information exposure from consumer iot devices: A multidimensional, network-informed measurement approach. In *Proceedings of the Internet Measurement Conference*, 2019.

[178] Pieter Robyns, Peter Quax, Wim Lamotte, and William Thenaers. Gr-lora: An efficient LoRa decoder for GNU Radio. *Zenodo Ed*, 10:5281, 2017.

[179] Ribana Roscher, Bastian Bohn, Marco F. Duarte, and Jochen Garcke. Explainable machine learning for scientific insights and discoveries. *Ieee Access*, 8:42200–42216, 2020.

[180] Erik C. Rye, Robert Beverly, and kc claffy. Follow the Scent: Defeating IPv6 Prefix Rotation Privacy. In *Proceedings of the 21st ACM Internet Measurement Conference*, pages 739–752, November 2021.

[181] Said Jawad Saidi, Oliver Gasser, and Georgios Smaragdakis. One Bad Apple Can Spoil Your IPv6 Privacy, March 2022.

[182] Yaman Sangar and Bhuvana Krishnaswamy. WiChronos: Energy-efficient modulation for long-range, large-scale wireless networks. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, London United Kingdom, April 2020. ACM.

[183] Cristiana Santos, Nataliia Bielova, Vincent Roca, Mathieu Cunche, Gilles Mertens, Karel Kubicek, and Hamed Haddadi. Feedback to the European Data Protection Board's Guidelines 2/2023 on Technical Scope of Art. 5(3) of ePrivacy Directive, February 2024.

[184] Amardeo C. Sarma and João Girão. Identities in the Future Internet of Things. *Wireless Personal Communications*, 49(3):353–363, May 2009.

[185] David W. Scott. Sturges' rule. *WIREs Computational Statistics*, 1(3):303–306, November 2009.

[186] Narmeen Shafqat, Daniel J. Dubois, David Choffnes, Aaron Schulman, Dinesh Bharadia, and Aanjhan Ranganathan. ZLeaks: Passive Inference Attacks on Zigbee based Smart Homes. *arXiv:2107.10830 [cs]*, July 2021.

[187] Vishal Sharma, Ilsun You, Giovanni Pau, Mario Collotta, Jae Deok Lim, and Jeong Nyeo Kim. LoRaWAN-Based Energy-Efficient Surveillance by Drones for Intelligent Transportation Systems. *Energies*, 11(3):573, March 2018.

[188] Guanxiong Shen, Junqing Zhang, Alan Marshall, Linning Peng, and Xianbin Wang. Radio frequency fingerprint identification for LoRa using spectrogram and CNN. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021.

[189] Meng Shen, Mingwei Wei, Liehuang Zhu, and Mingzhong Wang. Classification of Encrypted Traffic With Second-Order Markov Chains and Application Attribute Bigrams. *IEEE Transactions on Information Forensics and Security*, 12(8):1830–1843, August 2017.

[190] Taeshik Shon and Jongsub Moon. A hybrid machine learning approach to network anomaly detection. *Information Sciences*, 177(18):3799–3821, 2007.

[191] Sandra Siby, Marc Juarez, Claudia Diaz, Narseo Vallina-Rodriguez, and Carmela Troncoso. Encrypted DNS –> Privacy? A Traffic Analysis Perspective. *arXiv:1906.09682 [cs]*, October 2019.

[192] Ritesh Kumar Singh, Michiel Aernouts, Mats De Meyer, Maarten Weyn, and Rafael Berkvens. Leveraging LoRaWAN technology for precision agriculture in greenhouses. *Sensors*, 20(7):1827, 2020.

[193] Daniel J. Solove. The myth of the privacy paradox. *Geo. Wash. L. Rev.*, 89:1, 2021.

[194] Junhyuk Song, Radha Poovendran, Jicheol Lee, and Tetsu Iwata. The aes-cmac algorithm. Technical report, 2006.

[195] Pietro Spadaccino, Francesco Giuseppe Crinó, and Francesca Cuomo. LoRaWAN Behaviour Analysis through Dataset Traffic Investigation. *Sensors*, 22(7):2470, March 2022.

