Enabling the multi facets of privacy in modern communication systems via cutting-edge techniques and protocols

Department of Electrical and Information Engineering

INDUSTRY 4.0 Ph.D. Program

SSD: IINF-03/A – TELECOMMUNICATIONS

Final Dissertation

Enabling The Multi Facets of Privacy in Modern Communication Systems via Cutting-edge Techniques and Protocols

by

Ingrid Huso

Supervisors:

Prof. Gennaro Boggia

Prof. Giuseppe Piro

Coordinator of Ph.D. Program:

Prof. Caterina Ciminelli

Course n°37, 01/11/2021 - 31/10/2024

LIBERATORIA PER L'ARCHIVIAZIONE DELLA TESI DI DOTTORATO

Al Magnifico Rettore
del Politecnico di Bari

La sottoscritta Ingrid Huso nata a Trani (BA) il 19/04/1996 residente a Trani (BT) in via Firenze 14 e-mail ingrid.huso@gmail.com iscritta al 3° anno di Corso di Dottorato di Ricerca in Industria 4.0 ciclo XXXVII ed essendo stato ammesso a sostenere l'esame finale con la prevista discussione della tesi dal titolo:

"Enabling The Multi Facets of Privacy in Modern Communication Systems via Cutting-edge Techniques and Protocols"

____

DICHIARA

1) di essere consapevole che, ai sensi del D.P.R. n. 445 del 28.12.2000, le dichiarazioni mendaci, la falsità negli atti e l'uso di atti falsi sono puniti ai sensi del codice penale e delle Leggi speciali in materia, e che nel caso ricorressero dette ipotesi, decade fin dall'inizio e senza necessità di nessuna formalità dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni;
2) di essere iscritto al Corso di Dottorato di ricerca INDUSTRIA 4.0 ciclo XXXVII, corso attivato ai sensi del "*Regolamento dei Corsi di Dottorato di ricerca del Politecnico di Bari*", emanato con D.R. n.286 del 01.07.2013;
3) di essere pienamente a conoscenza delle disposizioni contenute nel predetto Regolamento in merito alla procedura di deposito, pubblicazione e autoarchiviazione della tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica;
4) di essere consapevole che attraverso l'autoarchiviazione delle tesi nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica del Politecnico di Bari (IRIS-POLIBA), l'Ateneo archivierà e renderà consultabile in rete (nel rispetto della Policy di Ateneo di cui al D.R. 642 del 13.11.2015) il testo completo della tesi di dottorato, fatta salva la possibilità di sottoscrizione di apposite licenze per le relative condizioni di utilizzo (di cui al sito http://www.creativecommons.it/Licenze), e fatte salve, altresì, le eventuali esigenze di "embargo", legate a strette considerazioni sulla tutelabilità e sfruttamento industriale/commerciale dei contenuti della tesi, da rappresentarsi mediante compilazione e sottoscrizione del modulo in calce (Richiesta di embargo);
5) che la tesi da depositare in IRIS-POLIBA, in formato digitale (PDF/A) sarà del tutto identica a quelle **consegnate**/inviate/da inviarsi ai componenti della commissione per l'esame finale e a qualsiasi altra copia depositata presso gli Uffici del Politecnico di Bari in forma cartacea o digitale, ovvero a quella da discutere in sede di esame finale, a quella da depositare, a cura dell'Ateneo, presso le Biblioteche Nazionali Centrali di Roma e Firenze e presso tutti gli Uffici competenti per legge al momento del deposito stesso, e che di conseguenza va esclusa qualsiasi responsabilità del Politecnico di Bari per quanto riguarda eventuali errori, imprecisioni o omissioni nei contenuti della tesi;
6) che il contenuto e l'organizzazione della tesi è opera originale realizzata dal sottoscritto e non compromette in alcun modo i diritti di terzi, ivi compresi quelli relativi alla sicurezza dei dati personali; che pertanto il Politecnico di Bari ed i suoi funzionari sono in ogni caso esenti da responsabilità di qualsivoglia natura: civile, amministrativa e penale e saranno dal sottoscritto tenuti indenni da qualsiasi richiesta o rivendicazione da parte di terzi;
7) che il contenuto della tesi non infrange in alcun modo il diritto d'Autore né gli obblighi connessi alla salvaguardia di diritti morali od economici di altri autori o di altri aventi diritto, sia per testi, immagini, foto, tabelle, o altre parti di cui la tesi è composta.

Bari, 05/11/2024 Firma Ingrid Huso

La sottoscritto, con l'autoarchiviazione della propria tesi di dottorato nell'Archivio Istituzionale ad accesso aperto del Politecnico di Bari (POLIBA-IRIS), pur mantenendo su di essa tutti i diritti d'autore, morali ed economici, ai sensi della normativa vigente (Legge 633/1941 e ss.mm.ii.),

CONCEDE

- al Politecnico di Bari il permesso di trasferire l'opera su qualsiasi supporto e di convertirla in qualsiasi formato al fine di una corretta conservazione nel tempo. Il Politecnico di Bari garantisce che non verrà effettuata alcuna modifica al contenuto e alla struttura dell'opera.
- al Politecnico di Bari la possibilità di riprodurre l'opera in più di una copia per fini di sicurezza, back-up e conservazione.

Bari, 05/11/2024 Firma Ingrid Huso

Department of Electrical and Information Engineering

INDUSTRY 4.0 Ph.D. Program

SSD: IINF-03/A – TELECOMMUNICATIONS

Final Dissertation

Enabling The Multi Facets of Privacy in Modern Communication Systems via Cutting-edge Techniques and Protocols

by

Ingrid Huso

Ingrid Huso

Referees:

Prof. Stefania Bartoletti

Prof. Stefano Tomasin

Supervisors:

Prof. Gennaro Boggia

Gennaro Boggia

Prof. Giuseppe Piro

Giuseppe Piro

Coordinator of Ph.D. Program:

Prof. Caterina Ciminelli

Course n°37, 01/11/2021 - 31/10/2024

Acknowledgements

Desidero innanzitutto esprimere il mio più sentito ringraziamento al mio Supervisor, Prof. *Gennaro Boggia*, per ogni possibilità di confronto e, soprattutto, per la fiducia riposta in me sin dal primo giorno della mia tesi triennale, introducendomi al mondo della ricerca. Un sincero grazie va anche al mio Co-Supervisor, Prof. *Giuseppe Piro*, per avermi guidato passo dopo passo in questo percorso, facendomi scoprire e approfondire la mia passione per la ricerca. Un grazie va anche al Prof. *Alfredo Grieco* e a tutto il *Telematics Lab*. Inoltre, vorrei ringraziare tutte le persone che ho avuto il piacere di incontrare durante i miei mesi di visiting ad Eindhoven.

Ringrazio, per il loro supporto, i miei ultimi compagni di avventure, *Enrico, Daniele, Salvatore* con cui ho condiviso momenti belli e divertenti negli ultimi mesi di dottorato.

Un grazie speciale va, invece, a quello che per me è il *"mio Lab"* che ha reso questo percorso indelebile facendomi conoscere veri amici prima che colleghi. *Giancarlo*, in parte responsabile del mio essere qui in questo momento, grazie per tutti i momenti trascorsi davanti ad un "gin all' acqua di mare" o allo spritz davanti al pantheon. *Antonio*, grazie per avermi sempre spronata e per ogni parola di conforto anche davanti al sushi, anzi ai 10 sashimi. *Francesco*, grazie per essere stata la mia spalla destra in questo percorso ma anche per tutti confronti quotidiani inclusi sfoghi e confidenze personali, durante le nostre lunghe camminate, ovviamente. *Federica*, grazie per aver fatto emergere la mia parte estroversa e perché in poco tempo e con pochi piantini sei diventata parte fondamentale del percorso.

Ringrazio anche persone non del campo che hanno contribuito ad alleggerire le giornate al Poliba. *Federica ("di Barletta")*, persona buona e dolce, grazie per ogni consiglio e confronto davanti ad un bicchiere di birra. Ringrazio di cuore per ogni singola parola di supporto e consiglio, *Annarita*, con cui in pochissimo tempo, ho costruito un rapporto sincero tanto da riuscire a capirci con uno sguardo.

Passando poi alle costanti, grazie alle persone che da quasi 15 anni mi sostengono in ogni mio passo: Rosa e Rossella. Grazie per essere sempre lì con me, per il continuo supporto e per la continua fiducia ma soprattutto per il non lamentarsi della mia "assenza" quotidiana.

Un grazie infinitamente speciale é per i miei *genitori* che passo passo mi hanno accompagnata, supportata e sopportata in questo percorso cercando di farmi vedere sempre quanto loro vogliano sempre il meglio per me, vi voglio bene. Un grazie lo devo anche ad *Antonio*, il fratello acquisito, per ogni confronto e conforto. Ringrazio, con tutto il cuore e tutta me stessa, mia *sorella* per essere sempre la parte migliore di me, per supportarmi, incoraggiarmi e spronarmi in ogni momento. Ti voglio bene piccola *Chiara*, "sempre insieme".

Per quanto emotivamente altalenante, sono grata a questo viaggio di poco più di tre anni per le esperienze vissute e per tutte le persone incontrate lungo il cammino. Grazie a tutto questo che ha contribuito a rendermi la persona che sono oggi, sia a livello personale che professionale.

« *Hey. Don't ever let somebody tell you: "You can't do something". Not even me.* Alright? You got a dream. You gotta protect it. *If you want something, go get it.* »

The Pursuit of Happiness

« *Vincere non è importante, è l'unica cosa che conta* »

Giampiero Boniperti

POLITECNICO DI BARI

Abstract

Department of Electrical and Information Engineering

Doctor of Philosophy

Enabling The Multi Facets of Privacy in Modern Communication Systems via Cutting-edge Techniques and Protocols

by Ingrid HUSO[*]

The rapid advancement of digital communication technologies, accelerated by the convergence of the Internet of Things (IoT), Industrial IoT (IIoT), and 5G networks, has revolutionized information sharing while introducing new privacy and security challenges in areas such as data privacy, trust management, and secure communications. This work introduces a privacy-centered approach to tackling emerging security challenges, starting with a novel fog-enabled Social Internet of Things (SIoT) architecture that integrates Trust Management Systems (TMS) to enhance service discovery, trustworthiness, and resource efficiency. To protect data at the network edge, the thesis develops a distributed data dissemination framework that utilizes searchable encryption and edge computing to ensure robust data privacy. Additionally, it explores the influence of carrier frequency on the resilience of Radio Frequency Fingerprinting (RFF) within Physical Layer Security (PLS), reinforcing device-level authentication with privacy-preserving elements. Finally, the thesis proposes an end-to-end cryptographic framework for Lawful Interception (LI) in Beyond 5G networks, aiming to strike a balance between stringent privacy protections and compliance with regulatory standards. Together, these contributions offer a multifaceted approach to privacy, blending architectural innovations, encryption mechanisms, and scalable frameworks for the next generation of secure communication networks.

[*] Ingrid Huso is grateful to Politecnico di Bari for supporting her PhD scholarship, whose scientific contributions and results of her research activities have been highlighted in projects funded by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, in the context of partnership on "Telecommunications of the Future" (PE000001 - program "RESTART", CUP: D93C22000910001) and partnership on "Cybersecurity" (PE000007 - program "SERICS", CUP: D33C22001300002, project ISP5G+).

Contents

List of Figures **x**

List of Tables **xi**

List of Acronyms **xiii**

Pubblications **xvii**

Scientific contribution **xix**

Introduction **1**

1 Introduction to Security in Modern Communication Systems **3**
1.1 Key Enabling Technologies in Modern Communication Networks 3
1.2 Security Issues and Challenges 4
1.2.1 Trustworthiness Management 5
1.2.2 Privacy Preservation in Data Distribution 6
1.2.3 Physical Layer Security 6
1.2.4 End-to-End Encryption against Lawful Interception 7

2 Trusted Service Provisioning in Social Internet of Things **9**
2.1 State of the Art on Trust Management Systems in the SIoT 9
2.2 The conceived Multi-tired SIoT architecture 11
2.3 Overview of the Proposed Trusted Service Provisioning Process 12
2.3.1 Conceived Trust model 13
2.3.2 Resource management evaluation 14
2.4 Performance Evaluation 15
2.4.1 Simulation results 15

3 Privacy-oriented Data dissemination **21**
3.1 Background concepts and literature review 21
3.1.1 Background concepts 22
3.1.2 Related works on Searchable Encryption in the IIoT 22
3.1.3 Literature review on Searchable Encryption in the IoT 23
Problem Description 24
3.2 Distributed and Privacy-Preserving Data Dissemination at the Network Edge via Attribute-Based Searchable Encryption 26
3.2.1 The Conceived Data Dissemination Scheme 27
Design Principles 27
Technical details about the data dissemination workflow 28
Running Example 31
3.2.2 Analytical evaluation 32
Analysis of the average search time 33
Analysis of the average delivery delay 35

3.3 Privacy-preserving data dissemination scheme based on Searchable Encryption, publish-subscribe model, and edge computing 38
3.3.1 The Conceived Data Dissemination Scheme 39
Data dissemination workflow 40
3.3.2 Security Proof 44
3.3.3 Performance evaluation 45
The followed Methodology 45
System setup description 47
Analysis of cryptographic operations 48
Impact of the Network Load on the Search Time 48
Average Delivery Delay 50
Energy Consumption 51

4 Radio Frequency Fingerprinting 53
4.1 Background and Related Works 53
4.1.1 Digital Modulation 53
4.1.2 Deep Learning 54
4.1.3 Related Works on Image-based RFF 54
4.2 Frequency Matters: On the Impact of Carrier Frequency on Privacy in Radio Fingerprinting 55
4.2.1 Reference Scenario and Adversary Model 56
Reference Scenario 56
Adversary model 57
4.2.2 Deployed Methodology 57
IQ sample collection 57
Image Generation 57
Multi-class Classification 58
Investigated Key Performance Indicators 58
4.2.3 Performance Evaluation 58
Experimental Testbed 58
Fingerprint Robustness to Carrier Frequency 59
Tracking Attacks 61
4.3 Jamming Echoes: On the Impact of Out-of-Band Interference on Radio Frequency Fingerprinting 62
4.3.1 Reference Scenario 62
4.3.2 RFF Methodology 63
Data Collection 63
Image Generation 63
Multi-class Classification 64
4.3.3 Experimental Measurements and Analysis 64
Experimental Testbed 64
Results 66

5 Design and implementation of a Looking-Forward Lawful Interception Architecture for Future Mobile Communication Systems 69
5.1 Background and Motivation 70
5.1.1 Lawful Interception 71
5.1.2 End-to-End Encryption 72
5.1.3 Key Escrow 73
5.2 The proposed methodology 74
5.2.1 Design Principles 74

5.2.2 Technical Details 75
5.2.3 Security proof and threat analysis 76
5.3 Proof-of-Concept Implementation 78
5.4 Performance Evaluation 81
5.4.1 LI for Real-Time VoIP Call 82
5.4.2 LI for End-to-end File Exchange 82
5.4.3 LI impact on the user QoS 84
5.4.4 Comparison: VoIP vs. File Exchange 86

Conclusions and Future Works **89**

A Cryptographic description of the SE algorithm presented in [94] **91**

B Cryptographic description of the SE algorithm presented in [88] **93**

C Detailed Description of the Key Escrow Algorithm **95**

Bibliography **97**

List of Figures

2.1 The proposed SIoT architecture. 11
2.2 The trusted service provisioning process. 13
2.3 Queued Requests Evaluation. 16
2.4 Queued Request increasing traffic load. 17
2.5 Average Delay increasing traffic load. 18
2.6 Malicious social objects detection. 19

3.1 The reference distributed network architecture. 27
3.2 Data dissemination workflow. 29
3.3 Running example. 32
3.4 Average search time vs number of MEC applications. 34
3.5 Average delivery delay vs number of MEC applications. 36
3.6 Average delivery delay vs numbers of attributes. 37
3.7 The reference distributed service architecture. 39
3.8 General Data dissemination Scheme. 41
3.9 Flowchart of the proposed search and data dissemination phase. 43
3.10 Testbed setup 46
3.11 System emulation scheme. 46
3.12 Search Time 48
3.13 Average Delivery Delay with 4 Edge Servers 50
3.14 Average Delivery Delay with 8 Edge Servers 50
3.15 SE energy consumption with 4 Edge Servers 51
3.16 SE energy consumption with 8 Edge Servers 51

4.1 Reference scenario: the transmitter communicates with the receiver on a pseudo-random channel through encrypted communication. 56
4.2 Adversary model: the adversary is challenged to identify the transmitter by resorting to RFF while exploiting (leaked) information associated with the DL-based RFF model of the transmitter on channel c_x. 56
4.3 Experimental testbed setup. 58
4.4 Alexnet accuracy confusion chart. Alexnet is trained and tested on 16 channels (one measurement per channel), using a test set of 40 images per channel. The confusion chart shows high accuracy when the test set is coming from the same channel as the training set (diagonal). 60
4.5 Maximum channel offset with misprediction greater than zero, considering various CNN. 61
4.6 RFF Accuracy at various channel distances $\delta \in [0, 15]$, using various CNN. . 61
4.7 Experimental Testbed—hardware and software components of the considered measurement setup. 64
4.8 BER (top) and Accuracy of RFF (bottom) as a function of the jammer frequency, with communication frequency at 1 GHz and jammer sweeping between 990 MHz and 1010 MHz. 66

4.9 BER (top) and Accuracy of RFF (bottom) as a function of the jammer frequency, with communication frequency at 2.4 GHz and jammer sweeping between 2395 MHz and 2405 MHz. 67
4.10 BER (top) and Accuracy of RFF (bottom) as a function of the jammer frequency, with communication frequency at 3 GHz and jammer sweeping between 2995 MHz and 3005 MHz. 67

5.1 5G 3GPP Lawful Interception architecture. 71
5.2 Technical workflow. 75
5.3 VoIP services implementation setup. 79
5.4 End-to-end file exchange implementation setup. 80
5.5 Intercepted Data. 81
5.6 Packet latency across the different LI stages of a 30 seconds VoIP call. 82
5.7 Statistics of the End-to-end LI latency phase per packet for the four different VoIP call durations. 83
5.8 Packet latency across the different LI stages of a 10^3 KB exchanged file. . . . 84
5.9 Statistics of the End-to-end LI latency phase per packet as a function of the four file sizes. 84
5.10 Packet End-to-End Latency of a 30 seconds VoIP call. 85
5.11 Packet End-to-End Latency of a 10^3 KB exchanged file. 85

List of Tables

2.1 Direct Social Factor rate . . . 13
2.2 Resource Capability Classes [38]. . . . 15
2.3 Services characteristics. . . . 15

3.1 Review of related works. . . . 25
3.2 Computational cost of cryptographic operations. . . . 32
3.3 Average Communication delays 35
3.4 Average Communication Latencies [114] [115]. . . . 47
3.5 Computational cost of cryptographic operations. . . . 47
3.6 Average Search Time Execution . . . 49

4.1 Communication settings parameters. . . . 65
4.2 Accuracy of RFF in interference-free scenarios. . . . 68

5.1 List of software and tools. . . . 79
5.2 Average delay difference experienced by each packet by deploying or not the proposed LI framework. . . . 86

List of Acronyms

3GPP 3rd Generation Partnership Project

5G Fifth generation mobile

5GNR 5G New Radio

5GCN 5G Core Network

ABE Attribute-Based Encryption

ABSE Attribute-Based Searchable Encryption

ADMF Administration Function

AI Artificial Intelligence

AMF Access and Mobility Management Function

AML Adversarial Machine Learning

AUSF Authentication Server Function

B5G Beyond 5G

BER Bit Error Rate

BPSK Binary Phase-Shift Keying

CA Certification Authority

CC Communication Content

C-LOR Co-Location Object Relationship

CNN Convolutional Neural Network

CP-ABE Ciphertext-Policy Attribute-Based Encryption

CSP Communications Service Provider

C-WOR Co-Work Object Relationship

DL Deep Learning

ES Edge Server

FGSM Fast Gradient Signed Method

GAN Generative Adversarial Networks

GDPR General Data Protection Regulation

gnB Next Generation Node B

HPC High-Performance Computing

IDBC ID-based Cryptosystem

IoT Internet of Things

IIoT Industrial Internet of Things

IM Instant Messaging

IMSI International Mobile Subscriber Identity

IQ In-Phase Quadrature

IRI Intercept Related Information

JWT JSON Web Tokens technology

KPI Key Performance Indicator

KP-ABE Key-Policy Attribute-Based Encryption

LEA Law Enforcement Agency

LEMF Law Enforcement Monitoring Facility

LI Lawful Interception

LICF Lawful Interception Control Function

LIPF Lawful Interception Provisioning Function

MDF Mediation and Delivery Function

MEC Multi-Access Edge Computing

ML Machine Learning

MQTT Message Queuing Telemetry Transport

NFV Network Function Virtualization

OOR Ownership Object Relationship

OTR Off-The-Record

PHY Physical

PKSE Public-Key Searchable Encryption

PLA Physical Layer Authentication

PLC Power Line Communication

PLS Physical Layer Security

POI Point of Interception

POSE Privacy-Oriented Search Engine

POR Parental Object Relationship

QoS Quality of Service

RFF Radio Frequency Fingerprinting

RF Radio Frequency

RGB Red-Green-Blue

RRC Root Raised Cosine

SDES Session Description Protocol Security Descriptions

SDN Software-Defined Networking

SDR Software Defined Radio

SE Searchable Encryption

SIP Session Initiation Protocol

SIoT Social Internet of Things

SIRF System Information Retrieval Function

SOR Social Object Relationship

SSE Symmetric Searchable Encryption

SRTP Secure Real-time Transport Protocol

TF Triggering Function

TKA Trusted Key Authority

TM Trust Management

TMS Trust Management System

TLS Transport Layer Security

UE User Equipment

UPF User Plane Function

VoIP Voice over IP

Pubblications

All the scientific contributions produced during the doctoral course are listed below.

International Journals:

- **Ingrid Huso**, Daniele Sparapano, Giuseppe Piro, and Gennaro Boggia, “Privacy-preserving data dissemination scheme based on Searchable Encryption, publish-subscribe model, and edge computing", Computer Communications (Elsevier), vol. 203, pp. 262-275, 2023.
- **Ingrid Huso**, Marco Olivieri, Leonardo Galgano, Adnan Rashid, Giuseppe Piro, and Gennaro Boggia, “Design and implementation of a looking-forward lawful interception architecture for future mobile communication systems,” Computer Networks (Elsevier), vol. 249, p. 110518, 2024.

International Conferences:

- Giancarlo Sciddurlo, **Ingrid Huso**, Domenico Striccoli, Giuseppe Piro, and Gennaro Boggia, "A Multi-tiered Social IoT Architecture for Scalable and Trusted Service Provisioning", Proc. of IEEE Global Communications Conference: Selected Areas in Communications: Social Networks (Globecom SAC SN), Madrid, Spain, Dec., 2021.
- **Ingrid Huso**, Giuseppe Piro, and Gennaro Boggia, " Distributed and Privacy-Preserving Data Dissemination at the Network Edge via Attribute-Based Searchable Encryption", Proc. of Mediterranean Communication and Computer Networking Conference (MedComNet), Paphos, Cyprus, may, 2022.
- Giuseppe Ungaro, Francesco Ricchitelli, **Ingrid Huso**, Giuseppe Piro, and Gennaro Boggia, "Design and Implementation of a Lawful Interception Architecture for B5G Systems Based on Key Escrow", 2022 IEEE Conference on Standards for Communications and Networking (CSCN)(CSCN’22), Thessaloniki, Greece, nov, 2022.

Under Review:

- **Ingrid Huso**, Savio Sciancalepore, Gabriele Oligeri, Giuseppe Piro, and Gennaro Boggia, "Frequency Matters: On the Impact of Carrier Frequency on Privacy in Radio Fingerprinting," in IEEE Wireless Communications Letters (submitted - under review).
- **Ingrid Huso**, Salvatore Carbonara, Savio Sciancalepore, Gabriele Oligeri, Giuseppe Piro, and Gennaro Boggia, “Jamming echoes: On the impact of Out-Of-Band interference on radio frequency fingerprinting,” in 2025 IEEE Wireless Communications and Networking Conference (WCNC)(WCNC 2025), Milan, Italy, Mar. 2025 (submitted - under review).

Scientific contribution

This work advances privacy-preserving techniques and security protocols for modern communication systems, addressing critical challenges in Internet of Things (IoT) and 5G network integration. By focusing on privacy, scalability, and trust, the research provides a deeper understanding of secure and efficient data exchange across interconnected ecosystems, fostering robust solutions for next-generation infrastructures. Foundational technologies such as Software-Defined Networking (SDN), Network Function Virtualization (NFV), Multi-Access Edge Computing (MEC), and network slicing enable dynamic, low-latency, and high-bandwidth infrastructures to meet the demands of IoT applications. However, their adoption introduces new challenges, particularly in securing data, managing trust, and ensuring privacy. Addressing these challenges requires innovative frameworks to enhance the reliability and security of modern networks.

To support secure interactions, a multi-layered, fog-enabled Social IoT Social Internet of Things (SIoT) architecture leverages trust and reputation management systems to evaluate device behavior and resource trustworthiness [1]. This decentralized approach ensures efficient service provisioning while reducing overhead and maintaining reliability. Additionally, entity virtualization has the potential to optimize resource orchestration, improving scalability in increasingly complex IoT environments.

In data-intensive environments, privacy-preserving frameworks based on Searchable Encryption (SE) enable secure keyword-based searches on encrypted data. Combined with a publish-subscribe communication model at the network edge, the proposed solution addresses critical privacy challenges in edge-computing scenarios, where data is distributed across resource-constrained and less secure components [2]. Experimental results demonstrate the efficiency of this approach, highlighting its suitability for lightweight, scalable, and secure data dissemination [3]. Optimizing edge server placement through advanced algorithms based on traffic load and processing dynamics will further enhance the framework's performance.

Moreover, Physical Layer Security (PLS) is explored with a focus on Radio Frequency Fingerprinting (RFF) for robust device authentication. Extensive experiments described in [4] and [5] using Software Defined Radio (SDR) validate the effectiveness of RFF-based techniques under multi-frequency operations and interference. These results demonstrate the feasibility of PLS as a complementary security layer, enhancing authentication and privacy directly at the hardware level. Addressing challenges such as interference and scalability will extend the applicability of PLS to more complex network environments.

In addition, to balance security and regulatory compliance, the work in [6] introduces an end-to-end encryption framework for Lawful Interception (LI) within modern communication networks. This cryptographic scheme enables secure data exchange, particularly for Voice over IP (VoIP) and messaging applications, while ensuring authorized access for regulatory purposes. The framework provides a balanced approach to privacy and compliance, paving the way for LI-compliant systems in multi-slice networks and edge-based infrastructures.

In conclusion, the research conducted in this thesis highlights the need for advanced security paradigms that integrate software and hardware approaches to address evolving challenges in privacy, trust, and Lawful Interception.

Introduction

The rapid evolution of digital communication technologies has transformed how information is shared, creating unprecedented opportunities and significant privacy challenges. With the advent of IoT, Industrial Internet of Things (IIoT), and Fifth generation mobile (5G) networks, communication systems have become increasingly interconnected and complex, facilitating high-speed data exchange and supporting diverse applications across sectors like healthcare, automotive, and industry [7]. The integration of IoT and 5G technologies is instrumental in establishing seamless connectivity among smart devices, forming the cornerstone of next-generation telecommunications, where key enabling technologies are essential in meeting the scalability and flexibility demands, thus fortifying the underlying architecture for an interconnected, intelligent network infrastructure [8].

In particular, SDN decouples network management from hardware, supporting programmability and dynamic control over network functions, essential for the scalability needs of IoT-driven 5G networks [9]. NFV enables traditional network services to be run as software on general-purpose hardware, reducing costs and accelerating the deployment of new services [10]. MEC brings computational resources to the network edge, allowing low-latency, data processing close to users, while isolating traffic from core network resources [11]. Finally, network slicing leverages virtualization to create multiple, isolated virtual networks over a single physical infrastructure, customizing resource allocation, quality of service, and security to meet the diverse application requirements within 5G environments [10].

However, these advances bring new challenges in protecting sensitive data from potential breaches and unauthorized access. As communication networks grow increasingly complex, the development of robust privacy-preserving techniques and protocols is essential to maintaining trust and security across diverse, interconnected systems [7].

Specifically, in large-scale IoT networks where heterogeneous devices independently establish social connections enhancing resource and service discoverability, the trustworthiness management of IoT nodes emerges as a crucial aspect [12]. Herein, the Trust Management System (TMS) functions as the module tasked with assessing the actions of social entities, automatically determining and updating their trust levels through systematic processes [13]. Therefore, a distributed IoT architecture ensuring the trustworthiness and availability of resources from service providers within the IoT environments is required.

Moreover, given the susceptibility of IoT devices in unsecured environments, malicious actors can intercept or compromise confidential data. Consequently, it is essential to secure data using strong cryptographic methods, directly applied by the data provider, by ensuring data privacy independently from the stored servers [14]. Thus, this underscores the significance of developing new approaches capable of enhancing security within the data dissemination process.

In addition, next-generation technologies pose the need for new security frameworks that go beyond traditional cryptographic approaches [10]. In response, PLS provides a range of innovative techniques, including advanced channel coding, secrecy coding for wiretap channels, privacy-protective transmission methods, secret key generation from shared randomness, physical unclonable functions for device authentication, and RFF for secure identification [15]. Specifically, RFF exemplifies a Physical Layer Authentication (PLA) technique that aims to verify the legitimacy of transmitting devices by examining inherent imperfections in their hardware components, which arise during the manufacturing process. These hardware-induced anomalies are expected to be unique to each device, rendering them exceptionally difficult for adversaries to imitate. Herein, in scenarios where IoT devices operate across multiple communication frequencies, it is essential to investigate the robustness of RFF with respect to variations in carrier frequencies.

Lastly, ensuring secure and reliable communication in modern systems necessitates robust end-to-end security, particularly for cross-trust-domain interactions in heterogeneous network environments [16]. In this context, whereas the pervasive adoption of end-to-end encryption significantly enhances communication security and privacy, it also complicates interception for authorized entities [17]. Addressing this challenge requires the development of advanced decryption techniques and the establishment of cooperative frameworks between telecommunications providers and Law Enforcement Agencies (LEAs).

This thesis aims to explore advanced security protocols and techniques designed for modern communication systems, focusing on their application within IoT and 5G networks. In detail, a brief summary of the thesis chapters is provided below. Chapter 1 introduces to 5G key enabling technologies and covers topics such as trust management, privacy preservation, physical layer security, and end-to-end encryption techniques tailored for 5G and beyond networks. Hence, Chapter 2 introduces an innovative multi-layered, fog-enabled SIoT architecture aimed at providing rapid service delivery, enhanced scalability, robust fault tolerance, and improved security. Additionally, Chapter 3 envisages a distributed and privacy-preserving data dissemination framework based on SE techniques and a publish-subscribe model at the network edge. Moreover, Chapter 4 investigates, through real experiments and the use of SDRs, (i) the critical dependence of RFF accuracy on the alignment between training and testing frequencies, and (ii) the impact of unintentional communication on the RFF accuracy. Then, Chapter 5 introduces an innovative LI framework integrating an end-to-end cryptographic scheme optimized for the interception at the application layer, offering a technological solution aimed at enhancing the 3rd Generation Partnership Project (3GPP) LI standard. Finally, the findings of this thesis work are summarized and potential future research directions are suggested.

Chapter 1

Introduction to Security in Modern Communication Systems

Integrating IoT technologies with 5G networks has catalyzed innovations across diverse sectors, including smart homes, autonomous vehicles, and critical infrastructure. With high-speed data transmission, ultra-low latency, and efficient power usage, 5G networks support billions of interconnected sensors, enabling real-time applications such as industrial automation and remote healthcare. Key enabling technologies within the 5G and emerging Beyond 5G (B5G) ecosystems (i.e., SDN, NFV, MEC, and network slicing) further enhance this framework by addressing scalability and flexibility requirements. As the convergence of 5G and IoT progresses, so does the imperative for robust security protocols to counter vulnerabilities in this increasingly complex network ecosystem. Ensuring the privacy and integrity of transmitted data remains a priority within the 5G-IoT landscape, fostering secure, resilient, and trustworthy communication.

1.1 Key Enabling Technologies in Modern Communication Networks

The proliferation of IoT services has had a profound impact across numerous industries, benefiting fields as varied as industrial systems, remote-controlled surgical equipment, connected vehicles, and essential infrastructure operations [7]. A key enabler of IoT advancements is the emergence of 5G networks, which have laid the foundation for connecting billions of sensors globally. Integrating 5G with IoT introduces high-speed data transmission capabilities, reduced energy consumption for low-power devices, and ultra-low latency (below 2 milliseconds), along with the seamless integration of diverse technologies and platforms [7]. Collectively, these features support the connection of billions of heterogeneous devices with sensors, providing the foundation for advanced services that span diverse applications.

In this rapidly evolving technological landscape, essential drivers within the 5G-IoT architecture are paving the way toward a fully connected environment. These technological advancements cover a broad range, including ultra-low latency communication systems that enable real-time applications such as autonomous vehicles and industrial automation. Additionally, energy-efficient designs are crucial to extending the battery life of IoT devices, which enhances device durability and helps to reduce operational costs over time. At the same time, the integration of edge computing, Machine Learning (ML), and Artificial Intelligence (AI) within the 5G-IoT ecosystem represents a paradigm change. Moreover, as the number of IoT devices expected to be incorporated into 5G networks grows exponentially, robust security protocols and scalable network structures are essential to protect data and manage this anticipated device influx effectively [7].

To support these advancements, the following key enabling technologies within the 5G and B5G ecosystem are necessary:

- **Software-Defined Networking** provides a centralized abstraction that enables programmability across the entire network. The primary goal of implementing SDN is to separate the control plane from network devices, such as switches, enabling an external network controller to dictate network functions and manage forwarding infrastructure, including routing and key management processes [9]. By decoupling these elements, SDN facilitates the deployment of new applications and enhances the flexibility of network management, which is essential for handling the exponential traffic growth anticipated in future mobile networks. As such, SDN plays a significant role as an enabler within the 5G-IoT environment [9].

- **Network Function Virtualization** transforms traditional network services, which historically required specialized hardware, by enabling these services to operate as software applications in cloud environments [10]. NFV uses general-purpose hardware, allowing network functions to be deployed dynamically on demand. This transformation reduces capital expenses (CAPEX) by minimizing reliance on proprietary devices and underutilized hardware. Furthermore, NFV accelerates the deployment of new services, shortening innovation cycles for providers, and supports multi-version and multi-tenant capabilities for network appliances, allowing a single platform to support various applications, services, and user groups simultaneously [9], [10].

- **Multi-Access Edge Computing** within 5G mobile networks enables localized cloud storage and computing resources at the network edge, facilitating data processing closer to the end-user [11]. By bringing computing applications, data processing, and analytics to the edge, MEC enables a network structure focused on data proximity, achieving extremely low latency, high data throughput, and improved intelligence and control [18]. Processing data close to the source allows for the exchange of analyzed insights instead of raw data, significantly reducing traffic and conserving network resources. Additionally, MEC isolates data traffic from the core network, decreasing the load on central resources and contributing to overall system efficiency [11].

- **Network Slicing** is a core feature of 5G networks, using virtualization to enable multiple logical or virtual networks to operate over a shared physical infrastructure. The primary purpose of network slicing is to manage physical resources effectively, grouping and isolating traffic according to the specific needs of different tenants and optimizing resources at a high level [19]. Each network slice is customized to support distinct use cases or operational requirements, and logical slicing divides a single physical network into several end-to-end virtual networks, maintaining strict isolation across each network in terms of access, transport, device, and core functions [10]. This approach allows for different service types to be delivered through dedicated virtual networks tailored to each application's specific requirements, including resource allocation, Quality of Service (QoS), and security [10].

These enabling technologies drive the capabilities of 5G and B5G networks, empowering the IoT landscape and advancing connectivity on a global scale. By harnessing SDN, NFV, MEC, and network slicing, the 5G-IoT ecosystem can meet the demands of a highly connected world, offering real-time, efficient, and secure services across diverse applications and industries.

1.2 Security Issues and Challenges

The advancement of mobile networks, previously discussed, aims to address the escalating demands for enhanced performance, portability, flexibility, and energy efficiency in emerging network services [7], as previously described. Whereas in earlier generations, security efforts primarily targeted billing integrity and basic user data protection, in the context of 5G, security must accommodate a far more complex ecosystem, one with an increasing number of connected devices and intricate service models [10]. Additionally, the 5G exceptional connectivity and the sensitive nature of IoT applications and mission-critical services in sectors such as healthcare and automotive heightens the need for robust protocols and data protection strategies overcoming specific security challenges.[7].

In particular, ensuring the integrity and confidentiality of data, protecting user privacy, securing network infrastructure, and countering sophisticated cyberattacks are all central concerns. To meet these challenges, robust encryption techniques, secure authentication protocols, and efficient intrusion detection systems are necessary. Furthermore, the exponential increase in IoT devices connected through 5G complicates efforts to secure an extensive and diverse array of interconnected devices [7].

Thus, addressing these core security and privacy concerns is essential to building a trustworthy and resilient 5G ecosystem capable of delivering on the promises of next-generation connectivity while upholding stringent privacy and security standards [7].

1.2.1 Trustworthiness Management

The swift growth of IoT applications across various areas has underscored the need to embed intelligence within IoT frameworks. Herein, it enables devices to make decisions and provide services autonomously, optimizing functionality and enabling innovative, intelligent services [20]. These advances, however, bring new security vulnerabilities, as malicious entities in IoT networks adopt increasingly sophisticated, intelligence-driven attack strategies that are difficult to detect. To address this, Trust Management (TM) has become an essential security approach. TM systematically evaluates the behavior of IoT entities over time, assessing their reliability to reduce potential risks [20].

Specifically, it is a set of techniques, strategies, and frameworks aimed at evaluating the reliability of IoT nodes [21]. Traditional TM models generally consist of three main stages: (i) gathering trust parameters, (ii) updating trust, and (iii) inferring trustworthiness [20]. In the first stage, the TMS collects relevant parameters either through direct assessment or indirect observations from neighboring nodes [21]. These parameters are chosen based on the specific context, such as node characteristics and the particular demands of IoT services. In the second stage, a trust value is calculated to reflect the node's reliability, using a predefined mathematical formula in the trust updating process [20]. This updating process further depends on two critical dimensions related to the: i) TMS architecture and ii) TMS basis. For the TMS architecture, trust management can be implemented through a centralized model, where a single node performs the trust evaluation, a distributed model, where nodes self-organize, or a hybrid model that integrates both centralized and distributed elements[20]. Moreover, for the TMS basis, trust management can operate on a time-based model, performing updates at regular intervals, an event-based model, triggered by specific occurrences, or a hybrid model combining both approaches [20]. Finally, the trust value is used to infer whether a node is reliable or not. If a node is deemed trustworthy, this result is recorded as part of its historical data to inform future trust updates [21].

Despite its critical role, existing TM frameworks exhibit limitations, such as continuous trust evaluation, limited metrics for service-based scenarios, and gaps in counteracting intelligent attacks [20]. Improving TM to integrate with service-oriented functions is vital to monitor trustworthiness at both local and network-wide levels, providing a robust defense against the rising threat of intelligent intrusions within IoT environments [20].

1.2.2 Privacy Preservation in Data Distribution

Privacy in 5G networks has become a crucial topic, as this technology is set to redefine access to digital services while introducing new challenges in data security, individual rights, and communication confidentiality [10]. Unlike previous mobile network generations, 5G introduces intricate, service-specific, and structural requirements, calling for rigorous privacy standards and regulations to foster trust among users and stakeholders [10].

The primary privacy concerns in 5G include data protection, location privacy, and identity management, each requiring unique protection, especially in sensitive domains like healthcare [7]. Specifically, the advanced capabilities of 5G enable highly personalized services, which necessitate different privacy levels depending on the service type. For example, location-based services continuously track users to enhance convenience but also introduce notable privacy risks related to real-time location tracking [10]. Thus, effective privacy solutions in 5G networks depend on several factors, such as adaptability, data management proficiency, and regulatory enforcement [10]. Organizations must conduct impact assessments and consider hybrid approaches—such as storing sensitive data closer to users at the edge cloud while keeping less critical data in central cloud storage [7].

In the context of 5G networks and IoT ecosystems, privacy is a critical concern due to shared environments and challenges in maintaining personal data ownership, particularly as information sharing can intensify data privacy risks [7]. The deployment of 5G supports shared network infrastructures or virtualized environments heightens the risk of unauthorized data access and unintended data exchanges, posing questions of accountability in cases of data loss [7].

Simultaneously, the proliferation of IoT devices has driven the need for secure data management as these devices generate significant volumes of sensitive data that must be carefully stored and processed [22]. To protect user privacy in cloud environments—often leveraged to handle this data, SE and other cryptographic techniques like homomorphic encryption are employed [22]. SE enables secure searches over encrypted data stored in the cloud, while homomorphic encryption allows computations on encrypted data without exposing its contents [22].

In particular, SE is a cryptographic technique enabling keyword search functionality directly over encrypted data, maintaining confidentiality while allowing data retrieval capabilities for authorized users [23]. This approach is critical for scenarios where data needs to remain secure yet accessible to entities beyond the data owner [23]. SE schemes employ various encryption techniques, including Symmetric Searchable Encryption (SSE), Public-Key Searchable Encryption (PKSE), and Attribute-Based Searchable Encryption (ABSE), each providing unique features and security assurances [24]. These schemes facilitate efficient and secure data querying in applications where confidentiality is paramount and ensure privacy-preserving storage and processing, addressing some of the critical security concerns in shared infrastructures of 5G and IoT ecosystems [22].

1.2.3 Physical Layer Security

The transition to 5G wireless communication, along with the advent of next-generation technologies, presents new challenges for the PLS community, necessitating security models that extend beyond conventional cryptographic methods [10]. To address these demands, PLS offers various innovative techniques, including advanced channel coding, secrecy coding for wiretap channels, transmission mechanisms that protect the user privacy, secret key generation from shared randomness, physical unclonable functions for device identification, and RFF for authentication [15].

In this context, PLA has garnered significant research attention due to its inherent security benefits. PLA allows for swift differentiation between legitimate and rogue transmitters directly at the physical layer, minimizing computational demands and reducing latency by bypassing upper-layer processing [25]. PLA is also highly adaptable, functioning effectively in heterogeneous systems where devices can interpret physical-layer signals even when they cannot decode each other's higher-layer data [25]. Importantly, PLA is intended as a complement to rather than a substitute for upper-layer authentication. For example, in two-factor authentication, PLA can confirm the authenticity of a user's device, while higher-layer methods can authenticate the user's identity—improving security in scenarios with varied device usage or distributed-antenna configurations [25].

PLA techniques are typically classified as either passive or active [25]. In passive methods, the receiver authenticates the transmitter by examining physical-layer properties, such as RF or channel characteristics. Conversely, active methods involve the transmitter embedding an authentication tag within the transmitted signal using a secret key, which the receiver then identifies to verify authenticity [25].

In particular, RFF represents a PLA approach that focuses on determining the authenticity of transmitting devices by analyzing unique imperfections in their hardware components, which are introduced during manufacturing [26]. These hardware-based anomalies are intended to be distinctive for each device, making them extremely challenging for an adversarial entity to replicate. RFF-based PLA can utilize either transient signal characteristics or stable features derived from the modulated In-Phase Quadrature (IQ) sample pairs [26]. Consequently, extensive research has been dedicated to evaluating the authentication efficacy of RFF techniques within this domain.

1.2.4 End-to-End Encryption against Lawful Interception

As mobile and wireless communication technologies evolve B5G, modern heterogeneous networks are expected to converge through the integration of diverse networking technologies. This integrated network environment introduces unique security challenges, particularly in establishing secure key agreements and preventing content theft during inter-domain communications [16]. To maintain secure and reliable communications, end-to-end security is critical, particularly for cross-trust-domain communications within these heterogeneous network environments [16].

In this context, today modern communication systems increasingly rely on applications such as Skype, Zoom, Telegram, and WhatsApp, which produce large volumes of multimedia content, including text, voice, and video. Consequently, the need for effective sensitive data protection systems has intensified to secure this diverse content. One approach is to implement an end-to-end encryption system that does not depend on any online services or centralized infrastructure. Many Voice over IP (VoIP) and Instant Messaging (IM) applications now claim to provide end-to-end encryption, which ensures that only the sender and designated recipient can access the contents of a message [27].

Within the scope of secure digital communication and private messaging applications, the Off-The-Record (OTR) protocol was developed to enable end-to-end encryption [28]. Although the OTR protocol has been available as a plugin for popular IM clients like Pidgin, its adoption has been limited due to usability challenges [29]. Public awareness around privacy concerns heightened after the Snowden disclosures, leading to the development of new encrypted messaging systems that address end-to-end encryption requirements by enhancing and adopting the OTR protocol [27]. To offer both end-to-end encryption and additional security features, such as forward secrecy and future secrecy, Open Whisper Systems introduced the Signal protocol. This innovative end-to-end encryption protocol supports both real-time (synchronous) and delayed (asynchronous) communication. The Signal protocol utilizes a key-distribution server to manage user identities and temporary keys, facilitating its use in both synchronous and asynchronous messaging contexts [30].

