MINISTÈRE DE L'E NSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ MOHAMED KHIDER-BISKRA

— FACULTÉ DES SCIENCES EXACTES ET DES SCIENCES DE LA NATURE ET DE LA VIE —
— DÉPARTEMENT D'INFORMATIQUE —

THÈSE

Présentée pour obtenir le diplôme de

DOCTORAT LMD

SPÉCIALITÉ : INFORMATIQUE

On Communication Privacy in the Internet of Things

Par
Asma Iman Kouachi

Soutenue le 20/05/2021, devant le jury composé de:

- Noureddine Djedi, Président, Professeur, Université de Biskra.
- Kamal Eddine Melkemi, Examinateur, Professeur, Université de Batna2.
- Mohamed Faouzi Zerarka, Examinateur, Maître de conférences, Université de Biskra.
- Abdelmalik Bachir, Directeur de thèse, Professeur, Université de Biskra.

بسم الله الرحمن الرحيم

(وَقُلْ رَبِّ زِدْنِي عِلْماً)

صدق الله العلي العظيم

سورة طه: من الآية114

Dedication

I was born for challenging anyone
who mocks to me

Asma Iman Kouachi

In the Name of ***Allah****, the Most Gracious, the Most Merciful*
All praise and gratefulness to Allah the Almighty and Peace and blessings be upon his prophet ***Mohammed****, the best teacher humanity has ever known.*

My PhD thesis is dedicated:

To my family, especially ***my parents Aissa Kouachi and Zineb Barket*** *who taught me that faith and patience are the keys of success.*

To my husband ***Mohamed Lamine Kelala*** *who helped me in this work.*

To my supervisor ***Prof Abdelmalik Bachir*** *who also taught me that patience is the most important thing during the PhD journey, he gave me much knowledge in many areas.*

To my friends whose have always been encouraged me and support me in the most difficult situations during my PhD journey.

To my university (UMK), my teachers

To my teacher at university of Setif ***Ph.D. Nabil Guellati*** *who has helped me in the practical side.*

Acknowledgement

I am thankful for all of those who said NO to me. It's because of them I'm doing it myself

Albert Einstein

In the Name of Allah, the Most Gracious, the Most Merciful First and foremost, I am faithfully grateful to Allah the Almighty who guided my steps and helped me consummate this modest work.

I would like to thank my parents **Aissa Kouachi and Zineb Barket** for their encouragement.

I would like to express my profound gratitude to my supervisor. **Prof Abdelmalik Bachir** for his guidance, assistance and advice throughout the accomplishment of this modest work.

I would like to thank the committee members: **Prof Noureddine Djedi, Prof Kamal Eddine Melkem and Dr Mohamed Faouzi Zerarka** for their acceptance to discuss my work.

I would like also to convey my heartfelt gratitude and deepest appreciation to my husband **Mohamed Lamine Kelala** for his persistent help, guidance and patience.

Abstract

We tackle the problem of privacy breaching in IPv6 Low power Wireless Personal Area Networks (6LoWPAN)-based Internet of Things (IoT) networks where an attacker may be able to identify the communicating entities.

We propose three contributions which are: (i) survey: we thoroughly expose the prime focus of the existing solutions on communication identifiers privacy in 6LoWPANs, clarifying the important information about: at which layer the solutions operate, based on which protocol, against which attack, for which application, based on simulations or real prototypes, which sensitive information or communication identifiers are protected, which Privacy-Preserving Technique (PPT) is used, and how long is the duration of the protection against privacy attacks. (ii) uOTA: based on the One Time Address (OTA) approach proposed for the traditional Internet, with a focus on low complexity, memory footprint, and energy consumption, uOTA uses just one IPv6 address to send or to receive one packet. (iii) ACFI which is based on: (1) anonymizing both IP and MAC addresses, as well as port number at the source host, using a random pseudonyming scheme, and (2) anonymizing the IP address and port number of the destination host, using a Tor-like network. We analysed the effect of the Tor entry node location on the performance of our solution in three different scenarios: the Tor entry node is located (a) inside the 6LoWPAN, (b) at the 6LBR gateway, or (c) completely outside the 6LoWPAN.

Using Cooja simulator, we showed that our solutions (uOTA and ACFI) outperformed state-of-the-art solutions by making it more difficult to identify communication flows by improving the anonymity and unlinkability of the communicating entities without significantly affecting energy consumption, communication delay, and network bandwidth.

Keywords: Internet of Things (IoT), Privacy, 6LoWPAN, Anonymity, Unlinkability.

Resumé

Nous abordons le problème de la violation de la vie privée dans les réseaux IoT basés sur 6LoWPAN où un attaquant peut être en mesure d'identifier les entités communicantes.

Nous proposons trois contributions qui sont: (i) un état de l'art approfondi: nous exposons en détail l'objectif principal des solutions existantes sur la confidentialité des identifiants de communication dans 6LoWPANs, en clarifiant les informations importantes sur: à quelle couche les solutions opèrent, en fonction de quel protocole, contre quelle attaque, pour quelle application, basée sur des simulations ou des prototypes réels, quelles informations sensibles ou identifiants de communication sont protégés, quelle technique de protection de la vie privée (PPT) est utilisée, et quelle est la durée de la protection contre les atteintes à la vie privée, (ii) uOTA: basé sur l'approche One Time Address (OTA) proposée pour l'Internet traditionnel, avec un accent sur la faible complexité, l'empreinte mémoire et la consommation d'énergie, uOTA utilise une seule adresse IPv6 pour envoyer ou recevoir un paquet, et (iii) ACFI : il est basé sur: (1) l'anonymisation des adresses IP et MAC, ainsi que le numéro de port de l'hôte source, en utilisant un schéma de pseudonyme aléatoire, et (2) l'anonymisation de l'adresse IP et du numéro de port de l'hôte de destination, en utilisant un réseau de type Tor. Nous avons analysé l'effet de l'emplacement du nœud d'entrée Tor sur les performances de notre solution dans trois scénarios différents: le nœud d'entrée Tor est situé (a) à l'intérieur du 6LoWPAN, (b) à la passerelle 6LBR, ou (c) complètement à l'extérieur du 6LoWPAN .

À l'aide du simulateur Cooja, nous avons montré que nos solutions (uOTA et ACFI) surpassent les solutions de pointe en rendant plus difficile l'identification des flux de communication en améliorant l'anonymisation et la dissociation des entités communicantes sans affecter de manière significative la consommation d'énergie, le délai de communication et la bande passante du réseau.

Mots clés: Internet des objets, Vie privée, 6LoWPAN, Anonymisation, Dissociation.

ملخص

نعالج مشكلة انتهاك الخصوصية في شبكة إنترنت الأشياء القائمة على 6LoWPAN. حيث قد يتمكن المهاجم من تحديد الكيانات المتصلة.

وعليه نقترح ثلاث مساهمات لحل هذه المشكلة و هي: (1) المسح: و هو الكشف بدقة عن الحلول الحالية المركزة على حماية خصوصية معرفات الاتصال في 6LoWPAN, و توضح المعلومات المهمة ما يلي: حول أي طبقة تعمل هذه الحلول, بناءا على أي بروتوكول, و ضد أي هجوم, على أي تطبيق, و استنادا على المحاكاة أو النماذج الأولية الحقيقية, ما هي المعلومات الحساسة أو معرفات الاتصال المحمية, و تقنية الحفاظ على الخصوصية المستخدمة (PPT), و مدة الحماية ضد هجمات الخصوصية.(2) uOTA: يعتمد على فكرة استعمال العنوان لمرة واحدة المقترحة للانترنيت التقليدية (OTA), مع التركيز على تخفيض التعقيد, و استهلاك الذاكرة و الطاقة, يستخدم uOTA عنوان IPv6 واحدا فقط لأرسال أو استلام حزمة واحدة. (3) ACFI: يعتمد على: (*) اخفاء هوية كل من عناوين IP و MAC, و كذلك رقم المنفذ في مضيف المصدر باستخدام نظام أسماء مستعارة عشوائية, و (**) اخفاء هوية عنوان IP ورقم منفذ مضيف الوجهة باستخدام شبكة تشبه Tor. حيث قمنا بتحليل تأثير موقع عقدة دخول Tor على أداء حلنا في ثلاثة سيناريوهات أين تقع عقدة دخول Tor: (أ) داخل 6LoWPAN, (ب) عند بوابة Tor, أو (ج) خارج 6LoWPAN.

عند استخدام برنامج محاكاة Cooja, أظهر لنا أن uOTA و ACFI التفوق على أحدث الحلول الموجودة, فمن خلال زيادة صعوبة تحديد تدفقات الاتصالات بواسطة تحسين إخفاء الهوية و الغاء ربط الكيانات المتصلة, مع عدم التأثير بشكل كبير على استهلاك الطاقة و تأخير الاتصال و عرض النطاق الترددي للشبكة.

الكلمات المفتاحية: إنترنت الأشياء, الخصوصية, 6LoWPAN, اخفاء الهوية, إلغاء الربط.

Journals, conferences, and posters

Don't limit your challenges,
challenge your limits

Jerry Dunn

Journals

- Asma Iman Kouachi, Abdelmalik Bachir, and Noureddine Lasla. Anonymizing communication flow identifiers in the internet of things. Computers & Electrical Engineering, 91:107063, Elsevier, 2021.

Conferences

- Asma Iman Kouachi and Abdelmalik Bachir. *Communication-Flow Privacy-Preservation in 6LoWPANs-Based IoT Networks*. In International Symposium on Modelling and Implementation of Complex Systems (pp. 33-47). Springer, Cham, 2020, October,

- Asma Iman Kouachi, Somia Sahraoui, and Abdelmalik Bachir. *OTFI: Communication privacy in the IoT based on One Time Flow Information*. In the 3rd International Symposium on Informatics and its Applications (ISIA 2018), M'sila (Algeria), November 6-7, 2018,

- Asma Iman Kouachi, Somia Sahraoui, and Abdelmalik Bachir. *Per Packet Flow Anonymization in 6LoWPAN IoT Networks*. In the 6th International Conf. on Wireless Networks and Mobile Communications (WINCOM'18), Marrakesh, Morocco, October 16-19, 2018. **Best Student Paper Award**

Posters

- Asma Iman Kouachi, Somia Sahraoui, and Abdelmalik Bachir. *Upllc-IoT: Ultra-Privacy and Low-Latency IoT-Communication*. In the Autumn School in the 7th IEEE International Conference on Smart Communications in Network Technologies, EL Oued, Algeria, October 27-28, 2018.

- Asma Iman Kouachi, Somia Sahraoui, and Abdelmalik Bachir. CFPe: *Communication Flow Privacy for E-Health in the Internet of Things*. In the 1st Ph.D Days, JDITA'2018, Biskra, Algeria, January 28-30, 2018.

- Asma Iman Kouachi and Abdelmalik Bachir: *Enhancing Communication Privacy in the IoT by OTCFI*. In the 3rd Scientific Days on Innovation, , Biskra, Algeria, October 29-30, 2019.

Contents

List of Figures

List of Tables

Acronyms

There are no secrets to success. It is the result of preparation, hard work, and learning from failure

Colin Powell

6LoWPAN *IPv6 Low power Wireless Personal Area Network*

IoT *Internet of Things*

PPTs *Privacy Preserving Techniques*

OTA *Micro One Time Addresses*

OTA *One Time Addresses*

ACFI *Anonymizing Communication Flow Identifiers*

TOR *The Onion Router*

LBR/6LBR *6LoWPAN Border Router*

OS *Operating System*

GPS *Global Policy and Strategy*

LoRaWAN *Long Range Wide Area Network*

OSI *Open Systems Interconnection*

SOA *Service Oriented Architecture*

WSN *Wireless Sensor Networks*

RFID *Radio Frequency IDentification*

IETF *Internet Engineering Task Force*

MTU *Maximum Transmission Unit*

BR *Border Router*

SCM *Supply Chain Management*

FSC *Food supply chain*

E-health *Electronic Healthcare*

VPN *Virtual Private Network*

PIR *Private Information Retrieval*

Mix-Net *Mix-Networks*

IID *Interface IDentifier*

CGA *Cryptography Generated Address*

SP *Service Provider*

TP *Third Party*

PIR *Private Information Retrieval*

OR *Onion Router*

DS *Directory Server*

AS *Autonomous System*

CR *Core Router*

AR *Access Router*

HID *Host IDentifier*

FID *Flow IDentifier*

ASN *Autonomous System Number*

MT6D *Moving Target IPv6 Defence*

CH *Cluster Head*

CM *Cluster Member*

BLE *Bluetooth Low Energy*

ND *Neighbor Discovery*

ICMP *Internet Control Message Protocol*

uMT6D *Micro Moving Target IPv6 Defence*

TW *Time Windows*

CSP *Communication Security and Privacy*

PID *Permanent ID*

SID *Session ID*

NAT *Network Address Translation*

APT *Advanced Persistent Threat*

IOR *IoT Onion Routers*

PeDAAC *Privacy Enabled Disjoint and Dynamic Address Auto-Configuration Protocol for 6LoWPAN*

DAD *Duplicate Address Detection protocol*

UID *Unique ID*

MQTT *Message Queue Telemetry Transport*

LPM *low power mode*

Rx *Receive Mode*

Tx *Transmit Mode*

RPL *Routing Protocol for Low-Power and Lossy Networks*

Chapter 1

General Introduction

Contents

Failure is simply the opportunity to begin again, this time more intelligently

Henry Ford

1.1 Context

Connection anytime and anywhere is a recent technology which is used in our daily applications, this technology is called the Internet of Things (IoT) which connects many things with each other over the Internet. The constant need for new services in our modern lifestyle coupled with the democratization of low power technology has led to the innovation of a large number of small devices with various capabilities including sensing, processing, information storage, as well as wireless communication. The need for providing Internet connectivity to these devices, also commonly called things, while taking full advantage of exiting Internet architecture and protocols, and facing the stringent constraints on resources and capabilities has led to the design of 6LoWPAN protocol suite, which provides an adaptation of the IPv6 protocol to resource-constrained things. This adaptation is mainly achieved through the operations of datagram compression and fragmentation to make typically long IP packets fit in the 127-Byte IEEE 802.15.4 maximum packet length [1].

1.2 The problem studied in the thesis

According to [2], there are five kinds of sensitive information in the IoT: identity, location, device type, time, and data. For each kind, there are specific privacy techniques. Thus, privacy is not only limited, for an attacker can discover the device's information like the device's name or its identifier [3]. This attack can also infer the device's place, with whom it communicates? the device's purpose of communication, and its service anytime and anywhere. Communication breached privacy problem is that a traffic analysis attack can infer that two entities are communicating with each other (with whom it is communicating?). Thus, one of the major privacy-preserving challenges is hiding that a communication is taking place between a source and a destination entity.

In a typical scenario, even if the data payload is encrypted, the packet header information related to the communication flow is generally transmitted in clear form for the packets to be correctly routed to the destination entity. Such information included in packet headers allows the full identification of the communication entities. A flow linking two communicating entities is usually defined by 5-tuple information: the source and destination addresses, the sources and destination port numbers, and the transport mode (TCP or UDP).

Communication breaching problem is centered around the persistence of the five flow information/communication identifiers. Communication identifiers privacy can be achieved when anonymity and unlinkability properties are achieved. The anonymity consists in hiding the original identities of entities (source anonymity and destination anonymity), and the unlinkability aims at breaking any pattern that would reveal the relationship between them [4] (source-destination unlinkability). In other words, if the flow information is anonymous and unlinkable, the communication will be anonymous and unlinkable automatically, and the traffic analysis attack can not link the communication entities (the source and the destination), and vice versa.

Preserving communication privacy in the IoT is harder than in IP communication in the Internet [5], because IoT-devices have many constraints on their limited-resources in energy and computing power (must use lightweight computation) for cryptography or others computations.

1.3 Contributions

In this thesis, we propose three contributions which are:

- State-of-the-art review: we collect the recent 6LoWPAN-based solutions for preserving communication identifiers privacy, and analyze them by extracting the most important information which are: (1) on which layer this solution works?, (2) what is the protocol that this solution is based on?, (3) what is the kind of attack that this solution is powerful against?, (4) to which application this solution applies?, (5) with which simulator that this solution was validated?, (6) what are the sensitive information/communication identifiers that this solution considers?, (7) what is the Privacy-Preserving Technique (PPT) in use?, (8) for how long does this solution keep privacy preserved?. The main goal of our survey is to help researchers to propose an efficient 6LoWPAN-based solution in the future for achieving communication identifiers privacy,

- uOTA: as it can be seen in the related work chapter, little has been done to protect IoT things from flow-identification attacks particularly for the promising TCP/IP-compatible 6LoWPAN architecture [2]. In this scheme, we propose a solution to cope with this problem and provide host-flow unlinkability to make it difficult for attackers to determine what things are communicating with. Our solution called micro OTA (uOTA) is inspired from the OTA approach proposed for the traditional Internet but with adaptations and optimization for 6LoWPAN architecture. The main contributions of uOTA can be summarized as:
  - Design of a flow anonymization solution that is compatible with 6LoWPAN networks to cope with the flow-identification problem in the IoT. We also provide a detailed description of the proposed solution,
  - A concrete implementation on Cooja simulator to validate the proposed solution on emulated motes, as well as a comparison with existing solution for flow anonymity in 6LoWPAN networks,
  - A communication flow privacy metric that measures the levels of flow identification at different locations in the global internet.
- ACFI: we consider a global solution that takes into account anonymizing both source and destination addresses and port numbers as well as source MAC addresses to make it difficult for attackers to de-anonymize sources or destinations or find relationships between them. The main contributions of ACFI are as follows:
  - Achieving communication-flow privacy by ensuring source and destination anonymity and unlinkability,
  - Anonymizing MAC, IP and port numbers at the source host by using random pseudonyms,
  - Anonymizing IP addresses and port numbers at the destination host by using Tor-like network,
  - Studying the effect of Tor-network choices on energy, consumption, end-to-end latency, and network bandwidth.

1.4 Thesis Structure

This thesis is divided into two parts which are:

- Part I: State of the Art. In this part we have three chapters as follows:
  - Chapter 2: Internet of Things Essentials, we present the most important information about the IoT that we need them in the next chapters,
  - Chapter 3: Privacy Essentials, we present a detailed point on privacy issue that we help us to understand the next chapters,

- Chapter 4: Related Work. We provide an extensive review of the proposed solutions related to a communication flow privacy in the IoT.

,

- Part II: Contributions, in this part we explain our proposals to solve the communication privacy problem in the IoT followed by their evaluation and results, this part contains:
  - Chapter 5: our first solution uOTA
  - Chapter 6: our second solution ACFI
  - Chapter 7: evaluation and results, this chapter shows evaluation metrics and the results obtained with our solutions (uOTA and ACFI) compared to the state-of-the-art solutions.

Finally, conclusion marks and future directions are given in Chapter 8.

Part I

State of The Art

Chapter 2

Internet of Things (IoT) Essentials

> Never give up on what you really want to do. The person with big dreams is more powerful than the one with all the facts.
>
> *Albert Einstein*

Contents

2.1 Introduction

In 1998, the concept of IoT was created by Kevin Ashton [6, 7], and introducing the term IoT was in 1999 by Kevin Ashton in Procter & Gamble company (P&G) [8, 9, 10, 11, 12, 13].

The basic idea of IoT technology is to enable the interaction of many devices (things) with each other for users' life. The IoT thing means an embedded computing device that can send and receive information in the network. An embedded system is based on micro-controllers with small memory utilization [14]. According to the website of Statista [15] as shown in Fig. 2.1 (from 2015 to 2025), the number of IoT-devices in the world will be increased from 30.7 billion in 2020 to 75.44 billion in 2025. IoT devices have some limitations as memory, computation, energy, etc [16].

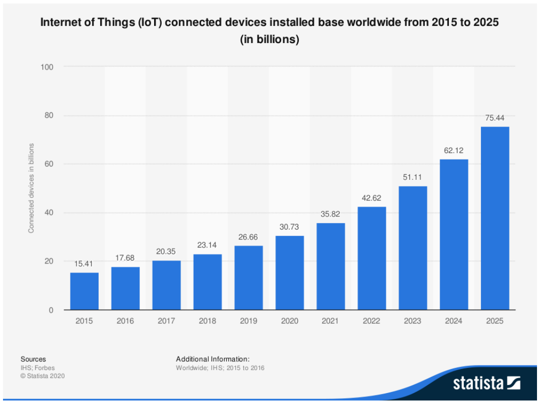


Figure 2.1: Number of IoT-devices from 2015 to 2025 in the world.
[15]

IoT has big researches in academic and industrial areas [17]. In this chapter we explain IoT essentials: in Section 2.2, we give some definitions of IoT, Section 2.3 discuss the six important components in the IoT. We propose a taxonomy to classify the IoT architecture according to four properties in Section 2.4. We explain more 6LoWPAN in Section 2.5. IoT-characteristics and IoT-applications are discuss in Sections 2.6, and 2.7, respectively followed by IoT-benefits in Section 2.8. We highlight some IoT-challenges in Section 2.9. We need Operating Systems (OSs) and simulators to apply proposed solutions. Thus, we discuss the most popular IoT-OSs in Section 2.10. Finally, we summarize this chapter in Section 2.11.

2.2 IoT-Definitions

There are numerous IoT definitions, we select some from them as follows:

- IoT is the connection between networks connected over the Internet [18],
- Things of users are connected over the Internet [19],
- IoT is divided into two words which are (i) Internet, i.e. network, and (ii) things, i.e. objects. IoT is a huge network of connected objects over the Internet [6],
- IoT is a network of connected sensors, actuators, vehicles, and other objects over the internet [17]. According to [20, 21, 22], a sensor is a device which aims to measure or to sense physical information like blood sugar levels, and sends the results to another entity. Example on sensors like temperature sensor, humidity sensor, light sensor. Actuator: aims to transfer the input information (results of sensor) into action, i.e. it focuses on doing the action according to the results of the sensor like switching between turn on/turn off the air conditioner. Example on actuator like pneumatic actuator, thermal actuator [23].
- IoT is everything communicates with everything, everyone in any-where, at any-time with any-path over the Internet to do any service. IoT is the next generation of the Internet [24, 25],
- As shown in Fig. 2.2, IoT is the combination of six Any [26].

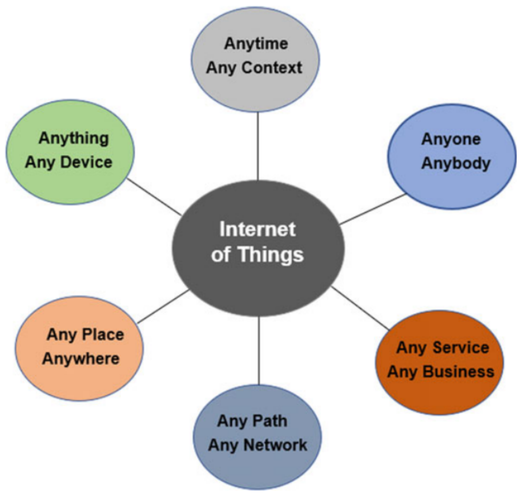


Figure 2.2: Six Any in IoT definition
[26].

2.3 IoT-Components

According to [27, 28, 29] IoT is a combination between six components as you can see in Fig. 2.3:

- Devices: are any physical objects like tablet, TV, car, etc,

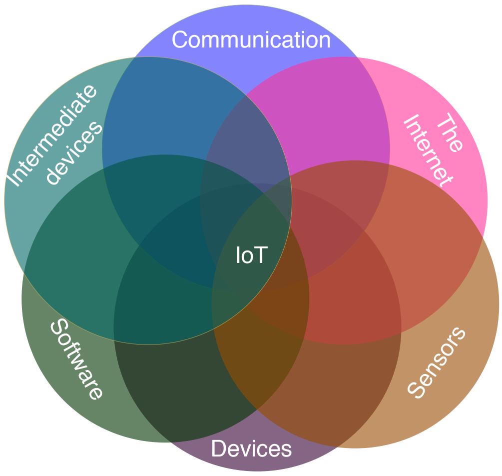


Figure 2.3: IoT-Components.

- Sensors: for capturing, collecting and transforming physical information. According to [30], there are three types of sensors which are (i) motion sensor like accelerometer sensor, (ii) environmental sensor like temperature sensor, and (iii) position sensor like Global Policy and Strategy (GPS) sensor, etc,
- Software: IoT devices provide many services to others which must own a software to access them, and to analyze information too,
- Internet: is the bridge of communication between everything (e.g. human, machines, animals, etc) in the world,
- Intermediate devices: are transmitter devices for transmitting information, they are devices between sender device and receiver device,
- Communication channel: transmitted information from sender device to receiver device is used by communication protocols over channel.

2.4 IoT-Architectures

According to our study, we classify IoT-architectures to four types as follows:

2.4.1 IoT-Architecture Based on Infrastructure

In [31], there are three kinds of the IoT architecture according to edge computing, fog computing, and cloud computing technologies, we select cloud computing to explain the IoT-architecture based on infrastructure. According to [8], this architecture is illustrated in Fig. 2.4.