[196] Pietro Spadaccino, Domenico Garlisi, Francesca Cuomo, Giorgio Pillon, and Patrizio Pisani. Discovery privacy threats via device de-anonymization in LoRaWAN. In *2021 19th Mediterranean Communication and Computer Networking Conference (MedComNet)*, pages 1–8, June 2021.

[197] Vijay Srinivasan, John Stankovic, and Kamin Whitehouse. Protecting your daily in-home activity information from a wireless snooping attack. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, pages 202–211, Seoul Korea, September 2008. ACM.

[198] Paul Staat, Simon Mulzer, Stefan Roth, Veelasha Moonsamy, Markus Heinrichs, Rainer Kronberger, Aydin Sezgin, and Christof Paar. IRShield: A countermeasure against adversarial physical-layer wireless sensing. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1705–1721. IEEE, 2022.

[199] Jothi Prasanna Shanmuga Sundaram, Wan Du, and Zhiwei Zhao. A Survey on LoRa Networking: Research Problems, Current Solutions and Open Issues. *arXiv:1908.10195 [cs, eess]*, August 2019.

[200] Latanya Sweeney. K-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–570, 2002.

[201] Xiaopeng Tan, Shaojing Su, Zhiping Huang, Xiaojun Guo, Zhen Zuo, Xiaoyong Sun, and Longqing Li. Wireless sensor networks intrusion detection based on SMOTE and the random forest algorithm. *Sensors*, 19(1):203, 2019.

[202] Vishal A. Thakor, Mohammad Abdur Razzaque, and Muhammad RA Khandaker. Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities. *IEEE Access*, 9:28177–28193, 2021.

[203] Oliver Thompson, Anna Maria Mandalari, and Hamed Haddadi. Rapid IoT Device Identification at the Edge. In *Proceedings of the 2nd ACM International Workshop on Distributed Machine Learning*, pages 22–28, December 2021.

[204] Min Ye Thu, Wunna Htun, Yan Lin Aung, Pyone Ei Ei Shwe, and Nay Min Tun. Smart Air Quality Monitoring System with LoRaWAN. In *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, pages 10–15, November 2018.

[205] Stefano Tomasin, Simone Zulian, and Lorenzo Vangelista. Security analysis of lorawan join procedure for internet of things networks. In *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 1–6. IEEE, 2017.

[206] Nuno Torres, Pedro Pinto, and Sérgio Ivan Lopes. Security Vulnerabilities in LPWANs—An Attack Vector Analysis for the IoT Ecosystem. *Applied Sciences*, 11(7):3176, January 2021.

[207] Kun-Lin Tsai, Fang-Yie Leu, Ilsun You, Shuo-Wen Chang, Shiung-Jie Hu, and Hoonyong Park. Low-power AES data encryption architecture for a LoRaWAN. *IEEE Access*, 7:146348–146357, 2019.

[208] A. Selcuk Uluagac, Sakthi V. Radhakrishnan, Cherita Corbett, Antony Baca, and Raheem Beyah. A passive technique for fingerprinting wireless devices with Wired-side Observations. In *2013 IEEE Conference on Communications and Network Security (CNS)*, pages 305–313, October 2013.

[209] Publications Office of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance). https://op.europa.eu/en/publication-detail/-/publication/3e485e15-11bd-11e6-ba9a-01aa75ed71a1/language-en, April 2016.

[210] Muhammad Usama, Junaid Qadir, Aunn Raza, Hunain Arif, Kok-Lim Alvin Yau, Yehia Elkhatib, Amir Hussain, and Ala Al-Fuqaha. Unsupervised machine learning for networking: Techniques, applications and research challenges. *IEEE access*, 7:65579–65615, 2019.

[211] Christian Vaas, Mohammad Khodaei, Panos Papadimitratos, and Ivan Martinovic. Nowhere to hide? Mix-Zones for Private Pseudonym Change using Chaff Vehicles. In *2018 IEEE Vehicular Networking Conference (VNC)*, pages 1–8, Taipei, Taiwan, December 2018. IEEE.