While the widespread adoption of end-to-end encryption greatly improves the security and privacy of communications, it also makes intercepting communications significantly more difficult for authorized entities [17]. In this context, the European Union has observed a marked increase in the activities of criminal networks engaged in cybercrime, terrorism-related offenses, and illicit trade [31]. To combat these threats, Law Enforcement Agency (LEA) relies extensively on LI tools to prevent, detect, and investigate criminal and terrorist activities. However, the continuous evolution of network architectures and associated services poses significant challenges to the development and implementation of advanced LI technologies. The advent of converged Beyond 5G network architectures, driven by novel key enabling technologies and communication paradigms, coupled with the pervasive adoption of robust end-to-end encryption mechanisms, necessitates innovative scientific and technical solutions [32]. When end-to-end encryption is employed, intercepted data can appear to LEAs as little more than a sequence of bits with minimal readable information. Consequently, balancing privacy and security with the needs of LI becomes critically important. Addressing these challenges effectively requires the development of robust decryption methods and the creation of cooperative frameworks between telecommunications providers and LEAs.

Chapter 2

Trusted Service Provisioning in Social Internet of Things

In the Social Internet of Things paradigm, the Trust Management System computes trust values of involved social objects, identifies trusted relationships, and selects the most suitable object able to provide a target service. State-of-the-art mechanisms conceived to address these tasks generally avoid considering the actual availability of social objects and demand the implementation of complex algorithms to constrained nodes.

The scientific literature already formulated different methodologies addressing these key functionalities. Most of the solutions, including those presented in [13], [33]–[37], implement the service provider selection without considering the availability of the actual resources. Consequently, requests may be frequently directed to social objects with higher trust values, favoring network congestion episodes and increasing latencies. Furthermore, some other valuable contributions expect to implement trust computation and service provider selection directly in the SIoT nodes [13], [33], [34], [37], [38]. Nevertheless, as explicitly highlighted [39], this represents an evident drawback for SIoT devices with limited storage and computation capabilities. Indeed, to the best of the authors' knowledge, the design of a more effective SIoT architecture able to jointly address these issues still represents an uncovered research goal.

To bridge this gap, the work presented herein conceives a novel multi-tiered SIoT architecture, where key functionalities are properly implemented to guarantee low latency, high scalability, fault tolerance, and security. Specifically, the lower level of the architecture embraces physical objects and their logical abstractions, exposing resources and services. The TMS entity, hosted at the first fog layer of the architecture, jointly addresses the trustworthiness of service providers under its control and the monitoring of the availability of resources exposed by related social objects. In this way, it can support an effective service provider selection without burdening on the constrained capabilities of social objects and preventing the network from blocks and slowdowns. Blockchain shares available services, relationships, and trust values across organizations and service domains at the second fog layer of the architecture. This allows to securely extend the boundaries of offered novel applications also at a large scale. The effectiveness of the proposed approach is investigated through computer simulations in a realistic SIoT scenario. The performance gain with respect to the baseline solution that does not leverage the conceived enhanced functionalities for the TMS is evaluated as well. Obtained results demonstrate that the proposed approach can serve incoming requests faster while guaranteeing a trusted and scalable service provisioning.

2.1 State of the Art on Trust Management Systems in the SIoT

The SIoT paradigm was recently born thanks to the promising integration of Social Network capabilities in the IoT domain [40]. By autonomously generating social relationships, smart objects can improve resource visibility, object reputation assessment, and service discovery [12][41]. Specifically, the social relationships are classified through different categories to promote trustworthy interactions in a service-oriented environment [40] [42]:

- Ownership Object Relationship (OOR): established among objects belonging to the same owner;
- Parental Object Relationship (POR): established among objects that are part of the same family and generally produced by the same manufacturer;
- Co-Work Object Relationship (C-WOR): established among objects working together for a common goal or in the same application;
- Co-Location Object Relationship (C-LOR): established among objects always located in the same place;
- Social Object Relationship (SOR): established among objects without common attributes or characteristics coming into contact because their owners come in contact or have a social relationship.

As a result, to generate any form of relationship, each social object must verify some conditions, such as the examination of the owner profile (OOR and POR), the geographical position (C-LOR), and the operational context (C-WOR and SOR). In a typical SIoT deployment, the TMS is the logical entity that evaluates the behavior of social objects and dynamically assigns them trust values through automatic mechanisms. Then, identifying trusted relationships supports the selection of the most suitable object able to supply a given request [43]. This latter task is referred to as *service provider selection*.

A social TMS in charge of evaluating and managing the trustworthiness of social objects was introduced, for the first time, in [13]. That study investigated a centralized architecture and identified its main deployment issues (e.g., single point of failure, low scalability). From now on, the scientific literature proposed many other SIoT system architectures, discussing the design of recommendation schemes based on the trust evaluation and defining different strategies aiming to offer an appropriate service provider selection through the TMS.

For example, the paper [33] faces the service provider search among nodes in a distributed manner with a new approach in a fast and autonomous way. The proposed strategy allows reaching the suitable provider, considering the energy constraints of nodes to increase the network lifetime. However, it neglects the aspects of load balancing, storage-saving, and the management of service requests that offer high scalability to the network. The work proposed in [34] defines functions and parameters to compute competence and willingness to quantify a trust value in a SIoT environment. Nevertheless, the entire algorithm computation for trust value is in charge of the IoT devices, not optimizing the computational loads. The contribution in [38] provides a scheme of access service recommendation for the SIoT, addressing both load balancing and network stability aspects. Here, within a distributed architecture, each node stores the profiles of the other nodes involved in the network. However, the nodes involved may not have sufficient storage capacity to keep track of the whole set of information needed. The authors in [35] propose a service-based grouping decentralized architecture for SIoT network as an approach to reduce the service discovery time. They exploit fog computing technology to boost the computational system capability. Nonetheless, the proposal does not provide any secure distributed storage technology for the management of social relationships.

The studies [37] and [36] present a blockchain-based trustful architecture for information spreading in SIoT environments. The described models provide a secure and transparent mechanism for trust evaluation. However, the first proposes efficient interactions to find the most suitable service provider in the network without considering any factor related to the

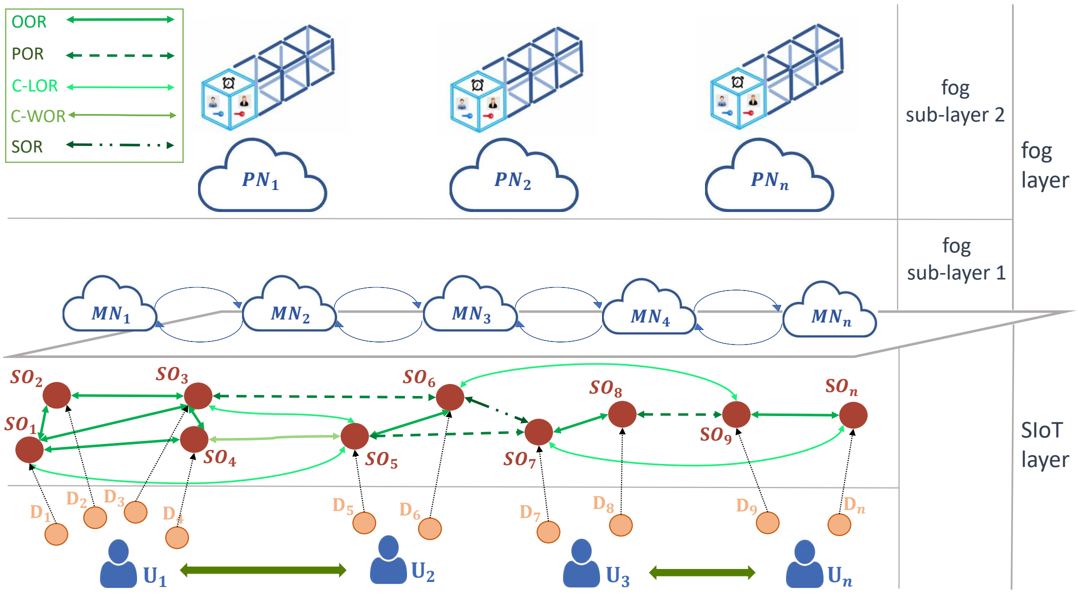


FIGURE 2.1: The proposed SIoT architecture.

employment of device resources. The latter, instead, presents an algorithm that exploits the information entropy to increase the system security but turns out to be effective only for specific time intervals.

Unfortunately, none of the studies discussed so far presents a well-defined paradigm that jointly embraces all the aspects of efficient resource management, scalability, and reliability of the Social Network, as well as trustworthiness and resource availability of service providers in SIoT environments.

2.2 The conceived Multi-tired SIoT architecture

Fig.2.1 depicts the novel SIoT architecture proposed in this work. It leverages a multi-layered decentralized configuration based on fog computing technology. Such a configuration allows for improving efficiency, increasing responsiveness, and reducing the computational loads of the network nodes by exploiting the higher computational capability of the fog nodes.

The lower layer of the architecture, namely SIoT layer, manages object virtualization. Social objects reproduce the digital counterparts of physical IoT devices while also collecting social skills not explicitly supported in the real world. The attributes that identify a social object and characterize its profile are:

- Device ID, which represents a device unique identifier;
- Owner ID, an identifier of the owner of the device;
- Manufacturer ID, useful to define the device manufacturer;
- Context, which indicates the type of task or service that the device can perform. A device can have more context-related identifier values, depending on the number of tasks/services it can accomplish.
- Resource capabilities, that indicates the resources a device can employ to provide services;

- Master node list, which indicates the set of master nodes responsible for managing all information related to the device;
- Friend list, which stores all the relationships identified by a social object within the Social Network.

Master nodes constitute the first sub-layer of the fog layer, namely fog sub-layer 1. The primary role of the fog sub-layer 1 is to perform the TMS for the management of service requests. A social object can act either as a service requester or a service provider. Moreover, to encourage service discovery, they are grouped into service communities according to the service they can provide. It is supposed that they can provide more than one type of service, thus belonging to more than one service community at the same time. In turn, each community is managed by a master node, which handles the Social Network of providers for the specific service, potentially generating a virtual topology for each service community. After establishing social relationships, a social object communicates to the network its availability to perform services. Accordingly, it searches an existing community characterized by the services that it can provide in the master nodes. If it cannot join any of the existing service communities, it creates a new one. Since the SIoT encompasses several services, a service-based grouping approach strongly minimizes the latencies in the service discovery procedure [35].

The fog sub-layer 2 interacts with the fog sub-layer 1 below. It deploys the primary nodes, characterized by high storage capacities, storing all the information related to social object profiles, social relationships, and reputations on a distributed database. Such information pool is hosted on a Blockchain, enabling privacy and security for the stored information defining the SIoT environment. Adopting a Blockchain in this type of framework ensures strong and secure traceability of the nodes, supporting the identification process of the most suitable social object to provide a service with a high degree of trustworthiness. Furthermore, such a hierarchical, distributed, and decentralized approach turns out to be of fundamental importance for the TMS execution since it allows increasing scalability and efficiency. Indeed, supposing that the information related to the reputation of a service requester is not available at the master node, it can be anyhow retrieved from the Blockchain on the primary node.

2.3 Overview of the Proposed Trusted Service Provisioning Process

The fundamental objective of the proposed architecture is to improve network navigability and boost the service search process. This is done by carefully considering service communities and social relationships settling the Social Network of objects. Most scientific works in this context perform this task by focusing on service providers' trustworthiness, mainly associated with evaluating the users' behavior. Differently, the strategy proposed herein goes one step further by jointly investigating the service trustworthiness and resource consumption assessment, thus promoting the service discovery beyond reliability and security.

Figure 2.2 shows the overall service provisioning procedure. A social object issues a service request and sends it to the closest master node. If the master node does not manage the related service community, the request is forwarded to the master node able to process the request. Through this procedure, the suitable service provider identification would not be restricted to the limited knowledge of the requester or the fog node directly connected to it, allowing a global view of each service provider's trustworthiness. Through the trust model and the resource management functionality (further described below), the TMS determines a trust ranking of providers among the social objects that can potentially provide the requested service. Then, the most suitable provider in the ranking is selected for the service execution.

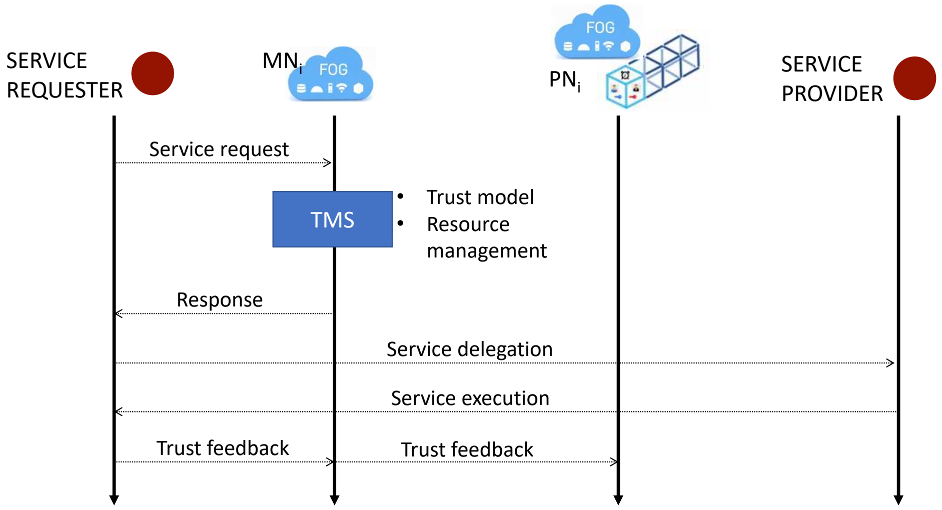


FIGURE 2.2: The trusted service provisioning process.

Finally, the service requester provides to the system its degree of satisfaction for the service received. The feedback is expressed with a value equal to 1 for service accomplished. A value equal to 0, instead, is given for a service not correctly completed. Both the fog sub-layers store the feedback for subsequent evaluations of the trust level.

2.3.1 Conceived Trust model

Given the i-th object requesting a service s_k and the j-th object exposing a service, the TMS calculates the trust value $T_{s_k}(i, j)$. In summary, $T_{s_k}(i, j)$ is defined through two main factors, which are the sociality factor and reputation.

The sociality factor, $S_f(i, j)$, rates the relationship established between the considered social objects by describing the degree of confidence in the case of both direct and indirect friendship (e.g., a friend of friends). In the case of direct friendship, it is set as $S_f(i, j) = SO_{ij}$, according to the type of social relationship (see Table 2.1). In indirect friendship, instead, it is evaluated by considering the social objects' common friends. Precisely, assuming that the number of i and j common friends is equal to C, $S_f(i, j)$ is computed as: $S_f(i, j) = \frac{\sum_{c=1}^{C} SO_{jc}}{C}$, where SO_{jc} represents the direct social factor rate between j and its common friends with i.

On the other hand, the reputation, $R_{s_k}(i, j)$, represents the opinion on the trustworthiness of a service provider for the service s_k, based on past experiences through feedback values assigned to previous interactions among social objects. It is calculated as a linear combination of three different contributions:

- the direct feedback $\Delta_{s_k}(i, j)$ describes how the i-th requester evaluated the j-th provider for the service s_k in the past;

TABLE 2.1: Direct Social Factor rate

Type of relationship	POR	OOR	C-LOR	C-WOR	SOR
SO_{ij}	0.9	0.8	0.7	0.6	0.5

- the indirect feedback $\Theta_{s_k}(i,j)$ describes how the friends of the i-th requester evaluated the j-th provider for the service s_k in the past. Assuming that the considered requester has F friends, $\Theta_{s_k}(i,j)$ is computed as:

$$\Theta_{s_k}(i,j) = \frac{1}{F}\sum_{f=1}^{F}\Delta_{s_k}(f,j), \tag{2.1}$$

  where $\Delta_{s_k}(f,j)$ is the feedback given by the f-th friend of the i-th requester.

- the indirect non-friend feedback $\Pi_{s_k}(i,j)$ specifies how the other non-friend social objects evaluated the j-th provider for the service s_k. Assuming that the total number of non-friends that have previously evaluated the provider j is equal to P, $\Pi_{s_k}(i,j)$ is computed as:

$$\Pi_{s_k}(i,j) = \frac{1}{P}\sum_{\pi=1}^{P}\Delta_{s_k}(\pi,j), \tag{2.2}$$

Finally, the reputation factor is obtained as:

$$R_{s_k}(i,j) = \alpha\Delta_{s_k}(i,j) + \beta\Theta_{s_k}(i,j) + \gamma\Pi_{s_k}(i,j), \tag{2.3}$$

where α, β, and γ are the weights ($0 < \alpha, \beta, \gamma < 1$ and $\alpha + \beta + \gamma = 1$) that determine the relevance for each factor considered in the evaluation of the reputation.

To conclude, the trust value associated with the i-th object requesting a service s_k and the j-th object exposing a service is obtained as:

$$T_{s_k}(i,j) = S_f(i,j) \cdot R_{s_k}(i,j). \tag{2.4}$$

2.3.2 Resource management evaluation

Leveraging the trust model, it is possible to recognize trusted social objects, discarding all providers below a configured threshold from the service provider selection. Moreover, by determining the trust value for each service provider, the master node obtains a ranking based on social object reliability.

Besides, the proposed TMS considers a further investigation addressing the resource capability of social objects. This contribution is restrictive for the trust evaluation, especially in an environment constituted by nodes with limited resources. In fact, in the case of several service requests assigned to the same social object, which entertains low resource capability, it would increase the risk of skipping the execution of the service due to a lack of available resources. As a matter of fact, its opportunity to provide the requested service decreases, causing possible congestion in the network. Hence, the master nodes monitor the status of the social objects and the resources required for the service execution. Precisely, after the ranking computation, the resource capacity of the candidate provider is monitored to verify the social object availability for the service execution. If this check fails, the candidate provider is temporarily dropped from the list. The master node updates the ranking and performs the same investigation on the new candidate until the service provider that meets the required resource consumption to execute the service is found.

TABLE 2.2: Resource Capability Classes [38].

Social object class	Resource Capability
Smartphone	0.8
Smart gateway	0.6
Smart camera	0.4
Sensor	0.2

TABLE 2.3: Services characteristics.

Service ID	1	2	3	4	5	6	7
Resource Consumption	0.3	0.2	0.2	0.1	0.1	0.2	0.1
Execution Time [s]	2	7	3	7	2	8	5

2.4 Performance Evaluation

The performance of the proposed SIoT is investigated herein through computer simulations. To this end, a MATLAB script is used to model a Social Network with heterogeneous objects, and various traffic loads, together with all the procedures described in section 2.3.

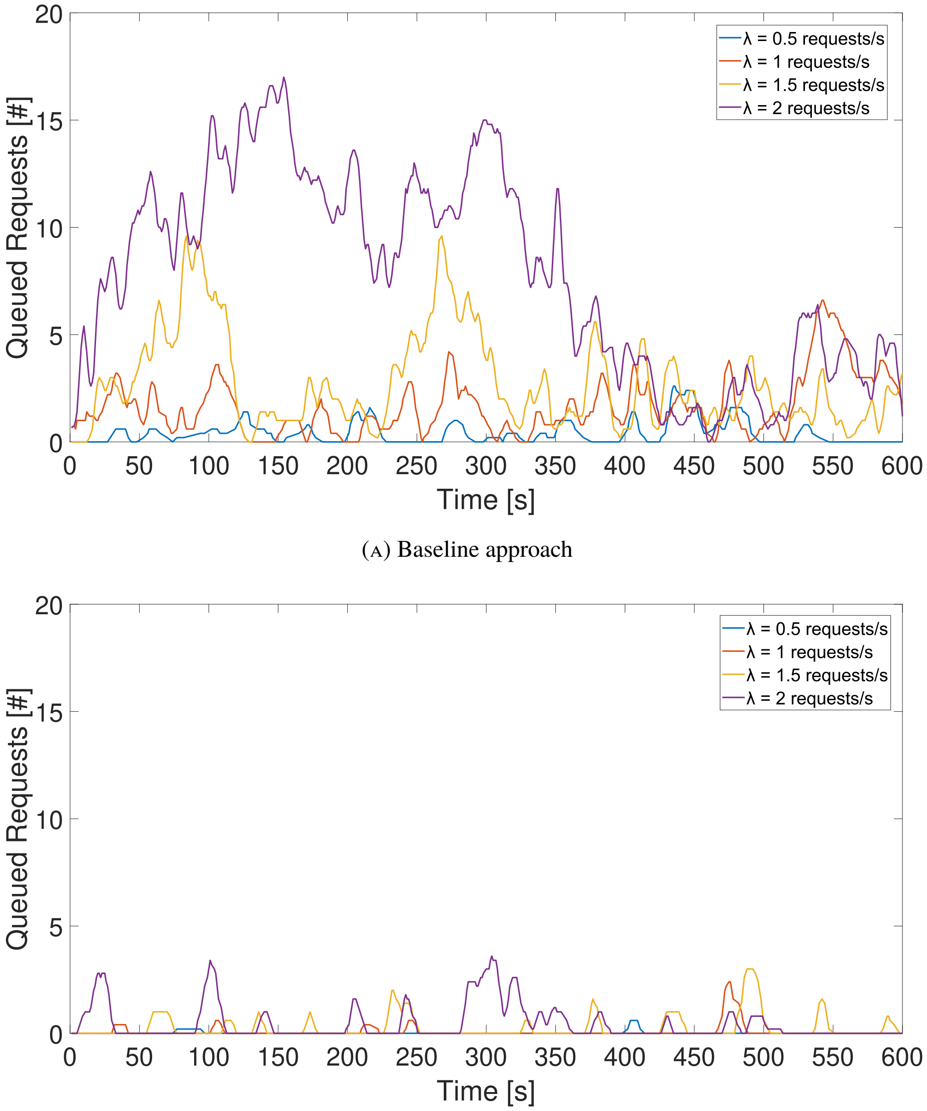
The proposed scenario considers a fog layer composed of five master nodes coordinated by a primary node for service request management. The number of social objects ranges from 50 to 300. They are uniformly distributed among four computing classes (smartphones, smart cameras, sensors, and smart gateways). A social object randomly generates a service request according to a Poisson distribution with different λ values (from 0.5 to 2 requests/s), to simulate different traffic loads. According to what is described in Section 2.2, each social object is characterized by an ID, an owner ID, a manufacturer ID, the geographical position, a list of services it can provide, and its resource capability. In line with [38], resource capabilities are set as summarized in Table 2.2. The Social Network is created by considering POR, OOR, and C-LOR relationships, based on the knowledge of the owner, manufacturer, and geographical position attributes, respectively.

Seven different types of service communities are configured and distributed among all the master nodes. Each social object joins the proper service community handled by a master node, following the procedure explained in Section 2.2. As reported in Table 2.3, each service is identified by an ID, the resource consumption needed to be accomplished (spanning from 0.1 to 0.3), and the execution time (spanning from 2 s to 8 s).

Finally, the performance of the proposed approach is compared with the baseline architecture, where the TMS calculates the trustworthiness of social objects by taking into account relationships and reputation parameters without any resource control.

2.4.1 Simulation results

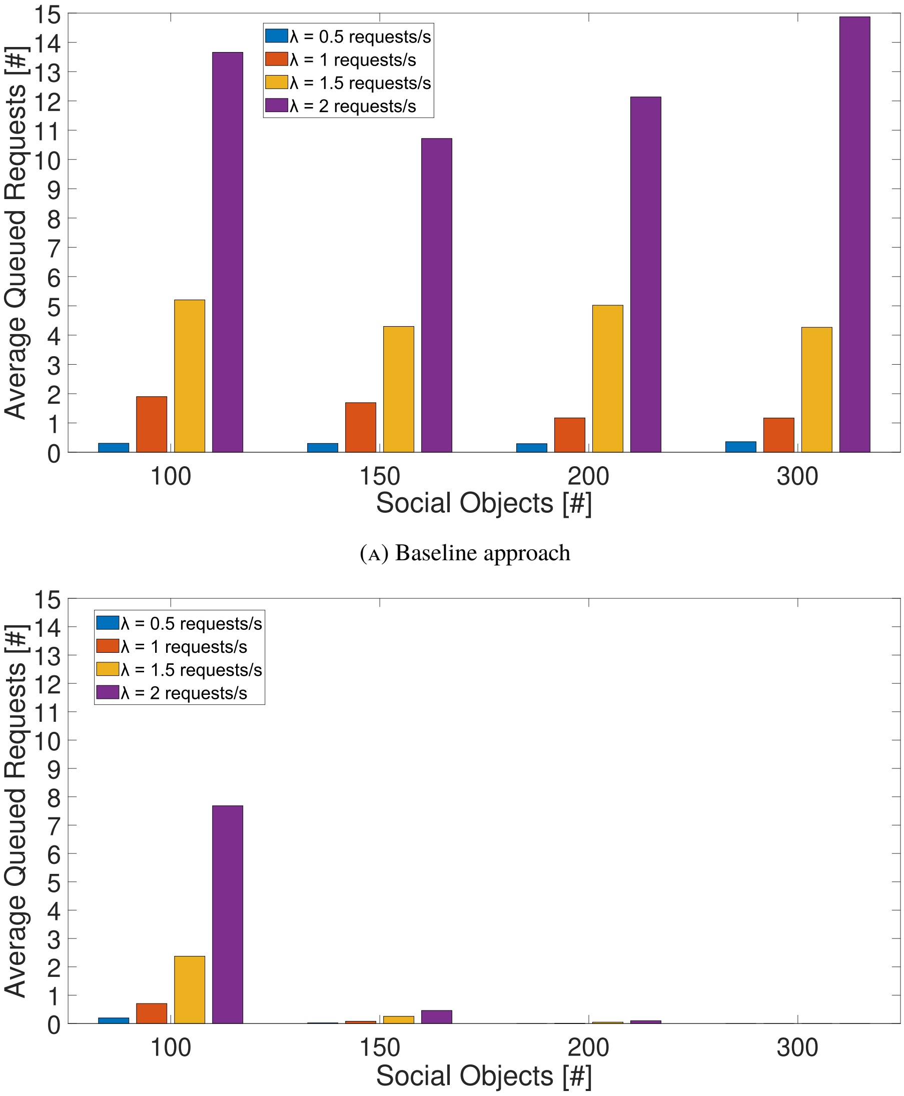
Figure 2.3 shows the number of queued requests during the time for different traffic loads. The results are obtained for a single scenario conducted on a social network with 150 social objects. For each λ, results are obtained over ten simulation runs to account for different network topologies and service distributions, and are averaged on a five-second time window sliding by one second. Herein, the network topology refers to the number of SIoT objects in the network and the service distribution is related to the request per second generated by each social object. Reported curves highlight the ability of the proposed approach to handle most of the requests in real-time: service requests are distributed to available trusted objects while preventing unpleasant queuing phenomena. On the contrary, the baseline approach distributes the requests without considering resource availability. As a consequence, most requests are relegated to a small set of trusted social objects, which monopolize scheduler assignments and overload their available resources in a short time. Therefore, a service provider selected to run a service by the TMS may not have sufficient resources, contributing to the formation of a queue of pending requests and, in turn, to a latency increase.



(A) Baseline approach

(B) Proposed approach

Figure 2.3: Queued Requests Evaluation.

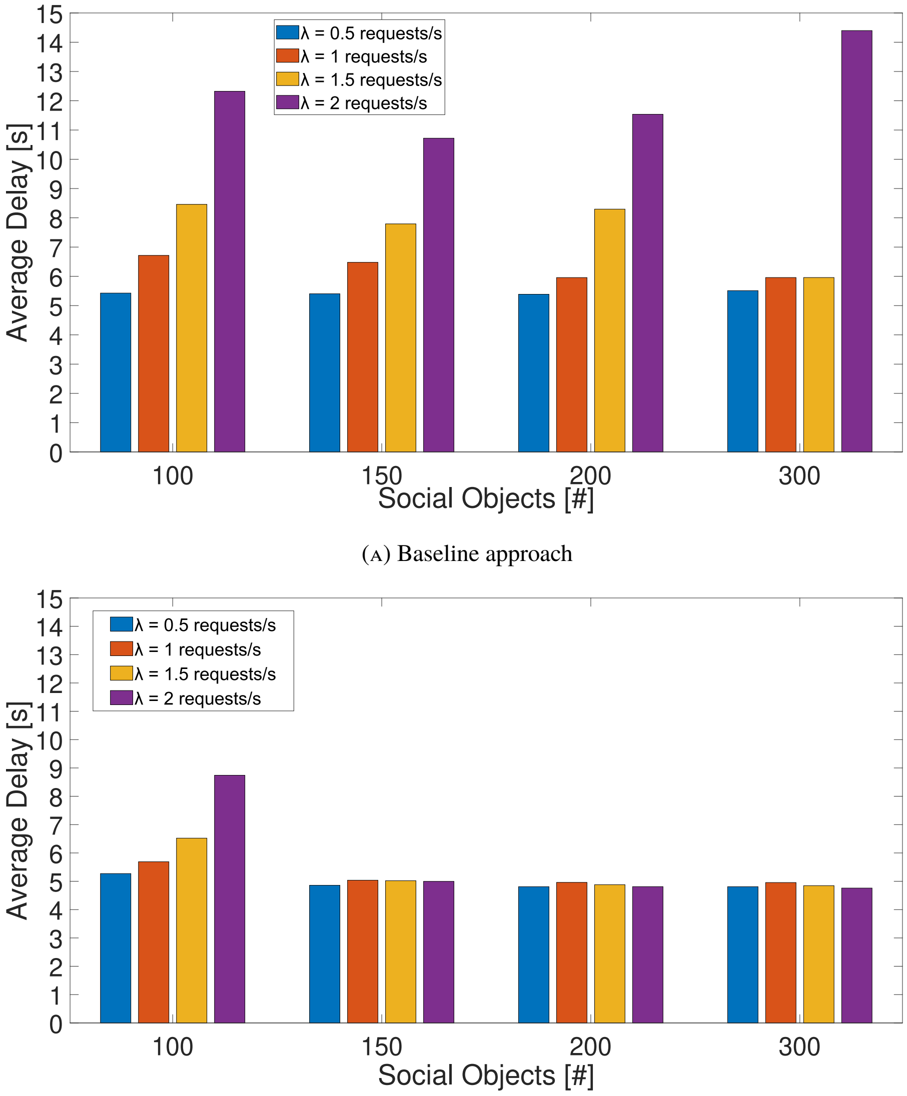


(A) Baseline approach

(B) Proposed approach

FIGURE 2.4: Queued Request increasing traffic load.

In order to generalize the afore-discussed findings, Figure 2.4 shows the average queued requests for different traffic loads. Resource management allows minimizing the number of pending requests, thus improving network scalability. On the contrary, in the baseline approach, the network fails to handle large amounts of traffic, testifying the increase of queued requests and, consequently, the average delay in accomplishing them all. Moreover, the average queued requests scheduled in the proposed TMS on the data plane decrease when the number of nodes increases. It demonstrates the explicit scalability improvement differently from the baseline approach. Indeed, this allows the network to react effectively to a substantial traffic increase (i.e., from 0.5 requests/s to 2 requests/s) without overloading the resources of social objects.



(A) Baseline approach

(B) Proposed approach

FIGURE 2.5: Average Delay increasing traffic load.

Figure 2.5 depicts the average delay experienced for a request in the proposed and the baseline approach. Delay does not consider the time needed to exchange control messages or interactions between master nodes, since it is negligible if compared to the service execution time. Thanks to the intelligent management of the available resources of the social objects, the delay performance of the proposed approach is almost the same for high numbers of social objects, also outperforming the baseline approach, especially for high traffic loads. Unlike the baseline approach, the proposed test does not reveal any performance decay in terms of average delay in fulfilling requests. In fact, by increasing the number of nodes and the request rate λ, the average delay is still constant. The variations between the two approaches are more visible for most populated configurations. In fact, considering 300 social objects and an average request rate equal to 2 requests/s, the average delay experienced reduces by up to 60%

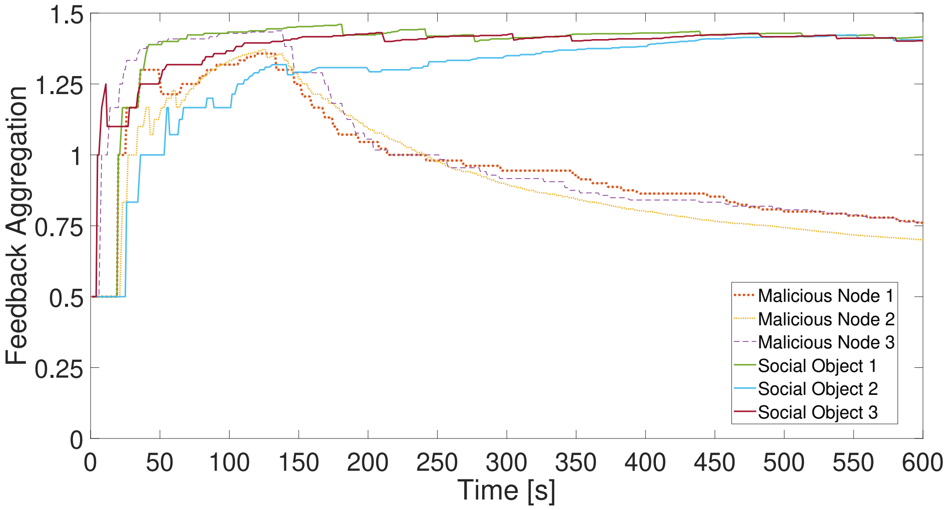


FIGURE 2.6: Malicious social objects detection.

Finally, to provide further insight, Figure 2.6 shows the evolution over time of the feedback aggregation for six service providers. Three of them have been forced to act as malicious nodes, not accomplishing a service after a request and receiving negative feedback accordingly. Indeed, this result testifies to the capability of the conceived TMS to detect potential malicious nodes. In fact, after a warm-up period of about 100 s, the identification of malicious nodes appears unmistakable. Since the proposed model is reputation-based, a trustworthiness reduction due to negative feedback causes the system to choose only the trustworthy social objects for the scheduled requests in the service provider selection.

Chapter 3

Privacy-oriented Data dissemination

With the proliferation of the IoT and IIoT ecosystems, secure and privacy-preserving data management has become essential. In these environments, seamless connectivity and autonomous management provide numerous societal benefits, yet also introduce critical security and privacy concerns, particularly regarding sensitive data protection. Given the vulnerability of IIoT devices in potentially unsecured environments, there is a substantial risk of data breaches and unauthorized access [14]. To mitigate these risks, advanced cryptographic techniques like Attribute-Based Encryption (ABE) and SE are crucial in protecting data from unauthorized access and curious cloud entities that may attempt to view confidential information [44].

ABE is widely regarded as an essential technique for access control, as it allows data to be encrypted in a manner that limits access strictly to authorized users based on specific attributes [45]. By encrypting data at the source, ABE prevents unauthorized entities—including potentially "honest but curious" clouds—from accessing sensitive information, thus safeguarding data privacy independently of where the data is stored (e.g., in remote clouds or at the network edge) [14].

In addition, SE supports data protection while enabling efficient keyword searches over encrypted data, making it particularly suitable for IIoT applications where data storage and retrieval must be both secure and functional. SE allows the data owner to encrypt and store data in the cloud, and authorized users can retrieve specific information by issuing a keyword query. This keyword-based retrieval process ensures that only relevant encrypted data is delivered, while maintaining confidentiality [23]. Together, ABE and SE provide a promising solution for privacy-preserving data management in IIoT environments by enabling secure access and search functionality without exposing sensitive information to unauthorized parties.

However, despite the promise of ABE and SE, most current solutions focus on single cryptographic operations and primarily rely on cloud-based architectures. These approaches often overlook the distributed nature of IIoT systems, which involve multiple data producers and consumers. Existing methods typically centralize data storage and search functions in the cloud, leading to high computational loads and increased end-to-end communication latency [46] [47]. Addressing these challenges with privacy-centric, distributed solutions that leverage edge computing remains an open research area, with the potential to improve efficiency, reduce latency, and ensure privacy-preserving data sharing within the IIoT network.

3.1 Background concepts and literature review

This Section presents background concepts and reviews the state-of-the-art on Searchable Encryption.

3.1.1 Background concepts

MEC. The hugely powerful performance requirements of 5G and B5G networks motivated the diffusion of the Multi-access Edge Computing (MEC) concept, which optimizes the spatial layout of network applications and services by utilizing pervasive computing, communication, and storage resources at the network edge [48] [49]. According to ETSI-MEC specifications [50], each MEC host may integrate multiple MEC applications that, by taking advantage of strong tools and computational resources, are able to process (e.g., data mining and fusion) heterogeneous data produced by IoT devices as well as to offer cutting-edge and specialized services closer to the end-users.

Attribute-Based Access Control (ABAC). The National Institute of Standards and Technology (NIST) formulates a concrete solution for fine-grained authorization in dynamical IoT scenarios [51]. The ABAC methodology assumes that any resource is protected via dedicated access control policies, defined as a combination of access grants and properties. Indeed, to access a specific resource, an end-user must prove to possess a subset of attributes satisfying the access control policy uniquely associated with the resource. Some noteworthy cryptographic algorithms directly include ABAC logic into the encryption and decryption procedures. ABE, Key-Policy Attribute-Based Encryption (KP-ABE) [52], and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [53] are among them. Indeed, these strategies may be used in IoT domain to provide data protection as well as flexible access control.

Searchable Encryption (SE). The growing number of IoT devices creates amounts of data to be collected and processed by means of computing and storage services (e.g., the Cloud). Indeed, to ensure privacy, the acquired data are encrypted before being stored in the public cloud [22]. However, despite the many advantages that cloud storage offers, protecting the privacy of sensitive data remains a difficult problem since the cloud servers are considered to be honest but curious [24]. This indicates that, although cloud service providers can be trusted for their services, they may also be interested in the data of their customers.

In this context, Searchable Encryption (SE) emerges as a preliminary turning point [44]. In cloud computing environments, SE offers a useful solution for issuing search queries on encrypted files based on specific keywords. Specifically, SE systems are constructed on a client/server architecture, in which the data owners and consumers serve as the clients during storage and retrieval, while the cloud acts as the server [24]. The data owner is responsible for outsourcing a collection of data and a list of keywords in an encrypted form [24]. The data user is authorized to retrieve data from the cloud by sending encrypted queries, namely Trapdoors, to the Cloud. The cloud server saves the documents submitted by the data owner and also handles search tasks: when a data user submits a Trapdoor, the cloud searches across the encrypted keywords and returns to the data user the documents that contain that specific keyword [24]. At the end, the data user can decrypt the received data [24].

3.1.2 Related works on Searchable Encryption in the IIoT

In the IIoT context, recent studies [44] and [54] declare that devices, networks, and application vulnerabilities are affecting everyday life and the overall industrial sector, raising the need to enhance privacy, data security, and access control. This highlights the importance of providing new methodologies able to improve the security in the data transmission flow.

A concrete solution offering fine-grained authorization, namely Attribute-Based Access Control (ABAC), has been formulated by the National Institute of Standards and Technology (NIST) [51]. The ABAC logic assumes that any resource is protected by means of dedicated access control policies, defined as a combination of properties/access grants. To access a specific resource, an end-user must prove the possession of a subset of attributes that satisfies the access control policy uniquely coupled with the resource. Some interesting cryptographic mechanisms integrate the ABAC logic directly within encryption and decryption processes. They include ABE, KP-ABE [55], and CP-ABE [56]. Indeed, these techniques can be used in the IIoT to jointly offer robust data security and flexible access control.

Regarding data dissemination, most of the available solutions (i.e., proposed in the scientific literature or implemented and ready to be used) leverage cloud-based approaches: data are distributed via remote clouds [57] [58] [59]. In these cases, however, to correctly deliver data to legitimate end-users, the server should know something about data sources, service type, end-users, and so on. If on the one hand, this can be an evident problem from the privacy perspective, from another hand this methodology is unfeasible in the presence of data protected with ABE.

SE emerges as a preliminary turning point [44]. In cloud computing environments, SE offers a useful solution for issuing search queries on encrypted files based on specific keywords. The work presented in [60] represents the first SSE scheme where the symmetric key encryption method is used to build the searchable ciphertexts and to allow users to generate trapdoors through the shared key. Later, the contribution in [61] integrates keyword searching with public key encryption techniques, allowing users to securely recover the requested files over encrypted data using user-defined keywords. The PKSE works with both public and private keys enabling data owners and users to do encryption with their public keys and produce trapdoors with their private keys. Following that, the scientific literature presents numerous PKSE systems with various capabilities, such as single keyword search [62] [63], fuzzy keyword search [64], verified keyword search [65], and ranked keyword search[66].

However, the mentioned SE systems do not allow data owners to give end-users fine-grained search capabilities. Indeed, studies [67] and [68] have recently looked at the integration of ABE and SE systems. Nevertheless, these approaches can only be utilized to find a particular keyword, limiting the flexibility and accuracy of data retrieval. Thus, works in [69], [70], and [71] have also looked into attribute-based multi-keyword search algorithms. Moreover, [72] suggests an enhanced ABE method with multi-keyword search to facilitate simultaneous numeric attribute comparison, hence significantly increasing the flexibility of ABE encryption in a dynamic IoT context.

As far as IoT is concerned, novel lightweight SE approaches are proposed in edge and fog computing environments since there are considered promising solutions able to bring data storage and computation capabilities closer to IoT devices [73]. Indeed, [74] envisages a dynamical SE process with a multi-keyword search for smart grids in a cloud-edge architecture where the search algorithm is running through a cooperation between the edge nodes and the cloud server. While, the studies in [75], [45], and [76] introduce three different SE schemes in fog-based IoT scenarios, where fog nodes both help the cloud in the searching and forwarding process and partially decrypt the retrieved documents in order to reduce the computational workload on IoT devices.
Recently, the work in [47] introduces a distributed SE scheme in the healthcare domain, demonstrating the capability of fog nodes to decrease the computational workflow with respect to the cloud environment. Nevertheless, it provides fog nodes with cryptographic capabilities to partially decrypt and encrypt searched queries.

3.1.3 Literature review on Searchable Encryption in the IoT

Recently, several research works in the field of IoT [77] [78] [79] emphasize how the proliferation of smart devices results in huge amounts of data being exchanged through the network, reducing its security and raising the need to improve privacy and data protection. This underlines the significance of introducing new approaches capable of improving data transmission security.

In this context, the majority of existing solutions, present in the scientific literature, rely on cloud-based approaches where data is disseminated via remote clouds [59] [58] [57]. Herein, however, the server needs to be aware of the data sources, service type, end-users, and other relevant information in order to correctly transmit data to authorized end-users.

Thus, Searchable Encryption (SE) appears as a key enabling technology for providing privacy preservation, [44]. In cloud computing contexts, SE represents a valuable option for performing search queries on encrypted files based on specific keywords [22]. The work provided in [60] constitutes the first SSE system in which the searchable ciphertexts are built using the symmetric key encryption approach, and users are able to create trapdoors using the shared key. Later, the contribution in [61] combines keyword searches with public key encryption methods, enabling users to safely recover the requested files over encrypted data using user-defined keywords. By utilizing both public and private keys, the PKSE allows data owners and users to encrypt data with their public keys and create trapdoors with their secret keys. After that, a variety of PKSE schemes with different capabilities are presented in the scientific literature, including single keyword searches [62], fuzzy [63] [64] and ranked keyword searches [66] as well as verified ones [65]. Since the aforementioned SE solutions are not feasible for bringing fine-grained search capabilities to end-users, recent studies [67]–[72], [80] have recently looked at the integration of ABE schemes and SE techniques, significantly boosting ABE adaptability in a dynamic IoT environment. Herein, to guarantee high efficiency of encrypted data retrieval from cloud servers, the works in [81] [82] introduce a multi-user SE scheme, while, recent several works [83], [84], and [85] propose multi-owner scenarios. In IoT scenarios, new lightweight SE techniques are offered in edge and fog computing contexts as potential options for bringing data storage and processing capabilities closer to IoT devices [22]. Indeed, the studies in [75], [45], [76], and [86] describe four distinct SE schemes in fog-based IoT scenarios, where fog nodes help the cloud with searching and forwarding while also partially decrypting the documents that are retrieved to lessen the computational load on IoT devices. Moreover, [74] and [87] envision dynamical SE schemes with a multi-keyword search for smart grids in a cloud-edge architecture and edge computing respectively, where the search algorithm runs in collaboration between the edge nodes and the cloud server. While, recent works (such as [47] [88], [89], [90], and [91]) offer distributed SE methods in different IoT domains, proving the possibility of fog and edge nodes to reduce the computational workload with regard to the cloud environment. Nevertheless, they equip fog or edge nodes with cryptographic capabilities to partly decode and encrypt searched queries.

Problem Description

The review of related publications reveals that the present scientific literature focuses on the cryptographic features of the SE schemes, with the main purpose of finding and retrieving specific encrypted files. Furthermore, from the summary reported in Table 3.1 it is possible to conclude that:

- Available studies investigate the computational complexity of SE operations as a function of security parameters (i.e., the number of attributes forming the access policy) and the number of files (i.e., the encrypted data) to be processed. Thus, none of them evaluate SE in realistic scenarios where coexist heterogeneous data producers and end-users.

TABLE 3.1: Review of related works.

Works	Multi-user	Multi-owner	Edge/Fog Computing	Privacy-preserving	Searchable Encryption at the network edge	Data dissemination	Cryptographic Security proof	Protocol Security proof	Publish-Subscribe model
[14]	✓			✓			✓		
[59]				✓			✓		
[46]			✓	✓			✓		
[47]			✓				✓		
[67]				✓			✓		
[70]				✓			✓		
[72]	✓			✓			✓		
[80]	✓			✓			✓		
[81]		✓		✓			✓		
[82]	✓			✓			✓		
[83]		✓		✓			✓		
[84]		✓		✓			✓		
[85]			✓	✓			✓		
[75]			✓	✓			✓		
[45]			✓	✓			✓		
[76]			✓	✓			✓		
[86]			✓	✓			✓		
[74]			✓	✓			✓		
[87]			✓	✓			✓		
[88]			✓	✓			✓		
[91]			✓	✓			✓		
This Work	✓	✓	✓	✓	✓	✓	✓	✓	✓

- Most of the existing works leverage a cloud-based approach, where search and dissemination tasks are directly implemented by the remote cloud. In recent works, computational capabilities at the network edge have been used to implement encryption and decryption operations, thus limiting the complexity expected for constrained devices. However, the chance of performing SE operations directly at the edge of the network has not yet been investigated.