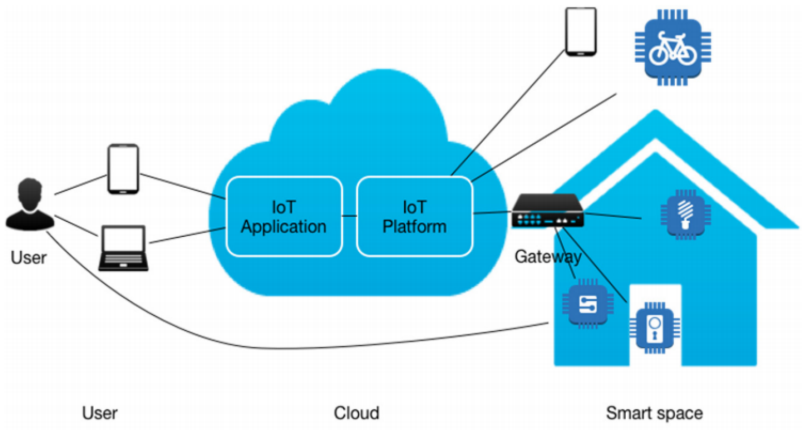


Figure 2.4: IoT-Architecture Based on Infrastructure.
[8].

As shown in Fig. 2.4, IoT-architecture based on infrastructure has three parts which are:

- Smart space: contains devices. According to the ability of devices, these devices (i) can connect with the Internet directly by sending information to IoT platforms, i.e. is a software in the cloud for integrating devices. Its goal is to search for sensor, service, to access to information, etc, (ii) or over gateways by using:
  - Wired communication (e.g. Ethernet),
  - Wireless communication, it can be:
    * Mobile network (e.g. 4G),
    * The gateway is near to smart space (e.g. Bluetooth),
    * The gateway is far to smart space (e.g. Long Range Wide Area Network (LoRaWAN)).
- Cloud: includes IoT platform and IoT services such as monitoring data, management data, composition data, processing data, etc,
- User: meaning the final application by using user's phone or tablet.

2.4.2 IoT-Architecture Based on Formation Model

According to [25, 10, 6, 32, 33, 30, 34, 35, 36, 37, 24, 12, 38, 39, 24]. In this kind of architecture, there are papers that said there are four layers, others said there are five layers, and others said six layers. So, we explain below the six layers (see Fig. 2.5) in this kind of architecture:

- Coding layer: aims to identify devices in the IoT system (unique ID for each device),

- Perception layer: or edge technology/hardware/objects/sensing layer. This layer is like physical layer in the traditional model Open Systems Interconnection (OSI), Its goals are collecting the physical information that sensed from sensors, and conversion the physical information into digital information for sending them to the network layer,

- Network layer: it is also named access gateway layer. (i) Connecting all devices together, (ii) addressing, (iii) routing, (iv) and transmitting information from perception layer to middleware layer are done at this layer,

- Middleware layer: it is called service management or service layer too. This layer is an interface between hardware and application layers, its goals are providing various needed services, managing devices and data, and it can aggregate and filter data too, etc. The most popular services in this layer are (i) service discovery, i.e. finding the device which satisfies users needs, and (ii) service composition,

- Application layer: or interface layer, it is a software for using the service by users,

- Business layer: its goal is managing the IoT system by creating diagrams, graphs, flow charts, curbs, etc according to the reached data from the application layer to analyze the IoT system.

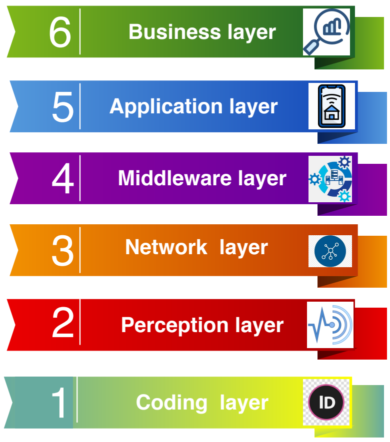


Figure 2.5: IoT-Architecture Based on Formation Model.

2.4.3 IoT-Architecture Based on Communication Model

In [25], this type of architecture includes:

- Device to Device communication: for exchanging information between IoT devices, there is not any intermediate node. Zigbee or Bluetooth protocols can be used in this communication.

- Device to cloud communication model: all services are in the cloud, device communicates the provider of the cloud service directly,

- Device to gateway communication model: there is an intermediate node, i.e. gateway between device and the cloud in order to access to services,

2.4.4 IoT-Architecture Based on Services

According to [6, 40, 28, 41, 42], this architecture is according to the perspective of functionalities. So, the Service Oriented Architecture (SOA) contains five layers as shown in Fig. 2.6, which are:

- Objects layer: is called hardware layer too, IoT includes many building technologies such as Wireless Sensor Network (WSN) and Radio Frequency IDentification (RFID). This layer is a set of devices in the network to collect information about things,
- Object abstraction layer: it is one level from the three middleware levels architecture, setting a common language and functions to access to various devices at this adaptation layer,
- Service management layer: services have to be available to objects (like service discovery),
- Service composition layer: composing many services to get a single one to realize a specific task,
- Applications layer: interaction between the user and the system.

As shown in Fig. 2.6, the architecture of middleware itself includes three layers which are: (a) object abstraction, (b) service management, and (c) service composition.

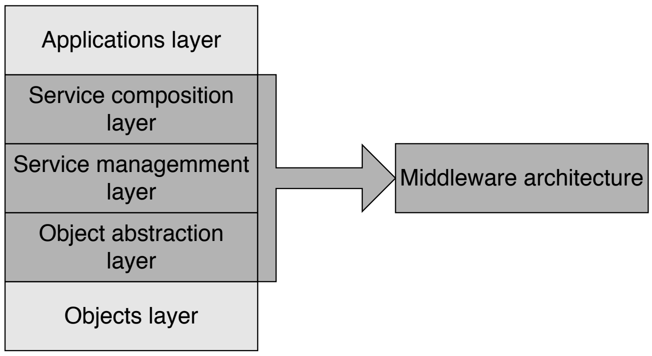


Figure 2.6: IoT-Architecture Based on SOA.

2.5 IoT-6LoWPAN

The growth of objects in the world and with the birth of IoT leads to reduction of IPv4 addresses (address space <number of objects), one solution is to use IPv6 addresses. In our study the

type of object is a sensor that produces the use of an adaptation layer (6LoWPAN), sensors in WSN connected to each other over low power protocols such as IEEE 802.15.4 (250 kb/s). 6LoWPANs are characterized by: small packet sizes, low bandwidth, battery supplied devices, low cost, large number of devices, unknown node positions, and long idle periods during when communications interfaces are turned off to save energy.

2.5.1 6LoWPAN Layer

IPv6 Low Power Wireless Personal Area Networks. 6LoWPAN is defined by Internet Engineering Task Force (IETF), in RFC 6282, 6LoWPAN is an adaptation of IPv6 to constrained objects that have limited battery, processing, memory, and storage resources. The main role of this layer is to make IPv6 addresses IPv6 adaptable to IEEE 802.15.4 frames. This requirement is challenging because there is a difference between both technologies, particularly the difference between the Maximum Transmission Unit (MTU) size for IPv6 packets which about 1280 bytes and MTU size of IEEE 802.15.4 packet which is about 127 bytes [1, 28, 5]. This layer is laying between the network and data link layers as shown in Fig. 2.7.

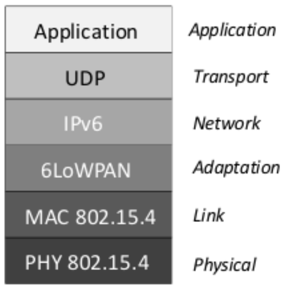


Figure 2.7: 6LoWPAN Protocols.
[5]

The main role of the 6LoWPAN layer is how to make address IPv6 adaptable to IEEE 802.15.4 protocols, by compression and fragmentation, such that the maximum size of a frame is 127 bytes and the minimum size of an IPv6 packet is 1280 bytes [5].

2.5.2 6LoWPAN Architecture:

The 6LoWPAN network contains a set of devices and routers connected to the Internet by the 6LBR like a gateway. There are three types of 6LoWPAN [43]: Ad hoc LoWPAN, i.e. without infrastructure, simple LoWPAN, i.e. a single Border Router and extended LoWPAN with multiple Border Routers (see Fig. 2.8). In this thesis, we focus on a simple 6LoWPAN, its architecture is illustrated in Fig.2.9.

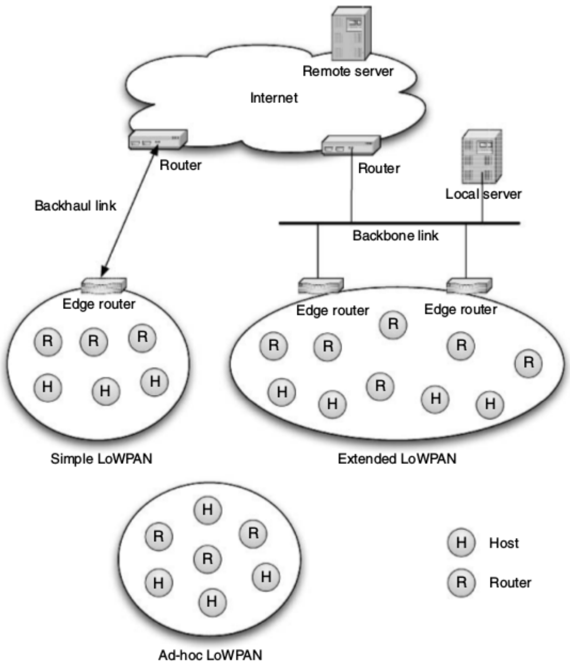


Figure 2.8: 6LoWPAN Architectures.
[43]

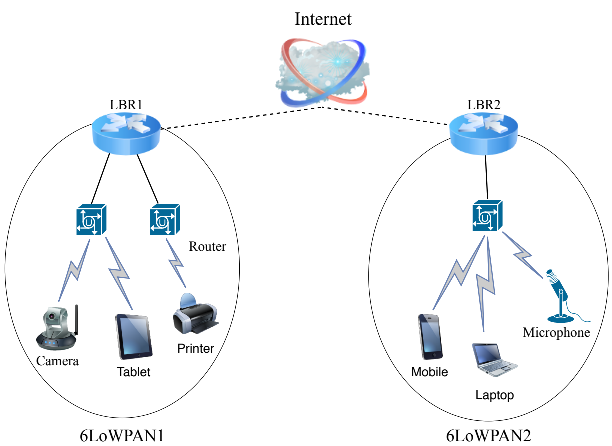


Figure 2.9: Simple 6LoWPAN Architecture.

2.6 IoT-Characteristics

According to [26, 44], the characteristics of the IoT are:

- Large scale: as we have said shortly before in the introduction section that the IoT devices are increasing during the time, i.e. they are in billions devices,
- Intelligence: integration of new software and algorithms into the hardware makes the IoT devices smart, these smart IoT devices work intelligently,
- Sensing: sensor is a part of the IoT. The sensor senses the physical information to send it to another IoT devices,
- Complex system: IoT devices, applications and operating systems in the IoT are heterogeneous in term of capabilities, so the organization between all these difference is very difficult,
- Dynamic environment: the user can turn-on/turn-off/update, etc any IoT devices dynamically,
- Massive amount of data: each sensor sends its information to the cloud or even to the fog node, so a huge information have to analyze, process, etc;
- Heterogeneity: in terms of hardware, applications, OSs, platforms, communication protocols, etc.
- Limited energy: IoT devices have constrained limitations in term of processing, computation, memory and energy,
- Connectivity: any IoT device can connect with other different IoT devices in order to create new applications and services,
- Self-configuring: IoT device has the ability to configure, update, and work itself, i.e. there is not any human intervention,
- Unique Identity: each IoT device has its unique identifier as IP address,
- Context awareness: sensors in the IoT system can make decisions according to the sensed information, i.e. so the sensor is aware of the context.

2.7 IoT-Applications

According to [35, 24, 25, 33, 10, 22, 42, 21] there are numerous IoT applications as follows:

- Electronic commerce and business transactions [19],

- Automation home or smart home, smart offices, and smart cities: we find four applications in the smart home which are: (a) energy, (b) security, (c) healthcare, and (d) entertainment. According to the architecture of smart home in [45], there are three levels which are: (i) devices, i.e. hardware as sensors, actuators, gateway, Bluetooth hub (beacon) [37, 46] and smart objects (device contains sensor and actuator like smart locks), (ii) communication, i.e. communication protocols inside the 6LoWPAN network for sensors communicate with other sensors or with their gateway like Zigbee, and network communication protocols like Bluetooth to connect with the Internet, and (iii) services which are software applications [45]. In smart cities, we control noise, traffic, fire, energy consumption, etc [12],

- Recycling: use of electronic waste, and collecting recyclable materials, etc by IoT devices [6],

- Media and entertainment: sharing new news according to users' locations [6],

- Security and thefts: meaning alarms and surveillance at home and work [8, 38, 12],

- Agriculture, smart farming, and breeding: monitoring animals, detecting contagious disease between animals, minimizing the number of farmers in a wide area by IoT applications, and smart irrigation too [6], smart greenhouses, livestock monitoring, and using agricultural drones in wide-area places [44, 38, 8, 12, 46],

- Transportation and assisted driving: meaning smart transport, i.e. monitoring traffic, screening and protecting privacy passengers and their luggage in airports, facilitating to track airline operations, finding optimized route, planing route, transport security, emergency, smart parking, etc [6, 17, 41, 8],

- Environment monitoring: using IoT devices in nature for preserving the environment like water shortage detection [12, 6, 17],

- Process: to equip gas, oil, and petrol containers by sensors to monitor their safety, their storage, etc [6],

- Social networking: sharing information and news about the assisted people and places, etc, [38]

- Manufacturing: monitoring products from production step to disposal step [6],

- Retail and Supply Chain Management (SCM): monitoring stocks constantly, tracking in-of-stocks and out-of-stocks, checking receipt, and recognizing of shoplifting, etc [6, 47, 46],

- Pharmaceutical: providing safety of pharmaceutical products, informing patients about expiration date of medicine, and reminding them to take their medicine too, etc [6],

- Independent living: for monitoring users' activities by wearable and ambient sensors [6],

- Automotive industry: making smart cars, smart buses, smart trains, and smart bicycles [6, 8],

- Aerospace and aviation: protecting security and preserving privacy in the aviation industry [6],

- Food Supply Chain (FSC) : means the farm-to-plate. So, it is this chain of agriculture to production to processing to storage to distribution to consumption. The goal of IoT in this application is safety, efficiency, and sustainability of FSC [41, 46],

- Mine safety: tracking and controlling the communication between surface and underground in underground mines works for preventing accidents [41],

- Firefighting: detecting fire and providing warnings to workers and people for vacating the fireplace [41],

- Energy: applications that manage or control the energy consumption in any other applications, for example controlling the energy consumption in smart home applications [17, 21],

- Prediction: prediction about the natural calamities, e.g. hurricane, earthquake, volcanic eruptions to propose possible solutions before they occur [12],

- Education: in [9], [38], IoT uses also in school education, medical education, medical training, vocational education, and vocational training,

- Medical and Healthcare: among the IoT applications that will be dominant in the world is the Electronic Healthcare application (E-Health). E-Health system is based on the collection of patients' personal information and vital information (blood pressure, blood sugar, humidity, temperature, heart rate, pulse rate, brain activity and so on) by special E-Health sensors. After the collection step, these information are sent to the cloud via the Internet in order to store, secure and manage them constantly [48]. The patient is in a hospital or in his room, or a worker which is in a hazardous place (underground mine), and there are sensors which are in his room, or they are wearable. Attaching sensors to patients in order to capture vital signs (temperature of body, blood pressure, pulse, etc) and biometric or physical information (fingerprint, voice, etc). Sensors capture information of patient's room and patient's information itself to manage and diagnostic the illness. The main benefit for healthcare in the IoT is resolution of patients problems will be solved rapidly. Healthcare uses IoT to monitor the healthcare equipment too (refilling/calibrating), and to alert medicines and nurses in special situations [18, 49, 38, 6, 17, 12, 41, 8, 46].

2.8 IoT-Benefits

- Improving data collection, customer interactions [25],

- Efficiency: IoT manipulates, processes, shares, computes, and analyzes data, that makes the IoT system more efficient [44],
- Transparency: IoT makes the IoT devices connect at anytime, and anywhere to provide services, which means transparency property [44],
- Monitoring, controlling and automation: controlling and monitoring the IoT devices, software without human intervention [44],
- Safety and comfort: aims to provide users' safety and their comfort [44],
- Time: reducing the time with IoT [44],
- Information: user can access to huge information to take his decisions [44],
- Security: enhancing security at home/office by using camera sensor for example[44]
- Cost/money: saving a large amount of money, reducing errors, operations-efficient, etc [44, 25],
- Accuracy: reducing making errors, and downtime,
- Improvement patients care [18, 49], and business process [41],
- Minimize resources [18],
- Users are comfortable [18],
- Enhancing user's quality life [45],
- Dynamic and self adaptation, e.g. the air conditioner at a smart home can change its condition (adding or reducing temperature) dynamically [29],
- Self configuration, i.e. a device in the IoT system can work with other IoT device to do a specific service, and these IoT devices can configure the network and software of the system without human help [29],
- Reduction users' efforts [29].

2.9 IoT-Challenges

There are many issues in the IoT-technologies that researchers must overcome. These issues cause many challenges. The important challenges are:

- Security: all IoT-applications inside data and hardware should be secure against attackers, especially e-health application [18, 19, 50, 42, 25, 41, 8, 22, 12, 28, 27, 45, 49, 51, 17, 24, 10, 21, 52, 25, 6, 41],

- Privacy: is an important issue in IoT for all applications particularly sensitive applications like e-health applications or military applications, where an attacker can infer all sensitive personal, medical and vital information about a patient, it impacts the patient's life negatively [28, 45, 49, 51, 17, 24, 21, 52]. In [52], all IoT applications need to preserve privacy, but the sensitive applications like healthcare applications that have to preserve more patients' privacy, and their communications too. The privacy can be a part of security in some papers like in [53, 18, 41, 19, 42, 8, 10, 12],

- Trust: the information transmitted by sensors have to be correct, attackers cannot corrupt this information to achieve trust [18],

- Technical challenges[41]: in term of (i) creation a new service with taking account IoT devices limitations, (ii) scalability due to the huge number of devices causes a problem in processing and management information, and in routing information too, (iii) IoT is a communication between heterogeneous networks, so there are not a common communication protocol, a common technology used, a common platform, etc, and (vi) integration IoT with the existing traditional applications and systems,

- Safety of IoT-devices [12],

- Architecture design: to provide reliable communication from device to device [21],

- Standardization: means that devices from different countries in the world have the possibility to exchange information in a specific IoT application [41, 21, 12],

- Scalability: when the number of IoT devices that are connected to the cloud is increased, here there is a problem of the capacity of the cloud (all devices send and receive information over the cloud). There are some solutions to solve that like 5G, i.e. by increasing the bandwidth, edge and fog computing solutions too, the IoT system has to update due to these proposed solutions [8],

- Identification: how we get a unique identifier for each device in our IoT system without any redundancy [6, 41, 12],

- Architecture: focuses on SOA architecture [6, 41].

- Communication: at communication modality (single-hop/multi-hop communication), infrastructure, network topology, connectivity (continuous/sporadic), network size, lifetime, etc [6, 41],

- Network: means technologies for the IoT like RFID, WSN, etc for achieving communication connectivity [6, 41],

- Software and algorithms: proposing algorithms and applications in the IoT system [6], [41],

- Hardware: means all things related to hardware devices [6], [41],

- Data and signal processing: is a mechanism at middleware level to process the captured information by sensor [6], [41],
- Discovery: the device provides a specific service, and this service has to discover from others devices, so we need to service discovery in IoT system [6], [41],
- Relationship network management: includes security, reliability, and performance [6], [41],
- Power and storage: aims to minimize the energy consumption and storage due to IoT devices limitations [6], [41],
- Standardization: standardization at communication protocol, interface, platform, etc [6], [41],
- Service management: allow creating a new service to the user's application [6], [41].

2.10 IoT-Operating Systems (OSs)

According to [46], we summarize the IoT-OS in the table below Table. 2.1. In [25], there are others simulators for the IoT which are: iFogSiM, Cloud2Sim, IoTSim, SimpleIoTSimulator, MBTASS, MobIoTSim, Arduino Unit, IoTFIY, and MAMMotH.

Operating System (OS)	Year of publication	Open source	Programming Languages	Simulators	Documentations
Contiki	2004	Yes	C	Cooja	Yes
TinyOS	2000	Yes	Nes C	TOSSIM	Yes
RIOT	2013	Yes	C, C++	Cooja,IoT-LAB	Not available
Nano_RK	2005	Yes	C	AVR studio	Not available
LiteOS	2008	Yes	LiteC++	Avorara	Yes
MantisOS	2005	Yes	C	Avorara	Yes
SOSOS	2005	Yes	C	Java SOS	No
RETOS	2007	Yes	C	RMtool	No

Table 2.1: Operating Systems in the IoT.

2.11 Conclusion

The increased deployment of tiny devices in all daily applications over the Internet introduced the IoT technology. We discussed essential information on the IoT that we need in this thesis.

IoT technology makes a controversial subject on security and privacy, they are the most prominent issues for the majority of IoT applications. In this thesis, we focus on privacy issues. How to preserve communication privacy in the IoT is still a sensitive problem, especially in sensitive applications such as e-health applications. We are going to explain essential information on privacy in the next chapter.

Chapter 3

Privacy Essentials

Success is no accident. It is hard work, perseverance, learning, studying, sacrifice and most of all, love of what you are doing or learning to do

Pele

Contents

3.1 Introduction

Privacy is one of the most prominent issues for the majority of IoT applications. How to preserve privacy is still a sensitive problem, especially in sensitive applications such as e-health.

In this chapter, we outline the essentials of information on privacy: Section 3.2 gives the most common definitions of the privacy term. We propose a taxonomy of privacy-kinds through the aggregation of the existing taxonomies in Section 3.3. The most important sections in this chapter are: (i) Section 3.4 which explains the privacy threats, (ii) Section 3.5 that discuss the major privacy properties in the Internet and in the IoT, (iii) Privacy-preserving techniques are explained in Section 3.6, (iv) anonymous communication shows in the IoT in Section 3.7, and finally (v) the best anonymous solution means which metrics that have to find in any anonymous solution in Section3.8. There is a difference between security and privacy, we summarize the most different points in Section 3.9. At the end, Section 3.10 gives a conclusion of this chapter.

3.2 Privacy-Definitions

There are many definitions of privacy, and they are different according to which they are where it is used. In computer science, privacy-definitions are as follows:

- According to [54], the user has the right to control his personal information as to his name, his address, his age, his job, etc, and to prevent the disclosure of unauthorized people to know his information,
- In [3], meta-data contains communication information as communication entities, time of sending/receiving, location, etc. Privacy does not achieve when meta-data is clear even if data is encrypted to protect the confidentiality, i.e. protecting security. Thus, to achieve communication privacy, hiding meta-data has to do,
- According to [55, 56], privacy is the user's right to be alone, there is not anyone who can watch or disturb him,
- The user has the ability to select when, how, and what his personal information that are sharing with others. In addition, the user has to control his personal information too, i.e. select who can access to his personal information and who can not [55, 3, 26, 57, 58, 56],
- In [59], privacy is the unwillingness to know any personal and sensitive information, people can say this sentence "It's none of your business," for referring to preserve their privacy,
- According to [60], to achieve privacy, you have to hide the sensitive header information like source IPv6 address, destination IPv6 address, source port number, destination port number, and transport mode as well as you can,
- In [45], communication privacy is preserving confidentiality when information is transmitted.

3.3 Privacy-Kinds Taxonomy

According to the literature, there are many privacy-taxonomies. In this section, we try to aggregate these taxonomies in a new taxonomy. In our new taxonomy, there are five metrics: (1) activities, (2) levels, (3) time, (4) information-centric or not, and (5) quality of preserving privacy. Fig. 3.1 illustrates our new taxonomy according to the five metrics, we discuss our new taxonomy as follows:

- Privacy-based on activities: in [61], the authors provided a new taxonomy of privacy according to four activities to understand privacy's problem:
  - Collection: many entities can collect the information of the user, for example, vital information in e-health applications,
  - Processing: owners of data collection are called also data holders, they process the data collection by storing, manipulating, and using it,
  - Dissemination: data holders transfer information processing to others, like health's ministry,
  - Invasion: in the three previous activities, an attacker can breach the user's privacy remotely, but here, he can invade the user directly.
- Privacy-based on levels: according to [62, 26], privacy is related to:
  - Body: protection of the privacy of users in physical aspects from harm, i.e. put caps and dress gloves,
  - Communication: prevent any third entity to know that two entities are communicating by hiding meta-data, i.e. communication between workers and doctor,
  - Territory: build limits on a specific place like a hospital, place of workers like a stadium,
  - Information: preserve personal information like vital information of workers [62].
- Privacy-based on time: according to [57], there are four kinds of privacy according to data phase which are:
  - Privacy at collection time: the smart things collect any information about users or about their information, at this time, the privacy can be breakthrough,
  - Privacy at transmission time: privacy during the communication: it is the security of communication protocols. Here, the attacker can analyze packets linking, for example, [24]
  - Privacy at storage time: the device stores data has to be private, and the stored data have to be anonymized,
  - Privacy at processing time: protecting data against the third party at processing step [24],

- Privacy-based on information: this kind focuses on which information will be breakthrough. So, there are:
  - Network-centric information: aims to interest in meta-data information, e.g. transport mode [63],
  - User-centric information: focuses on user's information as to his name, his age, his job, etc [63],
  - Device-centric information: the attacker can change the destination device according to the device's information like its ID, its location, etc to breakthrough the device privacy [63, 24],
  - Application-centric information: here the goal of an attacker is to break through the used application by disabled its services for example [55].
- Privacy-based on quality: according to [56], there are two types of privacy:
  - Hard privacy: its goal is data minimization, i.e. providing as little personal data as possible to entities, to preserve privacy effectively,
  - Soft privacy: access control, policies, and lows of privacy. Its goal is to provide specific properties on personal data, data store and data process [56].