[212] Eef Van Es, Harald Vranken, and Arjen Hommersom. Denial-of-Service Attacks on LoRaWAN. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, pages 1–6, Hamburg Germany, August 2018. ACM.

[213] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo S. Cardoso, and Frank Piessens. Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '16, pages 413–424, New York, NY, USA, May 2016. Association for Computing Machinery.

[214] Dmitrii Vekshin, Karel Hynek, and Tomas Cejka. DoH Insight: Detecting DNS over HTTPS by machine learning. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, pages 1–8, Virtual Event Ireland, August 2020. ACM.

[215] Paul A. Vixie. Extensions to DNS (EDNS). Internet Draft draft-ietf-dnsind-edns-03, Internet Engineering Task Force, August 1998.

[216] Paul A. Vixie. Extensions to DNS (EDNS1). Internet Draft draft-ietf-dnsext-edns1-03, Internet Engineering Task Force, August 2002.

[217] Tien Dang Vo-Huu, Triet Dang Vo-Huu, and Guevara Noubir. Fingerprinting Wi-Fi Devices Using Software Defined Radios. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pages 3–14, Darmstadt Germany, July 2016. ACM.

[218] Sandra Wachter. Normative challenges of identification in the Internet of Things: Privacy, profiling, discrimination, and the GDPR. *Computer law & security review*, 34(3):436–449, 2018.

[219] Isabel Wagner and David Eckhoff. Technical Privacy Metrics: A Systematic Survey. *ACM Computing Surveys*, 51(3):1–38, May 2019.

[220] Yinxin Wan, Kuai Xu, Feng Wang, and Guoliang Xue. Characterizing and Mining Traffic Patterns of IoT Devices in Edge Networks. *IEEE Transactions on Network Science and Engineering*, 8(1):89–101, January 2021.

[221] Samuel Warren and Louis Brandeis. The Right to Privacy. In Tom Goldstein, editor, *Killing the Messenger*, pages 1–21. Columbia University Press, 1890.

[222] Stephan Wesemeyer, Ioana Boureanu, Zach Smith, and Helen Treharne. Extensive security verification of the LoRaWAN key-establishment: Insecurities & patches. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 425–444. IEEE, 2020.

[223] Alan F Westin. *Privacy And Freedom*. 1968.

[224] Maarten Weyn, Glenn Ergeerts, Rafael Berkvens, Bartosz Wojciechowski, and Yordan Tabakov. DASH7 alliance protocol 1.0: Low-power, mid-range sensor and actuator communication. In *2015 IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 54–59. IEEE, 2015.

[225] Maarten Weyn, Glenn Ergeerts, Luc Wante, Charles Vercauteren, and Peter Hellinckx. Survey of the DASH7 Alliance Protocol for 433 MHz Wireless Sensor Communication. *International Journal of Distributed Sensor Networks*, 9(12):870430, December 2013.

[226] Tim Wicinski. DNS Privacy Considerations. Request for Comments RFC 9076, Internet Engineering Task Force, July 2021.

[227] Sanjay Yadav and Sanyam Shukla. Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, pages 78–83. IEEE, 2016.

[228] Yi Yang, Min Shao, Sencun Zhu, and Guohong Cao. Towards statistically strong source anonymity for sensor networks. *ACM Transactions on Sensor Networks*, 9(3):1–23, May 2013.

[229] Lin Yao, Lin Kang, Pengfei Shang, and Guowei Wu. Protecting the sink location privacy in wireless sensor networks. *Personal and Ubiquitous Computing*, 17(5):883–893, June 2013.

[230] Shui Yu, Guofeng Zhao, Wanchun Dou, and Simon James. Predicted Packet Padding for Anonymous Web Browsing Against Traffic Analysis Attacks. *IEEE Transactions on Information Forensics and Security*, 7(4):1381–1393, August 2012.

[231] Qiong Zhang and Kewang Zhang. Protecting Location Privacy in IoT Wireless Sensor Networks through Addresses Anonymity. *Security and Communication Networks*, 2022:e2440313, April 2022.