- No contributions envisage the opportunity of sharing and disseminating data through the network, and in particular at the network edge, in a distributed, efficient, and privacy-preserved manner by using SE schemes and publish-subscribe model.

3.2 Distributed and Privacy-Preserving Data Dissemination at the Network Edge via Attribute-Based Searchable Encryption

As well known, the IoT paradigm allows seamless connectivity and autonomous management in heterogeneous environments (without human interaction) and provides several important societal services via completely intelligent and automated systems [92]. The IIoT, also known as Industry 4.0, further improves user experiences and promises to develop new profit streams by leveraging IoT device capabilities and data processing/analytics in the industrial domain. At the time of this writing, IIoT allows to connect smart devices and sensors to construct autonomous systems that gather, exchange, and analyze real-time data, while delivering important insights to enhance efficiency, security, and energy usage in the industry [57].
In conjunction with its development, IIoT is facing several security problems. The first one refers to the privacy protection issue. Due to the vulnerability of IIoT devices in an unsecured environment, malicious users can steal or breach sensitive data. Therefore, data must be protected through robust cryptographic techniques, directly implemented by the data provider. In this way, privacy can be guaranteed independently from the part of the network where such data will be stored (e.g., remote cloud or network edge) [14]. Secondly, the flexibility in the access control represents a critical point in data sharing because IIoT systems are no longer limited to one-to-one authorization [58]. To reduce the danger of unauthorized actions, flexible access policies are needed to regulate the accessibility and usability of services. Thirdly, differently from conventional cloud-based storage systems, upcoming IIoT deployment should deeply leverage the potentials of edge computing and the possibility to store data at the network edge (i.e., very close to the data consumers), for providing customized complex services to actuators, robots, mobile agents, controlled devices, and human workers [93].

Very promising and data-centric solutions include ABE schemes and SE algorithms [44]. ABE is essential in access control because it protects data from unauthorized users [45]. SE technology is a cryptographic function able to encrypt data in a searchable manner: it allows retrieving the specific encrypted data by searching related keywords, while ensuring confidentiality [23]. Recently scientific literature presents several cryptographic schemes, where the above-mentioned techniques are combined in order to guarantee privacy-preserving solutions in file storage servers. Most of the solutions, including the ones in the IoT field proposed in [94] and [95], introduce ABSE schemes in cloud environments where IoT devices upload encrypted documents to cloud servers and authorized users can retrieve and read them by submitting a query to the cloud, which in turn performs the search algorithm to find the required document.

Anyway, from the study of the state of the art (see Section 3.2 for more details) it emerges that available solutions generally focus the attention on single cryptographic operations and propose cloud-based approaches (sometimes supported by a lightweight scheme [58] [59] or exploiting edge/fog nodes implementing part of security tasks [46] and [47]). Nevertheless,

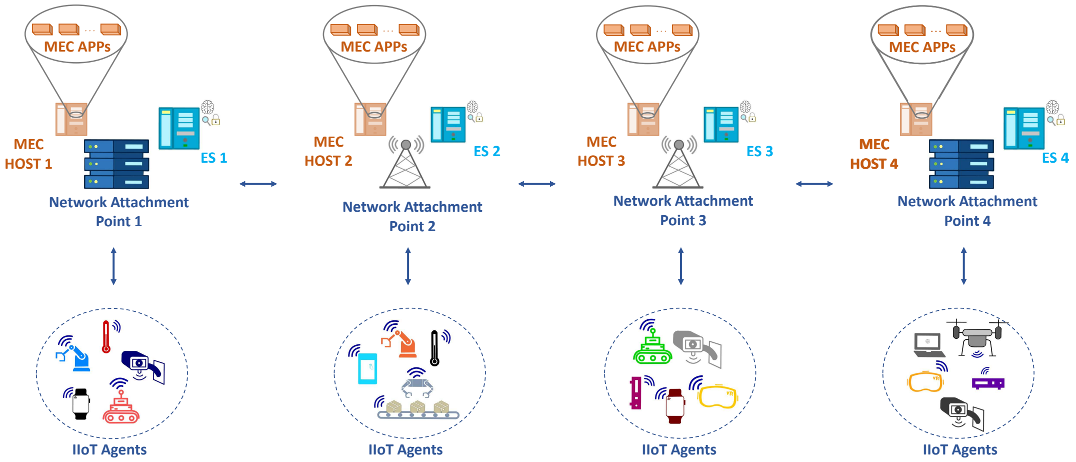


FIGURE 3.1: The reference distributed network architecture.

no works investigate the adoption of these techniques in scenarios with multiple data producers and end-users. Also, to the best of the authors' knowledge, the chance of sharing and disseminating data through the IIoT network in a distributed, effective, and privacy-oriented way by exploiting SE solutions represents an uncovered research goal.

To bridge this gap, this work envisages a novel methodology offering an efficient, scalable, and privacy-preserving data distribution at the network edge, by applying SE.

The reference architecture embraces heterogeneous data producers attached to a distributed network infrastructure through Network Attachment Point, MEC servers hosting applications, and Edge Servers. More specifically, MEC applications express the interest to receive specific data by sending Trapdoors to Edge Servers, data producers protect their contents through ABE and send them to Edge Servers, which implement SE to disseminate received contents only to MEC nodes hosting the applications that generated valid Trapdoors. The resulting scheme is privacy-preserving because Edge Servers are not endowed with cryptographic material. Moreover, it registers lower dissemination delays with respect to cloud-based solutions.

3.2.1 The Conceived Data Dissemination Scheme

Fig. 3.1 depicts the distributed network architecture considered in this work. Here, heterogeneous Network Attachment Points offer wireless or wired connectivity to groups of IIoT agents (i.e., sensors, wearables, fixed robotic arms, mobile robots, drones, and industrial processes). MEC hosts and Edge Servers are deployed at the edge of the network. According to ETSI-MEC specifications [50], each MEC host may integrate multiple MEC applications that, by exploiting powerful tools and computational resources, are able to process (e.g, data mining and fusion) heterogeneous data generated by IIoT agents, as well as to provide advanced and specialized services close to the end-users. On the other hand, Edge Servers are in charge of processing the received traffic flow while routing and forwarding data at the network edge. Without loss of generality, it is assumed that beyond each Network Attachment Point are available one Edge Server (ES) and one MEC host (handling many MEC applications). As anticipated in the introduction, data dissemination is autonomously handled by the ES, via ABSE.

Design Principles

The conceived approach leverages the following design principles.

First of all, security is enforced by an Authority that is a fully trusted third party responsible for the system setup. Specifically, it deals with system security initialization parameters, key material generation, attribute management, and policy enforcement.

MEC applications, which are in possession of a precise set of attributes (generated and released by the aforementioned trusted Authority), request data identified with a set of *keywords*. For example, a monitoring application can be interested in knowing the variables measured by all the available sensors, an AR/VR application is interested in retrieving all the data associated with a given industrial process, an indoor navigation process needs to know the location of robots and packages, and so on. According to the ABSE scheme (whose technical details are presented in the next sub-section), each request is encoded via *search Trapdoors*, based on the selected keywords and attributes. Each MEC host collects the Trapdoors generated by its MEC applications and shares them with all the available Edge Servers. Since Trapdoors hide the search keywords and attributes through cryptographic operations, Edge Servers can not retrieve any information about application interests and related access capabilities (i.e., privacy-oriented approach).

IIoT agents generate data (e.g., multimedia contents, time-series values, and so on) and outsource them to the closest ES. In other words, they represent the data producers. To this end, they select the specific keywords associated with the generated data, encrypt both keywords and data through ABE, and deliver the overall output to the closest ES.

Each ES handles a *Trapdoor table*, which jointly stores application requests and references the MEC host. Note that the Trapdoor table is a completely new entity envisaged in this contribution, and properly used to distribute data directly at the network edge, in an effective, distributed, and privacy-oriented way. Edge Servers have a twofold contribution: i) running the *search algorithm* over encrypted data, and ii) *disseminating data* towards specific MEC hosts. Therefore, when new data is received, ES scrolls the Trapdoor table in order to find the Trapdoors that match the keywords and the policies defined in the encrypted data. The search procedure returns the list of MEC hosts that previously sent valid MEC applications trapdoors that match both keywords and policies of data producers. Also in this case, it is worth mentioning that the search procedure does not provide any meaningful information on the search content to Edge Servers. Accordingly, the resulting approach ensures a privacy-preserving behavior: elements at the network edge receive and distribute data without revealing the related contents, since SE is used.

Technical details about the data dissemination workflow

This subsection formalizes both search and data dissemination processes, by providing technical details about security operations to be implemented. It is very important to remark that this contribution does not propose a novel ABSE algorithm, but it aims at integrating one of the techniques already provided in the current scientific literature for supporting fast and privacy-oriented data dissemination at the network edge. As a consequence, any ABSE mechanism can be integrated within the overall data dissemination workflow discussed herein. However, without loss of generality, the ABSE algorithm presented in [94] is taken as a reference example since it has been found to be less computationally expensive than others, and the overall complexity remains constant even as the number of users' attributes increases.

The overall data dissemination workflow is divided into five distinct phases, illustrated in Fig. 3.2 and detailed below.

Phase 1: system setup.

The selected ABSE scheme considers two groups of order p, $\mathbb{G}$ and $\mathbb{G}_T$, and a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. At first, the trusted Authority randomly selects $\alpha, \gamma \in \mathbb{Z}_p$ and $g, h_1, h_2 \in \mathbb{G}$, and considers three hash functions $H_1, H_2, H_3 : \{0,1\} \rightarrow \{0,1\}^{log_p}$. Then, it generates the master secret key, which is M_k, and the public parameters, which are P_b, as in

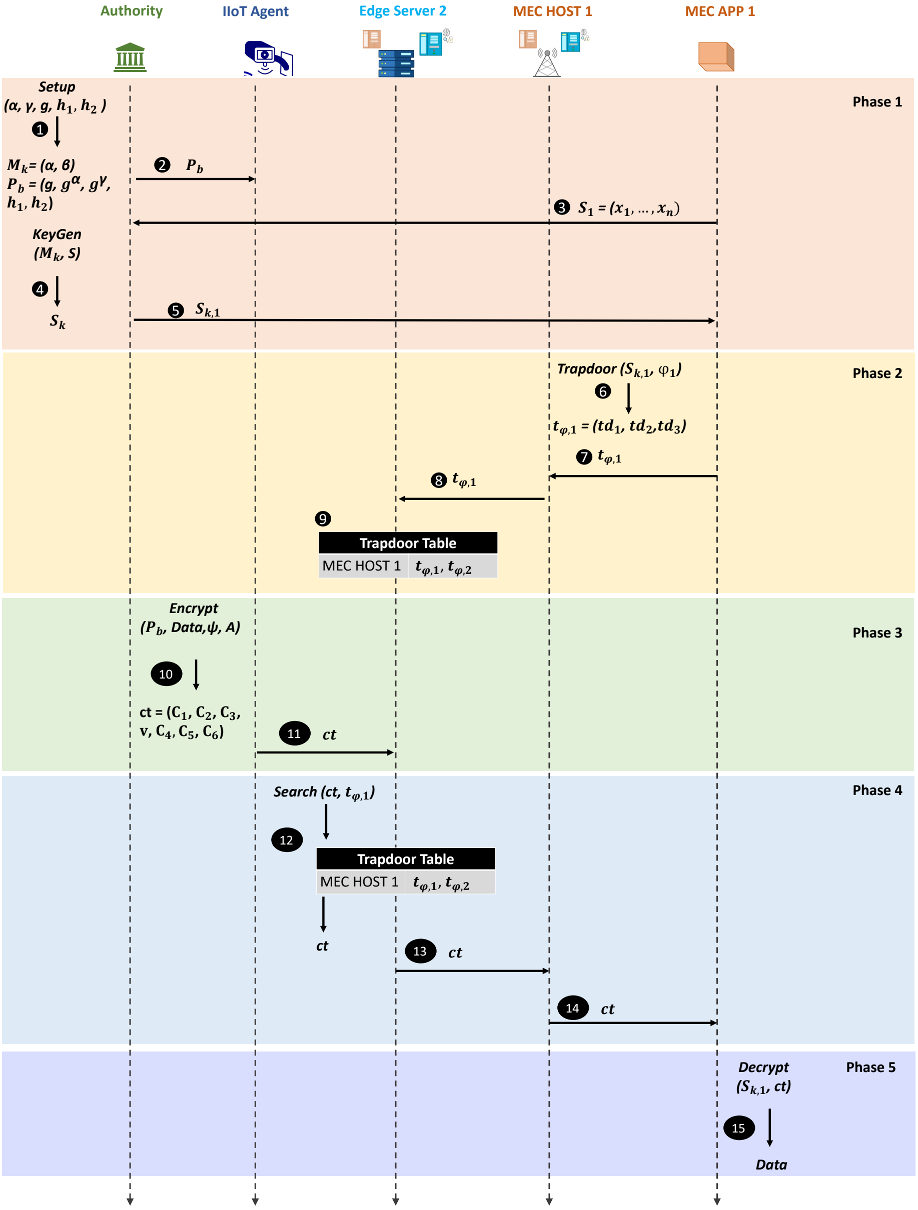

Authority
IIoT Agent
Edge Server 2
MEC HOST 1
MEC APP 1
Phase 1
Setup
(α, γ, g, h_1, h_2)
❶
M_k= (α, β)
P_b = (g, g^α, g^γ, h_1, h_2)
❷ P_b
❸ S_1 = (x_1, ..., x_n)
KeyGen
(M_k, S)
❹
S_k
❺ $S_{k,1}$
Phase 2
Trapdoor ($S_{k,1}$, φ_1)
❻
$t_{\varphi,1}$ = (td_1, td_2,td_3)
❼ $t_{\varphi,1}$
❽ $t_{\varphi,1}$
❾
Trapdoor Table
MEC HOST 1 $t_{\varphi,1}$, $t_{\varphi,2}$
Phase 3
Encrypt
(P_b, Data,ψ, A)
10
ct = (C_1, C_2, C_3, v, C_4, C_5, C_6)
11 ct
Phase 4
Search (ct, $t_{\varphi,1}$)
12
Trapdoor Table
MEC HOST 1 $t_{\varphi,1}$, $t_{\varphi,2}$
ct
13 ct
14 ct
Phase 5
Decrypt
($S_{k,1}$, ct)
15
Data

FIGURE 3.2: Data dissemination workflow.

what follows:

$$\begin{cases} M_k = (\alpha, \gamma) \\ P_b = (g, g^\alpha, g^\gamma, h_1, h_2). \end{cases} \tag{3.1}$$

The master secret key, which is used to create users' secret keys, is kept private. The public parameters, instead, are published by the Authority.

Moreover, by exploiting an AND-gate access structure based on n attributes and assuming that each attribute can assume different values, the Authority generates MEC applications attributes set and data producers policies respectively denoted by: $\mathrm{X} = (x_1, x_2, ..., x_n)$ *and* $A = (a_1, a_2, ..., a_l)$.

After receiving a set of attributes from the MEC application, the Authority produces the secret key for that application. Basically, a MEC application that joins the industrial network sends its set of attributes $\mathrm{X} = (x_1, x_2, ..., x_n)$ to the Authority. Then, the Authority chooses a random $r \in \mathbb{Z}_p$ and implements the key generation algorithm:

$$\begin{cases} \rho_1 = (h_1 g^{-r})^{\frac{1}{\alpha - \sum_{i=1}^{n} H_1(x_i)}} \\ \rho_2 = (h_2 g^{-r})^{\frac{1}{\gamma - \sum_{i=1}^{n} H_1(x_i)}}. \end{cases}$$

Accordingly, the secret key of the MEC application, S_k, is computed as:

$$S_k = (r, \rho_1, \rho_2),$$

and shared with the reference application.

Phase 2: trapdoor generation and forwarding. During this phase, the MEC application generates the search Trapdoor, that is t_Φ. Specifically, starting from its secret key S_k, the set of k keywords $\Phi = (\phi_1, \phi_2, ..., \phi_k)$ of its interest, and a random number $z_p \in \mathbb{Z}_p^*$, the Trapdoor is calculated as:

$$t_\Phi = (td_1, td_2, td_3), \tag{3.2}$$

where $td_1 = \rho_2^{z_p \cdot \sum_{i=1}^{k} H_2(\phi_i)}$, $td_2 = r \cdot z_p \cdot \sum_{i=1}^{k} H_2(\phi_i)$, and $td_3 = h_2^{z_p}$.

As anticipated in the previous sub-section, the Trapdoor is shared with all the Edge Servers in the system.

Phase 3: encryption and outsourcing. Let M be the data to encrypt and outsource to the ES. $\Psi = (\psi_1, \psi_2, ..., \psi_z)$ denotes the list of z keywords associated with that data. Moreover, $A = (a_1, a_2, ..., a_l)$ represents the list of attributes forming the access policy used to protect the data against unauthorized users. The encryption algorithms consider in input the public parameters P_b, the data M, the set of keywords Ψ, and the access policy A. Indeed, by extracting a random $s \in \mathbb{Z}_p^*$, the ciphertext is obtained as:

$$ct = (C_1, C_2, C_3, v, C_4, C_5, C_6)\ , \tag{3.3}$$

where:

$$\begin{cases} C_1 = g^{\alpha s} \cdot g^{-s \cdot \sum_{i=1}^{l} H_1(a_i)} \\ C_2 = e(g,g)^s \\ C_3 = M \cdot e(g, h_1)^{-s} \\ v = H_3(C_1, C_2, C_3) \\ C_4 = g^{\gamma v} \cdot g^{-v \cdot \sum_{i=1}^{l} H_1(a_i)} \\ C_5 = e(g,g)^v \\ C_6 = g^{v \cdot \sum_{i=1}^{z} H_2(\psi_i)} \end{cases}$$

Finally, the data producer sends the ciphertext *ct* to the reference ES.

Phase 4: search and data forwarding. This phase involves the ES, which performs the search algorithm to determine whether the received encrypted data matches one or more queries stored into the Trapdoor table. *Differently from the current scientific literature*, the procedure proposed herein operates in a scenario with multiple IIoT agents and multiple MEC applications.

In details, for each received data *ct* and for each stored Trapdoor t_Φ, the ES verifies that the following equation holds:

$$e(C_4, td_1) \cdot C_5^{td_2} = e(C_6, td_3). \tag{3.4}$$

The validity of the equation proves that i) the set of keywords Ψ in *ct* contains the keywords Φ retrieved from t_Φ and ii) the set of attributes S belonging to the MEC application matches the access policy A used to protect the considered data. In case of matching, the search algorithm produces in output 0, otherwise it returns 1.

During the search algorithm, all the Trapdoors are processed. However, if multiple Trapdoors received from the same MEC host produce a match, the ES delivers the encrypted data to that MEC host only once, denoting the list of interested MEC applications. In this way, the proposed approach also ensures a reduction in bandwidth consumption.

For the sake of clarity, search and data forward operations are defined in the Algorithm 1.

Algorithm 1 The proposed search and data forwarding phase

```
Each MEC application sends $t_{\phi,i} = (td_1, td_2, td_3)$ to the MEC host
The MEC host forwards $t_{\phi,i}$ to ESs
Each ES stores $t_{\phi,i}$ in its Trapdoor table
The ES receives $ct = (C_1, C_2, C_3, v, C_4, C_5, C_6)$
from a data producer
For each MEC host registered into the ES Trapdoor table
while Search(ct, $t_{\phi,i}$) = 0 do
    if $e(C_4, td_1) \cdot C_5^{td_2} = e(C_6, td_3)$ then
        Search($ct$,$t_{\phi,i}$) = 1
        if $ct$ has not been sent to the MEC host then
            the ES forwards $ct$ to the MEC host
            The ES records that $ct$ is sent to the MEC host
        end if
    else
        Search($ct$,$t_{\phi,i}$) = 0
    end if
end while
```

Phase 5: decryption. This phase allows the MEC application to decrypt the received cyphertext ct, by using its $sk = (r, \rho_1, \rho_2)$:

$$M = C_3 \cdot e(C_1, \rho_1) \cdot C_2^r. \tag{3.5}$$

Running Example

This section presents a running example willing to better explain operations and the interactions to be performed. The use case scenario considers 3 Network Attachment Points equipped with a MEC host and an ES. Moreover, as illustrated in Fig. 3.3, each ES has a Trapdoor table where all receiving Trapdoors are stored and listed with respect to the referred MEC host. The example use case considers an AR/VR application and a sensing control as MEC applications and a mobile robot with an integrated camera as a data producer. Specifically, the MEC host 1, holds two MEC applications: an AR/VR application and a sensing control one. These two generate query Trapdoors and the MEC host 1 forwards them to all the Edge Servers. Let's assume that both Trapdoors contain "AR", "video", and "robot" as query keywords and that they are asking for a video stream flow outsourced from a mobile robot with an integrated camera, which is attached at the ES 2. When the mobile robot outsources the data to the ES 2, this one checks within its Trapdoor table by comparing the encrypted data with the encrypted query keywords. As soon as it finds the matching Trapdoor (i.e., the one of the AR/VR application) it sends the data to the MEC host 1 and records that the data has been sent to that MEC host. In this way, when the Trapdoor referred to the sensing control application matches the ciphertext, the ES 2 does not re-send the same data to the same MEC host.

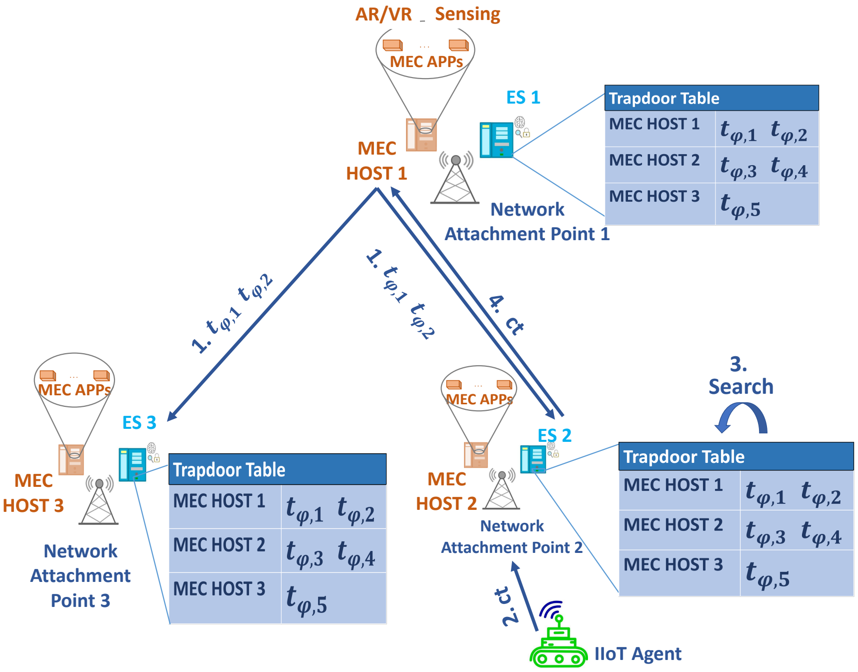

AR/VR
Sensing
MEC APPs
ES 1
MEC HOST 1
Network Attachment Point 1
Trapdoor Table
MEC HOST 1 $t_{\varphi,1}$ $t_{\varphi,2}$
MEC HOST 2 $t_{\varphi,3}$ $t_{\varphi,4}$
MEC HOST 3 $t_{\varphi,5}$
1. $t_{\varphi,1}$ $t_{\varphi,2}$
1. $t_{\varphi,1}$ $t_{\varphi,2}$
4. ct
3. Search
MEC APPs
ES 3
MEC HOST 3
Network Attachment Point 3
Trapdoor Table
MEC HOST 1 $t_{\varphi,1}$ $t_{\varphi,2}$
MEC HOST 2 $t_{\varphi,3}$ $t_{\varphi,4}$
MEC HOST 3 $t_{\varphi,5}$
MEC APPs
ES 2
MEC HOST 2
Network Attachment Point 2
Trapdoor Table
MEC HOST 1 $t_{\varphi,1}$ $t_{\varphi,2}$
MEC HOST 2 $t_{\varphi,3}$ $t_{\varphi,4}$
MEC HOST 3 $t_{\varphi,5}$
2. ct
IIoT Agent

FIGURE 3.3: Running example.

TABLE 3.2: Computational cost of cryptographic operations.

Cryptographic operation	**Execution time [ms]**
Pairing in $\mathbb{G}$ (P)	27.98
Exponentiation in $\mathbb{G}$ (E)	18.62
Exponentiation in $\mathbb{Z}_p$ (E_z)	0.759
Multiplicative in $\mathbb{Z}_p$ (M_z)	0.0058385
Search Time	$T_{SE} = 1 * E + 2 * P =$ $= 74.58$
Encryption	$T_{enc} = 3 * P + 8 * E + 2nM_z =$ $= 232.9 + 2 * n * 0.0058385$
Decryption	$T_{dec} = 1 * P + 1 * E + 2 * M_z =$ $= 46.61$

3.2.2 Analytical evaluation

To demonstrate the great potential of the conceived privacy-preserving data dissemination scheme, this section presents a numerical investigation conducted in different scenarios. To this end, a MATLAB script has been developed to model a distributed IIoT environment. The investigated KPIs include: i) the *average search time*, defined as the amount of time the node implementing the Searchable Encryption algorithm takes to check the received data with all the available Trapdoors, and ii) the *average delivery delay*, expressed as the average amount of time needed to deliver the generated data to the MEC applications that issued the right Trapdoors. Specifically, the *average search time* is evaluated as the time taken to execute the number of cryptographic operations needed. Results are compared against those registered by a baseline approach, where data and Trapdoors are managed by a remote cloud (which performs, in a centralized way, searching and delivery tasks).

The study considers a network with a variable number of Network Attachment Points, ranging from 2 to 10. Indeed, the number of MEC hosts and Edge Servers available in the considered network infrastructure range from 2 to 10, as well. Let N_{ES} be the number of available ES.

A variable number of data producers, N_{DP}, are randomly and uniformly distributed among network cells served by the aforementioned Network Attachment Points. Specifically, N_{DP} is set to 10, 50, or 100. Without loss of generality, it is assumed that these devices generate data for S different services. Therefore, these data are protected according to the access policies configured for the type of service they belong to. Each test randomly maps a data producer to one of the available service types. Moreover, the access policy is defined through a combination of n attributes.

On the other hand, a total number of MEC applications, N_{app}, are randomly and uniformly distributed among the available MEC hosts. In line with the previous assumptions, each test randomly maps an MEC application to one of the available service types. Thus, each MEC application is configured to request (via Trapdoors and according to the protocol discussed in the previous Section) all the data belonging to a given service type. The number of MEC applications is chosen in the range from 20 to 100.

Analysis of the average search time

The study was conducted by using a Windows system with an Intel Core i7 CPU at 2.60GHz. According to [94], the computational cost associated with a single search operation in the ABSE scheme considers one pairing in $\mathbb{G}$ (P), one Exponentiation in $\mathbb{G}$ (E), and one Multiplicative in $\mathbb{Z}_p$ (M_z). The resulting search time, namely T_{SE} is reported in table 3.2.

The scientific literature also demonstrated that the amount of time to perform multiple search operations linearly increases with the number of generated data or Trapdoors to be checked (see [96] for example). Indeed, given the number of MEC applications N_{app}, the number of data producers managed by the $i-th$ ES N^i_{DP}, and by assuming that all these data producers generate data within the same observation time interval (worst case), the resulting search time is equal to:

$$\hat{T_{SE}}\Bigg|_{proposal} = T_{SE} + \beta(N^i_{DP}N_{app} - 1). \tag{3.6}$$

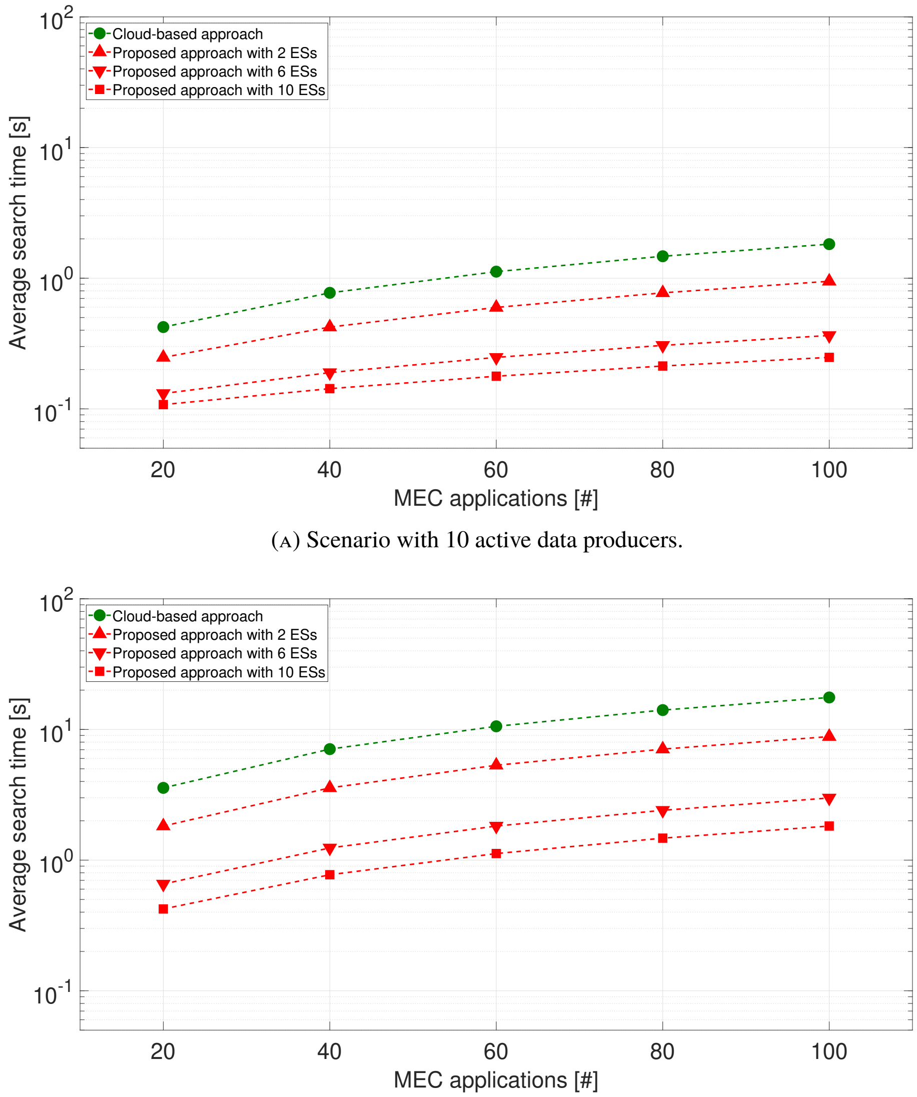
Indeed, when a single data producer (e.g., $N^i_{DP} = 1$) and one MEC application (e.g., $N_{app} = 1$) are considered, the search time results in:

$$\hat{T_{SE}}\Bigg|_{proposal} = T_{SE}.$$

Now, considering that the average number of data producers managed by the $i - th$ ES is equal to $\bar{N}_{DP} = E[N^i_{DP}] = N_{DP}/N_{ES}$, the average search time achievable by each single ES is equal to:

$$\bar{T}_{SE}\Bigg|_{proposal} = E[\hat{T_{SE}}] = T_{SE} + \beta\left(\frac{N_{DP}N_{app}}{N_{SE}} - 1\right). \tag{3.7}$$

A different story is experienced for the cloud-based approach. In this case, the search operation is performed only in a single node of the network. Therefore, given the number of MEC applications N_{app}, the total number of data producers N_{DP}, and by assuming that all these data producers generate data within the same observation time interval (worst case), the



(A) Scenario with 10 active data producers.

(B) Scenario with 100 active data producers.

FIGURE 3.4: Average search time vs number of MEC applications.

resulting average search time is equal to:

$$\bar{T}_{SE}\Big|_{cloud} = T_{SE} + \beta(N_{DP}N_{app} - 1). \tag{3.8}$$

The analysis of the state of the art suggests setting $\beta = 1.75$ [96]. Based on these premises, the average search time achievable in different scenarios is reported in Fig. 3.4. Reported results demonstrate that the average search time of both approaches increases with the number of data producers and MEC applications. The proposed approach, by distributing the search procedure on different Edge Servers on the edge of the network, permits obtaining a shorter search process time.

Indeed, by considering the scenario with 100 data producers and 10 MEC applications, the average search time is reduced by about 81.6% with 6 Edge Servers and about 88.2% with 10 Edge Servers. In the same way, by introducing 100 data producers and 100 MEC applications, the search time decreases by about 83% with 6 Edge Servers and about 89.6% with 10 Edge Servers. Finally, focusing the attention on 100 data producers, it could be noticed how passing

TABLE 3.3: Average Communication delays

Communication type	Average RTT [ms]	Delay [ms]	Number of hops [#]
$\bar{T}_{radio}$	1.309	0.6545	1
$\bar{T}_{edge}$	16.594	8.297	1
$\bar{T}_{cloud}$	42.08	21.04	22

from 10 to 100 MEC applications the average search time increases by 14 seconds for the cloud and 1.4 seconds for the proposed approach with 10 Edge Servers.

Analysis of the average delivery delay

In order to estimate the average delivery delay, it is necessary to introduce the following variables: i) $\bar{T}_{radio}$, that represents the delivery delay in the radio interface, ii) $\bar{T}_{edge}$, that represents the delivery delay at the network edge, and iii) $\bar{T}_{cloud}$, that represents the delivery delay experienced when a remote cloud is contacted. These variables have been evaluated through Ping and Trace Route tests. In particular, by using a computer connected to the network of Politecnico di Bari, for each test the average RTT on 10^3 consecutive pings has been considered. First, a test on a remote Amazon server has been made to estimate the communication delay between a data producer and the remote cloud server (i.e., $\bar{T}_{cloud}$). The second test has been done to evaluate the average communication delay experienced for contacting the closest Network Attachment Point (i.e., $\bar{T}_{radio}$). Finally, a test on another device connected to the same network at the Politecnico di Bari has been done to estimate the communication time at the edge (i.e., $\bar{T}_{edge}$). Results, used to evaluate the average delivery delay, are reported in Table 3.3.

The analysis of the average delivery delay should also consider the amount of time required to encrypt and decrypt data, denoted with T_{enc} and T_{dec}, respectively. Their values are reported in Table 3.2.

Regarding the methodology proposed in this work, the average delivery delay can be evaluated as:

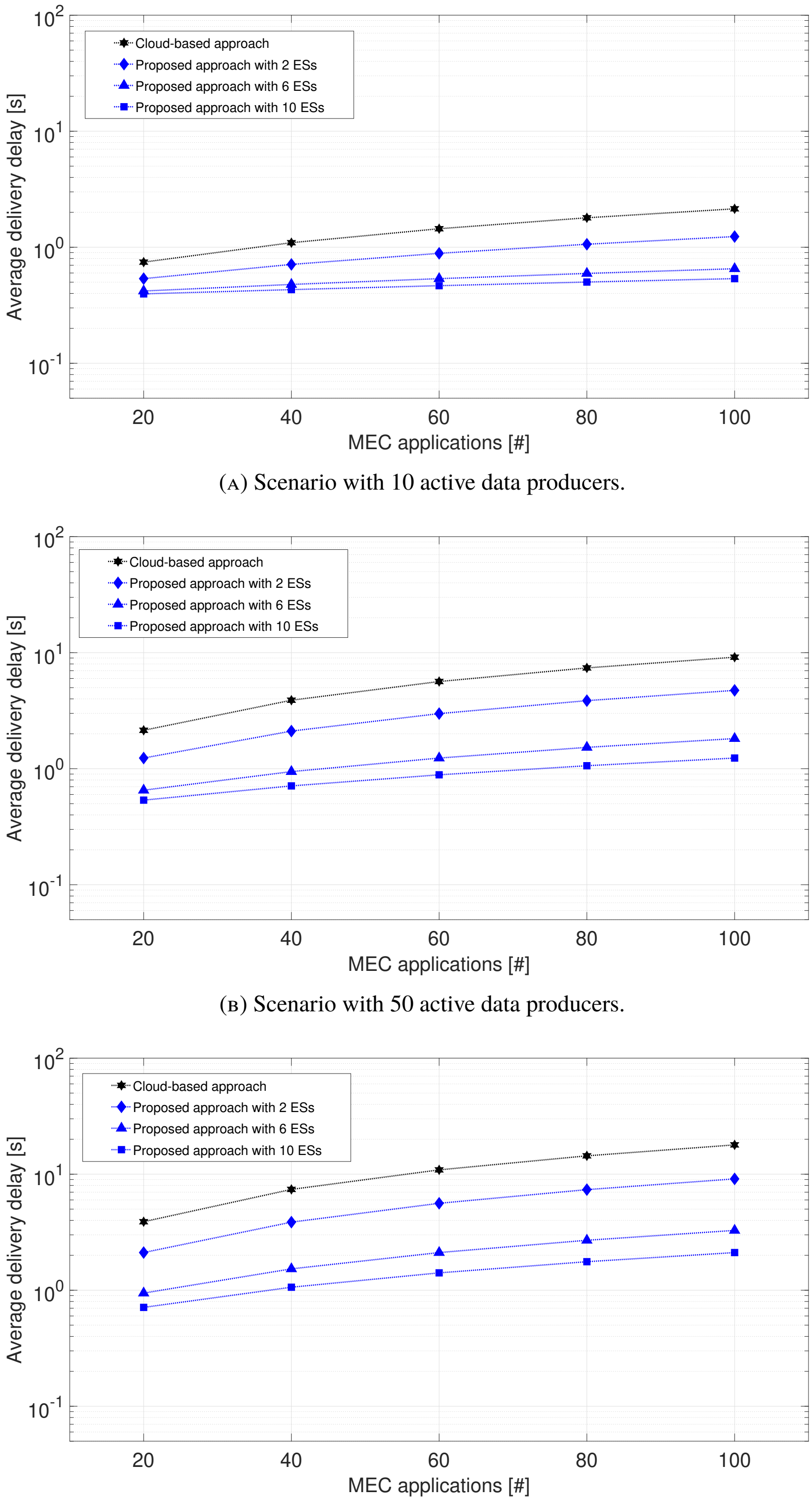
$$\begin{aligned}\bar{T}_{del}\Big|_{proposal} &= T_{enc} + \bar{T}_{radio} + \\ &+ \left[T_{SE} + \beta\left(\frac{N_{DP}N_{app}}{N_{SE}} - 1\right)\right] + \\ &+ \bar{T}_{edge} + T_{dec}.\end{aligned} \tag{3.9}$$

Differently, in the cloud-based solution, the average delivery delay can be evaluated as:

$$\begin{aligned}\bar{T}_{del}\Big|_{cloud} &= T_{enc} + 2\bar{T}_{cloud} + \\ &+ \left[T_{SE} + \beta\Big(N_{DP}N_{app} - 1\Big)\right] + \\ &+ T_{dec},\end{aligned} \tag{3.10}$$

where, $2\bar{T}_{cloud}$ is the sum of the time needed to send the data to the cloud server and the time required to deliver the searched ciphertext back to the MEC host.

Fig. 3.5 shows the average delivery delay as a function of the number of MEC applications and Edge Servers. The three sub-figures refer to scenarios with different numbers of data



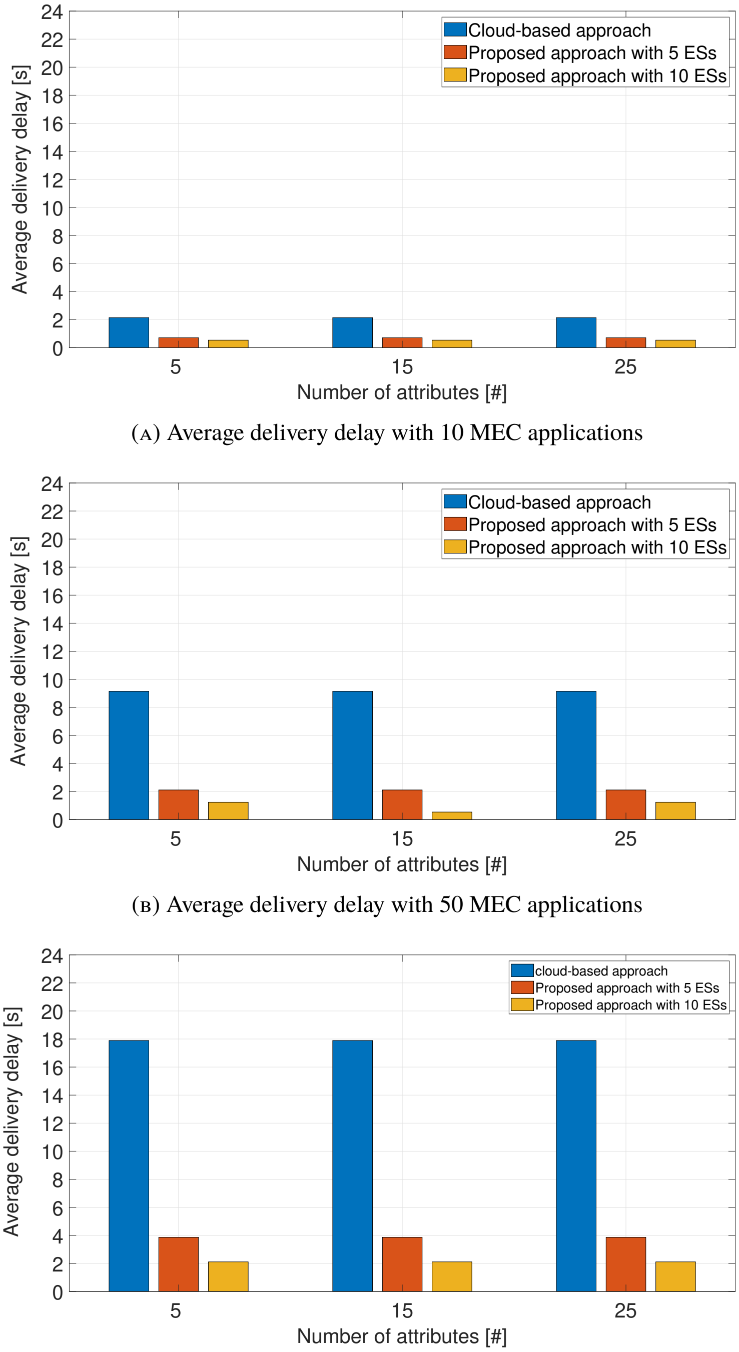
(A) Scenario with 10 active data producers.

(B) Scenario with 50 active data producers.

(C) Scenario with 100 active data producers.

FIGURE 3.5: Average delivery delay vs number of MEC applications.

producers. Results highlight how distributing search operations on the edge of the network allows for decreasing the amount of time needed for retrieving the query data flow. Indeed, as the number of data producers increases, the distance between the average delivery delay of the cloud-based approach and the proposed one increases, passing from a difference of a few seconds with 10 data producers to a difference of about 10 seconds with 100 data producers. Thus, the deployment of search operation directly on the edge of the network allows for reducing the average delivery delay up to 45% with respect to the baseline approach.



(A) Average delivery delay with 10 MEC applications

(B) Average delivery delay with 50 MEC applications

(C) Average delivery delay with 100 MEC applications

FIGURE 3.6: Average delivery delay vs numbers of attributes.

Finally, since table 3.2 shows that the number of attributes affects the encryption time, Fig. 3.6 reports the evaluation of the average delivery delay with different attributes. In line with [94], it can be noticed that, by fixing the number of data producers to 100, the variation of the number of attributes causes a marginal change in the average delivery delay, while increasing the number of MEC applications.

3.3 Privacy-preserving data dissemination scheme based on Searchable Encryption, publish-subscribe model, and edge computing

Today, security and privacy are considered fundamental requirements for the communication infrastructures used in any application domain [22] [97]. For this reason, many national and international regulations (like the General Data Protection Regulation in Europe [98]) impose the adoption of sophisticated mechanisms and tools ensuring confidentiality, data protection, access control, and other privacy-oriented security services. Moreover, to efficiently achieve this goal, some cryptographic algorithms, such as Attribute-based Encryption (ABE), have been conceived to natively enforce security directly on the data and prevent unauthorized entities (including *honest but curious clouds* [14]) accessing confidential information.

In this context, Searchable Encryption (SE) is emerging as a highly promising technique supporting data protection and keyword search over encrypted data [44]. Typically integrated into cloud-based applications, it assumes that [23]: i) the data owner (also referred to as data producer) encrypts and uploads the data in the cloud, ii) the authorized data users (also referred to as data consumer) issues a cryptographic keyword query to the cloud, and iii) the cloud delivers specific encrypted data to the requester only in case of search matching.

Recently, the scientific literature investigated the usage of SE in Internet of Things (IoT) scenarios [94] [77] [78]. Moreover, given the high computational complexity of SE cryptographic operations, some contributions formulated lightweight techniques [58] [59] or proposed to offload some security tasks to edge/fog nodes [46] [47] [77].

Nevertheless, despite these very valuable studies, available approaches still consider the remote cloud the only entity able to store data, manage keyword search over encrypted data, and deliver them to requesting users. Accordingly, the heavy computational load generated in scenarios with multiple data producers and data consumers (never studied yet) and larger end-to-end communication latencies inevitably compromise system performance.

