Internet of Things Data Contextualisation for Scalable Information Processing, Security, and Privacy

A thesis submitted in fulfilment of the requirements for

the degree of Doctor of Philosophy

Ali Yavari

Master of Communication Systems from KTH Royal Institute of Technology

Bachelor of Information Technology Engineering from IAU

School of Science

College of Science, Engineering and Health

RMIT University

March 2019

Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed. I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Ali Yavari

School of Science
RMIT University
15 March 2019

Abstract

The Internet of Things (IoT) interconnects billions of sensors and other devices (i.e., *things*) via the internet, enabling novel services and products that are becoming increasingly important for industry, government, education and society in general. It is estimated that by 2025, the number of IoT devices will exceed 50 billion, which is seven times the estimated human population at that time. With such a tremendous increase in the number of IoT devices, the data they generate is also increasing exponentially and needs to be analysed and secured more efficiently. This gives rise to what is appearing to be the most significant challenge for the IoT: Novel, scalable solutions are required to analyse and secure the extraordinary amount of data generated by tens of billions of IoT devices. Currently, no solutions exist in the literature that provide scalable and secure IoT scale data processing.

In this thesis, a novel scalable approach is proposed for processing and securing IoT scale data, which we refer to as contextualisation. The contextualisation solution aims to exclude irrelevant IoT data from processing and address data analysis and security considerations via the use of contextual information. More specifically, contextualisation can effectively reduce the volume, velocity and variety of data that needs to be processed and secured in IoT applications. This contextualisation-based data reduction can subsequently provide IoT applications with the scalability needed for IoT scale knowledge extraction and information security. IoT scale applications, such as smart parking or smart healthcare systems, can benefit from the proposed method, which improves the scalability of data processing as well as the security and privacy of data.

The main contributions of this thesis are: 1) An introduction to context and contextualisation for IoT applications; 2) a contextualisation methodology for IoT-based applications that is modelled around observation, orientation, decision and action loops; 3) a collection of

contextualisation techniques and a corresponding software platform for IoT data processing (referred to as contextualisation-as-a-service or ConTaaS) that enables highly scalable data analysis, security and privacy solutions; and 4) an evaluation of ConTaaS in several IoT applications to demonstrate that our contextualisation techniques permit data analysis, security and privacy solutions to remain linear , even in situations where the number of IoT data points increases exponentially.

Acknowledgements

Completion of this thesis was made possible by the support and collaboration of several people. I would like to express my special appreciation and thanks to my supervisor, Professor Dimitrios Georgakopoulos, for his continuous support during my PhD study, his patience, inspiration and immense knowledge. His supervision helped me to learn new things throughout my research and the writing of this thesis. I would like to thank Professor Timos Sellis for making me passionate about this PhD study, introducing Dimitrios to me, co-supervising my PhD, and for his continuous support and mentorship during my research. I also express my gratitude to Professor Heinrich Schmidts for ,co-supervising my PhD, and his inspirational feedback and wonderful ideas that extended my research. A special thanks to my co-supervisors who supported me during my PhD study, including Professor Arkady Zaslavsky, Professor Xun Yi and Dr Surya Nepal. I would like to express my sincere gratitude to Dr Prem Jayaraman and the other collaborators on my publications for everything I have learned from them.

Thank you to my friends and colleagues at RMIT University for making my PhD journey enjoyable. Thanks to Jesse for always distracting me by having irrelevant answers to my relevant questions. Thanks to Marco for all the gelati and picanhas. Special thanks to Josip for proofreading this thesis and taking responsibility for its English and grammar issues.

I would like to acknowledge Farhad for introducing me to the Computer Engineering world by helping me learning my first programming language and for all his unforgettable support.

Last but not least, a very special thanks to my parents, Zohre and Hamid, for their unconditional help, for motivating me to study, for their endless kindness, and for making me what I am. Thanks also to Sadra and Hoda for being my favourite siblings.

Credits

Portions of the material in this thesis have previously appeared in the following peer-reviewed publications:

- Ali Yavari, Prem Prakash Jayaraman, Dimitrios Georgakopoulos, and Surya Nepal. "ConTaaS: An Approach to Internet-Scale Contextualisation for Developing Efficient Internet of Things Applications." In Proceedings of the 50th Hawaii International Conference on System Sciences. pp. 5932-5940. 2017. [IBM/ISSIP Best Paper Award]

- Ali Yavari, Prem Prakash Jayaraman, and Dimitrios Georgakopoulos. "Contextualised service delivery in the internet of things: Parking recommender for smart cities." In Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on, pp. 454-459. IEEE, 2016.

- Ali Yavari, Arezou Soltani Panah, Dimitrios Georgakopoulos, Prem Prakash Jayaraman, and Ron van Schyndel. "Scalable role-based data disclosure control for the internet of things." In Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on, pp. 2226-2233. IEEE, 2017.

- Dimitrios Georgakopoulos, Ali Yavari, Prem Prakash Jayaraman, and Rajiv Ranjan. "Towards a RISC Framework for Efficient Contextualisation in the IoT." In Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on, pp. 1993-1996. IEEE, 2017.

– Arezou Soltani Panah, Ali Yavari, Ron van Schyndel, Dimitrios Georgakopoulos, and Xun Yi. "Context-driven Granular Disclosure Control for Internet of Things Applications." IEEE Transactions on Big Data (2017).

– Prem Prakash Jayaraman, Ali Yavari, Dimitrios Georgakopoulos, Ahsan Morshed, and Arkady Zaslavsky. "Internet of things platform for smart farming: Experiences and lessons learnt." Sensors 16, no. 11 (2016): 1884.

– Prem Prakash Jayaraman, Xuechao Yang, Ali Yavari, Dimitrios Georgakopoulos, and Xun Yi. "Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation." Future Generation Computer Systems 76 (2017): 540-549.

– Tejal Shah, Ali Yavari, Karan Mitra, Saguna Saguna, Prem Prakash Jayaraman, Fethi Rabhi, and Rajiv Ranjan. "Remote health care cyber-physical system: quality of service (QoS) challenges and opportunities." IET Cyber-Physical Systems: Theory & Applications 1, no. 1 (2016): 40-48.

– Reza Soltanpoor and Ali Yavari. "CoALA: Contextualization Framework for Smart Learning Analytics." In Distributed Computing Systems Workshops (ICDCSW), 2017 IEEE 37th International Conference on, pp. 226-231. IEEE, 2017.


watermarking
obfuscation
technique
security
applications
context
parking
disclosure
watermark
privacy
data
different
sensitive
digital
research
services
application
location
smart
access
service
model
sensor
devices
relevant
sensors
control
scalable
Security
system
described
presented
framework
ConTaaS
context-aware
scalability
International
key
user
value
Internet
query
related
provide
vehicle
scale
architecture
contextualisation
processing
contextual

Contents

Glossary

AALFI Ambient Assisted Living Flexible Interface

ABE Attribute Based Encryption

ACE Application Context Engine

ACI Application Context ID

ADL Activities of Daily Living

API Application Programming Interface

AR Augmented Reality

BSN Body Sensor Network

CBD Central Business District

CID Context ID

CoMon Cooperative Context Monitoring System

ConTaaS Contextualisation-as-a-Service

CPU Central Processing Unit

DAC Discretionary Access Control

DBN Dynamic Bayesian Network

DCCI Decentralised Checking of Context Inconsistency

DCE Data Context Engine

DCI Data Context Identifier

DDP Disclosure Decision Point

DES Data Encryption Standard

DL Description Logic

DLFSR Dynamic Linear Feedback Shift Register

DOG Domain Ontology Graph

EAT Effective Assertional Triple

ECC Elliptic Curve Cryptography

FDAC Fine-grained Distributed Access Control

FOL First-order Logic

FPGA Field-Programmable Gate Array

GPS Global Positioning System

GRBAC Generalised Role-based Access Control

HBAS Home and Building Automation System

HDP Hierarchical Dirichlet Processes

IBE Identity Based Encryption

ICT Information and Communications Technology

ID Identification

IDIM Incremental and Distributed Inference Method

IoT Internet of Things

IRI Internationalised Resource Identifier

KMS Key Management System

LBAC Lattice-Based Access Control

LC Linear Complexity

LFSR Linear Feedback Shift Register

LoCCAM Loosely Coupled Context Acquisition Middleware

LSB Least Significant Bits

MAS Multi-Agent System

MQ Multivariate-Quadratic

MRI Mapped Rule Index

OF Obfuscation Function

OODA Observation, Orientation, Decision, and Action

OSGi Open Service Gateway Initiative

OWL Web Ontology Language

PKC Public Key Cryptography

PN Pseudo Noise

POI Point of Interest

PSN Personal Social Network

PSS Personal Smart Space

QoC Quality of Context

RAM Random-Access Memory

RBAC Role-Based Access Control

RDF Resource Description Framework

RFID Radio-Frequency Identification

RQ Research Question

SCIMS Social Context Information Management System

SKC Secret Key Cryptosystem

SLR Systematic Literature Review

SPARQL SPARQL Protocol and RDF Query Language

SQL Structured Query Language

SS Spread Spectrum

SSL Secure Sockets Layer

SSN Semantic Sensor Network

SVM Support Vector Machine

SWE Sensor Web Enablement

SWRL Semantic Web Rule Language

TIF Transfer Inference Forest

TLS Transport Layer Security

TRN Truly Random Number

UTM Universal Transverse Mercator

VANET Vehicular Ad-hoc Network

VID Virtual Identity

VM Vector Machine

XML Extensible Markup Language

CHAPTER 1

Introduction

The Internet of Things (IoT) currently incorporates approximately 15 billion IoT devices, and there are estimates this will grow to 50+ billion by 2020 [1]. IoT devices produce a tremendous amount of data, which we refer to as IoT data. IoT data underpins the development of IoT applications that support many novel products and services that aim to make cities, healthcare, manufacturing (Industry 4.0), energy generation and distribution, and agriculture more data-driven and, therefore, 'smarter'.

Many IoT applications typically follow the Observation, Orientation, Decision, and Action (OODA) loop paradigm [2]. To explain this further, consider an IoT-based application for a smart city that, among other services, finds and recommends parking spaces to drivers who commute to work. At the observation phase this application collects a variety of data that include (1) streamed IoT data from traffic, parking, roadside, and public transport sensors, (2) stored data such as schedules from public transport databases, and (3) real-time IoT data such as location and navigation information from the vehicles and mobile phones of participating drivers. The orientation phase of the parking recommendation service involves contextualising all such information to recommend a specific parking spot to each driver or car. The first step of contextualisation involves filtering out all the IoT data that do not relate to parking recommendation before any further data processing takes place. The next step in the parking-finding orientation involves aggregating the destinations

of the drivers/vehicles and all the remaining IoT parking-related information. The final orientation step correlates the preferences of the driver, such as parking cost and distance from their destination, to a recommendation. Each driver selects a recommendation from those provided, which is set as their destination in their navigation system. Many other IoT applications and related services have some form of OODA loop in their data processing logic.

There are many existing research solutions for context management and contextualisation that can be broadly classified as database techniques, semantic web and rule-based context management approaches, and machine learning and data science-based contextualisation methods. However, most of the existing contextualisation approaches are incompatible as they consider different notions of context and propose heterogeneous and incompatible contextualisation techniques. Furthermore, none of these existing approaches can support data processing at the scale of the IoT, as most are highly inefficient from scalability and performance perspectives.

Another related concern in many IoT applications [3] is how to keep the data collected from the IoT private. Examples of sensitive IoT data include the physiological data collected by wearable or attached bio-medical sensors, and location data collected by the Global Positioning System (GPS) and mobile phones. Disclosure of such data creates opportunities for criminal activity and can result in loss of property, serious harm or even death. Thus, despite its benefits, the IoT presents significant security and privacy challenges, which are exacerbated by the unprecedented scale of IoT devices [4] and the amount of data they generate. Traditionally, such security issues have been addressed with the aid of encryption and privacy preservation techniques. However, IoT devices are extremely limited in computational power and memory resources and therefore these techniques add further data processing challenges [5].

The main aim of this thesis is to propose novel, real-time, IoT contextualisation techniques that use contextual information to significantly speed up the processing of IoT data for data analysis purposes. Furthermore, this thesis will also propose a novel combination of contextualisation and watermarking techniques that provide for both highly

scalable and lightweight role-based access control and data obfuscation techniques to secure IoT data and the privacy of users of the IoT ecosystem.

1.1 Research Questions

The Research Question (RQ)s covered in this thesis are of two main categories:

- Category 1: How can contextualisation improve performance and scalability in IoT data processing?

  RQ1 : How can we characterise the performance and scalability of existing techniques and algorithms that use context in the analysis of IoT scale data?
  RQ1 will review the scalability and performance of the techniques and algorithms investigated in Section 2.2.

- Category 2: How can we utilise contextualisation to improve scalability, security, and privacy in IoT applications?

  RQ2 : How can we perform scalable and performance oriented contextualisation of IoT data?
  For RQ2, we propose a scalable and performance oriented contextualisation technique that can be applied to IoT scale data.

  RQ3 : How can we design a sensor cloud solution for contextualisation of IoT data?
  We aim to explore how we can design an architecture for the proposed contextualisation technique that will be deployed in the cloud environment.

  RQ4 : Can we implement and demonstrate the proposed model by developing a proof-of-concept implementation, and validate its scalability and performance through experiments?
  We will discuss how we can implement the architecture proposed in RQ3. Further, we will run several experiments to evaluate the scalability and performance of the proposed technique.

RQ5 : How we can utilise contextualisation to improve the security and privacy of IoT Scale data?

We will investigate how the proposed contextualisation technique can be applied to IoT scale data to improve security and privacy.

1.2 Methodology

For RQ1, we develop a Systematic Literature Review (SLR) methodology in Section 2.2 to investigate research papers covering data processing approaches that consider context. The SLR outcome including the research gaps and challenges identified, is presented in Section 2.2.4. For RQ2, we propose an IoT data contextualisation technique and describe relevant definitions and contextual operations in Section 3.1. We have described a scalable contextualisation architecture (ConTaaS) for IoT data processing in a cloud environment in Section 3.2 to cover RQ3. Later in Section 4.1, we describe the implementation of Contextualisation-as-a-Service (ConTaaS) for IoT data processing in a scalable parking recommender use-case scenario for RQ4. For RQ5, we introduce a novel scalable data obfuscation technique that combines contextualisation with digital watermarking based on the disclosure privilege of matching roles in Section 5.1. Finally, in Section 5.2, we propose a scalable and context-aware granular obfuscation technique for preserving privacy of spatial-temporal IoT scale data.

1.3 Contributions

The main contributions of this thesis are of three main categories:

1. **Contextualisation for Scalable Data Processing and Analysis (Chapter 4)**

   - *Systematic literature review of state-of-the-art research publications related to IoT contextualisation.*

     To consider relevant published papers that discuss contextualisation aspects, we developed a novel systematic literature review methodology that based on

the impact of existing publications. The selected studies have been investigated based on the literature review methodology described in Section 2.2 .

- *Design of a scalable contextualisation architecture for IoT data processing (ConTaaS).*

  Although contextualisation can be viewed as a subclass of data analytics, many of the latest high-performance processing techniques for Big Data, such as MapReduce [6], are not ideal for IoT contextualisation because they fall short in supporting the incremental data processing requirements of contextualisation, and the near real-time requirements of many IoT applications. In order to facilitate IoT scale data contextualisation, a novel and innovative architecture is designed to support high-performance and scalable contextualisation in real-time (Chapter 3). This will include formal definitions for IoT contextualisation, design of the high performance and real-time contextualisation operators utilising prime factorisation, and implementation of ConTaaS design.

- *Cloud-based implementation of ConTaaS that utilises commercially available cloud infrastructure services.*

  ConTaaS architecture has been implemented in the Amazon Web Services EC2 cloud infrastructure [7]. This implementation applies the proposed contextualisation and IoT data processing approach proposed in this thesis to a 'Smart City' application referred to as Smart Parking Recommender (Section 4.1). This implementation is able to represent and contextualise data from IoT devices and provides an efficient way to for IoT applications to query contextualised IoT data.

- *Experimental evaluation of the proposed contextualisation technique.*

  The contextualisation technique proposed in this thesis is evaluated in terms of query processing time in an experimental scenario (Section 4.3). In this experiment we consider thousands of cars searching for parking spots in Melbourne. We utilise a dataset provided by the City of Melbourne augmented with a synthetic dataset of parking sizes and descriptions. The contextualisation

computation remains linear even in situations where the number of IoT data points (i.e., cars) increases exponentially.

2. **Contextualisation for Scalable Security (Chapter 5)**

   - *Introduction of a novel scalable data obfuscation technique that combines contextualisation with digital watermarking based on the disclosure privilege of matching roles.*

     In this thesis a lightweight yet highly scalable data obfuscation technique is proposed that combines contextualisation with digital watermarking based on the disclosure privilege of matching roles to govern access to IoT data. A digital watermarking technique is used to control perturbation of sensitive data enabling legitimate users to de-obfuscate perturbed data. The proposed technique utilises ConTaaS (Section 3.2) to achieve real-time aggregation and filtering of IoT data for a large number of designated users to enhance the scalability. ConTaaS contextualises sensitive data to reduce data size prior to data obfuscation. Reversibility of the obfuscated data is also provided to users with the appropriate disclosure privileges. Therefore, only the perturbed versions of the original data are available to the public as described in Section 5.1.

   - *Experimental evaluation of the proposed contextualisation for scalable security.*
     We evaluate the performance of the scalable data obfuscation technique with over 15 days worth of patient data. In this experiment, we compare the query processing time in a healthcare-related use-case. The experimental data presented in Section 5.1.3 demonstrate that the proposed technique is effective and lightweight and can make data processing 160 times faster, on average, than not using contextualisation.

3. **Contextualisation for Scalable Privacy (Chapter 5)**

   - *Proposing a scalable and context-aware granular obfuscation technique for spatial-temporal data.*

     The highly connected and distributed nature of the IoT opens up the possibility

of compromising privacy before obfuscation takes effect. Therefore, privacy enforcement should be deployed at earlier stages. Additionally, classical privacy treatments are too restrictive for the IoT, where coarser or finer data details may be required by different applications. In this thesis, a framework for privacy preservation in IoT environments is proposed that is capable of multi-granular obfuscation by enforcing context-aware disclosure policies (Section 5.2).

- *Experimental evaluation of the proposed scalable and context-aware granular obfuscation technique.*

  In Section 5.2.6, we evaluate the performance of the proposed context-aware granular obfuscation technique in a smart vehicle use-case scenario. For this purpose, a large-scale urban vehicular mobility dataset is used, which contains car traffic trajectories for the city of Cologne [8]. The performance of the proposed technique is investigated in terms of the processing time required to retrieve trajectories. This includes the time required for reversing the privacy transformations and obfuscation. The proposed technique significantly outperforms its encryption counterpart in terms of response time and data protection at the sensor and database levels. The evaluation presented in Section 5.2.6 shows that our novel technique can preserve privacy nine times faster than the most common encryption algorithms.

CHAPTER 2

Background and Related Work

In this chapter, we describe the background and related work information related to this thesis. Particularly, in Section 2.1 we have explained challenges for IoT contextualisation, in Section 2.2 we have explained the systematic literature review for IoT contextualisation, and in Section 2.3 we provide a literature review of IoT security and privacy.

2.1 Challenges for Internet of Things Contextualisation

This section provides a background to IoT contextualisation. More specifically Section 2.1.1 introduces heterogeneity in sensor data and semantic approaches, Section 2.1.2 discusses current notions for 'context' in respect to general contextualisation, while Section 2.1.3 involves the querying of sensor data.

2.1.1 Heterogeneity of Sensor Data and Semantic Approaches

Heterogeneity makes contextualisation more difficult, so homogeneous sensor data can help with IoT contextualisation. Over the past 10-15 years, sensors have been used in several different areas, such as environmental monitoring, traffic control and healthcare. Sensors are typically small devices capable of sensing, storing and transmitting data, as well as being actuated over wired and wireless networks. One of the main challenges in sensor networks

is in transforming data from heterogeneous sensing devices manufactured by different vendors for different applications into homogeneous, discoverable and usable information presented in human- and machine-readable forms. There have been several recent efforts to tackle this challenge through metadata tagging or semantic annotation of sensor data [9, 10, 11]. While metadata can be any sort of informal information attached to data, semantically annotated data is associated with ontologies [12] that expressively and formally define and describe the type, properties and interrelationships of the data. Semantically annotated data is not only more understandable but can also be used for reasoning and to deduce new knowledge and, subsequently, to increase the expressiveness of the data. The Sensor Web Enablement (SWE) [13] standard from the Open Geo-spatial Consortium is an international effort to standardise all types of sensors, transducers and sensor data repositories that are accessible and discoverable via the internet. SWE consists of the following: 1) Sensor Model Language, which includes a standard model and an Extensible Markup Language (XML) Schema [14] for describing sensor characteristics, specifications and capabilities, such as their location. 2) The Observation and Measurements standard model and schema for describing observations and measurements from sensors and sensor networks. 3) The Observation interface for entering queries and retrieving observation and sensory data. SWE standards and XML schema are able to describe sensor data and observations with metadata to some extent, but they do not support the semantic reasoning, abstraction and classification provided by semantic technologies. The Semantic Sensor Network (SSN) [15] adds semantics describing sensors and sensor networks. The SSN ontology is compatible with SWE and extends semantic support for SWE. The SSN ontology expressively represents sensors and observations of the environment.

2.1.2 The Variety of Context Notions

Context, in IoT contextualisation, is shared among different applications. On the other hand, there should be a clear boundary to distinguish data from context. Much of the related work has attempted to define and use context in developing a wide range of intelligent applications ranging from mobile computing to artificial intelligence [16, 17, 18].

Context has been introduced in the literature by several researchers. The most common definition in the literature is by Dey et al. [16], who define context as "any information that can be used to characterise the situation of an entity", where "an entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." While this definition is sufficient for some of the related work, it does not necessarily capture context from the perspective of the IoT or other large-scale and multi-application environments. Moreover, there is no clear separation between data and context in this definition. Context in the IoT is closely aligned with the notion of context in context-aware computing, due to the fact that context-aware computing and the IoT have similarities in terms of data. However, they do not have the same scales and processing demands.

2.1.3 Querying Sensor Data

Contextualisation often requires the querying of context and sensor data that is maintained in a specific data model. For example, if sensor and context data are managed by a relational database, the relational database model and Structured Query Language (SQL) [19] constrain the contextualisation that can be achieved. Relational databases require sophisticated resources to deal with complex queries. Moreover, relational databases cannot easily adapt their schema to accommodate contextual changes. Non-relational databases (also referred to as noSQL or nonSQL) provide data models that are more suitable for Big Data and distributed and scalable data storage [20]]; however, the data models and query processing capabilities they provide are heterogeneous/nonstandard. The Resource Description Framework (RDF) [21] provides a more powerful and standardised data model than SQL and noSQL databases, but query processing is complex and limited in scalability. Therefore, the management of contextual data (often referred to as *contextual information*) is a challenge.

2.2 Contextualisation in the Literature

In this section we provide a SLR of existing context and contextualisation techniques. SLR is a well-defined and established methodology for identifying, analysing, and interpreting all the available evidence that is (to a reasonable degree) related to a specific research question, and doing so in a way that is unbiased. SLR contrasts with expert review based on an ad-hoc selection of literature related to a specific subject [22, 23].The research contributions that presented in this thesis include the SLR presented in this chapter, which is based on a specific sequence of tasks adapted from [22].

2.2.1 Identification of Related Work

To find and review existing publications that discuss contextualisation, we searched Google Scholar (scholar.google.com) for relevant publications, of which we found over 3000 [1]. In particular, we organised our searches into four sets according to the search phrases listed in Table 2.1. This table also shows the total number of publications found by each query, and the number of such publications per year.

Table 2.1: Publications related to each search phrase.

Query Search Phrases	Total	2011	2012	2013	2014	2015	Date are not specified
'context reasoning'	1706	370	386	373	317	195	65
'context inference'	1287	267	241	253	284	203	39
'inference of context'	101	24	21	19	27	10	0
'reasoning on context'	82	22	17	15	13	15	0

Next, we reduced these four set of related publications to the most relevant one via the following publication selection process:

1. **Removal of duplicates**

   Related publications that included more than one of the search phrases in Table 2.1

[1] All the queries were conducted at the same time in November 2015

were included in more than one of the four result sets. Such duplicates were removed.

2. **Removal of articles by publishers lacking an excellent reputation**
   Therefore, we only considered publications from well-known publishers, including Elsevier [2], IEEE [3], ACM [4], and Springer [5].