We clarify which kind of privacy that we are working on. According to the five metrics of kinds of taxonomy, we have:

- Privacy-based on activities: dissemination,
- Privacy-based on levels: communication,
- Privacy-based on time: privacy at transmission time,
- Privacy-based on information: network-centric information,
- Privacy-based on quality: hard privacy.

3.4 Privacy-Threats Taxonomy

We divide the privacy-threats into two classes which are:

- Privacy-threats based on quality type: the authors of [4] proposed seven threats privacy **LINDDUN** acronym means: (L) **L**inkability, (I) **I**dentifiability, (N) **N**on-repudiation, (D) **D**etectability, (D) Information **D**isclosure, (U) Content **U**nawareness, (N) Policy/Consent **N**oncompliance.

  We have said shortly before that there are two kinds of privacy on quality which are (i) hard privacy, and (ii) soft privacy. Thus, there are also two kinds of privacy-threats according to this type which are: (a) privacy-threats based on hard privacy and (b) privacy-threats for soft privacy. We discuss them as follows:

Figure 3.1: Privacy-Kinds Taxonomy.

- Privacy-threats in hard privacy:
  * Linkability: an attacker can know the relationship between two or more Items of Interest (IOIs). IOI can be entities, data flow, data store, actions, etc,
  * Identifiability: is the threat of anonymity and pseudonymity, an attacker can link the identity of the host by its actions easily,
  * Non-repudiation: is a threat of plausible deniability, where an attacker is able to know all real user's activities because the user did not deny any things about his actions for an attacker,
  * Detectability: an attacker can infer if one item of IOIs exists or not,
  * Information Disclosure: user's personal information are accessible by an authorized party or by an attacker.
- Privacy-threats in soft privacy:
  * Content unawareness: unawareness influences negatively by making wrong decisions and activities. For example, sharing user's personal information with others even with attacks,
  * Policy, and consent Noncompliance: an attacker can exist inside the system, so he can provide a spurious policy to the user for breaching its privacy, there are also problems in the management of the system's policies in causing the noncompliance of policy and consent. Even if the system announces its privacy policies and privacy laws to its users, the system can disclose the personal information of its user easily, i.e. the system cannot be a trusted party.

In this thesis, we focus on linkability, identifiability threats of the first kind of privacy-threats, i.e. privacy-threat based quality, especially in hard privacy threats.

3.5 Privacy-Properties Taxonomy

According to [56, 64, 65, 58, 4, 55, 66, 67, 68] privacy properties are privacy objectives, they are illustrated in Fig. 3.2. According to [56], privacy properties are divided into two parties according to the privacy type and privacy threats (soft and hard) as you can see them in Fig. 3.2:

- Unlinkability: an attacker cannot know the relationship between two or more IOIs, e.g. hiding the relationship between two packets sent by the same source host, or different user's actions, sessions of communication can not be linked,
- Anonymity and pseudonymity: firstly, anonymity is the inability to reveal the original user's identity, e.g. IPv6 address. In other words, it means an attacker cannot link the identity of the host with its activity or with its other information, e.g. anonymous sender of a message. Secondly, pseudonymity is a designation of a user identifier, e.g. user's name by many identifiers instead of his real identifier, so pseudonymity is using various

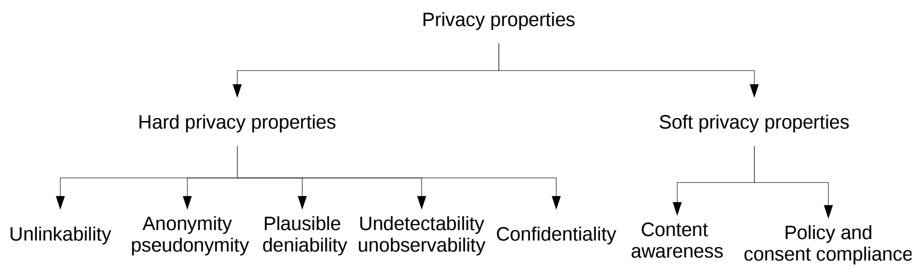


Figure 3.2: Taxonomy of Privacy Properties.

identifiers (name of the device, etc), e.g. a sender sends many messages to the same destination under different pseudonyms,

- Plausible deniability: possibility of the user to deny his sayings and actions, i.e. an attacker cannot prove that the user has done or said something, e.g. the source can deny that a specific message is sent to the destination,

- Undetectability, and unobservability: the attacker cannot disclose the user's activities. Undetectability is also called covertness, an attacker cannot detect IOI exists or not, e.g. an attacker cannot know that the sender exists in a given location. Unobservability means achieving anonymity and making messages indistinguishable, i.e. undetectability or dummy traffic (adding meaningless messages to the network when less communication). So, unobservability=anonymity+undetectability,

- Confidentiality: confidentiality in preserving privacy is hiding or controlling user's data content, e.g. the source can send its encrypted messages to the destination, or a server can apply access control to discover its services by a subset of clients (authorization clients). In other words, this property is selecting the authorized entities that can be access to the personal or sensitive information,

- Content awareness: the user is aware of his personal data anytime and anywhere, he can control his information too, he knows how and with who share his information, in particular in e-health applications,

- Policy, and consent compliance: it is a set of rules which explain how all components in the system protect the user's data, so the user can use this system if he accepts the privacy policies of this system.

Fig. 3.3 represents the relationship between the most popular privacy-threats and privacy-properties. In this thesis, we focus on unlinkability, anonymity, and pseudonymity privacy-properties.

	Privacy properties	Privacy threats
HARD	Unlinkability	**L**inkability
	Anonymity & Pseudonymity	**I**dentifiability
	Plausible deniability	**N**on-repudiation
	Undetectability & Unobservability	**D**etectability
	Confidentiality	**D**isclosure of information
SOFT	Content awareness	content **U**nawareness
	Policy and consent compliance	policy and consent **N**oncompliance

Figure 3.3: Relation between LINDDUN and privacy-properties.
[4]

3.6 Privacy Preserving Techniques (PPTs) Taxonomy

According to [57, 69, 55, 2, 62, 26, 54, 60, 3, 51, 70], Fig. 3.4 illustrates our novel taxonomy of PPTs which includes the most used and popular techniques for preserving privacy as follows:

- Data perturbation techniques: according to [57], their goal is focusing on hiding or modifying the sensitive information, these techniques include:
  - Noise addition techniques: or randomization techniques. They aim to add noise to the sensitive information in order to disorganize the communication. With the randomization technique, the linkability between communication peers will be hard to achieve [70],
  - Anonymization techniques: are the most important techniques for preserving privacy. They focus on hiding the original identities [26] with protecting the accuracy of information [55]. There are many ways to achieve anonymity as anonymity tools in [70] as follows:
    - Generalization: is based on an indirect way to preserve privacy. For example, to preserve the personal information of the user as his birthday, to know the user's age range, we can approximate his age range through the information with which he created an account instead of to his birth date directly. Note that this technique is better when we want to preserve data privacy not communication privacy. In other words, generalization technique is used with identifiers, not identities,
    - Multiple identities: working on identities rather than on identifiers enhances better communication privacy. Using many identities makes the linkability between identity and other items difficult. Multiple identities technique include:
      - Pseudonymization: typically, this technique focuses on identifiers rather than identities, but on identities is possible to use. Use not real identities during the communication instead of the real identity,

  · Digital identity management: the goal of this technique is like the previous one. This technique focuses on two things as follows: (i) generating many identities as pseudonyms, and (ii) management process on these pseudonyms (when the user has to use them), i.e. use the first pseudonym with Facebook, the second with Google, etc.

    * Communication obfuscation: obfuscation is hiding information by adding fake information to the original information for example in order to degrade the quality of information. Thus, this solution impacts on the accuracy of information negatively [55]. Communication obfuscation focuses on preserving meta-data in order to preserve communication privacy as well as possible. Communication obfuscation is used more when we browse online. This tool is divided into the below techniques which are:

      · Virtual Private Network (VPN): is an approach based on a proxy to achieve anonymity. The proxy here is the VPN. At the endpoints of the VPN, the traffic analysis attack can infer all meta-data. Thus, communication privacy does not preserve very well. According to [70], VPN is based on tunneling, i.e. encapsulating the data at the tunnel,

      · Proxy server: there is an intermediate entity, i.e. proxy between the sender and the receiver. To preserve sender anonymity, the proxy communicates with the receiver by its IP address instead of the sender's IP address. So, the proxy hides the IP address of the sender like in the anonymizer [71],

      · Mix-Networks (Mix-Net): is extended from the previous technique, so this technique includes a set of proxy entities (mixes) instead of a single proxy entity. Use a random path from the sender to the receiver in order to provide preserving privacy. MIX-Net is focusing on preserving two properties of privacy in the sender and receiver sides, i.e. anonymity and unlinkability. Its main idea is taking a set of messages in order to scramble/ delay/re-encode them to make the tracking the communication flows difficult to the traffic analysis attack [3],

      · The Onion Routing (Tor): is based on Mix-Net, Tor is the famous protocol of anonymization of communication in the Internet. Anonymization is done by hiding the identity of communication entities [51]. We are going to explain it with more details in the next chapter, i.e. related work.

- Data restriction and minimization techniques: focusing on the limitation of data use. In other words, data minimization technique reduces/retains the necessary and the relevant sensitive information to do the desired goal, i.e. providers have to use/access only to the sensitive information which they really need [62], Data restriction and minimization techniques include:

  – Access control/privacy by design technique: the user should have the ability to control his information, and he should select a set of individuals who can access to his personal information. In other words, the user decides that his information are accessed by selected authorization entities,

  - Cryptography-based technique: by lightweight operations and algorithms due to IoT devices' limitations,
  - Privacy awareness: lack of awareness on privacy (how user's personal information are processed, who can see these information, when these information are collected, etc [54]) at users leads to breach of their privacy easily. In other words, the user should control and manage his information, and he has to realize the privacy risks.

- Header Information Protection : according to [60], there are two techniques to protect header information which are:
  - Temporary Stateless Address Auto-configuration: focusing on modifying the interface identifier (IID) of the IPv6 address (the IPv6 address contains prefix field and IID field) from time to time. Temporary address needs to change prefix and IID fields to achieve better the privacy,
  - Cryptography Generated Address (CGA): this technique is proposed to prevent spoofing or stolen attack to get the IPv6 address.

- Based on trust term: in [2], there is a taxonomy on privacy approaches for preserving location privacy according to the trust term. This taxonomy contains four sets which are:
  - Approaches based on trust of a Service Provider (SP): these approaches include three approaches which are:
    * Working with data: focusing on preserving the privacy of the personal information of users by adding noise, deleting data, removing data, encryption, minimizing data, etc,
    * Access control and request: allowing the user to edit, access and remove his information, he can allow the service provider to use his information or no, etc,
    * Awareness, policy and low: aiming at the knowledge of the user about: (i) breaching privacy, (ii) his rights, (iii) privacy policies and lows, etc,
  - Approaches based on trust of a Third Party (TP): include:
    * Obfuscation and land marking: the first approach is to change the real personal information or the real sensitive information by transformation functions. The second approach for preventing the attacker to know the real location of the user according to his known places,
    * K-anonymity and cloaking area: aim to hide the ID of user for k-anonymity, and to hide the real location for cloaking approach,
    * Mix Zone: the user uses his new pseudonym in each zone where his area is divided into many zones.

- Approaches based on trust peers: here there is a collaboration between the sender and the receiver in preserving privacy, these approaches include two others which are:
  * Cooperation: communication entities collaborate with each other directly in order to preserve their privacy from the third trust party and from the service provider too,
  * Caching: storing some answers to queries in order to use them in the future queries. The goal of this approach is to reduce the number of connections with the service provider.
- Approaches based on no trust: the last set includes three approaches as follows:
  * Anonymity: we have explained it shortly before, it is the best approach to preserve privacy from all parties,
  * Dummies: the user sends a set of false queries with his real query instead to send it clearly,
  * Private Information Retrieval (PIR): according to [55], PIR for protecting the content of the communication, PIR preserves only destination anonymity, getting answers about the user queries from the server without disclosure on his real identity.

3.7 Anonymous Communication in the IoT

According to [72], anonymous communication in the IoT is one of the IoT challenges, the traditional anonymous communication systems need large capacities in terms of power, computation, and bandwidth, whereas the IoT-devices have limitations on these. There are two kinds of anonymous communication solution in the IoT [72]:

- Anonymous communication solutions based on computation offloading: the main idea of these solutions is using the traditional anonymous communication solutions at gateway (like 6LBR),
- Anonymous communication solutions based on lightweight cryptography: this kind is also divided into two types:
  - Anonymous communication based on identity encryption,
  - Anonymous communication based on pseudonym encryption.

3.8 The Best Anonymous Solution

The best anonymous solution has to guarantee Deployability (D), Usability (U), Flexibility (F), and Simple Design (S) DUFS[73], as it can be seen in Fig. 3.5, where:

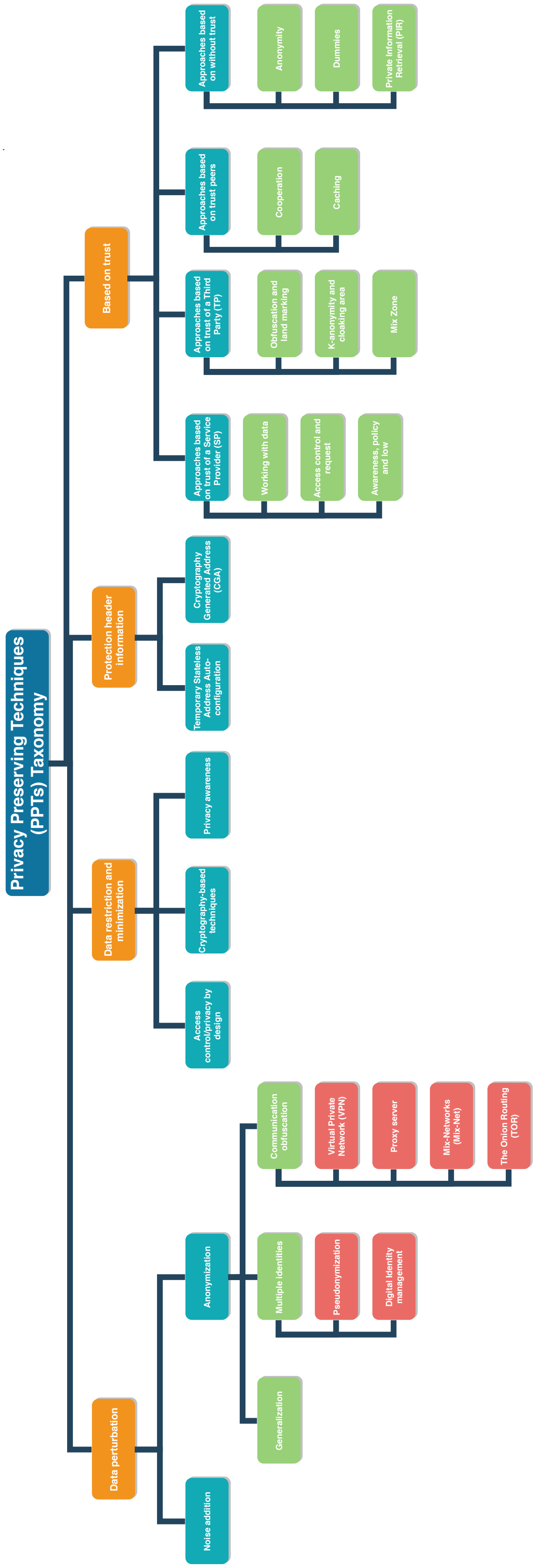


Figure 3.4: Privacy Preserving Techniques (PPTs) Taxonomy.

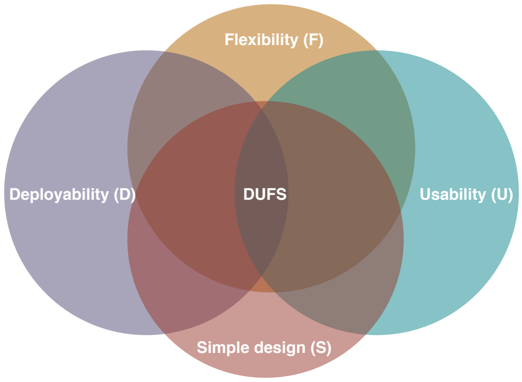


Figure 3.5: The Best Anonymous Solution.

- Deployability (D): the solution can be used in the real applications,
- Usability (U): the solution will be compatible with any application, operating systems and platforms. Stated differently, the solution will not need heavier configurations, more latency, or bandwidth, etc,
- Flexibility (F): the solution can resolve any problems that we may encounter with the in any new applications. The future applications, architectures will not need a reinventing the design of the solution,
- Simple Design (S): the design will be very understood and very clear.

3.9 Security Vs Privacy

Hiding the payload does not mean preventing the traffic analysis attack (is an attack on privacy that focuses on tracking the communication flows [3]) to observe the communication [3]. According to [2], we classify the difference between security and privacy in the table blow (Table. 3.1):

Metric	Security	Privacy
Data protection	Data and device's information.	User and his information.
Penetration	Detecting information, e.g. password.	By collecting user's information and analyzing them.
Kind of attack	Protection data between the sender and the receiver which are trusted parties from a foreign attack.	Any party between the sender and the receiver can be an adversary. The attack can be a foreign attack and can be a non-foreign attack.
Properties	Confidentiality, integrity, availability, non-repudiation, etc.	Anonymity, pseudonymity, unlinkability, Plausible deniability, undetectability and unobservability, confidentiality, content awareness, policy and consent compliance.
Approaches	Access control, encryption, digital signature, etc.	Tor, data perturbation, data restriction, randomization, anonymization, pseudonymization, obfuscation, etc.

Table 3.1: Security Vs Privacy.

3.10 Conclusion

The most important features of preserving communication privacy are: (i) making communication ambiguous, (ii) harder to track the traffic, and (iii) confusing to recognize communication entities, etc. Achieving privacy is the ability to prevent the linking of meta-data, i.e. communication entities, messages timing, location of the communication entities timing, sizing, etc and flow information like source/destination network and transport addresses and transport mode from the third party, because these information are only known from communication entities.

We have seen the essentials of privacy. Currently, in this thesis, we are working on communication privacy in the IoT. The latter is preventing the privacy attacks to infer that communication is taking place between communication entities. In the next chapter, i.e. related work, we are going to discuss the recent solutions that preserved communication privacy in the IoT.

Chapter 4

Related Work

> Being positive does not mean you don't ever have negative thoughts. It just means you don't let those thoughts control your life
>
> *Jay Shetty*

Contents

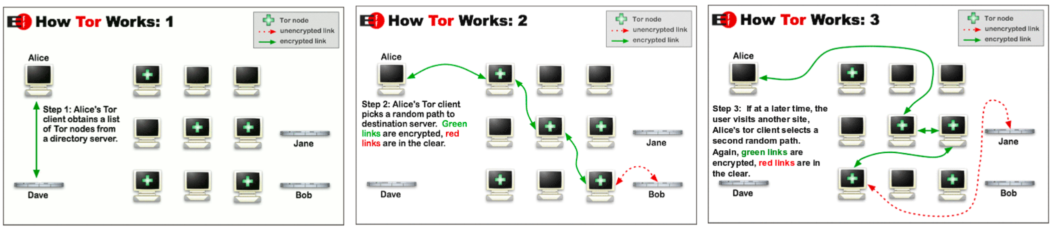
4.1 Introduction

According to [74, 75, 76], preserving communication identifiers privacy is a very important property for the success of IoT applications. However, traditional solutions cannot apply directly to 6LoWPANs due the constrained capabilities of devices [16, 2, 77]. In this chapter, we touch upon the main solutions for preserving communication identifiers privacy in the Internet in Section 4.2, and in 6LoWPANs considering the most important features (Section 4.3), i.e (1) in which layer this solution operates? to know where the communication privacy will be achieved, (2) what is the protocol that this solution is based on? the solution can be based on a traditional protocol or it is a new one itself, (3) what is the kind of attacks that this solution is powerful against? there is not any solution which is powerful against all communication privacy attacks, (4) to which application this solution apply?, some solutions are specific to applications whereas others are not, (5) what is the simulator used for solution validation? popular general simulators such ns-2, and ns-3, as well as IoT-specific simulators such as Cooja, TinyBLE, have also been used, (6) what are the sensitive information that this solution focuses on? the sensitive information are source and/or destination MAC, IP addresses and/or port numbers, (7) what is the privacy-preserving technique in use? and, (8) how long does a given solution last? The smaller this parameter, the better the solution. Section 4.4 summarizes the IoT-solutions in a table. Finally, Section 4.5 is the conclusion of this chapter.

4.2 Internet-based Solutions

4.2.1 The Onion Routing (Tor)

The Onion Routing (Tor) [78], commonly referred to as Tor, hides source and destination IP addresses. Tor architecture includes (see Fig. 4.1): (a) Tor client, its main operations are (1) Tor client picks set of Onion Routers (ORs) to use them as a Tor circuit, (2) it generates and manages cryptography keys to anonymize the communication, (b) ORs or Tor nodes: for the sake of illustration we assume that there are three ORs: entry router (OR1), middle router (OR2) and exit router (OR3). Tor network is the set of these ORs, their goal is reaching source's data to the destination by an anonymous way, (c) Directory Server (DS): all ORs' information are available in this DS, and (d) Tor server: communicates with the Tor client over the Tor network anonymously [79, 51, 73, 80, 81, 82]. Each OR knows only its successor and its predecessor.



(a) Step 1. (b) Step 2. (c) Step 3.

Figure 4.1: Tor Architecture and Operations. [80].

Note that the current version of Tor does not support IPv6 entirely. In other words, the communication between Tor client and Tor network, and between Tor network and Tor destination can support IPv6, but the communication between Tor routers does not support IPv6, it is still used IPv4 [83, 84]. There are many mechanisms of IPv4/IPv6 translation [85]. With Tor, there are some powerful traffic analysis attacks who focus on sizing and timing by using sophisticated statistical techniques to breach communication privacy, i.e. the attacker can track the incoming and outcoming traffic from/to Tor network, so he can confirm with a high probability that the specific communication patterns are communicated with each other [80, 73, 79]. An attacker can be a malicious OR1 (entry router), so it can de-anonymize the client Tor easily. We can find a censorship attack in Tor [82], i.e. he can block the entry router or the exit router. In addition, Tor has a limitation about the transport protocol, it worked only with TCP connection.

4.2.2 One Time Addresses (OTA)

OTA is a communication architecture based on One Time Addresses [86, 87]. Its main idea is to use a single address to send or receive just one packet by achieving both flow-packet and host-flow unlinkability.

OTA Architecture

OTA operations are based on the general Internet architecture as presented in Fig. 4.2 which shows an example of two Autonomous Systems (ASs) connected to the Internet. We assume that the sender is served by AS1 and the receiver is served by AS2. Within each AS, there are Border, Core, and Access Routers denoted by BRs, CRs, ARs, respectively. OTA changes the structure of IPv4 packets. It defines two fields that are required in its structure: Host IDentifier (HID) coded on 4B, it is like an IPv4 address, and Flow IDentifier (FID) also coded on 4B. The OTA address is the encryption of the combination of HID and FID by ARs.

OTA End-to-End Communication

Before a communication starts between a source and a destination node, there is a need for two operations: connection establishment and address-pool creation. Connection establishment

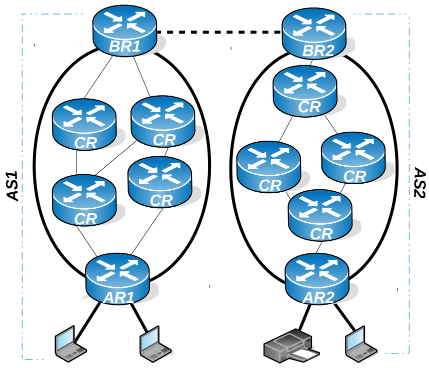


Figure 4.2: An example of two Autonomous Systems connected to the Internet.

is done in two phases. During the first phase, both authentication of hosts and negotiation of encryption keys between ASs and between the components of each AS must happen. The encryption keys will be used later for the encryption/decryption of OTA addresses and reply OTA addresses, which is the address that the other host needs to use when sending a packet back to the first host.