Based on these premises, to achieve an important step forward in this direction, this work proposes a novel methodology offering a privacy-oriented data dissemination at the network edge. It is important to note that this contribution does not propose a new SE algorithm. Instead, it investigates the adoption of any SE algorithms within a novel service architecture handling the data dissemination process in a distributed manner and without the help of remote clouds. Differently from the current state of the art, this work provides the following main scientific contributions:

- First, it designs a novel service architecture supporting a scalable, efficient, and privacy-oriented data dissemination by combining attribute-based SE approaches, a publish-subscribe communication model, and edge computing capabilities. The reference architecture includes heterogeneous data producers and data consumers served by B5G base stations and some Edge Servers deployed at the network edge. The former group of users generates and encrypts data via attribute-based SE and publishes them to the closest Edge Server. While the second group issues encrypted subscription requests (that are encrypted queries, namely *Trapdoors*) and share them with all the available Edge Servers. The keyword search over encrypted data is implemented at the network edge, e.g., by the Edge Servers, in a distributed manner. Indeed, once new data is published, the reference Edge Server delivers the encrypted data to the consumers who issued a valid Trapdoor.

- Second, it presents an implementation of the aforementioned service architecture modeling the most significant functions of an Edge Server and its interactions with data producers and other remote Edge Servers (including the receiving of data published by

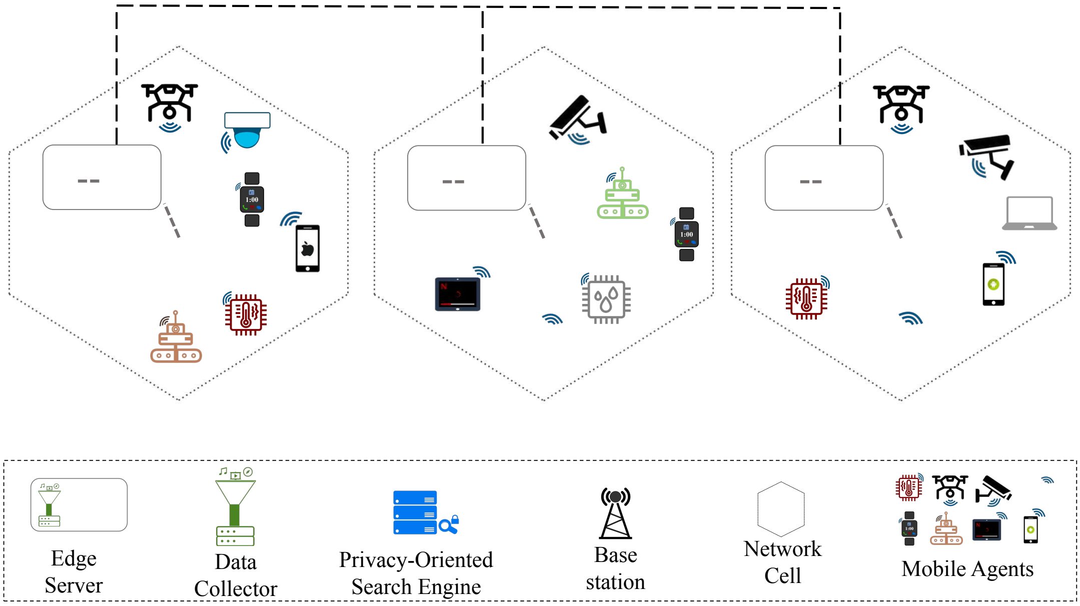


FIGURE 3.7: The reference distributed service architecture.

data producers on the Edge Server via a memory-less Poisson process, the execution on the Edge Server of the cryptographic operations, and the resulting data dissemination tasks). Here, communication latencies between logical nodes are enforced according to the values proposed in the literature.

- Third, it experimentally evaluates the performance of the proposed approach in realistic scenarios where coexist heterogeneous data producers and end-users. Note that the conducted study does not only investigate the computational complexity of SE operations as a function of security parameters (which represents the main objective of all the contributions available in the current scientific literature). Nevertheless, it explores the impact of network settings (i.e., number of Edge Servers) and loads (i.e., number of data consumers and various number of publications over time, modeled through Poisson-distributed rates) to these three Key Performance Indicators (KPIs): latencies associated with both cryptographic operations and the overall dissemination process, as well as the consumed energy. The obtained results demonstrate the unique ability of the proposed solution to achieve shorter delays and less energy consumption values compared to cloud-based alternatives.

3.3.1 The Conceived Data Dissemination Scheme

Given the problem description presented in Section 3.1.3, this work presents a novel decentralized service architecture that provides privacy-preserving data distribution by combining Attribute-Based Searchable Encryption algorithms, publish-subscribe communication paradigm, and edge computing capabilities.

The reference architecture, shown in Fig. 3.7, is a typical B5G network, where base stations offer wireless connectivity to mobile agents and the edge network hosts computing platforms.

Here, mobile agents are divided into two specific groups: data consumers and data producers, properly distributed among different cells. Their interaction follows the publish-subscribe model.

Data consumers issue their service requests by generating SE *Trapdoors* and share them across all the available servers, placed at the network edge. While, data producers encrypt data (e.g., AR/VR video stream, temperature or humidity data, and healthcare information from wearable sensors) via an Attribute-Based SE algorithm and publish them at the edge of the network.

Customized Edge Servers, deployed at the network edge, interact with each other and with data producers and data consumers to implement the following tasks: i) collect subscription requests into a Trapdoor table, ii) receive encrypted data published by data producers iii) implement keyword search over encrypted data through SE cryptography, and iv) deliver encrypted data only to authorized consumers. To this end, each Edge Server embraces a *Data Collector* and a *Privacy-Oriented Search Engine (POSE)*. These entities can be seen as two separate MEC applications running into the same MEC host, deployed close to the B5G base station, as depicted in Fig. 3.7. For the sake of completeness, it is important to remark that the Data Collector entity is in charge of receiving the encrypted data published by data producers, while the POSE entity runs all the other main functionalities. Moreover, to guarantee a privacy-oriented approach each Edge Server is unable to retrieve any information from the requests since Trapdoors hide required service keywords using cryptographic schemes.

Differently from conventional and cloud-based architectures, the proposed service architecture implements SE operations at the network edge in a distributed way. The term "decentralized" refers to the possibility of distributing search operations at the network edge rather than assigning all to one single remote entity (i.e., in the cloud).

In summary, the following benefits are achieved:

- each Edge Server, and specifically each POSE entity, can implement keyword search operations only on encrypted data generated by the mobile agents served by a specific base station. This ensures a reduction of the overall computational burn with respect to cloud-based approaches and guarantees high levels of scalability in distributed environments;

- data dissemination managed at the network edge reduces communication latencies with respect to cloud-based approaches;

- the POSE entity implements the aforementioned keyword search over encrypted data by considering the list of subscriptions stored within the Trapdoor table. By storing in the Trapdoor table the details about the location of data consumers, it can be possible to deliver only once the encrypted data towards the base station serving consumers interested in the same data. This aspect ensures bandwidth and energy reduction.

Data dissemination workflow

This section describes both the search and data dissemination procedures by giving technical specifics concerning general security measures to be conducted.

It is noteworthy to mention that this contribution does not present a new SE method, but it rather tries to integrate one of the strategies currently available in the scientific literature to enable faster and privacy-oriented data dissemination at the network edge. As a result, any SE technique may be included into the entire data distribution procedure detailed here. Furthermore, to provide concrete examples, the technical details related to the SE algorithms presented in [94] and [88] and their integration in the proposed data dissemination scheme are reported in Appendix A and Appendix B, respectively.

The conceived data transmission procedure is organized into five different stages, as shown in Fig. 3.8 and explained below.

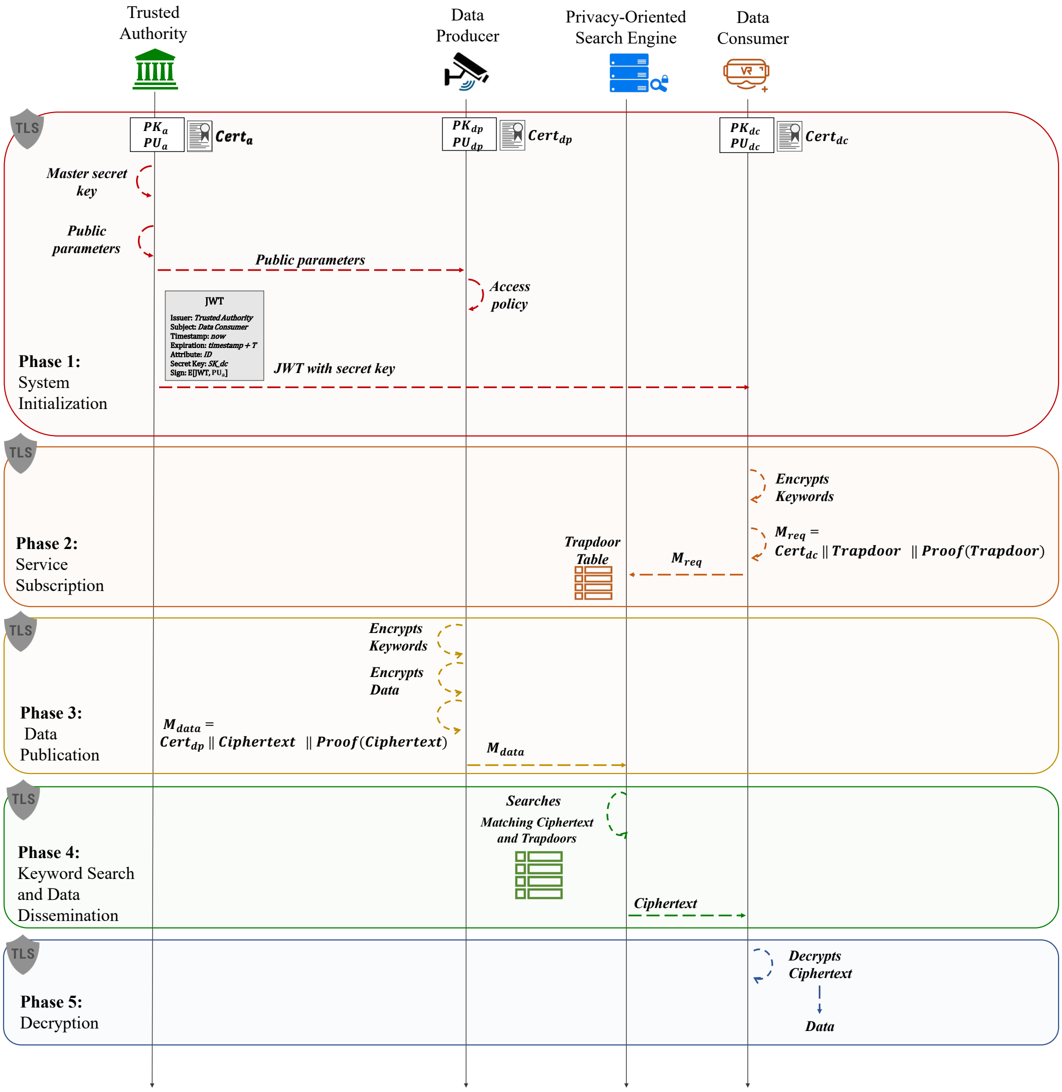

Trusted Authority
Data Producer
Privacy-Oriented Search Engine
Data Consumer
TLS
PK_a PU_a $Cert_a$
PK_{dp} PU_{dp} $Cert_{dp}$
PK_{dc} PU_{dc} $Cert_{dc}$
Master secret key
Public parameters
Public parameters
Access policy
JWT
Issuer: Trusted Authority
Subject: Data Consumer
Timestamp: now
Expiration: timestamp + T
Attribute: ID
Secret Key: SK_dc
Sign: E[JWT, PU_a]
JWT with secret key
Phase 1: System Initialization
Encrypts Keywords
$M_{req} =$ $Cert_{dc} \| Trapdoor \| Proof(Trapdoor)$
Trapdoor Table
M_{req}
Phase 2: Service Subscription
Encrypts Keywords
Encrypts Data
$M_{data} =$ $Cert_{dp} \| Ciphertext \| Proof(Ciphertext)$
M_{data}
Phase 3: Data Publication
Searches
Matching Ciphertext and Trapdoors
Ciphertext
Phase 4: Keyword Search and Data Dissemination
Decrypts Ciphertext
Data
Phase 5: Decryption

FIGURE 3.8: General Data dissemination Scheme.

For the sake of clarity, it is important to remark that the interaction between the trusted authority, data producers and consumers, and the POSE entity is protected via the Transport Layer Security (TLS) protocol.

Phase 1: system initialization.

Phase 1 includes three different steps: initialization of public key cryptography, setup of the access control policies, and attributes processing.

This phase involves a Trusted Authority which is responsible for enforcing system security. It is a completely trusted third party in charge of system configuration by addressing system security setups, key material creation, attribute management, and policy enforcement.

Firstly, during this first phase, a private-public key pair is given to each entity in the network. Public keys are stored in trusted X.509 certificates. This will help achieving peer authentication within the following phases.

Secondly, depending on the encryption algorithm, the Authority generates the master secret key and public parameters. The master secret key is kept secret by the Authority and it is used to create secret keys for data producers and consumers. Instead, the public parameters are exposed by the Authority to all the data producers in the network that use it and their attributes to generate their access control policies.

Thirdly, each data consumer needs to obtain its attributes and its secret key. Token-based standard data structures, are used to send secret keys and associated attributes to data consumers.

A token is commonly used in literature as a container for security-related information, allowing to transfer authentication/authorization information among different communication entities. Basically, the token is a straightforward method that successfully implements the decoupling between authentication and authorization processes [99]. The conceived solution employs the JSON Web Tokens technology (JWT) [100]. In particular, a JWT connects any type of information chosen by the token author (i.e., claims) to the identity of the data consumer for which the token was produced. Moreover, all the information within the token is encrypted [101]. Few standardized claims are already present in a JWT, such as the issuer (i.e., token generator), subject (i.e., token receiver), timestamp, and expiration date [100]. While the secret key and a human-readable string of the attribute set are added. Finally, the sign field is annexed to the end of the container to guarantee the integrity and validity of the token.

Thus, the Authority sends a JWT to each data consumer that will be in possession of a series of attributes and the related cryptographic material.

Phase 2: service subscription. This phase leads data consumers to generate and publish search Trapdoors. Specifically, when a data subscriber needs to retrieve specific data, it chooses a set of keywords, takes its attributes and its secret key, and calculates the Trapdoor by encrypting it with attribute-based encryption.

Then, it asks for its certificate to the Trusted Authority and it generates a trusted and authentic service request message, namely M_{req}, as follows:

$$M_{req} \;=\; Cert_{dc} \,||\, Trapdoor \,||\, Proof(Trapdoor) \;, \tag{3.11}$$

where $Cert_{dc}$ is the data consumer certificate, $Trapdoor$ is the above-created Trapdoor, and $Proof(Trapdoor)$ is the digital signature of the Trapdoor created by using the private key associated with the public key stored in the certificate. Finally, as previously described, the data consumer subscribes the service request message to all the POSE entities in the network.

Phase 3: data publication. Herein, before publishing data on the closest pose entity, a data producer selects specific keywords associated with the data flow and encrypts them with the attribute-based encryption algorithm. It takes as input the data, the keyword, the secret key, and the access policy, while it gives as output the ciphertext.

Additionally, the data producer retrieves its certificate from the Trusted Authority and creates a trusted and authentic message, namely M_{data}, as follows:

$$M_{data} \;=\; Cert_{dp} \,||\, ciphertext \,||\, Proof(ciphertext) \;, \tag{3.12}$$

where $Cert_{dp}$ is the data producer certificate, $ciphertext$ is the published data (protected with the ABE cryptosystem), and $Proof(ciphertext)$ is the digital signature of the ciphertext obtained with the private key linked to the public key stored in the certificate. Subsequently, the message is published on the referenced POSE entity.

Phase 4: keyword search and data dissemination. This phase involves the POSE entity, which runs the search algorithm to verify whether the published encrypted data matches one or more subscriptions stored in the Trapdoor table. Herein, differently from the current scientific literature, the proposed methodology addresses a scenario with multiple data producers and data consumers. In detail, for each received data, the search procedure progressively handles each subscribed Trapdoor. Thus, the POSE entity verifies their equality through specific equations depending on the cryptographic algorithm. The equation's validity demonstrates that i) the set of keywords in the ciphertext contains the keywords of a subscribed request,

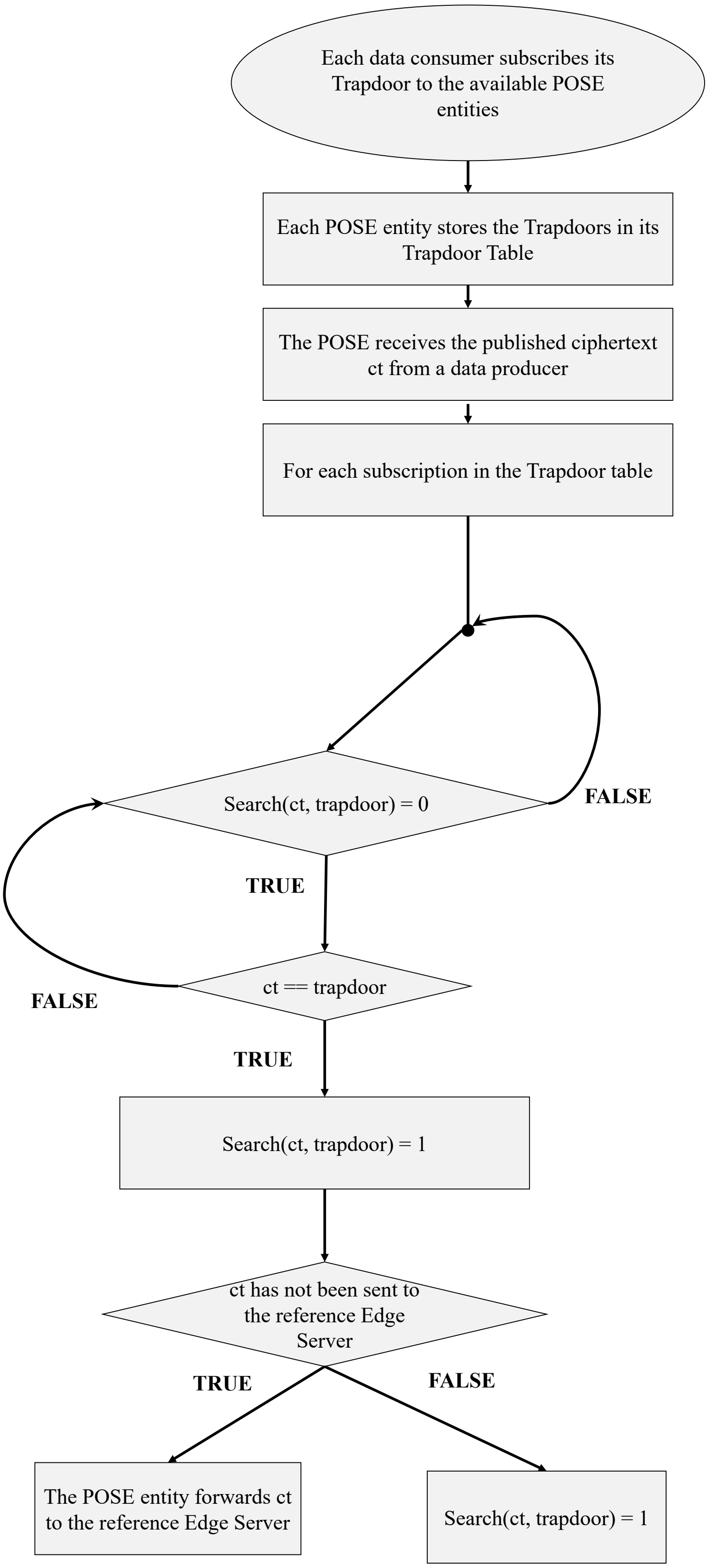

Each data consumer subscribes its Trapdoor to the available POSE entities
Each POSE entity stores the Trapdoors in its Trapdoor Table
The POSE receives the published ciphertext ct from a data producer
For each subscription in the Trapdoor table
Search(ct, trapdoor) = 0
FALSE
TRUE
ct == trapdoor
FALSE
TRUE
Search(ct, trapdoor) = 1
ct has not been sent to the reference Edge Server
TRUE
FALSE
The POSE entity forwards ct to the reference Edge Server
Search(ct, trapdoor) = 1

FIGURE 3.9: Flowchart of the proposed search and data dissemination phase.

and ii) the data consumer attributes match the data producer access policy. The search algorithm returns 0 in case of mismatching, or 1 otherwise.

However, subscriptions of different data consumers connected to the same base station may result in a match. In this context, to guarantee energy efficiency at the network edge, the POSE entity disseminates only once the encrypted data towards the base station by listing all matched data consumers.

For the sake of clarity, search and data dissemination operations are defined in the flow chart in Fig. 3.9.

Phase 5: decryption. Finally, the last phase allows the data consumer to decrypt the received cyphertext with the decryption algorithm taking as input the secret key and retrieving the data.

3.3.2 Security Proof

This section formulates a security proof for the proposed service architecture, by jointly considering cryptographic operations and communication protocols.

Regarding cryptographic operations, it is significant to note that many works in the scientific literature proposing SE algorithms analyze the cryptographic security of their conceived technique. For example, the two algorithms integrated and evaluated in the proposed service architecture, respectively presented in [94] and [88], have been proven to be robust against both Chosen Keyword Attack (CKA) and Chosen Plaintext Attack (CPA). As a result, our envisioned solution is secured by design against the aforementioned cryptographic threats.

Regarding the security analysis of the rest of the service architecture, the designed solution leverages well-known security building blocks (such as TLS, X.509 certificates, and ABE cryptosystems). These remain independently constructed, and their security has been previously established and formally described in the reference contributions listed below. Thus the security analysis of each used functionality is proved as follows:

- **Secure end-to-end communication.** The proposed distributed service architecture, by using TLS (i.e., TLS version 1.3) protocol, guarantees the establishment of a secure end-to-end channel communication between each involved entity of the network (e.g., Trusted Authority, Data Producer, Data consumer, and POSE entity). Specifically, it helps providing data confidentiality and mutual authentication. Moreover, this makes the communication network resilient against Man-In-The-Middle (MITM) attacks. TLS is a well-known and extremely widespread security protocol, thus, its security proof has already been explored in [102], [103], and [104].

- **Peer and data authentication.** Independently from the usage of TLS, the service architecture presented in this work must ensure that submission requests and published data are generated by trusted entities. Therefore, data and peer authentication is achieved by using X.509 certificates and digital signatures. In fact, the digital signature is a crucial aspect of the exchange of messages in the "service subscription" and "data publication" phases, as it serves to verify the authenticity and integrity of the message. This signature is generated using the private key associated with the public key included in the X.509 certificate, and is added to the message along with the mobile agent certificate and the Trapdoor or ciphertext (as shown in Eq. 3.11 and Eq. 3.12). Indeed, the robustness of the cryptographic technique used to generate the digital signature greatly affects the security of X.509 certificates. While classical algorithms such as ECDSA and RSA are commonly used and have been demonstrated to be secure in previous studies, [105] and [106] respectively. Newer quantum-resistant signature schemes such as CRYSTALS-Dilithium [107] and SPHINCS+ [108] may also be used to ensure the security of the digital signature. Additionally, the security of the used standardized data structure determines the security of the cryptographic data associated with each data consumer (i.e., attributes and secret key). The security of the encrypted JWT token depends on the public-key cryptography algorithm used to create the digital sign. RSA and ECDSA techniques can be applied in this situation as well. Their security has been proven in [106] and [105], respectively. Thus, the peer and data authentication procedure helps the conceived methodology being resilient to collusion and replay attacks.

- **Access control.** The "data publication" and "decryption" phases make use of ABE cryptographic techniques (i.e., CP-ABE[53]) for guaranteeing data protection and flexible access control. In this context, the Trapdoor matches guarantee that data consumers asking for a specific resource prove to be in possession of a subset of attributes that satisfies the access control policy uniquely coupled with the resource and chosen by the data producers. This ensures that only authorized mobile agents access to the protected data.

- **Network traffic monitoring.** Intrusion Detection Systems (IDSs), as a unique network security approach, are an important defense solution [109]. Specifically, the IDS collects network traffic, and security logs, and determines whether or not the network has been compromised by examining some indicators [109]. Thus, in line with recent scientific literature [110],[111], and [112], the proposed service architecture can integrate IDSs at the network edge to ensure resistance to Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. Here, IDSs may be placed between the Edge Servers and the mobile agents, to filter and analyze encrypted traffic generated by data producers and received by data consumers. In this context, however, note that this contribution does not carefully investigate the behavior of IDS, which will be studied in future research activities.

3.3.3 Performance evaluation

To illustrate the significant potential of the proposed privacy-preserving data dissemination strategy in realistic scenarios, this Section investigates its performance through experimental tests.

The followed Methodology

Differently from our preliminary numerical analysis presented in [2], the cryptographic operation expected for the two state-of-the-art SE algorithms proposed in [94] and [88] and presented in Appendix A and Appendix B, respectively, have been implemented through the Pairing Based Cryptography (PBC) library and executed within a proper experimental computing test environment modeling the conceived data dissemination process.

Conducted tests consider a network with a variable number of base stations (i.e., 4 and 8 cells). Herein, from 10 to 50 data consumers are uniformly distributed. Moreover, the study assumes to consider some data producers, uniformly attached to the base stations. Data producers may need to examine their data stream and publish new data because the network requirements are subject to change. To properly handle with this specification, the reviewed data publications related to the data producers are modeled via Poisson at different rates (i.e., from 40 to 320 new publications/s) during a total time of 180 seconds. Specifically, tests make use of the Message Queuing Telemetry Transport (MQTT) protocol which is based on a publish/subscribe communication mechanism. Publishers are clients who send messages, while subscribers are the ones who receive them. Their interaction is promoted by a central point (i.e., broker) that receives the messages from the publisher and delivers them to the subscribers [113]. Thus, the proposed methodology deploys MQTT to publish new data at different rates and to subscribe for receiving specific data. The impact of i) the number of service subscriptions (equivalent to the number of Trapdoors generated by data consumer

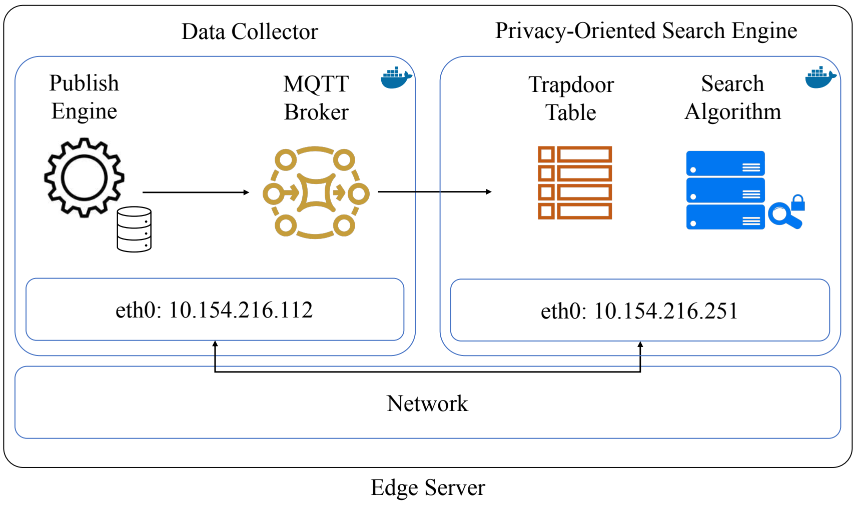


FIGURE 3.10: Testbed setup

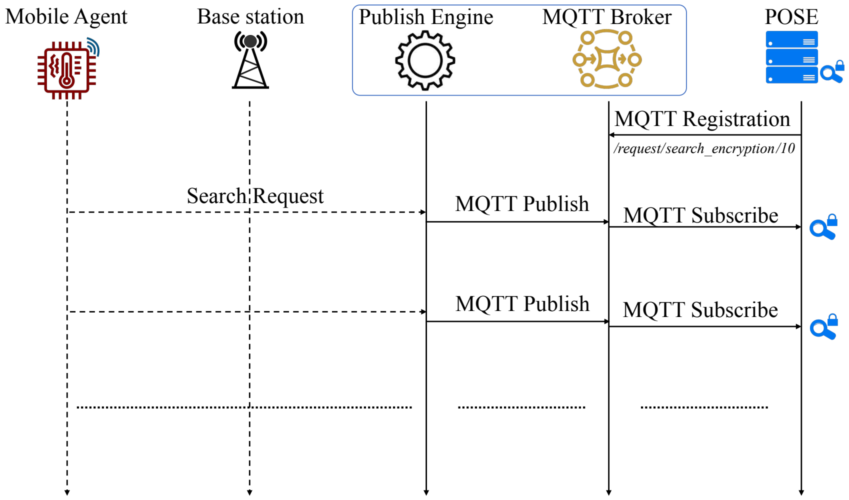


FIGURE 3.11: System emulation scheme.

and stored within each single Trapdoor table), ii) the average amount of data publications over time, and iii) the number of Edge Servers involved into the privacy-preserving data dissemination strategy has been carefully studied by measuring three main Key Performance Indexes (KPIs). The first one is the *average search time*, defined as the average amount of time needed to complete the execution of the Searchable Encryption algorithm over all the service subscriptions stored within the Trapdoor table. The second one refers to the *average delivery delay*, computed as the average amount of time required to deliver published data to the subscriber. While the average search time highlights the computational impact of both the number of Trapdoors and network load on the implementation of the SE function, the average delivery delay extends the previous KPI by also reporting the impact of delivery delays of data across the distributed network. Finally, the third KPI regards the *energy consumption*. In particular, the consumed energy to run SE operations is examined as a function of both the number of subscriptions and the number of new publications changed. The analysis of these KPIs highlights the significant performance gain the proposed architecture achieves against a conventional cloud-based approach. Indeed, it demonstrates the benefits of distributing service subscriptions, collection of published data, SE tasks, and data delivery at the edge of the network rather than deploying a centralized and remote application available in the cloud.

TABLE 3.4: Average Communication Latencies [114] [115].

Network End-Points	Average Communication Latency [s]
Radio interface	0.0036
Network Edge	0.0052
Remote Cloud	0.015

TABLE 3.5: Computational cost of cryptographic operations.

Cryptographic operation	Average execution time [s]	
	SE algorithm presented in [94]	SE algorithm presented in [88]
Encryption	0.02694	0.03264
Decryption	0.02568	0.03339
Search Algorithm	0.02954	0.03233

System setup description

Experimental tests have been carried out on a workstation running the Ubuntu 22.04.1 LTS operating system, with an Intel Xeon Bronze 3106 @1.70GHz processor, 96 GB of RAM and 180 Watt of consumed peak power.

Specifically, the workstation models the most important functionalities of an Edge Server and its interaction with respect to data publishers and other remote Edge Servers. Here, two lightweight networked containers are built by using Docker to model the Data Collector and the POSE entity (see Fig. 3.10).

The Data Collector is in charge of collecting all data published by data producers served by a specific base station. Furthermore, it delivers the collected data to the POSE entity. As depicted in Fig. 3.10, these two tasks are executed by the Publish Engine entity and MQTT Broker, respectively.

On the other hand, the POSE entity hosts the Trapdoor table and implements the SE algorithm. Moreover, the interaction between the Data Collector and POSE entity is implemented through a client-server application, established by using the MQTT protocol version 3.1.1. In particular:

- The Publish Engine uses a C++ script to emulate the reception of data published by IoT agents. Here, data are generated according to a memory-less Poisson process, where the average number of new data published over time in a single cell varies from 40 to 320 new publications/s.

- As soon as a new data is published, the Publish Engine in the Data Collector forwards MQTT Publish messages to the MQTT Broker (i.e., Eclipse Mosquitto message broker *), by using the Paho MQTT C library. Then, the MQTT Broker sends a MQTT Subscribe message to the POSE entity, previously registered to receive MQTT messages on a specific topic (e.g., `/request/search_encryption/trapdoor_number`).

- The POSE entity receives the MQTT Subscribe messages via the Flask API framework and runs the search algorithm by executing a binary file containing the cryptographic operations. Herein, to properly configure the scheme of the two attribute-based SE algorithms (i.e., [94] and [88]), which number of attributes is set to 5, Type A pairings are constructed on the curve $y^2 = x^3 + x$ over the field Z_P. Additionally, the dimensions of the $\mathbb{G}$ and $\mathbb{G}_T$ group elements are set to 1024 bits and to 160 bits for the Z_P ones.

*https://mosquitto.org/

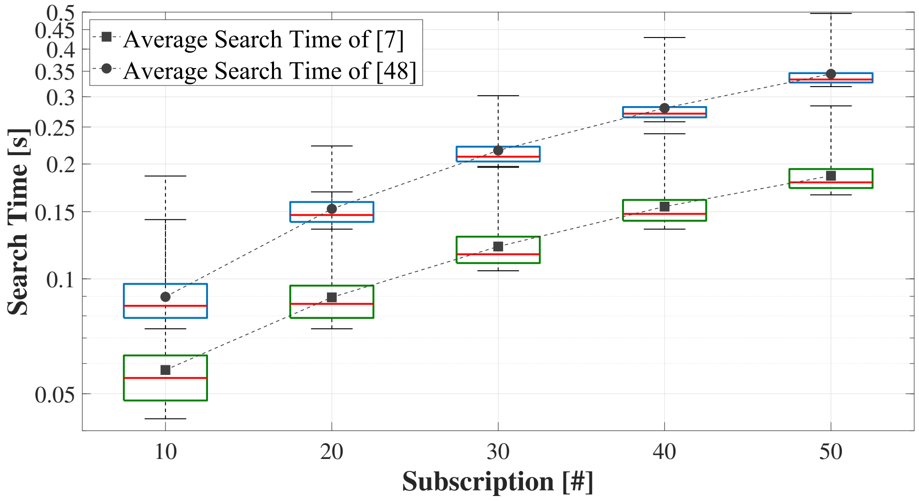


FIGURE 3.12: Search Time

In addition, the implemented experimental setup also emulates communication latencies on the various network segments to measure the average delivery delay. Table 3.3 shows latency values taken from the scientific literature [114] [115].

Analysis of cryptographic operations

This section evaluates the computational cost of cryptographic operations and measures statistical information on search algorithm execution time.

Firstly, by adopting the above-described simulation setup and running 10^2 tests, the average execution time of encryption and decryption operations, as well as the single search accomplishment are calculated and reported in table 3.5. It shows that the algorithm in [94] is less computationally intensive than the algorithm in [88].

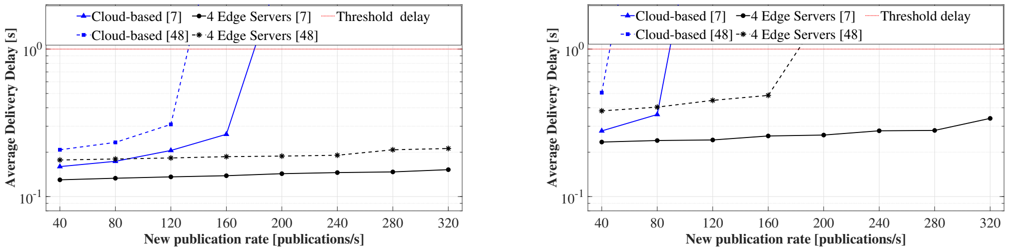
Secondly, with respect to the conceived methodology presented in Section 3.2.1, the average search time is evaluated and defined as the amount of time needed for each POSE entity to execute sequentially the number of subscribed trapdoors. Fig. 3.12 depicts the statistical information of the search time execution run over 10^2 tests. For both the algorithms, the 25th, 50th, and 75th percentiles as well as the lowest value and maximum values, are reported. In addition, the respective average search time value is also shown. Thus, differently from the preliminary numerical results presented in our previous work [2], here the empirical results prove that the search execution time linearly increases with the number of subscriptions. Specifically, Fig. 3.12 demonstrates that the algorithm presented in [94] performs better than the one described in [88]. While, both algorithms highlight a double average search time in the execution of 10 sequentially Trapdoors with respect to the single execution. The results prove that when subscriptions pass from 10 to 50, the average search time triples in [94] and quadruples in [88].

Impact of the Network Load on the Search Time

This section analyzes the impact of network load on the search time execution, employing the simulation setup described in Section 3.3.3. Since POSE entities and cloud servers both have finite processing capability, herein it is proved that by gradually increasing the number of newly published data, the number of queued search executions increases, exposing a longer average search time execution.

Table 3.6: Average Search Time Execution

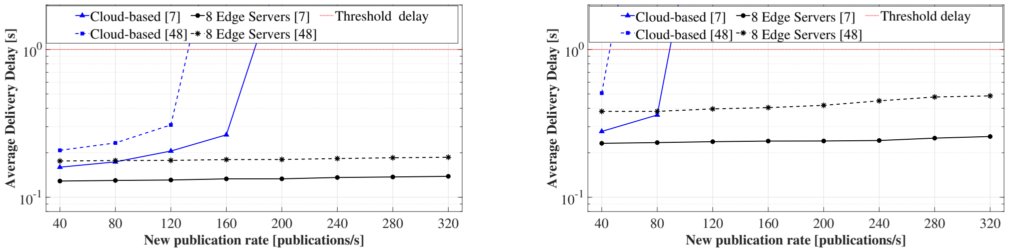
Subscription No. [#]	**Edge Server No. [#]**	**New publication rate [publication/s]**	**Cloud-based Average Search time execution [s]**		**Proposed architecture Average Search time execution [s]**	
			SE algorithm in [94]	SE algorithm in [88]	SE algorithm in [94]	SE algorithm in [88]
10	**4**	**40**	0.0770	0.1116	0.0685	0.1023
		80	0.0908	0.1369	0.0717	0.1049
		120	0.1228	0.2126	0.0746	0.1079
		160	0.1818	13.0792	0.0771	0.1116
		200	3.5898	37.2815	0.0815	0.1133
		240	21.5363	61.2140	0.0840	0.1158
		280	39.0132	84.1684	0.0855	0.1328
		320	43.1322	107.4534	0.0908	0.1369
	8	**40**	0.0771	0.1116	0.0673	0.1010
		80	0.0908	0.1370	0.0685	0.1023
		120	0.1227	0.2126	0.0694	0.1027
		160	0.1818	13.0792	0.0717	0.1049
		200	3.5898	37.2815	0.0718	0.1052
		240	21.5364	61.2140	0.07459	0.1079
		280	39.0132	84.1684	0.0757	0.1102
		320	43.1322	107.4534	0.0771	0.1116
40	**4**	**40**	0.1961	0.4105	0.1729	0.3063
		80	0.2775	53.9431	0.1785	0.3293
		120	33.2482	122.7198	0.1806	0.3742
		160	71.9170	190.0141	0.1961	0.4105
		200	110.0377	256.6541	0.1998	1.7370
		240	147.2419	281.9454	0.2178	19.3829
		280	187.1398	387.6209	0.2194	37.3034
		320	210.0170	445.3315	0.2775	53.9431
	8	**40**	0.1961	0.4105	0.1703	0.3063
		80	0.2775	53.9431	0.1729	0.3063
		120	33.2483	122.7199	0.1762	0.3211
		160	71.9170	190.0142	0.1785	0.3293
		200	110.0377	256.6541	0.1788	0.3438
		240	147.2419	281.9455	0.1806	0.3742
		280	187.1398	387.6210	0.1897	0.4020
		320	210.0170	445.3315	0.1961	0.4105



(A) Average Delivery Delay with 10 subscriptions

(B) Average Delivery Delay with 40 subscriptions

FIGURE 3.13: Average Delivery Delay with 4 Edge Servers



(A) Average Delivery Delay with 10 subscriptions

(B) Average Delivery Delay with 40 subscriptions

FIGURE 3.14: Average Delivery Delay with 8 Edge Servers

Table 5.2 displays the average search time execution, calculated as the average value of all single search executions performed on the newly published data (generated with a specific Poisson distribution rate). Specifically, it shows the average search time execution of both considered algorithms (i.e., [94] and [88]) as a function of number of both submitted Trapdoors and network cells.

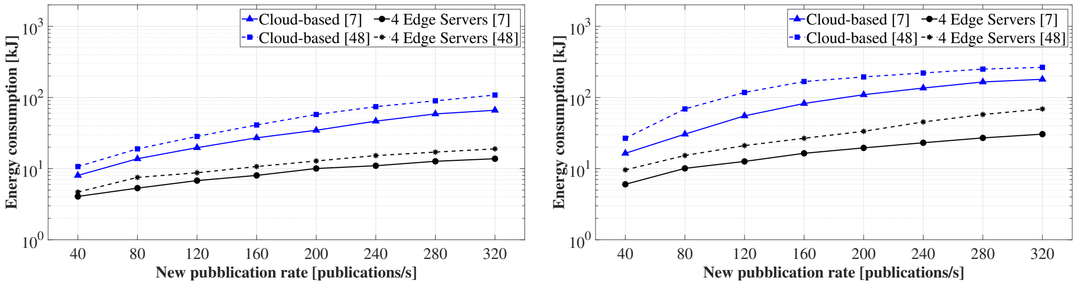
It emphasizes how moving search operations at the network edge would boost search operations on newly published data. Indeed, considering the algorithm in [94] with 10 subscriptions, by passing from 40 to 320 publications/s, the average search time execution within the proposed architecture increases by just a few milliseconds (22 ms and 70 ms with 4 and 8 Edge Servers, respectively). While augments of 43 seconds with the Cloud-based approach. Similarly, the average search time execution for the suggested architecture rises to 53 seconds and 100 milliseconds with 4 and 8 Edge Servers, respectively. While, it increases by 200 seconds in the Cloud-based scenario using the algorithm in [88] with 40 subscribers.

As a result, since the needed search time is not negligible, a technique that distributes tasks at the edge of the network might result in significant advantages.

Average Delivery Delay

By referring to the simulation setup specified in Section 3.3.3, this section compares the average delivery delay in the cloud-based and proposed scenario.

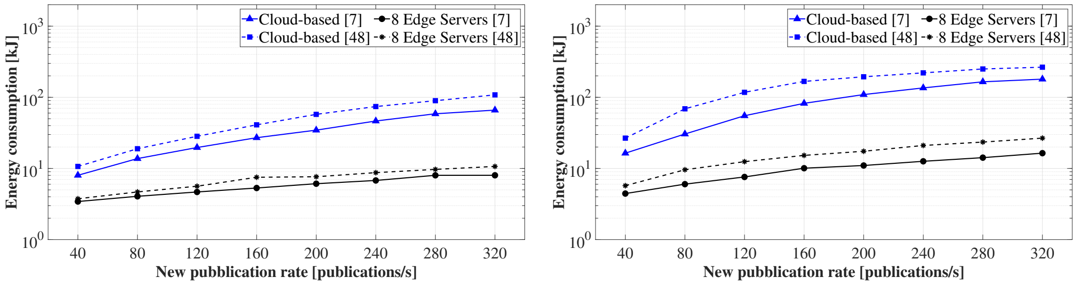
To properly evaluate the average delivery delay, the average communication latencies displayed in table 3.3, the average time required to encrypt and decrypt data recorded in table 3.5, and the average search time execution reported in table 5.2 have been considered. On one hand, the average cloud-based delivery delay is evaluated as the sum of cryptographic operations and latencies due to reach the cloud server. On the other hand, following the proposed architecture, it is calculated as the sum of the following three contributions: cryptographic operations, latency in the radio interface, and latency experienced at the edge of the network. Supposing to evaluate a scenario (e.g., monitoring and control applications) where the maximum acceptable delay is 1 s, Fig. 3.13 and Fig. 3.14 depicts the average delivery delay,



(A) 10 subscriptions

(B) 40 subscriptions

FIGURE 3.15: SE energy consumption with 4 Edge Servers



(A) 10 subscriptions

(B) 40 subscriptions

FIGURE 3.16: SE energy consumption with 8 Edge Servers

respectively in a 4 and 8 cells network, as a function of new published data rates. Specifically, only below threshold values are displayed. In line with the search time execution, the algorithm in [94] achieves shorter delivery delays than the algorithm in [88]. Moreover, results of both 10 and 40 subscriptions highlight how distributing search operations at the network edge allows obtaining a higher tolerable data publishing rate. Indeed, Fig. 3.13 shows that by passing from 10 to 40 subscriptions and using the algorithm in [7], the acceptable new published data rate for the proposed approach is not reduced, differently for the cloud-based one where it reduces by 60%. While, by exploiting the algorithm in [88], a maximum tolerated delivery delay is obtained reducing the new published data rate up to 10% for the proposed approach and 70% for the cloud-based one. Similarly, this happens in an 8-cell network depicted in Fig. 3.14. By comparing the two figures, it is possible to understand that by increasing the number of cells and the new publication rates, the average delivery delay slightly varies in a scenario with 10 subscriptions for both the Cloud-based and proposed approaches. While it significantly differs by increasing the subscription number to 40. Indeed, on the one hand, the proposed service architecture allows to maintain a delivery delay lower than the threshold. On the other hand, the Cloud-based approach allows both algorithms to suddenly reach the threshold value.

Energy Consumption

The SE algorithms require a significant and not negligible amount of time and energy to run, both on Edge Servers and in remote clouds.

To measure the energy consumption due to the execution of *SE tasks*, conducted tests tracked the number of active and pending SE operations over time. Indeed, by dividing the time into small intervals ΔT (e.g., 1 ms each), the energy consumed during each interval is calculated using the percentage of peak power $P(n_{SE})$, based on the number of active/pending SE operations (i.e., n_{SE}), and each time slot ΔT. This allows evaluating the energy consumption E as:

$$E = \sum_{n_{\Delta T}} P(n_{SE}) \cdot \Delta T, \tag{3.13}$$

where, $n_{\Delta T}$ is the total number of observed time slots. In this context, $n_{\Delta T}$ is the amount of time needed to run all the SE operations in a specific scenario.