[232] Yue Zhang and Zhiqiang Lin. When Good Becomes Evil: Tracking Bluetooth Low Energy Devices via Allowlist-based Side Channel and Its Countermeasure. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3181–3194, Los Angeles CA USA, November 2022. ACM.

FOLIO ADMINISTRATIF

THESE DE L'INSA LYON, MEMBRE DE L'UNIVERSITE DE LYON

NOM : **PÉLISSIER** DATE de SOUTENANCE : **27 septembre 2024**

Prénoms : **Samuel**

TITRE : **Privacy-preserving communications for the IoT**

NATURE : **Doctorat** Numéro d'ordre : **2024ISAL0075**

École Doctorale : **InfoMaths**

Spécialité : **Informatique**

RÉSUMÉ :
Les dernières décennies ont été témoins de l'émergence et de la prolifération d'objets connectés, communément appelés *Internet des Objets* (IdO). Cet écosystème divers correspond à une large gamme de dispositifs spécialisés, allant de la caméra IP aux capteurs détectant les fuites d'eau, chacun conçu pour répondre à des objectifs et des contraintes de consommation d'énergie, de puissance de calcul ou de coût. Le développement rapide de nombreuses technologies et leur connexion en réseau s'accompagne de la génération d'un important volume de données, soulevant des préoccupations en matière de vie privée, en particulier dans des domaines sensibles tels que la santé ou les maisons connectées.
Dans cette thèse, nous exploitons les techniques d'apprentissage automatique (*machine learning*) pour explorer les problèmes liés à la vie privée des objets connectés via leurs protocoles réseau. Tout d'abord, nous étudions les attaques possibles contre LoRaWAN, un protocole longue distance et à faible coût d'énergie. Nous explorons la relation entre deux identifiants du protocole et montrons que leur séparation théorique peut être contrecarrée en utilisant les métadonnées produites lors de la connexion au réseau. En nous appuyant sur une approche multi-domaines (contenu, temps, radio), nous démontrons que ces métadonnées permettent à un attaquant d'identifier les objets connectés de manière unique malgré le chiffrement du trafic, ouvrant la voie au traçage ou à la ré-identification. Nous explorons ensuite les possibles contre-mesures, en analysant systématiquement les données utilisées lors de ces attaques et en proposant des techniques pour les obfusquer ou réduire leur pertinence. Nous démontrons que seule une approche combinée offre une réelle protection. Par ailleurs, nous proposons et évaluons diverses solutions de pseudonymes temporaires adaptées aux contraintes de LoRaWAN, en particulier la consommation énergétique.
Enfin, nous adaptons notre méthodologie d'apprentissage automatique à DNS, un protocole largement déployé dans l'IdO grand public. À nouveau basées sur les métadonnées, notre attaque permet d'identifier les objets connectés, malgré le chiffrement du flux DNS-over-HTTPS. Explorant les contre-mesures potentielles, nous observons un non-respect des standards liés au padding, entraînant la compromission partielle de la vie privée des utilisateurs.

MOTS-CLÉS : **Vie privée ; Internet des Objets ; Objets connectés ; Sécurité ; Protocoles ; Réseaux sans fil ; Métadonnées ; Empreinte ; Re-identification ; Inférence d'activités ; LoRaWAN ; LPWAN ; DNS ; DNS-over-HTTPS.**

Laboratoire(s) de recherche : **CITI**

Directeurs de thèse : **Mathieu CUNCHE** (Professeur des Universités, INSA Lyon) & **Vincent ROCA** (Chargé de recherche, Inria)

Président du Jury : **Hervé RIVANO** (Professeur des Universités, INSA Lyon)

Composition du Jury :

MITTON	**Nathalie**	Directeur de Recherche	Centre Inria de l'Université de Lille	Rapporteuse
RASMUSSEN	**Kasper**	Professeur des Universités	University of Oxford	Rapporteur
CARNEIRO VIANA	**Aline**	Directeur de Recherche	Centre Inria Saclay	Examinatrice
FRANCILLON	**Aurélien**	Professeur des Universités	EURECOM	Examinateur