- OTA Address-pool generation: we explain the second part of the connection establishment. We start by detailing the pool generation process in the destination host (the pool of addresses is generated in the same way with the sender host). After the establishment of the connection between both hosts, the source host in AS1 sends a request to the destination host which contains the size N of the destination's reply address. The destination host informs the AR2 in AS2 with its size for its reply addresses, AR2 creates N reply addresses with the same HID and FID of the destination host. Finally, AR2 sends the packet to the source host. The packet contains the encryption set of the destination's reply addresses by the shared key between AS1 and AS2. The content of packet is the pool address of destination host as in source host,

- OTA Packet routing: the source host picks one address from its pool (set of OTA destination), and sends the packet to AR1. When AR1 receives this packet, it generates OTA for the source, and it sends the packet to BR1. This latter decrypts OTA destination to infer the AS destination. Note that, in the destination OTA there is an information called Autonomous System Number (ASN) to determine the AS destination. The ASN in OTA destination determines the border router to which the packet will be forwarded. In our case, the packet will be forwarded in BR2. When BR2 receives the packet, it recognizes the AR2 by the decryption of the OTA destination because there is a shared key inside each AS. Thus, BR2 infers the AR2, then it forwards OTA destination to the next hop CR, which in turn continues forwarding the packet to other CRs until it reaches to AR2. When AR2 receives the packet, it extracts the destination host (HID2) by decryption of destination OTA in order to send the packet to the destination host.

OTA is a good approach to prevent an attacker from tracking the traffic or inferring hosts' flow information in traditional Internet by guaranteeing host-flow and flow-packet unlinkability. Its main idea consists of using a different address per transmitted or received packet. The authors changed the structure of IP packet headers. They used an OTA address instead of an IPv4 address where the OTA size is 28B and the size of the OTA header is 82B. This new structure introduced by OTA requires that all routers take into consideration this new structure.

4.2.3 Moving Target IPv6 Defense (MT6D)

To avoid flow identification, the authors of MT6D in [88] proposed to rotate source/destination MAC and IPv6 addresses, as well as source/destination port number according to the host which wants to send a packet (source/destination) several times dynamically. The main processes at beginning of each interval time are: (1) calculating the new MAC, IPv6 and port number for the host which will send the packet (new identifiers) by hashing: its original MAC, IPv6 addresses and its port number, shared session key and time stamp. (2) Informing the routers about the new identifiers which will use them at next time. The new identifiers are located in a new MT6D header. The flow identifying information in the original IPv6 header are the original source and destination IPv6 addresses. (3) Encrypting/decrypting the original IPv6 packet and encapsulating/decapsulating it in MT6D packet. The flow identifying information in the new MT6D header are the new source and destination IPv6 addresses. The authors implemented their solution in two modes: (i) gateway device mode; they set two Access Routers (ARs). AR1 is like a gateway between the source and the Internet, and AR2 is like a gateway between the destination and the Internet. The ARs apply the MT6D processes, (ii) embedded software mode where the host is the one which applies all the MT6D processes. We choose the second mode of implementation for explaining the main processes of MT6D which we mentioned previously. For example a `src-host` wants to send a packet to a `dst-host`, we focus here on the IPv6 addresses; the `src-host` sends its packet to AR1 with its original IPv6 address (`src-addr`). After that, AR1 hashes `src-addr` to get the new address, e.g. `src-addr'`. AR1 puts `src-addr'` in the new MT6D header, and encrypts the original IPv6 packet. Finally, AR1 encapsulates the encrypted IPv6 packet in the new MT6D packet, and sends the MT6D packet to AR2 over the Internet. We determine the destination gateway (AR2) by the flow information included in the MT6D packet. AR2 decapsulates the received MT6D packet to extract the original encrypted IPv6 datagram that will be deciphered by AR2 to get the original IPv6 datagram. AR2 sends the obtained packet to the destination host where the original destination's IPv6 address is within the original IPv6 datagram. Finally, the destination host receives systematically the IPv6 datagram [89, 90]. At this stage, the authors added the new MT6D header (62B) to the original IPv6 header. In addition, MT6D is worthy to note that host-flow unlinkability property is only achieved between MT6D gateways in the first mode. Moreover, this property cannot be ensured during the interval of time (e.g. the interval of time = 10 seconds, MT6D keeps the same new addresses between [1s-10s]) in both modes where a communication session is divided into many time intervals.

4.3 IoT-based solutions

4.3.1 Tor for Smart Home

The original Tor protocol for privacy protection on the Internet has a limitation about the transport protocol. It works only with TCP connections. To integrate the Tor into the IoT technology, the authors of [91] used the Tor protocol for the smart home applications in order to anonymize the communication. The authors set a gateway that runs Tails (The Amnesic Incognito Live System) [92]. Tails ensures that all connections pass through the Tor, and no MAC address spoofing, i.e. no traffic tracking based on MAC addresses. It is to be noted that this solution did not address the context of 6LoWPANs and has only been tested with IPv4 addresses. Therefore, its applicability in 6LoWPANs may need major modifications.

4.3.2 Generating IPv6 Pseudonyms

In [93], the authors propose a lightweight IPv6 address auto-generation algorithm in 6LoWPANs with the aim of preserving communication privacy without introducing additional headers. In particular, they consider that each 6LoWPAN has one LBR and a few clusters. For each cluster, there is a cluster head (CH) acting as Access Router (AR) and some cluster members (CM) acting as terminals. The structure of IPv6 address in this paper has three fields: (a) global routing prefix, (b) cluster ID, and (c) member ID, i.e. hierarchical addressing. A CM gets its IPv6 address by three-step procedure: (i) its prefix is obtained from the LBR, (ii) its cluster ID is obtained from its cluster head (AR), and (iii) its member ID is generated by itself by picking a member ID randomly from its pool for each time window. With this solution, there is a complexity at the CH as it is responsible for allocating and sending the information about address changes to all its CMs. In addition, CMs are the only ones that change their member ID. A problem with this solution is that it keeps using the same MAC address. Therefore, even when a CM changes its IPv6 address every time window, it keeps using the same MAC address, and thus could be used by an attacker to link changing IPv6 addresses together from their corresponding MAC address.

4.3.3 Using MAC pseudonyms (CryptoCop)

In [94], the authors identified MAC addresses as a vulnerability that limits the use of pseudonyms at the routing layer. They considered the situation of Bluetooth Low Energy (BLE) protocols in addition to 6LoWPANs. In fact, in these protocols devices keep using the same MAC address even when pseudonyms are used at the IP layer. As a solution, they proposed to change the source MAC address for each frame to preserve source anonymity thereby achieving sender-frames unlinkability. However, their solution still suffers from receiver-frames linkability as it only achieved source anonymity. In addition, it is not clear how the sender informs the destination about its new MAC address.

4.3.4 Changing IPv6 and MAC Address Each Time Window

In [95], the authors proposed a new neighbor discovery extension called 6LoWPAN-ND with the goal of minimizing the overhead of operations of the traditional neighbor discovery scheme and to change IPv6 and MAC addresses for each time window. According to the traditional architecture of 6LoWPANs consisting of a LBR, ARs, and devices, their scheme relies on the following operations. To get a new MAC address, a device starts by sending a request to its corresponding AR with its information (device ID, nonce, etc.). The AR forwards the request to the LBR which selects a new MAC address from its pool and sends the reply back to the AR with device information, which in turn forwards the reply back to the device. Once the device receives its new MAC address, it makes use of it to create a new IPv6 address. Although this solution provide a certain level of anonymity through the use of pseudonyms both the link and routing layers, it has a non negligible overhead caused by the number of messages exchanged between the device, the AR and the LBR. In addition, this solution creates pressure at the LBR due the centralization of the operations of MAC assignment to the devices. Moreover, the solution does not seem to have effect on those destination nodes located outside the 6LoWPAN (e.g. a destination located in the global internet).

4.3.5 Mutual Change of Source and Destination MAC Addresses

In [96], the authors proposed a scheme called Ephemeral to hide both source and destination MAC addresses of devices within a 6LoWPAN at each time window. Their proposal makes some assumptions and operates as follows. Each device has two keys K_1 for encrypting the payload of the frame, and K_2 for generating new MAC addresses. The generation of new source and destination MAC addresses called nsMAC and ndMAC respectively relies on the use of cryptographic functions. Specifically, the use: (1) a random value r which is on 14B or 15B long, (2) a counter c which is 1B long, there are two kinds of the counter, one for the source s_c and other for the destination d_c, (3) a key K_2, and (4) the initial source and destination MAC addresses called sMAC and dMAC, respectively. Each node knows the MAC address and the initial r of its neighbors. In addition, the updated r will be sent over Internet Control Message Protocol (ICMP) messages for a time window. The source entity has to generate the two new MAC addresses nsMAC and ndMAC to replace the initial source and destination MAC addresses (sMAC and dMAC). After that, the source entity adds two fields (2B long) to the frame header which are s_c and d_c, and sends the frame to the destination entity which needs to decrypt the frame header to infer the original source and destination MAC addresses (sMAC and dMAC). Although Ephemeral provides source and destination MAC anonymity, it has some limitations as it adds extra overhead (2B field in the header). In addition, it is not clear how it operates for flows going outside the 6LoWPAN where destination nodes can be located anywhere in the global Internet.

4.3.6 Changing All Communication Identifiers

In [97], the authors propose uMT6D which is an adaptation of the traditional MT6D [88] for the IoT technology. uMT6D changes the source and destination IPv6 addresses for each time window, as well as source and destination MAC addresses and port numbers to prevent the network traffic attack from linking the communication between two entities. Rotating addresses is based on a lightweight hash function. uMT6D encapsulates the original encrypted IPv6 packet into the new uMT6D packet with new addresses. Rotation, encryption, and encapsulation addresses can be done in the gateway or at the device. uMT6D added a new header to the original IPv6 header to preserve communication privacy which leads to more energy consumption and increased latency.

4.3.7 Using Lightweight IDs

In [98], the authors proposed a solution called Communication Security and Privacy (CSP) support for 6LoWPANs, which relies on the use of lightweight IDs instead of IPv6 addresses. CSP defines three types of IDs: Permanent ID (PID), Session ID (SID), and Temporary ID (TID). Each node in the 6LoWPAN communicates with its 6LBR to get its PID and SID, using the source address TID as a MAC address, and the destination address is 6LBR PID. The 6LBR maintains a mapping table with correspondences between PIDs and SIDs. There are two cases according to the location of the destination which are: (i) the source and the destination are in the same 6LoWPAN network, (ii) the source and the destination are in different 6LoWPAN networks. While this solution aims at hiding all communication identifiers, the use of IDs instead of IPv6 addresses may cause compatibility issues. In addition, there is a need for a coordination protocol between routers for the updating of the changing IDs.

4.3.8 NAT-Inspired Solutions

In [99], the authors proposed a solution called new Advanced Persistent Threat (APT) to increase communication identifiers privacy between two entities inside ASs. Their main idea is similar to the technique used by Network Address Translation (NAT) as they define two types of addresses: public address which uses for external communication toward another AS, and private address which uses for internal communication within the same AS. The BR is an important part of this solution as it has the role of ensuring address translation from and to external ASs. As such it is considered as the most vulnerable point in the entire architecture. In addition to this weakness, APT did not envisage MAC address change so communication privacy within the local network might not be preserved. Moreover, the way the mapping is performed by the BR makes it vulnerable to traffic correlation attacks at the BRs.

4.3.9 Using Tor for UDP Communications (Tor-UDP)

In [100], the authors proposed to use Tor to provide communication identifiers anonymity in 6LoWPANs. However, due to the limited computation, storage, and energy resources, most communications in 6LoWPANs are based on UDP. Therefore, they proposed to modify existing Tor operations, which work on TCP traffic only, to be compatible with UDP communications. They defined IoT Onion Routers (IORs) to anonymize the source and destination of communications entities as well as port numbers. As other solutions, it is not clear to which device class this solution can apply as it does not seem to operate on very constrained devices. In addition, there is no changing of MAC addresses which makes the solution vulnerable to local attackers.

4.3.10 Using One Time Address (uOTA)

In [101], the authors proposed Micro One Time Address (uOTA) which is a solution that aims to prevent both (i) insider attacks (attacker situated in the same cluster or in the same 6LoWPAN with the communication entities), and (ii) outsider attacks (attacker situated outside the 6LoWPAN) from de-anonymizing the communication between the two communicating entities and linking the relationship between them. The main idea of uOTA is mainly inspired from OTA [86] and adapted to operate in constrained 6LoWPANs. uOTA uses one IPv6 address to send or to receive just one packet as well as port numbers. As other similar proposals, uOTA has some shortcomings as it (i) introduces extra delay due to IPv6 translations, (ii) the coordination between routers has to be done, and (iii) uOTA does not envisage changing MAC addresses which makes it vulnerable to local attackers.

4.3.11 Using Congruence Classes to Allocate Addresses

In [102], the authors proposed a solution called Privacy-enabled Disjoint and Dynamic Address Auto-Configuration (PeDAAC) protocol for 6LoWPANs with the aim of allocating MAC and IPv6 addresses dynamically without the need for a Duplicate Address Detection (DAD) protocol while achieving anonymity and unlinkability of communication as well as location privacy against spatial and temporal correlations. The main idea of PeDAAC is based on the use of a congruence classes which have interesting mathematical properties in creating non duplicate addresses, the authors did not explicitly show how the coordination between new addresses is achieved such as the creation of AR-Unique ID (UID) and node-UID fields in the structure of IPv6 addresses. In addition, as shown in [103], their solution is not resistant against attackers situated within the same cluster. Moreover, PeDAAC makes assumptions that layer 2 messages exchanged are secure without explicitly showing how this is achieved.

4.3.12 Using Tor for MQTT Protocol (MQTT-Tor)

In [100], the authors propose to solve the problem of communication identifiers for the Message Queue Telemetry Transport (MQTT) protocol stack based on Tor, called MQTT-Tor. In the traditional MQTT protocol, there is a broker acting as a server for publishing or subscribing topics through clients. In MQTT-Tor, a set of brokers is set as Tor routers and clients (devices) play the role of Tor clients. The main limitations of this solution is that it does not apply to very constrained devices and does not cope with local attackers as MAC addresses are not changed.

4.3.13 Delegating Tor Operations (Tor-Delegation)

In [104], the authors showed that the original Tor cannot be applied directly to IoT networks due to the constrained nature of IoT devices and the incompatibility of protocols. They proposed to delegate and offload those sophisticated encryption operations to a router or a web server of the device owner which would act as entry node. The delegation server establishes the Tor circuit during the connection step and encrypts the data, the source and destination IP addresses many times depending on the number of ORs along the Tor circuit. Although this might be a good solution to the communication identifiers privacy in IoT networks, the use of the delegation server breaks the end-to-end reachability that is advocated by the design of 6LoWPANs. In addition, the solution is also vulnerable to local attackers as MAC addresses are not changed.

4.4 Summary of IoT-based Solutions

We sum up the 6LoWPAN-based communication identifiers privacy preserving solutions in Table 4.1. In our study on communication flow privacy preservation in 6LoWPANs, we find that many papers are based on obfuscating communication by proxies or by Tor routers, and are based on cryptography techniques. We can conclude that these techniques are the most effective for preserving communication privacy in the 6LoWPANs. In addition, a good proportion of the cited papers in this study have not been designed for a specific application and validated their proposals based on simulations.

Approach	Year	Layer	Based on	Against attack	Application	Simulator or tool	Sensitive information	PPT	Duration
Tor-smart home [91]	2014	Network and MAC	Tor	Outsider attack	Smart home	Experimental e.g. on Skype	sIPv4, dIPv4, sMAC, dMAC	Tor	Peer session
APS [93]	2015	Network	IPv6	Eavesdropper attack	Smart home	ns-2	sIPv6	Randomization	Time window
CryptoCop [94]	2016	MAC	BLE	Eavesdropper attack	Heart rate prototype	TinyBLE	sMAC	Cryptography	Peer packet transmission
6LoWPAN-ND extension [105]	2016	Network and MAC	6LoWP-AN-ND	Eavesdropper attack	Unspecified	ns-3	sMAC, dMAC sIPv6, dIPv6	NAT proxy	Time window
Ephemeral [96]	2016	MAC	IEEE 802.15.4	Traffic analyzing	Unspecified	Cooja and WSNET	sMAC, dMAC	Cryptography	Time window
uMT6D [97]	2017	Network and Transport	MT6D	Network traffic	Unspecified	Cooja	sIPv6, dIPv6 sPort, dPort	Proxy	Time window
CSP [98]	2017	Network	IPv6	Address exhaustion, and disclosure information	Patient monitoring	ns-2	sIPv6, dIPv6	Cryptography	Peer session
APT [99]	2017	Network	IPv4	External attack	Smart cars	Unspecified	sIPv4, dIPv4	NAT proxy	Communication session
Tor-UDP [100]	2017	Transport	Tor	De-anonymizer attack	Unspecified	Java	sIPv4, dIPv4 sPort, dPort	Tor	Peer session
uOTA [101]	2018	Network and Transport	OTA	Traffic attack	Healthcare	Cooja	sIPv6, dIPv6 sPort, dPort	Multiple IDs pseudonymization	Packet trans-mission or reception
PeDAAC [102]	2018	Network and MAC	IPv6	Eavesdropper attack	Indoor scenario	Cooja	sIPv6, dIPv6 sMAC, dMAC	Multiple IDs pseudonymization	Time window, node movement, link failure.
MQTT-Tor [100]	2019	Application	MQTT	De-anonymizer attack	Unspecified	Python	Requests	Tor	Peer session
Tor-Delegation [104]	2019	Network	Tor	Network attacks	Unspecified	Cooja	sIPv4, dIPv4	Tor	Peer session

Table 4.1: Summary of main communication flow privacy protection solutions

4.5 Conclusion

In this chapter, we explained some traditional solutions for preserving communication privacy in the Internet (Internet-based solutions). There are some IoT-based solutions inspired by the traditional ones, and there are other solutions completely new. For each IoT-based solution, we gave its main idea and its advantages and drawbacks, and we summarized the IoT-based solutions in the table according to the important points, i.e. layer, based-protocol, against attack, application, simulator, sensitive information, PPT, and duration. We are going to explain our ideas in the next chapters according to our motivations for these recent IoT-based solutions for preserving communication privacy.

Part II

Contributions

Chapter 5

Micro One Time Address: uOTA

A person who never made a
mistake never tried anything new

Albert Einstein

Contents

5.1 Introduction

In this chapter, we proposed a new solution for preserving communication privacy in the IoT which is Micro One Time Address (uOTA). To design uOTA protocol, we follow the same principle of traditional OTA, while taking into account the particularities of 6LoWPAN networks. Our main purpose is to solve the communication flow privacy in the IoT by enabling a host-flow unlinkability. uOTA is based on a per-packet One Time Address, i.e. a given address is used exactly once to send or receive one packet. The remainder of this chapter is presented with a detailed description of uOTA.

5.2 uOTA Threat Model

In some previous solutions, a complete host-flow unlinkability cannot be guaranteed. An attacker may detect or link a flow to one communication end and escalate this into identifying the other end. For example, Tor has been shown to be vulnerable to some flow-correlation attacks based on timing analysis [106, 107]. In addition, other solutions such as basic OTA are not suitable in the context of 6LoWPAN due to their complexity and memory footprint. As it can be seen in the related work section, little has been done to protect IoT things from flow-identification attacks particularly for the promising TCP/IP-compatible 6LoWPAN architecture [2] where the flow identification threat model in IoT is represented in Fig. 5.1. We propose a solution to cope with this problem and provide host-flow unlinkability to make it difficult for attackers to determine what things are communicating with. Our solution called micro OTA (uOTA) is inspired from the OTA approach described in [86] proposed for the traditional Internet but with adaptations and optimization for 6LoWPAN architecture. Our main contributions of uOTA can be summarized as:

- Design of a flow anonymization solution that is compatible with 6LoWPAN networks to cope with the flow-identification problem in the IoT. We also provide a detailed description of the proposed solution,
- A concrete implementation on Cooja simulator to validate the proposed solution on emulated motes, as well as a comparison with existing solutions for flow anonymity in 6LoWPAN networks,
- A communication flow privacy metric that measures the levels of flow identification at different locations in the global internet.

5.3 uOTA Architecture

As shown in Fig. 5.2, a 6LoWPAN network is composed of a set of low-power devices (sensors) and low-power routers, and connected to the global Internet through the 6LBR. Within the 6LoWPAN network, the closest router to the source/destination is called AR. Depending on source and destination capabilities: source and destination hosts may be low-power devices such as beacons (Apple iBeacon, Google Eddystone, etc.), or powerful ordinary devices such as printers, and locations, the routers between the source and the destination may be either traditional powerful routers or low-power routers. As shown in Fig. 5.2, we assume that a communication is taking place between a source and destination nodes called `src-host` and `dst-host` through access routers `AR1` and `AR2`, and `LBR` routers (`LBR1` and `LBR2`), respectively. We assume that the real addresses of source and destination hosts are globally. We assume also that substitution/recovery algorithm used by uOTA is based on the last 64b of the IPv6 address. We divide 64b into

Src-host address	Dst-host address	Location of attacker
aaaa :: 212:7402:1:101	aaaa :: 212:7402:2:202	The source 6LoWPAN network
aaaa :: 212:7402:1:101	aaaa :: 212:7402:2:202	The source AS
aaaa :: 212:7402:1:101	aaaa :: 212:7402:2:202	Elsewhere in the global Internet
aaaa :: 212:7402:1:101	aaaa :: 212:7402:2:202	The destination AS
aaaa :: 212:7402:1:101	aaaa :: 212:7402:2:202	The destination 6LoWPAN network

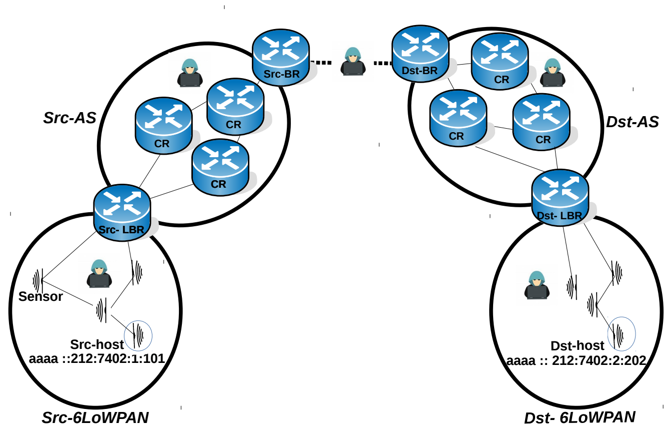


Figure 5.1: Flow Identification Threat Model in IoT. Here an attacker is interested in identifying that a communication is taking place between a source and destination entities. Depending on the location of the attacker, different types of flow identifying information can be obtained. The attacker may be located in the source 6LoWPAN network, the destination 6LoWPAN network, the source AS, the destination AS, or elsewhere in the global Internet.

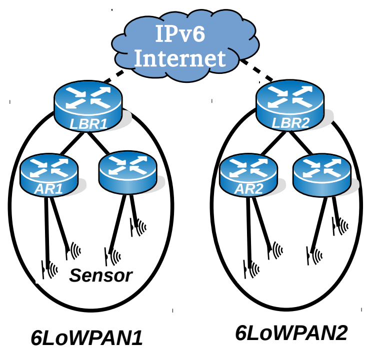


Figure 5.2: uOTA Architecture.

Notation	Description
`addr`	an IPv6 address
$\mathtt{S}_s$`addr`=$\mathtt{S}_s$`(addr)`	a new address to send a new packet
$\mathtt{S}_r$`addr`=$\mathtt{S}_r$`(addr)`	is a new address to receive a new packet
$\mathtt{S}_s$`(addr)`	substitution operation on `addr` at step i
$\mathtt{S}_r$`(addr)`	substitution operation on `addr` at step $i+1$
$\mathtt{S}_r^{-1}$`(`$\mathtt{S}_r$`addr)`	recovery of `addr` from $\mathtt{S}_r$`addr`
$(A,B/msg)$	sending a *msg* from A to B

Table 5.1: uOTA Notations.

four parts P_1,P_2,P_3,P_4 where each part equals 16b. The main function in our algorithm is represented in formula 5.1, where $P_{1,2,3,4}$ are the real parts in the real IPv6 address, $P'_{1,2,3,4}$ are the new parts of the new IPv6 address, and $R_{1,2,3,4}$ are random numbers for each part ($R_1 \in [0-10]$, $R_2 \in [0-20]$, $R_3 \in [0-30]$, $R_4 \in [0-40]$).

$$\mathrm{P'}_{1,2,3,4} = \mathrm{P}_{1,2,3,4} \mod (R_{1,2,3,4} + \text{Current\_time (s)}) \tag{5.1}$$

In addition, there is an address mapping table in each AR containing two addresses for each host: a new source address $\mathtt{S}_s$`addr` used to replace the `src-addr` to send a packet and a new source address $\mathtt{S}_r$`addr` used to replace `src-addr` to receive the packet, $\mathtt{S}_r$`addr` is also called a reply address. When `AR1` receives a packet from `AR2` with $\mathtt{S}_r$`addr` address, `AR1` can not know that this packet will be sent to `src-host` which sent the packet at the beginning to `AR1` without the mapping table address. We also assume that the communication between `src-host` and `dst-host` is confidential because $\mathtt{S}_r$`addr` and ($R_{1,2,3,4}$ + Current-time (s)) are sent encrypted in the payload of the packet.