Fig. 3.15 and Fig. 3.16 illustrate the consumed energy in the function of the number of subscriptions, publication rates, and Edge Servers as well as the time needed to run the SE tasks. These figures confirm that the algorithm presented in [88] consumes more energy than the algorithm in [94]. This is due to the higher computational cost required by the algorithm in [88] to perform SE operations. Moreover, in a scenario with 4 Edge Servers and 10 subscriptions (Fig. 3.15a), the proposed solution permits keeping low energy values, ranging from 5 kJ to 20 kJ. Differently, the usage of the cloud-based method results in a higher energy consumption passing from 10 kJ to 100 kJ. In addition, in a scenario with 8 Edge Servers and 10 subscriptions (Fig. 3.16a), the consumed energy only increases by 8 kJ, passing from 40 to 320 publications/s within the proposed service architecture. Instead, with the cloud-based method, it increases by 90 kJ. The same pattern occurs in a network with 8 Edge Servers and 40 subscriptions (Fig. 3.16b). Here, the energy consumption values are lower than 20 kJ with the proposed approach, while it rises by a factor of ten, passing from 40 to 320 publications/s within the cloud-based one. By comparing the two figures, it is clear that the energy consumption resulting from the execution of SE operations in a scenario with 10 subscribers, for both the Cloud-based and proposed approach, marginally varies by increasing the number of cells and the new publishing rates. While it considerably differs by raising the number of subscriptions to 40, especially for the cloud-based methods.

Chapter 4

Radio Frequency Fingerprinting

Wireless networks are foundational to IoT applications but remain vulnerable to multiple security threats due to the open nature of the wireless communication channel. This openness exposes devices to unauthorized access and privacy violations [116]. Thus, ensuring authenticity within wireless networks is a major challenge, as traditional cryptographic methods, while effective, often demand significant computational resources and require modifications to devices' firmware and software [117].

Within this context, PLA has emerged as a promising solution, leveraging unique physical layer characteristics from transmitted signals to verify a device's identity [117]. These inherent attributes do not require changes or computational overhead at the transmitting device, and thus act as unique identifiers [117]. RFF, in particular, harnesses the specific hardware traits of a transmitter, which are inadvertently embedded in the transmitted signal. These characteristics allow a passive receiver to uniquely identify the transmitting device [118]. Components such as oscillators, amplifiers, and modulators introduce unique variations in the signal's phase and frequency, without affecting its quality [119]. By analyzing these characteristics, RFF can uniquely identify—or fingerprint—a device, much like a biometric identifier for humans. This process involves collecting Radio Frequency (RF) emissions at a receiver, extracting relevant features, and training a model to recognize specific transmitters in the wild [119].

Wireless channel conditions, however, pose a substantial challenge for RFF-based device identification, particularly in wideband communication systems [120]. Traditional RFF approaches extract features based on both device-specific hardware impairments and channel conditions, with channel variations frequently overshadowing the intended device-specific signal markers [121]. Consequently, RFF accuracy diminishes with channel fluctuations, introducing high location-dependency [121]. Recent research indicates that deep learning (DL)–based, image-inspired RFF models can achieve more consistent performance than traditional models by mitigating the multipath effects induced by the wireless channel [122]. This approach preprocesses physical layer data (I-Q samples) into image-like representations for input into image classification algorithms such as Convolutional Neural Networks (CNNs). Due to their robustness, image-based RFF models have been widely adopted in various wireless environments [123][124][125].

4.1 Background and Related Works

In this section, preliminary concepts that will be useful for later sections, i.e., digital modulation techniques and CNNs are introduced.

4.1.1 Digital Modulation

Wireless communication systems use digital modulation techniques to convert baseband to high-frequency signals suitable for transmission over the wireless channel [126]. In detail, a digital modulation scheme generates a modulated signal characterized by an in-phase (I) and a quadrature (Q) component, commonly represented as complex IQ values $I + jQ$, where I and Q denote the real and imaginary parts, respectively. The transmitter maps a bit sequence into symbols and then I-Q samples using a specific modulation scheme. The receiver decodes the original bit sequence from the received I-Q samples, by associating to the received IQ value the symbol characterized by the minimum error, under the assumption that the noise affecting the received signal is minimum. In the remainder of this work, the Binary Phase-Shift Keying (BPSK) modulation scheme is considered, according to which, the bit values $\{0, 1\}$ are mapped into the symbols $+1, -1$. It is worth noting that BPSK requires only the in-phase component ($I = \pm 1$) while the quadrature component is set to zero $Q = 0$.

4.1.2 Deep Learning

Deep Learning (DL) techniques have been recently adapted to the wireless communication and image recovery domains [127]. In this context, CNN models demonstrate to achieve better performances, especially for image processing tasks [128]. Indeed, by exploiting their ability to learn and extract features, CNNs are deployed into a wide range of scenarios, including image classification [129]. Specifically, CNNs are a widely used DL architecture mostly adopted in computer vision applications [127] and image processing [129]. In this context, by using images or labeled data, CNNs learn to generate hierarchical representations of the data, which can then be used effectively for accurate and reliable target classification [130] reaching high accuracy [128]. CNNs consist of three types of layers, i.e., convolutional, pooling, and fully-connected layers, where neurons perform convolutional operations and enhance the performance of the model through a process of iterative learning [131]. Convolutional layers primarily handle feature extraction by applying convolutional filters to the input data, producing a corresponding feature map. Subsequently, the pooling layers are used to reduce computational overhead by downsampling the spatial dimensions of the feature map. Finally, fully connected layers are responsible for high-level feature processing and for making final predictions [131]. CNNs have gained popularity in the literature mainly for their remarkable performance in classifying images.

4.1.3 Related Works on Image-based RFF

RFF have recently gained popularity in the scientific community as a novel approach for authenticating RF devices by analyzing their unique Physical (PHY)-layer signal characteristics [117]. Overall, scientific approaches dealing with transmitter identification from PHY-layers signals can be divided into two primary categories: i) traditional methods based on statistical analysis, and ii) approaches leveraging DL algorithms [118]. Traditional RF fingerprinting methods relying on customized features often face challenges in generalizing to real-world environments [132]. In contrast, DL automatically extracts complex features by directly using as input values the raw I-Q data, enhancing accuracy and adaptability [133]. While effective in many scenarios, this strategy produces sensitive fingerprint models, that struggle to adapt to varying channel conditions, mobility, and power cycling of RF devices [122]. In this context, image-based RFF systems, converting raw I-Q samples into 2-D or 3-D images, have demonstrated superior identification performance over previous techniques under challenging channel conditions and across power cycles of the devices [122], [134]. Image-based RFF systems have been recently used also for the detection of several attacks in wireless scenarios, e.g., Adversarial Machine Learning (AML) [135] and jamming [124]. To mention a few relevant works using image-based RFF, Papangelo et al. [135] investigate the effectiveness of AML techniques in attacking and improving the robustness of image-based RFF systems, demonstrating that adversarial training enhances system resilience against attacks like Fast

Gradient Signed Method (FGSM) and Generative Adversarial Networks (GAN) with a minor impact on classification accuracy. Alhazbi et al. [123] propose a solution for early jamming detection and identification in mobile scenarios, leveraging DL to analyze image transformed I-Q samples at the PHY layer and accurately detect and classify jamming types, including Gaussian noise and tone jamming. Along the same line, Sciancalepore et al. [124] perform jamming detection by focusing on an indoor scenario and extending the work in [123] with different modulation techniques, enhanced adversary models, and sparse autoencoders on image transformed I-Q samples. Moreover, Irfan et al. [125] present an approach for detecting jamming signals in Power Line Communication (PLC) systems by converting PHY-layer I-Q samples into images and applying CNN for classification.

Overall, although image-based RFF approaches have gained popularity in the PHY-layer security domain, current research primarily focuses on detecting jamming attacks, while the impact of interferences (intentional or not) on the accuracy of RFF remains unexplored. Moreover, none of these works analyzes the impact of an out-of-band interference on the performance of RFF. Indeed, although the Bit Error Rate (BER) is minimally impacted when jamming occurs in close channels, RFF is much more fragile, and can be potentially corrupted even when interference does not occur on the same channel and bandwidth of the main communication.

4.2 Frequency Matters: On the Impact of Carrier Frequency on Privacy in Radio Fingerprinting

On one hand, RFF presents a compelling solution for ensuring authenticity in wireless communications; on the other hand, it raises privacy concerns, particularly regarding data security. Devices operating on a particular frequency generate RF emissions that are captured by a dedicated receiver, which then creates radio fingerprints and stores them on a server, such as a database within a security service [136]. Unauthorized access to such servers could lead to significant data leakage [137]. Since each device has a unique fingerprint, any leakage of the RFF model—or portions of it—may allow adversaries to infer individuals' locations, behaviors, and even social interactions, thereby facilitating surveillance and infringing on personal privacy [138].

Despite these risks, there has been little evaluation of how partial leakage of the RFF model might impact devices' anonymity and privacy. Assuming an adversary gains access to an RFF model, no research has yet assessed RFF performance when there is no information about the carrier frequency on which the model was trained. Existing studies commonly assume that a device's RFF model is unique and remains unaffected by operational factors, such as the frequency at which the receiver collects RF emissions. However, the literature has not sufficiently investigated two critical aspects: (i) the consistency of a device's RFF model across various carrier frequencies, and (ii) the effectiveness of tracking attacks conducted via RFF when there is a mismatch between the carrier frequency used for training and that used during testing—a common scenario in cases of partial RFF model leakage.

The work presented herein explores the impact of partial leakage of an RFF model on the overall RFF process. Through a series of controlled experiments utilizing SDRs and state-of-the-art image-based RFF models, it is demonstrated that an RF device's RFF model is highly dependent on the carrier frequency used in communications. While RFF models generated at closely related frequencies often display similarity, significant discrepancies arise as frequency differences increase. When attackers attempt to utilize a leaked RFF model without knowledge of the original carrier frequency, these discrepancies lead to notable declines in RFF accuracy, potentially reducing performance to random guessing, even when only a small number of devices are present in the network.

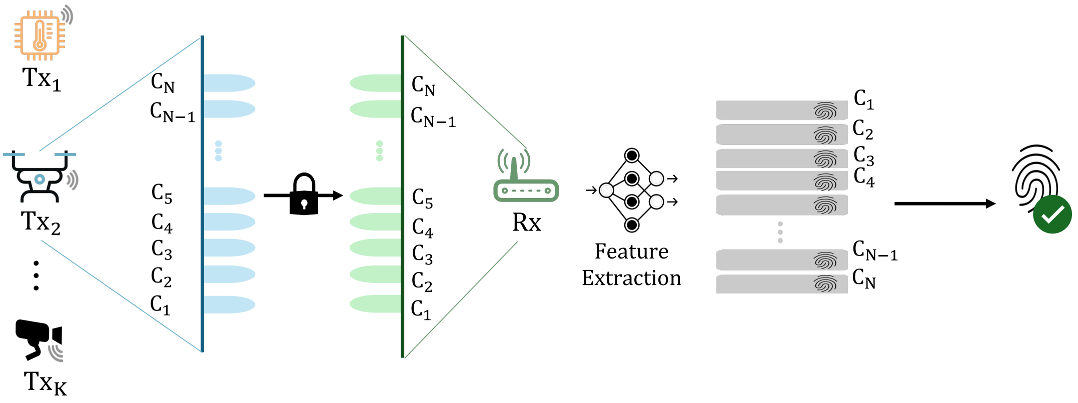


FIGURE 4.1: Reference scenario: the transmitter communicates with the receiver on a pseudo-random channel through encrypted communication.

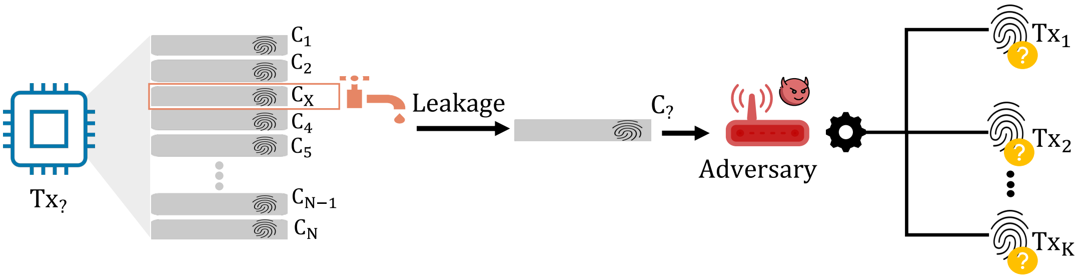


FIGURE 4.2: Adversary model: the adversary is challenged to identify the transmitter by resorting to RFF while exploiting (leaked) information associated with the DL-based RFF model of the transmitter on channel c_x.

This analysis shows, for the first time, that lack of knowledge about the training frequency renders a leaked RFF model nearly unusable for attackers, compelling them to develop a new device profile, which incurs both time and cost. Frequency hopping is thus proposed as a viable approach for RF devices to counteract RFF-based tracking attacks, potentially preserving anonymity and location privacy. To encourage reproducibility and further exploration of these findings, all related data is released as open-source at [139].

4.2.1 Reference Scenario and Adversary Model

Reference Scenario

Fig. 4.1 depicts the reference scenario, including one or more RF transmitters and one RF receiver. The RF devices communicate wirelessly on a pseudo-random channel $c(t)$ chosen by the transmitter and the receiver as a function of previously established secrets. The main objective of a transmitter is to stay anonymous to all devices in the network except to the receiver, which in turn performs the transmitter's physical-layer authentication via RFF. To this aim, the receiver resorts to a dataset of pre-trained models $\mathcal{M} = \{M_1, \ldots, M_N\}$ constituted by N RFF models, each of them referring to a specific channel, which is stored on a server. Therefore, the receiver tunes to the pre-defined channel c_x, select M_x, with $x \in [1, N]$, initiates the reception of the signals from the transmitter, and finally validates the transmitter applying M_x. Specifically, in line with the contribution in [135], the receiver deploys image-based RFF techniques using CNN pre-trained DL models for image classification achieving robustness against channel fluctuations and mobility.

Adversary model

Fig. 4.2 shows the adversary model. It is assumed that one of the RFF models $M_a \in \{M_1, \ldots, M_N\}$ with $a \in [1, N]$ has been leaked to the adversary, as a result of the adversary gaining unauthorized access to the server where the RFF model is stored. Specifically, the adversary is aware of the RFF model, but they do not know which channels are associated with that specific RFF model. Thus, the adversary aims to identify the transmitter (i.e., fingerprint) by leveraging the leaked model M_a. Note that, given the high-security constraints, i.e., a random selection of the communication channel, transmitter anonymity, and unawareness of the association of a model to a particular channel, the adversary task is particularly challenging. In this work, the impact of M_a leakage on transmitter anonymity and the influencing configuration parameters are investigated.

4.2.2 Deployed Methodology

This section reports the methodology used by both the legitimate receiver and the adversary to assess the impact of partial leakage of the RFF model and carrier frequency on the RFF. The deployed approach involves converting raw samples of the signal taken from the radio spectrum into images [122], [123], [134], [140], [141] due to its remarkable robustness to noise and other side effects. In line with such a methodology, the RFF problem is transformed into an image classification problem. The main steps involve: (i) *IQ sample collection*, (ii) *images generation*, and finally (iii) *multi-class classification*.

IQ sample collection

The IQ samples are collected by tuning both the transmitter and the receiver on the same channel. The BPSK modulation scheme is considered, where the in-phase component assumes as a value either -1 or +1, while the quadrature component is always zero. By mapping the I component and the Q component to the real and imaginary parts of a complex number, respectively, the BPSK decoding process follows Eq. 4.1:

$$x(t) = \begin{cases} -1 \; cos(2\pi f_0 t), & \text{if b = 0,} \\ +1 \; cos(2\pi f_0 t), & \text{if b = 1,} \end{cases} \tag{4.1}$$

where $x(t)$ is the transmitted modulated signal, f_0 is the carrier frequency, and b is the bit value. Thus, given a carrier frequency f_0, the couples [-1, 0] and [1, 0] represent the theoretical position of the received IQ samples in the IQ plane. However, due to radio imperfections, the collected IQ samples are distributed in the IQ plane, generating a specific pattern that identifies the device's fingerprint.

Image Generation

The image generation phase processes the collected raw IQ samples and generates gray-scale images following the baseline procedure described in [122]. Specifically, the procedure involves collecting K IQ samples and then dividing the IQ plane and the clouds of points created by such IQ samples into $N \times M$ tiles, with the values of N and M determining the image dimensions. Afterward, for each tile (m, n), the IQ samples that fall in the tile (bivariate histogram) are counted. To guarantee that such value maps to a correct pixel value in the generated image, if the count exceeds 255 (i.e., the maximum possible value of the pixel of an image), the value is truncated to 255. To this aim, it is fundamental to calibrate the number of IQ samples per image to minimize the loss of information, i.e., too many tiles exceeding 255 samples.

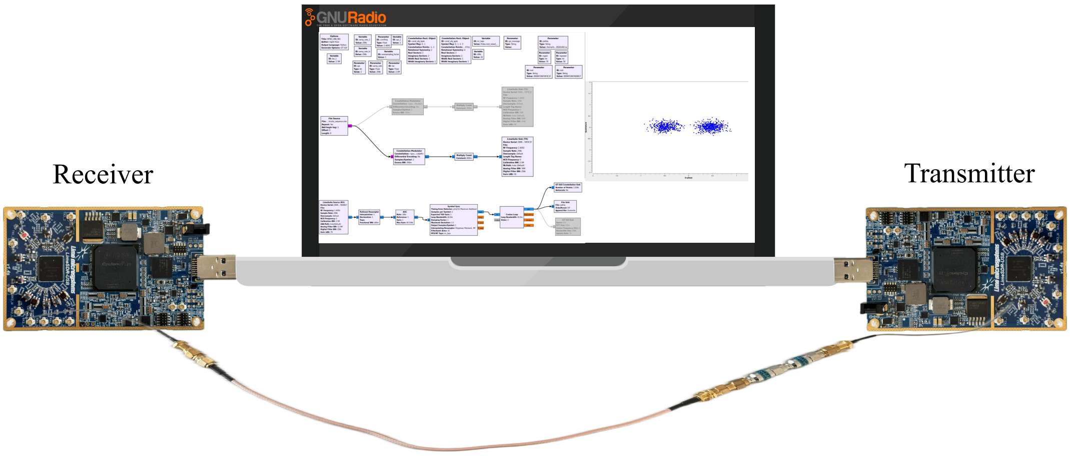


FIGURE 4.3: Experimental testbed setup.

Multi-class Classification

This task involves classifying images generated during the previous phase. Specifically, in line with the adversary model described earlier, the aim of the proposed multi-class classification problem is to identify the transmitter device. Firstly, the collected IQ samples are divided into three subsets, i.e., training, validation, and testing, Secondly, state-of-the-art CNNs pre-trained on the ImageNet database [142] are considered, i.e., Alexnet, Resnet-18, Resnet-50, Resnet-101, Inceptionv3. The procedure used their implementation in MatLab2023b, where the input and output layers are adapted to fit the classification problem. Specifically, the input layers are re-sized to fit the size of the images generated from raw IQ samples, while the output layers are re-designed to accommodate the number of classes in the specific experiment.

Investigated Key Performance Indicators

Two main experiments are considered. Firstly, the impact of the carrier frequency on the RFF model of a device is investigated. Specifically, it aims to assess to what extent the fingerprint of a particular device changes when changing the carrier frequency. Thus, a multi-class classification task is performed where the number of classes is coincidental with the number of tested channels. Secondly, the impact of partial information leakage of the RFF model is evaluated by analyzing the mismatch in the training and testing channel used for the RFF, in line with the adversary model discussed in Sec. 4.2.1. Herein, a multi-class classification problem where the number of classes is coincidental with the number of distinct devices in the setup is performed. For each test and device, the RFF model is trained on the IQ samples acquired at a specific channel, and it is tested using the IQ samples acquired at a different channel. Thus, it is denoted δ as the absolute value of the difference between the channel considered for training and the one used for testing, i.e., $\delta = |ch_{train} - ch_{test}|$, and the performance of the RFF models for increasing values of δ is evaluated.

4.2.3 Performance Evaluation

Experimental Testbed

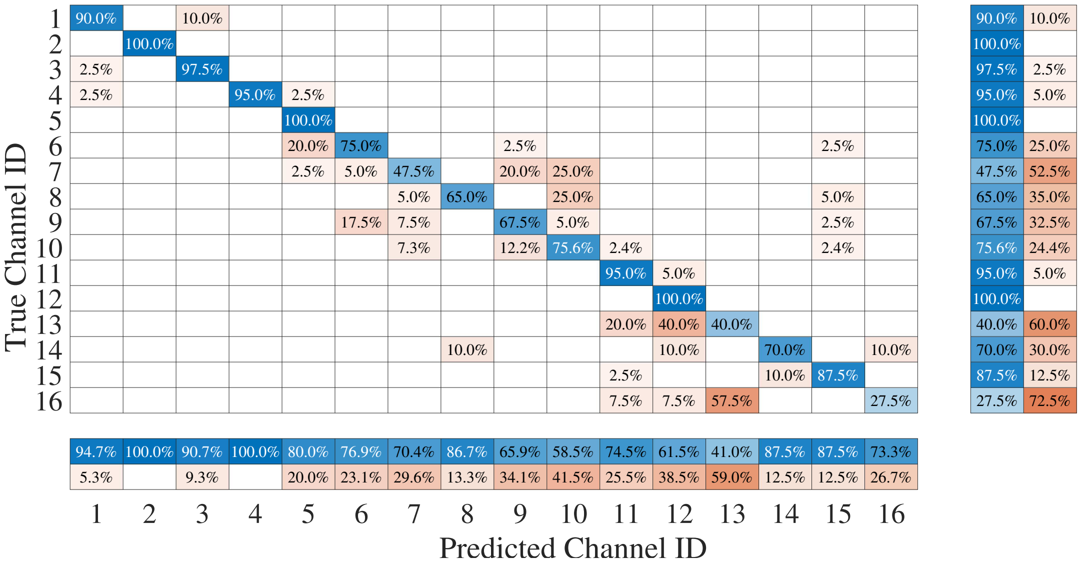
The experimental testbed used to collect the IQ samples resorts to four LimeSDRs devices. The considered SDRs feature the *LMS7002M* RF Transceiver, capable of running any wireless standard and mobile communication, including WiFi and 4G [143]. The SDRs are connected to a Ubuntu 22.04 workstation, equipped with a 12th Gen Intel(R) Core(TM) i7 @2.70 GHz processor. The conducted experiments utilize a direct wired connection between the transmitter and the receiver. In fact, investigating the impact of carrier frequency mismatch on RFF performance is challenging due to various factors that influence the RFF accuracy in real-world environments, such as multipath propagation, shadowing, and interference. The use of the wired connection minimizes the impact of such noise sources and allows to focus on the specific effect of carrier frequency on the RFF, enabling a more reliable analysis of its influence on system performance. At the same time, note that this setup constitutes an extreme advantage for the adversary: indeed, as a legitimate receiver, the adversary can also focus only on the RFF task, without worrying about the mentioned noise figures.

To drive the behavior of the SDR, the *GNU Radio 3.10* software is used, offering the possibility to configure the radios with the desired communication parameters. Thus, the transmitter and the receiver gain are set to 50 dB and 70 dB, respectively. The communication frequencies and channels defined by the IEEE 802.15.4 communication technology [144] and used by devices compliant with the Zigbee specification [145] are considered. Thus, 16 channels in the frequency range $2405-2480$ MHz are used, each characterized by a bandwidth of 2 MHz and an inter-channel spacing of 5 MHz, according to the IEEE 802.15.4 standard specification. The transmission chain defined on GNURadio consists of four blocks: i) a *File Source*, used to generate a message consisting of a string of 256 bytes with incremental values; ii) a *Constellation Modulator*, configured to handle the BPSK modulation scheme; iii) a *Multiply Constant*, used to adjust the amplitude of the signal to avoid saturation, and iv) the *LimeSuite Sink*, where the radio signals are up-converted to the selected carrier frequency, with a sample rate of 256K samples per second. On the receiver, six main blocks are defined: i) the *LimeSDR Source*, used to receive the radio signal at the selected carrier frequency; ii) a *Rational Resampler*, acting as a filter; iii) an *AGC*, used for mitigating channel fluctuations; iv) a *Symbol Sync*, in charge of decoding the digital signal; v) a *Costas Loop*, deployed for phase and frequency mitigation offsets, and vi) a *File Sink*, storing the output of the whole reception chain into a ".iq" file.

Specifically, a selected LimeSDR is deployed as the receiver while the other three radios are connected alternately to act as transmitters (as depicted in Fig. 4.3). For the sake of clarity, to avoid the influence of power cycling on the RFF [122], the data collection campaign is conducted without switching off the receiver LimeSDR. Overall, 60 tests lasting 90 seconds are run for each of the 16 channels on each transmitter. Subsequently, all the collected data are uploaded to a centralized server for running the tests. The collected data are available open-source at [139]. For the data analysis, the High-Performance Computing (HPC) cluster available at TU/e in Eindhoven (NL), providing 2 GPUs Tesla V1000 with 256 GB of RAM is used. Moreover, $K = 10^6$ IQ samples per image are utilized and images of size $M \times N = 225 \times 225$ are obtained, in line with the size of the images of the ImageNet database in MATLAB2023b. For the classification procedure, the 60%, 20%, and 20% of the data for training, validation, and testing, respectively are used, and due to the tested number of channels, the performance of the various RFF models are tested by considering values of δ in the interval $\delta = [0, 15]$.

Fingerprint Robustness to Carrier Frequency

This section investigates the performance of a model trained on a specific channel when applied to data from other channels. The analysis begins with a preliminary example using one device and a single CNN model, specifically AlexNet. A model was trained on images generated from all 16 available channels, and then tested on images from each individual channel. The training set consists of 84 minutes of measurements per channel, while the test set includes 6 minutes, totaling 560 images for training and 40 images for testing per channel. Figure 4.4 displays the resulting confusion matrix.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
1	90.0%		10.0%														90.0%	10.0%
2		100.0%															100.0%	
3	2.5%		97.5%														97.5%	2.5%
4	2.5%			95.0%	2.5%												95.0%	5.0%
5					100.0%												100.0%	
6					20.0%	75.0%			2.5%						2.5%		75.0%	25.0%
7					2.5%	5.0%	47.5%		20.0%	25.0%							47.5%	52.5%
8							5.0%	65.0%		25.0%					5.0%		65.0%	35.0%
9						17.5%	7.5%		67.5%	5.0%					2.5%		67.5%	32.5%
10							7.3%		12.2%	75.6%	2.4%				2.4%		75.6%	24.4%
11											95.0%	5.0%					95.0%	5.0%
12												100.0%					100.0%	
13											20.0%	40.0%	40.0%				40.0%	60.0%
14								10.0%				10.0%		70.0%		10.0%	70.0%	30.0%
15											2.5%			10.0%	87.5%		87.5%	12.5%
16											7.5%	7.5%	57.5%			27.5%	27.5%	72.5%
	94.7%	100.0%	90.7%	100.0%	80.0%	76.9%	70.4%	86.7%	65.9%	58.5%	74.5%	61.5%	41.0%	87.5%	87.5%	73.3%		
	5.3%		9.3%		20.0%	23.1%	29.6%	13.3%	34.1%	41.5%	25.5%	38.5%	59.0%	12.5%	12.5%	26.7%		

FIGURE 4.4: Alexnet accuracy confusion chart. Alexnet is trained and tested on 16 channels (one measurement per channel), using a test set of 40 images per channel. The confusion chart shows high accuracy when the test set is coming from the same channel as the training set (diagonal).

The results indicate that classification accuracy is highest when the test set data originates from the same channel as the training set data. Furthermore, the confusion matrix reveals that misclassifications typically occur with adjacent channels; for instance, channel 13 is predicted as channels 11, 12, and 13 in 20%, 40%, and 40% of cases, respectively. Similarly, channel 7 achieves an accuracy of about 47% on the same channel, while the remaining 53% of predictions are distributed over channels 5, 6, 9, and 10. This pattern holds consistently across all channels, with non-diagonal cells only showing notable accuracy when close to the diagonal, suggesting that neighboring channels retain similar feature representations for the model.

Minor fluctuations in classifier accuracy are observed, likely attributable to unpredictable real-world variations in channel conditions. Despite these fluctuations, the overall classification trend remains consistent across channels. After training a model on a specific channel, device fingerprinting classification errors predominantly occur with adjacent channels, while misclassifications on distant channels are rare. Notably, mispredictions happen when features from a transmitter on one channel resemble those on another channel, making a model associated with a particular channel (M_a) capable of identifying a transmitter on an adjacent channel. Some exceptions are observed, such as channel 6 being predicted as channel 15, a phenomenon that warrants further investigation.

The analysis also examines the maximum offset between the training and testing channels that leads to mispredictions. For example, as seen in Fig. 4.4, channels 1, 2, and 6 show maximum offsets of 2, 0, and 9, respectively. This analysis considers the maximum offset independently of the misprediction error, following a conservative approach where most mispredictions are characterized by low accuracy.

For comparison, multiple CNNs were evaluated, including AlexNet, Inceptionv3, ResNet-18, ResNet-101, and ResNet-50. Figure 4.5 presents the maximum channel offset that yields a misprediction greater than zero as a function of the reference channel. This figure confirms that only adjacent channels tend to retain transmitter features effectively, with the gray-shaded area primarily concentrated within a range of ± 5 (maximum offset). Some exceptions are notable, such as channels 5 and 14, which experience larger offsets across all networks. Importantly, this analysis follows a conservative assumption of considering mispredictions greater than zero regardless of their values. For instance, channel 6 has a maximum offset of

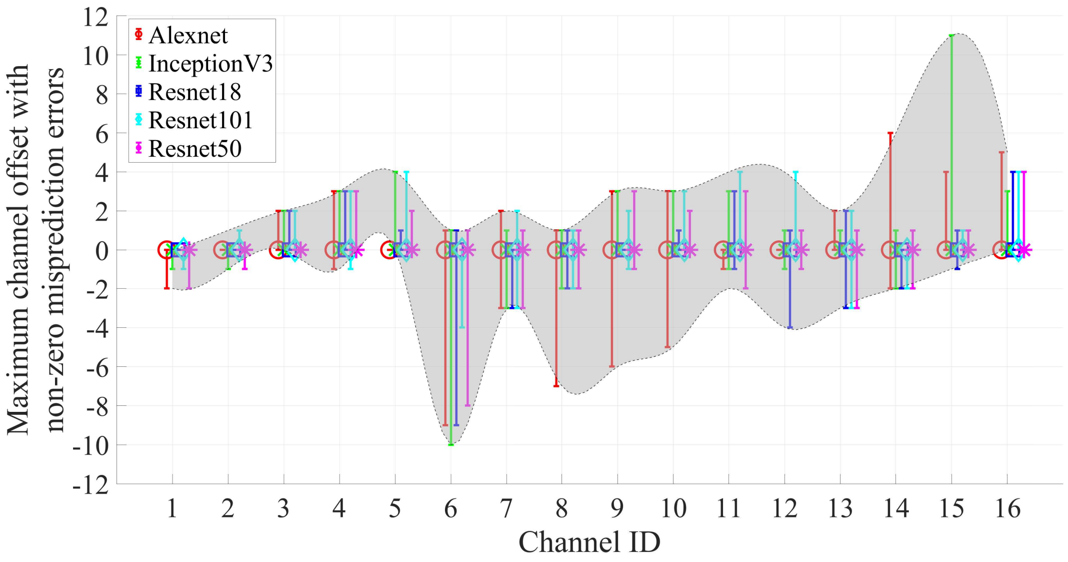


FIGURE 4.5: Maximum channel offset with misprediction greater than zero, considering various CNN.

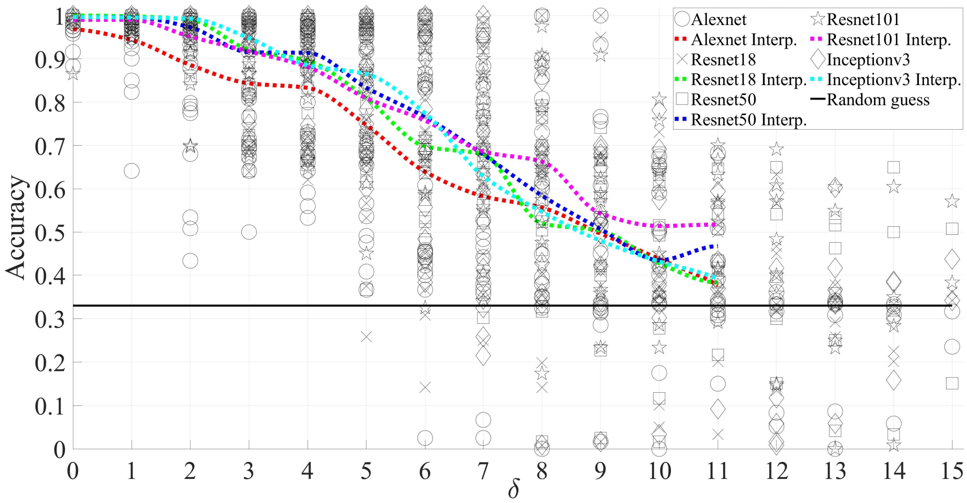


FIGURE 4.6: RFF Accuracy at various channel distances $\delta \in [0, 15]$, using various CNN.

9 (channel 15) with only a 2.5% misprediction rate.

Overall, the findings confirm that features extracted from signals on a specific channel can be effectively used to test a model on the same or neighboring channels, albeit with lower accuracy on adjacent channels.

Tracking Attacks

The focus is on tracking attacks, specifically identifying a device from its leaked profile across various channels during the training and testing phases. In this setup, three transmitting devices and one receiver are used. The configuration follows the previous test approach, with a training set comprising 252 minutes (84 minutes per device) of measurements per channel across 16 channels, and a test set containing 18 minutes (6 minutes per device) of measurements per channel.

Various CNNs models are trained on measurements from a single transmitter on a specific channel, and subsequently evaluated by identifying the same transmitter among others using test measurements obtained on channels at varying distances, denoted as $\delta \in [0, 15]$.

Fig. 4.6 illustrates the accuracy of the CNN in correctly identifying the transmitter, represented as a function of the offset δ between the training and testing channels. The optimal scenario occurs when $\delta = 0$, where testing is conducted on the same channel as training, resulting in a high likelihood of correct identification, with accuracy exceeding 0.95 across all CNNs considered. At an offset of $\delta = 2$, average accuracy (represented by dashed lines) for different networks remains above 0.9, though with increased variance (0.5 to 1). However, accuracy declines as the offset increases, eventually approaching random guess levels (0.33) at an offset of approximately 11.

These findings confirm that models trained on RF device data are effective in identifying devices primarily around the carrier frequency at which data was collected, with diminishing utility on more distant carrier frequencies. Conversely, RF devices could maintain anonymity more effectively by hopping between available channels, thereby increasing the time required to build a reliable profile for RFF.

4.3 Jamming Echoes: On the Impact of Out-of-Band Interference on Radio Frequency Fingerprinting

The vast majority of the literature on RFF suggests performing RFF when the quality of the link between the transmitter and the receiver is high (low BER), in order to minimize the effect of the multipath fading. However, to the best of the authors' knowledge, no research has yet explored the effect of (intentional) jamming or (unintentional) interfering signals active near the communication bandwidth of the transmitter-receiver link.

This work systematically investigates the impact of out-of-band interference on the accuracy and robustness of RFF systems. Using extensive controlled real-world experiments with Software-Defined Radios (SDRs) and state-of-the-art image-based RFF models, this study evaluates the effects of an interfering source (or jammer) on RFF. Results demonstrate that out-of-band interfering signals, while only marginally affecting the BER of the communication link, significantly degrade RFF performance, reducing system accuracy to random chance levels. This analysis highlights out-of-band interference as an additional challenge for achieving reliable device identification from PHY-layer data in practical deployments.

4.3.1 Reference Scenario

The reference scenario considers N devices transmitting RF signals over the air to communicate with one another. For ease of discussion, signals are assumed to be modulated according to the BPSK modulation scheme, although the considerations apply independently of the specific digital modulation technique. The analysis does not consider any specific carrier frequency used for communication for an analysis of the impact of the carrier frequency on these findings.

Additionally, an RF receiver is deployed specifically to collect the signals emitted on the wireless spectrum. As part of the network, the receiver always knows the frequency at which a specific communication may occur, enabling it to collect the raw PHY-layer information (IQ samples) corresponding to such transmissions. Consequently, the receiver gathers IQ samples corresponding to valid received packets and delivers them to a central processing unit, which is responsible for classifying the device emitting such packets using RFF. This scenario exploits state-of-the-art image-based RFF models, similar to the one adopted in [134], chosen for its enhanced robustness to channel variation and real-world effects characterizing embedded systems, such as radio reboot [122] and firmware reload [146]. Moreover, given that the RFF system knows the RF profile of all devices capable of transmitting on the wireless channel, it employs multi-class classification via CNNs to classify the device(s) that emit signals.

In alignment with standard RFF research, the assumption is that RF profiles are generated using wireless signals collected before deployment in a controlled, interference-free environment. However, at runtime during testing, wireless interference may occur both in-band, i.e., on the same channel(s) as the regular communications, and out-of-band, i.e., on frequencies close but not coinciding with those within the bandwidth of the regular communication channel. Throughout this paper, these interferences are referred to as *jamming*, irrespective of the nature of the interference, which could be unintentional (benign) or intentional (malicious). The impact of such out-of-band interference on the accuracy of image-based RFF is examined in the following sections.

4.3.2 RFF Methodology

This section describes the methodology used to assess the impact of out-of-band interference on the RFF. The considered approach transforms I-Q samples into images, in line with state-of-the-art methods relying on image-based RFF [122], [123], [134], [140], [141]. Image-based RFF provides considerable resilience against multi-path and other disturbances. Furthermore, utilizing state-of-the-art tools for RFF aligns with the main objective of this research, namely, demonstrating the impact that out-of-band interference has on RFF.

In accordance with this methodology, the RFF problem is reformulated as an image recognition problem. The methodology involves three main steps: (i) *Data Collection*, (ii) *Image Generation*, and (iii) *Multi-class Classification*.

Data Collection

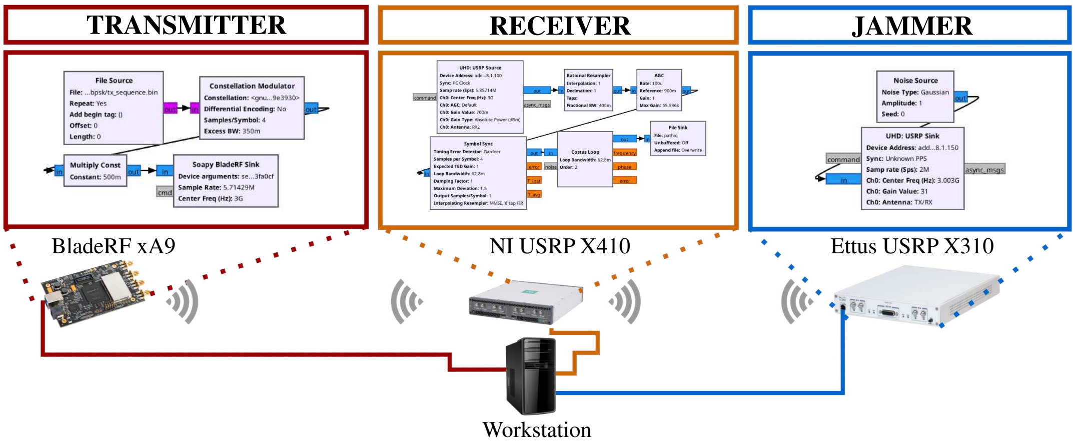
Raw PHY-layer data is collected in the form of IQ samples from the wireless channel by aligning both the transmitter and receiver to the same frequency channel. The BPSK modulation technique is employed, where the in-phase component takes on values of either -1 or +1, while the quadrature component is zero. The I and Q components are mapped to the real and imaginary parts of a complex number, respectively, as outlined in Eq. 4.2.

$$s(t) = \begin{cases} -1 \ \cos{(2\pi f_c t)}, & \text{if b} = 0, \\ +1 \ \cos{(2\pi f_c t)}, & \text{if b} = 1, \end{cases} \tag{4.2}$$

Here, $s(t)$ represents the transmitted signal, f_c denotes the carrier frequency, and b indicates the bit value. For a given carrier frequency f_c, the pairs [-1, 0] and [1, 0] correspond to the theoretical positions of the transmitted I-Q samples on the I-Q plane. However, due to imperfections in radio hardware and fluctuations in wireless propagation, the actual received I-Q samples are dispersed across the IQ plane, forming a pattern that embeds a unique fingerprint for the device.

Image Generation

This step involves processing the acquired raw I-Q samples to produce Red-Green-Blue (RGB) images, in accordance with the baseline procedure outlined in [122]. The process entails slicing the received IQ samples into chunks of $K = 10^5$ IQ samples and dividing the IQ plane, along with the corresponding point clouds produced by the IQ samples, into $Y \times J$ tiles. The parameters Y and J determine the dimensions of the final image. For each tile $i_{y,j}$, the quantity of IQ samples contained within the tile is determined, generating a bivariate histogram.



File Source
File: ...bpsk/tx_sequence.bin
Repeat: Yes
Add begin tag: ()
Offset: 0
Length: 0

Constellation Modulator
Constellation: <gnu...9e3930>
Differential Encoding: No
Samples/Symbol: 4
Excess BW: 350m

Multiply Const
Constant: 500m

Soapy BladeRF Sink
Device arguments: se...3fa0cf
Sample Rate: 5.71429M
Center Freq (Hz): 3G

UHD: USRP Source
Device Address: add...8.1.100
Sync: PC Clock
Samp rate (Sps): 5.85714M
Ch0: Center Freq (Hz): 3G
Ch0: AGC: Default
Ch0: Gain Value: 700m
Ch0: Gain Type: Absolute Power (dBm)
Ch0: Antenna: RX2

Rational Resampler
Interpolation: 1
Decimation: 1
Taps:
Fractional BW: 400m

AGC
Rate: 100u
Reference: 900m
Gain: 1
Max Gain: 65.536k

Symbol Sync
Timing Error Detector: Gardner
Samples per Symbol: 4
Expected TED Gain: 1
Loop Bandwidth: 62.8m
Damping Factor: 1
Maximum Deviation: 1.5
Output Samples/Symbol: 1
Interpolating Resampler: MMSE, 8 tap FIR

Costas Loop
Loop Bandwidth: 62.8m
Order: 2

File Sink
File: pathiq
Unbuffered: Off
Append file: Overwrite

Noise Source
Noise Type: Gaussian
Amplitude: 1
Seed: 0

UHD: USRP Sink
Device Address: add...8.1.150
Sync: Unknown PPS
Samp rate (Sps): 2M
Ch0: Center Freq (Hz): 3.003G
Ch0: Gain Value: 31
Ch0: Antenna: TX/RX

FIGURE 4.7: Experimental Testbed—hardware and software components of the considered measurement setup.

More in detail, an image is represented as a matrix of dimensions $[Y \times J \times 3]$, where each color is one layer, and each pixel value in the range of 0 to 255 is assigned based on the tile value, according to the following scheme:

- If $0 \leq n_T < 255$, then $p_R = 0$, $p_G = 0$, and $p_B = n_T$,
- If $256 \leq n_T < 511$, then $p_R = 0$, $p_G = n_T - 255$, and $p_B = 255$,
- If $n_T > 511$, then $p_R = n_T - 510$, $p_G = 255$, and $p_B = 255$,

where n_T represents the tile value derived from the bivariate histogram, and p_R, p_G, and p_B correspond to the red, green, and blue pixel values, respectively.

To ensure that the value corresponds to a valid pixel intensity in the generated image, if the count exceeds 255 (the maximum allowable pixel value), such values are truncated to 255. It is crucial to adjust the number of IQ samples per image to minimize information loss, particularly from having too many tiles with a sample count exceeding 255.

Multi-class Classification

This phase allows performing RFF by correctly classifying the images generated in the previous step. In accordance with the scenario illustrated in Section 4.3.1, the objective of the proposed multi-class classification problem is to identify the transmitting device in a pool of N transmitters. To achieve this, the collected dataset of I-Q samples is split into three subsets: training, validation, and testing. Moreover, several state-of-the-art CNNs pre-trained on the ImageNet database [142], specifically ResNet-18, are considered. The implementation of these models provided by MATLAB 2024a is utilized, with modifications made to the input and output layers to suit the specific classification task. The input layers are resized to match the dimensions of the images generated from raw IQ samples, while the output layers are restructured to account for the number of classes in the experiments, based on the number of transmitters.

4.3.3 Experimental Measurements and Analysis

Experimental Testbed

Figure 4.7 illustrates the implemented experimental testbed used for the real-world tests. The testbed includes seven SDRs, with five of them working alternatively as transmitters, one

TABLE 4.1: Communication settings parameters.

Parameter	**Value**
Reference Frequencies (Tx-Rx)	1 GHz 2.4 GHz 3 GHz
Communication Bandwidth	2 MHz
Roll-off factor (α)	0.35
Sample per symbol (Sps)	4
Sample rate	5.8 Msps
Jammer Carrier Frequencies	0.995-1.005 GHz 2.395-2.405 GHz 2.995-3.005 GHz
Jamming Bandwidth	2 MHz

working as jammer and one working as the receiver. Specifically, the testbed is composed of the following devices:

i) Five *BladeRF 2.0 micro xA9* devices, used as transmitters, equipped with *LMS6002D* RF transceivers capable of supporting various wireless standards and mobile communication protocols. These devices can transmit wireless signals in the bandwidth $[47 - 6,000]$ MHz, with a gain up to 66 dB.

ii) One *NI Ettus USRP X410*, used as the receiver, providing four independent transmit and receive channels, each supporting up to 400 MHz of instantaneous bandwidth, and covering frequencies from 1 MHz to 7.2 GHz using a two-stage superheterodyne architecture.

iii) One *Ettus Research USRP X310*, employed as a jammer, supporting frequency coverage from DC to 6 GHz, with a maximum baseband bandwidth of 160 MHz.