3. **Removal of articles based on secondary and tertiary studies (surveys)**
   Publications can be classified as primary, secondary or tertiary studies. Primary studies are normal research publications that include research contributions. A study that reviews primary studies on a specific research topic is a *secondary study* [22]. *Tertiary studies* review other reviews. Therefore, in this step, we removed all publications that were secondary or tertiary studies (e.g., with the word 'survey' in the title).

4. **Removal of articles that had never been cited**
   This step allowed us to consider citations as a metric of the research impact of a publication. Therefore, we did not consider papers without any citations.

5. **Focus on articles with the highest research impact**
   To select the highest impact publications we utilised the μ_j value [Equation 2.1] proposed for primary studies. In this equation, α_i is the number of citations of each paper i published in year j, and n_j is the total number of related papers published in that particular year. To select the highest impact research papers we set n_j to be our set of publications after step 4, and j for the past five years.

$$\mu_j = \left\lceil \frac{\sum_{i=1}^{n_j} \alpha_i}{n_j} \right\rceil \tag{2.1}$$

6. **Manual filtering of the remaining papers**
   Finally, the last selection phase was to exclude papers that were not relevant to this

[2] www.elsevier.com
[3] www.ieee.org
[4] www.acm.org
[5] www.springer.com

thesis, such as papers from disciplines other than computer science and engineering (e.g., *context* in social science).

Figure 2.1 shows the number of papers remaining after each step of the selection process.

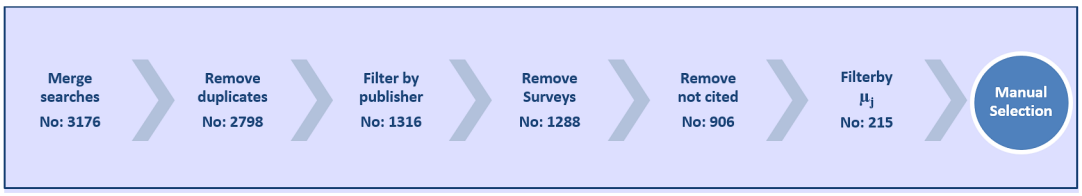

Figure 2.1: Literature selection process.

2.2.2 Primary Studies

In this section, we summarise the contributions of related publications (mainly highly-relevant primary studies with high research impacts, as discussed in Section 2.1).

- Baladron et al. [24], presented a converged framework for context management as a driver for service adaptation in the future internet. It allows integration, monitoring and control of heterogeneous sensors and devices under a single context-aware service manager (i.e., a service manager that utilises contextual information and provides contextualisation techniques). This module can use clustering algorithms to take advantage of user histories for inference and prediction of missing contexts.

- Roussaki et al. [25], presented a context management architecture suitable for pervasive services combined with social networking and explored its value to users all over the world. They presented the first results from their work in SOCIETIES , a European Information and Communications Technology (ICT) research project that aims to bring together pervasive computing and social networking paradigms.

- Lee et al. [26], presented the implementation of MobiCon: a practical context-monitoring middleware for context-aware application development. This middleware mediates context-aware applications (i.e., an application that can utilise contextual

information to deliver services) and personal sensor networks, offering Application Programming Interface (API)s as well as run-time environments for applications. The system guarantees accurate context recognition (i.e., a process that identifies user and application contextual information from sensor data) by means of five modules, namely, context processor, sensor manager, resource coordinator, application broker and sensor broker. Some requirements for the system are high-rate data acquisition from multiple sensors, feature extraction and context recognition with large amounts of context data, along with intermediate transmission of results. MobiCon supports multiple applications and can adapt to sensor availability.

- Okeyo et al. [27], described an integrated architecture that combines ontological and temporal knowledge representation formalism for composite activity recognition. They also presented models, methods and algorithms that are capable of supporting the recognition of simple and composite activities. In this paper, contextual user data was analysed to recognise users' ongoing activities. They claimed that the paper presents the first effort to use a purely knowledge-driven approach that addresses temporal representation as well as reasoning requirements to recognise both simple and composite activities.

- Yndurain et al. [28], presented an architecture that enables search engines to take into account contextual information to enhance the search results. They proposed a technique to adapt the queries to deal with signals and produce intermediate context states. They experimented with the use of heuristic rules for contextual reasoning. They propose that, in the future, search engines must become more context-aware.

- Rodriguez et al. [29], presented a fuzzy ontology that aims to solve the limitations of other ontology-based activity recognition techniques in dealing with imprecision and uncertainty, as well as vague or incomplete data. The proposed approach is incremental and allows different levels of granularity that, as they state, allows behavioural abstraction and more accurate recognition. In their fuzzy ontology, rules

can be established to recognise human behaviour using a sequence of observations, a behaviour specification structure and handling of uncertainty.

- Roy et al. [30], proposed a framework for sensor networks that fuses data from multiple sensors and uses the context state to support context-aware services that handle ambiguity by reasoning efficiently about the state. The focus is on considering the computational aspects of data related to sensors and providing context-aware services. The main goals of the work are to construct a framework that can deal with information redundancy and guarantees an application's quality in terms of contextual bounds. They proposed a system that includes dynamic Bayesian network techniques and uses sensor data to derive context states via a fusion process. They also used reasoning techniques from information theory to determine optimal contextual attribute values and minimise state ambiguity. They proposed an indicator, the Quality of Context (QoC), to evaluate the framework and used Sun Small Programmable Object Technology (SunSPOT) to build a system and validate their proposal.

  propose a framework for sensor networks that fuses data from sensors and uses the context state to support context-aware services that handle ambiguity by reasoning efficiently about the state. Focus is set on considering the computational aspects of data related to sensors and providing context-aware services. The main goals of the work are to construct a framework that can deal with information redundancy and guarantees an application's quality in terms of context bounds. They propose a system that includes dynamic Bayesian Network techniques and uses the data from sensors to derive context states by means of a fusion process. They also use reasoning techniques from information theory to select context attributes' optimal values and minimise state ambiguity. They proposed an indicator, the QoC, to evaluate the framework and used Sun Small Programmable Object Technology to build a system and validate their proposal.

- Song et al. [31], proposed an interactive middleware architecture for lifelog-based

context awareness (which is a technology that automatically provides a service based on the contextual information) in distributed and ubiquitous environments. The middleware is a system that can distribute and manage situational information in mobile nodes using mobile devices in distributed and ubiquitous environments. The system shares service contents through interactive middleware through publication. Their proposal aims to unify and share multiple middleware modules to lay the foundation for providing extended functions to applications using services to improving performance.

- Rahman et al. [32], presented SenseFace, a framework to create a personal social network that offers personalised and context-specific interaction with different services and communities of interest. SenseFace provides the following features: 1) It is capable of extracting the user's personal social network from the internet; 2) it extracts content from Body Sensor Network (BSN) and Personal Social Network (PSN); 3) it extracts context information from the BSN and PSN and creates context primitives; 4) it combines the generated context primitives that define a user's context; 5) it employs a novel ubiquity stack that maps each user's context to a subset of services and social ties; and 6) it dynamically assigns a priority to each service so that the visualisation maintains a layout of each service based on its earned credit.

- Martin et al. [33] explored the use of smartphones for activity recognition. They studied how to approach the building of an activity recognition system through continuous background execution in a smartphone. The architecture they proposed is an embedded, stand-alone, physical activity estimation approach that uses the sensing, processing and storage capabilities of devices in order to estimate significant movements or postures (e.g. walking at slow, normal and rushing paces, running, sitting, standing, etc.). They demonstrated the feasibility of the system by 1) collecting an activity dataset of 16 individuals, and 2) training a set of classifiers (Naïve Bayes, decision table and decision tree) working on different selections of sensor data. They measured the accuracy, computational cost and memory fingerprint of

their classification system.

- Kabir et al. [34], presented a Social Context Information Management System (SCIMS) to support the development of applications utilising social information. They proposed an ontology-based model for representing and storing both people- and object-centric relationships and used these to compute the status information of SCIMS users. Based on information acquired from various sources, SCIMS derives a rich social context. The system uses Facebook, LinkedIn, Twitter and Google Calendar to acquire users' social context information (i.e., social roles, social relationships and status) and provides rich semantic support for representing and inferring from such contexts. They propose a way to preserve privacy by allowing users to adjust the granularity of information stored.

- Rossi et al. [35], presented a real-time ambient sound recognition system for smartphones called AmbientSense. They described the design, implementation and evaluation of the system. The system samples ambient sound data, extracts features from the data and produces a context recognition result by means of a Support Vector Machine (SVM) classifier using auditory scene models. There is a training phase for creating models based on an audio data training set. The recognition stage is implemented in two modes: autonomous mode and server mode. In server mode, classification is performed by transmitting features to a server and then receiving the resulting class. Both modes of operation were evaluated with a set of 23 daily life ambient sound classes in terms of recognition performance, phone CPU load and recognition delay.

- Meditskos et al. [36], proposed a hybrid framework (SP-ACT) for complex activity recognition via a semantics solution that utilises Web Ontology Language (OWL), SPARQL Protocol and RDF Query Language (SPARQL) for recognition of complex activities. Ontologies provide a common vocabulary for representing activity-related contextual information and SPARQL rules derive high-level activity interpretations. The temporal relations among activities are handled by SPARQL functions and the

derivation of new composite activities exploits the native capabilities of SPARQL to update the underlying activity model.

- Papadopoulou et al. [37], proposed the idea of a Personal Smart Space (PSS), which is a set of devices connected using peer-to-peer connections, and described the interactions that may occur among them. They presented an architecture for the PSS that includes several layers and blocks that structure the management of information from the device level to actions at the environment level. They claim that the PSS approach easily separates the personalisation needs of the user from the control of the devices.

- Okeyo et al. [38], presented an approach to dynamic sensor data segmentation, which is where sensor data streams are segmented into fragments, each of which can be mapped to an activity description. They used this approach for knowledge-driven activity recognition that is capable of continuous real-time operation. The main contributions of this work are: 1) A sensor data segmentation model was proposed based on time windows that is applicable to a wide range of activity recognition scenarios; 2) description of mechanisms for the dynamic manipulation of model parameters, such as the setting, shrinking and expansion of the time window; 3) incorporation of the data segmentation method into an ontology-based technique for activity recognition; 4) evaluation of the performance of the proposal in real-time activity recognition; 5) implementation of a prototype system to evaluate the above that consists of a synthetic Activities of Daily Living (ADL) data generator, an ADL ontology, a sensor data simulator for ADL data playback and a real-time activity recognition system. This evaluation method utilised accuracy as its main metric and the results demonstrated the feasibility of this approach.

- Wei et al. [39], explored an approach that uses a middleware system, called CAMPUS, to automatically perform context-aware adaptation of decisions at run time. CAMPUS utilises semantic technologies to dynamically make adaptation decisions based on contextual information. They also proposed a new programming model based on

compositional adaptation for constructing context-aware applications and facilitating adaptation decisions. CAMPUS formulates a comprehensive ontology-based model that allows the capture of important concepts and relationships between entities, which are necessary for automated context-aware adaptation decisions. Based on these ontologies, the CAMPUS employs Description Logic (DL) and First-order Logic (FOL) to infer and make context-aware adaptation decisions automatically.

- Boytsov and Zaslavsky [40], proposed, developed and evaluated a technique for formal verification of context models and situations. Situations, in this context model, include dependencies among situations as well as situation definitions in terms of context features . The paper demonstrated that the definitions complied with the expected properties and provided a complete set of counterexamples that illustrated situation inconsistency.

- Carreira et al. [41], proposed a prototype system that uses context information to automatically detect and solve conflict situations in Home and Building Automation System (HBAS). The contributions of this work include the following: 1) a conflict taxonomy that includes a formal representation of intelligent environment conditions and components, and 2) a foundation for systems capable of automatic conflict detection and resolution. The research also included an investigation of the nature of classification of conflicts in home automation, a model that allows specification of the intelligent environment components that can cause conflicts, and a prototype system that can analyse the environment, detect the existence of conflicts and determine whether they can be resolved.

- Yuan and Herbert [42], presented a fuzzy-based context modelling and reasoning framework for a proposed pervasive healthcare architecture referred to as CARA. The reasoning component of CARA fuses physiological, behavioural and environmental information to support home healthcare monitoring. CARA also provides more accurate emergency situation detection via incorporation of real-world environmental data to supplement medical sensor data. The paper also presented some details of

the CARA architecture, including its remote monitoring, data and video review, and healthcare reasoning components.

- Lee et al. [43], presents Cooperative Context Monitoring System (CoMon) for addressing the high-energy usage problem in mobile content management that occurs when context information is shared among mobile users. Their solution includes techniques for continuity-aware co-operator detection and benefit-aware negotiation. According to their estimations and tests, the monitoring system enables mobile applications to monitor the environment with much lower energy consumption than other techniques.

- Nath [44], presents a middleware system for efficient and continuous sensing of the contexts of mobile phone users. The proposed middleware dynamically learns rules about the relationships among various context attributes using a novel technique for association rule mining. The rules learned are exploited for optimising inference caching and speculative sensing. Inference caching allows the middleware to infer one context attribute from another already-known attribute without requiring any sensor data. Speculative sensing enables the middleware to occasionally infer the value of an expensive attribute by sensing cheaper attributes.

- Ha et al. [45], described an assistive system based on Google Glass devices for users in cognitive decline (e.g., those with Alzheimer's disease or mild cognitive impairment) and survivors of stroke. They described the architecture of the system, which is called Gabriel, and presented a prototype implementation. Google Glass devices are used to perform first-person image capture, sensing, processing and communication. The main system architecture characteristics include being multi-tiered, offering low end-to-end latency bounds on computing-intensive actions, taking into account the limited battery capacities and processing capabilities of these devices, support for Vector Machine (VM)-based extensibility for customisation, and graceful degradation of offload services in case of network failure and unavailability.

- Mehrotra et al. [46], proposed a middleware system solution called SenSocial that builds and binds social and physical context data streams for applications. SenSocial performs remote management of streams and filters their data to refine contextual (physical and social) data streams. Next, it separates and delivers relevant parts of such context data to various apps as they require. SenSocial offers privacy management functionality that allows the developer to manage the type and granularity of sensed contextual data that is stored and shared. It is also able to manage the sampling of users' physical contexts once an online social network action is detected and pairs the sensed physical context with the social network information.

- Motik et al. [47], described a knowledge management system that supports context-aware applications called Delta-Reasoner. This system uses RDF as the data model and OWL for semantically representing background knowledge. It uses incremental reasoning for dealing with changes in reading calculations from the sensors. Delta-Reasoner is part of the Intelligent Mobile Platform, a system that exploits semantic technologies to represent different situations and related applications.

- Maia et al. [48], presented a context management middleware system for the Android platform called Loosely Coupled Context Acquisition Middleware (LoCCAM). LoCCAM provides self-adaptive gathering of contextual information from nearby devices. The context management component of LoCCAM is built as an extension of the Open Service Gateway Initiative (OSGi) framework [49], with the feature of dynamic reconfiguration of the acquisition layer during application execution. LoCCAM employs a model for publication and notification of contextual information based on tuple spaces.

- Xu et al. [50], explored the feasibility of inferring a user's inputs on the touchscreen of a smartphone using the data collected from motion sensors. They show that there are unique patterns of tap events (in terms of changes in acceleration) and that statistical approaches can be used to detect the occurrence of tapping. Similarly, they showed that there is a correlation between the tap position and the gesture change

(as detected by the orientation sensor) during a tap event. They also designed what they called a TapLogger, which is a Trojan horse application that utilises observed sensor data to secretly log users' touchscreen inputs.

- Wei and Jin [51], presented a context-aware service discovery architecture for the IoT that aimed to provide an efficient infrastructure to support smart service provisioning. An ontology-based model that utilises Dynamic Bayesian Network (DBN) was proposed to handle uncertain and temporal contexts. They also examined the function of context and relations with entities in the IoT. To achieve this, they used DBN to obtain high-level contexts from low-level time-series data streams by reasoning about these contexts.

- Liu et al. [52],] presented an ontology learning model called Domain Ontology Graph (DOG). This model 1) supports the definition of ontology graphs that provide knowledge conceptualisation, and 2) supports the ontology learning process that guides semiautomatic domain ontology learning and generates corresponding ontology graphs. Two kinds of ontological operations are introduced in this paper: document ontology graph generation and ontology-graph-based text classification.

- Chon et al. [53], proposed a method for the autonomous construction of a Point of Interest (POI) map called LifeMap, which includes a service that provides information about various locations without needing a centralised server. LifeMap exploits onboard accelerometers and electronic compasses to track the locations of mobile device users. It incorporates a room-level, fingerprint-based, place-learning technique that generates logical locations from the properties of Wi-Fi radio signals. The paper also presented an implementation of the system on Android phones and validated its practical usage in everyday life.

- Zhu et al. [54], proposed an authorable (able to generate Augmented Reality (AR) content) and context-aware system for assisting maintenance technicians. This system, called ACARS, enables AR developers to create contextual information for maintenance purposes via a 2D desktop user interface. Existing authoring tools

for maintenance tasks are unidirectional but are not context-aware, so proving this combination is a significant contribution. ACARS consists of context management, AR visualisation, database, offline authoring and on-site authoring components. The context management module collects contexts from user inputs and sensors, infers new contexts and transmits all the contexts to an AR-based visualisation module.

- Choi et al. [55], proposed a dynamic access control model they called Onto-ACM. Onto-ACM provides ontology reasoning and semantic analysis for managing the security level required to access resources. Onto-ACM offers a mechanism that can prevent misuse of access rights by dynamically changing the permissions of any user role based on context information.

- Zhang et al. [56], proposed a scheme of Decentralised Checking of Context Inconsistency (DCCI) in pervasive computing environments. They claimed that DCCI is the first scheme in pervasive computing environments that is capable of checking context inconsistency in a fully distributed manner. Their solution for doing this builds a shortcut structure that aims to reduce the communication overhead and improve the checking accuracy by exploiting a preference-based locality.

- Okeyo et al. [57], presented a hybrid knowledge-driven approach to composite activity modelling that combines ontological and temporal modelling. This approach enhances ontology-based activity models with qualitative temporal information based on Allen's temporal logic [58]. The paper proposed 1) ontological modelling constructors describing composite activities, and 2) temporal modelling operators. These provide for the modelling of both static and dynamic characteristics of activities. The paper used these modelling approaches to demonstrate several composite activity models. Finally, a set of inference rules was proposed to achieve composite activity recognition.

- Coradeschi et al. [59], presented a system called GiraffPlus, consisting of a network of home sensors that collects vital sign measurements such as weight, blood pressure, blood glucose, environmental signals from smoke sensors, temperature sensors, fall sensors, etc. This information is then processed by a context recognition system

that recognises activities, monitors health and assesses wellbeing. GiraffPlus can subsequently trigger alarms or reminders to its users or their caregivers. A telepresence robot, the Giraff robot, is part of the proposed solution. This robot can be moved around the home by somebody else controlling it over the internet. It is equipped with a video camera, display, microphone, speakers and touchscreen, and can be used for information collection and communication.

- Ju et al. [60], proposed a coordinated sensing flow execution engine, referred to as Symphoney, for concurrent sensing applications. Symphoney supports frame externalisation that identifies semantic structures embedded in sensor data streams. Symphoney also provides a data flow programming model via an XML interface. It supports developers by giving them tools to flexibly compose customised sensing flows. This allows rapid prototyping of complex sensing flows, reducing the time and effort needed.

- Nguyen et al. [61], proposed a framework for discovering latent patterns that employs Hierarchical Dirichlet Processes (HDP). Latent patterns are relations amongst contexts that are hidden inside the data and that are not easily inferred. HDP is a hierarchical Bayesian non-parametric model mainly used for the purpose of modelling grouped data. The paper includes results of an experiment with users from the authors' lab. These experiments show that data can be represented and learned with HDP and these can find parameter clusters that are similar to those yielded by other methods that search for latent patterns.

- Zaslavsky et al. [62],presented CAROMM, a mobile data stream mining system that aims to provide improved scalability of mobile data collection and run-time analytics (on-the-move mining). The paper shows that CAROMM is capable of collecting and processing data from a large number of mobile devices. The key component of this work is CAROMM's data analysis-cluster engine, which provides the mobile data analytics functionality and related scalability.

- Sudhana et al. [63], proposed an ontology for a context-aware adaptive e-learning application that delivers learning material by taking into account the context of the e-learner. In this work, context is modelled as ontological profiles. The paper describes alternative categorisation approaches for contextual information in the e-learning domain based on learner perspective-based context acquisition. The paper also proposed an architecture model for a context-aware e-learning application.

- Liu et al. [64], presented an Incremental and Distributed Inference Method (IDIM) for large-scale ontologies (incremental RDF datasets) that utilise MapReduce [6]. This solution works by constructing a Transfer Inference Forest (TIF) and Effective Assertional Triple (EAT). These help reduce the required storage and simplify the reasoning process required. These, in turn, allow users to execute their query more efficiently (other related work requires computing and searching over the entire RDF closure).

- McNaull et al. [65], proposed an Ambient Assisted Living Flexible Interface (AALFI) that is controlled by a Multi-Agent System (MAS). This solution aims to provide help to older people so they can continue living in their own home for longer. This research provides such support via an adaptive multi-modal interface that is driven and updated by a MAS and complements the current support offered by the NOCTURNAL project [66] by providing interaction through visual and auditory modalities. The users are provided with advice that is based on criteria such as the quality of sleep during the night and possible breaches of safety during the day. These help its user to carry out corrective measures and/or seek further assistance. Interactions are personalised to support each user's needs and are either visual or auditory.

2.2.3 Definitions of Terms for the Review of Related Work

The following terms are used in the review of related publication in Section 2.2.2:

- **Contextual Operations -** These are the operations that utilise context information and perform IoT data contextualisation. They include the following:
  - *Filter*: Operations that only allow certain information to pass through them;
  - *Aggregate*: Operations that compute a single value from a collection of values by using mathematical operations such as averaging or summarising;
  - *Infer*: Operations that deduce new information by reasoning on the data.
- **Techniques -** This includes any algorithms for the above and other contextual information.
- **System -** This includes the implementation, architecture and APIs:
  - *API and Software tools*: Any external data communication with other services or applications;
  - *Implementation and architecture:* Include the architecture and implementations described in the papers.
- **Semantics -** This includes the use of semantic-based models and techniques for defining and utilising content and performing contextualisation:
  - *Ontology*: Specified ontologies that have been used in the paper;
  - *Context*: The definition or description of the context used in the paper;
  - *Context Attributes*: Context attributes described in the paper;
  - *Context Reasoning*: Deduction of context based on available information;
  - *Context Model*: Describes how the context data is structured.
- **Evaluation -** This includes evaluations of techniques and systems for contextualisation:
  - *Evaluation Definition:* Evaluation of the contribution of the paper;
  - *Evaluation Parameters:* Parameter used for evaluation of the paper's contribution.

- **Big Data and IoT -** This includes specific references and solutions for Big Data and the IoT:
  - *Big Data*: Any reference to Big Data and its definitions;
  - *Big Data Parameters (3V)*: Investigation of main Big Data parameters including volume, variety, and velocity;
  - *Internet of Things*: discussion or contribution related to the IoT.

2.2.4 Systematic Literature Review Outcome

This section presents the conclusion of the SLR.

To assess the related work summarised in Section 2.2.2, we consider the contextualisation operations in the literature review. The majority of the related work proposes reasoning and inferring operations for context and contextualisation. Filtering operations are encountered in a few publications [54, 46, 32, 26]. However, they do not necessarily describe if or how context is utilised in the filtering operation. Furthermore, we could not find any systematic description of filtering and aggregating operations or a comprehensive description of contextualisation operations in any of the related works.

Related work that explores the semantic aspect of context and contextualisation lacks any clear definition of *context* and provides no clear boundaries between sensor data, application data and context (or contextual information). In addition, definitions of *context* are typically application specific. Similarly, the majority of related work defines ontology by themselves or they are not using any formally defined ontologies. Furthermore, there are no common methods of using ontologies.

Related work does not provide a common evaluation methodology, and there is no common framework for the performance evaluation of the provided contextualisation solutions. Furthermore, the scalability of such contextualisation solutions is not considered in most of the papers. Finally, no related work has demonstrated scalable and/or near real-time IoT data processing in applications involving context and contextualisation.

No related work has investigated and provided solutions for dealing with high-volume, -velocity and -variety sensor data, which remain a challenge in IoT and related Big Data applications. In fact, most related work does not consider the IoT at all. There were a few papers that mention applications and services using sensors but these are not involved any internet-related issues and, hence, are not necessarily IoT applications.