5.4 uOTA Address Substitution/Recovery

Before a communication starts, a connection establishment procedure is launched between the `src-host` and `dst-host` for the selection of a substitution/recovery algorithm. uOTA notations are represented in Table 5.1.

5.5 uOTA Routing Packet

This operation is applied in both hosts, each host informs its partner about its reply address. Note: when routing the first packet, `src-host` uses its original `addr` to send the first packet, `dst-host` uses its original `addr` to receive the first packet. After that, each host has a reply address of the other host ($\mathtt{S}_r$`src`/$\mathtt{S}_r$`dst`), and they work like destination addresses for receiving

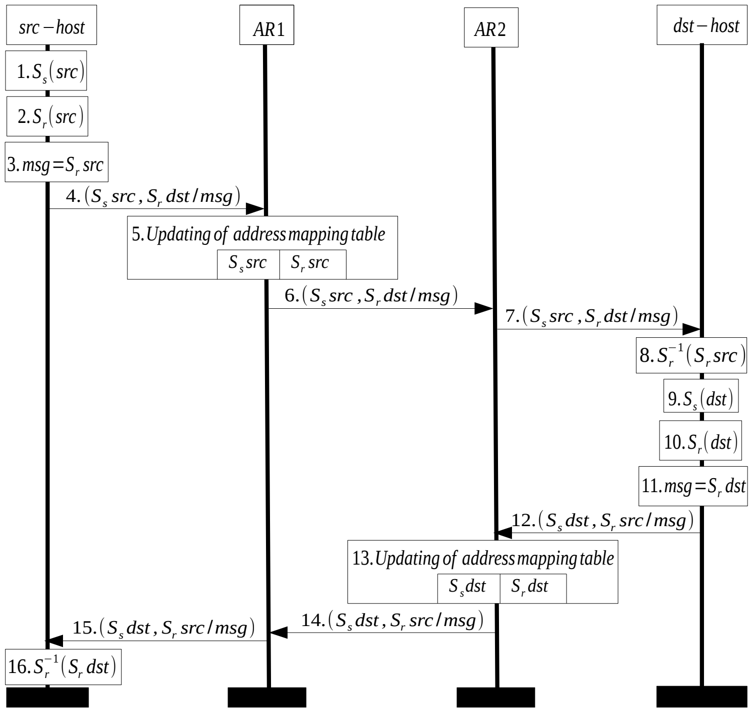


Figure 5.3: uOTA Routing Packets.

the rest of packets. As shown in Fig. 5.3, `src-host` generates its new address to send its new packet (Line 1), and generates its new address to receive a new packet (Line 2). After that, it puts its new reply address in the payload of packet (Line 3), and sends its packet to AR1 (Line 4). AR1 updates the address mapping table for `src_host` (Line 5) and forwards the packet to AR2 (Line 6). AR2 forwards the packet to `dst_host` (Line 7). `dst_host` has to know the original address of its partner, so it recovers the reply address of `src_host` in order to know *addr* (Line 8). `dst_host` undergoes the same phases of `src-host` in Lines (1, 2, 3 and 4) which are the same as in Lines (9, 10, 11 and 12) in `dst-host`. AR2 updates the address mapping table for `dst-host` (Line 13) and forwards the packet to AR1 (Line 14). The latter forwards the packet to `src-host` (Line 15). `src-host` recovers the reply address of `dst-host` (Line 16) to know the original address of `dst_host` in order to complete the communication.

5.6 uOTA Vs OTA

The deference between OTA and uOTA are discussed in table 5.2.

	Privacy protocols	
Criteria	OTA	uOTA
Compatibility	with the Internet	with 6LoWPAN IoT
Address is based	on IPv4	on IPv6
Size of address	28B	16B
Size of packet header	82B	40B
Memory footprint	Big	Small

Table 5.2: The deference between OTA and uOTA.

Packet <number, from, to>	Time (ms)	Source address	Destination address
Packet <P1, patient, doctor>	1500	aaaa :: 212:7402:1:101	aaaa :: 212:7402:2:202
Packet <P2, patient, doctor>	1620	aaaa :: 213:7502:47:25	aaaa :: 124:8402:10:98
Packet <P3, patient, doctor>	1698	aaaa :: 202:87:8:a1	aaaa :: 14:870:ba:47
Packet <P4, doctor, patient>	1874	aaaa :: 74:748:a3:78	aaaa :: 78:a7:12:188
Packet <P5, doctor, patient>	1909	aaaa :: 88:142:78:12	aaaa :: 57:200:87:785

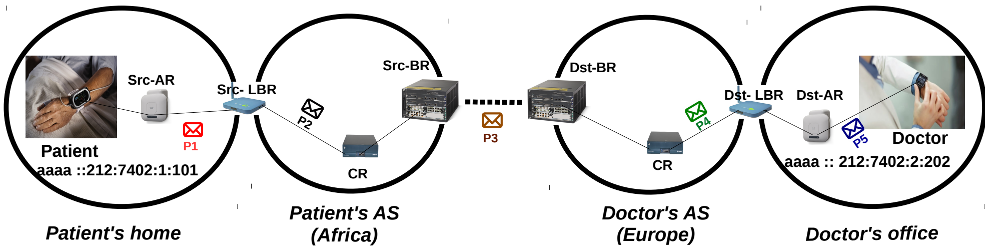


Figure 5.4: E-health scenario uses uOTA to preserve communication privacy by anonymizing the flow identifying fields in 6LoWPAN packet header.

5.7 Application Scenario of uOTA in IoT

E-health is the most sensitive application in the world. Thus, hiding the communication flows in this application is very important and interesting. There are some recent e-health applications which focused on a technical side, however, some other applications focused on protection data healthcare. These applications did not use any protocol which anonymized the e-health communication flow. As we can imagine an e-health IoT threat model, especially e-health communication flow threat model like an attacker who can know communication entities anywhere and anytime. Moreover, the attacker can infer host-flow linkability because the communication entities (e.g. patient and doctor) keep using the same flow identifying information in all packet headers, especially their IPv6 addresses. Fig. 5.4 represents a simple e-health scenario which uses uOTA to preserve communication privacy where the patient is located in his home (6LoWPAN1) in Africa continent (AS1). The doctor is located in his office (6LoWPAN2) in Europe continent (AS2) who diagnoses and analyses the patient over the Internet. Here, both the patient and the doctor use one IPv6 address to send/receive exactly one packet. Thus, the attacker cannot infer that the communication is taking place between the patient and the doctor because we have achieved host-flow unlinkability.

5.8 Conclusion

In this chapter, we presented our first solution which is uOTA. uOTA is based on disguising flow-identifying information such as source and destination addresses. We designed our solution according to the 6LoWPAN architecture to be fully compatible with IoT vision that allows full compatibility with the existing TCP/IP Internet.

Chapter 6

Anonymizing Communication Flow Identifiers: ACFI

There are two types of people who will tell you that you cannot make a difference in this world: those who are afraid to try and those who are afraid you will succeed.

Ray Goforth

Contents

6.1 Introduction

In this chapter, we propose Anonymizing Communication Flow Identifiers (ACFI) [108]. In this approach, we consider a global solution that takes into account anonymizing both source and destination addresses and port numbers as well as source MAC addresses to make it difficult for attackers not only to de-anonymize sources or destinations but also to find relationships between them. The main contributions of this approach are the following: (i) Achieving source and destination anonymity at layer 2 and layer 3 at the same time, (ii) Achieving source-destination unlinkability, and (iii) Reducing the usage of the energy for achieving lightweight communication privacy protocol. The remainder of this chapter describes ACFI in details.

6.2 Motivation Scenario

We consider an IoT network deployed to perform monitoring, event detection, or surveillance type application. In these kinds of applications, things typically transmit regular messages on some kind of activity. To ensure data confidentiality, security solutions are typically implemented at the application layer and thus leave important information such as flow identifying information transmitted accessible to attackers. In some scenarios, knowing flow identifying information may provide some indication that reveals private information. For example, assume that in a monitoring application the fact that a given thing is sporadically transmitting a series of messages may indicate that there is something happening in the nearby. In the case of healthcare applications this might be revealing that some health-related complications are happening.

In this solution, our goal is to make it difficult for an attacker to determine that a communication between two nodes (typically a thing and a remote server) is taking place. We first anonymize the source address and port number at the sender by using a random pseudonym scheme, and then use a Tor-like network to anonymize the destination address and port number.

For the sake of illustration, let us consider the following scenario. Assume that there is a large number of construction workers operating in a particular area and the health of these workers is being monitored by e-health service providers located somewhere in the global Internet. Assume that the workers are carrying e-health sensors which are connected to the 6LoWPAN gateway which allows them to send their physiological information to the e-health service provider.

Fig. 6.1 represents our motivation scenario where the 6LoWPAN has a single 6LBR as a gateway, wearable devices at workers, and ARs. The 6LoWPAN network can be divided into sets of nodes. These sets are called clusters, there is one AR for each cluster.

Fig. 6.1 shows also two cases of communications: (i) periodic communication or stable frequency which is presented by the green flow, i.e. all workers are sending their physiological information to the health provider periodically (regularly), (ii) event communication or variable frequency which is presented by the red flow, which includes event-driven applications (e.g.

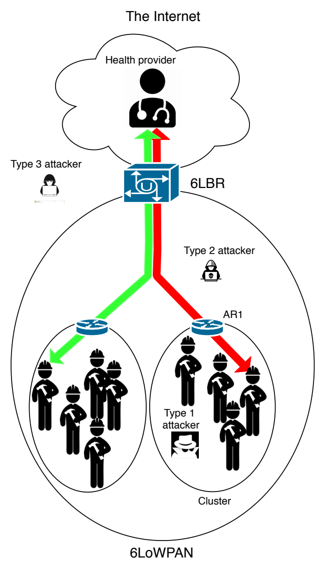


Figure 6.1: Motivation scenario

when an event occurs from a worker X suddenly, the corresponding embedded sensors start sending related physiological information to the health provider non periodically (irregularly). As it can be seen in Fig. 6.1, we consider three types of attackers as follows:

- Type 1 Attacker: the attacker is in the same cluster where the sender is located,
- Type 2 Attacker: the attacker is in the same 6LoWPAN where the sender is located,
- Type 3 Attacker: the attacker is outside the 6LoWPAN where the sender is located.

To preserve a high level of communication privacy, we need to hide sensitive information such as source MAC as well as source and destination IP addresses such that:

- Hiding the destination MAC and IPv6 addresses inside and outside the 6LoWPAN network: to preserve destination anonymity very well, destination MAC and IPv6 addresses have to be hidden. In both cases (periodic and non-periodic communications), we have to hide the destination MAC (the source and the destination are in the 6LoWPAN network) and IPv6 addresses (the source is inside the 6LoWPAN network, and the destination is outside the 6LoWPAN network) in order to prevent attacker type 1, type 2, and type 3 from knowing with whom the worker is communicating. In our solution, MAC address is a part of IPv6 address. So, when the IP address changes, MAC address will be changed automatically,
- Hiding the source MAC and IPv6 addresses inside and outside the 6LoWPAN network: in periodic-communication, the attacker type1, type 2, and type 3 can infer that workers

are communicating with another side because traffic frequency can be captured (all workers are sending their health information each 8 hours in our motivation scenario), there is no problem if the attacker knows that a communication is taking place with workers and another side regularly (the attacker does not know the destination side because we have to hide the destination MAC and IPv6 addresses inside and outside the 6LoWPAN network), so we do not need to hide the source MAC and IPv6 addresses in the periodic communication, in this case, we preserve only source-destination unlinkability and destination anonymity, i.e. we do not preserve source anonymity, but it is better if we preserve source anonymity too, so we hide the source MAC and IPv6 addresses inside and outside the 6LoWPAN network too. Here, all kinds of attackers can see that the worker X is communicating with another side outside the specific period. In other terms, all attackers (type 1, type 2, and type 3) can infer who is the worker that the event has occurred with. Here, the attacker can broadcast this news to the public, or they can prevent this communication at all in order to do harm on this worker, etc. So, here we have to hide the source MAC and IPv6 addresses against all attackers in order to prevent the attacker's ability to know who is the worker that the event has occurred with and to achieve source-destination unlinkability.

6.3 ACFI Motivations

Based on our related work, following important properties of communication privacy protocol in the IoT is highlighted, all of which have not existed together in any single solution:

1. Source and destination anonymity should be achieved in front of type 1, type 2, and type 3 attackers,
2. Source-destination unlinkability should be robust against all kind of attackers (type 1, type 2, and type 3),
3. Solution has to consume less memory, less energy, and less latency.

ACFI achieves the three above points at the same time by hiding source and destination IPv6 addresses, ports number, as well as MAC addresses inside and outside the 6LoWPAN network.

6.4 ACFI Core Idea

The core idea of ACFI is: (i) anonymizing the source IPv6 address and port number, as well as MAC address at the sender auto-randomly, and (ii) anonymizing the destination address and port number in Tor-like network.

ACFI prevents the three types of attackers (previously mentioned, i.e. type 1, type 2, and type 3) from (a) de-anonymizing source and destination addresses and port numbers, as well as from (b) making linkability between source and destination.

6.5 ACFI Architecture

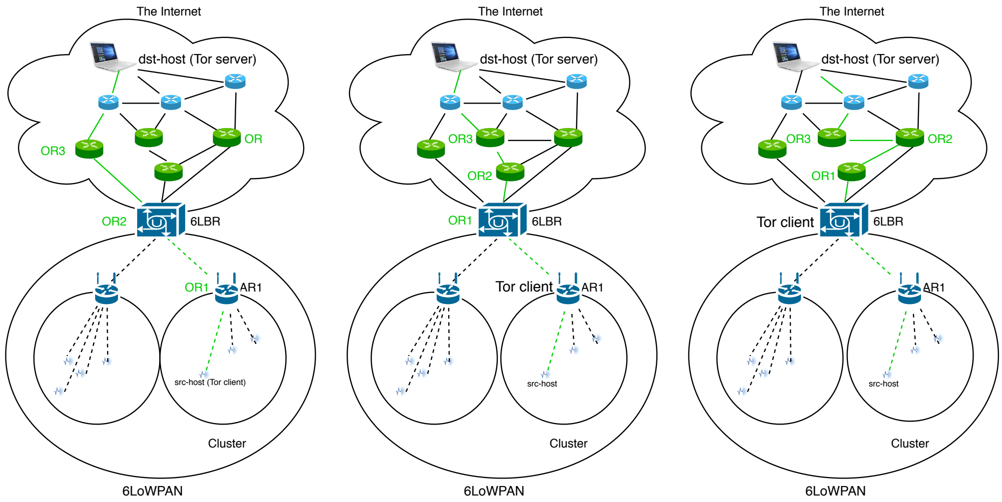
As shown in Fig. 6.2, a typical IoT network according to the 6LoWPAN protocol stack is composed of one 6LBR, number of ARs, and a number of things. A thing is directly linked to one AR that manages incoming and outgoing traffic toward and from the thing linked to it. While 6LBR does not have particular constraints on power and capabilities compared to low power ARs and things, ARs are generally devices with higher capacities compared to other things. The set composed of one AR and the things associated with it can be seen as a cluster.

According to our motivation scenario, the communication is taking place between a source host `src_host` located inside a 6LoWPAN network and a destination host `dst_host` located somewhere in the global Internet outside the 6LoWPAN network. Between these source and destination hosts, we assume that there is a set of nodes forming a Tor-like network. For the sake of simplicity, we consider that the Tor-like anonymizing network is composed of three *Tor-nodes*: OR1, OR2, and OR3. We consider three scenarios depending on the location of Tor-network, particularly where the entry node OR1 is located, which are:

- Scenario 1 (Fig. 6.2a): OR1 is running in an AR1 located inside a 6LoWPAN, Tor client is on `src_host`, and Tor-like network is composed of the following nodes : AR1 (as OR1), 6LBR as (OR2), and OR3,
- Scenario 2 (Fig. 6.2b): OR1 is running in the 6LBR, Tor client is on AR1, and Tor-like network is composed of: 6LBR (as OR1), OR2, and OR3,
- Scenario 3 (Fig. 6.2c): OR1 is outside the 6LoWPAN network, Tor client is on 6LBR, and Tor-like network is composed of OR1, OR2, and OR3.

6.6 ACFI IPv6 Address Structure

As shown in Fig. 6.3, IPv6 addresses used in our solution are composed of two parts: (i) Global Routing Prefix (`Prefix_ID`), and (ii) Interface ID (`IID`) representing the MAC address of the host [109]. In this work, we set that the `Prefix_ID` is on 80b, and `IID` is on 48b. We consider also that the `IID` itself could be composed of two parts which we call: (1) `MAC_ID` (8b), and (2) Node ID (NID) on 40b. As shown in Fig. 6.4, there are two connections between source and destination nodes. We assume that communication time for each session can be discretized into several Time Windows (TWs) as shown in the example of Fig. 6.4. We ensure source address



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Figure 6.2: ACFI Architecture

Prefix_ID	MAC_ID(8b)	NID(40b)
Global routing prefix(80b)	IID is the MAC address (48b)	

Figure 6.3: IPv6 Address Structure in ACFI

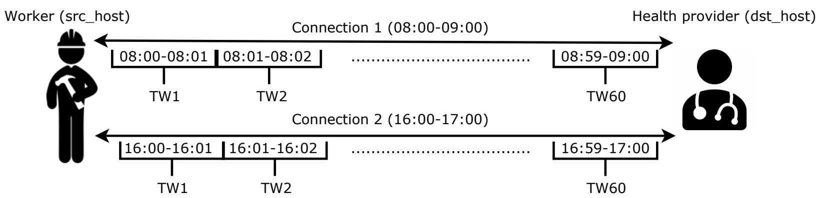


Figure 6.4: Communication between source and destination nodes. In this example, we consider that the source node is the worker and the destination node is the health provider, and that there are two sessions with one hour duration each. We also consider that the first session takes effect from 08:00 to 09:00 and the second session from 16:00 to 17:00. We take an interval of one minute as granularity of time discretization (i.e we take TW = 1 minute).

anonymization by changing source IPv6 address every TW. The new generated IPv6 address for each TW has to be different from previous ones and be unique within the 6LoWPAN. We do this by keeping the same `MAC_ID` in all IPv6 addresses for the same node, and changing only the NID field to get the new IPv6 address which is unique automatically because `MAC_ID` is unique (an example about that will be seen in Fig. 6.16).

6.7 ACFI Session Establishment

For each session, ACFI randomly selects the three anonymizing nodes (OR1, OR2, and OR3) from a set of nodes available on the anonymizing network, which allows the source node to hide its destination address and port number. To generate new source address and port number as well as MAC address randomly, source and destination nodes, AR1, and selected anonymizing nodes use shared seeds like shared keys in the generation technique of the new IPv6 addresses. At the end of session establishment, we find five kinds of shared seeds chosen randomly which are communicated between involved parties in a secure way:

- $seed_{AR}$(AR1,6LBR,nodes): is shared seed between AR1, 6LBR, and nodes associated with it,
- $seed_{nodes}$(i,j) is the shared seed between communicating nodes i and j. It is unique and only known nodes i and j, and AR1. With $seed_{nodes}$(i,j), we prevent type 1 attackers from communication privacy violation. The AR1 uses $seed_{nodes}$(i,j) to generate its new IPv6 address too,
- $seed_{TOR}$(TorClient,OR1) the shared seed between Tor client and OR1,
- $seed_{TOR}$(TorClient,OR2) the shared seed between Tor client and OR2,
- $seed_{TOR}$(TorClient,OR3) which is the shared seed between Tor client and OR3.

Note that when a source node sends a packet with an old IPv6 address, the destination node has to accept it. We assume that there is information about the sequence number of the time window during which it has been transmitted (e.g. TW_1, TW_2, etc). Therefore, the destination host can link the used addresses to the flow of packets.

6.8 ACFI Main Operations

There are two main operations in our proposal which are:

- Obfuscation: is generating new IPv6 address and port number at `src_host` and AR1 for the chosen scenario (i.e., Scenario 2). Our proposal ACFI uses the same value of the five shared seeds in the same session. Shared seeds do not change for each TW, but they only change for each session. During obfuscation operation, ACFI works on the last 40b of the old IPv6 address, i.e. on the NID field. The result of obfuscation is the generation of the new NID of the new IPv6 address, referred to as N_NID. We have:
  - $N\_NID_{src} = O\_NID_{src} + seed_{nodes}(i,j)$,
  - $N\_NID_{AR1} = O\_NID_{AR1} + seed_{AR}(AR1,6LBR,nodes)$,

- $N\_NID_{dst} = O\_NID_{dst} + seed_{TOR}(TorClient, OR3) + seed_{TOR}(TorClient, OR2) + seed_{TOR}(TorClient, OR1)$.

- Recovery: to recover the old IPv6 address or port number for each connection. AR1 and `dst_host` need to recover the O_NID of the old IPv6 address of `src_host` as follows $O\_NID_{src} = N\_NID_{src} - seed_{nodes}(i,j)$. For each connection, each Tor router recovers the old destination IPv6 address. We have:
  - O_NID of `dst_host` at OR1 is $O\_NID_{dst} = N\_NID_{dst} - seed_{TOR}(TorClient, OR1)$,
  - O_NID of `dst_host` at OR2 is $O\_NID_{dst} = N\_NID_{dst} - seed_{TOR}(TorClient, OR2)$,
  - O_NID of `dst_host` at OR3 is $O\_NID_{dst} = N\_NID_{dst} - seed_{TOR}(TorClient, OR3)$.

Note: ACFI hides the source and destination IPv6 addresses and port numbers too. In the rest, we focus on how ACFI hides IPv6 addresses.

6.9 ACFI Routing Packets

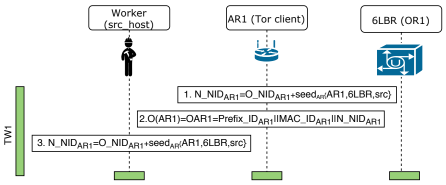
We use the notation presented in Table 6.2 to show the operation of ACFI on an example of two sessions as shown in Table 6.1.

Table 6.1: ACFI properties shown on an example of two sessions

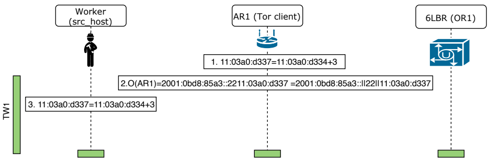
		Session 1	Session 2
src	Prefix_ID	2001:0bd8:85a3::/80	2001:0bd8m85a3::/80
	MAC_ID	8a	8a
	NID	2e:0370:7334	2e:0370:7334
	MAC address	8a2e:0370:7334	8a2e:0370:7334
	IPv6 address	2001:0bd8:85a3::8a2e:0370:7334/80	2001:0bd8:85a3::8a2e:0370:7334/80
AR1	Prefix_ID	2001:0bd8:85a3::/80	2001:0bd8:85a3::/80
	MAC_ID	22	22
	NID	11:03a0:d334	11:03a0:d334
	MAC address	2211:03a0:d334	2211:03a0:d334
	IPv6 address	2001:0bd8:85a3::2211:03a0:d334/80	2001:0bd8:85a3::2211:03a0:d334/80
OR1	IPv4 address	192.168.12.1	192.168.100.1
	IPv6 address	64:ff9b::c0a8:0c01/96	64:ff9b::c0a8:6401/96
OR2	IPv4	192.168.123.1	192.168.200.1
OR3	IPv4 address	192.168.98.1	192.16s8.14.1
	IPv6 address	64:ff9b::c0a8:6201/96	64:ff9b::c0a8:0e01/96
dst	Prefix_ID	2001:ab73:67db::/80	2001:ab73:67db::/80
	MAC_ID	73	73
	NID	24:5698:0d31	24:5698:0d31
	MAC address	7324:5698:0d31	7324:5698:0d31
	IPv6 address	2001:ab73:67db::7324:5698:0d31/80	2001:ab73:67db::7324:5698:0d31/80
Seeds	$\text{seed}_{\text{nodes}}$(src,dst)	3	7
	seed_{TOR}(AR1,OR1)	2	4
	seed_{TOR}(AR1,OR2)	4	8
	seed_{AR}(AR1,6LBR,devices)	3	7
	seed_{TOR}(AR1,OR3)	5	7

Table 6.2: ACFI Notations

Notation	Description
addr	source/destination IPv6 address
P_i	Packet *i*
IP	Initial Packet
NP	New Packet
src_{IP}	source IPv6 address of IP packet
dst_{IP}	destination IPv6 address of IP packet
src_{NP}	source IPv6 address of NP packet
dst_{NP}	destination IPv6 address of NP packet
NID	Node ID (40b)
O_NID	Old NID, NID at the previous TW
N_NID	New NID
MAC_ID	MAC address ID (8b)
Prefix_ID	Global routing prefix (80b)
$\|\|$	Concatenation operation
$seed_{AR}$(AR1,6LBR, devices)	Shared seed between AR, 6LBR and AR's devices
$seed_{nodes}$(i,j)	Shared seed between i and j nodes, and AR
$seed_{Tor}$(TorClient, OR1 or OR2 or OR3)	Shared seed between TorClient and OR1 or OR2 or OR3
O(addr)	Obfuscation operation on addr IPv6 address
Oaddr=O(addr) O_1addr=Oaddr	Result of obfuscation operation, new IPv6 address
O^{-1}Oaddr	Recovery operation of addr from *Oaddr*
addr=O^{-1}Oaddr addr=$O_1^{-1}(O_1addr)$ $O_i^{-1}O_i addr=O_{i-1}(O_i addr)$	Result of recovery operation for i time, old IPv6 address
<A,B/ M>	Sending the message M from A to B
<src_{NP}, dst_{NP}<src_{IP}, dst_{IP}>>	Encapsulating the initial packet where the source address is src_{IP}, and the destination address is dst_{IP} inside the new packet where its source address is src_{NP}, and its destination address is dst_{NP}.