The receiver is connected to a workstation running Linux Ubuntu 24.04 and equipped with an AMD Ryzen Threadripper PRO 5965WX @3.28 GHz processor and an NVIDIA GeForce RTX 4070 Ti, responsible for running RFF.

The software development toolkit GNU Radio, version 3.10, is used to control the operation of the SDRs and to customize the RF behavior with the required communication settings, as defined in Table 4.1. In this context, the GNU Radio *transmitter chain* used on the Blade RF devices consists of three main blocks: i) a *File Source*, used to generate a (repeating) message made up of a string of 256 bytes with incremental values; ii) a *Constellation Modulator*, featuring a Root Raised Cosine (RRC)-filter, configured for the BPSK modulation scheme; and iii) a *Soapy BladeRF Sink*, which takes complex data as input and streams them to the BladeRF process unit to be then transmitted over the air.

The *receiver* chain exploits the following five main blocks: i) a *UHD Source*, which acquires the I-Q samples from the USRP and streams them for further processing in the signal chain; ii) a *Rational Resampler*, which changes the sample rate of the received I-Q samples; iii) an *AGC*, which dynamically adjusts the gain of the signal to maintain a constant output amplitude, despite fluctuations in the channel; and finally, iv) a *Symbol Sync*, which is utilized to perform clock recovery by synchronizing with the symbols in the digital signal, subsequently decoding the digital signals; and v) a *Costas Loop*, which locks onto the center frequency.

Finally, as for the *jammer*, a *Gaussian Noise Source* generator is connected directly to *UHD Sink* block, configured with a sample rate of 2 Msps, so to obtain an interference signal characterized by a nominal bandwidth of 2 MHz.

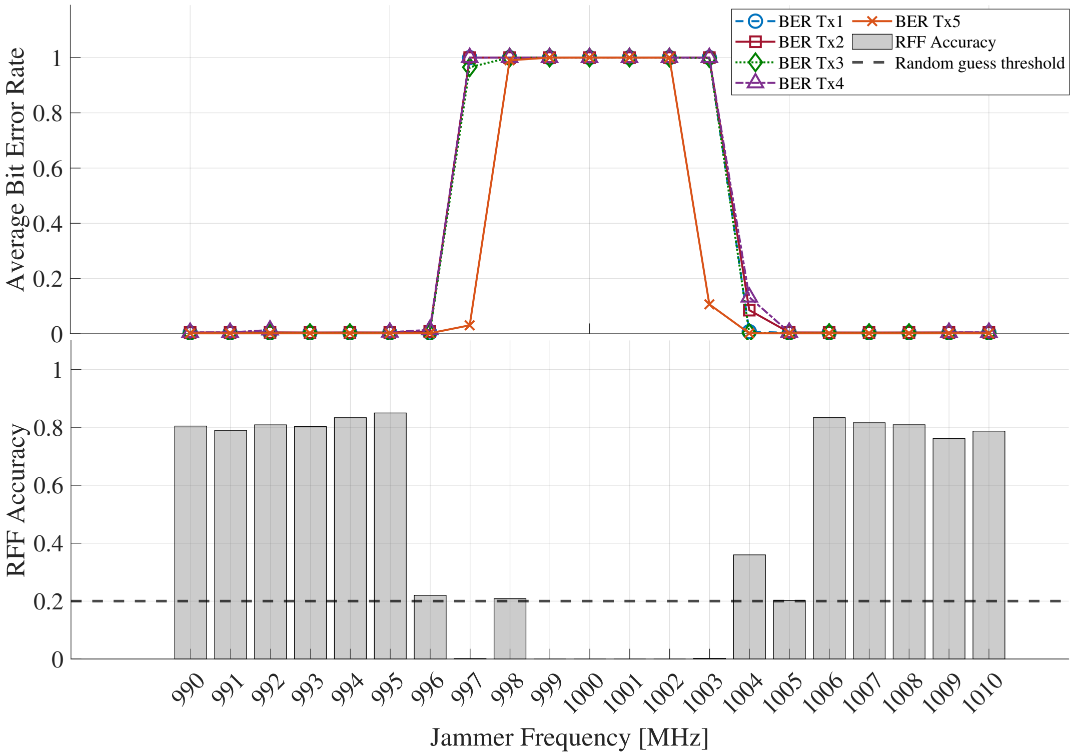


FIGURE 4.8: BER (top) and Accuracy of RFF (bottom) as a function of the jammer frequency, with communication frequency at 1 GHz and jammer sweeping between 990 MHz and 1010 MHz.

During the real-world tests, to minimize the impact of FPGA reload [146] and power cycling on the RFF [122], executed data collection is executed without turning off the receiver and the jammer, while alternatively using the five *BladeRF* as transmitters. Three bandwidths are considered, i.e., $[990, 1010]$ MHz, $[2395, 2405]$ MHz, and $[2995, 3005]$ MHz, where the five signal transmitters are let to emit signals of bandwidth 2 MHz on the carrier frequencies 1000 MHz, 2400 MHz and 3000 MHz, respectively. For each of such carrier frequencies, two distinct experiments are conducted. First, in an interference-free scenario, 10 data acquisition sessions per communication frequency on each transmitter are performed, with each session lasting 25 seconds. Then, the jammer is activated, injecting interference with bandwidth 2 MHz on all frequencies in the bandwidth of interest with a step of 1 MHz, and executed 5 data collection sessions per communication frequency on each transmitter for each jammer carrier frequency.

After data collection, for RFF, images using $K = 10^5$ IQ samples are generated, with image dimensions set to $225 \times 225 \times 3$, aligning with the image size in the ImageNet database in MATLAB 2024a. Then, for the multi-class classification, the dataset is split into 60% for training, 20% for validation, and 20% for testing, and the CNN resnet18 is employed for the classification, in line with recent relevant scientific contributions on RFF [135], [134].

Results

In the following, the results of the investigation are presented by resorting to two metrics, i.e., the average BER of the communication link and the average accuracy of the considered RFF technique. Firstly, the five (5) transmitters are considered in an interference-free environment, so to assess the (best) performance of the considered RFF technique. Table 4.2 shows the average accuracy of the CNN *ResNet-18*, denoting the capability of the RFF system to identify

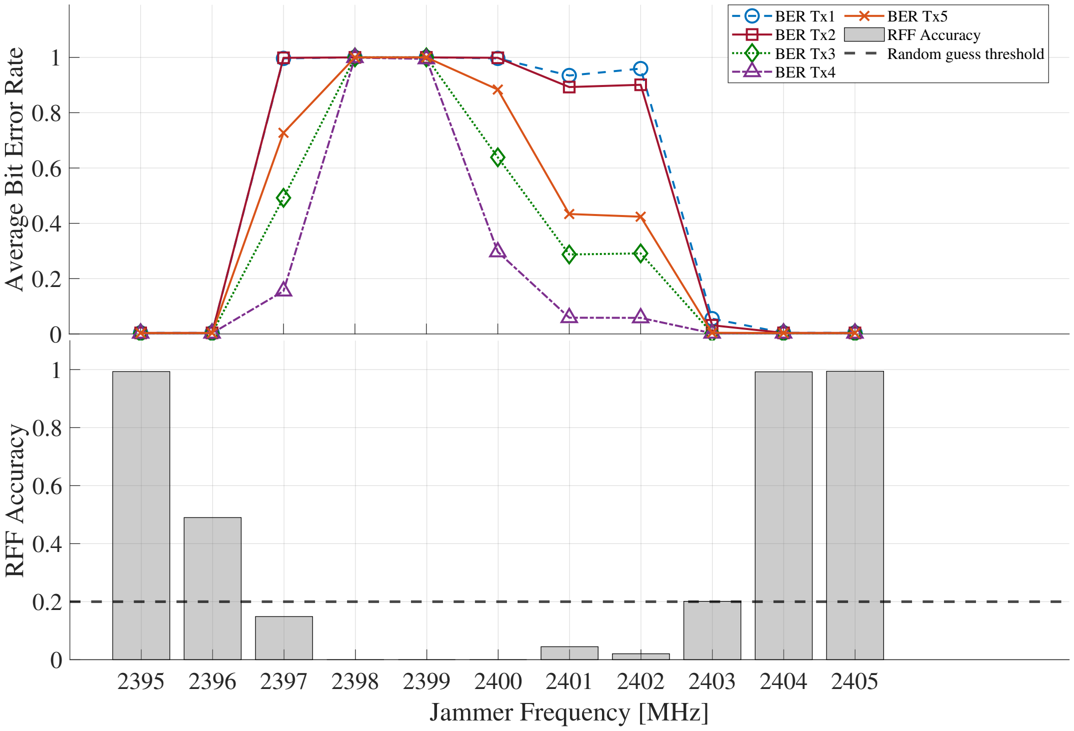


Figure 4.9: BER (top) and Accuracy of RFF (bottom) as a function of the jammer frequency, with communication frequency at 2.4 GHz and jammer sweeping between 2395 MHz and 2405 MHz.

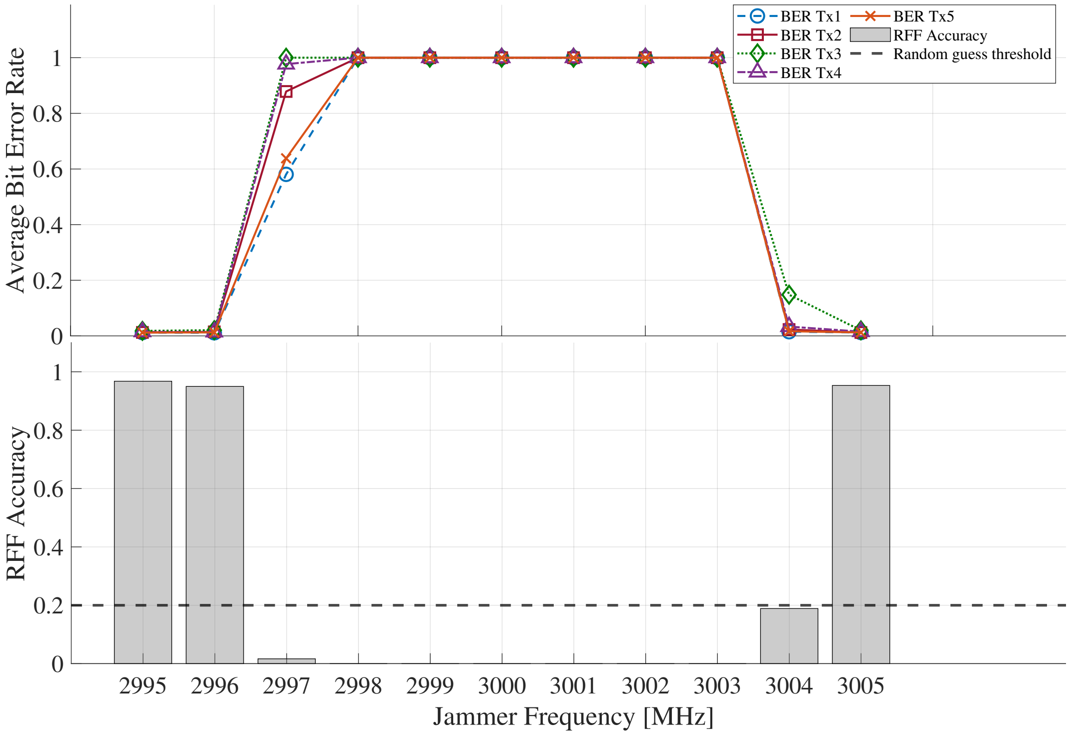


Figure 4.10: BER (top) and Accuracy of RFF (bottom) as a function of the jammer frequency, with communication frequency at 3 GHz and jammer sweeping between 2995 MHz and 3005 MHz.

Table 4.2: Accuracy of RFF in interference-free scenarios.

Reference Frequencies	**RFF Accuracy**
1 GHz	0.9831
2.4 GHz	0.9921
3 GHz	0.9692

each of the 5 radios in the pool when such radios transmit on three different reference frequencies, i.e., 1 GHz, 2.4 GHz, and 3 GHz. It is observable that the accuracy is always higher than 0.96. Thus, the considered RFF technique can detect and identify each of the transmitters in the radio spectrum from PHY-layer data.

Figures 4.8, 4.9, and 4.10 show the results of the analysis for the three reference frequencies considered in this work, i.e., 1 GHz, 2.4 GHz, and 3 GHz, respectively. These results are obtained in each figure by setting the communication frequency between the transmitter and the receiver to the reference frequency, and then by sweeping the frequency of the jammer as indicated in the x-axis of the figures. Firstly, can be seen that a Gaussian noise jammer featuring a baseband of 2 MHz affects the quality of the communication link (on average) in the range $[-3, +3]$ MHz with respect to the reference frequency. Thus, the modulated noise signal is characterized by a passband of about 6 MHz —this phenomenon being the result of the side lobes of the modulated noise signal.

In such an area, the BER is equal to 1, and communication between the transmitter and the receiver is prevented. Outside that frequency range, when BER is equal to zero, the accuracy of the RFF is always between 0.8 and 1. This represents this work ground truth and confirms that the RFF system works effectively.

It is important to notice that the actual behavior of the interference in the RF domain (bandwidth amplitude being equal to 6 MHz) is out of the scope of this work, while focusing on the analysis of the edges. Indeed, the most interesting phenomenon occurs at the edge of the range previously discussed when the BER changes from 0 to 1 and vice versa. Notably, there are frequencies where both the BER and the RFF accuracy are low. A few examples are 996 MHz and 1004 MHz (1 GHz, Fig. 4.8), 2396 MHz and 2403 MHz (2.4 GHz, Fig. 4.9), and finally, 3004 MHz (3 GHz, Fig. 4.10). At such frequencies, the high quality of the link (low BER) is not a sufficient condition to justify the performance of the RFF. Therefore, through these results, is possible to claim that low BER is a necessary but not sufficient condition to perform successful RFF.

Chapter 5

Design and implementation of a Looking-Forward Lawful Interception Architecture for Future Mobile Communication Systems

The European Union (EU) has witnessed a significant increment of criminal networks involved in cybercrime, terrorism-related offenses, and outlawed trades [31]. The most recent report on police-recorded offenses within the EU presents statistical insights spanning the years from 2016 to 2021 [147]. It encompasses various criminal activities across EU member states, defining occurrences such as acts against computer systems with approximately 110k cybercrime events recorded in 2021, participation in organized criminal activities, reflecting around 7.5k registered activities during the same year, and unlawful acts involving controlled drugs or preceding, accounting for over 1150k events in 2021. Moreover, it emerges an increasing level of participation by EU member states in these initiatives. For instance, the number of cybercrimes doubled across major European countries between 2018 and 2021. Therefore, LEAs are seeking innovative and efficient LI tools that are compatible with the evolving 5G and Beyond 5G network architectures and are capable of preventing, detecting, and investigating criminal and terrorist activities.

In contrast to the conventional technologies, the 5G and Beyond 5G networks provide incomparable data rates, high channel capacity, and low latency by introducing a highly dynamic and distributed architecture with the use of emerging technologies such as SDN, NFV, network slicing, and Edge Computing [148]. To date, this technology integration is required to cope massive increase of data generated by IoT devices and applications where the majority of the data may be encrypted or in plaintext. These emerging technologies enable efficient resource allocation and on-demand network customization, making it challenging to identify precise interception points and employ advanced analytic tools for real-time interception, processing, and analysis of data within the future network infrastructure [32].

Furthermore, 5G networks use new security protocols such as enhanced encryption and random mobile identifiers [148]. Therefore, if in the past radio monitoring techniques (e.g., IMSI-catchers [149]) were used to intercept network identifiers, in the new 5G Core Network (5GCN) it is no longer feasible since the International Mobile Subscriber Identity (IMSI) is transmitted in a concealed form to protect their privacy [150]. Moreover, new-generation mobile systems are increasingly dependent on IM and VoIP platforms (e.g., Telegram and WhatsApp), allowing, by the privacy-by-design paradigm, real-time communication and secure sharing of private information through the usage of end-to-end encryption [27]. It is a secure communication mechanism that permits only the parties involved to correctly send and receive messages since the encryption keys are only accessible to each participant and not to the service provider [151] [17].

Nevertheless, while this represents a significant achievement in communication security, it makes conventional LI techniques, based on existing 3GPP specifications, largely ineffective [152]. Despite this, the LEAs can still intercept communication flows but the encrypted data remains fully unintelligible [153]. This introduces significant challenges for the advancement of LI methodologies, thus requiring the design and the investigation of novel technical solutions [32]. It is important to note that this challenge has gained attention from the European Commission, researchers, and security specialists [32], [152], and [153].

Recent position papers such as [154] and [155], highlight the importance of addressing the management of LI in Beyond 5G and 6G systems and standardizing legal requirements [156]. However, it should be noted that their primary goal is not to provide any original or effective methodology for solving this problem. Meanwhile, a machine learning-based LI architecture, as described in [157], has been designed to analyze and classify audio and video content. Nonetheless, it does not offer the possibility to decrypt the multimedia flows, as well as to deliver them to LEAs. As a result, the usage of end-to-end encryption in an ever larger amount of applications highlights the importance of introducing more sophisticated techniques supporting effective LI features.

To bridge this fundamental gap, the work* presented herein provides the following main scientific contributions:

- This work present a novel LI framework offering new interception capabilities on top of the existing 3GPP standardized architecture. The proposed LI framework leverages a secure configuration and usage of an inspection-friendly end-to-end cryptography scheme (i.e., Key Escrow algorithm) at the application layer and allows authorized LEAs to decipher end-to-end encrypted data intercepted (via conventional LI procedures) in the core network. Here, data privacy is guaranteed against the mobile operator, which is still unable to guess intercepted contents because encrypted. Moreover, the security proof study demonstrates the ability of the proposed LI framework to resist two adversarial scenarios.

- A proof-of-concept implementation of the proposed LI framework is presented, based on the Linux-based Docker containers, emulating a 5G network via Open5Gs and UERANSIM environments. Herein, the OpenLI software is used to ensure the standard-compliant LI implementation by employing four containers representing the entities of the LI framework. The implementation, using Python scripts and its cryptographic libraries, demonstrates functionalities such as end-to-end encrypted data exchange, data interception, and decryption through Key Escrow mechanisms at the application layer.

- The performances of the proposed approach are evaluated by considering two different use cases: an end-to-end data exchange (i.e., encrypted end-to-end file exchange) and a cloud-based deployment (i.e., VoIP service). The obtained results validate the effectiveness and reveal the real-time-like latency performances and scalability of the proposed LI framework.

5.1 Background and Motivation

This Section explains the technicalities of the standardized 3GPP LI architecture, presents background concepts on End-to-End encryption techniques, and describes the state of the art on Key Escrow schemes.

*This work represents a substantial extension of a preliminary contribution previously presented by the same author in a recent conference paper [158].

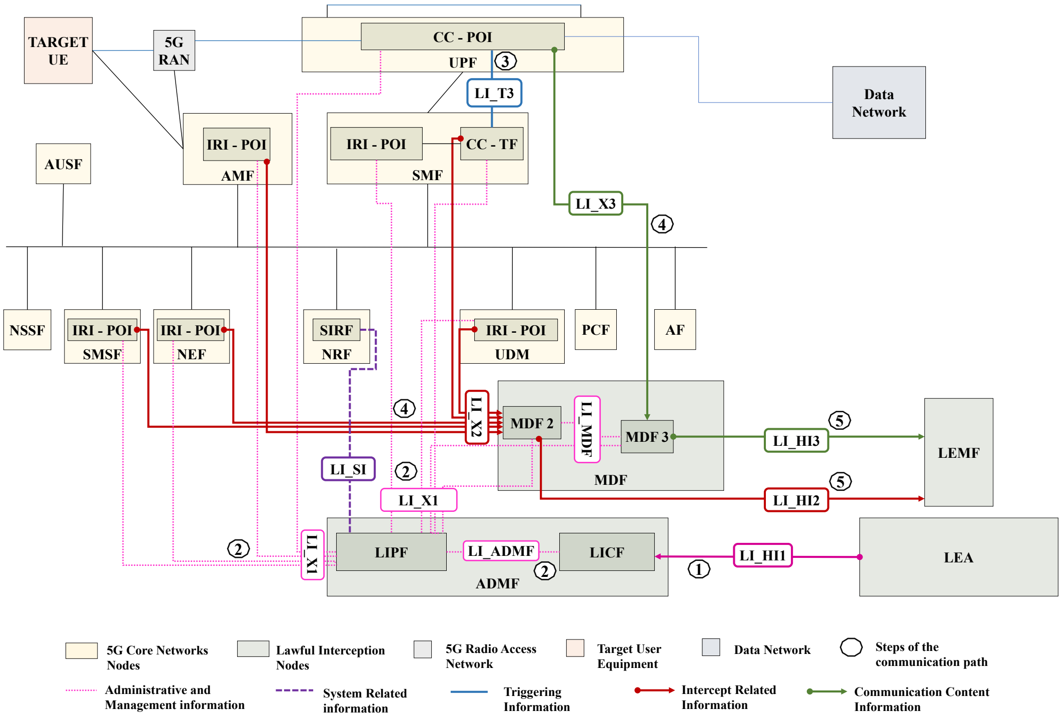


FIGURE 5.1: 5G 3GPP Lawful Interception architecture.

5.1.1 Lawful Interception

The LI refers to the technological methods employed by Communications Service Providers (CSPs) to collect, retain, and transmit communication data to law enforcement databases [159]. The 3GPP Technical Specifications on LI provides: i) LI requirements in TS 33.126 [159], ii) LI architecture and functions in TS 33.127 [160], and iii) LI protocol and procedures in TS 33.128 [161].

Fig. 5.1 illustrates a high-level description of the 5G 3GPP LI architecture, highlighting LI nodes and interfaces. Within this illustration, the communication paths proceed through five main steps, which are as follows:

- ***Step 1.*** Given a targeted User Equipment (UE) which needs to be intercepted, the LEA submits a valid warrant to the CSP through the LI_HI1 interface, which in turn starts all the required standard procedures [159].

- ***Step 2.*** Herein, in accordance with [160], the Administration Function (ADMF), by exploiting the LI_ADMF and all LI_X1 interfaces are responsible for the administrative and management functions of the LI capability within the CSP. These functions encompass the provisioning, modification, and deactivation of Point of Interception (POI), Triggering Function (TF), and Mediation and Delivery Functions (MDFs).

  Specifically, the ADMF comprises two main logical sub-functions, communicating via the LI_ADMF interface.
  First, the Lawful Interception Control Function (LICF) manages the entire life cycle of a warrant while acting as the central repository for all sensitive information and LI configuration data. Additionally, it holds the ultimate responsibility for all decisions made within the LI system.
  Second, the Lawful Interception Provisioning Function (LIPF) serves as a secure intermediary that enables the LICF to interact with the LI modules that are necessary for

the CSP network to function. Indeed, it is in charge of interacting with the System Information Retrieval Function (SIRF), which gives interface system-related information via the LI_SI, so that the latter may carry out the steps required to set up and sustain interception of the target service.

Indeed, the LIPF performs a passive function during this step. By routing LI_X1 communications from and to the LICF, or an active function by receiving triggering information and passing the trigger to the relevant POI.

- ***Step 3.*** This step begins when the relevant POI, located in the User Plane Function (UPF), is triggered via the LI_T3 interface enabling it to i) detect the target communication, ii) extract Intercept Related Information (IRI) or Communication Content (CC) from the target, and iii) deliver the output to the MDF [161].

- ***Step 4.*** In this step, the architecture provides multiple POIs distinguished into two groups based on the type of information they transmit to the MDF, which comprises two modules (i.e., MDF2 and MDF3). Therefore, the IRI-POI delivers IRI information through the LI_X2 interface to the MDF2, while the CC-POI delivers CC data over the LI_X3 to the MDF3.

- ***Step 5.*** At this point, the MDF generates the IRI and CC messages from the MDF2 and MDF3 and delivers them, via the LI_HI2 and LI_HI3 interfaces, respectively, to the Law Enforcement Monitoring Facility (LEMF) [161]. Finally, the LEA easily accesses the intercepted traffic.

5.1.2 End-to-End Encryption

As previously anticipated in the Section 2.1, one of the most relevant challenges regards *encryption and privacy*. The widespread implementation of encryption mechanisms in 5G and Beyond 5G networks poses substantial obstacles to the LI process. People increasingly depend on applications such as Skype, Zoom, Telegram, WhatsApp, or similar applications, which generate extensive multimedia content, including text, voice, and video. Consequently, the demand for robust sensitive data protection systems has risen to protect this vast multimedia content. One way to achieve this is to put in place an end-to-end encryption system that doesn't rely on any online services or centralized infrastructure. Indeed, more VoIP and IM applications claim to support end-to-end encryption which guarantees that only the sender and the intended receiver can decipher the contents of a message [27].

In the realm of secure online communication systems and private chat applications, the off-the-record (OTR) protocol emerged to facilitate end-to-end encryption [28]. Despite being integrated as a plugin for widely used IM clients such as Pidgin, its limited adoption can be attributed to usability issues [29]. Increased consciousness of privacy concerns emerged after the Snowden revelations. As a result, new encrypted messaging systems have evolved to address end-to-end encryption problems by expanding and adopting the OTR protocol [27]. To provide both end-to-end encryption and advanced security features, such as forward secrecy and future secrecy, Open Whisper Systems introduced Signal, a groundbreaking end-to-end encryption protocol that supports both synchronous and asynchronous communication settings. The Signal protocol requires a key-distribution server to maintain user identities and ephemeral keys, as it functions in synchronous and asynchronous messaging situations [30].

Currently, the majority of end-to-end encryption applications either employ the Signal protocol (e.g., Signal and WhatsApp) or use Signal-like proprietary protocols (e.g., Telegram and Zoom) [27]. For example, the WhatsApp end-to-end encryption requires a user to initiate a voice or video connection by creating encrypted sessions with each of the receiver devices, and after the call is initiated, Secure Real-time Transport Protocol (SRTP) is used to protect it by using master secret keys created for each receiver device [162]. Whereas, the Telegram end-to-end encryption functionality is implemented in one-to-one chats and calls using its proprietary protocol, known as the MTProto protocol. In this protocol the cryptographic keys are exchanged via the Diffie-Hellman protocol and the participating devices exchange these keys after establishing a Secret Chat [163].

Even if the adoption of end-to-end encryption significantly enhances the security and privacy of communications, it simultaneously renders the interception of the communication more challenging [17]. In case of end-to-end encryption, in fact, the intercepted traffic can be interpreted by LEAs just as a string of bits with limited information. In this context, effectively managing the trade-off between privacy or security and the requirements of authorized interception becomes crucial. Therefore, the mitigation of these challenges demands the development of resilient decryption capabilities and the establishment of collaborative frameworks between telecommunication service providers and LEA.

5.1.3 Key Escrow

Generally speaking, Key Escrow represents a technique that helps in recovering the secret key used for application encryption and, when specific criteria are met, assists the authorized entities (e.g., the LEA in the considered case) in decrypting the ciphertext [164].

The Clipper Chip was proposed by the United States government in the 1990s as an initial effort to build a key escrow system [165]. Herein, the Skipjack symmetric encryption was employed and the encryption keys were partitioned into distinct components and securely entrusted to various government entities [166]. Nonetheless, the Clipper Chip faced intense criticism and censure owing to concerns about its susceptibility to security flaws and the inherent hazards of unauthorized access to the escrowed keys, weakening its usefulness and public trust [167]. To overcome the previous issues and strike a balance between enabling lawful interception and mitigating the potential for unauthorized access, an alternative strategy for Key Escrow entails the engagement of a trusted third-party entity responsible for preserving the decryption keys on behalf of users [168]. To facilitate this form of Key Escrow, various protocols have been suggested, including the ones in [164], [169], and [170], which aim to provide the necessary framework for effective implementation and management.

Apart from these contributions, the adoption of Key Escrow techniques in the context of LI has not received the deserved attention in recent years. The main reason refers to the native design principle of the related interception approach: the introduction of the General Data Protection Regulation (GDPR) led to the prohibition of previous Key Escrow schemes that operated on SIM private keys, as users were unaware of the voluntary backdoors [171]. Remarkably, to the best of the authors' knowledge, there has been no attempt to employ a Key Escrow system at the application level for LI purposes.

In contrast, it is noteworthy to notice that Key Escrow schemes, applied to end-to-end cryptography, may achieve a compromise between the need for individual privacy and the lawful requirements of government agencies to conduct surveillance or interception activities for criminal investigations [167].

Indeed, it underscores the need for further research and development in addressing the evolving landscape of privacy regulations and technological advancements and offering valuable insights into the potential integration of Key Escrow mechanisms within application frameworks for enhanced LI capabilities.

5.2 The proposed methodology

This Section aims to propose a feasible technical solution that fosters further discussions on defining inspection-friendly end-to-end encryption schemes to address the LI challenges. It introduces a novel LI framework that enhances interception capabilities by adding new features on top of the conventional 3GPP standardized architecture. This enhancement is achieved through the secure configuration and utilization of a Key Escrow cryptographic scheme at the application layer, thereby enabling LEAs to decrypt end-to-end encrypted data intercepted in the core network.

5.2.1 Design Principles

The proposed LI framework, illustrated in Fig. 5.2 has been designed starting from the following two main hypotheses.

1) Standard-compliant hypothesis. In accordance with the guidelines provided by 3GPP [160], the mobile network infrastructure consists of the Next Generation Node B (gnB) that facilitates wireless connectivity for UEs through the 5G New Radio (5GNR) interface, as well as the 5GCN. The mobile network operator possesses control over the network infrastructure, enabling the end-user's identification through the International Mobile Subscriber Identity (IMSI) and determining the corresponding UPF within the 5GCN. Notably, the UPF also serves as the hosting entity for the POI, which owns the capability to intercept specific communications as elaborated upon below.

2) High level definition of the proposed LI framework. The traffic generated or received by the UE results in encrypted application data, meaning that the intercepted CC potentially comprises a series of encrypted data. Thus, the technical approach assumes that the end-to-end application traffic is secured through a Key Escrow system, which assists authorized entities such as the LEAs in decrypting the ciphertext [168].

Indeed, the conceived LI framework is defined on top of the conventional 3GPP scheme reported in Fig. 5.1. Specifically, it does not require new 3GPP entities and it considers the following four main entities:

- **Subscribers.** Two users (i.e., UE A and UE B) enabling secure communication via end-to-end encryption.
- **Law Enforcement Agency (LEA).** An authorized enforcement entity that, under the legislation, requests the content of the communication and gets IRI and CC from the CSP.
- **Authentication Server Function (AUSF).** A responsible component of the 5GCN for subscribers' identity verification. In response to the LEA request, it sends to the LEA the appropriate decryption material and interception-related information and provides the application encryption material to the subscribers.
- **Trusted Key Authority (TKA).** A fully trusted third party that requests and provides encryption keys for communication sessions to the AUSF, LEA, and subscribers.

Without loss of generality, the proposed solution leverages the Key Escrow algorithm presented in [168], constructed on an ID-based Cryptosystem (IDBC), and investigates, for the first time, its adoption on LI tasks within Beyond 5G systems. Specifically, the proposed LI framework builds upon the conventional 3GPP LI architecture [160] by introducing a new entity, namely TKA. This entity, acting as a fully trusted third party with the same degree of trustworthiness as a Certification Authority (CA), assumes system configuration responsibility by addressing both application security setups and application session key material

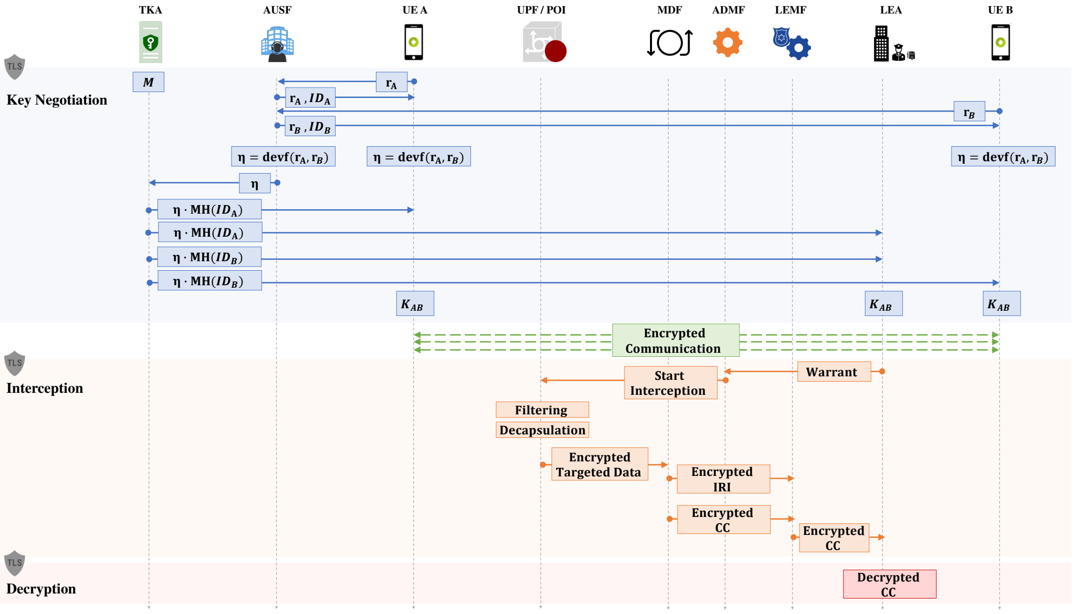


FIGURE 5.2: Technical workflow.

generation. It is necessary to emphasize that the TKA only retains its secret master key and does not store the keypairs of any registered user.

Moreover, it is important to highlight that the conceived LI framework requires an end-to-end encrypted application provider to agree on the specifically defined algorithm as a key-exchange solution.

To ensure clarity, the conceived LI framework outlines above the standardized 3GPP LI architecture [160], which independently manages the aspects related to mobility behavior. In particular, in line with the standard [160], each piece of mobility information (e.g., location and cell IDs) is stored within the IRI content. Firstly, when a target UE connects to the 5G network in the registration procedure, the IRI-POI in the Access and Mobility Management Function (AMF) creates the registration xIRI, which contains information on the registration mobility update. Subsequently, the location update xIRI is produced each time the IRI-POI in the AMF determines that the targeted UE's location has changed due to UE mobility or when the AMF sees target UE location data while performing a service operation. Furthermore, if the information in the AMF includes one or more cell IDs, all of them must be transmitted to the LEMF whenever location reporting is activated at the AMF.

5.2.2 Technical Details

This section better describes the conceived LI framework as depicted in the Fig. 5.2. To ensure clear understanding, it is important to note that the interaction between the TKA and 5GCN or 5GNR nodes is protected via the Transport Layer Security (TLS) protocol.

Moreover, based on the two premises introduced in the Section 5.2.1, system configuration and interception operations can be described through three main phases: *key negotiation*, *interception*, and *decryption*.

1. ***Key Negotiation Phase***: This phase relies on the involvement of both the mobile operator and TKA. In this context, the formulated LI framework performs most of the application-level cryptographic operations by introducing hash functions, bilinear pairings, and derivative functions denoted as $\mathcal{H}(\cdot)$, $e(\cdot)$, and η respectively. The TKA owns a master secret key and computes the UEs public/private key pairs based on their

unique identity. Moreover, based on the Key Escrow algorithm presented in [168], since pre-shared keys are distributed between the AUSF and two UEs, the AUSF securely transmits and receives nonces to the UEs. Thus, the two UEs are equipped to independently compute the derivation functions, obtain the application session key, and protect the communication using end-to-end encryption at the application layer (i.e., $k_{AB} = e(\eta \cdot M\mathcal{H}(ID_A), \mathcal{H}(ID_B))$ and $k_{BA} = e(\mathcal{H}(ID_A), \eta \cdot M\mathcal{H}(ID_B)))$. At the same time, the AUSF calculates the derivation function and shares it with the TKA, which, in turn, forwards it to the authorized LEA. Thus, the LEA owns the cryptographic material for deriving the same application session key while ensuring that users remain unaware of lawful interception activity. Fig. 5.2 presents the detailed cryptographic operations. Please refer to the Appendix C and to [168] for in-depth details about the summarized cryptographic scheme.

2. ***Interception Phase*:** During this phase, the LEA issues a valid interception warrant to the ADMF. The ADMF validates the warrant and subsequently grants permission to commence the interception procedure. Herein, the POI decapsulates and filters the targeted GTP data. Specifically, after the ADMF validates the warrant, the POI investigates the traffic passing through the UPF. In line with the *Standard-compliant hypothesis*, the packets containing the data exchanged between the two UEs in the end-to-end encrypted communication are encapsulated according to the following protocol structure: IP over GTP over TCP over IP. Indeed, the POI performs a decapsulation operation on each packet to obtain the raw end-to-encrypted application data. Subsequently, the LEMF entity collaborates with the MDF to collect precise information about the targeted communication from POI, including IRI and encrypted CC.

3. **Decryption Phase:** During this phase, the LEA gains the ability to decrypt the previously received encrypted CC. This decryption process involves the utilization of the application session key, denoted as $k_{AB} = e(\eta \cdot M\mathcal{H}(ID_A), \mathcal{H}(ID_B))$.

To ensure clarity, the conceived LI framework is established on the existing 3GPP architecture and does not modify the established 5G security protocol. The above-described cryptographic procedures are meant to be done at the application layer and do not involve the UE SIM, which has limited computing capabilities. Moreover, it is important to remark that since the application session key is derived from random numbers exchanged in every new negotiated communication, the LEA cannot reuse the same key material for intercepting other communication sessions.

5.2.3 Security proof and threat analysis

This Section provides the security proof for the designed LI framework, considering the security requirements related to the communication protocols and cryptographic techniques. It is important to note that [168] examines the security proof of the selected Key Escrow technique in terms of cryptographic operations. Concerning the protocol security analysis for the proposed framework, the developed LI framework incorporates a widely recognized security building block known as TLS. Its security has been previously established and fully defined in the reference contributions given below, and it remains independently created. As a result, the following proves the security of each employed security requirement:

- **Secure End-to-End Communication:** Through the use of the TLS (i.e., TLS version 1.3) protocol, the proposed LI framework ensures the establishment of a secure end-to-end channel communication between each non-5G entity and 5G standardized network architecture. In particular, it facilitates mutual authentication and data secrecy. Furthermore, it allows the communication network to be resistant against Man-in-the-Middle

(MITM) attacks. Since TLS is a well-known and widely used security protocol, [102], [103], and [104] have already investigated its security proof.

- **Subscriber Non-Engagement:** This security requirement ensures that subscribers cannot determine whether their communication is being monitored since they do not take part in key escrowing which mainly involves the TKA and the AUSF. The work in [168] provides formal proof of it.

- **Warrant Validity:** This security requirement relates to the failure of an interception in an unauthorized session and to the prevention of a replay attack. Specifically, as detailed in Appendix C, the LEA receives from the TKA the cryptographic material (i.e., τ_2) for calculating the session key after verifying the previously submitted specific warrant. Moreover, since the Key Escrow algorithm selects nonces r_A and r_B randomly in each session and the application session key is derived from a function involving these nonces (i.e., $\eta = devf(r_A, r_B)$), each application session will introduce a different session key not managed from the submitted warrant, as demonstrated by [168].

- **Key Escrow Effectiveness:** The generic PKI-based Key Escrow models need the TKA to store a large number of public key pairs, while the proposed LI framework only requires the storage of the master key, which is always kept secure and never delivered, in line with [168].

Moreover, to ensure the compliance with LI specifications and standards (i.e., [159], [160], and [161]) which allow only authorized LEA with a valid warrant to intercept the communication, this Section aims at studying the security of the proposed LI framework under two attack scenarios involving the presence of a malicious user (i.e., unauthorized LEA) trying to eavesdrop the secure communication, in line with [168].

Adversarial scenario 1: absence of a valid warrant. Let UE A and UE B be the subscribers willing to initiate an end-to-end encrypted communication and let the eavesdropper E be the malicious user trying to intercept the encrypted communication. The procedure pursues the following steps in line with the algorithm described in Appendix C:

1. The UE A sends the chosen random number r_A and its signature $sign_A(r_A)$ to the AUSF.

2. The AUSF verifies r_A and the signature $sign_A(r_A)$ of UE A and forwards them to the UE B.

3. UE B, in turn, verifies the UE A signature, generates its random number r_B, and delivers it together with its signature $sign_B(r_B)$ to the AUSF.

4. The AUSF proves r_B and the signature $sign_B(r_B)$ of UE B and forwards them to the UE A.

5. The two subscribers now compute their application session key $k_{AB} = e(devf(r_A, r_B) \cdot P_A, p_B)$ and $k_{BA} = e(p_A, devf(r_A, r_B) \cdot P_B)$, respectively, and start their end-to-end encrypted communication (please refer to Appendix C for detailed description.).

6. The eavesdropper E tries to intercept the communication, but it does not have any related cryptographic material from which to retrieve the application session key because it didn't present any warrant to let the TKA generate and forward it. Specifically, it does not have any information about the derivation function and fails to compute any correct application session key for decrypting the communication between UE A and UE B.

Adversarial scenario 2: expired session. Let UE A and UE B be the subscribers that already had an end-to-end encrypted session correctly intercepted by an authorized LEA, and let the eavesdropper E be the malicious user capturing the application session key used for the above communication session (i.e., $k_{AB} = e(devf(r_A, r_B) \cdot P_A, p_B)$). Assuming now that UE A and UE B start a new communication session, they compute the new application session key as follows:

1. The UE A forwards the chosen random number r'_A and its signature $sign_A(r'_A)$ to the AUSF.

2. The AUSF verifies r'_A and the signature $sign_A(r'_A)$ of UE A and sends them to the UE B.

3. UE B, in turn, proves the UE A signature and generates its random number r'_B and delivers it together with its signature $sign_B(r'_B)$ to the AUSF.

4. The AUSF verifies r'_B and the signature $sign_B(r'_B)$ of UE B and onwards them to the UE A.

5. The two subscribers calculate their application session key $k'_{AB} = e(devf(r'_A, r'_B) \cdot P_A, p_B)$ and $k'_{BA} = e(p_A, devf(r'_A, r'_B) \cdot P_B)$, respectively, and start their end-to-end encrypted communication (please refer to Appendix C for detailed description.).

6. The eavesdropper E attempts to capture the communication, but it has the wrong application session key derived from r_A and r_B, which is different from the newly defined one and fails to decrypt the communication between UE A and UE B. Specifically, even having a previous session key and considering $P_A = M\mathcal{H}(ID_A)$, retrieving the value of $devf(r'_A, r'_B)$ from $devf(r'_A, r'_B) \cdot M\mathcal{H}(ID_A)$ is not possible due to the computational infeasibility of the Elliptic Curve Discrete Logarithm Problem (ECDLP), as proved in [172].

5.3 Proof-of-Concept Implementation

This Section presents a proof-of-concept of the proposed LI framework which is implemented for end-to-end data exchanges (i.e., encrypted end-to-end file exchange) and cloud-based deployments (i.e., VoIP services) to prove the effectiveness of the proposed solution. Precisely, the implemented LI framework offers the opportunity to achieve the following functionalities:

- Enabling two UE devices to exchange end-to-end encrypted data across the 5GCN
- Allowing the LEAs to intercept and access downlink end-to-end encrypted data.
- Facilitating decryption of the application intercepted data through Key Escrow mechanisms.

The testbed is deployed on a workstation with an Intel(R) Core(TM) i5-9400 CPU @ 2.90GHz processor and 16 GB of RAM. It hosts Linux-based Docker containers (i.e., Ubuntu 20.04), with a dedicated container for each 5G entity network. For emulating the 5GNR and the communication between UEs and the gnB, *UERANSIM* is installed. Simultaneously, *Open5gs* is configured to emulate the 5GCN. The *OpenLI* framework is deployed into a Docker-based environment to ensure a standard-compliant LI implementation. More specifically, four containers such as Provisioner, Collector, Mediator, and Agency are used to emulate the ADMF, POI, MDF, and LEMF.

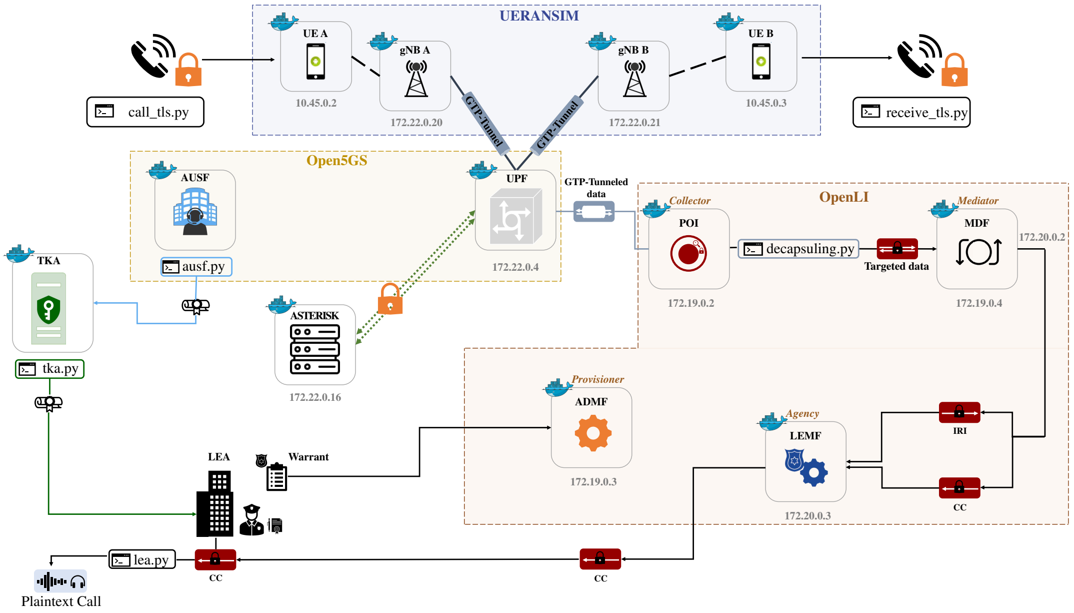


FIGURE 5.3: VoIP services implementation setup.