Finally, there is no generic cloud solution for using context and contextualisation in the IoT and related Big Data applications.

2.3 Scalable Security and Privacy in Internet of Things

Typical security threats that can compromise IoT applications include eavesdropping, impersonation, modification and data breaches. Moreover, to protect the privacy-sensitive data of individual IoT devices, e.g. in the case of healthcare applications, it is important to provide privacy-aware access to data without exposing the actual data. The IoT is an important new internet technology with great potential for developing smart buildings and cities, assisted living and healthcare, precision agriculture and environmental monitoring, manufacturing, and security and defence. IoT systems and their applications must deal with malicious disclosure and attacks and provide mechanisms that protect sensitive data such as patients' physiological data, energy consumption data from smart meters, and the locations of mobile users. Existing techniques for protecting privacy-sensitive data in the IoT include the following.

There are several aspects of the IoT that present security and privacy problems, including IoT device communications, constrained resources (e.g., limited battery life), variety (e.g., different types of devices made by multiple manufacturers), and scale (billions of devices) [67]. Among the plethora of recent research solutions [68, 69, 70, 71, 72] for protecting sensitive IoT data, some related research (e.g., [73, 68, 70]) focuses on security and privacy preservation policies while others (e.g., [68, 69, 71, 72]) focus on encryption and the design of privacy preserved frameworks for the IoT [74]. Although most proposed techniques can ensure security and privacy, their ability to scale up to support millions of IoT devices and their data has not been validated.

Ensuring the scalability of privacy-preservation solutions for millions of IoT devices is a significant problem. The solution proposed in this section couples watermarking with contextualisation to protect the privacy of a virtually unlimited number of IoT data points. As noted earlier, there are clear parallels between the disclosure technique proposed in this thesis and access control, and there has been a considerable volume of research on developing both access and disclosure control methods.

The most common access control mechanisms are Discretionary Access Control (DAC), Lattice-Based Access Control (LBAC), and Role-Based Access Control (RBAC) [75]. DAC is discretionary in the sense that the owner of the requested resource controls the access to that resource. Each access request is checked against the specified authorisations. If there exists an authorisation stating that the user can access the resource in a specific mode (read or write), access is granted, otherwise it is denied. LBAC enforces unidirectional information flow via a predefined lattice of security labels that are associated with every resource and user in the system. RBAC determines the access level via the role abstraction, rather than simply by the identity or clearance of the requester. In this model, a role is a semantic construct, which is often a representation of a job in an organisation.

In an IoT setting where both data and access control policies can change rapidly, the above access models cannot deal with such frequent changes. To deal with such changes, another trend in research enriches access polices with contextual information. For instance, several extensions to the basic RBAC model have been proposed to incorporate context variables such as the Generalised Role-based Access Control (GRBAC) model [76]. GRBAC introduces environmental information such as temperature or location to activate roles based on the value of conditions in the environment where the request was made. A similar context-aware RBAC model has been proposed for health-care applications [77], where the contextual information invokes the relevant access policies for a specific role. A major deficiency of these approaches is that data access is either granted or denied.

In order to provide flexibility for situations where different data granularities are needed, disclosure control methods are advantageous. Existing disclosure control techniques are divided into identity and data disclosure control. Identity-based disclosure techniques, such

as k-anonymity and l-diversity or pseudonymity, attempt to detach or replace identifiers from data, whereas the latter techniques protect the data itself. In this section, we discuss only data disclosure control techniques. A comprehensive review of identity disclosure control techniques was conducted by Aggarwal and Philip [78].

Common techniques for data disclosure control include generalisation and suppression, data swapping and noise addition. Data generalisation attempts to prevent data linkages for the privacy preservation of published datasets. An example would be replacing an exact date of birth with only the year. Suppression techniques can be viewed as the ultimate form of generalisation since no information is released. Unfortunately, these techniques cause information loss and, also, are not appropriate for real-time applications because of the complexity of the required calculations.

The watermarking aspect of the technique we propose in Section 5.1 is similar to a noise addition technique, as digital watermarking techniques are used to obfuscate sensitive data. In contrast to noise addition techniques, this technique is reversible which enables tuning of the obfuscation parameters based on the access privilege of the users.

Achieving multi-granular disclosure requires the use of obfuscation techniques. Data obfuscation, in this case, involves generalising or degrading sensitive data to establish the desired level of granularity for disclosure. Existing obfuscation techniques include data randomisation, data anonymisation, random sampling, or data swapping [5]. For instance, Mivule [79] investigated techniques for adding noise to sensitive data, including additive noise, multiplicative noise, logarithmic multiplicative noise and differential privacy, with respect to the statistical preservation of a published dataset. Preserving privacy in an IoT setting only at the time of data dissemination may not be effective and the whole data life-cycle needs to be considered to ensure end-to-end data privacy.

2.3.1 Existing Techniques for Protecting Privacy-sensitive Data in Internet of Things

In this section we will describe existing techniques for protecting privacy-sensitive data in IoT.

- Key Agreement Protocols

  To secure communication in the IoT, it is important to encrypt the data sent between sensor nodes, gateways and other devices due to the public nature of the internet. Keys for encryption must be agreed upon by communicating nodes [80]. Due to resource constraints, key agreement in IoT is non-trivial. Many key agreement schemes used in general networks, such as Kerberos [81] and RSA [82], may not be suitable for the IoT because it usually has no trusted infrastructure. Pre-distribution of secret keys to all pairs of nodes is not viable due to the large amount of memory used when the network size is large. To overcome this problem, a random key pre-distribution scheme [83] has been proposed, where each sensor node receives a random subset of keys from a large key pool before deployment. Any two nodes can find a common key within their subsets and use it to secure their communication. Without requiring any key pre-distribution, data sensed within the IoT has been used to establish the common secret key. For example, in [84], two sensors, in a BSN, used the common electrocardiogram signals of a patient to establish a secret key.

  Roman et. al [85], Du et.al. [86] and Camtepe et. al. [87] analysed the applicability of several link-layer oriented Key Management System (KMS), which establish keys for sensor nodes within the same WSN using techniques such as linear algebra, combinatorics and algebraic geometry. However, the authors mention that not all mathematical-based KMS protocols can fulfil the IoT context. According to their analysis result, only [86] and [88] might be suitable for some IoT scenarios.

- Identity Protection Protocols Hu et.al [89] proposed an identity-based system that, protects the location information of IoT devices during emergency situations. In this approach, each user communicates with others using Virtual Identity (VID), which does not contain any real information about the user. Under this architecture, users' privacy can be protected well because they only send VID(s) to communicate, and VID is anonymous and unlinkable to users. The location information will finally be sent to the user making a request only after verification of their identity. In the

IoT, verifying the identities of "things" is crucial to preventing unauthorised access to users' private data, and granting access to legitimate users only. Liu et.al. [90] propose an authentication protocol for IoT systems. Under the proposed protocol, "things" and objects are end nodes, and each node has a unique global address for connecting over the internet. To establish a session key, both secret-key Secret Key Cryptosystem (SKC) and Public Key Cryptography (PKC) have been considered for IoT environments, but they all suffer several problems. For example, SKC requires large amount of memory to store key chains and PKC suffers from high energy consumption. Kalra et.al. [80] proposed an Elliptic Curve Cryptography (ECC) based key establishment method suitable for IoT environments. Their analysis indicates that the proposed protocol can prevent eavesdropping, man-in-the middle attacks, key control attacks, and replay attacks.

- Attribute-Based Encryption Schemes As a large amount of sensed data is stored in sensor nodes or databases, it is important to control access to it. Attribute Based Encryption (ABE) [91] was used to control access to sensor data in [92], [93]. In traditional public-key cryptography, a message is encrypted for a specific receiver using the receiver's public-key. Identity Based Encryption (IBE) [94] changed the traditional understanding of public-key encryption by allowing the public-key to be an arbitrary string, e.g., the email address of the receiver. ABE goes one step further and defines the identity not atomic but as a set of attributes, e.g., roles, and messages can be encrypted with respect to subsets of attributes (key-policy ABE - KP-ABE) or policies defined over a set of attributes (ciphertext-policy ABE - CP-ABE). The key issue is, that someone should only be able to decrypt a ciphertext if the person holds a key for "matching attributes". User keys are issued by a trusted party. In CP-ABE, a user's private-key is associated with a set of attributes and a ciphertext specifies an access policy over a defined universe of attributes within the system. A user will be able to decrypt a ciphertext, if and only if their attributes satisfy the policy of the respective ciphertext. Policies may be defined over attributes using conjunctions, disjunctions. For instance, let us assume that the universe of attributes is defined

to be {A=General, B=Nurse, C=Doctor, and D=Specialist} and User 1 receives a key to attributes {A,B} while User 2 to attribute {D}. If a ciphertext is encrypted with respect to the policy $(A \wedge C) \vee D$, then User 2 will be able to decrypt, while User 1 will not be able to decrypt. In KP-ABE, an access policy is encoded into the users secret key, e.g., $(A \wedge C) \vee D$, and a ciphertext is computed with respect to a set of attributes, e.g., {A,B}. In this example the user would not be able to decrypt the ciphertext but would for instance be able to decrypt a ciphertext with respect to {A,C}. Based on KP-ABE, Fine-grained Distributed Access Control (FDAC) was proposed for IoT in [92]. FDAC is resistant against user collusion, i.e., cooperation by colluding users will not lead to the disclosure of additional sensor data. Based on CP-ABE, another fine-grained access control scheme for IoT was proposed in [93] which allows AND-based policies only.

- k-Anonymity Techniques IoT data are valuable for knowledge discovery. Given that the IoT is regarded as the next generation worldwide network that connects every necessary object to facilitate our daily lives, privacy is a major concern and challenge. Current solutions to this problem include the [95] use k-anonymity techniques to anonymise sensor data before releasing it for analysis. The concept of k-anonymity was first formulated by Latanya Sweeney in [96] as an attempt to solve the following problem: "Given person-specific field-structured data, produce a release of the data with scientific guarantees that the individuals who are the subjects of the data cannot be re-identified while the data remain practically useful." A release of data is said to have the k-anonymity property if the information for each person contained in the release cannot be distinguished from at least k-1 individuals whose information also appears in the release. For example, if k = 5 and the potentially identifying variables are age and gender, then a k-anonymised data set has at least five records for each combination of age and gender. The most common implementations of k-anonymity use transformation techniques such as generalisation, global recoding, and suppression.

CHAPTER 3

Contextualisation

Contextualisation is defined as the process of filtering, aggregating, and inferring IoT data by using relevant information to the applications using the data. Contextualisation of IoT scale data requires techniques that can process large volumes of heterogeneous data arriving at very high velocity. Due to the increasing number of devices in the IoT, scalability is an important challenge to be overcome. Therefore, contextualisation of IoT data should be scalable in such a way that the IoT data input velocity, scale, and variety can be handled by the available computing resources. The contextualisation techniques proposed in this thesis are designed and developed for generic, scalable IoT ecosystems with consideration that the data will be used by multiple applications in real-time.

3.1 Internet of Things Data Contextualisation

In this section we present the main aspect of contextualisation that we used to provide our solution and also help deal with the challenges identified in Chapter 2. In particular, in Section 3.1.1 we provide working definitions for IoT contextualisation, while Section 3.1.2 presents contextual operations. A conceptual architecture for contextualisation is presented in Section 3.1.3.

3.1.1 Definitions

In this section we define the main components of the contextualisation approach we propose in this thesis:

- *Triples*: A triple is a statement describing a data item in the form of the following three parts: $< Subject, Predicate, Object >$. *Subject* is the identifier of the entity that the data is describing/is an attribute of. *Object* is the description of the *Subject* in terms of the relation described in *Predicate*. For example, $< RMITUniversity, hasEmail, info@rmit.edu.au >$, describes that RMIT University (*Subject*) has an email address (*Predicate*) which is "info@rmit.edu.au" (*Object*).

- *RDF Triples*: An RDF triple is a formal *triple* in such a way that the *Subject* can be a blank-node or Internationalised Resource Identifier (IRI) [97]. The *Predicates* are only IRI and the *Object* can be IRI, literals or blank-nodes. A blank-node in an RDF is a node that does not contain any data, but groups data as a parent node [Figure 3.1].

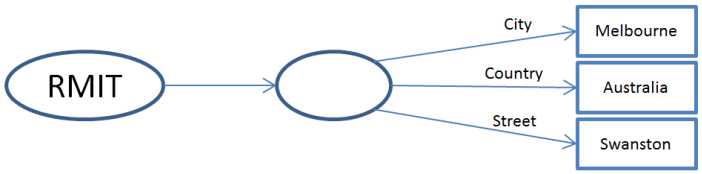

Figure 3.1: Blank Node

- *Context*: Context represented as triples, where the *Subject* is a specific *application*, *Predicates* describe the relevancy of the entity with the information and the *Object* is the information. For example, the context (triple): <App1, RestaurantType, vegetarian> represents that *App*1 is interested in vegetarian restaurants. Subsequently, *context* is defined as any combinations of the *Predicates* and *Objects* that is relevant to a given application.

- *Contextualised IoT Data*: Contextualised data for a particular application is a subset

of RDF triples that is filtered, aggregated and inferred according to the context relevant to the given application.

- *Context ID (CID)*: CID is a label assigned to each particular context to uniquely identify it. In this thesis, a unique prime number is exclusively assigned to each context triple (i.e., to each combination of a particular predicate and a particular object).

- *Contextual Preference*: Contextual preference is a set of contexts that are relevant to a given application as defined by the application itself or by the user of it.

- *Application Context ID (ACI)*: ACI is a label that represents all the contextual preferences of a given application. ACI may not be unique for each particular application and can dynamically change based on the changing contextual preference of each application. In this thesis, the ACI number computed and assigned to each application by multiplying the CIDs is relevant (i.e., in the conceptual preference) to that particular application. For example, assuming that $App1$ is an application that has two contexts as:

$$< App1, Location, Melbourne >, CID = 7$$

and

$$< App1, RestaurantType, Vegetarian >, CID = 29$$

the ACI number of these two contexts will be 203, i.e., the multiplication of 7 and 29. The ACI numbers identify:

1. The contexts of a given application
2. Applications with similar contextual preferences

For any application A with an ACI number n:

$$n = \prod_{i=1}^{w(n)} p_i \tag{3.1}$$

where $w(n)$ identifies the number of contexts relevant to application A, and each distinct prime factor p_i of n is one of the CIDs relevant to that *Subject*. For example, in order to derive contexts of an application with $ACI = 77$, by prime factorisation of 77 we will obtain 7 and 11 as relevant CIDs. If the ACI number for an application is a prime number, it indicates that the *Subject* has only one context.

- *Contextual Query*: Contextual Query is a query that considers CIDs or ACIs as a part of the query.

3.1.2 Contextual Operations

Contextualisation is often dynamic and is always achieved by performing operations on IoT data based on the contexts that are relevant to each application. In this thesis, we propose three main classes of contextual operations, namely: *filter*, *aggregate*, and *infer*.

The *filter* operation applies to IoT data input and its output is a subset of the input IoT data that satisfies a contextualisation condition. This condition specifies which IoT data has contextual relevancy with one or more applications. Filtering does not modify the data. It only determines if the data should be considered in queries issued by the application.

The *aggregate* operation receives several IoT data inputs and mathematically or statistically processes them to compute the IoT data output. For example, in a room with multiple temperature sensors, an aggregate operation can calculate the average as an approximation of the room's temperature. Aggregation in contextualisation is any mathematical operation that can combine two or more input triples into a single output triple.

The *infer* operation is used to deduce new knowledge from the input IoT data and output it as data. For example, if RMIT University is in Melbourne and Melbourne is in Australia, infer can deduce that RMIT University is in Australia The contextual filtering, aggregation, and inference operations and their scalability properties are defined in more detail next.

- *Contextual Filter*: Processing IoT data generated by potentially millions of sensors

may not be possible due to the limited scalability of available computing resources. The contextual filtering operation labels triples in such a way that only triples relevant to at least one application will be considered. Contextual filtering converts triples to quads by adding the CID to the output triples. In this way, any triples that are not contextually relevant to any particular application can be excluded from the contextual queries. A contextual filter can use any labels as long as are uniquely defined and, as noted earlier, here we will use prime numbers for labelling. The CID calculated for triples indicates:

  - Triples that satisfy the contextual preferences of one or more applications, and
  - Triples that identically satisfy the same contextual preferences.

- *Contextual Aggregation*: This operation aggregates two or more context triples based on the similarity of applications context references. For example, considering the following triples:

$$< App3, Location, Melbourne, 7 >$$

$$< App3, Symptom, \text{“}Headache\text{”}, 3 >$$

$$< App3, Symptom, \text{“}Pain\text{”}, 5 >$$

$$< App2, Symptom, \text{“}Headache\text{”}, 3 >$$

$$< App2, Symptom, \text{“}Pain\text{”}, 5 >$$

$$< App1, Food, \text{“}Vegetarian\text{”}, 2 >$$

  In this set of triples there is no particular application interested in “Headache” or “Pain” individually. However, if there are applications that are interested in both “Headache” and “Pain” simultaneously additional triples with blank-nodes are generated [Figure 3.1] with the new CIDs as follows:

$$< App3, Location, Melbourne, 7 >$$

$$< App2, Aggregated, \_ : b, 11 >$$

$$< App3, Aggregated, \_ : b, 11 >$$

$$<_{:} b, Symptom, \text{“}Pain\text{”}, 3 >$$

$$<_{:} b, Symptom, \text{“}Headache\text{”}, 5 >$$

$$< App1, Food, \text{“}Vegetarian\text{”}, 2 >$$

Next, the CIDs of the triples that satisfy all the aggregated contexts are updated with the appropriate blank-node.

- Contextual Inference: Inference is the process of deducing new knowledge. Contextual inference takes contexts as input to deduce new knowledge. For example, suppose that application $App1$ has ACI number 210 and application $App2$ has ACI number 30. It can be inferred that all the contextual data relevant to $App2$ are also relevant to $App1$. Furthermore, by dividing 210 by 30 and conducting prime factorisation of the result it can be inferred that all the data relevant to $App1$ can be relevant to $App2$ if $CID = 7$ becomes a new contextual preference for $App2$.

3.1.3 Architecture (Contextualisation-as-a-Service)

Figure 3.2, presents the proposed ConTaaS architecture. Contextualisation using the depicted ConTaaS Architecture is performed in a sequence of steps. In step I, the raw data from IoT devices are annotated using semantic representations such as the SSN ontology [15]. Please note that such semantic annotations can be compliant with any other semantic data annotation framework, e.g., SensorML [98].

Semantically annotated IoT data is then converted to RDF triples and may be stored for further processing. Step II involves: 1) The application context that is specified and represented in the system as the domain context (this may also include user context such as user preferences), and 2) the contextual filter, aggregate, and infer operations described in Section 3.1.2. In Step III, the output of Step II (which is the contextualised data) will be presented to the IoT applications.

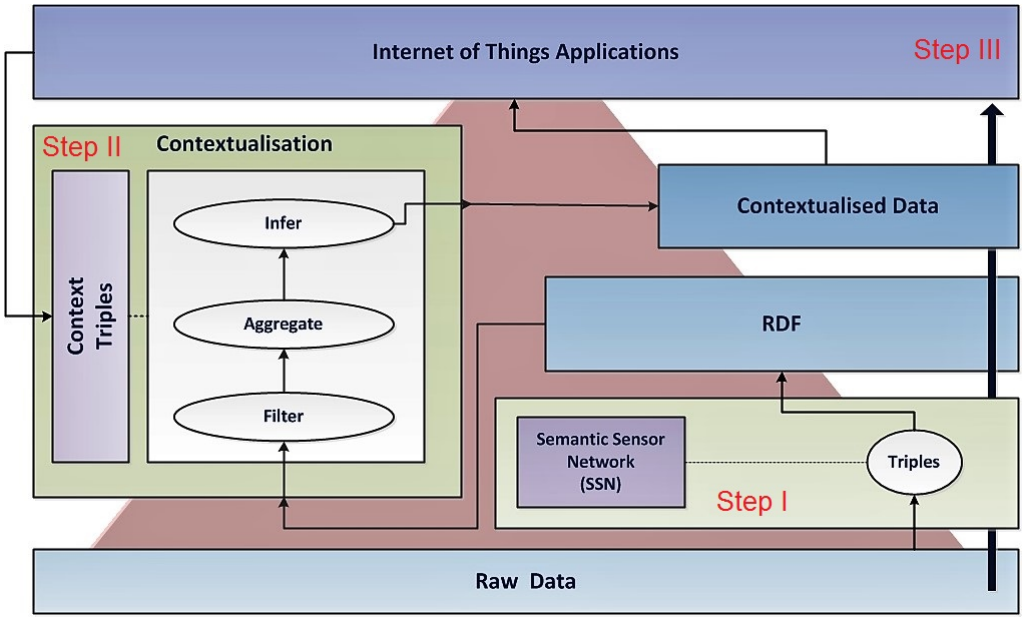

Figure 3.2: Cloud Solution Design of the ConTaaS.

The architecture proposed is general-purpose and can be used to realise IoT contextualisation in any domain. However, in this thesis we only give examples from the health and transportation domains.

To provide a specific ConTaaS example from the health domain consider the outbreak of the Ebola virus that occurred in March 2014 in Western Africa. This was an international public health emergency that according to the World Health Organisation resulted in more than 4500 deaths. To stop viral transmission via physical contact in such a public health emergency, it is vital to do the following as soon as possible:

1. Diagnose the virus as soon as possible. The most common symptoms of the Ebola virus are fever, fatigue, loss of appetite, vomiting, diarrhoea and headache [99].

2. Isolate patients by limiting contact with other people.

3. Start infection control and treatment.

Speeding up diagnosis by identifying persons with all or most of these symptoms and determining whether they travelled in a high-risk area during a specific time period can

be lifesaving. More specifically, to deal with Ebola, medical authorities must solve the following problems:

- Check all residents and travellers to determine whether they had been in Africa during the first few months of 2014 and whether they have Ebola symptoms.
- Determine whether those identified in (1) had any physical contact with anybody known to be infected.
- Transfer those identified in (2) to a hospital.

Just like Ebola, the World Health Organisation has also identified Zika as an international public health emergency. Zika is mainly transferred through the bites of infected Aedes mosquitoes and causes symptoms such as fever, conjunctivitis, joint pain and skin rash. Zika was originally considered to be a mild virus [100]. However, recent scientific research shows that Zika virus can cause microcephaly in the unborn babies of mothers who are infected by the virus during their pregnancy. The first step in mitigating Zika is similar to that for Ebola, as it is important to know if anyone visited Brazil or other high-Zika-risk areas at the same time they manifested Zika symptoms. However, as Zika is a mild virus and does not have any particular treatment, the only advice for infected people is to rest and avoid pregnancy until the virus disappears completely from the body, which takes approximately six months.

With current advances in mobile smartphone and wearable technology, it is now feasible to collect the data that are needed to mitigate such infectious diseases in people. Furthermore, records of the symptoms exhibited by individuals can also be obtained from hospitals, medical checks and wearable devices such as smartwatches. Such data streams collected from citizens are potentially massive and medical authorities need to frequently repeat their analyses of such evolving datasets. Managing and analysing such datasets requires sophisticated computing resources. ConTaaS has the potential to solve such problems by reducing the complexity of the data analysis query and extracting valuable knowledge from such data in near real-time. To illustrate these capabilities, we introduce

two data analysis applications for Ebola and Zika, namely, namely *EbolaApp*, and *ZikaApp*. *EbolaApp* processes context triples that are described as follows:

$$< EbolaApp, Location, Australia >$$

$$< EbolaApp, Symptom,' fever' >$$

$$< EbolaApp, Symptom,' fatigue' >$$

$$< EbolaApp, Symptom,' lossofappetite' >$$

$$< EbolaApp, Symptom,' vomiting' >$$

$$< EbolaApp, Symptom,' diarrhoea' >$$

$$< EbolaApp, Symptom,' headache' >$$

$$< EbolaApp, Visited, Africa >$$

Examples of *ZikaApp* context triples include:

$$< ZikaApp, Location, Australia >$$

$$< ZikaApp, Symptom,' fever' >$$

$$< ZikaApp, Symptom,' conjunctivitis' >$$

$$< ZikaApp, Symptom,' jointpain' >$$

$$< ZikaApp, Symptom,' skinrash' >$$

$$< ZikaApp, Visited, Brazil >$$

Table 3.1 shows sample data records from five persons. The CIDs of the data records in Figure 3.3 have been assigned by the contextual filter. The person named Ava does not have any CIDs. Lack of any CIDs indicates that this person is not relevant to either *EbolaApp* and *ZikaApp*. The next step of contextualisation involves contextual aggregation and generates the aggregated CIDs shown in Figure 3.4. The lack of an aggregated CID for

Table 3.1: Example symptoms of people screened by EbolaApp and ZikaApp (top) and corresponding CIDs (bottom).