(a) Operations of generating new AR1 pseudonym.



(b) Example on generating new AR1 pseudonym.

Figure 6.5: Generating new AR1 pseudonym.

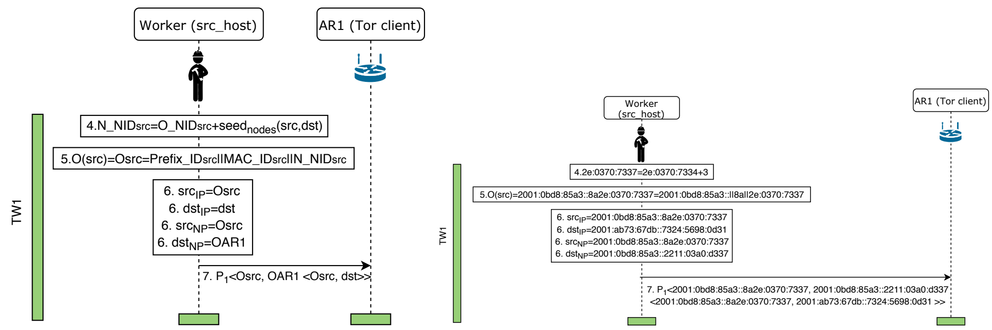
6.9.1 Routing Packets from Source to Destination Hosts

- **From Source Host to AR1:** The AR1 is playing the role of Tor client and the 6LBR is Tor entry node. The AR1 generates a new pseudonym to use for itself or for the node as described in the instructions below:
  - $N\_NID_{AR1} = O\_NID_{AR1} + seed_{AR}(AR1,6LBR,src)$ (line 1).
  - $O(AR1) = OAR1 = Prefix\_ID_{AR1} \,||\, MAC\_ID_{AR1} \,||\, N\_NID_{AR1}$ (line 2)
  - `src_host` generates also the new pseudonym of AR according to $seed_{AR1}$ for sending the packet to it (line 3).

  Generating new AR1 pseudonym at AR1 and source host is illustrated in Fig. 6.5. Fig. 6.5a represents the operations for this and Fig. 6.5b represents an example about it. According to our motivation scenario, preserving more anonymity and unlinkability properties in both kinds of communication (periodic and non-periodic) leads to generating new source and destination communication flows identifiers for each TW in periodic communication, and if an event is done in non-periodic communication.

  Fig. 6.6 (Fig. 6.6a, 6.6b) represents the communication between the source host and the AR1. The source host generates its new IPv6 address every TW according to its old IPv6 address during the previous TW as follows and informs its AR1:
  - $N\_NID_{src} = O\_NID_{src} + seed_{nodes}(src, dst)$ (line 4),
  - $O(Src) = OSrc = Prefix\_ID_{src} \,||\, MAC\_ID_{src} \,||\, N\_NID_{src}$ (line 5),

  For each packet, the source host sets its new IPv6 address (Osrc) as the source address of the initial packet (IP), the destination address of the Initial packet is dst. The source host encapsulates the initial packet (IP) into a new packet (NP). The source address of the new packet src_{NP} is Osrc and the destination address of the new packet dst_{NP} is OAR1. ACFI changes the IPv6 address of the AR1, i.e. ACFI generates OAR1 to prevent type 2 attackers from recognizing to which sub-network (cluster) the source host belongs (line 6). Finally, the source host sends the NP to its AR1 like $P_1 <Osrc,OAR1<Osrc, dst>>$ (line 7). Fig. 6.6a,6.6b shows an example of communication between source host and AR1.



(a) Operations of the communication between Source Host and AR1

(b) Example on the communication between Source Host and AR1

Figure 6.6: Communication between Source Host and AR1

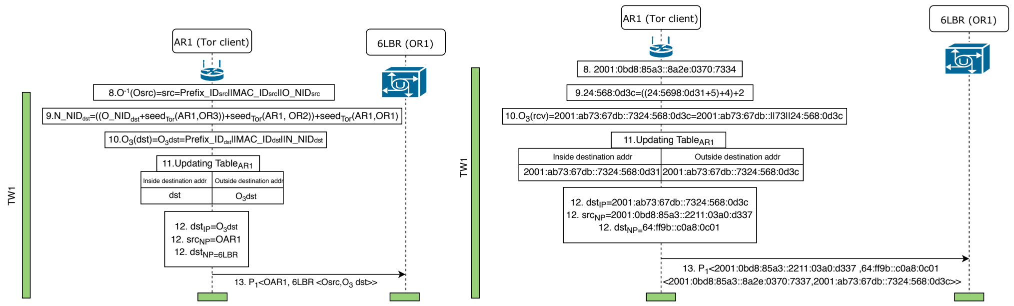
- **From AR1 to 6LBR**: AR1 performs address obfuscation (see Fig. 6.5) to prevent type 2 attackers from recognizing to which sub-network (cluster) the source host belongs. AR1 also performs Tor client operations as shown in Fig. 6.7. For each connection, AR1 recovers the real sender IPv6 address, and it obfuscates the initial destination IPv6 address (dst) three times, i.e. O_3(dst) for hiding the destination address in front of type 2 and type 3 attackers by these instructions:
  - $O^{-1}O_{src}$ = src = Prefix_ID_{src} || MAC_ID_{src} || O_NID_{src} (line 8),
  - $\text{N\_NID}_{dst} = (\text{O\_NID}_{dst} + \text{seed}_{Tor}(\text{AR1,OR3}) + \text{seed}_{Tor}(\text{AR1,OR2}) + \text{seed}_{Tor}(\text{AR1,OR1})$ (line 9),
  - $O_3(dst) = O_3dst = \text{Prefix\_ID}_{dst}$ || MAC_ID_{dst} || N_NID_{dst} (line 10).

  In line 11, AR1 updates its mapping address table which is called Table_{AR}, the latter contains two fields which are the inside and the outside address of destination host, i.e. `dst_host`.

  After that, AR1 changes dst_{IP} with O_3dst, src_{NP} with OAR1, and dst_{NP} with 6LBR (OR1) (line 12). Finally, it sends the NP as follows $P_1 <\text{OAR1, 6LBR} < \text{OSrc}, O_3\text{dst}>>$ (line 13). Fig. 6.7, Fig. 6.7a, and Fig. 6.7b represent the communication between Tor client (AR1) and entry node (6LBR or OR1) followed by an example.

- **From 6LBR to OR2**: the 6LBR recovers O_2dst from O_3dst by the opposite operation of obfuscation as follows:
  - $\text{O\_NID}_{dst} = \text{N\_NID}_{dst}$ - seed_{Tor}(AR1, OR1) (line 14).
  - $O_3^{-1}(O_3dst) = O_2dst = \text{Prefix\_ID}_{dst}$ || MAC_ID_{dst} || O_NID_{dst} (line 15).

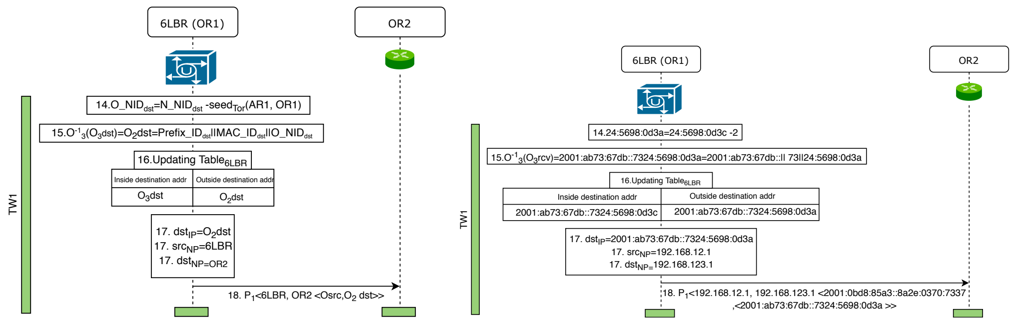
  Table_{6LBR} has to be updated for each connection (line 16). The 6LBR changes dst_{IP} with O_2dst, src_{NP} with 6*LBR*, and dst_{NP} with OR2 (line 17). Finally, it sends the NP as follows



(a) Operations of the communication between AR1 and 6LBR

(b) Example on the communication between AR1 and 6LBR

Figure 6.7: Communication between AR1 and 6LBR



(a) Operations of the communication between OR1 and OR2

(b) Example on the communication between OR1 and OR2

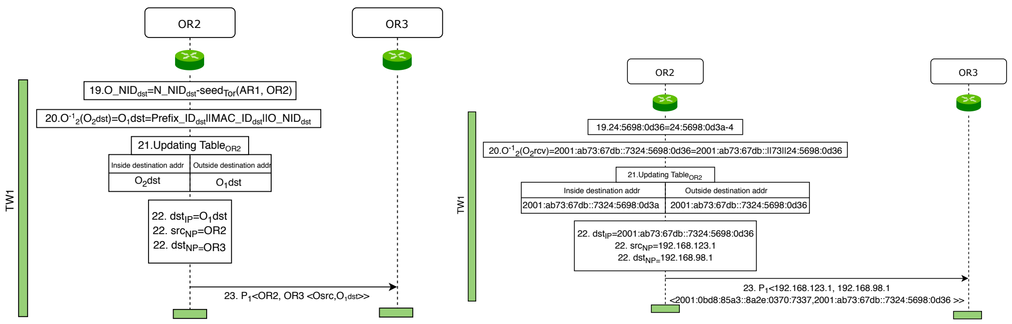
Figure 6.8: Communication between OR1 and OR2

P_1 <6LBR , OR2 < OSrc,O_2dst>> (line 18). The communication between OR1 and OR2 and an example about it are illustrated in Fig. 6.8, Fig. 6.8a, and Fig. 6.8b.

- **From OR2 to OR3**: as shown in Fig. 6.9, Fig. 6.9a, and Fig. 6.9b. The OR2 recovers O_1dst from O_2dst by the following operations:
  - $O\_NID_{dst} = N\_NID_{dst}$ - $seed_{Tor}(AR, OR2)$ (line 19).
  - $O_2^{-1}(O_2dst) = O_1dst = Prefix\_ID_{dst} \,||\, MAC\_ID_{dst} \,||\, O\_NID_{dst}$ (line 20).

  $Table_{OR2}$ is the mapping table address in the OR2. The latter updates $Table_{OR2}$ for each connection (line 21). The OR2 changes dst_{OP} with O_1dst, src_{NP} with OR2, and dst_{NP} with OR3 (line 22). The OR2 sends the NP as follows P_1 <OR2, OR3<OSrc,O_1dst>> (line 23).

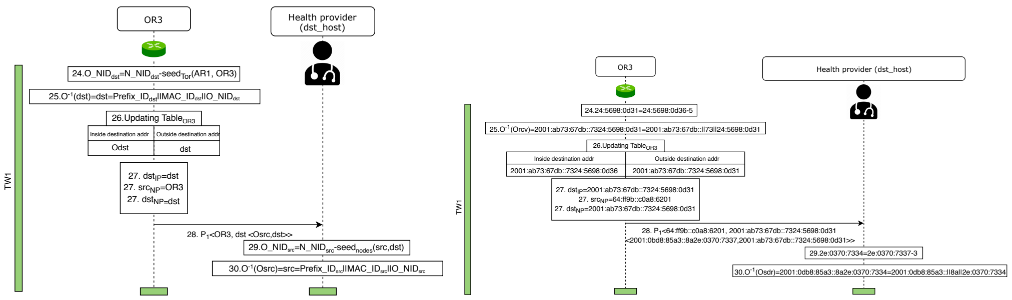
- **From OR3 to Destination Host**: In Fig. 6.10, the OR3 recovers dst from O_1dst as follows:



(a) Operations of the communication between OR2 and OR3

(b) Example on the communication between OR2 and OR3

Figure 6.9: Communication between OR2 and OR3



(a) Operations of the communication between OR3 and Destination host

(b) Example on the communication between OR3 and Destination host

Figure 6.10: Communication between OR3 and Destination host.

- O_NID_{dst} = N_NID_{dst} - $seed_{Tor}$(AR1, OR3) (line 24).
- $O^{-1}(O_1 dst)$ = dst = $Prefix\_ID_{dst}$ || MAC_ID_{dst} || O_NID_{dst} (line 25).

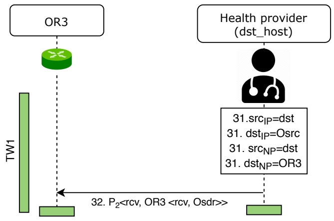
In line 26, $Table_{OR3}$ is updated by the OR3, The OR3 changes dst_{IP} with dst, src_{NP} with OR3, and dst_{NP} with dst (line 27). In line 28, the OR3 sends the NP as follows $P_1 < OR3, dst < OSrc, dst >>$.

The `dst_host` has to know the original sender in e-health application. So, it recovers the original IPv6 address of the sender as follows:

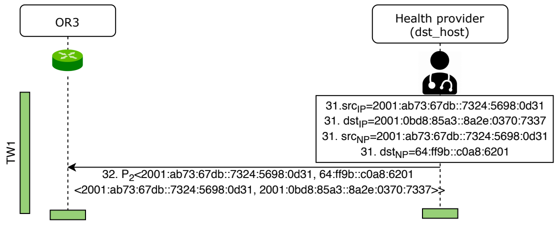
- O_NID_{src}=N_NID_{src}-$seed_{nodes}$(src, dst) (line 29).
- O^{-1}(Osrc) = Src = $Prefix\_ID_{src}$ || MAC_ID_{src} || O_NID_{src} (line 30).

6.9.2 Routing Packets From Destination to Source Hosts

- **From Destination Host to OR3**: as shown in Fig. 6.11, Fig. 6.11a, and Fig. 6.11b, the dst wants to send a reply packet (P_2) to `src_host`, so src_{IP}, dst_{IP},src_{NP}, and dst_{NP} will be

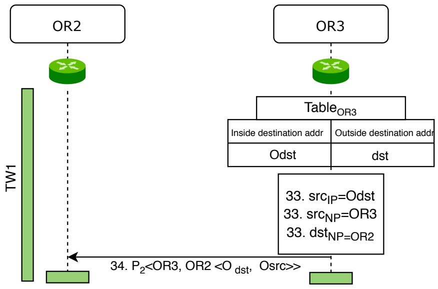


(a) Operations of ACFI routing packet from Destination Host to OR

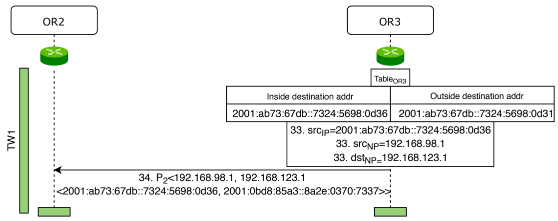


(b) Example on ACFI routing packet from Destination Host to OR3

Figure 6.11: ACFI routing packet from Destination Host to OR3



(a) Operations of ACFI routing packet from OR3 to OR2



(b) Example on ACFI routing packet from OR3 to OR2

Figure 6.12: ACFI routing packet from OR3 to OR2

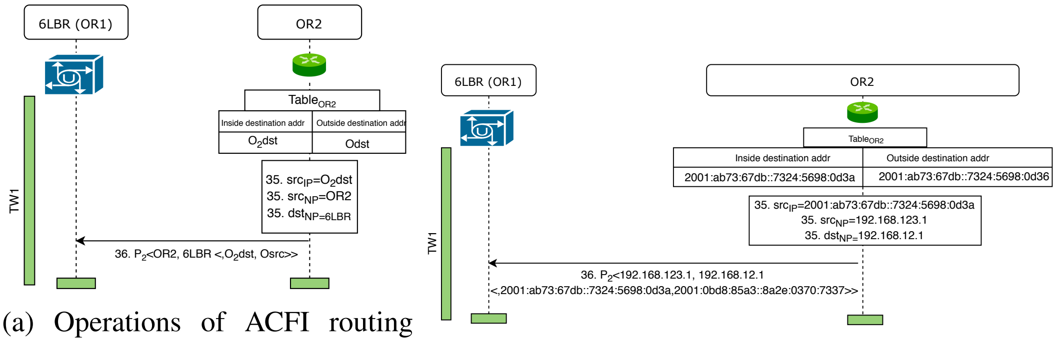
dst, Osrc, dst, and OR3, respectively (line 31). In line 32, the `dst_host` sends the NP as follows $P_2 < dst, OR3 < dst, OSrc >>$.

- **From OR3 to OR2**: Fig. 6.12 represents the communication between OR3 and OR2. Here, the OR3 changes src_{IP} with *Odst* according to its table, i.e. $\text{Table}_{\text{OR3}}$, and src_{NP}, dst_{NP} with OR3, and OR2, respectively (line 33). In line 34, the OR3 sends the replay packet P_2 to OR2 like $P_2 < OR3, OR2 < Odst, OSrc >>$.

- **From OR2 to 6LBR (OR1)**: as shown in Fig. 6.13, the OR2 undergoes the same steps of OR3 in lines (33 and 34) which are the same as in lines (35 and 36) in OR2.

- **From 6LBR to AR1**: the OR1 undergoes the same operations of OR3 and OR2 (see Fig. 6.14). So, the OR2 undergoes the same steps of OR2 in lines (35 and 36) which are the same as in lines (37 and 38) in OR1.

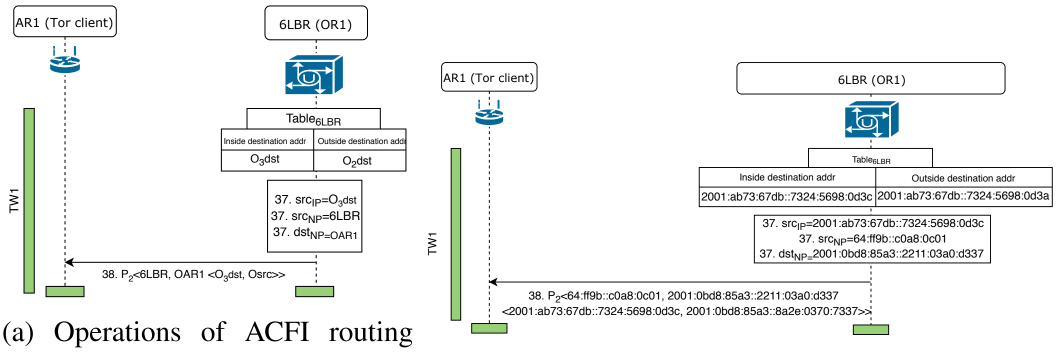
- **From AR1 to Source Host**: this is the last step in our proposal. The AR1 undergoes the same steps of OR1, OR2, and OR3. As shown in Fig. 6.15, the AR1 changes the addresses of the initial and new packet (line 39) in order to send the reply packet P_2 to the worker (line 40).



(a) Operations of ACFI routing packet from OR2 to OR1

(b) Example on ACFI routing packet from OR2 to OR1

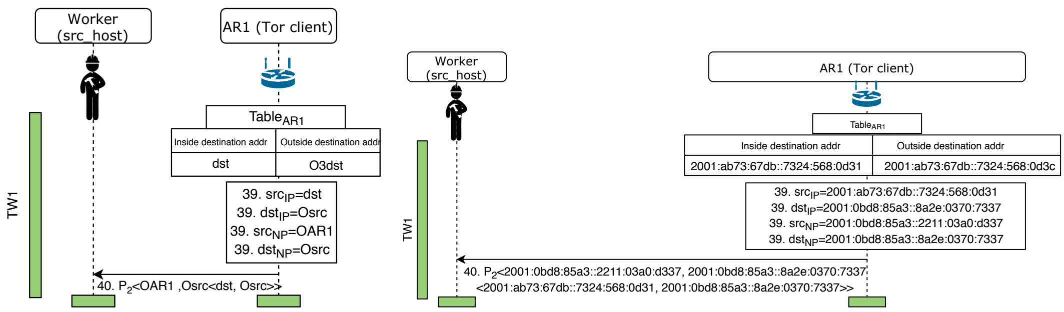
Figure 6.13: ACFI routing packet from OR2 to OR1 (6LBR)



(a) Operations of ACFI routing packet from OR1 to AR1

(b) Example on ACFI routing packet from OR1 to AR1

Figure 6.14: ACFI routing packet from OR1 to AR1



(a) Operations of ACFI routing packet from AR1 to Source Host

(b) Example on ACFI routing packet from AR1 to Source Host

Figure 6.15: ACFI routing packet from AR1 to Source Host

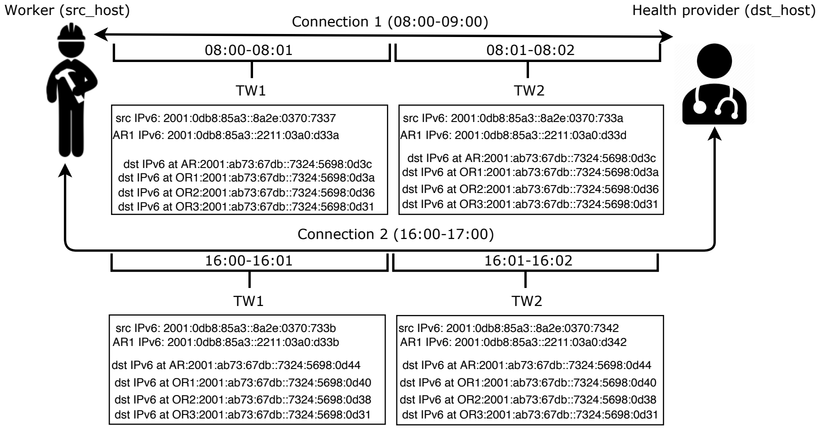


Figure 6.16: Results of ACFI communication between worker and health provider in two different connections

6.10 Conclusion

We have proposed ACFI for 6LoWPAN-based IoT networks. Our solution makes it difficult for an attacker to determine that a communication between two nodes or applications is taking place. ACFI is composed of two parts which guarantee both anonymization of source and destination identifying information such as addresses and port numbers, as well as makes sure that source and destination could not be made linkable to each other while communicating.

Chapter 7

Evaluation and Results

> Writing the perfect paper is a lot like a military operation. It takes discipline, foresight, research, strategy, and, if done right, ends in total victory
>
> *Ryan Holiday*

Contents

7.1 Introduction

In this chapter, we evaluate our two proposals: (i) uOTA (Section 7.2), and (ii) ACFI (Section 7.3) in terms of energy consumption, latency, network overhead. We analyse anonymity and unlinkability properties in the last proposal, i.e. ACFI. In each section, we discuss the results of the simulations. In the end, we summarize this chapter in Section 7.4.

7.2 uOTA Simulations and Evaluation

We evaluate energy consumption, network overhead, latency and privacy-protection metrics with uOTA compared to a uMT6D (we chose the embedded software mode of uMT6D implementation). We implemented both solutions in Cooja and ran simulations with varying numbers of nodes and thus communicating pairs. We set the simulation time to 900s where nodes would have exchanged a significant number of packets. We use Routing Protocol for Low-Power and Lossy Networks (RPL) [110] for routing packets through the simulated 6LoWPANs. We considered the following metrics.

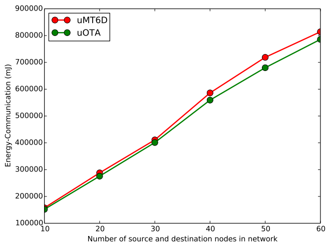
7.2.1 Energy Consumption

Cooja/Contiki OS measure the time spent in active mode (CPU), low power mode (LPM), (TX), and in receive mode (RX) in ticks. The main formula for estimating energy consumption has been presented in formula (7.1) [111].

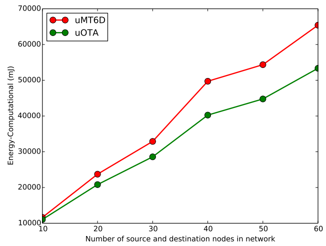
$$E_i = I_i * V_i(V) * T_i(s). \tag{7.1}$$

Where i is the mode $i \in \{\text{CPU, LPM, TX, and RX}\}$ and E_i (resp. I_i, V_i, T_i) is the energy (resp. current, tension, time) used in mode i. Note that to get the elapsed time in seconds in Cooja, we need to convert time from ticks into seconds where 32768 ticks are generated per second. In our simulations, we used the values provided in Contiki 3.0 for sky mode sensor, which are V_i=3V, I_i = 1.8mA, 0.0545mA, 17.7mA, and 20mA for i =CPU, LPM, TX, and RX, respectively.