To effectively meet the requirements of each involved node, this work designs and configures the network architectures illustrated in Fig. 5.3 and Fig. 5.4. This process includes executing individual environments, assigning dedicated network interfaces, and establishing their interactions. To enhance clarity, the Python scripts, by using the libraries listed in the Table 5.1, implement the main functionalities of each participating entity as well as the LI framework cryptographic operations.

TABLE 5.1: List of software and tools.

5G network and Lawful Interception	**Software**
Access Network	UERANSIM
5G Core Network	Open5gs
Lawful Interception	OpenLI
End-to-end communication	Netcat
VoIP services	Asterisk server and PJSIP library
Cryptographic Operation	**Adopted libraries**
Hash function	Hashlib, libnum
Key derivation function	PyCryptodome
Encryption, decryption	PyCryptodome
Bilinear paring	Tate_bilinear_pairing
Post-processing step	**Software**
Interception	OpenLI and libtrace
Packet decapsulation	Scapy
Reassembly	TCPReassembly

Upon the establishment of the 5G network, the testing process begins with the exchange of cryptographic material between the AUSF and TKA using the `ausf.py` script. Subsequently, the `tka.py` script forwards this cryptographic material to the LEA for the session key computation. Once the key-negotiation phase is completed, the first deployment of the proposed framework involves the implementation of *VoIP services implementation* using the Asterisk

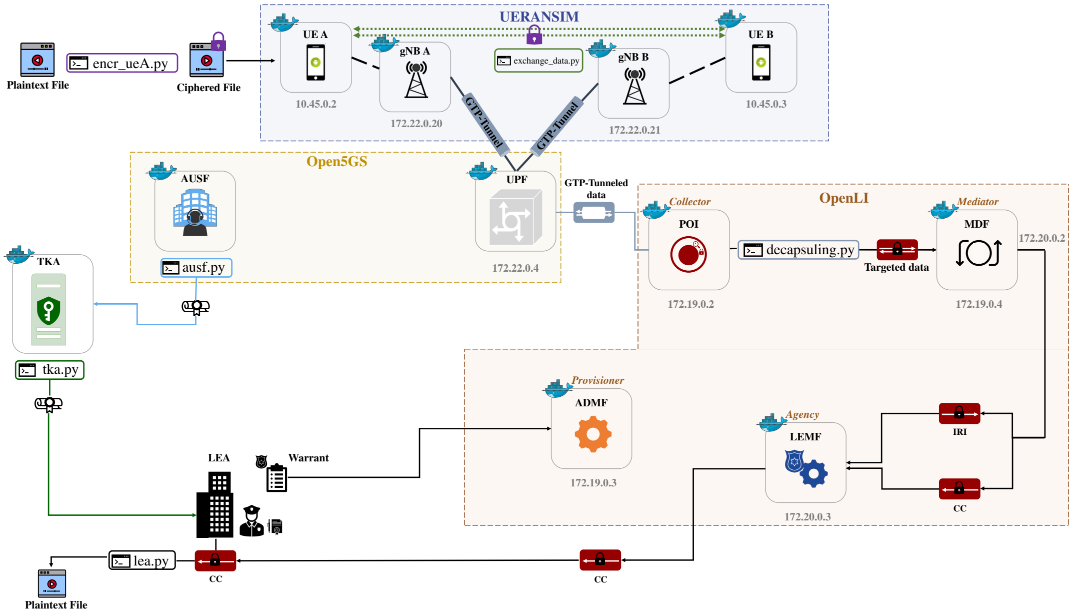
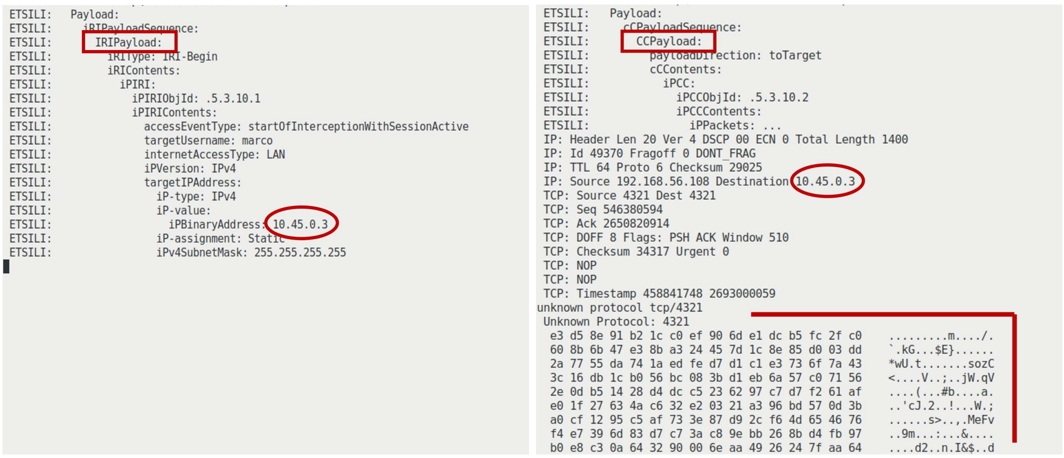


Figure 5.4: End-to-end file exchange implementation setup.

server, as depicted in the Fig. 5.3. Without loss of generality, the implemented proof-of-concept leverages on a key-exchange solution agreement between the VoIP provider and the designed LI framework. Specifically, the two users equipment (i.e., UE A and UE B) are registered through the `pjsip` library by starting a TLS session into the server to make or receive VoIP calls. In this way, the VoIP call will be encrypted using *SRTP/Session Description Protocol Security Descriptions (SDES)* as a key-exchange solution. After the TLS handshake, encrypted Session Initiation Protocol (SIP) messages traverse the network while the SRTP stream is encrypted by the algorithm selected during the SRTP/SDES key exchange system. Consequently, the UE A, through the `call_tls.py` script, utilizes the obtained session key to encrypt and authenticate the initial SRTP stream. Meanwhile, the UE B, employing the `receive_tls.py` script, can respond or terminate the incoming call.

Alternatively, a second scenario, illustrated in the Fig. 5.4, involves the implementation of encrypted end-to-end file exchange. Herein, the first UE encrypts a file containing plaintext media content using the `encr_ueA.py` script and obtains the ciphertext file. The latter is then forwarded to the second UE using the `exchange_data.py` script, employing the netcat package.

Meanwhile, the interception phase starts when the LEA sends a warrant containing all the interception requirements for the downlink interception. In detail, `HTTP` requests with `JSON` files are sent to the Provisioner via the `REST API`. The main `JSON` file defines some of the warrant characteristics (e.g., *LEA ID, LEA IP and ports, interception ID, targeted UE IP, targeted UE mobile operator, and session ID*). Thus, the Provisioner can accept the interception request and activate the Collector to start the interception by letting it access the above-described `JSON` file. During the downlink phase, GTP-encapsulated SRTP data and encrypted data traverse the 5GCN in the first and second deployments, respectively. Herein, using the `decapsulating.py` script, which utilizes `Scapy` library, the collector filters and decapsulates the GTP traffic and obtains the encrypted TCP payload. Categorically, it performs a decapsulation operation on each packet to read the destination IP address of the GTP payload. If there is a match between the IP address of the analyzed GTP payload and the target IP address specified in the warrant. Moreover, the captured GTP payload is transmitted to the Collector by adding an appropriate Ethernet 802.3 header. Thus, the Collector captures the whole traffic, and by using *OpenLI* services it identifies the encrypted target data and forwards



```
ETSILI:   Payload:
ETSILI:     iRIPayloadSequence:
ETSILI:       IRIPayload:
ETSILI:         iRIType: IRI-Begin
ETSILI:         iRIContents:
ETSILI:           iPIRI:
ETSILI:             iPIRIObjId: .5.3.10.1
ETSILI:             iPIRIContents:
ETSILI:               accessEventType: startOfInterceptionWithSessionActive
ETSILI:               targetUsername: marco
ETSILI:               internetAccessType: LAN
ETSILI:               iPVersion: IPv4
ETSILI:               targetIPAddress:
ETSILI:                 iP-type: IPv4
ETSILI:                 iP-value:
ETSILI:                   iPBinaryAddress: 10.45.0.3
ETSILI:                 iP-assignment: Static
ETSILI:                 iPv4SubnetMask: 255.255.255.255
```

(A) IRI.

```
 ETSILI:   Payload:
 ETSILI:     cCPayloadSequence:
 ETSILI:       CCPayload:
 ETSILI:         payloadDirection: toTarget
 ETSILI:         cCContents:
 ETSILI:           iPCC:
 ETSILI:             iPCCObjId: .5.3.10.2
 ETSILI:             iPCCContents:
 ETSILI:               iPPackets: ...
 IP: Header Len 20 Ver 4 DSCP 00 ECN 0 Total Length 1400
 IP: Id 49370 Fragoff 0 DONT_FRAG
 IP: TTL 64 Proto 6 Checksum 29025
 IP: Source 192.168.56.108 Destination 10.45.0.3
 TCP: Source 4321 Dest 4321
 TCP: Seq 546380594
 TCP: Ack 2650820914
 TCP: DOFF 8 Flags: PSH ACK Window 510
 TCP: Checksum 34317 Urgent 0
 TCP: NOP
 TCP: NOP
 TCP: Timestamp 458841748 2693000059
unknown protocol tcp/4321
 Unknown Protocol: 4321
  e3 d5 8e 91 b2 1c c0 ef 90 6d e1 dc b5 fc 2f c0    .........m..../.
  60 8b 6b 47 e3 8b a3 24 45 7d 1c 8e 85 d0 03 dd    `.kG...$E}......
  2a 77 55 da 74 1a ed fe d7 d1 c1 e3 73 6f 7a 43    *wU.t.......sozC
  3c 16 db 1c b0 56 bc 08 3b d1 eb 6a 57 c0 71 56    <....V..;..jW.qV
  2e 0d b5 14 28 d4 dc c5 23 62 97 c7 d7 f2 61 af    ....(...#b....a.
  e0 1f 27 63 4a c6 32 e2 03 21 a3 96 bd 57 0d 3b    ..'cJ.2..!...W.;
  a0 cf 12 95 c5 af 73 3e 87 d9 2c f6 4d 65 46 76    ......s>..,.MeFv
  f4 e7 39 6d 83 d7 c7 3a c8 9e bb 26 8b d4 fb 97    ..9m...:...&....
  b0 e8 c3 0a 64 32 90 00 6e aa 49 26 24 7f aa 64    ....d2..n.I&$..d
```

(B) CC.

FIGURE 5.5: Intercepted Data.

all corresponding packets to the Mediator. The Mediator receives and uses the packet-level tracing environments such as `tracepktdump` and `tracesplit` and splits encrypted targeted data in IRI and CC payload (see detailed packet inspection in the Fig. 5.5). Later it forwards them to the Agency within the standardized interfaces (i.e., HI2 and HI3). Finally, during the decryption phase, by running the `lea.py` script and its *decryption* function, the LEA can decode the SRTP flow or decipher the target traffic and acquire the clear VoIP conversation or obtain the plaintext media file, respectively.

5.4 Performance Evaluation

This Section investigates the significant potential of the proposed LI framework through experimental tests. Specifically, it analyzes the impact of i) *several processed packets*, ii) *the durations of VoIP call and the sizes of media files by measuring the latency involved in the LI procedure*, and iii) *the deployment of the proposed LI framework on the experienced user QoS*. For this reason, four *KPIs* are considered for the real-time LI latency and one *KPI* for the experienced user QoS, as follows:

1. **UPF Acquisition Latency:** it defines the starting point of the interception procedure, and it specifies the time duration for each packet to arrive at the UPF.

2. **POI Capturing Latency:** it specifies the time duration required for each packet to be captured by the Collector.

3. **LEMF Collecting Latency:** it is the time duration in which each targeted packet is delivered to the Agency.

4. **End-to-end LI Latency:** it is the time required to process each packet during the interception process. It is considered as the sum of the above three metrics.

5. **End-to-end User Latency:** it defines the end-to-end delay experienced by each packet delivered by UE A in reaching UE B.

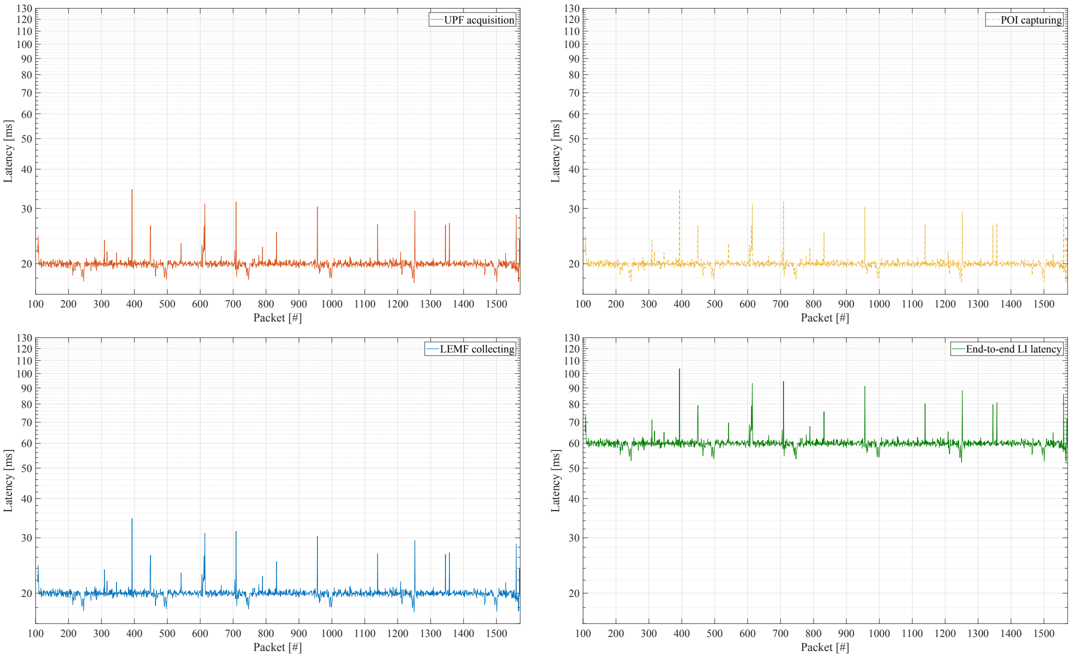


Figure 5.6: Packet latency across the different LI stages of a 30 seconds VoIP call.

5.4.1 LI for Real-Time VoIP Call

Tests examine four VoIP call conversations of varying time durations (15, 30, 45, and 60 seconds). The used VoIP simulation setup is described in the Section 5.3. Each run is repeated 10^2 times over multiple seeds, with an average of the KPI measurements.

The initial evaluation assesses the impact on the number of processed packets. Fig. 5.6, displays for each SRTP packet, the average latency into the four phases within 30 Sec VoIP call (i.e., 1550 SRTP packets). Herein, it is important to highlight that the mean latency experienced by each packet between the *UPF Acquisition Latency* and *LEMF Collecting Latency* is of a microsecond order, emphasizing the potential of processing real-time interception activities. Additionally, the *UPF Acquisition Latency*, *POI Capturing Latency*, and *LEMF Collecting Latency* consistently hover around 20 ms, where only the 1% of packets reach higher latency times (i.e., between 30 ms and 40 ms). Moreover, analyzing the *End-to-end LI Latency*, each targeted SRTP packet reaches the LEA in less than 0.07 s, demonstrating the capacity to manage real-time interceptions even during real-time VoIP calls.

Secondly, the influence of VoIP call duration on the entire interception procedure is analyzed. Fig. 5.7 shows the average and the statistics information of the End-to-end LI Latency per packet for the four different VoIP call durations. Specifically, it illustrates the 25th, 50th, and 75th percentiles, as well as the lowest and highest values of the End-to-end LI Latency, reached by each packet during the LI framework tests and it envisages how there is not any notable difference between the four VoIP call durations. In reality, the End-to-end LI Latency for each packet is typically between 55 ms and 65 ms. Furthermore, it also shows that the average End-to-end LI Latency per packet stays within 60 ms during the four VoIP calls, proving the scalability of the proposed methodology.

5.4.2 LI for End-to-end File Exchange

By exploiting the end-to-end file exchange implementation setup presented in the Section 5.3, tests consider four media files of different sizes (i.e., 10 KB, 10^2 KB, 10^3 KB, and 10^4 KB),

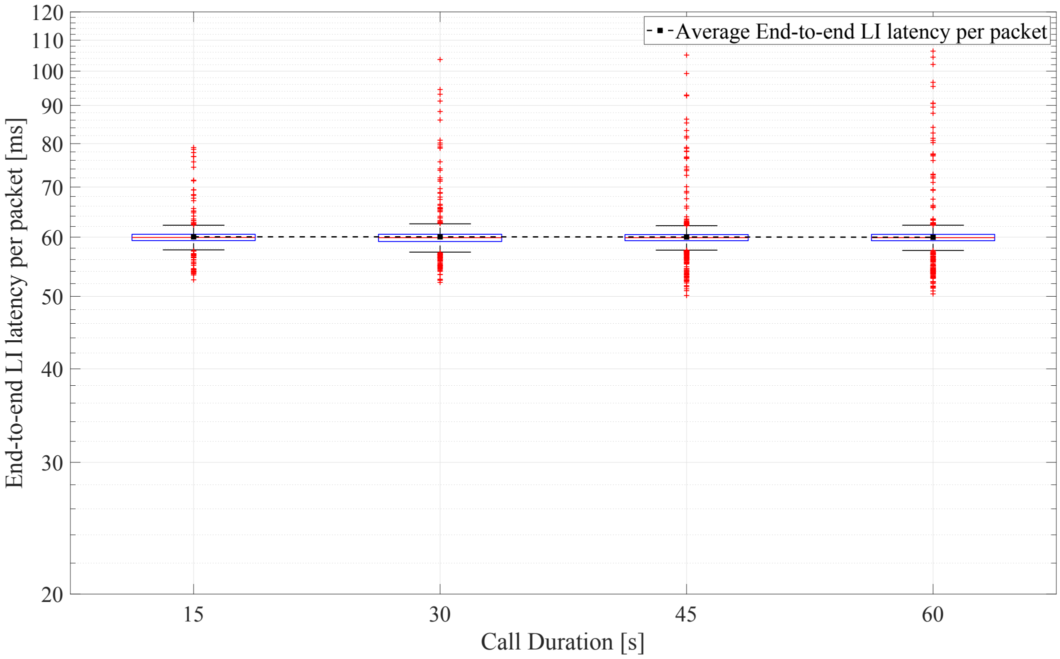


FIGURE 5.7: Statistics of the End-to-end LI latency phase per packet for the four different VoIP call durations.

where each run is repeated 10^2 times over multiple seeds and average KPI measurements are collected.

The initial test evaluates the impact on the number of processed packets. Fig. 5.8 reveals the average latency for each packet across the four phases within a specific number of packets (i.e., 7000 packets). The comprehensive results show that the differences between the main three phases are negligible, as no single phase significantly affects the End-to-end LI Latency more than the others.

In detail, the End-to-end LI Latency for each packet consistently hovers around 0.25 ms, with a small percentage of packets reaching latency times of 0.5 ms. However, it is evident that each targeted packet arrives at the LEA mostly in less than 0.5 ms by highlighting the opportunity and ability of the proposed solution to process real-time interceptions during end-to-end file exchanges.

Secondly, the influence of the exchanged file sizes on the entire duration of the end-to-end file exchange interception is evaluated. Fig. 5.9 shows the statistical data of the End-to-end LI Latency phase per packet as a function of the four file sizes. Herein, the average End-to-end LI latency per packet and the lowest and highest values are displayed, together with the 25th, 50th, and 75th percentiles. It should be noted that the minimum and maximum latencies reached by each packet are strongly dependent on the file size. Indeed, in small files (i.e., 10 KB) the latency varies between 0.12 ms and 0.14 ms, while in the heavier ones, it ranges between 0.08 ms and 0.5 ms. Additionally, the packet average End-to-end LI Latency verifies a dependency on the file size since it grows as the size of the exchanged file increases. In detail, when passing from a file size of 10 KB to 10^4 KB, the average latency exhibits an increase of two orders of magnitude. Nevertheless, Fig. 5.9 displays that even with a media file of 10^4 KB, the average packet End-to-end LI Latency at the LEA side is just over 0.25 ms by ensuring that each packet is averagely processed in real-time by the LEA.

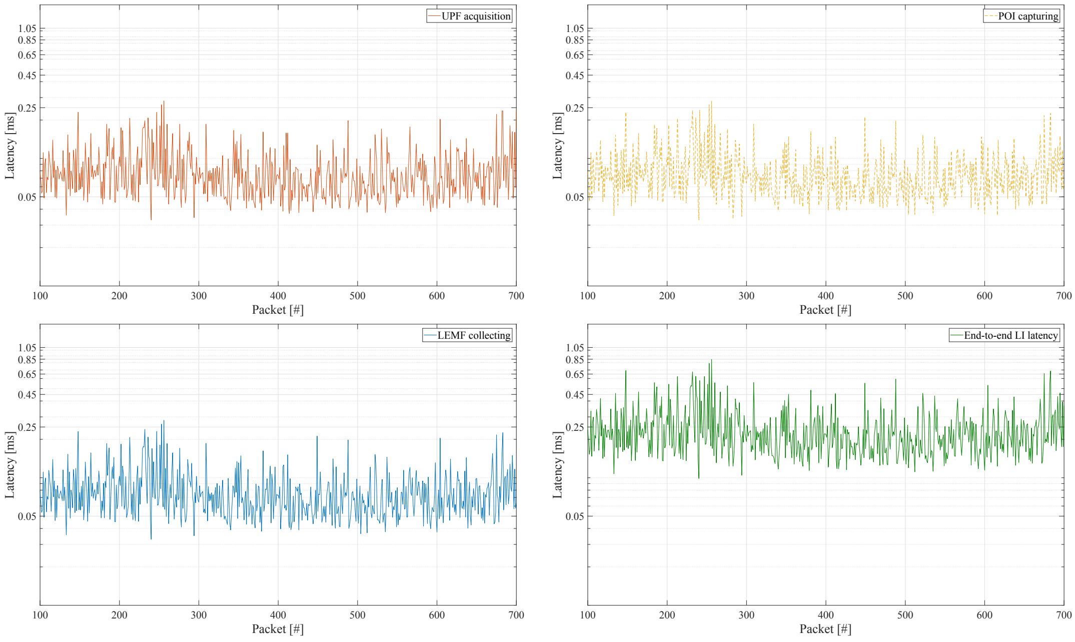


FIGURE 5.8: Packet latency across the different LI stages of a 10^3 KB exchanged file.

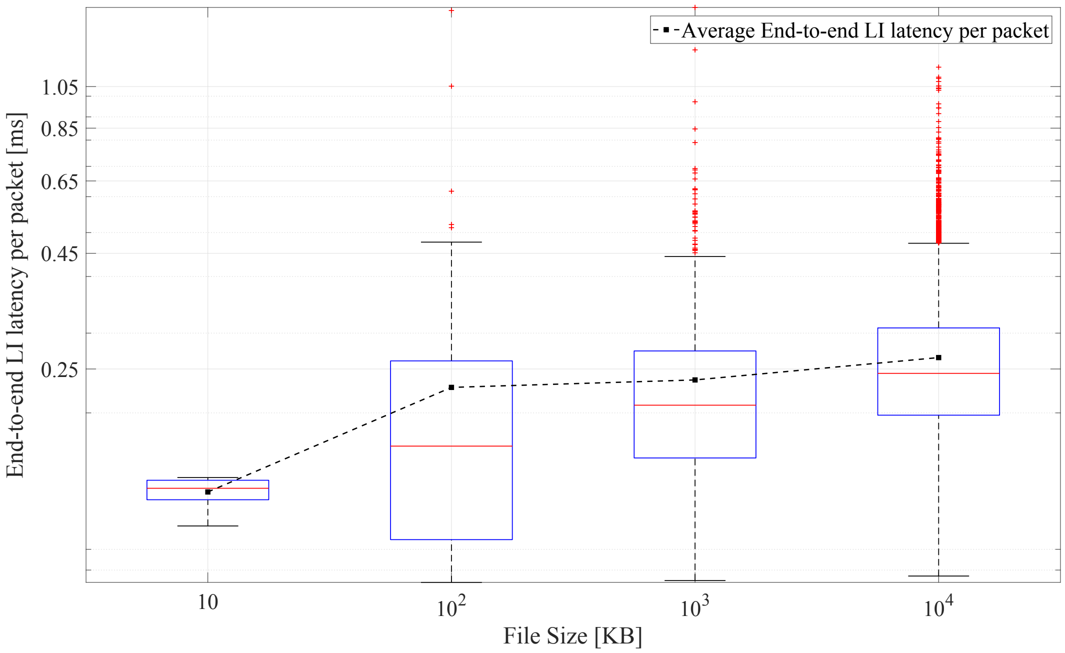


FIGURE 5.9: Statistics of the End-to-end LI latency phase per packet as a function of the four file sizes.

5.4.3 LI impact on the user QoS

This section aims to evaluate how the deployment of the proposed LI framework affects the experienced user QoS by studying the behavior of the proposed LI framework proof of concept by activating and deactivating the LI services in both real-time VoIP calls and file exchange scenarios.

Specifically, the tests examine four VoIP call conversations of different time durations (15, 30, 45, and 60 seconds) and four media files of different sizes (i.e., 10 KB, 10^2 KB,

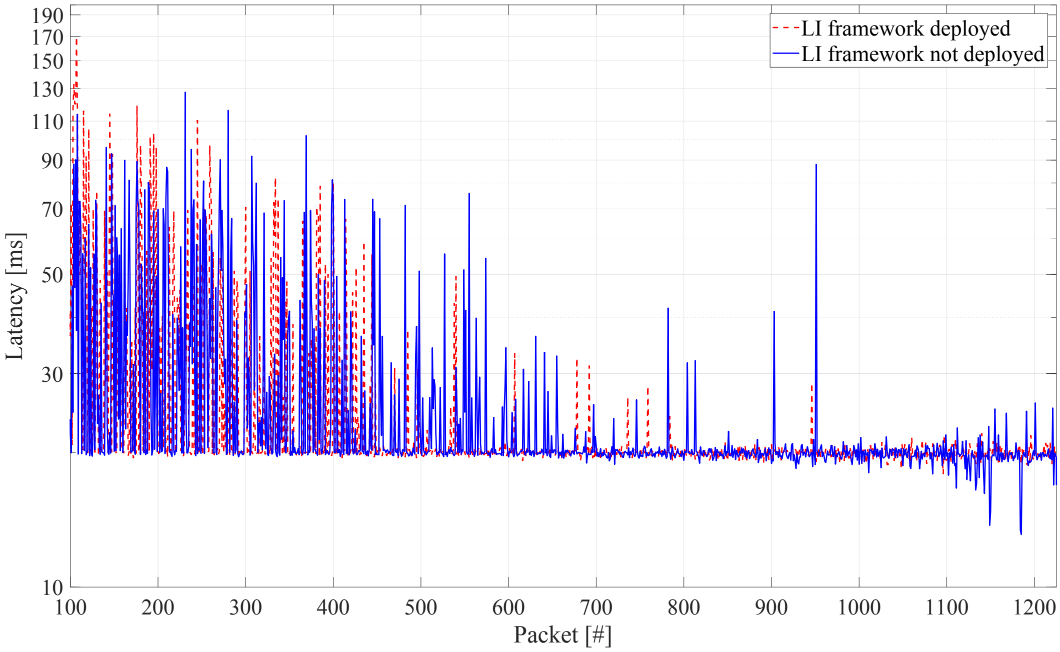


FIGURE 5.10: Packet End-to-End Latency of a 30 seconds VoIP call.

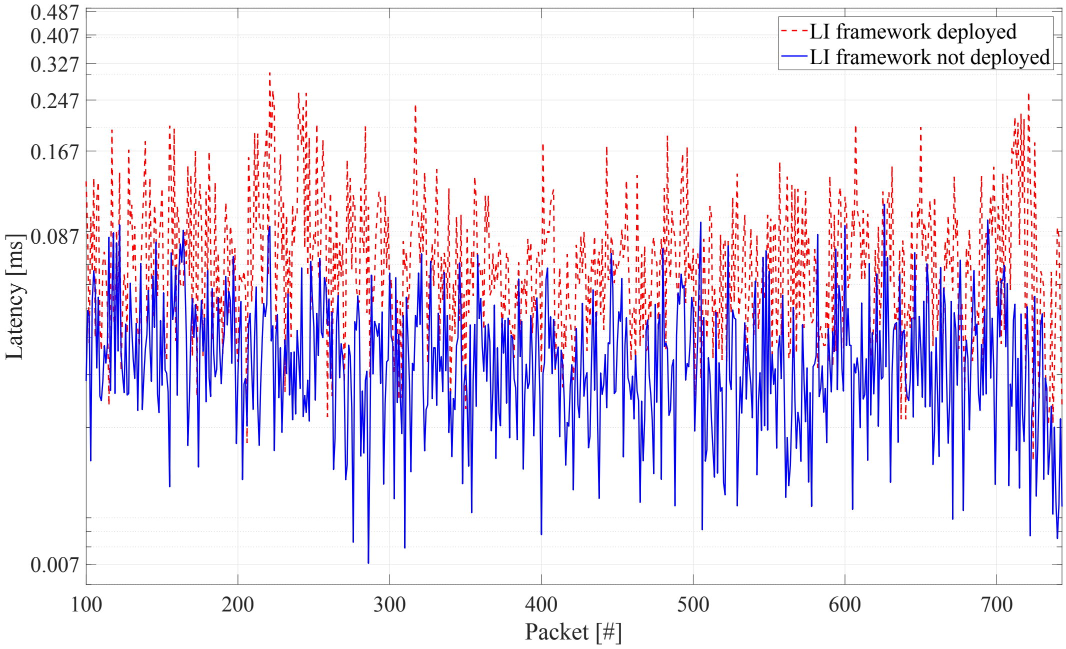


FIGURE 5.11: Packet End-to-End Latency of a 10^3 KB exchanged file.

10^3 KB, and 10^4 KB). The VoIP and file exchange simulation setups used are described in Section 5.3. Each run is repeated 10^2 times over multiple seeds, with an average of the KPI measurements.

In particular, Fig. 5.10 illustrates the end-to-end user latency experienced by each packet delivered from UE A to UE B during a 30-second VoIP call. It is noticeable that there is not a significant variation in the end-to-end user latency in terms of delay generated within the proposed LI framework. Indeed, for almost all packets, the time each packet takes to reach the UE B device when LI services are not going on is equivalent to the time it takes when the proposed LI framework is active. In detail, by adding LI services, each packet encounters an

TABLE 5.2: Average delay difference experienced by each packet by deploying or not the proposed LI framework.

VoIP call	**Call Duration [s]**	**Average delay difference experienced by a single packet [ms]**
	15	0.077452
	30	0.038487
	45	0.033085
	60	0.049420
File exchange	**File Size [KB]**	**Average delay difference experienced by a single packet [ms]**
	10	0.000012
	10^2	0.000017
	10^3	0.000031
	10^4	0.000035

average delay of around 38 microseconds. The reality is that although the proof-of-concept employs exclusive containers, the tests run on a single workstation, which may have influenced and caused the above-mentioned minor delay.

Meanwhile, Fig. 5.11 depicts the end-to-end latency experienced by each packet delivered from UE A to UE B for transmitting a 10^3 KB exchanged file. Here, the suggested LI framework implementation results in a slight increase in the packets' end-to-end latency. In particular, using LI services, each packet experiences an average delay difference of around 0.031 microseconds, which still allows the proposed LI framework to effectively work in real-time scenarios. The truth is that even though the proof-of-concept uses exclusive containers, it is executed on a single workstation, which may have affected and caused the aforementioned slight delay.

In conclusion, it is important to emphasize that the deployment of the proposed LI framework has no significant impact on the user QoS in both end-to-end file exchange and real-time VoIP call use cases.

5.4.4 Comparison: VoIP vs. File Exchange

In terms of packet latency, both scenarios demonstrate the LI framework's capability to achieve real-time processing. In real-time VoIP calls, the End-to-end LI Latency per packet is consistently around 60 ms, allowing for efficient interception even in short-duration VoIP calls. On the other hand, file exchange interception exhibits a slightly lower latency, with an average End-to-end LI Latency per packet of around 0.25 ms. This implies that the LI framework can handle real-time interception for both scenarios, with a more granular efficiency observed in file exchanges. This difference arises because, when implementing the VoIP call, all cryptographic operations are performed at the same time as sending each SRTP packet, resulting in a higher packet End-to-end LI Latency.

Secondly, the impact of different parameters is noteworthy. In real-time VoIP calls, the call duration does not significantly affect the End-to-end LI Latency, maintaining a constant range across various VoIP call durations. In contrast, in file exchange, the file size has a more pronounced effect on the End-to-end LI latency, with larger files leading to increased average latency, suggesting that the LI framework performance in file exchange is more influenced by the processed packet data size.

Thirdly, by evaluating the packet End-to-End user Latency of both a 30s VoIP call and a 10^3 KB file exchange, it is evident that the packet size influences the user-experienced latency when the LI framework is active. It is not true, however, that the suggested LI framework appears to have a more significant influence on end-to-end file exchange performances. Indeed, Table 5.2 displays the average delay difference experienced by each packet by deploying or not deploying the proposed LI framework. Here, it is evident that, in the case of VoIP conversations, the difference is three orders of magnitude more than that of the file exchange scenario.

Lastly, it is noteworthy to highlight that for both real-time VoIP call and end-to-end file exchange scenarios, the impact on the QoS experienced by the user introduced by deploying the proposed LI framework is negligible.

In summary, the scalability of the proposed LI framework becomes apparent in both scenarios. The ability to handle diverse scenarios with minimal impact on latency performance underscores the robustness and adaptability of the proposed LI framework, making it a promising solution for multiple interception applications, ranging from VoIP calls to end-to-end file exchanges.

Conclusions and Future Works

In conclusion, this thesis has examined advanced privacy-preserving techniques and security protocols within modern communication systems, underscoring the imperative role of privacy and trust in the rapidly evolving landscape of IoT and 5G networks. The integration of these networks is transforming various sectors, creating highly interconnected ecosystems where secure and efficient data exchange is essential. Each chapter has contributed to this theme by addressing specific technological challenges and proposing solutions that promote privacy, scalability, and trustworthiness across complex network environments.

Chapter 1 introduced and described the foundational technologies and enablers crucial to support the integration of IoT in 5G architectures, SDN, NFV, MEC, and network slicing. Together, these technologies create a dynamic infrastructure capable of meeting the high demands of IoT applications, enabling low-latency, high-bandwidth, and energy-efficient data transmission. As these enabling technologies become more prevalent, they have transformed information sharing but have also introduced significant privacy and security challenges, particularly in areas like data protection, trust management, and secure communications.

In response to the need for secure interactions in SIoT environments, Chapter 2 introduced an innovative multi-layered, fog-enabled SIoT architecture that leverages trust and reputation management to secure IoT services. This architecture integrates TMS to assess the behavior and reliability of devices, effectively balancing resource demands and ensuring service provision from the most trusted sources. The conceived architecture has been shown to enhance the reliability of SIoT environments, providing a decentralized, secure, and scalable model for future IoT networks. Upcoming research activities on this topic will delve deeper into enabling sophisticated services within the IoT ecosystem by utilizing entity virtualization, facilitating enhanced orchestrations within increasingly intricate network infrastructures.

Building upon the need for privacy in data-intensive environments, Chapter 3 introduced a privacy-preserving data dissemination framework based on SE combined with a publish-subscribe model at the network edge. SE allows for efficient keyword-based searches on encrypted data, addressing one of the core privacy challenges in IoT applications by enabling secure, private data queries. This framework supports a secure and efficient exchange of sensitive data, especially critical in edge-computing scenarios where data is often distributed across less secure network components. The proposed model has demonstrated that secure, privacy-focused solutions are feasible for IoT networks and suggests the need for further research into lightweight and scalable privacy-preserving techniques adaptable to various IoT use cases. In this context, future research works plan to formulate an optimized algorithm for distributing Edge Servers at the network edge based on traffic load, communication requirements, heterogeneous processing, and user dynamics.

Chapter 4 explored advancements in PLS with a focus on RFF as an authentication method. Through real-world experiments and the usage of SDRs, this chapter demonstrated the robustness of RFF-based device authentication, even when devices operate across multiple frequencies or experience interference. This investigation into PLS highlights the potential of physical-layer techniques as a complementary security layer in IoT networks, where authentication and privacy can be reinforced directly at the hardware level. This research also underscored the need for future studies on physical-layer methods, especially in environments with high interference or complex multi-frequency operations.

Finally, Chapter 5 addressed the balance between security and regulatory compliance by proposing an end-to-end encryption framework tailored for LI within modern communication networks. Recognizing the growing demand for encrypted communication in applications such as VoIP and messaging, the proposed LI framework introduces a cryptographic scheme that enables secure data exchanges while still meeting regulatory requirements for authorized access. This approach offers a balanced solution that could guide the development of LI-compliant systems as encryption becomes ubiquitous. Future research in this area will investigate the application of this technology within multiple network slices and the employment of interception procedures at the edge of the network.

The findings presented in this thesis emphasize that the convergence of modern communication systems requires new security paradigms that go beyond traditional methods, encompassing both software and hardware layers to maintain privacy and trust. Moreover, as IoT applications become more integrated into critical infrastructure, ensuring data privacy as well as the capability of conducting lawful interception in modern communication systems will be essential, especially within highly dynamic scenarios.

Appendix A

Cryptographic description of the SE algorithm presented in [94]

The technical details of the algorithm in [94] are described as follows.

Phase 1: system initialization. This attribute-based SE scheme considers two groups of order p, $\mathbb{G}$ and $\mathbb{G}_T$, and a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. At first, the trusted Authority randomly selects $\alpha, \gamma \in \mathbb{Z}_p$ and $g, h_1, h_2 \in \mathbb{G}$, and considers three hash functions $H_1, H_2, H_3 : \{0,1\} \to \{0,1\}^{log_p}$. Then, it generates the master secret key, that is M_k, and the public parameters, that are P_b, as in what follows:

$$\begin{cases} M_k = (\alpha, \gamma) \\ P_b = (g, g^{\alpha}, g^{\gamma}, h_1, h_2). \end{cases} \tag{A.1}$$

The master secret key, which is used to create users' secret keys, is kept private. The public parameters, instead, are published by the Authority. Moreover, by exploiting an AND-gate access structure based on n attributes and assuming that each attribute can assume different values, the Authority generates data consumers attributes set and data producers policies respectively denoted by: $\mathrm{X} = (x_1, x_2, ..., x_n)$ *and* $A = (a_1, a_2, ..., a_l)$.

After receiving a set of attributes from the users, the Authority produces the secret key for that data consumer. Basically, a data consumer that joins the network sends its set of attributes to the Authority. Then, the Authority chooses a random $r \in \mathbb{Z}_p$ and implements the key generation algorithm:

$$\begin{cases} \rho_1 = (h_1 g^{-r})^{\frac{1}{\alpha - \sum_{i=1}^{n} H_1(x_i)}} \\ \rho_2 = (h_2 g^{-r})^{\frac{1}{\gamma - \sum_{i=1}^{n} H_1(x_i)}}. \end{cases}$$

Accordingly, the secret key of the data consumer, S_k, is computed as:

$$S_k = (r, \rho_1, \rho_2),$$

and shared with the reference data consumer.

Phase 2: service subscription. During this phase, the data consumer generates the search Trapdoor, that is t_{Φ}. Specifically, starting from its secret key S_k, the set of k keywords $\Phi = (\phi_1, \phi_2, ..., \Phi)$ of its interest, and a random number $z_p \in \mathbb{Z}_p^*$, the Trapdoor is calculated as:

$$t_{\Phi} = (td_1, td_2, td_3), \tag{A.2}$$

where $td_1 = \rho_2^{z_p \cdot \sum_{i=1}^{k} H_2(\phi_i)}$, $td_2 = r \cdot z_p \cdot \sum_{i=1}^{k} H_2(\phi_i)$, and $td_3 = h_2^{z_p}$.

Then, the data consumer subscribes the Trapdoor to all the Edge Servers in the system.

Phase 3: data publication. Let M be the data to encrypt and to publish to the Edge

Server. $\Psi = (\psi_1, \psi_2, ..., \psi_z)$ denotes the list of z keywords associated with that data. Moreover, $A = (a_1, a_2, ..., a_l)$ represents the list of attributes forming the access policy used to protect the data against unauthorized users. The encryption algorithms consider in input the public parameters P_b, the data M, the set of keywords Ψ, and the access policy A. Indeed, by extracting a random $s \in \mathbb{Z}_p^*$, the ciphertext is obtained as:

$$ct = (C_1, C_2, C_3, v, C_4, C_5, C_6) \ , \tag{A.3}$$

where:

$$\begin{cases} C_1 = g^{\alpha s} \cdot g^{-s \cdot \sum_{i=1}^{l} H_1(a_i)} \\ C_2 = e(g,g)^s \\ C_3 = M \cdot e(g, h_1)^{-s} \\ v = H_3(C_1, C_2, C_3) \\ C_4 = g^{\gamma v} \cdot g^{-v \cdot \sum_{i=1}^{l} H_1(a_i)} \\ C_5 = e(g,g)^v \\ C_6 = g^{v \cdot \sum_{i=1}^{z} H_2(\psi_i)} \end{cases}$$

Phase 4: keyword search and data dissemination. This phase involves the POSE entity, which performs the search algorithm to determine whether the published encrypted data match one or more subscriptions stored in the Trapdoor table. In details, for each published data and for each stored subscription, the Edge Server verifies that the following equation holds:

$$e(C_4, td_1) \cdot C_5^{td_2} = e(C_6, td_3). \tag{A.4}$$

The validity of the equation proves that i) the set of keywords Ψ in ct contains the keywords Φ retrieved from t_Φ and ii) the set of attributes X belonging to the data consumer matches the access policy A used to protect the considered data. In case of matching, the search algorithm produces in output 0, otherwise it returns 1.

Phase 5: decryption. This phase allows the data consumer to decrypt the received cyphertext ct, by using its $sk = (r, \rho_1, \rho_2)$:

$$M = C_3 \cdot e(C_1, \rho_1) \cdot C_2^r. \tag{A.5}$$

Appendix B

Cryptographic description of the SE algorithm presented in [88]

The technical details of the algorithm in [88] are described as follows.

Phase 1: system initialization. This attribute-based SE scheme considers two groups of order p, $\mathbb{G}$ and $\mathbb{G}_T$, and a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Moreover, it takes g_0 and g_1 as G generators and $U = \{h_1, \ldots, h_u\}$ as the attribute set. At first, the trusted Authority randomly selects $\alpha, a \in \mathbb{Z^*}_p$ and $g_2, \hat{g_2}, h_1, h_2, \ldots, h_u \in \mathbb{G}$, and considers an hash function $H_1 : \{0,1\} \to \mathbb{Z}_p$. Then, it generates the master secret key, that is M_k, and the public parameters, that are P_b, as in what follows:

$$\begin{cases} M_k = ({g_0}^{\alpha}, a) \\ P_b = (H_1, g_0, ({g_0}^{a}, g_2, , \hat{g_2}, e(g_0, g_0)^{\alpha}, e(g_0, g_0)^{a}, U, \mathbb{G}, \mathbb{G}_T). \end{cases} \tag{B.1}$$

The master secret key, which is used to create users' secret keys, is kept private. The public parameters, instead, are published by the Authority.

Moreover, lets consider n attributes and assuming that each attribute can assume different values, the Authority generates data consumers attributes set and data producers policies respectively denoted by: $\mathrm{X} = (x_1, x_2, ..., x_n)\ and\ A = (a_1, a_2, ..., a_l)$.

After receiving a set of attributes from the users, the Authority produces the secret key for that data consumer. Basically, a data consumer that joins the network sends its set of attributes $\mathrm{X} = (x_1, x_2, ..., x_n)$ to the Authority. Then, the Authority chooses a random $k, v, z \in \mathbb{Z}_p$ and implements the key generation algorithm:

$$\begin{cases} K = {g_0}^{\alpha}{g_0}^{a_k} \\ A = {g_0}^{v} \\ B = {g_2}^{a}\hat{g_2}^{v} \\ A_x = h_x^v \end{cases}$$

Accordingly, the secret key of the data consumer, S_k, is computed as:

$$S_k = (K, A, B, \{A_x\}),$$

and shared with the reference data consumer.

Phase 2: service subscription. During this phase, the data consumer generates the search Trapdoor, that is t_Φ. Specifically, starting from its secret key S_k, the keyword Φ of its interest, and a random number $u \in \mathbb{Z}_p^*$, the Trapdoor is calculated as:

$$t_\Phi = (td_1, td_2, td_3), \tag{B.2}$$

where $td_1 = B\hat{g_2}^{H_1(\Phi)}$, $td_2 = A{g_0}^{uH_1(\Phi)}$, and $td_{3,x} = A_x * h_x^{uH_1(\Phi)}$.

Then, the data consumer subscribes the Trapdoor to all the Edge Servers in the system.