Name	Symptom	Visited	Location
John	fever, fatigue, loss of appetite , diarrhoea	Africa, Sweden	Australia
Sophia	diarrhoea, vomiting, headache	France	Germany
Ava	sore throat, cough	Germany	Canada
Jacob	conjunctivitis, skin rash, fever, joint pain, headache	Brazil, Italy	Australia
Emily	conjunctivitis , joint pain, skin rash, fever	Brazil, Australia	Japan

Name	CID	Aggregate CIDs
John	13,7,5,37,31,11	13, 11
Sophia	37, 2, 17	
Ava		
Jacob	19, 3, 29, 13, 23, 17, 11	47,13,11
Emily	19, 23, 29, 3, 13	47,13

Sophia determines that she does not need to be considered further as she is not relevant to *EbolaApp* or *ZikaApp*.

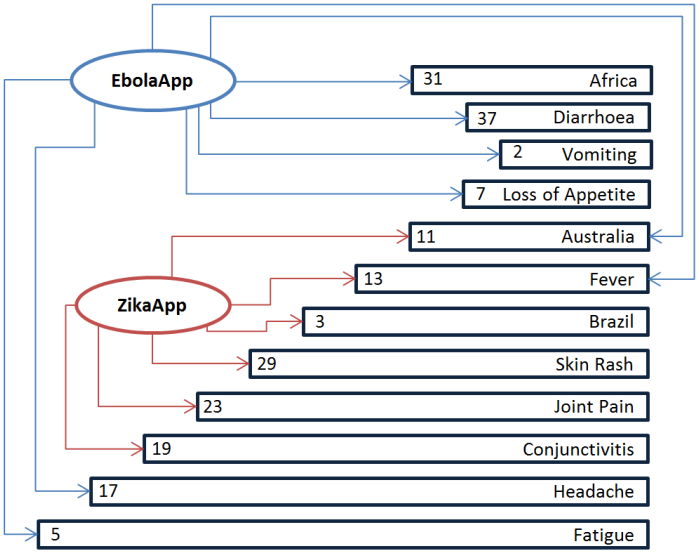

Figure 3.3: Context IDs.

Finally, the ACI numbers for these apps are calculated as $EbolaApp = 5863$ and $ZikaApp = 6721$. The ACI numbers for the remaining persons are computed (i.e., $John = 143$, $Jacob = 6721$ and $Emily = 611$), and based on this, ConTaaS can determine that

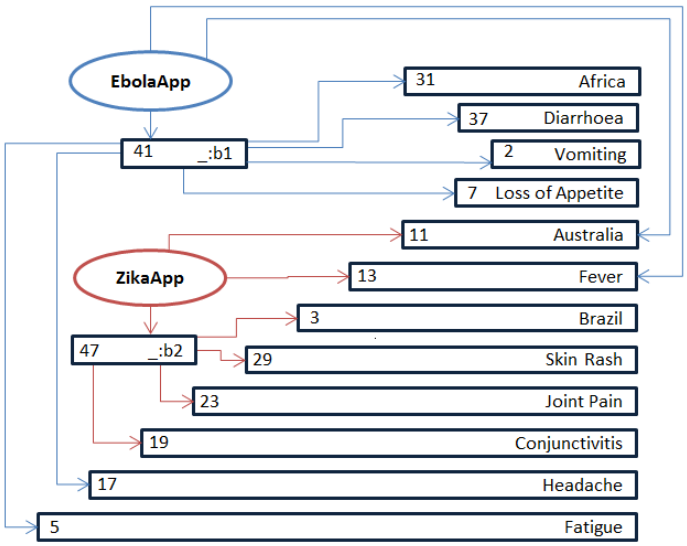

Figure 3.4: Aggregated context IDs.

Jacob is the only person that satisfies the *ZikaApp* context and no one satisfies the *EbolaApp* context. Contextual Inference can determine that Emily is at risk of Zika infection. However, her current location is not Australia. Subsequently, a list of triples with $ACI = 611$ can be used within Australia's borders to detect high-risk passengers.

CHAPTER 4

Contextualisation for Scalable Data Processing in the Internet of Things

In Chapter 3 we defined *contextualisation* as a process of identifying data relevant to an entity based on the entity's contextual information. For example, in a smart city setting, an entity may be a driver, a passenger, a vehicle, an area of the city or the entire city; while in a smart farming setting, entities may be crops, farmers and farms. In this chapter, we show that contextualisation drastically reduces the time and resources required to process (e.g., analyse) IoT data, provides unprecedented scalability and reduces the response time required to distil valuable information from massive amounts of IoT data (in the rest of this thesis, we use the term IoT scale data to refer to data generated by millions of IoT sensors).

In particular, contextualisation-based scalability of IoT scale data is achieved by the IoT contextual *filter*, *aggregate* and *infer* operations introduced in Section 3.1.2. These operations output drastically fewer data than what was input by eliminating IoT data from entities that are irrelevant to the IoT application at hand.

This chapter mainly focuses on IoT applications that can be represented as OODA loops [2]. Such IoT applications consist of the following four phases: *Observation* involves collecting IoT data; *Orientation* includes contextualising such IoT data to reduce the dataset to a subset that is relevant to the aims of the application; *Decision* pertains to making appropriate decisions; and *Action* consists of an actuation that is based on the decisions made [Figure 4.1]. Each OODA phase may be comprised of an additional OODA loop, so OODA-based IoT applications can be nested. Given that there is no shortage of IoT applications that can be modelled as OODA loops (a few examples include finding parking in a smart city, deciding what crop to plant in a farm, monitoring and improving productivity in a manufacturing plant, preventing refuelling and driving away without paying, deciding what to display on an advertising board based on the past purchases of those in close proximity, and detecting environmental pollution and raising alerts), we believe that focusing on OODA-based IoT applications does not limit the importance of our contribution. Therefore, in this chapter, we focus on how contextualisation can be applied to an OODA-based IoT application (we use a smart city application as a case study) and the show the scalability benefits of contextualisation by benchmarking and comparing data analyses made with and without contextualisation.

We selected smart cities as a case study for illustrating contextualisation benefits because they include millions of sensors. However, existing solutions for capturing data from all these sensors, analysing them and providing recommendations in near real-time are currently unavailable. Furthermore, IoT sensors and data sources are added every day, while others are replaced or taken offline. This IoT environment volatility makes the IoT data processing problem harder.

Contextualisation in our smart city case study includes the following steps: (1) consideration of multiple contexts originating from drivers, vehicles and the smart city, (2) continuously computing relevant contexts, and (3) providing an instant response to common parking queries. Contextualisation-based scalability is achieved by finding shared contexts among multiple users and processing them only once to efficiently answer all user queries. A greater number of users and queries typically results in more scalability.

As noted earlier, contextualisation can achieve similar scalability benefits in any OODA-based application. This will involve contextualisation and benchmarking processes that are similar to those described for the smart city case study presented in the following section. Chapter 5 illustrates the benefits of contextualisation in maintaining privacy and security. This chapter also include IoT application studies that are not OODA-based and illustrate the generality of contextualisation and its outcomes.

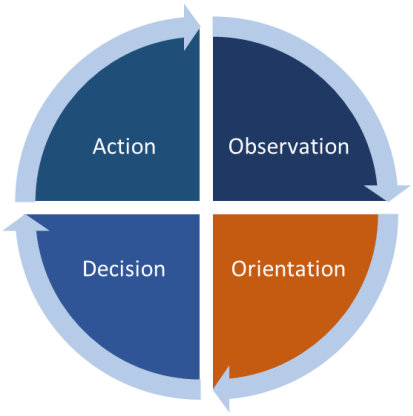

Figure 4.1: The observation, orientation, decision, and action loop.

4.1 Development of a Scalable Parking Recommender

There are many existing parking recommendation solutions that direct drivers to empty parking spaces. Some even provide an estimated average waiting time. Unlike existing solutions, which only consider parking information (e.g., available parking spaces and their locations), the contextualisation approach we propose in this chapter takes into account each driver's context, which may include their preferences (e.g., covered parking), driving experience (e.g., avoid narrow parking spaces), their car's location (e.g., collected from the driver's smartphone), the vehicle's properties (e.g., type, length, height, etc.), and other parking properties (e.g., shaded, covered, etc.). The principal challenge in delivering such contextualised services lies in the ability to contextualise IoT scale data (based on available contextual information) and to do this efficiently and in near real-time.

Parking is becoming an expensive resource in any major city and finding the most appropriate parking space is always regarded as a challenge. Existing solutions (Section 4.4) follow the following approaches to make parking recommendations:

- Use IoT data from parking facilities (e.g., from their parking sensors).
- Utilise vehicle-provided data (e.g., from onboard accelerometers) to compute empty parking spots based on each vehicle's kinetic state (i.e., moving, stopped, etc.) and location (such information is typically crowd-sourced [101]).
- Employing machine learning models to predict queue lengths for parking in shopping centres.

Most of these approaches do not scale up to process more context(s) in real-time. Unlike such existing approaches, the solution proposed in this thesis provides the following:

- Allows IoT services to take into consideration multiple contexts originating from each driver, their car and a smart city section of interest (e.g., parking in a shopping centre),
- Permits instant responses to common parking queries by continuously contextualising, and
- Combines the contexts of multiple drivers to efficiently answer parking queries.

The architecture of the smart parking recommendation application that incorporates the proposed contextualisation and processing of IoT data is illustrated in Figure 4.2. The data will be collected from parking sensors as well as drivers. The collected data will be stored in a database. Then, the contextualisation server will contextualise the data and store it in the contextualised data database. Car ontology will be used to describe the specifications of the cars (e.g., size). The smart parking recommender exemplifies the need to contextualise IoT data and the advantages it provides.

Contextualisation of IoT data involves the following:

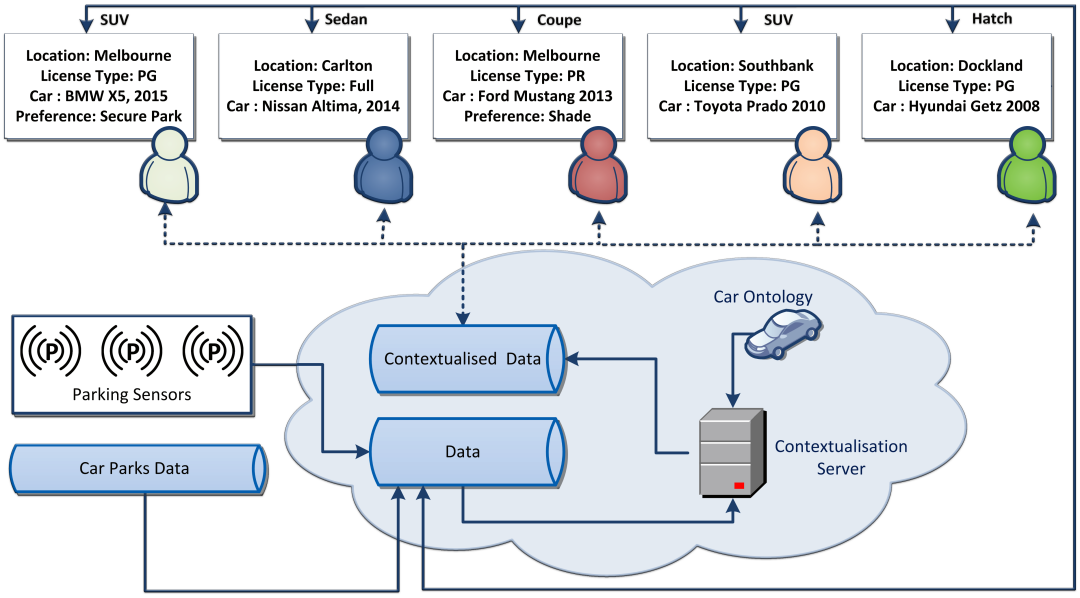

Figure 4.2: Architecture of the smart parking recommender.

1. Context Collection (and deduction): User context information collected from user's smart-phones, wearable devices, or manually provided by the user. Moreover, cloud services can help to deduce new context information from the collected context. For example, a service equipped with car ontology and related data can deduce the user's car size by knowing the car manufacturer and model.

2. IoT Contextualisation: The contextualisation of IoT data in this chapter is based on two main operations including contextual filter and contextual aggregation that were introduced in Section 3.1.2. Contextual filter filters the data originating from IoT devices and services based on any given context. For example, data received from a parking sensor located in a particular location (e.g., a parking space in a Melbourne suburb) can be excluded from further data processing and related queries whenever there is no particular user looking for parking in that particular location. Contextual Aggregation combines potentially filtered data based on contextual similarities and relevance. For example, if all the current users searching for parking spots in Melbourne have SUV vehicles we can then aggregate the SUV and Melbourne contexts and treat it as a new context.

3. Delivery of the contextualised data.

To develop the smart parking recommender we utilised ConTaaS described in Section 3.2. To perform a Contextual Filter operation (Section 3.1.2), a set of available contexts from drivers in each particular city area is initially created. Next, a unique context identifier (a prime number in this thesis) is assigned to each context. Finally, all available parking spots in the selected city area are searched, and the resulting triples are converted from N-triples to N-quads by adding the unique context identifier to the N-triples. If the triple is already N-quad (indicating that the triple is already contextualised for other contexts), the context identifier is updated by multiplying the prime number of the context identifier with the number the triple already has. For instance, if a secure parking space $Parking1$ has a context with identifier 11, the triple of $Parking1$ is converted as follows:

$$< Parking1, hasType, closed > to < Parking1, hasType, closed, 11 >$$

Similarly, if $Parking2$ already has another context with identifier 7, its triple is converted as follows:

$$< Parking1, hasType, closed, 7 > to < Parking1, hasType, closed, 77 >$$

After performing the contextual filter operation, all of the triples that remain in the form of N-triples are not relevant and can be eliminated from further processing. To perform contextual aggregation, multiple contexts are merged and a new prime number is assigned. For example, if in one particular location all the drivers are searching for parking spots with one context followed by another (e.g., large size and secure), then these two contexts are aggregable.

The above contextualisation system was implemented in the Java programming language. It uses Apache Jena [102] for the semantic triple store and the ARQ Query Engine [103] for SPARQL to query the contextualised data. For evaluation, the system was deployed on an Amazon EC2 [7] "M3 General Purpose" instance with 30 GB ram and 8 'vCPU'.

4.2 Evaluation Dataset for the Scalable Parking Recommender

Consider 50,000 users (i.e., drivers) searching for parking spots in the Central Business District (CBD) and surrounding suburbs [Figure 4.3] that are provided by the City of Melbourne [1] [Figure 4.4]. To evaluate the scalable parking recommender outlined in Section 4.1, we use this dataset and augment it with a synthetic dataset of parking sizes and descriptions.

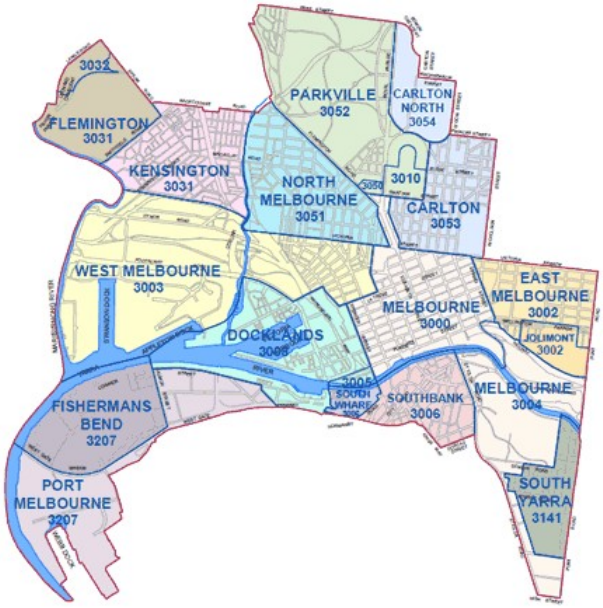

Figure 4.3: Melbourne central business district and surrounding suburbs.

The information considered in the dataset is as following:

- Driver information
  - Licence type (Full = experienced, Green P = drivers with one year experience, Red P = beginners).
  - Car Model Specification (e.g., BMW X5 2015)
  - Preferences (e.g., secure car park, car park with shade, etc.)

[1] http://data.melbourne.vic.gov.au

- Parking information
  - Location (suburbs)
  - Type (closed, open-air)
  - Space (number of available spots)
  - Acceptable car body types (SUV, hatch, sedan, etc.)
  - Weather conditions (sunny, rainy, snowy, etc.)

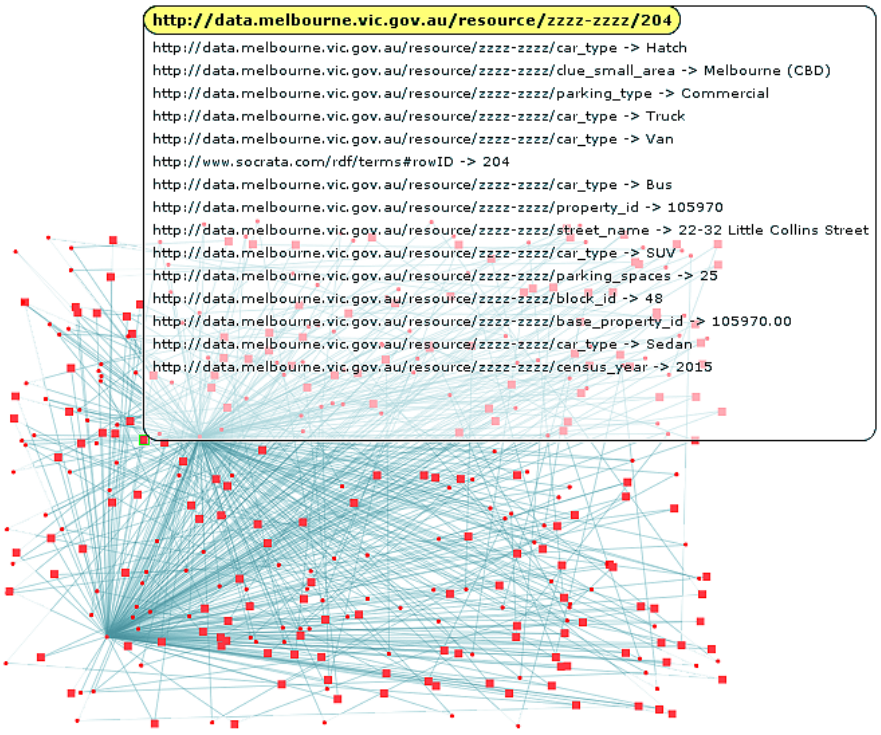

Figure 4.4: Parking spots dataset.

4.3 Performance and Scalability Evaluation of the Scalable Parking Recommender

The objective of this evaluation is to validate the following hypothesis: Contextualisation helps process driver queries faster and can handle IoT scale IoT datasets. Since we keep

track of shared contexts among multiple users, each individual query is independent (please note that this is different from typical query caching mechanisms). For instance, a user may share their contextual preference (e.g., shaded parking spots) with a second user while another contextual preference (e.g., acceptable car type) is shared with a third user.

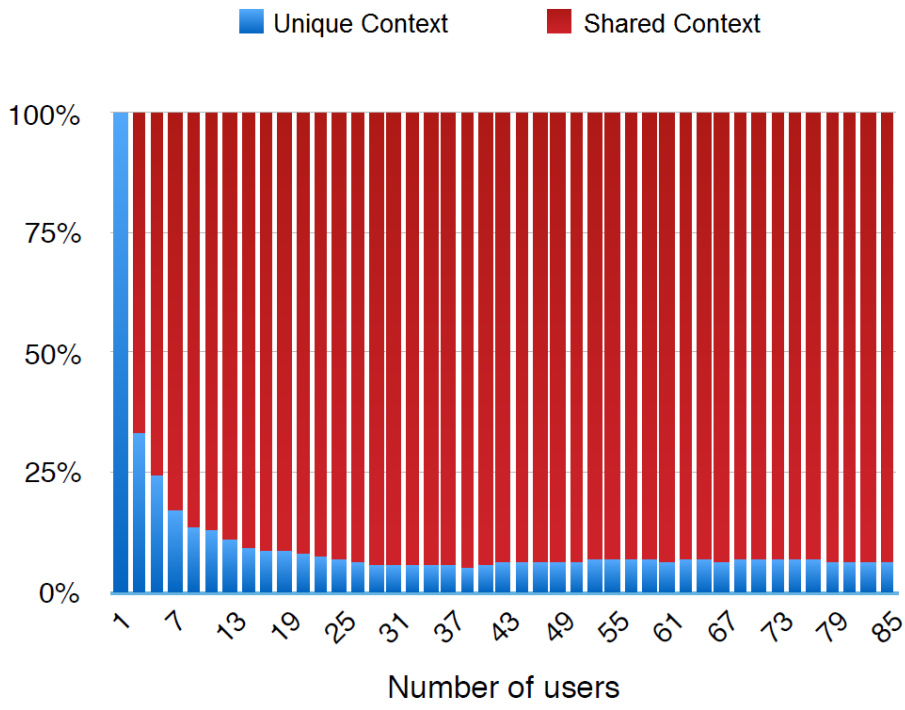

Figure 4.5: Percentage of shared contexts according to the number of users.

Figure 4.5 presents the percentages of shared contexts among 85 random users. As in many real-world situations, the shared context (and related context data) among users increases rapidly as the number of users increases. This result also illustrates the fact that the number of shared contexts among users of IoT services (as illustrated in the smart parking recommender), is typically much larger than the number of unique contexts.

Figure 4.6 shows the the percentage of active users' contexts relevant to the parking data at any given moment. For example, with 85 driver contexts in the system and 3368 parking data triples, the total number of users whose contexts match the parking data triples (e.g., finding enclosed parking in Melbourne CBD) is 27.5%.

In Figure 4.7, the total query processing times with and without the contextualisation

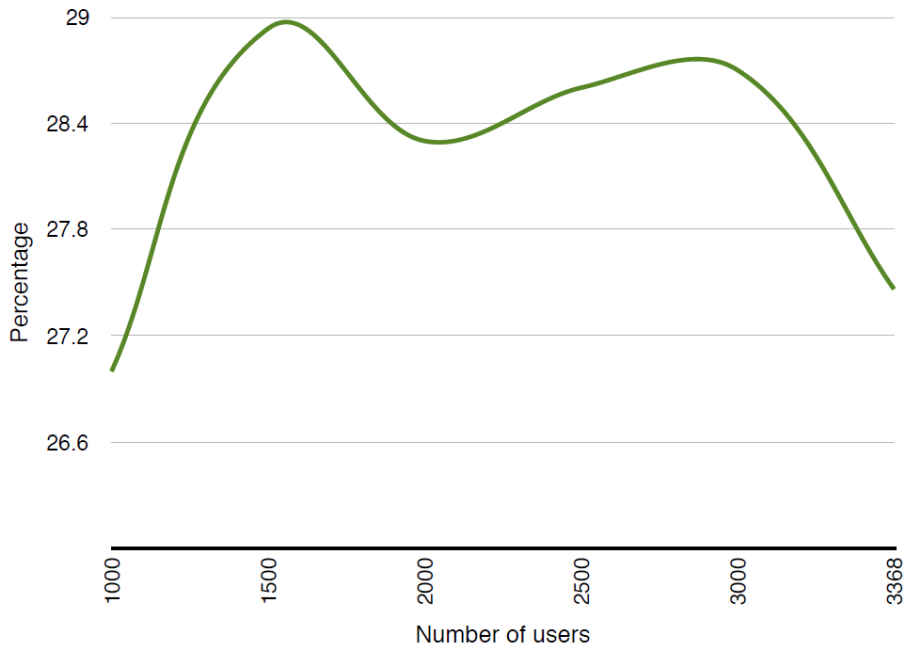

Figure 4.6: Percentage of the relevant data according to the number of users.

process are presented. Please note that as the similarity among the context preferences of drivers increases (as in Figure 4.5), contextualisation can greatly help by significantly reducing the total query processing time. This is due to the fact that the proposed contextualisation solution has already resolved the queries of users with similar contexts. Hence, to satisfy a request from a new user whose contexts match those of existing users in the system, contextualisation simply maps the responses of existing user queries by matching the relevant contexts to the new requests.