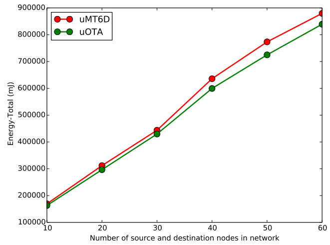
For the energy consumed in communication, we observe that uOTA and uMT6D have approximately the same communication-energy consumption (see Fig. 7.1a). This is so because their main operations consists in repeatedly changing `src-addr` and `dst-addr`, i.e. computation instead of communication. However, for the energy consumed in computation, as shown in Fig. 7.1b, uOTA consumes less energy than uMT6D. This is due to many operations performed by uMT6D, we mentioned these operations in related work where uMT6D applies the same MT6D operations. In addition, uMT6D calculates the new IPv6 `src-addr` and `dst-addr` by SHA-256 hash algorithm which is more energy consuming than the operations performed by uOTA. By considering both, the energy consumed in communication and that consumed in computation, we show that uOTA still offers an advantage compared to uMT6D, as shown in Fig. 7.1c.



(a) Energy Spent in Communication.



(b) Energy Spent in Computation.



(c) Total Energy.

Figure 7.1: Energy Consumption.

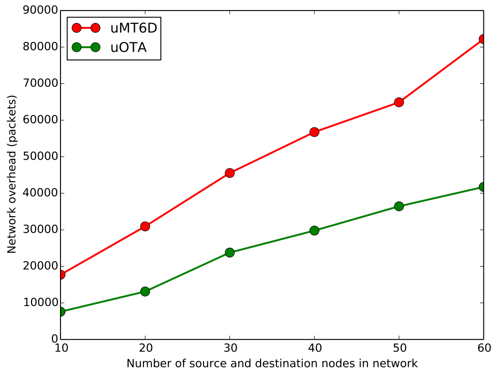


Figure 7.2: Network Overhead.

7.2.2 Network Overhead

We estimate the network overhead by counting the number of packets generated by each simulated protocol during the simulation run, we consider all packets generated by `src-host` and `dst-host`. As shown in Fig. 7.2, uOTA network overhead is smaller than uMT6D network overhead. The reason is that uMT6D needs more packets to encapsulate the IPv6 packets in the uMT6D packets unlike in uOTA.

7.2.3 Latency

Latency is the expression of the elapsed time between sending a packet from the `src-host` to its reception by the `dst-host`. We estimate the latency of packets for all source and destination nodes in the network. The result in Fig. 7.3 shows that the latency of packets in uOTA is smaller than that of uMT6D. The operation that is used by uMT6D to increase the latency compared to uOTA is the encapsulation and the decapsulation of the uMT6D packet.

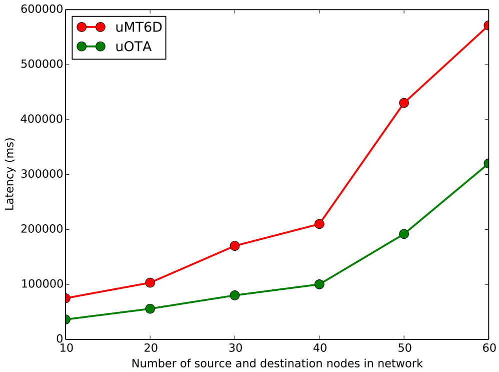


Figure 7.3: Latency.

7.2.4 Privacy-Preservation

To estimate the extent of privacy preservation with uOTA compared to other solutions such uMT6D, we define a simple privacy preserving metric that allows us to measure the quantity of attackers that are able to identify part of the flow identifying information. We present our metric about Privacy Preservation in the IoT (PP_{IoT}) in formula 7.2.

$$PP_{\mathrm{IoT}}(\%) = \frac{\sum_{i=1}^{2}(ND_i * PP_{\mathrm{6LoWPAN}}) + \sum_{j=3}^{5}(ND_j * \alpha)}{\sum_{k=1}^{5} ND_i} \tag{7.2}$$

Where: *ND* (devices): number of devices; i: 6LoWPAN1 network/6LoWPAN2 network; j: AS1/elsewhere in the global Internet/AS2; k: i/j; α (%): the privacy-preservation in j; we suppose that α is a rate between 85% and 95%; and PP_{6LoWPAN} (%): we estimate the Privacy-Preservation inside 6LoWPAN1 network/6LoWPAN2 network (PP_{6LoWPAN}) by formula 7.3.

$$PP_{\mathrm{6LoWPAN}}(\%) = TPP_{\mathrm{6LoWPAN}} - PL_{\mathrm{6LoWPAN}}. \tag{7.3}$$

Where: TPP_{6LoWPAN} (%): Total Privacy Preservation in 6LoWPAN network, so $TPP_{\mathrm{6LoWPAN}} =$ 100%; and PL_{6LoWPAN} (%): is the Privacy Loosing in 6LoWPAN network. From the results shown in Fig 7.4 where the number of devices in AS1, AS2 and elsewhere in the global Internet is set to 800 devices, we can see that privacy protection with uOTA is better than uMT6D. This is because uOTA prevents an attacker from discovering flow-identifying information (by setting for each sent or received packet only one address). However, in uMT6D, an attacker can still infer relevant flow-identifying information during the interval time in a communication session. We also underline that when the number of devices is scaled up in 6LoWPAN1 and 6LoWPAN2 networks, a decline in privacy-protection for both protocols is observed. We assumed above that the privacy-protection is almost well-protected outside 6LoWPAN networks (α) where the privacy-protection is measured by PP_{6LoWPAN} inside the 6LoWPAN networks.

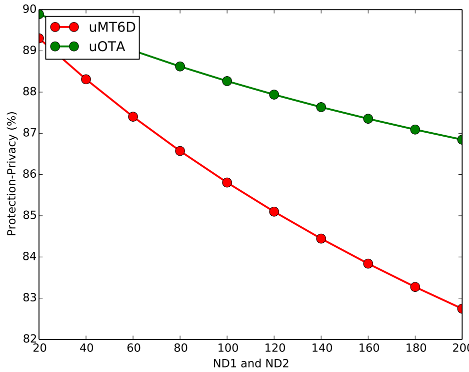


Figure 7.4: Privacy-Protection.

Table 7.1: Simulation parameters

Simulation parameter	Value
Simulation time	900s
Time window	60s
Kind of network	6LoWPAN
Radio driver	cc2420
Radio model	UDGM (Unit Disk Graph Medium)
Radio frequency	2.4 Ghz
Simulation area	100m x100m
Data rate	250 kbps
Mote transmission range	50m
Size of the pool in PeDAAC	10

7.3 ACFI Simulations, Analysis and Evaluation

Our simulation scenario includes several source hosts, destination hosts, ARs, 6LBR, and Tor routers. We simulate PeDAAC and ACFI on its three scenarios according to where the OR1 is located, i.e. ACFI-Scenario 1, ACFI-Scenario 2, and ACFI-Scenario 3 with Cooja simulator on Contiki OS [112]. RPL [110] is the routing protocol in each communication privacy protocol, i.e. ACFI and PeDAAC. Type 1, type 2, and type 3 attackers have been simulated in our experiments.

7.3.1 Settings and Metrics

Simulation parameters of our simulation are represented in Table 7.1. Nodes are placed randomly in the simulation area which is 100m*100m. We consider the following performance metrics:

- Network overhead: is the number of additional packets generated by the system [20],
- Memory footprint with MSP430 micro-controller (ROM and RAM usage). We use similar settings in papers [113, 20].
- Energy: Cooja measures the time spent in active mode, low power mode, transmit mode, and in receive mode in ticks. The main formula for estimating energy consumption has been presented in (7.4) [111, 113].

$$E_i = I_i * V_i * T_i. \tag{7.4}$$

  i is the mode $i \in \{\text{CPU, LPM, Tx, and Rx}\}$ and E_i (resp. I_i, V_i, T_i) is the energy (resp. current, tension, time) used in mode i. Note that to get the elapsed time in seconds in Cooja, we need to convert time from ticks into seconds where 32768 ticks are generated per second. In our simulations, we used the values provided in Contiki 3.0 for sky mote, which are V_i=3V, I_i = 1.8mA, 0.0545mA, 17.7mA, and 20mA for i =CPU, LPM, Tx, and Rx, respectively,
- Latency: is the time elapsed from sending the packet from the sender to its reception by the receiver [20].

7.3.2 Performance Evaluation

We evaluate the performance of ACFI-Scenario 1, ACFI-Scenario 2, ACFI-Scenario 3, and PeDAAC protocols by changing the number of nodes in the 6LoWPAN network from 10 to 100 nodes. We explain our simulation results on our simulation metrics as follows:

Network overhead

As shown in Fig. 7.5, ACFI network overhead in the three scenarios is the same, and it is very smaller than in PeDAAC. The reason is that PeDAAC sends more additional packets that contain information about changing addresses, i.e. CS_{6LBR}, CS_{AR}, etc. in some cases during the connection unlike in ACFI. These cases can be: (i) the mobility of nodes from cluster to cluster in the same 6LoWPAN network, or (ii) failure of AR1 or device. However, these cases cause changing CS_{6LBR} and CS_{AR} and the change of these seeds cause adding more network overhead, i.e. if CS_{6LBR} or CS_{AR} change, the 6LBR will send additional packets to all ARs, and each AR will send additional packets to nodes associated with it. As it can be seen in Fig. 7.5, there is a variation of network overhead in PeDAAC solution, i.e. increments and decrements at the network overhead. This variation has a relation with the occurrence of the case (i) or (ii). In our proposal, at the beginning of each connection, ACFI sends additional packets that contain the shared seeds ($seed_{nodes}$(i,j), $seed_{TOR}$(TorClient, OR1), $seed_{TOR}$(TorClient, OR2), $seed_{TOR}$(TorClient, OR3), and $seed_{AR}$) only once. In other words, ACFI does not send any additional packets for each time window or even the occurrence of the case (i) or (ii). This is what led to the reduction of overhead in our solution.

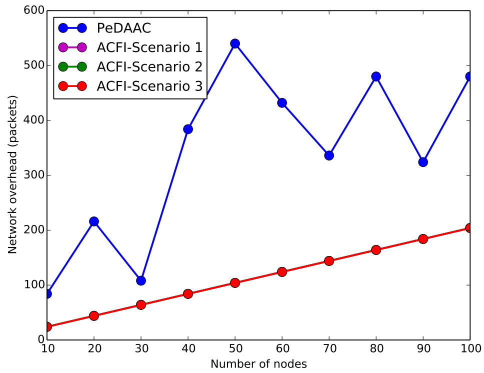


Figure 7.5: Network overhead in PeDAAC, ACFI-Scenario 1, ACFI-Scenario 2, and ACFI-Scenario 3 protocols

Memory footprint with MSP430-size micro-controller

For SkyMote Things, the maximum usage of RAM and ROM are 10KB, 48KB, respectively. According to Fig. 7.6 and Table 7.2:

- ACFI-Scenario 1: needs 6832B of RAM from 10KB, and 44654B of ROM from 48KB,
- Both of ACFI-Scenario 2 and ACFI-Scenario 3: need 6830B of RAM from 10KB, and 44646B of ROM from 48KB,
- PeDAAC: uses 6884B of RAM from 10KB, and 44824B of ROM from 48KB.

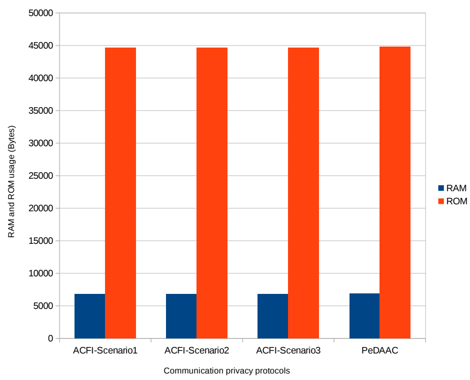


Figure 7.6: RAM and ROM usage

The memory footprint in the communication with ACFI in the three scenarios is approximately the same in the communication with PeDAAC memory footprint.

Table 7.2: RAM and ROM usage

Protocol	RAM (Byte)	ROM (Byte)
ACFI-Scenario1	6832	44654
ACFI-Scenario2	6830	44646
ACFI-Scenario3	6830	44646
PeDAAC	6884	44824

Energy

Energy consumption is shown in Fig. 7.7. We start to measure the energy consumed by the CPU Fig. 7.7a, the one in LPM mode in Fig. 7.7b, in Tx mode Fig. 7.7c, and in Rx mode Fig. 7.7d. The total energy consumption is shown in Fig. 7.7e. The latter shows that ACFI-Scenario 2, ACFI-Scenario 3, and PeDAAC consume approximately the same total energy, but ACFI-Scenario 1 consumes more total energy compared to them. The reason is that the source host in ACFI-Scenario 1 hides the source IPv6 address, and it plays the role of Tor client at the same time.

Latency

We measure the latency in milliseconds (ms). The results in Fig. 7.8 explains that ACFI-Scenario 1 latency and PeDAAC latency is approximately the same. ACFI-Scenario 3 latency is bigger than in ACFI-Scenario 1, ACFI-Scenario 2, and PeDAAC due to Tor-like network as the latency has a relation with the size of the routing path in Tor-like network. In ACFI-Scenario 3, the size of the routing path is bigger than in ACFI-Scenario 2, and ACFI-Scenario 1.

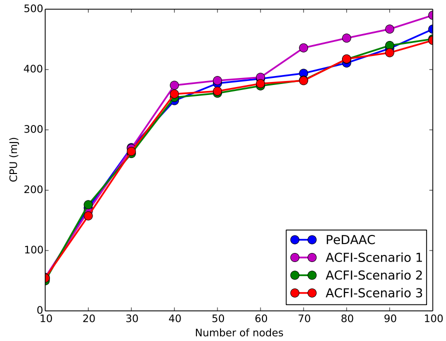
Analysis

In this section, we define two analytical models which are anonymity and unlinkability for analytical evaluation. In ACFI solution, type 1, type 2, or type 3 attackers correlate the old source and destination IPv6 addresses with the new IPv6 addresses to breach source and destination anonymity and source-destination unlinkability.

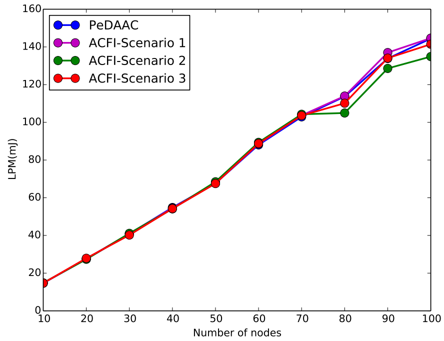
Anonymity Model

According to [114, 115], anonymity is making sure that an attacker is a target host inside an anonymity set. An attacker assigns for each source host i a probability p_i that this source host is the originator of a given packet. Assume that the set of all source nodes is $\mathscr{S}$, therefore, for a given packet, $\sum_{i \in \mathscr{S}} p_i = 1$. Source (or destination) Anonymity is the percentage of source (or destination) anonymity achieved by the system, where its value is calculated for each packet and is equal to $\sum_{i \in \mathscr{S}} p_i log_2(p_i)/log_2(|\mathscr{S}|)$.

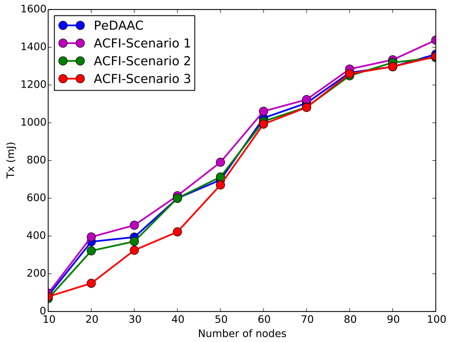
Unlinkability Model We consider measuring unlinkability in a similar way proposed in [115]



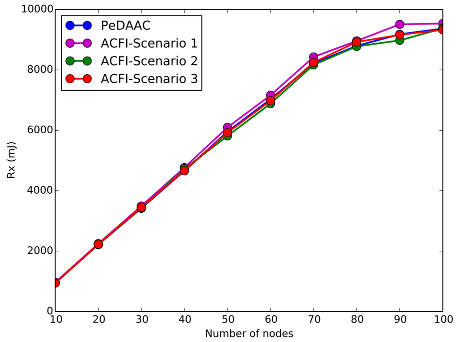
(a) Energy spent in CPU mode



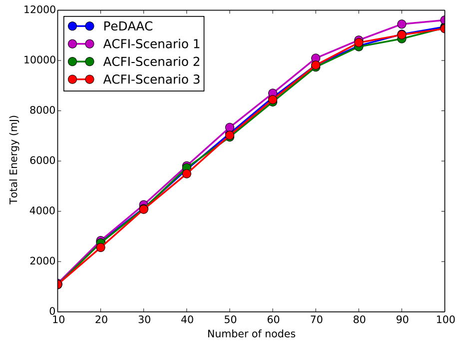
(b) Energy spent in LPM mode



(c) Energy spent in Tx mode



(d) Energy spent in Rx mode



(e) Total Energy

Figure 7.7: Energy consumption

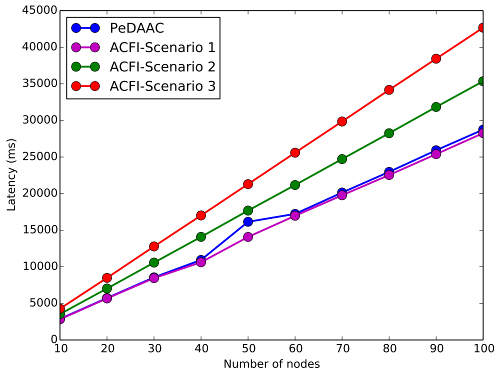


Figure 7.8: Latency (ms)

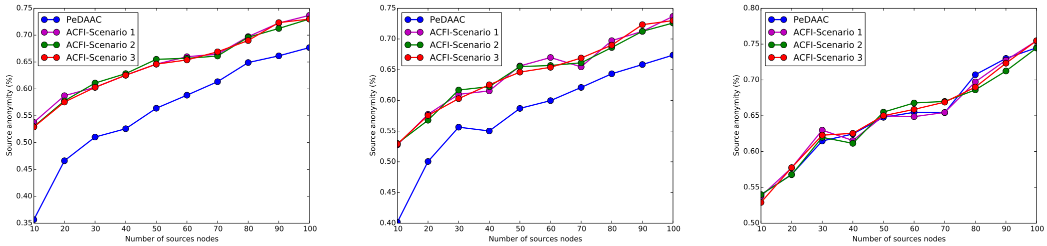
where it is related to the probability of linking that node i is communicating with another node j.

In our context, unlinkability means that it is difficult for an attacker to infer that a source node i is communicating with a destination node j. Similarly to the formula defined earlier for anonymity, and by considering that $p_{(i,j)}$ is the probability that node i is communicating with node j and that the set of all communicating pairs is $\mathscr{C}$, the unlinkability is equal to $\sum_{(i,j)\in\mathscr{C}} p_{(i,j)} log_2(p_{(i,j)})/log_2(|\mathscr{C}|)$

Results Analysis

We consider all type 1, type 2, and type 3 attackers to consider the various cases where an attacker can be located inside the cluster, inside the 6LoWPAN network, or outside the 6LoWPAN network, respectively. Results of analysis of the source anonymity, destination anonymity, and source-destination unlinkability models against type 1, type 2, and type 3 attackers in ACFI-Scenario 1, ACFI-Scenario 2, ACFI-Scenario 3, as well as PeDAAC protocols are explained as follows:

- Source anonymity, as shown in Fig. 7.9:
  - Type 1 attacker: ACFI-Scenario 1, ACFI-Scenario 2, and ACFI-Scenario 3 preserves source anonymity in front of type 1 attacker better than in PeDAAC (see Fig. 7.9a). Type 1 attacker can be a malicious device in the cluster, so it receives also the information of changing addresses like CS_{6LBR} and CS_{AR}, and the source device is located in the 6LoWPAN, especially in one of the cluster in our scenario. Type 1 attackers can infer the real source IPv6 address even if the source device changes its IPv6 address through CS_{6LBR} and CS_{AR} unlike in our proposal. Type 1 attacker can not infer the real source IPv6 address because the shared seed between communication pairs is only known between the source and destination devices,

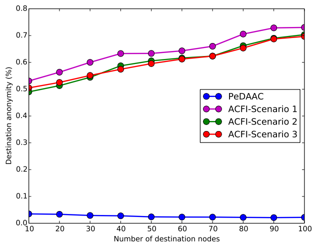


(a) Type 1 attacker (b) Type 2 attacker (c) Type 3 attacker

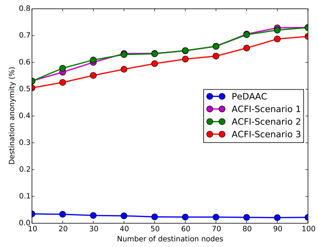
Figure 7.9: Source Anonymity

- Type 2 attacker: it can be a malicious AR, so it also receives the CS_{6LBR}. Type 2 attacker can infer the cluster which the source device located, and for that, source anonymity preservation in PeDAAC is smaller than in ACFI-Scenario 1, ACFI-Scenario 2, and ACFI-Scenario 3 (see Fig. 7.9b),
- Type 3 attacker: in Fig. 7.9c, source anonymity preservation is approximately the same in all protocols, the outsider attacker, i.e. Type 3 attackers can not infer the real source IPv6 address.

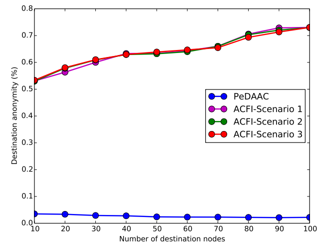
• Destination anonymity: see Fig. 7.10. As shown in Fig. 7.10a,7.10b,7.10c, destination anonymity does not preserve at all in PeDAAC in front of the three kind of attackers because PeDAAC preserves the anonymity of 6LoWPAN-devices (the source device), and in our scenario, the destination device is outside the 6LoWPAN network. In our proposal:
  - Type 1 attacker: in ACFI-Scenario 1, the destination anonymity is achieved very well compared to ACFI-Scenario 2, and ACFI-Scenario 3 because in ACFI-Scenario 1, the source device hides the destination IPv6 address three times, and it encapsulates the original packet which contains the new destination address into a new packet, so type 1 attacker can not infer the real destination address. In ACFI-Scenario 2 and ACFI-Scenario 3, the AR1 hides the destination IPv6 address three times, so the source device sends the new packet where the original packet contains the original destination address. Here, type 1 attacker can infer the original destination IPv6 address if it can recognize the original packet,
  - Type 2 attacker: ACFI-Scenario 1 and ACFI-Scenario 2 achieve destination anonymity very well in front of type 2 attackers compared to ACFI-Scenario 3. In ACFI-Scenario 3, the 6LBR hides the destination address, so type 2 attacker can know the original packet which contains the original destination IPv6 address,
  - Type 3 attacker: destination anonymity is achieved very well outside the 6LoWPAN network like the source anonymity. Type 3 attacker can not know the real destina-



(a) Type 1 attacker

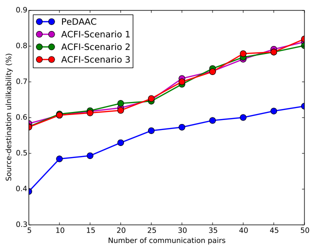


(b) Type 2 attacker

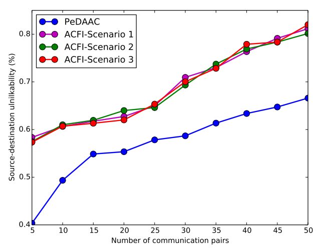


(c) Type 3 attacker

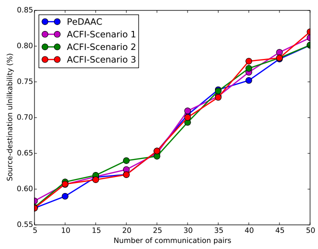
Figure 7.10: Destination Anonymity



(a) Type 1 attacker.



(b) Type 2 attacker.



(c) Type 3 attacker.

Figure 7.11: Source-destination unlinkability.

tion IPv6 address in the three scenarios, i.e. ACFI-Scenario 1, ACFI-Scenario 2, and ACFI-Scenario 3.

- Source-destination unlinkability:
  - Type 1 attacker: ACFI-Scenario 1, ACFI-Scenario 2, and ACFI-Scenario 3 achieve approximately the same level of source-destination unlinkability bigger than PeDAAC, because our proposal achieves source and destination anonymity better than in PeDAAC,
  - Type 2 attacker: the same explanation for the previous point,
  - Type 3 attacker: when the source anonymity is achieved very well outside the 6LoWPAN network in PeDAAC, type 3 attackers can not infer the relationship between communication pairs. Moreover, ACFI achieves source and destination anonymity very well compared to PeDAAC, so it preserves source-destination unlinkability very well.

7.4 Conclusions

We summarize the main points of simulation results on our two solutions are follow:

- uOTA: we evaluated the performance of our solution with uMT6D and shown that our solution achieves higher levels of privacy with lower costs in terms of energy consumption and reduced communication delays,
- ACFI: in the end, we evaluated our proposal against three attacks depending on where the attacker and the Tor entry point are located. We evaluated the network performance and the privacy-level of the three variants in terms of Anonymity, Unlinkabilily. Simulation results showed that ACFI achieves significant improvement over state-of-the-art proposals.