Phase 3: data publication. Let M be the data to encrypt and to publish to the Edge Server. $\Psi = (\psi_1, \psi_2, ..., \psi_z)$ denotes the list of z keywords associated with that data. Moreover, $A = (a_1, a_2, ..., a_l)$ represents the list of attributes forming the access policy used to protect the data against unauthorized users. The encryption algorithms consider in input the public parameters P_b, the data M, the keyword Ψ, and the access policy A. Indeed, by randomly extracting $d, f, \epsilon \in \mathbb{Z}_p^*$, the ciphertext is obtained as:

$$ct = (C_1, C_2, C_3, C_4, C_5, C_6, C_7)\ , \tag{B.3}$$

where:

$$\begin{cases} C_1 = F_d e(g_0, g_0)^{\alpha,d} \\ C_2 = g_0^d g_0^f \\ C_3 = g_1^d g^f \\ C_4 = g_0^d \; C_5 = e(g_0^a, g_2)^\epsilon \\ C_6 = g_0^\epsilon \\ C_7 = g_0^{H_1(\Phi)\epsilon} \end{cases}$$

Phase 4: keyword search and data dissemination. This phase involves the POSE entity, which performs the search algorithm to determine whether the published encrypted data match one or more subscriptions stored in the Trapdoor table. In details, for each published data and for each stored subscription, the Edge Server verifies that the following equation holds:

$$\frac{e(td_1, C_6)}{C_5} = \prod_{x \in X} (e(C_{6x}, td_2) e(C_7, td_3)). \tag{B.4}$$

The validity of the equation proves that i) the keyword Ψ in ct corresponds the keyword Φ retrieved from t_Φ and ii) the set of attributes U belonging to the data consumer matches the access policy A used to protect the considered data. In case of matching, the search algorithm produces in output 0, otherwise it returns 1.

Phase 5: decryption. This phase allows the data consumer to decrypt the received cyphertext ct, by using its secret key:

$$M = \frac{e(g_0, g_0)^{\alpha d} \cdot e(g_0, g_0)^{kad}}{e(g_0^{\alpha + ak}, g_0^d)}. \tag{B.5}$$

Appendix C

Detailed Description of the Key Escrow Algorithm

This Section aims at technically presenting an in-depth description of the used *IDBC Key Escrow Algorithm* designed and developed in [168]. Specifically, let's suppose that UE A is the under surveillance subscriber and that the LEA presents, via the interface HI1, the LI warrant for intercepting a specific communication session between UE A and UE B to the AUSF. Thus, the ***Key Negotiation Phase*** of the LI framework is detailed below.

Here, the TKA possesses a master secret key $M \in \mathcal{Z}_p{}^*$ and computes public/private key pairs for subscribers based on their unique identities. Assuming that UE A and UE B share their unique identities, ID_A and ID_B with the TKA, it employs a hash function $\mathcal{H} : \mathcal{Z}_p^* \to \mathcal{G}$ to generate:

$$\begin{cases} p_A = \mathcal{H}(ID_A) & \text{UE A public key,} \\ P_A = M\mathcal{H}(ID_A) & \text{UE A private key,} \end{cases}$$

and

$$\begin{cases} p_B = \mathcal{H}(ID_B) & \text{UE B public key,} \\ P_B = M\mathcal{H}(ID_B) & \text{UE B private key,} \end{cases}$$

where, how specified into [168], retrieving M is computationally complex as solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) has proved in [172].

Furthermore, as the TKA functions as the 'Escrow Agency,' there is no need for an additional Key Escrow process. Additionally, the initial sharing of the key k_A between AUSF and UE A, as well as the key k_B between AUSF and UE B, is established.

Firstly, UE A sends μ_1 to the AUSF using the equation:

$$\mu_1 = e_{k_A}(ID_A||ID_B||r_A||sign_A(r_A)),$$

where $||$ denotes concatenation, e_{k_A} denotes the encryption function adopting the shared key k_A, r_A is a random integer generated by the UE A, and $sign_A(r_A)$ is its corresponding signature.

Thus, the AUSF receives and decrypts μ_1, verifies r_A with the signature $sign_A(r_A)$, and then constructs μ_2 to be sent to UE B using the equation:

$$\mu_2 = e_{k_B}(ID_A \,||\, ID_B \,||\, r_A \,||\, \text{sign}_A(r_A)),$$

here e_{k_B} denotes the encryption function using the shared key k_B, r_A is the random integer generated by the UE A, and $sign_A(r_A)$ is its corresponding signature.

Upon receiving μ_2, UE B first decrypts and verifies r_A with the signature $sign_A(r_A)$. Subsequently, UE B forwards μ_3 to the AUSF using the equation:

$$\mu_3 = e_{k_B}(ID_B \,||\, ID_A \,||\, r_B \,||\, \text{sign}_B(r_B)),$$

Here, e_{k_B} represents the encryption function using the shared key k_B, r_B is a randomly generated nonce by UE B, and $sign_B(r_B)$ is the corresponding signature.

Moreover, UE B is able now to compute $\eta = devf(r_A, r_B)$, where $devf$ is a derivation function from r_A and r_B.

Upon verifying μ_3, the AUSF firstly calculates $\eta = devf(r_A, r_B)$ and then generates μ_4 and sends to the UE A:

$$\mu_4 = e_{k_A}(ID_B \,||\, ID_A \,||\, r_B \,||\, \text{sign}_B(r_B)),$$

where e_{k_B} denotes the encryption function using the shared key k_B, r_B is a randomly generated nonce by UE B, and $sign_B(r_B)$ is the corresponding signature.

Thus, the UE A is now able to verify the identity of UE B and then compute $\eta = devf(r_A, r_B)$.

Once the communication procedure among the subscribers is completed, the interception procedures continue with the AUSF sending τ_1 (containing the LEA request) to the TKA:

$$\tau_1 = \eta || ID_A || \text{LEA } request \ .$$

Subsequently, the TKA sends τ_2 to the LEA via the HI2 interface:

$$\tau_2 = \eta \cdot M\mathcal{H}(ID_A),$$

where the "$\cdot$" operator denotes the multiplication.

At this point, all entities possess the necessary cryptographic material to generate the communication session key k_{AB} through which the end-to-end communication session will be encrypted.

Firstly, UE A computes $k_{AB} = e(\eta \cdot M\mathcal{H}(ID_A), \mathcal{H}(ID_B))$. Secondly, UE B computes $k_{BA} = e(\mathcal{H}(ID_A), \eta \cdot M\mathcal{H}(ID_B))$.

Here, the function e($\cdot$) defines the bilinear function operation, and using IDBC-based model properties described in [168], the validity of the two equations is proven as follows:

$$\begin{aligned} k_{AB} &= e(\eta \cdot M\mathcal{H}(ID_A), \mathcal{H}(ID_B)) = \\ &= e(\mathcal{H}(ID_A), \mathcal{H}(ID_B))^{\eta \cdot M} = \\ &= e(\mathcal{H}(ID_A), \eta \cdot M\mathcal{H}(ID_B)) = k_{BA} \ . \end{aligned}$$

While the LEA employs a public hash function $\mathcal{H} : \mathcal{Z} \to \mathcal{P}$ to calculates $\mathcal{H}(ID_A)$ and $\mathcal{H}(ID_B)$, it uses τ_2 to compute the communication session key k_{AB}.

Bibliography

[1] G. Sciddurlo, I. Huso, D. Striccoli, G. Piro, and G. Boggia, "A multi-tiered social iot architecture for scalable and trusted service provisioning," in *2021 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2021, pp. 1–6.

[2] I. Huso, G. Piro, and G. Boggia, "Distributed and privacy-preserving data dissemination at the network edge via attribute-based searchable encryption," in *2022 20th Mediterranean Communication and Computer Networking Conference (MedComNet)*, 2022, pp. 122–130. DOI: 10.1109/MedComNet55087.2022.9810394.

[3] I. Huso, D. Sparapano, G. Piro, and G. Boggia, "Privacy-preserving data dissemination scheme based on searchable encryption, publish–subscribe model, and edge computing," *Computer Communications*, vol. 203, pp. 262–275, 2023.

[4] I. Huso, S. Sciancalepore, G. Oligeri, G. Piro, and G. Boggia, "Requency matters: On the impact of carrier frequency on privacy in radio fingerprinting," *IEEE Wireless Communication Letters*, 2024.

[5] I. Huso, S. Carbonara, S. Sciancalepore, G. Oligeri, G. Piro, and G. Boggia, "Jamming echoes: On the impact of Out-Of-Band interference on radio frequency fingerprinting," in *2025 IEEE Wireless Communications and Networking Conference (WCNC) (WCNC 2025)*, Milan, Italy, Mar. 2025.

[6] I. Huso, M. Olivieri, L. Galgano, A. Rashid, G. Piro, and G. Boggia, "Design and implementation of a looking-forward lawful interception architecture for future mobile communication systems," *Computer Networks*, vol. 249, p. 110 518, 2024.

[7] S. F. Ahmed, M. S. B. Alam, S. Afrin, *et al.*, "Toward a secure 5g-enabled internet of things: A survey on requirements, privacy, security, challenges, and opportunities," *IEEE Access*, vol. 12, pp. 13 125–13 145, 2024. DOI: 10.1109/ACCESS.2024.3352508.

[8] A. Vanelli-Coralli, A. Guidotti, T. Foggi, G. Colavolpe, and G. Montorsi, "5g and beyond 5g non-terrestrial networks: Trends and research challenges," in *2020 IEEE 3rd 5G World Forum (5GWF)*, 2020, pp. 163–169. DOI: 10.1109/5GWF49715.2020.9221119.

[9] R. Bajracharya, R. Shrestha, S. A. Hassan, H. Jung, and H. Shin, "5g and beyond private military communication: Trend, requirements, challenges and enablers," *IEEE Access*, vol. 11, pp. 83 996–84 012, 2023. DOI: 10.1109/ACCESS.2023.3303211.

[10] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, "A survey on security and privacy of 5g technologies: Potential solutions, recent advancements, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 196–248, 2020. DOI: 10.1109/COMST.2019.2933899.

[11] B. Kizilkaya, G. Zhao, Y. A. Sambo, L. Li, and M. A. Imran, "5g-enabled education 4.0: Enabling technologies, challenges, and solutions," *IEEE Access*, vol. 9, pp. 166 962–166 969, 2021. DOI: 10.1109/ACCESS.2021.3136361.

[12] L. Atzori, A. Iera, and G. Morabito, “From "smart objects" to "social objects": The next evolutionary step of the internet of things,” *IEEE Communications Magazine*, vol. 52, no. 1, pp. 97–105, 2014. DOI: 10.1109/MCOM.2014.6710070.

[13] M. Nitti, R. Girau, and L. Atzori, “Trustworthiness management in the social internet of things,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1253–1266, 2014. DOI: 10.1109/TKDE.2013.105.

[14] A. Ometov, O. L. Molua, M. Komarov, and J. Nurmi, “A survey of security in cloud, edge, and fog computing,” *Sensors*, vol. 22, no. 3, 2022, ISSN: 1424-8220. DOI: 10.3390/s22030927.

[15] M. Mitev, A. Chorti, H. V. Poor, and G. P. Fettweis, “What physical layer security can do for 6g security,” *IEEE Open Journal of Vehicular Technology*, vol. 4, pp. 375–388, 2023. DOI: 10.1109/OJVT.2023.3245071.

[16] J. Zhang, Z. Yan, S. Fei, M. Wang, T. Li, and H. Wang, “Is today’s end-to-end communication security enough for 5g and its beyond?” *IEEE Network*, vol. 36, no. 1, pp. 105–112, 2022. DOI: 10.1109/MNET.101.2100189.

[17] Y. Li, Y. Yu, W. Susilo, Z. Hong, and M. Guizani, “Security and Privacy for Edge Intelligence in 5G and Beyond Networks: Challenges and Solutions,” *IEEE Wireless Communications*, vol. 28, pp. 63–69, 2021.

[18] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, “Network slicing and softwarization: A survey on principles, enabling technologies, and solutions,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018. DOI: 10.1109/COMST.2018.2815638.

[19] P. Rost, C. Mannweiler, D. S. Michalopoulos, *et al.*, “Network slicing to enable scalability and flexibility in 5g mobile networks,” *IEEE Communications Magazine*, vol. 55, no. 5, pp. 72–79, 2017. DOI: 10.1109/MCOM.2017.1600920.

[20] R. Su, A. R. Sfar, E. Natalizio, P. Moyal, and Y.-Q. Song, “Ensuring trustworthiness in ioit/aiot: A phase-based approach,” *IEEE Internet of Things Magazine*, vol. 5, no. 2, pp. 84–88, 2022. DOI: 10.1109/IOTM.001.2100190.

[21] S. Sagar, A. Mahmood, K. Wang, Q. Z. Sheng, J. K. Pabani, and W. E. Zhang, “Trust–siot: Toward trustworthy object classification in the social internet of things,” *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1210–1223, 2023. DOI: 10.1109/TNSM.2023.3247831.

[22] Y. Harbi, Z. Aliouat, A. Refoufi, and S. Harous, “Recent security trends in internet of things: A comprehensive survey,” *IEEE Access*, vol. 9, pp. 113 292–113 314, 2021. DOI: 10.1109/ACCESS.2021.3103725.

[23] N. Andola, R. Gahlot, V. K. Yadav, S. Venkatesan, and S. Verma, “Searchable encryption on the cloud: A survey,” *The Journal of Supercomputing*, pp. 1–33, 2022.

[24] U. Varri, S. Pasupuleti, and K. Kadambari, “A scoping review of searchable encryption schemes in cloud computing: Taxonomy, methods, and recent developments,” *The Journal of Supercomputing*, vol. 76, no. 4, pp. 3013–3042, 2020.

[25] N. Xie, Z. Li, and H. Tan, “A survey of physical-layer authentication in wireless communications,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 282–310, 2021. DOI: 10.1109/COMST.2020.3042188.

[26] E. Illi, M. Qaraqe, S. Althunibat, *et al.*, “Physical layer security for authentication, confidentiality, and malicious node detection: A paradigm shift in securing iot networks,” *IEEE Communications Surveys & Tutorials*, 2024.

[27] M. Alatawi and N. Saxena, "Sok: An analysis of end-to-end encryption and authentication ceremonies in secure messaging systems," in *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '23, Guildford, United Kingdom: Association for Computing Machinery, 2023, pp. 187–201, ISBN: 9781450398596.

[28] N. Borisov, I. Goldberg, and E. Brewer, "Off-the-record communication, or, why not to use pgp," in *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society*, ser. WPES '04, Washington DC, USA: Association for Computing Machinery, 2004, pp. 77–84, ISBN: 1581139683.

[29] R. Stedman, K. Yoshida, and I. Goldberg, "A user study of off-the-record messaging," in *Proceedings of the 4th Symposium on Usable Privacy and Security*, ser. SOUPS '08, Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2008, pp. 95–104, ISBN: 9781605582764.

[30] K. Cohn-Gordon, C. Cremers, L. Garratt, J. Millican, and K. Milner, "On ends-to-ends encryption: Asynchronous group messaging with strong security guarantees," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18, Toronto, Canada: Association for Computing Machinery, 2018, pp. 1802–1819, ISBN: 9781450356930.

[31] The European Commission, *Eurostat, Recorded offences by offence category - Police data*. The European Commission, 2023.

[32] Council of the European Union and EUROPOL, "Position paper on 5G," The European Commission, Tech. Rep., Apr. 2019.

[33] A. Zannou, A. Boulaalam, and E. H. Nfaoui, "Siot: A new strategy to improve the network lifetime with an efficient search process," *Future Internet*, vol. 13, no. 1, 2021, ISSN: 1999-5903. DOI: 10.3390/fi13010004.

[34] L. Wei, J. Wu, C. Long, and B. Li, "On designing context-aware trust model and service delegation for social internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4775–4787, 2021. DOI: 10.1109/JIOT.2020.3028380.

[35] B. Farahbakhsh, A. Fanian, and M. H. Manshaei, "TGSM: Towards trustworthy group-based service management for social IoT," *Internet of Things*, vol. 13, p. 100 312, 2021, ISSN: 2542-6605. DOI: https://doi.org/10.1016/j.iot.2020.100312.

[36] Y. Yi, Z. Zhang, L. T. Yang, X. Deng, L. Yi, and X. Wang, "Social interaction and information diffusion in social internet of things: Dynamics, cloud-edge, traceability," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2177–2192, 2021. DOI: 10.1109/JIOT.2020.3026995.

[37] M. Amiri-Zarandi and R. A. Dara, "Blockchain-based trust management in social internet of things," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing*, 2020, pp. 49–54. DOI: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00024.

[38] Z. Chen, R. Ling, C.-M. Huang, and X. Zhu, "A scheme of access service recommendation for the social internet of things," *International Journal of Communication Systems*, vol. 29, no. 4, pp. 694–706, 2016. DOI: https://doi.org/10.1002/dac.2930.

[39] R. K. Chahal, N. Kumar, and S. Batra, "Trust management in social internet of things: A taxonomy, open issues, and challenges," *Computer Communications*, vol. 150, pp. 13–46, 2020, ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2019.10.034.

[40] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (siot) – when social networks meet the internet of things: Concept, architecture and network characterization," *Computer Networks*, vol. 56, Nov. 2012. DOI: 10.1016/j.comnet.2012.07.010.

[41] W. Z. Khan, M. Y. Aalsalem, M. K. Khan, and Q. Arshad, "When social objects collaborate: Concepts, processing elements, attacks and challenges," *Computers and Electrical Engineering*, vol. 58, pp. 397–411, 2017, ISSN: 0045-7906. DOI: https://doi.org/10.1016/j.compeleceng.2016.11.014.

[42] W. Z. Khan, Q. .-. .-. Arshad, S. Hakak, M. K. Khan, and Saeed-Ur-Rehman, "Trust management in social internet of things: Architectures, recent advancements and future challenges," *IEEE Internet of Things Journal*, pp. 1–1, 2020. DOI: 10.1109/JIOT.2020.3039296.

[43] R. Faqihi, D. Ramakrishnan, and D. Mavaluru, "An evolutionary study on the threats, trust, security, and challenges in siot (social internet of things)," *Materials today: proceedings*, Nov. 2020. DOI: 10.1016/j.matpr.2020.09.618.

[44] T. Soo Fun and A. Samsudin, "Recent technologies, security countermeasure and ongoing challenges of industrial internet of things (iiot): A survey," *Sensors*, vol. 21, no. 19, 2021.

[45] H. Li and T. Jing, "A lightweight fine-grained searchable encryption scheme in fog-based healthcare iot networks," *Wireless Communications and Mobile Computing*, 2019.

[46] Y. Tao, P. Xu, and H. Jin, "Secure data sharing and search for cloud-edge-collaborative storage," *IEEE Access*, vol. 8, pp. 15 963–15 972, 2020. DOI: 10.1109/ACCESS.2019.2962600.

[47] Mamta, B. B. Gupta, and M. D. Lytras, "Fog-enabled secure and efficient fine-grained searchable data sharing and management scheme for iot-based healthcare systems," *IEEE Transactions on Engineering Management*, pp. 1–13, 2022. DOI: 10.1109/TEM.2022.3143661.

[48] G. Gür, A. Kalla, C. de Alwis, *et al.*, "Integration of icn and mec in 5g and beyond networks: Mutual benefits, use cases, challenges, standardization, and future research," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1382–1412, 2022. DOI: 10.1109/OJCOMS.2022.3195125.

[49] K. Velasquez, D. Perez Abreu, M. Curado, and E. Monteiro, "Resource orchestration in 5g and beyond: Challenges and opportunities," *Computer Communications*, vol. 192, pp. 311–315, 2022, ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2022.06.019.

[50] ETSI, "Multi-access edge computing (mec): Framework and reference architecture," ETSI, standard ETSI GS MEC 003 v.2.1.1, 2019.

[51] NIST, "Guide to attribute based access control (abac) definition and considerations," NIST, standard NIST Special Publication 800-162, 2014.

[52] M. Rasori, P. Perazzo, G. Dini, and S. Yu, "Indirect revocable kp-abe with revocation undoing resistance," *IEEE Transactions on Services Computing*, vol. 15, no. 5, pp. 2854–2868, 2022. DOI: 10.1109/TSC.2021.3071859.

[53] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance cp-abe with efficient attribute revocation for cloud storage," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1767–1777, 2018. DOI: 10.1109/JSYST.2017.2667679.

[54] V. Pedreira, D. Barros, and P. Pinto, "A review of attacks, vulnerabilities, and defenses in industry 4.0 with new challenges on data sovereignty ahead," *Sensors*, vol. 21, no. 15, 2021.

[55] C.-J. Wang and J.-F. Luo, "A key-policy attribute-based encryption scheme with constant size ciphertext," in *2012 Eighth International Conference on Computational Intelligence and Security*, 2012. DOI: 10.1109/CIS.2012.106.

[56] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, 2007. DOI: 10.1109/SP.2007.11.

[57] S. Li, S. Zhao, G. Min, L. Qi, and G. Liu, "Lightweight privacy-preserving scheme using homomorphic encryption in industrial internet of things," *IEEE Internet of Things Journal*, 2021. DOI: 10.1109/JIOT.2021.3066427.

[58] K. Zhang, J. Long, X. Wang, H.-N. Dai, K. Liang, and M. Imran, "Lightweight searchable encryption protocol for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4248–4259, 2021. DOI: 10.1109/TII.2020.3014168.

[59] B. Chen, L. Wu, N. Kumar, K.-K. R. Choo, and D. He, "Lightweight searchable public-key encryption with forward privacy over iiot outsourced data," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 4, pp. 1753–1764, 2021. DOI: 10.1109/TETC.2019.2921113.

[60] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceeding 2000 IEEE Symposium on Security and Privacy. S P 2000*, 2000, pp. 44–55. DOI: 10.1109/SECPRI.2000.848445.

[61] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology - EUROCRYPT 2004*, C. Cachin and J. L. Camenisch, Eds., Springer Berlin Heidelberg, 2004, pp. 506–522.

[62] I. R. Jeong, J. O. Kwon, D. Hong, and D. H. Lee, "Constructing peks schemes secure against keyword guessing attacks is possible?" *Computer communications*, vol. 32, no. 2, pp. 394–396, 2009.

[63] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "Dual-server public-key encryption with keyword search for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 789–798, 2016. DOI: 10.1109/TIFS.2015.2510822.

[64] P. Xu, H. Jin, Q. Wu, and W. Wang, "Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack," *IEEE Transactions on Computers*, vol. 62, no. 11, pp. 2266–2277, 2013. DOI: 10.1109/TC.2012.215.

[65] J. Shen, C. Wang, A. Wang, S. Ji, and Y. Zhang, "A searchable and verifiable data protection scheme for scholarly big data," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 1, pp. 216–225, 2021. DOI: 10.1109/TETC.2018.2830368.

[66] W. Zhang, Y. Lin, and G. Qi, "Catch you if you misbehave: Ranked keyword search results verification in cloud computing," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 74–86, 2018. DOI: 10.1109/TCC.2015.2481389.

[67] K. Liang and W. Susilo, "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1981–1992, 2015. DOI: 10.1109/TIFS.2015.2442215.

[68] J. Cui, H. Zhou, H. Zhong, and Y. Xu, "Akser: Attribute-based keyword search with efficient revocation in cloud computing," *Information Sciences*, vol. 423, pp. 343–352, 2018, ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2017.09.029.

[69] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and fine-grained attribute-based data storage in cloud computing," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 785–796, 2017. DOI: 10.1109/TSC.2016.2520932.

[70] Y. Miao, X. Liu, R. H. Deng, *et al.*, "Hybrid keyword-field search with efficient key management for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3206–3217, 2019. DOI: 10.1109/TII.2018.2877146.

[71] J. Sun, L. Ren, S. Wang, and X. Yao, "Multi-keyword searchable and data verifiable attribute-based encryption scheme for cloud storage," *IEEE Access*, vol. 7, pp. 66 655–66 667, 2019. DOI: 10.1109/ACCESS.2019.2917772.

[72] Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, and H. Li, "Practical attribute-based multi-keyword search scheme in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3008–3018, 2018. DOI: 10.1109/JIOT.2017.2779124.

[73] N. Karnik, U. Bora, K. Bhadri, P. Kadambi, and P. Dhatrak, "A comprehensive study on current and future trends towards the characteristics and enablers of industry 4.0," *Journal of Industrial Information Integration*, p. 100 294, 2021.

[74] K. Fan, Q. Chen, R. Su, *et al.*, "Msiap: A dynamic searchable encryption for privacy-protection on smart grid with cloud-edge-end," *IEEE Transactions on Cloud Computing*, 2021. DOI: 10.1109/TCC.2021.3134015.

[75] R. Zhou, X. Zhang, X. Wang, G. Yang, H. Wang, and Y. Wu, "Privacy-preserving data search with fine-grained dynamic search right management in fog-assisted internet of things," *Information Sciences*, vol. 491, pp. 251–264, 2019.

[76] Y. Miao, J. Ma, X. Liu, J. Weng, H. Li, and H. Li, "Lightweight fine-grained search over encrypted data in fog computing," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 772–785, 2019. DOI: 10.1109/TSC.2018.2823309.

[77] H. Wang, K. Fan, K. Zhang, Z. Wang, H. Li, and Y. Yang, "Encrypted data retrieval and sharing scheme in space–air–ground-integrated vehicular networks," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5957–5970, 2022. DOI: 10.1109/JIOT.2021.3062626.

[78] K. Wang, C.-M. Chen, M. Shojafar, Z. Tie, M. Alazab, and S. Kumari, "Affirm: Provably forward privacy for searchable encryption in cooperative intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2022. DOI: 10.1109/TITS.2022.3177899.

[79] S. S. Chaeikar, A. Jolfaei, and N. Mohammad, "Ai-enabled cryptographic key management model for secure communications in the internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–, 2022. DOI: 10.1109/TITS.2022.3200250.

[80] Y. Bao, W. Qiu, P. Tang, and X. Cheng, "Efficient, revocable, and privacy-preserving fine-grained data sharing with keyword search for the cloud-assisted medical iot system," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 2041–2051, 2022. DOI: 10.1109/JBHI.2021.3100871.

[81] J. Cui, J. Lu, H. Zhong, Q. Zhang, C. Gu, and L. Liu, "Parallel key-insulated multiuser searchable encryption for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4875–4883, 2022. DOI: 10.1109/TII.2021.3110193.

[82] J. Li, X. Wang, Q. Gan, and F. Wang, "Mfpse: Multi-user forward private searchable encryption with dynamic authorization in cloud computing," *Computer Communications*, vol. 191, pp. 184–193, 2022.

[83] S. Abdelfattah, M. Baza, M. M. E. A. Mahmoud, M. M. Fouda, K. A. Abualsaud, and M. Guizani, "Multidata-owner searchable encryption scheme over medical cloud data with efficient access control," *IEEE Systems Journal*, vol. 16, no. 3, pp. 5067–5078, 2022. DOI: 10.1109/JSYST.2021.3123956.

[84] X. Tang, C. Guo, Y. Ren, C. Wang, and K.-K. R. Choo, "A global secure ranked multikeyword search based on the multiowner model for cloud-based systems," *IEEE Systems Journal*, vol. 16, no. 2, pp. 1717–1728, 2022. DOI: 10.1109/JSYST.2022.3157530.

[85] S. Gao, Y. Chen, J. Zhu, Z. Sui, R. Zhang, and X. Ma, "Bpms: Blockchain-based privacy-preserving multi-keyword search in multi-owner setting," *IEEE Transactions on Cloud Computing*, pp. 1–13, 2022. DOI: 10.1109/TCC.2022.3196712.

[86] Q. Zhang, G. Wang, W. Tang, K. Alinani, Q. Liu, and X. Li, "Efficient personalized search over encrypted data for mobile edge-assisted cloud storage," *Computer Communications*, vol. 176, pp. 81–90, 2021.

[87] D. Wang, P. Wu, B. Li, H. Du, and M. Luo, "Multi-keyword searchable encryption for smart grid edge computing," *Electric Power Systems Research*, vol. 212, p. 108 223, 2022, ISSN: 0378-7796. DOI: https://doi.org/10.1016/j.epsr.2022.108223.

[88] K. Gu, W. Zhang, X. Li, and W. Jia, "Self-verifiable attribute-based keyword search scheme for distributed data storage in fog computing with fast decryption," *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 271–288, 2022. DOI: 10.1109/TNSM.2021.3123475.

[89] J. Liu, Y. Li, R. Sun, *et al.*, "Emk-abse: Efficient multi-keyword attribute-based searchable encryption scheme through cloud-edge coordination," *IEEE Internet of Things Journal*, pp. 1–1, 2022. DOI: 10.1109/JIOT.2022.3163340.

[90] Q. Chen, K. Fan, K. Zhang, H. Wang, H. Li, and Y. Yang, "Privacy-preserving searchable encryption in the intelligent edge computing," *Computer Communications*, vol. 164, pp. 31–41, 2020, ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2020.09.012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366420319320.

[91] S. Niu, M. Song, L. Fang, F. Yu, S. Han, and C. Wang, "Keyword search over encrypted cloud data based on blockchain in smart medical applications," *Computer Communications*, vol. 192, pp. 33–47, 2022, ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2022.05.018.

[92] D. C. Nguyen, M. Ding, P. N. Pathirana, *et al.*, "6g internet of things: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 359–383, 2022. DOI: 10.1109/JIOT.2021.3103320.

[93] R. W. Coutinho and A. Boukerche, "Design of edge computing for 5g-enabled tactile internet-based industrial applications," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 60–66, 2022. DOI: 10.1109/MCOM.001.21261.

[94] H. Wang, J. Ning, X. Huang, G. Wei, G. S. Poh, and X. Liu, "Secure fine-grained encrypted keyword search for e-healthcare cloud," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1307–1319, 2021. DOI: 10.1109/TDSC.2019.2916569.

[95] T. Liu, Y. Miao, K.-K. R. Choo, *et al.*, "Time-controlled hierarchical multi-keyword search over encrypted data in cloud-assisted iot," *IEEE Internet of Things Journal*, 2021. DOI: 10.1109/JIOT.2021.3126468.

[96] M. Wang, Y. Miao, Y. Guo, C. Wang, H. Huang, and X. Jia, "Attribute-based encrypted search for multi-owner and multi-user model," in *ICC 2021 - IEEE International Conference on Communications*, 2021. DOI: 10.1109/ICC42927.2021.9500525.

[97] B. Ji, Y. Wang, K. Song, *et al.*, "A survey of computational intelligence for 6g: Key technologies, applications and trends," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 7145–7154, 2021. DOI: 10.1109/TII.2021.3052531.

[98] European Parliament and Council of the European Union, "The general data protection regulation," legislation, 2016.

[99] S. Sciancalepore, G. Piro, D. Caldarola, G. Boggia, and G. Bianchi, "On the design of a decentralized and multiauthority access control scheme in federated and cloud-assisted cyber-physical systems," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5190–5204, 2018.

[100] A. Alkhulaifi and E.-S. M. El-Alfy, "Exploring lattice-based post-quantum signature for jwt authentication: Review and case study," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5. DOI: 10.1109/VTC2020-Spring48590.2020.9129505.

[101] D. Das, S. C. Sethuraman, and S. C. Satapathy, "A decentralized open web cryptographic standard," *Computers and Electrical Engineering*, vol. 99, p. 107 751, 2022, ISSN: 0045-7906. DOI: https://doi.org/10.1016/j.compeleceng.2022.107751. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S004579062200060X.

[102] A. K. Ranjan, V. Kumar, and M. Hussain, "Security analysis of tls authentication," in *2014 International conference on contemporary computing and informatics (IC3I)*, IEEE, 2014, pp. 1356–1360.

[103] A. Ferreira, R. Giustolisi, J.-L. Huynen, V. Koenig, and G. Lenzini, "Studies in socio-technical security analysis: Authentication of identities with tls certificates," in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2013, pp. 1553–1558. DOI: 10.1109/TrustCom.2013.190.

[104] J. Zhang, L. Yang, W. Cao, and Q. Wang, "Formal analysis of 5g eap-tls authentication protocol using proverif," *IEEE Access*, vol. 8, pp. 23 674–23 688, 2020. DOI: 10.1109/ACCESS.2020.2969474.

[105] I. Blake, G. Seroussi, G. Seroussi, and N. Smart, *Elliptic curves in cryptography*. Cambridge university press, 1999, vol. 265.

[106] R. C. Standard, "Pkcs# 1 v2. 1," *RSA Laboratories, Jun*, vol. 14, p. 61, 2002.

[107] D. Soni, K. Basu, M. Nabeel, N. Aaraj, M. Manzano, and R. Karri, "Crystals-dilithium," in *Hardware Architectures for Post-Quantum Digital Signature Schemes*, Springer, 2021, pp. 13–30.

[108] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe, "The sphincs+ signature framework," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 2129–2146.

[109] L. Nie, Y. Wu, X. Wang, *et al.*, "Intrusion detection for secure social internet of things based on collaborative edge computing: A generative adversarial network-based approach," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 1, pp. 134–145, 2022.

[110] H. Bangui and B. Buhnova, "Lightweight intrusion detection for edge computing networks using deep forest and bio-inspired algorithms," *Computers and Electrical Engineering*, vol. 100, p. 107 901, 2022.

[111] A. Singh, K. Chatterjee, and S. C. Satapathy, "An edge based hybrid intrusion detection framework for mobile edge computing," *Complex & Intelligent Systems*, vol. 8, no. 5, pp. 3719–3746, 2022.

[112] A. S. Almogren, "Intrusion detection in edge-of-things computing," *Journal of Parallel and Distributed Computing*, vol. 137, pp. 259–265, 2020.

[113] A. Mileva, A. Velinov, L. Hartmann, S. Wendzel, and W. Mazurczyk, "Comprehensive analysis of mqtt 5.0 susceptibility to network covert channels," *Computers & Security*, vol. 104, p. 102 207, 2021, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2021.102207.

[114] T. Lackner, J. Hermann, F. Dietrich, *et al.*, "Measurement and comparison of data rate and time delay of end-devices in licensed sub-6 ghz 5g standalone non-public networks," *Procedia CIRP*, vol. 107, pp. 1132–1137, 2022.

[115] M. Xu, Z. Fu, X. Ma, *et al.*, "From cloud to edge: A first look at public edge platforms," in *Proceedings of the 21st ACM Internet Measurement Conference*, 2021, pp. 37–53.

[116] Q. Lu, Z. Yang, H. Zhang, F. Chen, and H. Xian, "MRFE: A deep-learning-based multidimensional radio frequency fingerprinting enhancement approach for iot device identification," *IEEE Internet of Things Journal*, 2024.

[117] S. Al-Hazbi, A. Hussain, S. Sciancalepore, G. Oligeri, and P. Papadimitratos, "Radio frequency fingerprinting via deep learning: Challenges and opportunities," in *2024 International Wireless Communications and Mobile Computing (IWCMC)*, 2024.

[118] A. Jagannath, J. Jagannath, and P. S. P. V. Kumar, "A comprehensive survey on radio frequency (RF) fingerprinting: Traditional approaches, deep learning, and open challenges," *Computer Networks (Elsevier)*, 2022.

[119] N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar, "A Review of Radio Frequency Fingerprinting Techniques," *IEEE Journal of Radio Frequency Identification*, 2020.

[120] A. Al-Shawabka, F. Restuccia, S. D'Oro, *et al.*, "Exposing the fingerprint: Dissecting the impact of the wireless channel on radio fingerprinting," in *IEEE Conference on Computer Communications*, 2020.

[121] H. Fu, L. Peng, M. Liu, and A. Hu, "Deep learning-based rf fingerprint identification with channel effects mitigation," *IEEE Open Journal of the Communications Society*, 2023.

[122] S. Alhazbi, S. Sciancalepore, and G. Oligeri, "The Day-After-Tomorrow: On the performance of radio fingerprinting over time," in *Proc. of ACSAC*, 2023.

[123] S. Alhazbi, S. Sciancalepore, and G. Oligeri, "BloodHound: Early Detection and Identification of Jamming at the PHY-layer," in *IEEE Consum. Commun. & Network. Conf.*, 2023.

[124] S. Sciancalepore, F. Kusters, N. K. Abdelhadi, and G. Oligeri, "Jamming detection in low-ber mobile indoor scenarios via deep learning," *IEEE Internet of Things Journal*, 2024.

[125] M. Irfan, A. Omri, J. H. Fernandez, S. Sciancalepore, and G. Oligeri, “Jamming Detection in Power Line Communications Leveraging Deep Learning Techniques,” in *Int. Symp. on Networks, Computers and Communications (ISNCC)*, 2023.

[126] T. S. Rappaport, *Wireless communications: principles and practice*. Cambridge University Press, 2024.

[127] H. He, C.-K. Wen, S. Jin, and G. Y. Li, “Deep learning-based channel estimation for beamspace mmwave massive mimo systems,” *IEEE Wirel. Commun. Lett.*, 2018.

[128] A. P. Hermawan, R. R. Ginanjar, D.-S. Kim, and J.-M. Lee, “CNN-Based Automatic Modulation Classification for Beyond 5G Communications,” *IEEE Commun. Lett.*, 2020.

[129] J. Xie, C. Liu, Y.-C. Liang, and J. Fang, “Activity Pattern Aware Spectrum Sensing: A CNN-Based Deep Learning Approach,” *IEEE Commun. Lett.*, 2019.

[130] Y. Dong, Q. Liu, B. Du, and L. Zhang, “Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification,” *IEEE Transactions on Image Processing*, 2022.

[131] A. Younesi, M. Ansari, M. Fazli, A. Ejlali, M. Shafique, and J. Henkel, “A Comprehensive Survey of Convolutions in Deep Learning: Applications, Challenges, and Future Trends,” *IEEE Access*, 2024.

[132] “Exposing the Fingerprint: Dissecting the Impact of the Wireless Channel on Radio Fingerprinting,” in *IEEE INFOCOM*, Toronto, ON, Canada, 2020.

[133] B. Hamdaoui and A. Elmaghbub, “Deep-learning-based device fingerprinting for increased lora-iot security: Sensitivity to network deployment changes,” *IEEE Network*, 2022.

[134] G. Oligeri, S. Sciancalepore, S. Raponi, and R. D. Pietro, “PAST-AI: Physical-layer authentication of satellite transmitters via deep learning,” *IEEE Trans. on Inf. Forens. and Secur.*, 2023.

[135] L. Papangelo, M. Pistilli, S. Sciancalepore, G. Oligeri, G. Piro, and G. Boggia, “Adversarial Machine Learning for Image-Based Radio Frequency Fingerprinting: Attacks and Defenses,” *IEEE Commun. Magaz.*, 2024.

[136] S. H. et al., “Radio fingerprinting for anomaly detection using federated learning in lora-enabled industrial internet of things,” *Future Generation Computer Systems (Elsevier)*, 2023.

[137] Z. L. et al., “Non-inducible rf fingerprint hiding via feature perturbation,” in *IEEE International Conference on Communications*, 2023.

[138] L. F. Abanto-Leon, A. Bäuml, G. H. Sim, M. Hollick, and A. Asadi, “Stay Connected, Leave no Trace: Enhancing Security and Privacy in WiFi via Obfuscating Radiometric Fingerprints,” *Proc. of ACM on Measurem. and Analys. of Comput. Systs.*, 2020.

[139] I. Huso, et al., *Open Source Data of SDRs on Different Channels*, https://tinyurl.com/rxjv5wrc, Accessed: 23-May-2024.

[140] A. Sadighian, S. Sciancalepore, and G. Oligeri, “SatPrint: Satellite Link Fingerprinting,” in *ACM Symp. on Applied Comput.*, 2024.

[141] A. Sadighian, S. Sciancalepore, and G. Oligeri, “FadePrint: Satellite Spoofing Detection via Fading Fingerprinting,” in *IEEE Consum. Commun. & Netw. Conf.*, 2024.

[142] “ImageNet Large Scale Visual Recognition Challenge,” *International journal of computer vision*, 2015.

[143] Lime Microsystems, “LMS7002M – FPRF MIMO Transceiver IC,” Data Sheet.

[144] IEEE, “IEEE Standard for Low-Rate Wireless Networks,” *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, 2020.

[145] ZigBee Alliance, “Zigbee specification,” Specification, Aug. 2015, ZigBee Document – 05-3474-21.

[146] M. Irfan, S. Sciancalepore, and G. Oligeri, “On the Reliability of Radio Frequency Fingerprinting,” *arXiv preprint arXiv:2408.09179*, 2024.

[147] The European Commission, *Eurostat, Recorded offences by offence category - Police data*. The European Commission, 2021.

[148] C.-X. Wang, X. You, X. Gao, *et al.*, “On the road to 6g: Visions, requirements, key technologies, and testbeds,” *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 905–974, 2023.

[149] I. Palamà, F. Gringoli, G. Bianchi, and N. Blefari-Melazzi, “Imsi catchers in the wild: A real world 4g/5g assessment,” *Computer Networks*, vol. 194, p. 108 137, 2021, ISSN: 1389-1286.

[150] 3GPP, “3gpp release 15 description,” 3rd Generation Partnership Project (3GPP), Technical Report (TS) 33.126, Nov. 2022, Release 15.

[151] T. Isobe and R. Ito, “Security analysis of end-to-end encryption for zoom meetings,” *IEEE Access*, vol. 9, pp. 90 677–90 689, 2021.

[152] M. Vidoni, E. Senior Course, and F. Police, “5g technology: New challenges for law enforcement agencies to face,” *European Law Enforcement Research Bulletin*, vol. 22, pp. 157–171, Oct. 2022.

[153] Scientists4Crypto, “Academic letter to the european commission on "encryption — security through encryption and security despite encryption",” *Scientists4Crypto*, Dec. 2020.

[154] M. Säily, O. N. C. Yilmaz, D. S. Michalopoulos, E. Pérez, R. Keating, and J. Schaepperle, “Positioning technology trends and solutions toward 6g,” in *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2021.

[155] D. Giustiniano, G. Bianchi, A. Conti, S. Bartoletti, and N. B. Melazzi, “5g and beyond for contact tracing,” *IEEE Communications Magazine*, vol. 59, no. 9, pp. 36–41, 2021.

[156] V. Doronin, ““lawful interception – a market access barrier in the european union”?” *Computer Law & Security Review*, vol. 51, p. 105 867, 2023, ISSN: 0267-3649.

[157] M. Monshizadeh, V. Khatri, M. Varfan, and R. Kantola, “Liaas: Lawful interception as a service,” in *2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2018.

[158] G. Ungaro, F. Ricchitelli, I. Huso, G. Piro, and G. Boggia, “Design and implementation of a lawful interception architecture for b5g systems based on key escrow,” in *2022 IEEE Conference on Standards for Communications and Networking (CSCN)(CSCN'22)*, Thessaloniki, Greece, Nov. 1, 2022, published.

[159] 3GPP, “Lawful interception (li) requirements,” 3rd Generation Partnership Project (3GPP), Technical Report (TS) 33.126, Nov. 2022, Release 18.0.0.

[160] 3GPP, “Lawful interception (li) architecture and functions,” 3rd Generation Partnership Project (3GPP), Technical Report (TS) 33.127, Sep. 2023, Release 18.5.0.

[161] 3GPP, “Protocol and procedures for lawful interception (li),” 3rd Generation Partnership Project (3GPP), Technical Report (TS) 33.128, Sep. 2023, Release 18.5.0.

[162] WhatsApp Messenger, “Whatsapp encryption overview,” WhatsApp Messenger, Tech. Rep., Sep. 2023, Accessed: December 1, 2023. [Online]. Available: https://scontent-ams4-1.xx.fbcdn.net/v/t39.8562-6/383236184_722587863039320_504065106322868039 3_n.pdf?_nc_cat=101&ccb=1-7&_nc_sid=b8d81d&_nc_ohc=2mCqgHDwvkkAX-SZwA-&_nc_ht=scontent-ams4-1.xx&oh=00_AfDx5tswN8agfAV4lVpUamJBsr21Bn0qtbIch35wpRE8RQ&oe=659961C4.

[163] Telegram Messenger LLP, “Telegram api documentation,” Telegram Messenger LLP, Tech. Rep., 2023, Accessed: December 1, 2023. [Online]. Available: https://core.telegram.org/api/end-to-end.

[164] T.-H. Kim, W.-B. Kim, D. Seo, and I.-Y. Lee, “A secure encapsulation schemes based on key recovery system,” in *Silicon Valley Cybersecurity Conference: First Conference, SVCC 2020, San Jose, CA, USA, December 17–19, 2020, Revised Selected Papers 1*, Springer, 2021, pp. 25–37.

[165] H. Farlow and B. M. Edwards, “Shining a light on ‘going dark’: A framework to guide the co-design and communication of decryption laws based on the passage of the telecommunications and other legislation (assistance and access) bill 2018,” *Computer Law & Security Review*, vol. 46, p. 105 726, 2022, ISSN: 0267-3649.

[166] T. Riebe, P. Kühn, P. Imperatori, and C. Reuter, “Us security policy: The dual-use regulation of cryptography and its effects on surveillance,” *European Journal for Security Research*, vol. 7, no. 1, pp. 39–65, 2022.

[167] C. Duan and J. Grimmelmann, “Content moderation on end-to-end encrypted systems: A legal analysis,” *Available at SSRN 4457414*, 2023.

[168] K. Han, C. Y. Yeun, T. Shon, J. Park, and K. Kim, “A scalable and efficient key escrow model for lawful interception of IDBC-based secure communication,” *International Journal of Communication Systems*, vol. 24, no. 4, 2011.

[169] T. Kim, W. Kim, D. Seo, and I. Lee, “Secure encapsulation schemes using key recovery system in iomt environments,” *Sensors*, vol. 21, no. 10, 2021, ISSN: 1424-8220.

[170] K. Han, C. Y. Yeun, and K. Kim, “New key escrow model for the lawful interception in 3gpp,” in *2009 Digest of Technical Papers International Conference on Consumer Electronics*, 2009, pp. 1–2.

[171] A. D. Felice, “Encryption: Finding the balance between privacy, security and lawful data access,” DIGITALEUROPE, Position paper on Encryption Policy, 2020. [Online]. Available: https://cdn.digitaleurope.org/uploads/2020/03/DIGITALEUROPE-Position-on-Encryption-Policy-.pdf.

[172] I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography* (London Mathematical Society Lecture Note Series). Cambridge University Press, 1999.