Finally, in Figure 6, the total time for the contextualisation process is presented, which includes converting driver and location (smart city) information to relevant N-Quads and resolving driver queries. As the result indicates, after 10,000 users, the process does not continue to improve significantly due to the fact that the shared contexts are the major contexts of all the users. This result validates the hypothesis, i.e., that contextualisation performs much better when the amount of data is of the IoT scale.

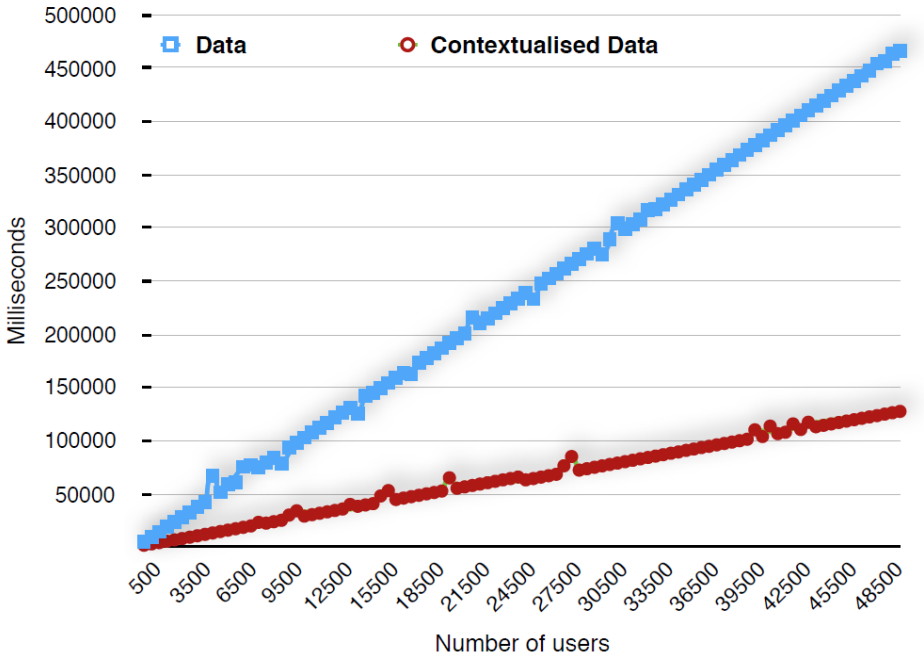

Figure 4.7: Query Time Comparison

4.4 Parking Recommenders in the Literature

In [104, 105], the authors describe the infrastructure currently embedded in smart cities that allows the development of smart parking solutions. Parking is considered a major problem in developing and developed cities. With the advent of technologies such as the IoT, cloud computing, and Big Data analytics tools, there have been a number of recent works focusing on smart parking management approaches. Most of these approaches focus on the following two dimensions for estimating parking availability: (1) Development of new IoT devices for different parking situations (e.g. garages, shopping centres, etc.) [106, 107, 108, 109, 110, 111], and (2) developing algorithms and methodologies in particular machine learning and queuing theory approaches [112, 113, 114, 115, 116, 117, 118].

In [107], the authors propose a Vehicular Ad-hoc Network (VANET) based smart parking scheme using vehicular communications through road-based infrastructure (road side units). It provides real-time parking navigation in large spaces and ensures user

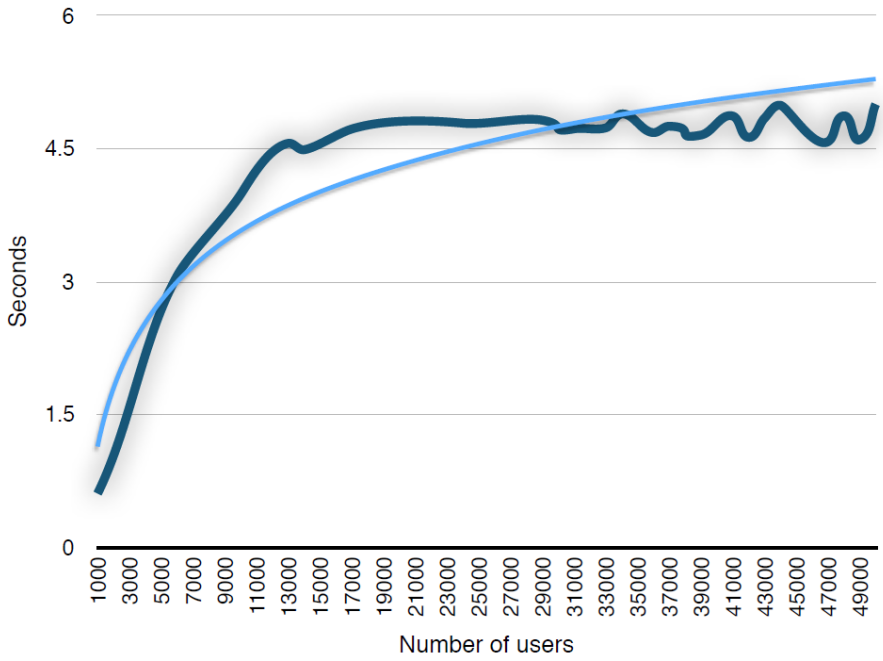

Figure 4.8: Contextualisation processing time according to number of users.

privacy. In [113, 110], the authors described a smartphone-based crowd-sensing approach providing parking place recommendations. They built statistical models of sensor data obtained from smart-phones to detect events such as arrival and departure from a parking spot (e.g., using accelerometer data). In [108], the authors proposed a parking estimation system using Arduino-based ultrasound sensors. In [111], the authors made use of existing IoT infrastructure deployed in parking spaces to provide a cloud-based parking space finder service. The focus of this work was on the middleware required to deliver the parking recommendation service. Similarly, in [106] the authors presented an architecture for parking management in smart cities. This system makes use of custom-developed IoT hardware, in particular, retractable bollards, magnetic loops to detect occupancy, Radio-Frequency Identification (RFID) readers, and ZigBee-based wireless transceivers. In [118], the authors proposed a vision sensor (camera)-based approach for estimating and recommending empty parking spots. In [109], the authors proposed a technique to predict parking space availability. Their approach was to identify the key features that

best describe parking availability and to use various machine learning algorithms such as regression trees and support vector regression to determine the strength and accuracy of these algorithms. In [114], the authors used anomaly detection and clustering to detect interesting patterns, such as the distribution of heavily-used parking spots, and to compare pricing versus security. In [117], the authors employ mixed-integer linear programming to solve the same problem. Their solution reduces the overall time required to find a parking spot. The solution proposed in [116] uses an online demand management model to provide parking spot recommendations to electric vehicles, while [115] proposed the use of contextual information from users and smart parking infrastructure to make more precise recommendations. This work concisely describes how context is represented and used.

Most of the above solutions focus on using IoT data to provide recommendations, with either very little or no consideration of context information obtainable from drivers or the smart city. Moreover, most of these approaches are tailor-made to work for closed garages or specific shopping malls. There is no consistent way of representing parking and driver data and all current approaches use different architectures. On the contrary, the proposed approach provides a unified solution for representing IoT data obtained from devices such as sensors, cars, wearables and smartphones, and also to efficiently query all such data.

CHAPTER 5

Contextualisation for Scalable Security and Privacy in Internet of Things

In Chapter 4, we described how the contextualisation of IoT data can improve the scalability of IoT applications. Moreover, we showed that contextualisation can also provide better security and privacy in the IoT. More specifically, Section 5.1 and Section 5.2, describe how contextualisation also improves the scalability of the security and access control in IoT via contextual access control and granular disclosure control.

5.1 Contextualisation-based Scalable Access Control for the Internet of Things

In this section, we introduce a novel obfuscation technique for IoT data that uses a combination of lightweight digital watermarking and scalable contextualisation [Chapter 3]. Digital watermarking is the practice of embedding extra information within digital content itself, which is also called host data, in a matter that does not interfere with the normal usage of host data [119]. Such techniques have been mainly used for the digital

rights management of multimedia content. The watermarking technique proposed in this section affects the sensitive data more or less depending on the disclosure privileges of the data requester. In particular, the proposed watermarking technique provides more effective obfuscation for the most sensitive data by increasing the intensity of the watermark. In the literature, little research, if any, has been conducted for perturbing sensitive IoT data using digital watermarking. In contrast to many other data obfuscation techniques, such as those described in [120] and [121], the proposed obfuscation technique is reversible only by authenticated users with the appropriate disclosure privilege(s). Since there is no information loss, in the proposed approach, data can be freely modified and retrieved by repeatedly having the right obfuscating parameters. In this regard, the technique presented in this section is reminiscent of role-based access control in which only users with matching roles can access the target data [120].

Embedding watermarks introduces a tuneable distortion in the target data. This enables masking any data for any application where privacy is of great importance.

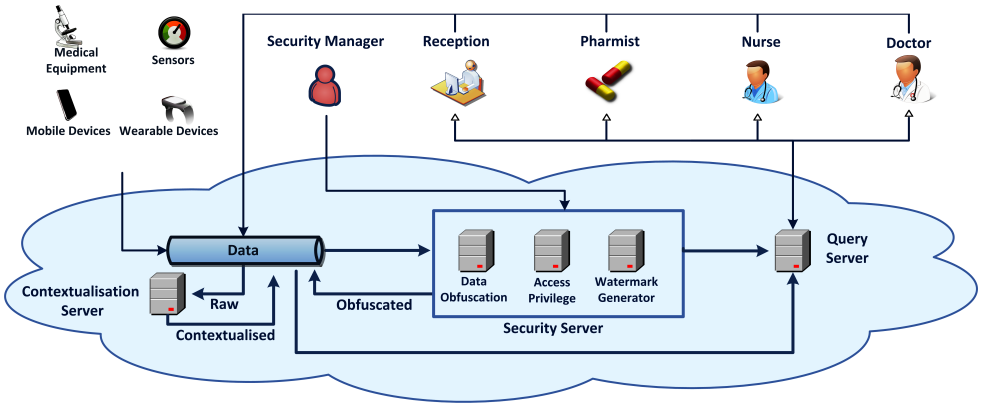

Figure 5.1: Conceptual architecture of the contextualisation-based scalable access control for the IoT.

The main innovation in the proposed IoT data obfuscation technique is in combining watermarking with a variation of our highly scalable contextualisation technique (i.e., ConTaaS; Chapter 3). As discussed in Section 3.1.3, ConTaaS excludes irrelevant data from consideration and reduces the volume of IoT data that needs to be managed and

analysed by IoT applications. This contextualisation-driven data reduction approach also improves the scalability and performance of security and privacy-preservation mechanisms implemented in the IoT. Moreover, it reduces the amount of computation (often referred to as *reasoning*) required to understand and measure the corresponding privilege level for accessing each specific data point.

5.1.1 Contextualised Role-based Data Disclosure Control

This section describes a novel IoT security service that utilises watermarking to provide confidentiality, controlled disclosure, authentication, and authorisation. More specifically, Section 5.1.1.1 describes control of IoT data disclosure, Section 5.1.1.2 presents a nested, role-based, disclosure model, Section 5.1.1.3 describes the conceptual architecture, Section 5.1.1.4 presents the data model, Section 5.1.1.6 describes the Watermarking as a Service model, Section 5.1.1.7 describes the obfuscation technique, and Section 5.1.1.8 presents the delivery of the obfuscated data utilised for role-based disclosure control.

5.1.1.1 Controlling IoT Data Disclosure

For contextual role-based disclosure, context is any information that can describe or impact the disclosure privilege of the IoT data for the relevant roles. In this Section we propose an IoT security service that provides aspects of confidentiality, controlled disclosure, authentication, and authorisation. Implementation of this service involves using cloud computing infrastructure and service-oriented computing principles to enable usage by anybody. In this section, the focus is only on a contextualised authentication and data disclosure control for IoT data. However, the proposed service is also capable of performing other IoT security services.

5.1.1.2 Role-based Disclosure Privilege Model

To explain the IoT security service, consider a nested role-based model of security privileges [Figure 5.2], where the lowest privilege is granted to individuals located in the inner-most region and the highest privilege is granted to individuals in the outer-most region. This

means that an individual in a particular region will also have all the privileges of users in regions contained within it. The number of privilege regions (PR) is denoted by d, where d is the number of predefined roles in the system. Moreover, each PR has a unique identifier assigned to it, denoted as rid. Therefore, a user with rid_k has all the disclosure privileges granted to all other users within regions rid_1 to rid_{k-1}. Please note that the region identifiers are only known to the security service. These notations and others used to describe role-base disclosure are listed in Table 5.1.

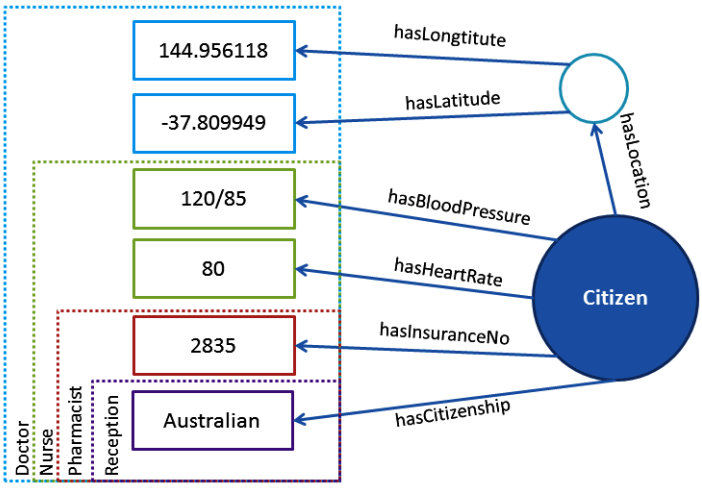

Figure 5.2: Role-based privilege model for the health-care scenario.

To grant access to data, the proposed IoT security service exploits knowledge of the existing roles of authenticating users (who interact with the service by issuing queries). Therefore, every user belongs to a specific PR and the security service verifies this membership via a key that is assigned to every user. If the user's key is valid, the associated region Identification (ID), i.e., rid, is retrieved to de-obfuscate the query result later.

To further explain the proposed role-based model, the following notation is used: Every region r_k is associated with a pair (key_k, rid_k), where key_k is the secret key for all users belonging to that region and rid is as defined above. Secret keys and the corresponding region identifiers are generated by taking advantage of a known ensemble construction, such as Kasami. More specifically, the binding key to a region is a PN code and the associated

Table 5.1: Notation used to describe role-base disclosure.

Symbol	Meaning
d	Number of disclosure privilege regions
r_k	$k-th$ region index ($1 \leq k \leq d$)
rid_k	$k-th$ region identifier
key_k	Binding key for region rid_k
$skey_i$	Session key for user i
Ξ	An ensemble of Pseudo Noise (PN) sequences with desired correlation properties
l	Length of the PN sequence/code
σ	Composite template key
$rindex_o$	Region index attached to data object O
u_i	i-th user/data-requestor
$Hash$	Hash functions, example SHA-1 or MD5
$h_{k,1}$, $h_{k,2}$	first half and the second half of the calculated hash for the region r_k
α_k	Scale factor (watermark amolitude) corresponding to region r_k
$puKey_{DS}$	Public key of the data delivery service
$prKey_{DS}$	Private key of the data delivery service

rid is the spatial shift value that can be used to generate other orthogonal PN codes as described in Section 5.1.1.5. Next, these notations are used to describe in detail how these values are generated and how the security service can retrieve the associated rids without having access to the individual keys (i.e., the PN codes).

5.1.1.3 Conceptual Architecture for Role-based Disclosure

IoT data are typically captured from various internet-connected devices such as smartphones, wearable devices and sensors. Such data is often not protected. Applying traditional security techniques such as encryption is not feasible due to the resource limitations of IoT devices. In this section, the aim is to protect IoT device data with a lightweight and scalable technique by using contextualisation and watermarking. Data disclosure control is achieved by a role-based privilege model, as described earlier.

Figure. 5.1 illustrates the conceptual architecture of the proposed novel role-based data disclosure control. The primary components of this architecture are the contextualisation, (IoT) security, and data delivery services. ConTaaS (introduced in Section 3.2) is used to contextually filter and contextually aggregate triples based on the their relevancy to the available roles. The contextualisation service deduces the associated access privileges (which

are represented by the label $rinedx_O$ as described in Section 5.1.1.7) for each individual data point based on the privilege ontology [1] and the policies defined by a security manager. The security service is comprised of disclosure privilege, data obfuscation and watermarking functions and is responsible for providing defined policies to the contextualisation service, watermarking data, and providing role-based authentication using watermarks. Finally, the data delivery service is provides privacy-preserving delivery of the query results to the users. This conceptual architecture will be explained via an example presented in Section 5.1.2.

5.1.1.4 Modelling and Querying Data for Role-based Disclosure

Data modelling and querying are based on existing semantic web standards, such as RDF and SPARQL. More specifically RDF-based N-triples [122] are utilised to describe IoT data in the form of $< Subject, Predicate, Object >$. Subject is the identifier of the entity that the data is describing; Object is the description of the Subject in terms of the relation described in Predicate. For example, a triple such as $< Patient1, hasHeartrate, 85 >$ indicates that $Patient1$ heart rate was 80 betas per minute. Data queries are formulated using SPARQL.

5.1.1.5 Digital Watermarking

Digital watermarking is currently used in the multimedia domain to provide copyright protection [119]. The watermark constitutes a piece of secret information which is blended within the digital content in such a way that it is *invisible* to the consumer. Recently, digital watermarking is also has been used for authenticate non-media data, such as time-series, biological sequences, graphs, spatial, spatio-temporal, and streaming data [123]. In such applications, watermark invisibility to human perception is no longer ensured, but interference with such data is detectable. Given that the IoT typically generates IoT data streams, digital watermarking in the IoT focuses on streaming IoT device data.

[1] Ontology is a formal way of describing taxonomies and defining the structure of knowledge. The privilege ontology is a knowledge repository describing the relationship between the privacy-sensitivity and the roles of data.

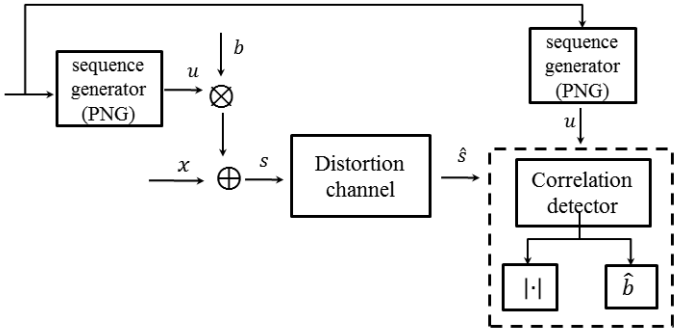

Figure 5.3: Main components of a spread spectrum digital watermarking system.

Spread Spectrum (SS) is a popular approach for the digital watermarking of IoT data. A watermark is constructed as a random sequence that is *imperceptibly* inserted in a spread-spectrum-like fashion into the IoT data values. Such sequences are often near-orthogonal codes of +1 and -1 symbols (i.e., streams of +1 and -1), and can be decoded through correlation between code pairs [Section 5.1.1.6]. The data security provided by the SS watermarking technique is highly dependent on the spreading sequences. Using truly random sequences is ideal so that no one other than the encoder can predict the watermark. Unfortunately, appropriate hardware for generating such codes is not generally available [124]. Besides, the decoder must generate the same random code to retrieve the watermarked IoT data, which is impossible as they are totally random. Instead, a PN sequence [Section 5.1.1.6] is used to resemble the random behaviour.

Randomness is an ensemble property and cannot be achieved in a single sequence [125]. To encode an ensemble of PN codes in the same data stream (either one data stream or an aggregated data stream such as a moving average [126]]), two other properties are needed: high auto-correlation of a PN code and low cross-correlation between any two PN codes in the same code family or set. Auto-correlation refers to the degree of correspondence between a code and a phase-shifted replica of itself. Cross-correlation is defined the degrees of agreement and disagreement between two codes.

An ensemble of periodic PN sequences with low off-peak auto-correlation and cross-correlation can be generated using maximal length sequences or m-sequences [125].

Table 5.2: Comparison of PN family sets.

Type	Length (l)	Maximum correlation bound	Family size	Normalised linear complexity
Gold	$2^n - 1$	$2^{(n+1)/2} - 1$ or $2^{(n+2)/2} - 1$	$l + 2$	$\frac{2n}{2^n - 1} \cong 0$
Small-Kasami	$2^{2n} - 1$	$\sqrt{l}$	$\sqrt{l}$	$\frac{1.5n}{2^{2n} - 1} \cong 0$
Large Kasami	$2^{4n+2} - 1$	$2\sqrt{l}$	$l \times \sqrt{l}$	$\frac{2n}{2^{2n} - 1} \cong 0$

For example, in [127], an ensemble of l PN codes are shifted versions of a primitive m-sequence. Nearly n bits can be encoded through the phase, i.e., the number of spatial shifts (with a cyclic wrap-around), of a $l = 2^n - 1$. To increase the number of possible PN codes, more primitive PN codes with low cross-correlation can be used. Two of the known ensembles of such are Gold and Kasami [126]. Gold is a set of $2^n + 1$ sequences of length $l = 2^n - 1$, $(n \neq 4)$ whose cross-correlation is three valued. For n odd, the values are optimal and bounded by $2^{(n+1)/2} - 1$. Kasami codes of length $l = 2^n - 1$ only exist for even values of n. There are two classes of Kasami sequences, namely small sets, and large sets. The small sets has better correlation properties compared to the gold and large sets. The summary of the described PN codes is listed in Table 5.2. Linear complexity in this table refers to the security level of PN codes in terms of unauthorised detection.

5.1.1.6 Watermark as a Service

Watermark generation and exchange are delivered 'as a service' to users in order to satisfy disclosure privilege requirements. This requires a trusted third party that only knows the summation of all shifted keys associated with all defined roles and maintains the *rid* of the data requester. In contrast, the contextualisation service is not trusted and therefore, only the obfuscated versions of data are stored in its database [Section 5.1.1.7].

Suppose there is an ensemble of PN sequences of length l (with low off-peak auto-correlation and cross-correlation), denoted as $\Xi = PN_1, PN_2, ..., PN_{|\Xi|}$. Examples of such ensembles are the gold and Kasami sets. From this set, a unique sequence $PN_j (1 \leq j \leq |\Xi|)$ is chosen as the key for all users in region r_k . On the server side, the received key is used

to retrieve the associated rid_k. Please note that this number is an integer value for shifting the chosen PN_j and should be less than the PN length i.e. $rid_k < l$; otherwise, the shifted PN codes will not be unique (because of cyclic wrap-around). This value must be retrieved before granting data access to the user.

Apart from the PN codes which are identical for all users in the same region, every user obtains a session key that makes the de-obfuscation process dependent on their unique credentials and therefore enhances the security of the proposed technique. This session key is generated by security service and is exchanged using a secure exchange protocol such as Secure Sockets Layer (SSL)/Transport Layer Security (TLS). The session key for user i is denoted as $skey_i$.

The process for retrieving region IDs (rids) is equivalent to the de-spreading of the secret PN code (keys). This is done by a correlation operation between the template PN sequence and the received PN code from the user. The underlying principle behind the decoding process is based on the observation that if in a cross-correlation between an embedded PN sequence and a template, the two differ only by a shift, then the correlation peak will be shifted by that amount. More detailed information about the decoding process can be found in [126].

The template sequence, σ is a composite PN sequence obtained from the summation of several shifted versions of the original PN codes that are assigned to different regions, i.e. $\sigma = \sum_{i=1}^{d} shift(PN_i, rid_i)$, where $shift()$ represent a spatial shift with cyclic wrap around. Then, the periodic correlation is expressed as $\rho(\tau) = \sum_{j=1}^{l-1} \sigma(j) PN_i(j+\tau)$. If PN_i is the correct key, the correlation values (ρ) reveal a significant peak at the position corresponding to rid_i. This value is passed to the data delivery service to de-obfuscate the data prior to sending it back to the user. If the key is not valid, then the retrieved rid will be incorrect which means the original information cannot be retrieved successfully.

The above disclosure control has three main advantages:

- First, the PN codes can be generated on the fly in the most compact Linear Feedback Shift Registers using Field-Programmable Gate Array (FPGA) which is a lightweight and cost-effective approach.

- Second, storing one composite key instead of individual keys eases the key management burden at the server end and makes the proposed scheme more scalable compared to storing different keys for different users. This additionally increases the security of the proposed scheme if the security service is compromised.

- Third, session keys are used to afford the ability to have a fine-grained disclosure privilege for authenticated users.