Chapter 8

Conclusions and Future Works

> Don't carry your mistakes around with you. Instead, place them under your feet and use them as stepping stones
>
> *Jay Shetty*

Our main contributions in this thesis consist of a comprehensive review of the state-of-the-art with a new taxonomy and two original solutions on communication privacy-preserving for 6LoWPAN-based IoTs networks. Our solutions (uOTA and ACFI) make it difficult for an attacker to determine that a communication between two nodes or applications is taking place. We summarize our contributions as follows:

- State-of-the-art Review: we shed light on the relevant 6LoWPAN-based solutions for preserving communication identifiers privacy. We have provided a comprehensive coverage of communication identifiers privacy problems and presented the design guidelines that have been the most important proposed in the literature focusing on analysing each of them from different aspects including the involved communication layer, protocols, type of attacks considered, application scenarios, validation through prototyping of by simulation, etc. We have concluded that successful solutions need to take into account communication flow preservation at all of link, network, and transport layers and should use effective PPTs for preserving communication privacy such as pseudonymization and obfuscation based on Tor or proxy techniques, and reduce the lifespan of the pseudonyms used,
- uOTA: we have presented uOTA. It is based on disguising source and destination addresses and port numbers as well as the transport mode used. We designed our solution

according to the 6LoWPAN architecture to be fully compatible with IoT vision that allows full compatibility with the existing TCP/IP Internet. We evaluated the performance of our solution with uMT6D and shown that our solution achieves higher levels of privacy with lower costs in terms of energy consumption and reduced communication delays,

- ACFI: it is composed of two parts which guarantee both anonymization of source and destination identifying information such as addresses and port numbers, as well as makes sure that source and destination entities could not be made linkable to each other while communicating. We evaluated our proposal against three attacks depending on where the attacker and the Tor entry point are located. We evaluated the network performance and the privacy-level of the three variants in terms of anonymity, unlinkability, memory footprint, energy consumption, as well as latency. While simulation results showed that ACFI achieves significant improvement over state-of-the-art proposals, it is to be noted that significant efforts should be made to deal with the increase of latency associated with privacy-preserving solutions in general. The problem of reducing latency is an important and challenging one and needs to be carefully addressed particularly in the context of e-health. The problem may be dealt with from various perspectives such as by defining classes of e-health applications with requirements on the maximum latency they may tolerate and aiming at finding the best Tor-like network in terms of the number of nodes as well as their locations meeting the set requirements of each class of applications while maximizing privacy preservation. Other directions may also investigate hardware implementations of privacy preserving primitives such as obfuscation and recovery at intermediate nodes to accelerate the processing of packets along the routing path.

While our solutions have been extensively evaluated by simulation using Cooja and shown good performance compared to state-of-the-art contributions, further evaluation and testing on real testbeds is required for a better refinement and improvement of these solutions.

Bibliography

[1] Zach Shelby and Carsten Bormann. *6LoWPAN*. John Wiley & Sons, Ltd, Chichester, UK, wiley seri edition, nov 2009.

[2] Adnan Ahmed Abi Sen, Fathy Albouraey Eassa, Kamal Jambi, and Mohammad Yamin. Preserving privacy in internet of things: a survey. *International Journal of Information Technology*, 10(2):189–200, 2018.

[3] Ralf C Staudemeyer, Henrich C Pöhls, and Marcin Wójcik. The road to privacy in iot: beyond encryption and signatures, towards unobservable communication. In *2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 14–20. IEEE, 2018.

[4] Mina Deng, Kim Wuyts, Riccardo Scandariato, Bart Preneel, and Wouter Joosen. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering*, 16(1):3–32, 2011.

[5] Christine Hennebert and Jessye Dos Santos. Security protocols and privacy issues into 6lowpan stack: A synthesis. *IEEE Internet of Things Journal*, 1(5):384–398, 2014.

[6] Debasis Bandyopadhyay and Jaydip Sen. Internet of things: Applications and challenges in technology and standardization. *Wireless personal communications*, 58(1):49–69, 2011.

[7] Rolf H Weber and Romana Weber. *Internet of things*, volume 12. Springer, 2010.

[8] Mario Kusek. Internet of things: Today and tomorrow. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 0335–0338. IEEE, 2018.

[9] Mostafa Al-Emran, Sohail Iqbal Malik, and Mohammed N Al-Kabi. A survey of internet of things (iot) in education: Opportunities and challenges. In *Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications*, pages 197–209. Springer, 2020.

[10] Amine Rghioui and Abdelmajid Oumnad. Internet of things: Surveys for measuring human activities from everywhere. *International Journal of Electrical & Computer Engineering (2088-8708)*, 7(5), 2017.

[11] Kiran Dewangan and Mina Mishra Dr. Internet of things for healthcare: A review. *International Journal of Advanced in Management, Technology and Engineering Sciences*, 8(2,526-534), 2018.

[12] Rishika Mehta, Jyoti Sahni, and Kavita Khanna. Internet of things: Vision, applications and challenges. *Procedia computer science*, 132:1263–1269, 2018.

[13] Xiaoyi Cui. The internet of things. In *Ethical Ripples of Creativity and Innovation*, pages 61–68. Springer, 2016.

[14] Micrium team. *Part2: The Thing|Micruim*, 2020 (accessed April, 2020). https://www.micrium.com/iot/thing/.

[15] Statista team. *Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions)*, 2020 (accessed April, 2020). https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

[16] Yuchen Yang, Longfei Wu, Guisheng Yin, Lijie Li, and Hongbin Zhao. A survey on security and privacy issues in internet-of-things. *IEEE Internet of Things Journal*, 2017.

[17] Jiangchuan Liu, Feng Wang, Xiaoqiang Ma, and Zhe Yang. Recent advances in wireless communication protocols for internet of things. *Wireless Communications and Mobile Computing*, 2017, 2017.

[18] Phillip A Laplante and Nancy Laplante. The internet of things in healthcare: Potential applications and challenges. *IT Professional*, (3):2–4, 2016.

[19] Wang Xi and Luo Ling. Research on iot privacy security risks. In *2016 International Conference on Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*, pages 259–262. IEEE, 2016.

[20] Kosmas Kritsis, Georgios Z Papadopoulos, Antoine Gallais, Periklis Chatzimisios, and Fabrice Theoleyre. A tutorial on performance evaluation and validation methodology for low-power and lossy networks. *IEEE Communications Surveys & Tutorials*, 20(3):1799–1825, 2018.

[21] Naser Hossein Motlagh, Mahsa Mohammadrezaei, Julian Hunt, and Behnam Zakeri. Internet of things (iot) and the energy sector. *Energies*, 13(2):494, 2020.

[22] Aanal Chokshi and Shivam Patel. Internet of things (iot) iot architecture security challenges in iot & role of iot in healthcare industry. *IJETSR*, 4:822–825, 2017.

[23] Farshad Firouzi, Bahar Farahani, and Mahdi Nazm Bojnordi. The smart “things” in iot. In *Intelligent Internet of Things*, pages 51–95. Springer, 2020.

[24] Surapon Kraijak and Panwit Tuwanut. A survey on iot architectures, protocols, applications, security, privacy, real-world implementation and future trends. 2015.

[25] Bhanu Chander and Gopalakrishnan Kumaravelan. Internet of things: Foundation. In *Principles of Internet of Things (IoT) Ecosystem: Insight Paradigm*, pages 3–33. Springer, 2020.

[26] Hany F Atlam and Gary B Wills. Iot security, privacy, safety and ethics. In *Digital Twin Technologies and Smart Cities*, pages 123–149. Springer, 2020.

[27] Andrew Whitmore, Anurag Agarwal, and Li Da Xu. The Internet of Things—A survey of topics and trends. *Information Systems Frontiers*, 17(2):261–274, 2015.

[28] Ammar Rayes and Samer Salam. Internet of things—from hype to reality. *The road to Digitization. River Publisher Series in Communications, Denmark*, 49, 2017.

[29] Balwinder Kaur and Vijay Dhir. Internet of things: Vision, challenges and future scope. *International Journal of Advanced Research in Computer Science*, 8(4), 2017.

[30] Amit Kumar Sikder, Giuseppe Petracca, Hidayet Aksu, Trent Jaeger, and A Selcuk Uluagac. A survey on sensor-based threats to internet-of-things (iot) devices and applications. *arXiv preprint arXiv:1802.02041*, 2018.

[31] T Joshva Devadas and R Raja Subramanian. Paradigms for intelligent iot architecture. In *Principles of Internet of Things (IoT) Ecosystem: Insight Paradigm*, pages 67–100. Springer, 2020.

[32] Dimitrios G Kogias, Emmanouel T Michailidis, Gurkan Tuna, and Vehbi Cagri Gungor. Realizing the wireless technology in internet of things (iot). In *Emerging Wireless Communication and Network Technologies*, pages 173–192. Springer, 2018.

[33] Anil Sharma and Renu Sharma. A review of applications, approaches, and challenges in internet of things (iot). In *Proceedings of ICRIC 2019*, pages 257–269. Springer, 2020.

[34] Bahar Farahani, Farshad Firouzi, Victor Chang, Mustafa Badaroglu, Nicholas Constant, and Kunal Mankodiya. Towards fog-driven iot ehealth: Promises and challenges of iot in medicine and healthcare. *Future Generation Computer Systems*, 78:659–676, 2018.

[35] Muhammad Junaid, Munam Ali Shah, and Imran Abbas Satti. A survey of internet of things, enabling technologies and protocols. In *2017 23rd International Conference on Automation and Computing (ICAC)*, pages 1–5. IEEE, 2017.

[36] T Poongodi, Rajalakshmi Krishnamurthi, R Indrakumari, P Suresh, and Balamurugan Balusamy. Wearable devices and iot. In *A Handbook of Internet of Things in Biomedical and Cyber Physical System*, pages 245–273. Springer, 2020.

[37] Faisal Alsubaei, Abdullah Abuhussein, and Sajjan Shiva. Security and privacy in the internet of medical things: taxonomy and risk assessment. In *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*, pages 112–120. IEEE, 2017.

[38] Debajyoti Misra, Gautam Das, and Debaprasad Das. Review on internet of things (iot): Making the world smart. In *Advances in Communication, Devices and Networking*, pages 827–836. Springer, 2018.

[39] Kai Zhao and Lina Ge. A survey on the internet of things security. In *2013 Ninth international conference on computational intelligence and security*, pages 663–667. IEEE, 2013.

[40] Shancang Li, Li Da Xu, and Shanshan Zhao. The internet of things: a survey. *Information Systems Frontiers*, 17(2):243–259, 2015.

[41] Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243, 2014.

[42] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[43] Zach Shelby and Carsten Bormann. *6LoWPAN: The wireless embedded Internet*, volume 43. John Wiley & Sons, 2011.

[44] Farshad Firouzi, Bahar Farahani, Markus Weinberger, Gabriel DePace, and Fereidoon Shams Aliee. Iot fundamentals: Definitions, architectures, challenges, and promises. In *Intelligent Internet of Things*, pages 3–50. Springer, 2020.

[45] Joseph Bugeja, Andreas Jacobsson, and Paul Davidsson. On privacy and security challenges in smart connected homes. In *2016 European Intelligence and Security Informatics Conference (EISIC)*, pages 172–175. IEEE, 2016.

[46] Farhana Javed, Muhamamd Khalil Afzal, Muhammad Sharif, and Byung-Seo Kim. Internet of things (iot) operating systems support, networking technologies, applications, and challenges: A comparative review. *IEEE Communications Surveys & Tutorials*, 20(3):2062–2100, 2018.

[47] Mohamed Ben-Daya, Elkafi Hassini, and Zied Bahroun. Internet of things and supply chain management: a literature review. *International Journal of Production Research*, 57(15-16):4719–4742, 2019.

[48] Daojing He, Ran Ye, Sammy Chan, Mohsen Guizani, and Yanping Xu. Privacy in the internet of things for smart healthcare. *IEEE Communications Magazine*, 56(4):38–44, 2018.

[49] Sureshkumar Selvaraj and Suresh Sundaravaradhan. Challenges and opportunities in iot healthcare systems: a systematic review. *SN Applied Sciences*, 2(1):139, 2020.

[50] Tatjana M Burkow, Lars Kristian Vognild, Trine Krogstad, Njål Borch, G Ostengen, Astrid Bratvold, and Marijke Jongsma Risberg. An easy to use and affordable home-based personal ehealth system for chronic disease management based on free open source software. *Studies in health technology and informatics*, 136:83–88, 2008.

[51] Qing Yang, Rongxing Lu, Yacine Challal, and Maryline Laurent. Security and privacy in emerging wireless networks. *Security and Communication Networks*, 2017, 2017.

[52] Kewei Sha, Wei Wei, T Andrew Yang, Zhiwei Wang, and Weisong Shi. On security challenges and open issues in internet of things. *Future Generation Computer Systems*, 83:326–337, 2018.

[53] Mauro Conti, Ali Dehghantanha, Katrin Franke, and Steve Watson. Internet of things security and forensics: Challenges and opportunities, 2018.

[54] Delfina Malandrino, Andrea Petta, Vittorio Scarano, Luigi Serra, Raffaele Spinelli, and Balachander Krishnamurthy. Privacy awareness about information leakage: Who knows what about me? In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*, pages 279–284, 2013.

[55] Michael Haus, Muhammad Waqas, Aaron Yi Ding, Yong Li, Sasu Tarkoma, and Jörg Ott. Security and privacy in device-to-device (d2d) communication: A review. *IEEE Communications Surveys & Tutorials*, 19(2):1054–1079, 2017.

[56] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. 2010.

[57] Faiza Loukil, Chirine Ghedira-Guegan, Aïcha Nabila Benharkat, Khouloud Boukadi, and Zakaria Maamar. Privacy-aware in the iot applications: a systematic literature review. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 552–569. Springer, 2017.

[58] Rolf H Weber and Romana Weber. Security and privacy. In *Internet of Things*, pages 41–68. Springer, 2010.

[59] Roger Tourangeau. Confidentiality, privacy, and anonymity. In *The Palgrave Handbook of Survey Research*, pages 501–507. Springer, 2018.

[60] Christine Hennebert and Jessye Dos Santos. Security protocols and privacy issues into 6LoWPAN stack: A synthesis. *IEEE Internet of Things Journal*, 1(5):384–398, 2014.

[61] J Solove Daniel. A taxonomy of privacy. *University of Pennsylvania law review*, 154(3):477–560, 2006.

[62] Noura Aleisa and Karen Renaud. Privacy of the internet of things: A systematic literature review. 2017.

[63] Javier Lopez, Ruben Rios, Feng Bao, and Guilin Wang. Evolving privacy: From sensors to the Internet of Things. *Future Generation Computer Systems*, 75:46–57, 2017.

[64] Emmanouil Vasilomanolakis, Jörg Daubert, Manisha Luthra, Vangelis Gazis, Alex Wiesmaier, and Panayotis Kikiras. On the security and privacy of internet of things architectures and systems. In *2015 International Workshop on Secure Internet of Things (SIoT)*, pages 49–57. IEEE, 2015.

[65] Rosa Sánchez-Guerrero, Florina Almenárez Mendoza, Daniel Diaz-Sanchez, Patricia Arias Cabarcos, and Andrés Marín López. Collaborative ehealth meets security: Privacy-enhancing patient profile management. *IEEE journal of biomedical and health informatics*, 21(6):1741–1749, 2017.

[66] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. 2010.

[67] Ralf C Staudemeyer, Henrich C Pöhls, and Marcin Wójcik. What it takes to boost internet of things privacy beyond encryption with unobservable communication: a survey and lessons learned from the first implementation of dc-net. *Journal of Reliable Intelligent Environments*, pages 1–24, 2019.

[68] Andreas Pfitzmann and Marit Hansen. Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management-a consolidated proposal for terminology. *Version v0*, 31:15, 2008.

[69] Lukas Malina, Gautam Srivastava, Petr Dzurenda, Jan Hajny, and Sara Ricci. A privacy-enhancing framework for internet of things services. In *International Conference on Network and System Security*, pages 77–97. Springer, 2019.

[70] Aurelia Tamò-Larrieux. Technical tools and designs for data protection. In *Designing for Privacy and its Legal Framework*, pages 101–148. Springer, 2018.

[71] Anonymizer Team. *IP Rotation for Commertial Enterprises*, 2020 (accessed April, 2020). https://www.anonymizer.com/.

[72] Shunye Wang, Yanhui Du, Tianliang Lu, Jing Wu, and Tengfei Wang. A survey of anonymous communication methods in internet of things. In *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 627–633. IEEE, 2019.

[73] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. Technical report, Naval Research Lab Washington DC, 2004.

[74] Asma Iman Kouachi and Abdelmalik Bachir. Communication-flow privacy-preservation in 6lowpans-based iot networks. In *International Symposium on Modelling and Implementation of Complex Systems*, pages 33–47. Springer, 2020.

[75] Renzo Efrain Navas. *Improving the resilience of the constrained Internet of Things: a moving target defense approach*. PhD thesis, Ecole nationale supérieure Mines-Télécom Atlantique, 2020.

[76] Renzo E Navas, Frédéric Cuppens, Nora Boulahia Cuppens, Laurent Toutain, and Georgios Z Papadopoulos. Mtd, where art thou? a systematic review of moving target defense techniques for iot. *IEEE Internet of Things Journal*, 2020.

[77] Mahmud Hossain, SM Riazul Islam, Farman Ali, Kyung-Sup Kwak, and Ragib Hasan. An internet of things-based health prescription assistant and its security system design. *Future generation computer systems*, 82:422–439, 2018.

[78] TOR Foundation. *Tor Project Anonymity Online*, 2019 (accessed November, 2019). https://www.torproject.org/.

[79] Saad Saleh, Junaid Qadir, and Muhammad U Ilyas. Shedding light on the dark corners of the internet: A survey of tor research. *Journal of Network and Computer Applications*, 114:1–28, 2018.

[80] Tor team. *Tor Project: Overview*, 2020 (accessed May, 2020). https://2019.www.torproject.org/about/overview.html.en#overview.

[81] Mayank Chauhan, Anuj Kumar Singh, et al. Survey of onion routing approaches: Advantages, limitations and future scopes. In *International conference on Computer Networks, Big data and IoT*, pages 686–697. Springer, 2019.

[82] Anna Engelmann and Admela Jukan. Defying censorship with multi-circuit tor and linear network coding. In *2019 12th CMI Conference on Cybersecurity and Privacy (CMI)*, pages 1–6. IEEE, 2019.

[83] Tor Project Foundation. *org/roadmaps/Tor/IPv6-Tor Bug Tracker & wiki*, 2019 (accessed November, 2019). https://trac.torproject.org/projects/tor/wiki/org/roadmaps/Tor/IPv6.

[84] Tor Project Foundation. *org/roadmaps/Tor/IPv6Features-Tor Bug Tracker & wiki*, 2019 (accessed November, 2019). https://trac.torproject.org/projects/tor/wiki/org/roadmaps/Tor/IPv6Features.

[85] Congxiao Bao, Xing Li, Mohamed Boucadair, Marcelo Bagnulo, and Christian Huitema. Ipv6 addressing of ipv4/ipv6 translators. 2010.

[86] Taeho Lee, Christos Pappas, Pawel Szalachowski, and Adrian Perrig. Communication based on per-packet one-time addresses. In *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, pages 1–10. IEEE, 2016.

[87] Tae-Ho Lee. *Towards an Accountable and Private Internet*. PhD thesis, ETH Zurich, 2017.

[88] Matthew Dunlop, Stephen Groat, William Urbanski, Randy Marchany, and Joseph Tront. Mt6d: A moving target ipv6 defense. In *2011-MILCOM 2011 Military Communications Conference*, pages 1321–1326. IEEE, 2011.

[89] Matthew William Dunlop. *Achieving Security and Privacy in the Internet Protocol Version 6 Through the Use of Dynamically Obscured Addresses*. PhD thesis, Virginia Tech, 2012.

[90] Christopher Frank Morrell. *Improving the Security, Privacy, and Anonymity of a Client-Server Network through the Application of a Moving Target Defense*. PhD thesis, Virginia Tech, 2016.

[91] Nguyen Phong Hoang and Davar Pishva. A tor-based anonymous communication approach to secure smart home appliances. In *2015 17th International Conference on Advanced Communication Technology (ICACT)*, pages 517–525. IEEE, 2015.

[92] Tails Foundation. *Tails-Privacy for anyone anywhere*, 2019 (accessed November, 2019). https://tails.boum.org/index.en.html.

[93] Xiaonan Wang and Yi Mu. Addressing and privacy support for 6lowpan. *IEEE Sensors Journal*, 15(9):5193–5201, 2015.

[94] Robin Snader, Robin Kravets, and Albert F Harris III. Cryptocop: Lightweight, energy-efficient encryption and privacy for wearable devices. In *Proceedings of the 2016 Workshop on Wearable Systems and Applications*, pages 7–12. ACM, 2016.

[95] Luca Brilli, Tommaso Pecorella, Laura Pierucci, and Romano Fantacci. A novel 6lowpan-nd extension to enhance privacy in ieee 802.15. 4 networks. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2016.

[96] Jessye Dos Santos, Christine Hennebert, JC Fonbonne, and Cédric Lauradoux. Ephemeral: Lightweight pseudonyms for 6lowpan mac addresses. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6. IEEE, 2016.

[97] Kimberly Zeitz, Michael Cantrell, Randy Marchany, and Joseph Tront. Changing the game: A micro moving target ipv6 defense for the internet of things. *IEEE Wireless Communications Letters*, 7(4):578–581, 2018.

[98] Xiaonan Wang and Yi Mu. Communication security and privacy support in 6lowpan. *Journal of Information Security and Applications*, 34:108–119, 2017.

[99] Byeong Ok Kwak and Tae Soo Chung. Trust domain based trustworthy networking. In *Information and Communication Technology Convergence (ICTC), 2017 International Conference on*, pages 1247–1259. IEEE, 2017.

[100] Yanina Protskaya. *Security in the internet of things*. PhD thesis, Università degli Studi di Parma. Dipartimento di Ingegneria e architettura, 2020.

[101] Asma Iman Kouachi, Somia Sahraoui, and Abdelmalik Bachir. Per packet flow anonymization in 6lowpan iot networks. In *2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pages 1–7. IEEE, 2018.

[102] Monali Mavani and Krishna Asawa. Privacy enabled disjoint and dynamic address autoconfiguration protocol for 6lowpan. *Ad Hoc Networks*, 79:72–86, 2018.

[103] Monali Mavani and Krishna Asawa. Resilient against spoofing in 6lowpan networks by temporary-private ipv6 addresses. *Peer-to-Peer Networking and Applications*, pages 1–15, 2019.

[104] Jens Hiller, Jan Pennekamp, Markus Dahlmanns, Martin Henze, Andriy Panchenko, and Klaus Wehrle. Tailoring onion routing to the internet of things: Security and privacy in untrusted environments. In *2019 IEEE 27th International Conference on Network Protocols (ICNP)*, pages 1–12. IEEE, 2019.

[105] Thomas Narten, William Allen Simpson, Erik Nordmark, and Hesham Soliman. Neighbor discovery for ip version 6 (ipv6). 2007.

[106] Ye Zhu, Xinwen Fu, Bryan Graham, Riccardo Bettati, and Wei Zhao. On flow correlation attacks and countermeasures in mix networks. In *International Workshop on Privacy Enhancing Technologies*, pages 207–225. Springer, 2004.

[107] Vitaly Shmatikov and Ming-Hsiu Wang. Timing analysis in low-latency mix networks: Attacks and defenses. In *European Symposium on Research in Computer Security*, pages 18–33. Springer, 2006.

[108] Asma Iman Kouachi, Abdelmalik Bachir, and Noureddine Lasla. Anonymizing communication flow identifiers in the internet of things. *Computers & Electrical Engineering*, 91:107063, 2021.

[109] Robert M Hinden and Stephen E Deering. Internet protocol version 6 (ipv6) addressing architecture. 2003.

[110] Tim Winter, Pascal Thubert, Anders Brandt, Jonathan W Hui, Richard Kelsey, Philip Levis, Kris Pister, Rene Struik, Jean-Philippe Vasseur, and Roger K Alexander. Rpl: Ipv6 routing protocol for low-power and lossy networks. *rfc*, 6550:1–157, 2012.

[111] Adam Dunkels, Fredrik Osterlind, Nicolas Tsiftes, and Zhitao He. Software-based online energy estimation for sensor nodes. In *Proceedings of the 4th workshop on Embedded networked sensors*, pages 28–32. ACM, 2007.

[112] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. Contiki-a lightweight and flexible operating system for tiny networked sensors. In *29th annual IEEE international conference on local computer networks*, pages 455–462. IEEE, 2004.

[113] Aleksandar Velinov and Aleksandra Mileva. Running and testing applications for contiki os using cooja simulator. 2016.

[114] Sandra Steinbrecher and Stefan Köpsell. Modelling unlinkability. In *International Workshop on Privacy Enhancing Technologies*, pages 32–47. Springer, 2003.

[115] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In *International Workshop on Privacy Enhancing Technologies*, pages 54–68. Springer, 2002.