5.1.1.7 Data Obfuscation

Before we explain the obfuscation process, recall that after contextualisation, a hierarchy of data is constructed for all required privileges. This is based on set of privilege policies provided by an administrator, such as "The ECG data can only be accessed by doctors", or "The blood pressure data can be accessed by nurses and doctors". Based on these policies and the role-based privilege model, the contextualisation service attaches a related tag to the IoT data, i.e., for every data object O, the region index $rindex_O$ is attached.

If the IoT data storage is located outside the trust enclave, the original data values are modified using an Obfuscation Function (OF), This is done in a way that only authenticated users with the right privileges can de-obfuscate the data and retrieve their original values. In the literature, there are many OFs for this purpose. As discussed in Section 2.3, random noise addition generated from a probabilistic distribution (such as Laplacian) can be used for OF. However, the use of truly random numbers makes the de-obfuscation process non-reversible. If highly sensitive data is involved (e.g., medical data), a reversible OF is desired. In this case, a *deterministic* OF by means of digital watermarking techniques can be used to provide a reversible obfuscation transformation.

If an additive watermarking approach is utilised for this the obfuscated data is simply constructed by adding a scaled watermark to the data. Following the notations, the watermarked data is obtained as $O^w = O + scale(w)$. Traditionally, $scale()$ updates the amplitude of the watermark w to make it imperceptible from the host data. If better obfuscation is desired, watermarks with larger scale values are embedded.

Our data obfuscation solution adds two more amendments to the above watermark encoding scheme to make the obfuscation technique dependent on the users' privileges. First, the embedded watermark is a keyed hashed value of the retrieved *rid* with the composite key σ. This makes the data obfuscation dependent on the security service and prevents calculation of the hash if an adversary intercepts the *rid*. Second, the watermark amplitude is tuned in such a way that for data with higher sensitivity, the scale factor is larger. The rationale for the latter is that more sensitive data require stronger privacy preservation than data that can be accessed with the lower disclosure privilege. Since the proposed de-obfuscation technique is reversible, it can add a large amplitude of watermark to the original data and, subsequently, subtract the added value to retrieve the original data. Again, these values should be selected a priori, based on the desired privilege policies.

In summary, the obfuscated or watermarked data object is generated as $O^w = O + \alpha_k \times decimal(Hash(rid_k, \sigma))$, where α_k is the associated scale factor for the region r_k and $decimal()$ returns the decimal hash value. Once a query is issued, the security service retrieves the parameters for that user and passes it to the data delivery service to de-obfuscate the data. However, one issue remains. For IoT data that can be accessed from multiple regions, the data delivery server cannot distinguish the obfuscating parameters. Consequently, invalid values will be reported.

A solution to this problem involves changing the watermark value and maintaining an extra table (such as Table 5.3) in the security service. This achieves the following: The hash value, $Hash(rid_k, \sigma)$ is split in half, say $h_{k,1}$ and $h_{k,2}$. The data is then obfuscated by the scaled version of the first half i.e., $O^w = O + \alpha_k \times decimal(h_{k,1})$, while the second half $h_{k,2}$ replaces the original region index $index_O$ that is used to find out the disclosure privilege of that data. Therefore, the obfuscated data can be represented by the quadruplet $< S, P, O^w, h_{k,2} >$. This means that for data de-obfuscation, the associated values including *rid*, α, h_1, and h_2 must be stored.

Table 5.3: Obfuscation Parameter Table

region index	roles	region id	scaling	first half of hash	second half of hash
1	role 1	rid_1	α_1	$h_{1,1}$	$h_{1,2}$
2	role 2	rid_2	α_1	$h_{1,1}$	$h_{1,2}$
.	.	.	.	.	.
k-1	role k-1	rid_{k-1}	α_{k-1}	$h_{k-1,1}$	$h_{k-1,2}$
k	role k	rid_k	α_k	$h_{k,1}$	$h_{k,2}$
k+1	role k+1	rid_{k+1}	α_{k+1}	$h_{k+1,1}$	$h_{k+1,2}$
.	.	.	.	.	.
.	.	.	.	.	.
d	role d	rid_d	α_d	$h_{d,1}$	$h_{d,2}$

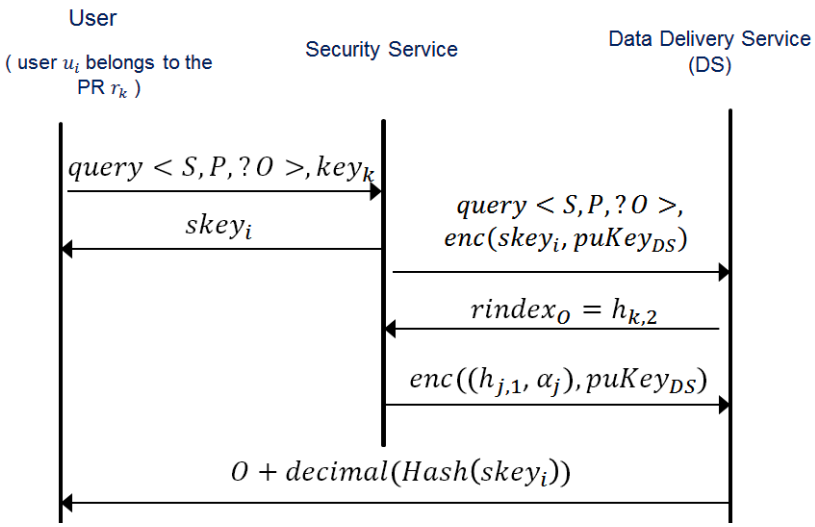

Figure 5.4: Sequence diagram of data delivery

5.1.1.8 Data Delivery

The data delivery service includes a query server that de-obfuscates the data based on the parameters received from the security service. The data delivery service re-obfuscates the data using the session key before sending the result to the user. From a technical standpoint, this not only limits data disclosure at rest, but also while it is being transmitted to the data requester.

Consider a user u_i belonging to the region r_k. The entire process is described step-by-

step as follows:

1. Users u_i sends their query for data object O (i.e., $< S, P, ?O >$), along with its secret (PN code) to the security service,

2. If the key is correct, the security service retrieves the associated rid_k and creates a session key, say $skey_i$, and sends back a copy of the session key to the user. Also another copy of the session key is created using the public key of the data delivery service (i.e., $enc(skey_i, puKey_{DS})$, where $enc()$ is an encryption function) and is sent along with the query to the Data Delivery Service,

3. The data delivery service sends the corresponding $rindex_O$ for the requested Object O to the security service, which is effectively $h_{k,2}$.

4. The security service searches the obfuscation parameter table for the equivalent hash value and retrieves the corresponding $h_{j,1}$, α_j values. These values are again encrypted with the public key of the data delivery service and sent to the data delivery service,

5. The data delivery service consequently decrypts the received information using its private key $prKey_{DS}$ to extract the scaling factor and the watermark and subtracts the multiplication of the two values from the obfuscated data,

6. The data delivery service re-obfuscates the data using the session key before sending it to the user. For this purpose, the hash value of the session key is calculated and its decimal value is added to the original data,

7. Finally, user u_i de-obfuscates the data by subtracting the hash value of the session key and obtains the original data.

The aforementioned steps are illustrated in Figure. 5.4.

5.1.2 Use Case

Wireless sensor networks and cloud computing are currently utilised in IoT applications to deliver smart health care services to citizens [128]. These services require access to sensitive patient data. However, protecting privacy is challenging due to the limited processing capabilities of the IoT devices utilised and the enormous amount of data that needs to be collected and analysed.

The scenario we describe in the next paragraph is an extension of the scenario described in Chapter 3. Consider again the outbreak of an epidemic disease such as Ebola. In order to control the disease, it is necessary to check and monitor the symptoms of all citizens continuously and as quickly as possible. Consider now that in addition to the medical staff who monitor citizens for symptoms, IoT devices (e.g., smart watches, smart phones) are also used to collect symptom data. To provide privacy in such an environment, consider now the four roles that are defined with different access privileges as shown in Figure. 5.2.

The contextual filter will exclude irrelevant data for all queries involving symptoms. For example, if there is no contextual preference for heart rate, people with heart rate-related symptoms will be filtered out and will not be considered. Next, contextual aggregation will generate aggregated nodes as described in [129]; for example, for all queries that are interested in heart rates greater than a particular value and are also interested in blood pressures less than a particular value.

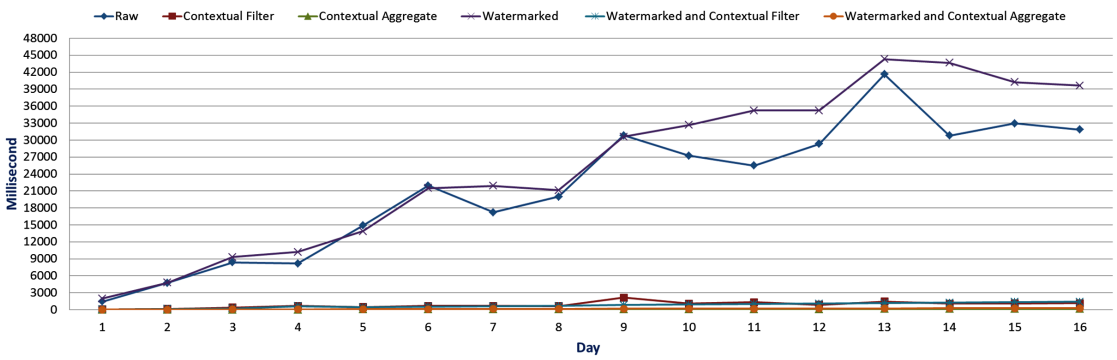

Figure 5.5: Query response times over time.

5.1.2.1 Evaluation Test-bed and Dataset

To evaluate the scalability of the proposed watermarking solution we developed a test-bed. The test-bed was developed on an Amazon EC2, "M4 General Purpose" instance, with 32 GB Random-Access Memory (RAM) and an 8 'vCentral Processing Unit (CPU)'. In the evaluation we utilised an synthesised RDF dataset consisting of IoT data such as the blood pressure, heart rate, and location of 500 users recorded every 10 minutes for 15 days. Data relating to the patients' insurance and citizenship were assumed to be entered into the dataset by medical or administration staff.

5.1.3 Evaluation

Figure. 5.5 shows the results of a performance evaluation conducted using 15 days worth of data collected from patients. The collected data were stored in the form of triplets represented on the horizontal axis of the graph. Some 7200 samples were collected on the first day. Performance of contextual filtering took 66 ms, while contextual aggregation took only 11 ms. Similarly, watermark insertion took 1996 ms, watermarking combined with contextual filtering took 80 ms, and watermarking combined with contextual aggregation took only 18 ms. This experiment clearly reveals just how lightweight the proposed watermarking technique is, as well as the effectiveness of the contextualisation technique and the superiority of their combination (taking only 284 ms to process 1,152,000 data points).

From a security standpoint, the proposed method suffers from two potential issues. First, the usage of an ensemble of PN sequences as authentication keys resolves the problem of generating keys for computationally-limited IoT devices, but it opens up the possibility of a brute force attack for guessing the secret *rid* values. Here, Small Kasami is used which provides $2^{2n} - 1$ possible values for region IDs - a relatively small pool of values. However, this problem can be solved by using a more secure PN codes with a larger set size, such as Moreno-Tirkel sequences [130], without changing the proposed technique.

The second issue is related to the watermark amplitudes for data obfuscation process, i.e., the α values. Here, a constant value is used to amplify the embedded watermark,

which makes the proposed scheme vulnerable to a Wiener attack in which an attacker can remove the watermark by using statistical estimation. In order to combat this type of attack, the power spectrum of watermark must resembles that of the data (referred to as the power-spectrum condition [131]). This feature can be easily added to the proposed model during the contextualisation process and makes the proposed watermarks robust against this type of attack.

5.2 Contextualisation-based Scalable Privacy Preservation for the Internet of Things

In Section 5.1, we proposed a novel solution for contextualisation-based scalable access control. IoT data obfuscation can also be used for privacy preservation and several solutions that use an obfuscation function to reduce the granularity of information currently exist (such as data generalisation and suppression, data masking and perturbation techniques such as random noise addition and data swapping) currently exist. These methods are useful for privacy preservation of a published dataset where the data is distilled based on the trust level of the data consumer, preferably in an irreversible manner to maximise data protection [5]. Applying these obfuscation techniques for IoT data is more difficult and requires an understanding of the privacy requirements of such collected data for the following reasons:

First, in IoT the environments, data is collected from highly distributed resources. This distributed nature of the IoT increases the possibilities for privacy breaches compared to a traditional data store model where the data is stored before it is used or transferred to a third party [132]. In the IoT, even the strongest privacy preservation method may not be effective at the time of data storage. Therefore, it is imperative to protect the privacy of IoT data during IoT data streaming and data collection. The essence of a comprehensive solution to protect the privacy of IoT data through the whole data life-cycle was also addressed recently by Bertino [133].

Second, the management and protection of large volumes of data generated by IoT

devices is a very complex task [132]. The generated data might be used by many different applications for different purposes; even unknown applications and/or without data owner consent. Hence, the data owners need to personally control the disclosure privileges of their data. Traditional authorisation models provide only two options to data owners: granting or denying access. However, having additional disclosure flexibility in addition to granting/denying access may be more appropriate in certain situations. For instance, a taxi driver may be willing to reveal their precise whereabouts to their taxi company (e.g., for logistics and to satisfy company policy). In contrast, the same taxi driver may prefer to send only their cloaked location (e.g., street name) to potential customers (e.g., for safety reasons). Furthermore, the data owners may not only need to be able to control *whom* to share data with, but also *how much* data to share.

Third, the above granular disclosure approach is not possible unless the data owners are provided with sufficient flexibility to specify situations in which they grant/deny access to their data. For example, the taxi driver from the previous example might restrict the taxi company's access to their exact whereabouts while on a break. Similarly, a patient might allow a doctor to access their health record only during an examination. Therefore, with the aid of some extra information such as location and time (for the taxi example) and physical co-location (for the healthcare scenario), granular data disclosure can be achieved. In the IoT literature, such information is called *context* and is often used to make more flexible and intelligent decisions.

The aim of this section is to propose a conceptual framework for privacy preservation of IoT data using contextual information to achieve *flexible privacy*. The general idea behind this research is depicted in Figure 5.6. Assume O is a privacy conscious object that could be either a person (such as a driver or a patient) or a resource (such as a sensor, RFID, etc). The object O has four sensitive data items that are shown here as different shapes (a square, circle, triangle, and pentagon). Depending on the contextual information, the goal is to deliver varying *granularity* to the data requester. Granularity, in this context, refers to the precision of the data which is access-, application-, user- or usage-dependent. This is distinct from precision itself, which is an absolute measure. For this purpose, an

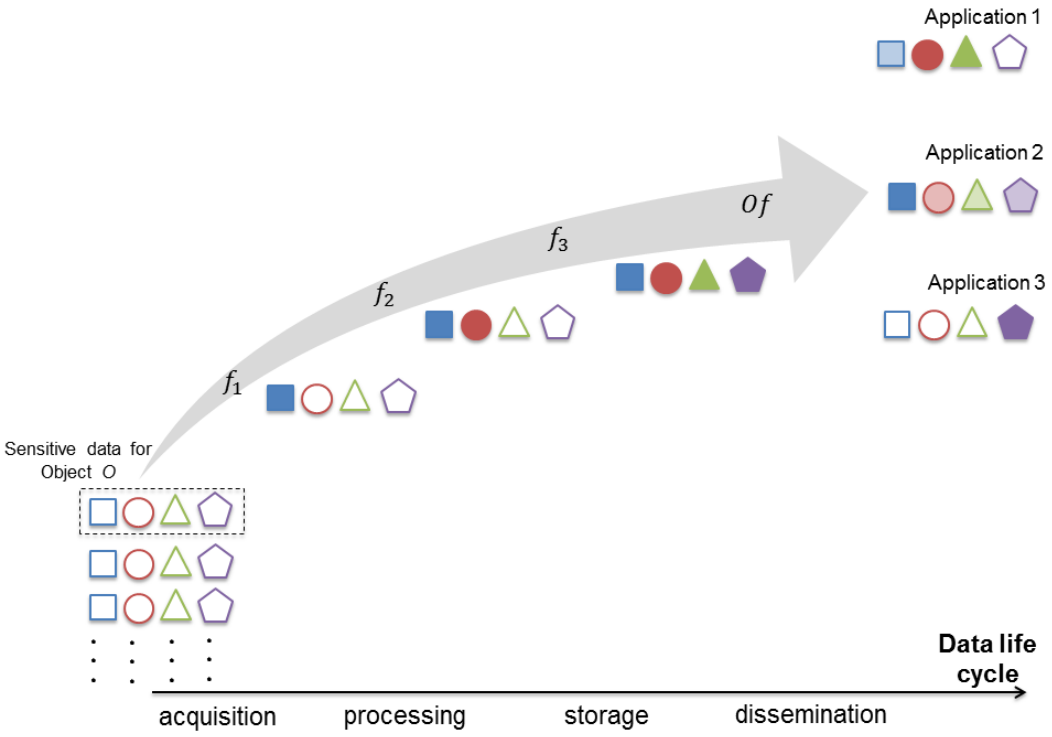

Figure 5.6: Multi-stage privacy protection scheme, where colour intensity represents noise granularity.

obfuscation function is used that returns an obfuscated version of O, i.e., $of : O \rightarrow O^{'}$ depending on the desired granularity level gl. Also, in order to protect the privacy of sensitive data, multiple privacy preservation functions f are applied to the data before data dissemination. One could consider these as sample and spatial-temporal precision variation, respectively. In Section 5.2.1.1, the requirements for these functions are looked at in detail.

Central to the proposed design is the notion of flexible privacy – the data owner should not only be able to control data access but also the accuracy of the data made accessible (to whom, how much). This control can be achieved based on the contextual information of either the data owner (such as location, time, and emergency situations) or data consumer (such as their role, physical co-location, or time of access). As information becomes more contextual, the disclosure granularity that can be achieved becomes finer. Apart from

that, in the suggested framework, the privacy protection is advocated at multiple phases of the data life-cycle to afford maximum data protection (which is different from data transmission security schemes). Previous data obfuscation methods are inferior to the proposed method for IoT settings where dynamic obfuscation is required (i.e., where the context information and contextual preferences change rapidly), but data should always be protected at multiple stages before its actual delivery. In this section, we propose a novel, lightweight, multi-tier privacy scheme suitable for tight-resource-constraint IoT environments, which makes the following contributions:

- A conceptual framework for privacy preservation of IoT data through the whole life-cycle, emphasising end-to-end protection;
- A context-aware granular obfuscation technique for spatial-temporal data; and
- A smart vehicle use case that implements and evaluates this conceptual framework and technique.

5.2.1 Multi-stage Privacy Preservation Framework

In this section, the conceptual privacy preservation framework is described as having two main parts: multi-stage privacy protection and dynamic obfuscation. These are further discussed in the rest of this thesis using the notation provided in Table 5.4.

5.2.1.1 Multi-stage Privacy Protection

To explain multi-privacy protection, recall that in the IoT environment, we aim to protect IoT data before storage and dissemination. In Section 5.2 we defined O as a privacy seeking object with k sensitive data items. To achieve privacy preservation of O, we enrich such sensitive data with a set of *pseudo-sensitive* context (i.e., a context where the privacy of the contextual information does not need to be preserved but the disclosure of sensitive data is affected). To explain this, consider a smart vehicle that periodically reports its vehicle ID and GPS coordinates. Suppose that it also provides some contextual information that includes the current local time and its speed. The vehicle ID and GPS coordinates

Table 5.4: Notation used in the multi-stage privacy preservation framework.

Symbol	Meaning
f	Privacy preservation function
of	Obfuscation function
k	Number of privacy preservation functions
d	Number of granularity level
gl	Granularity level for disclosure control
C^{app}, C^{data}	Application context and data context
CS^{app}, CS^{data}	Application context set and data context set for a given query
n ,m	Sizes of CS^{app} and CS^{data}
cid^{app}, cid^{data}	Atomic Context Identifier of Applications and Data exists in disclosure rules
CID^{app}, CID^{data}	Compound Context Identifier of Application and Data for a given query
Tr_{MO}	Trajectory of a moving object
$< p_i, t_i >$	Position coordinates of the form (x_i, y_i) with time stamp t_i
(α_x, α_y)	Scale factor or watermark amplitude for (x_i, y_i)
$DLFSR$	Dynamic Linear Feedback Shift Register
$(l_1, iv_1, poly_1)$	Number of registers, initial value, and polynomial of the primary LFSR in the DLFSR generator
$(l_2, iv_2, poly_2)$	Number of registers, initial value, and polynomial of the secondary LFSR in the DLFSR generator
π	Secure permutation function
$Hash$	Hash functions, for example SHA-1 or MD5
l, b	Hash output size and buffer length, respectively

are sensitive IoT data, whereas the contextual information (local time and vehicle speed) can always be revealed. Next, consider a case where the driver of the vehicle shares its exact GPS coordinates during the day but not at night. Even though time is not sensitive information here, it is used to determine the disclosure of sensitive data.

Table 5.4 defines k as the number of privacy preservation functions that protect the privacy of k sensitive data items. Each of these privacy preservation functions is denoted by f_i and protects the privacy of the corresponding sensitive data d_i. For our vehicle example, k was 2 (vehicle ID, and GPS coordinates) and therefore two functions are needed. The main requirements for such functions include:

- Must guarantee the privacy of the protected data,
- Must be reversible, in a sense that the original data can be recreated from obfuscated data during run time access, and
- Must be lightweight enough to meet IoT device constraints

The first requirement necessitates the existence of some *trapdoor* information held by a legitimate entity, such as a private key, or seed value. The second requirement ensures reversible privacy which is actually needed to achieve the dynamic obfuscation goal. Therefore, techniques such as generalisation and suppression or Gaussian random noise additions are not acceptable as they transform data into another form in an irreversible manner. Finally, any functions f relying on heavyweight cryptographic mechanisms such as homomorphic encryption [134] to obtain privacy guarantees are too demanding for IoT devices with tight resource constraints.

Please also note that the privacy preservation functions can be applied at different stages of the data life-cycle based on the security requirements. For example, in Figure 5.6, f_1 occurs at the collection point, while f_2 and f_3 are applied at data dissemination stages. Ideally, a successful security solution should provide end-to-end data protection, i.e., from data acquisition to the final destination. However, the real-time processing and computational capacity constraints of IoT devices make the end-to-end protection an ambitious goal. Therefore, a trade-off must be made to provide *sufficient security* in IoT. Some existing work, e.g., [133], recommends privacy protection at data collection and data dissemination stages.

Based on the stage at which the privacy function is applied, one can decide about the security aspect of the function. For instance, if function f resides on sensors (data collection stage), it should have low computational complexity. In contrast, if the protection is applied by a powerful computer at the storage stage, a more complex function is acceptable. Apart from stage considerations, the '3V' features of big data, namely, volume, variety and velocity, should be accommodated in the design of privacy functions. For instance, if the variety of sensitive data is low (such as vehicle ID), an RSA with a 256-bit key length can be used for such data. In cases where multiple sensitive data must be protected, even an encryption method with a relatively small key and cipher-text will decrease the efficiency of any IoT application (in terms of query response time).

For the rest of this chapter, the terms *data* and *context* are used to mean only the *sensitive* data and *pseudo-sensitive* context respectively, unless otherwise specified.

5.2.1.2 Dynamic Obfuscation

In the proposed multi-stage privacy preservation framework, an obfuscation method is used that offers coarser or finer granularity disclosure based on the contextual information considered in each query. The rationale behind this solution is that data that is obfuscated at stages earlier than the dissemination stage may require a static disclosure control model, which may not be suitable for IoT environments where a wide range of applications uses the data. Even when considering a single application, the disclosure granularity might be different for different users based on their contextual information, such as their role, location or time.

Before describing the proposed dynamic obfuscation method it is necessary to distinguish between two types of context: IoT data and IoT application context.

Definition 1: The IoT Data Context (C^{data}) refers to the context associated with the collected data (such as time and speed for the smart vehicle example). The IoT Application Context (C^{app}) includes the contextual information in which the queries are issued, such as the role of the data requester (e.g., physician, nurse, police, etc.) or physical co-location (e.g., of a physician with a patient).[2]

The C^{data} and C^{app} information have direct impact on data obfuscation. To this end, an obfuscation function of is introduced, with the goal to provide a varying degree content information (i.e., granularity) to the application (data consumer) based on both C^{data} and C^{app} values. In other words, $of(d)$ is a version of the original data with a less information precision depending on the allowed disclosure rules. This necessitates the existence of disclosure rules to determine the level of data obfuscation, which will be explained later. Following is the description of the challenges in achieving dynamic obfuscation in an IoT setting.

[2]Remember the term Context is used instead of Pseudo-sensitive context. This type of context is privacy-relevant information that affects the disclosure granularities of the data.

5.2.2 IoT Data Challenge

The question that arises here is whether flexible privacy can be accommodated with respect to IoT data characteristics?

Example 1. Assume Alice wants to find the current location of Bob. Some of the following disclosure policies may apply

- If Alice is a paramedic currently located in Melbourne, and Bob has an incident in Melbourne, the precise location of Bob is shared with Alice.
- If Alice is an employee of the Melbourne branch of 13CABS (Taxi Company), and Bob has a trip to Footscray suburb overnight, he only shares his street name with Alice.
- If Alice is an employee for VicRoads (roads authority), and Bob is driving in Heidelberg Road, Chandler Highway, or Malvern Road (streets of Melbourne) during weekdays, he shares his suburb name with Alice only during her working hours.

Given policies such as the above, it is clear that searching every data and application context (C^{data}, C^{app}) for every request is prohibitively expensive.

In the above scenario, one data requester (e.g., Alice) and one thing (e.g., Bob's GPS device) are considered. In IoT, millions of things are connected to the internet, and will generate big data at unprecedented scale. In such an IoT ecosystem, only a subset of things are typically queried by any application at the same time. Furthermore, contextual information, such as time and location, can be frequently updated and therefore the response latency can be extremely high.

To address the above two issues and making the proposed obfuscation scheme scalable, two aspects are considered in the design of our framework. First, making privacy flexible via policies, or so called *disclosure rules*, to dramatically reduce the need to search all individual contexts. In other words, once a query is issued and the existing rules are scanned to find a match, if a match is found, the data is obfuscated based on the stated granular rules; otherwise, the data is not revealed at all. This way the complexity is relative

to the number of existing rules, not the number of all possible combination of multiple contexts (i.e both $C^{data}$and C^{app}). Second, a rule indexing model is proposed to further speed up finding the corresponding rules. This model is based on prime factorisation to scales up the framework described in Chapter 3.

5.2.3 Disclosure Rules

In the proposed framework, a disclosure rule has two main parts in the form of *conditions* $\rightarrow$ gl. The conditions are the contextual information, possibly for both the data and the application. The gl is the disclosure granularity, the value of which depends on the characteristic of data. For instance, for GPS coordinates, location precision can be expressed in terms of logical locations such as the building name, street name, suburb name, city, etc. Also, one may decide to control discourse in terms of data precision by considering metrics such as metres, kilometres, feet, or other units [135], or apply precision rules to other numeric data, such as date/time, age, height, etc. This framework does not put any limitation for expressing gl values. d is considered as possible granularity levels $gl_1, gl_2, ..., gl_d$ for a particular data.

Additionally, a consensus on defining the disclosure rules is needed. Generally, regular expressions can facilitate rule representations to support even complex rules. There are also more specific options such as Platform for privacy preferences [136] or Semantic Web Rule Language (SWRL) [137] that can be used. In this section, SWRL is used to express disclosure rules, but other languages can be used interchangeably. For instance, the first rule of Example 1 can be expressed using SWRL as: (paramedic(?requester) $\wedge$ hasCurrentLocation(?requester, ?Melbourne) $\wedge$ hasIncident("Bob") $\rightarrow$ shareLocation (?maximumPrecison).

These rules need to be defined by security managers or whoever is in charge of preserving privacy of things (in association with people). Once the disclosure rules are defined, the next step is indexing those rules as explained below.

5.2.4 Rule Indexing Model

At this step, for every context that is present in the existing disclosure rules, an identifier is assigned which is the next available prime number. The identifiers for application contexts and data contexts are shown with cid^{app} and cid^{data}, respectively. For the first rule of Example 1, if we assume Bob's vehicle ID is 'car1234', then we will have $< Alice, Role, Paramedic >$, ${cid_1}^{\text{app}} = 5$ and $< Alice, Location, Melbourne >$, ${cid_2}^{\text{app}} = 11$ and for the data identifiers, we have $< car1234, Location, Melbourne >$, ${cid_1}^{\text{data}} = 7$ and $< car1234, incident, yes >$, ${cid_1}^{\text{data}} = 11$. It is important to note that cid^{app} is not unique for each particular application and can dynamically change based on the applications and their changing contextual preferences. The same is true for the data context identifiers. These assigned identifiers are then stored in two separate tables, namely ACI and Data Context Identifier (DCI) tables.

The next step is rule indexing, the idea of which is borrowed from [129, 138]. First, the atomic and compound context identifiers are defined as follows.

Definition 2: An atomic context identifier is a prime number that is assigned to every present context in the disclosure rules. A Compound Context identifier is constructed by multiplying several context identifiers. In fact, the described cid^{app} and cid^{data} are atomic. If we form $cid_1^{app} \times cid_2^{app} = 5 \times 11$ for example 1, the 55 is a called a compound context identifier. This type of context is denoted by capital letters, i.e. CID^{app} and CID^{data}.

The rule indexing method essentially translates a disclosure rule into two compound identifiers. In other words, the rule indexes are calculated from atomic identifiers to compound identifiers, then compound IDs for data and application are associated with each other according to the rules. For instance, the first rule of Example 1 will be mapped to $(CID^{app} = 55, CID^{data} = 77, gl = full)$. These rules are stored in a table which is referred to as a Mapped Rule Index (MRI) table. Once a query is issued, the corresponding context identifiers are found and multiplied together and then searched through the MRI table. This way, the rules can be stored in a more compact way and the query efficiency will therefore be improved.

Next, the main components of the proposed architecture are reviewed.

5.2.4.1 Security Service

In the proposed framework, the dynamic obfuscation is done by a Security Service that follows the principles of the Security-as-a-Service model [139]. However, the goal is to achieve *content* security as opposed to *transport* security to govern disclosure control. For a convincing explanation, several tasks of this service have decoupled into three main components of Application Context Engine (ACE), Data Context Engine, and Disclosure Decision Point. However, this does not mean there are three different components; in fact, they are all part of one entity, i.e., the security service.

Application Context Engine

The main task of the ACE is finding contextual information of the incoming data request. Upon receiving a query, ACE first authenticates the data requester. If the authentication is successful, then the next step is forming the context set, i.e., CS^{app} and CS^{data}. The ACE delegates the formation of the latter set to the Data Context Engine (DCE) and is itself responsible for forming the CS^{app}.

At this step, it can be assumed that the contextual information for the application is either stated in the query (in case the framework utilises contextual queries [Chapter 3], or the ACE extracts them. For instance, the provided credential could reveal the role of the data requester or the IP address can be used to find the location of the requester. Apart from that, some of the contextual information such as time might need to be translated to other high level context (such as "working hours" or "night") and therefore the application context engine should interpret this context.

Data Context Engine

DCE has two main tasks: forming the context set for the queried data, i.e., (CS^{data}), and obfuscating the data before its dissemination to the data requester.

For both tasks, the DCE might need to decode some of the information (either context or data) using the relevant privacy preservation function. A prior example was the use of the multi-stage privacy protection that is achieved by applying k different functions to

protect sensitive information. For instance, if a physician wants to have access to heartbeat data of patients who have a family history with cardiovascular disease, this information might be stored in the patient's health-record that is already encrypted (say by function f_1). Therefore, the DCE needs to reverse the transformation process prior to retrieving contextual information. Additionally, once the granularity rule for the data requester is found, the data itself might be protected in a database, if it has been defined as sensitive data (say by function f_2 for heartbeat data). Therefore, the data is first decoded and then obfuscated prior to its dissemination to the destination; this is emphasised on the low complexity and reversible privacy requirements of function f.

Disclosure Decision Point

Disclosure Decision Point (DDP) is where the tables ACI, DCI, and MRI have been stored. Once the CS^{app} and CS^{data} are obtained from the application and data context engine, the DDP translates each individual context to the prime identifier using table ACI for application context and DCI for data context. Then, the compound identifiers CID^{app} and CID^{data} are calculated by multiplying the relevant identifiers. Finally, the MRI table is scanned for a match. If a match is found, the corresponding granularity level (gl) is given to the data context Engine that obfuscates the data accordingly, and is then sent back to the data requester. Otherwise, the access disclosure is denied.

5.2.5 Case Study: Smart Vehicles

In this section, the proposed framework is customised for a smart city scenario. Smart cities rely on advanced technologies, such as the IoT, networking, data analytics, recommendations and decision support to deliver a better quality of life to citizens [129, 140]. The main building blocks of a smart city are smart healthcare, smart vehicles, smart grids, and so on. Although a general-purpose framework is proposed, for this case study we focus on smart vehicle deployment. Such connected vehicles are already on the market and it is estimated that by 2020, 75% of all cars shipped will be built with internet-connection hardware [141], which obviously raises privacy issues. The reason why smart vehicles were

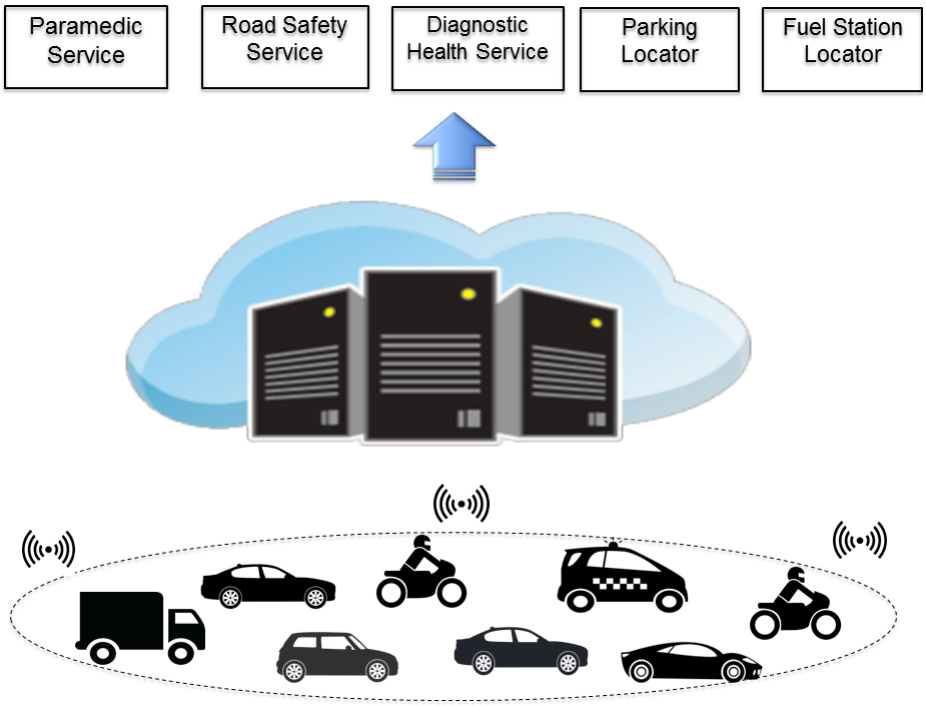

Figure 5.7: An overview of a smart vehicle system.

chosen as a focal point is that the types of data they generate include spatio-temporal streams that are changing frequently (i.e., trajectory data) and, therefore, context-aware privacy preservation can become a difficult task.

5.2.5.1 System Overview

The proposed smart vehicle system is shown in Figure 5.7. At the bottom of this figure, there are smart vehicles that, with the aid of sensors and RFIDs, periodically transmit their data to cloud storage. Five main services/applications are considered in the system: paramedics, road safety, a parking locator, a fuel station locator and a diagnostic health service. The paramedic service is an emergency service that is available to smart vehicles in the case of accidents. The road safety service provides traffic information such as road congestion, recommended paths and driving offences. The parking locator and fuel station locator are essentially location-based services for finding available parking spaces and nearby fuel stations, respectively. Finally, the diagnostic health service gathers information

about people who might be in the presence of contagious diseases. In Section 5.2.5.3, these services are related to trajectory data.

5.2.5.2 Data Model

A smart vehicle is a moving object that is equipped with one or more sensors. In the proposed data model, streaming trajectories obtained by these sensors are treated as sensitive information that need to be protected. For a moving object MO, a trajectory data stream is presented Tr_{MO} as a sequence of pairs $Tr_{MO} = \{< p_1, t_1 >, < p_2, t_2 >, ..., < p_{now}, t_{now} >\}$, where position p_i is a Cartesian point coordinates shown as x_i and y_i with ordered timestamps t_i. The (x_i, y_i) values could be easily obtained by mapping GPS coordinates i.e. longitude and latitude using a Universal Transverse Mercator (UTM) transformation.

Apart from the trajectory stream, every individual data point is enriched with a set of contextual information that is denoted by CS^{data}. For the particular dataset that is used for the implementation, the context set includes vehicle ID (vid), and current speed sp_i. Therefore, the information for the object MO at time t_i includes $(< p_i, t_i >, vid, sp_i)$. As the dynamic data obfuscation approach depends on contextual information of queried data and data requester, it is also needed to define the application context (CS^{app})for the system as described in the following section.

5.2.5.3 Spatio-temporal Granularities and Disclosure Rules

As discussed in Section 5.2.1.2, there are several ways to define granularities. Without loss of generality, for spatial granularity ($gl^{spatial}$), location precision (in terms of kilometres, meters, feet, etc) is considered and for the temporal granularity ($gl^{temporal}$), a binary granularity meaning the time information should be revealed or not is considered.

To clarify, let's review some of the discourse rules for different services. If a vehicle involves an incident, the precise location and time is shared with the Paramedic service. The Diagnostic Health service only has access to the spatial data with granularity of 1m, i.e ($gl^{spatial} = 1m$ and $gl^{temporal} = 0$). This disclosure rule for the proposed case study has been motivated by a scenario where a driver is suspected for a super-contagious diseases

such as Ebola, and therefore the Diagnostic Health needs to know the places that the driver has been visited in order to stop the spread of the disease. However, the order of visiting places does not matter to this service and thus does not need to be revealed. For a general scenario, having d granularity levels, the $(d-1)$ least significant bits of location coordinates to zero for the worst case (applications with the least discourse granularity) is set in order to mask the precise coordinates.

It is important to note that for a particular service, the granularity could change based on the contextual information. For instance, the road safety service might be authorised to have access the location information with ($gl^{spatial} = 100 feet$ and $gl^{temporal}$=0), but if the vehicle is travelling beyond the street limit, the exact location and time might be revealed to this service.

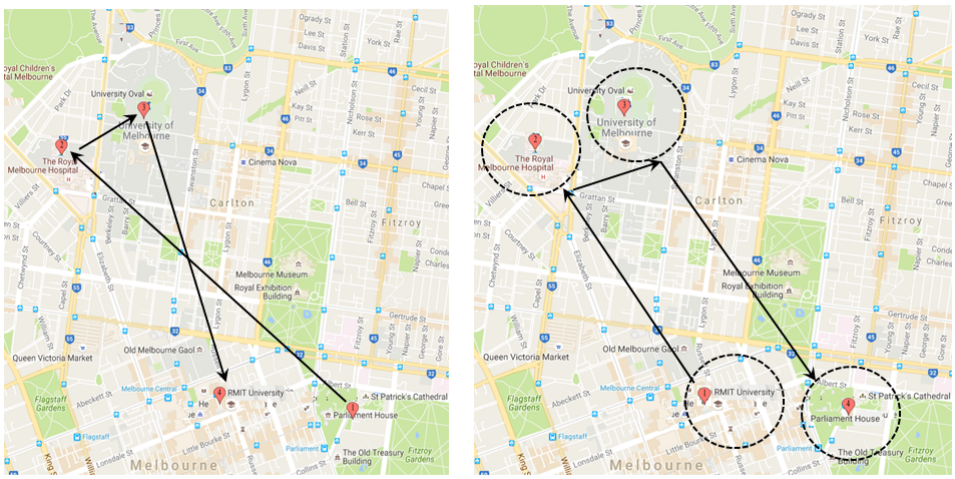

Figure 5.8: Spatio-temporal privacy preservation. (a) Original trajectory: Parliament House → Royal Melbourne Hospital → University of Melbourne → RMIT University, (b) modified trajectory: *cloaked*(RMIT University) → *cloaked*(Royal Melbourne Hospital) → *cloaked*(University of Melbourne) → *cloaked*(Parliament House).

.

5.2.5.4 Privacy Protection at a Glance

The trajectory stream is considered as sensitive data, the privacy of which needs to be protected. Therefore, there are two types of data in this example, spatial and temporal. In this regard, the system resembles the privacy protection against location-based services. Duckham [142] proposed a few rules as the key principles of research on location privacy, which make it different from other privacy preserving research. The author suggested to consider the predictable mobility of humans, the constraints of the area within which people move, and the importance of a formal definition of fundamental terms (such as the precision and accuracy of information) in the design of protection mechanisms. The majority of proposed methods for spatio-temporal privacy only focuses on location privacy by means of techniques such as randomising, discretising, sub-sampling, etc. However, revealing timeliness of spatial data opens up the possibility of time-and-location attack [143] and results in breach of privacy. There are only a few privacy methods that do not disregard the temporal cloaking of trajectory data.

In the proposed system, a two-tier privacy preservation (k=2) is proposed, one for privacy preservation of spatial data at the data collection stage, and the other for temporal data that is applied at data storage stage. Therefore, it can achieve spatio-temporal privacy while data is at rest. Additionally, an obfuscation function of is used that obfuscates data according to the desired granularity level at the time of data dissemination. Figure 5.8 illustrates an example of the spatio-temporal privacy preservation approach, where the original trajectory of a moving object is not only replaced with the cloaked locations, but also the sequence of these locations are perturbed.

In order to respect the low complexity of IoT devices, both suggested privacy preserving functions f_1 and f_2 and also the of function are lightweight, while attempting to make them as secure as possible, given IoT device constraints. For this purpose, digital watermarking methods and pseudo-random constructions are used because of their hardware-friendly nature. In fact, f_1, f_2, and of are pseudo-noise addition, (hashed-based) scrambling, and data masking (then can be coupled with one-time pad partial encryption, if a more secure transmission is needed), respectively.

5.2.5.5 Preliminaries

In this section, some preliminaries are briefly introduced to help understand the rest of Section 5.2.5.

Digital Watermarking

As we discussed earlier digital watermarking is a technique for copyright protection. The watermark constitutes a piece of secret information to be hidden within the digital content in such a way that it is not visible to the consumer. A digital watermark can be either distortion-based or distortion-free depending on whether the embedded marks introduce any distortion to the underlying data [144]. For example, adding random numbers to data samples results in changing the original values (distortion watermarks), whereas re-arranging data samples according to a secret watermark do not introduce any change into data values (distortion-free watermarks).

One of the recent driving forces in digital watermarking research is data obfuscation [145]. Because embedding watermarks introduces a tunable distortion in host data, it is possible to mask the original data for the applications where privacy is of great concern. Contrary to conventional watermarking, the visibility constraint can be relaxed; the reversible distortion introduced by the watermark is used to reduce data precision to below levels where privacy can be compromised. These levels are tunable to application-dependent granularity.

In this section, a distortion-based watermark (noise addition) and a distortion-free watermark (data scrambling) are used. For the former, taking advantage of a so-called *Linear feedback shift registers* to construct the noise/watermark will be described briefly in the following.

Linear Feedback Shift Registers

Generating random numbers has been studied thoroughly for many applications such as stream cipher design, watermarking codes, spread spectrum communications [146]. Unfortunately, generating Truly Random Number (TRN)s is an expensive task due the

complexity of required hardware (such as thermal noise of zener diodes or radioactive decay). In this regard, Linear Feedback Shift Register (LFSR) are favourite primitives due to their desirable statistical properties and hardware-friendly nature [146].

LFSRs are shift registers, generating new bits using a linear feedback polynomial. Figure 5.9 shows an example of an LFSR of order 4. In this example, a new bit is generated for each shift, based on a linear combination of the bit values of (in this case) 3 and 4 previous shifts. Certain feedback combinations produce a pseudo-random pattern of bits equal to 2^{order}-1. So in the case of Figure 5.9, a pattern of maximal period of 15 is produced (16-1 as the all-zeros case is excluded). For this reason, the sequence thus produced is called a *maximal-* or *m*-sequence.

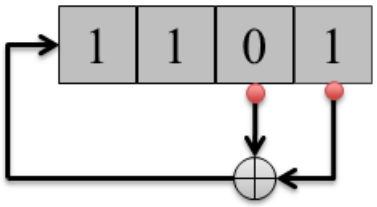

Figure 5.9: An LFSR of order 4 with characteristic polynomial $x^4 + x^3 + 1$. The red points show the positions of the taps.

An LFSR can be completely specified by means of its *characteristic polynomial* or the positions of the *taps*. For the previous figure, the characteristic polynomial is $g(x) = x^4 + x^3 + 1$. Once the $g(x)$ and non-zero initial value of registers (1101 for Figure 5.9) are known, the rest of the sequence is uniquely identified. In fact, for this purpose, the deterministic behaviour is desirable because of the reversibility requirement of the privacy preserving operation (function f). On the other hand, the security requirement of the privacy functions calls for unpredictability of PN sequences that is measured by *linear complexity*.

Linear Complexity (LC) is the length of the shortest LFSR that is able to generate a given sequence. An ideal binary PN sequence of length p, is one whose linear complexity is also p. In other words, the entire sequence is needed in order to predict future bit values.

For very long sequences, this is impractical, and thus is sufficiently secure. Unfortunately, the LC of an LFSR itself is poor ($\log_2 p$). In the literature, there are many proposals to generate PN sequences with higher LC such as: adding a source of a truly random number generator, combining multiple LFSR, or decimating generators irregularly. For this section, a method called Dynamic Linear Feedback Shift Register (DLFSR) is used that achieves high LC by frequently changing initial values and characteristic polynomials.

5.2.5.6 Privacy Preserving Data Collection-Spatial Cloaking

The spatial privacy in this section is achieved by adding PN values (watermarks) to the location coordinates at the sensors.

Watermark Generation using DLFSR

The binary DLFSR [147] is used that consists of two LFSRs (primary and secondary) and a counter. The primary LFSR is controlled by the counter, whose value depends on the internal state of the secondary LFSR and therefore the primary LFSR polynomial is changed in a round robin fashion. In other words, a secondary LFSR is used that cloaks regularly, combined with a primary LFSR with irregular cloaking.

The above DLFSR needs three values for both LFSRs in order to generate random numbers: number of LFSR stages (LFSR order), initial value of the register, and the initial selected polynomial. These values are denoted by $(l_1, iv_1, poly_1)$ and $(l_2, iv_2, irrpoly_2)$ for the primary and secondary LFSRs, respectively.

Note that while the examples explained below assume a binary LFSR (ie. alphabet of 2) for simplicity, LFSRs using a larger alphabet can also be considered, A, which is a prime number, and then balanced by subtracting the floor $(A/2)$. The DLFSR construction works identically provided both LFSRs use the same alphabet. The sequence values are then uniformly distributed from $-A/2$ to $+A/2$.

Watermark Insertion

Assume the i^{th} generated random number obtained from the DLFSR is r_i. The watermark is then a scaled version of the r_i that is added to the location coordinates to make it inaccurate. For the location $p_i = (x_i, y_i)$, the watermarked location $p_i^{'}$ is calculated as follows:

$p_i^{'} = (x_i^{'}, y_i^{'}) = (x_i + \alpha_x \times r_i, y_i + \alpha_y \times r_i)$

where α_x and α_y are scaling factors or watermark amplitudes.

Clearly, the higher the scale factors, the better the spatial privacy can be achieved as the accuracy of the data decreases. This way, trajectories can be hidden on the fly at the point of origin. The Security Service needs to be able to obtain the original values if necessary. Therefore, for every moving object MO, 8 secret parameters need to be exchanged *a-priori* with the Security Service including $\{(l_1, iv_1, poly_1), (l_2, iv_2, poly_2), (\alpha_x, \alpha_y)\}$.

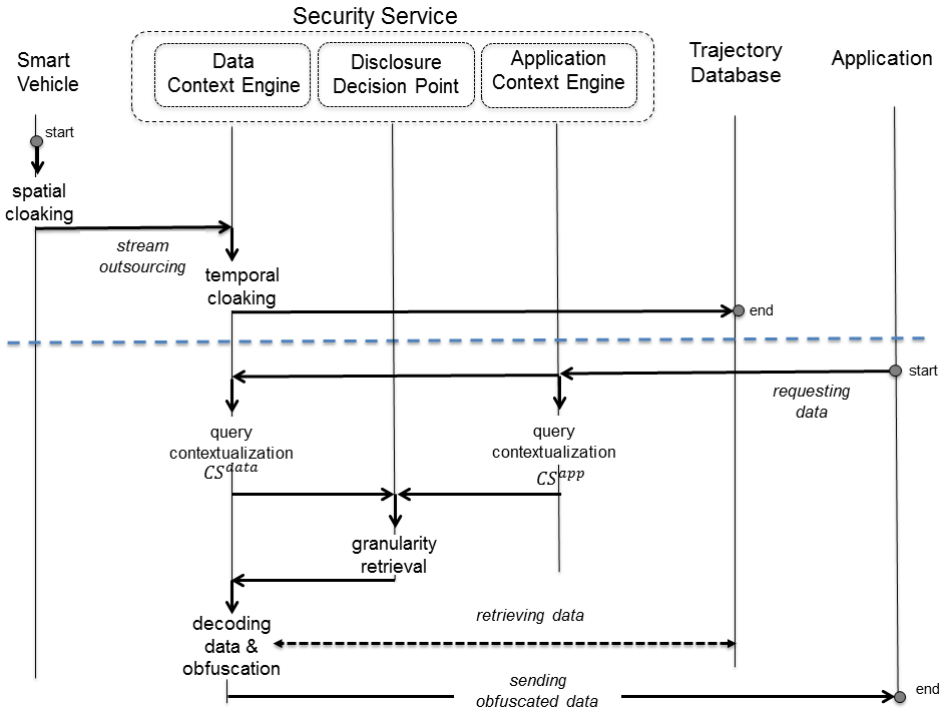

Figure 5.10: Sequence diagram for spatio-temporal privacy preservation. The dashed line separates the data collection and storage stages from the data dissemination stage (as explained in Section 5.3.4.1, these three components are part of the one security service and are only de-coupled here for demonstration purposes).

5.2.5.7 Privacy Preserving Data Storage-Temporal Cloaking

Temporal privacy can be achieved by scrambling the obfuscated locations. Note that the temporal cloaking does not change the timestamps. Instead, the coordinates are shuffled positionally to obscure the timeliness of location information. For instance, if vehicle "car1234" has been at University of Melbourne at time 5pm, and then RMIT University at time 8pm, the locations are swapped, but the time values themselves are not changed. At the time of data dissemination, the Security Service decides whether the correct order of trajectories needs to be revealed to the application.

In contrast to the spatial cloaking, the temporal cloaking is done by the Security Service. This means there is more freedom to choose a secure privacy preservation method (compared to the LFSR) as long as the time complexity is not high. For this purpose, a secure permutation is described as $\pi : [1..b] \rightarrow [1..b]$ to scramble data in a particular order such that the i^{th} watermarked point $p_i^{'}$ is substituted by $p_{\pi(i)}^{'}$ and b is the required buffer size (that is a power of 2). For example, a trajectory of length 4, $\left\{< p_1^{'}, t_1 >, < p_2^{'}, t_2 >, < p_3^{'}, t_3 >, < p_4^{'}, t_4 >\right\}$, is replaced with $\left\{< p_4^{'}, t_1 >, < p_1^{'}, t_2 >, < p_2^{'}, t_3 >, < p_3^{'}, t_4 >\right\}$ for a certain permutation π.

There are many different ways for having a secure permutation such as using block cipher, hash functions, or congruential random numbers, to name a few. Here, a simple yet effective secure scrambling method is chosen based on hash functions that is described below:

Assume $Hash()$ is a one-way keyed hash function with the output size of l bits and a buffer size of $b = 2^c$. A non-overlapping window of size c is applied to the hash output and encodes it to a decimal value. The window by c values are then advanced. This leads to l/c numbers that correspond to the permutation indices. However, it is possible that some of the numbers collide. In such circumstances, the duplicate values are skipped. The beauty of this scheme is that hashed value can be customised for individual moving objects by using a secret key or its combination with other contextual information such as vehicle id. The security of this permutation lies in the security of the hash function and the buffer size b. The longer the buffer, the higher the number of possible permutations there are,

meaning a brute force attack will be less effective. On the contrary, very long buffer results delay the reverse process and decrease the system efficiency in terms of query response time.

By coupling temporal scrambling with spatial cloaking, it is ensured that data is protected at rest.

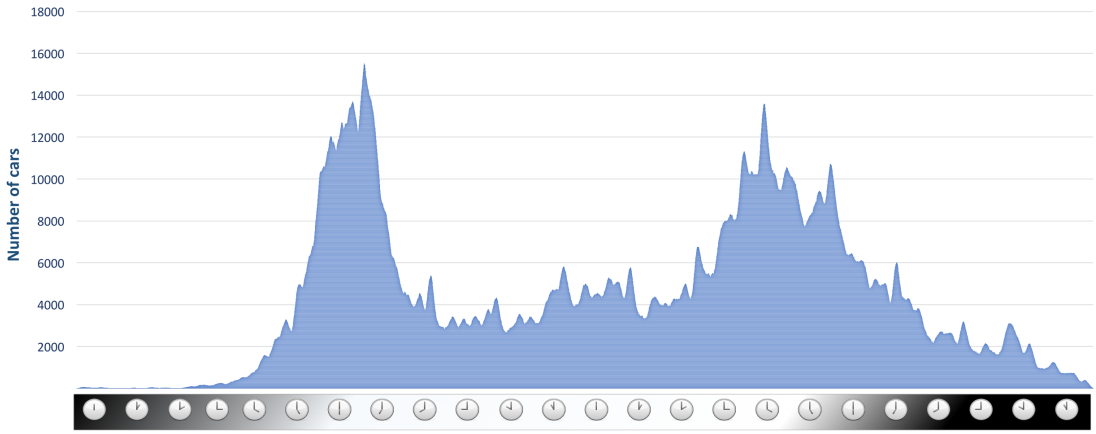

Figure 5.11: Number of vehicles per time interval (x-axis) over 24 hours (Dataset from [8]).

5.2.5.8 Privacy Preserving Data Dissemination-Dynamic Obfuscation

The data dissemination stage is triggered by receiving a query. At this point, it is assumed that the query is contextualised and therefore the corresponding granularities i.e $gl^{spatial}$ and $gl^{temporal}$ are retrieved from the Disclosure Decision Point. Before obfuscating data based on these two values, the Security Service first needs to obtain the exact location and time and then based on the $gl^{spatial}$, the least significant of the spatial points are masked. If the query is requesting the trajectory of an object for a time duration such as range queries (the example uses a range of times between 8am-12am), the Security Service decides whether the time information of the trajectories should be revealed based on the $gl^{temporal}$.

For obtaining original information, the Security Service uses the secret values to regenerate the random numbers for reversing scrambling and noise addition. For the

former, the queried data point needs to be retrieved from the permuted index $p'_{\pi(i)}$. Then, for the latter, the watermark values ($\alpha_x \times r_i$ and $\alpha_y \times r_i$) need to be subtracted from $p'_{\pi(i)}$. The last step is obfuscating the original coordinates by masking the least significant bits.

For implementation, a maximum precision of 1 meter is considered well below the precision of GPS coordinates, and at the same time sufficient for many applications. Then, consider 4 different granularity levels, such that for the lowest level, the three Least Significant Bits (LSB)s are set to zero, whereas for the highest level no bit will be masked. Geocentrically, the obfuscated location $of(p_i)$ can be represented by a circle with p_i at centre, and $2^{gl^{spatial}}$ is the radius in meters. The more undefined bits, the larger the circle area, and thus better privacy can be achieved while the utility of data might also becomes a concern. The entire privacy preservation process that has described so far is illustrated in Figure 5.10.

It is mentioned that the Security Service mainly deals with protection of data content as opposed to the data during transmission. If the data transmission is also required to be protected, it is recommended that partial encoding is used, such as a one-time pad with a user session key to protect the unmasked bits (i.e. most significant bits) of the location data to enforce transmission security as well. This way, there is no intermediate point from the point of origin to the final destination where data is left unprotected. For sensors with the capability, the above can be implemented using connections secured by SSL/TLS.

5.2.6 Performance Evaluation

The described smart vehicle system is developed as a proof of concept. For this purpose, a large-scale urban vehicular mobility dataset [8] is used which contains trajectories of the car traffic in the city of Cologne. The trajectories covers a region of 400 square kilometres for a period of 24 hours. The dataset comprises more than 700.000 individual car trips. Each record of this dataset contains the time (with 1-second granularity), the vehicle identifier, its position and speed. Figure 5.11 shows the distribution of vehicles over the 24 hours. Also, 5000 disclosure policies are defined as described in Section 5.2.1.2. All the experiments are performed on a workstation with a Intel i7-4790 3.26GHz CPU and a 8GB

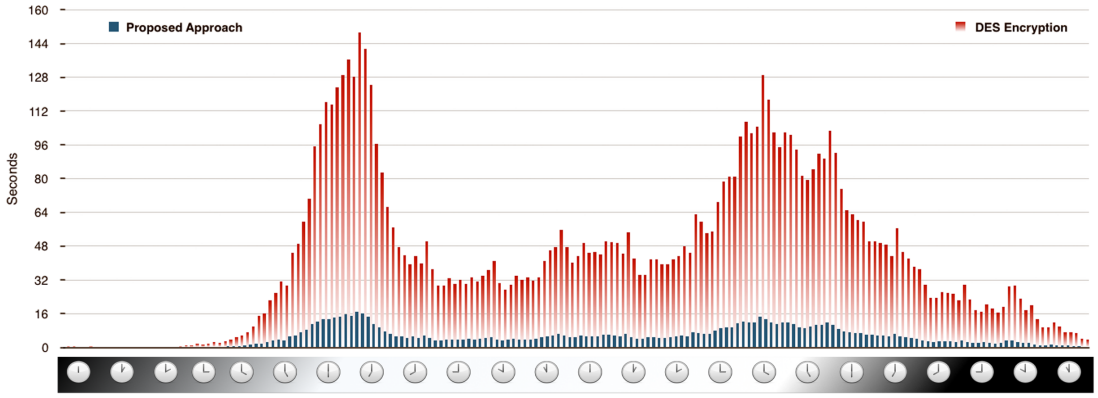

Figure 5.12: Comparison of the processing times of the proposed system (using DLFSR and SHA1-based permutation) and the DES-encryption baseline system.

RAM.

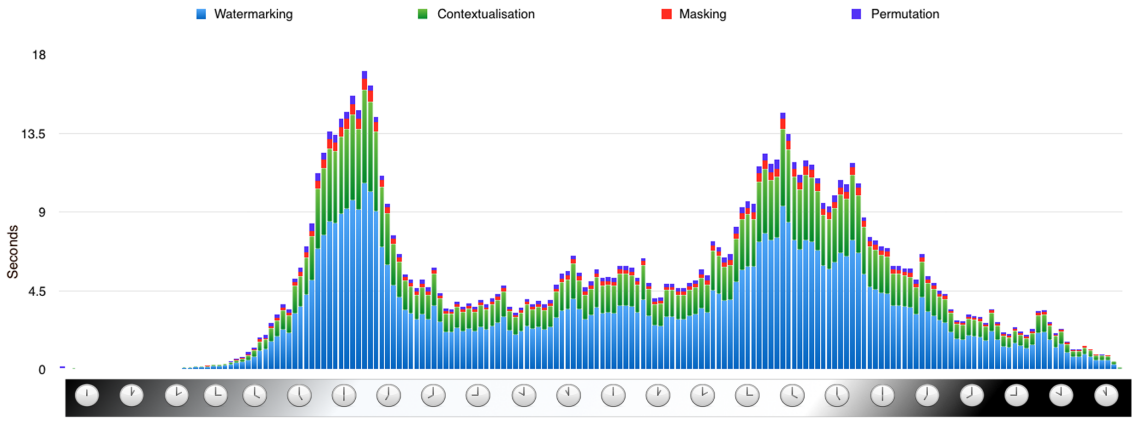

Figure 5.13: Running time breakdowns for different parts of the proposed system including query contextualisation, finding permuted indexes, extracting watermarks, and dynamic data masking.

The performance of the proposed system is investigated in terms of processing time for retrieving trajectories. For the system, this includes the time that is required for reversing the privacy transformations (permutation and watermark extraction) and dynamic obfuscation (data masking). For simulating queries, the 24 hours are divided into time intervals of size 7.5 minutes, and each query retrieved the trajectory of vehicles that are present in that particular interval for different services that are named in Section 5.2.5.1.

Additionally, the proposed method is compared against a baseline system that encrypts the trajectories with Data Encryption Standard (DES) using a 56-bit key and generates 64-bit ciphertexts. In other words, instead of spatio-temporal cloaking, GPS coordinates are encrypted to measure the security overhead for protecting data at the database. Therefore, every time data leaves the system, it needs to be decrypted to retrieve original values and then masked with the desired granularity level that completes the dynamic obfuscation goal.

The results are shown in Figure 5.12. It is interesting that the processing time of both systems follows the data distribution behaviour. For instance, during the peak hours (such as morning and afternoon), the processing time of retrieving trajectories for both systems is higher as there are more cars in those intervals. As you can see, the processing time for the proposed method is much better (by approximately 10 times) than the encryption system. Apart from that, the data cannot be protected by encryption methods at the sensor level due to high complexity of the operations and therefore the data is protected only at the database. Therefore, the proposed system significantly outperforms its encryption counterpart in terms of both response time and data protection, and at sensor level and database as well.

To better understand the security overhead of different parts of the proposed scheme, the running time that is taken by the Security Service for query contextualisation, reversing temporal cloaking (permutation), reversing the spatial cloaking (watermarks), and dynamic obfuscation (data masking) are further explained in Figure 5.13. According to this figure, the most time consuming part is related to permutation and query contextualisation, though this is still an acceptable cost for having multi-granular obfuscation.

5.2.6.1 Discussion

The above results confirm that the cost of security is reasonable in the proposed framework. This suggests that the proposed method is more efficient and scalable than the encryption alternative for an IoT setting. In this section, the synchrony between the watermark encoder (i.e., sensors) and decoder (security service) is considered. In other words, if some

data is missing during data transmission to the cloud, or if the sensors stop working for a while, it is possible that the decoding process begins to fail. One possible solution to this problem is to use synchronisation invariant watermarks, such as the twin watermarks proposed in [148].

From a security standpoint, two parts influence the proposed privacy preservation: spatial and temporal cloaking. Firstly, the security of spatial cloaking is based on the linear complexity of the DLFSR generator. For the implementation, a DLFSR is used with polynomials of the order 16 and 4, which generates random numbers with period and linear complexity of $p = 7864200$ and $LC = 1920$ [147]. This increases the linear complexity by a factor of almost 120 compared to a simple LFSR –and, this is achieved without losing the good statistical properties of the LFSR[3]. It is also possible to use higher order LFSRs for the primary and secondary polynomials to further increase the security. Because the linear complexity of this DLFSR generator is not mathematically investigated for a general case by the authors [147], this length is chosen to be able to compare it with the basic LFSR. Apart from the random numbers, the scale factors are part of the secret information that is only known to the data owner and the Security Service.

Secondly, the security of temporal cloaking is related to the security of the permutation. For this purpose, a one-way keyed hash function is used to scramble watermarked data points. A good hash function $Hash()$ must have the two properties:

1. One way transformation: Given a hash value h it should be difficult to find any message m such that $h = Hash(m)$;

2. Collision resistance: It should be difficult to find two different messages m_1 and m_2 such that $Hash(m_1) = Hash(m_2)$. Such a pair is called a hash collision.

Due to the one-way property of hash function, even if given $h = Hash(M, K)$ in the temporal cloaking, the attacker is unable to obtain the secret key K from the hash value. In addition, hash functions are often used in the generation of pseudo-random bits (e.g., NIST special publication [149]). The randomness of hash functions guarantees the

[3]The period and linear complexity of an LFSR with order 16, is 65535 and $\log_2 2^{16} = 16$, respectively as described in Section 5.2.5.5.

permutations for scramble temporal cloaking is unpredictable without knowledge of the secret key K. Additionally, the uniformity of the distribution of hash values guarantees that all valid permutations can be generated on the basis of hash values.

CHAPTER 6

Conclusion

In this thesis, we have introduced a novel framework and corresponding architecture for IoT data contextualisation that supports more efficient IoT data processing, security and privacy.

These contributions provide scalable contextualisation in real-time and include rigorous definitions of IoT contextualisation concepts, IoT contextualisation operators that utilise prime factorisation, and a use-case that demonstrates the practical benefits of contextualisation at the IoT application level.

The IoT contextualisation architecture was then applied to IoT scale data processing, which illustrated the scalability achieved via IoT contextualisation in a smart parking use-case and a related performance benchmark. More specifically, the experimental evaluation conducted using the parking recommender use-case demonstrates that contextualisation of IoT data can reduce query times for IoT services by more than three times that of a scenario where no contextualisation is applied to the same query workload.

Another contribution of this thesis is the role-based disclosure control technique, which ensures data security in any IoT application where the dissemination of IoT data may violate the privacy of its users (who submit queries or/and provide IoT devices for use by others). To develop a security technique, we combined our IoT contextualisation architecture with digital watermarking. The resulting technique achieves a comprehensive

solution that is lightweight enough to be supported by resource-constrained IoT devices. To assess the impact of this technique, we studied a healthcare-related use-case. Via this use-case, we showed that the proposed combination of contextualisation and data obfuscation significantly reduces the amount of data that needs to be processed while also permitting reversibility of data obfuscation. Experimental evaluation of this use-case shows that contextualisation reduces the obfuscation- and de-obfuscation-related data processing requirements by an average of approximately 160 times.

The final contribution of this thesis is the novel contextualisation-based technique that enables multi-granular privacy preservation that is suitable for IoT-constrained environments. For this, we focused on protecting data during the different stages of the IoT application life-cycle. The proposed technique was studied in a smart vehicle use-case, whereby the IoT data stream generated by the smart vehicle is protected by data obfuscation comprised of a combination of digital watermarking and data scrambling. Contextualisation directs the obfuscation intensity applied to the IoT data generated by the smart vehicle based on application-specific rules (e.g., the vehicle's location or driver). This reduces data obfuscation-related processing for specific aspects of the application (e.g., when the vehicle is at a specific location or is driven by a specific driver, as the data generated during these times does not need to be protected). Benchmarking of our use-case demonstrated that this novel technique achieves privacy preservation nine times more rapidly than the most common encryption algorithms.

Research Objectives and Questions Revisited

In particular, in this thesis we have revisited the following research objectives with satisfactory results:

- **Studying the performance and scalability of current state-of-the-art in processing of IoT scale data using contextual information.** [**RQ1**]
  In the Section 2.2.4 we presented the results of the SLR based on the methodology described in Section 2.2. As well as discussing performance and scalability, the inves-

tigation also provided details of the operations performed by the systems described in the papers investigated, and whether or not these operations are contextual.

- **Proposing a scalable and performance-oriented contextualisation technique for IoT data [RQ2]**

  One of the main objectives of this thesis was to provide an IoT contextualisation architecture that is generic and scalable.

  The ConTaaS Architecture presented in Section 3.2 satisfies these stringent requirements. It is general-purpose, allows the user to define the contextual information relevant to their domain, and can be used to realise IoT contextualisation in any application area because of its inherent adaptability.

  Although this architecture is general and can be used in any application area, in this thesis, we only presented two example application areas: health and transportation.

  In addition to this novel, general-purpose, IoT architecture, we looked at the dynamic natures of the different contextual operations performed by systems that use contextualisation. In this thesis, we generalised the types of contextual operations by asserting that any operation can be categorised as one of three classes: filter, aggregate or infer.

- **Designing a sensor cloud solution for contextualisation of IoT data [RQ3]**

  Many of the latest high-performance processing techniques for Big Data, such as MapReduce [6], are not suitable for IoT scale applications as they cannot handle the real-time constraints of IoT scale data processing.

  These latest high-performance processing techniques are designed for batched work, and ignore the incremental data processing requirements of many IoT applications. This leads to the batched nature of these big-data processing techniques not supporting the near real-time requirements of IoT scale applications, as processing data in batches causes re-computation of work already completed. This re-computation simplifies the processing model for big-data applications, but the extra latency introduced by repeating work unnecessarily renders these techniques unsuitable.

In order to facilitate IoT scale data contextualisation, a novel and innovative architecture was designed to support high-performance and scalable contextualisation in real time. This architecture is described in Chapter 3, which also includes an example scenario in which this architecture flourishes.

- **Implementing and demonstrating the proposed model by developing a proof-of-concept implementation, and validating its scalability and performance through experiments [RQ4]**

  We chose to demonstrate the benefits of the novel ConTaaS architecture by implementing it in a smart parking recommender system. A parking recommender in a smart city is hard to design, as they can receive data from millions of IoT sensors (including in the users' cars and devices) while requiring near real-time latency to be useful to drivers. However, solutions for capturing data from the myriad of sensors that could be used, analysing the heterogeneous data produced by these sensors, and then processing it to provide recommendations in near real-time is currently unfeasible due to the difficulties inherent in such a large and heterogeneous system.

  The ConTaaS architecture was implemented on the Amazon Web Services EC2 cloud infrastructure [7] to tackle this formidable IoT contextualisation task. The implementation of the ConTaaS architecture created for this thesis is able to represent and contextualise large amounts of data from IoT devices while providing near real-time, correct and efficient responses that query the contextualised IoT data and recommend parking to users.

- **Utilising contextualisation to improve the security and privacy of IoT Scale data [RQ5]**

  Finally, a lightweight yet highly scalable data obfuscation technique was proposed that combines contextualisation with digital watermarking. The digital watermarking technique is reversible and parameterised. This allows the security system to use this technique to control the perturbation of sensitive data, enabling legitimate users to de-obfuscate perturbed data with ease.

The proposed technique utilises ConTaaS to achieve real-time aggregation and filtering of IoT data for large numbers of designated users, as ConTaaS enhances scalability, as described in Section 5.1.

Next, we proposed a scalable and context-aware granular obfuscation technique for spatial-temporal data. This technique is used for privacy preservation in IoT environments and is capable of multi-granular obfuscation by enforcing context-aware disclosure policies, as described in Section 5.2.

6.1 Future Research

In this section we will briefly discuss future research relevant to this thesis as following:

- Context Acquisition

  The contexts used in this thesis were assumed to be provided by the application or users. A future research direction could explore the development, mapping and integration of additional advanced data analytics methods for contextualisation, such as machine learning and deep learning algorithms, to allow for automated or semi-automated collection and processing of context information.

- Dynamic Context

  The context information considered in this thesis was static. This means we did not consider changes in context information or the relationships between context data. Dynamic context changes are common due to mobility, changes of interest or the task at hand, or changes in the environmental contexts that entities are subject to . One future research direction would be to define dynamic context within IoT ecosystems in such a way that it not only captures changes and variation in context but also assesses the relationships between pieces of context information.

- Context Verification

  This thesis assumed that the context provided is always accurate. However, this is not the case, especially in an IoT ecosystem. Hence, new techniques need to be developed

that can differentiate between good and bad contexts. A future research direction could be to develop a verification method that can validate context (e.g., by using ontologies or formal methods) to improve the effectiveness of contextualisation. These methods should take into account the real-time and time boundary requirements of IoT scale applications.

- Incremental Context Processing

  The architecture described in this thesis is designed to be incremental, though the incremental functionalities of the techniques have not yet been addressed. One future research direction would be to investigate the performance of ConTaaS in advanced processing environments such as incremental MapReduce [150].

- Complex Reasoning

  Investigation of more complex reasoning methods using more complex or even multiple ontologies is another direction for future work. In this thesis, we investigated a few ontologies with respect to our application scenarios. However, ConTaaS has the capability to be extended to take advantage of linked data and more complex ontologies.

Bibliography

[1] L Ericsson. More than 50 billion connected devices. *White Paper*, 14:124, 2011.

[2] John R Boyd. The essence of winning and losing. *Unpublished lecture notes*, 12(23):123–125, 1996.

[3] Andrew Whitmore, Anurag Agarwal, and Li Da Xu. The internet of thingsa survey of topics and trends. *Information Systems Frontiers*, 17(2):261–274, 2015.

[4] Prem Prakash Jayaraman, Xuechao Yang, Ali Yavari, Dimitrios Georgakopoulos, and Xun Yi. Privacy preserving internet of things: From privacy techniques to a blueprint architecture and efficient implementation. *Future Generation Computer Systems*, 2017.

[5] David E Bakken, Rupa Parameswaran, Douglas M Blough, Andy A Franz, and Ty J Palmer. Data obfuscation: Anonymity and desensitization of usable data sets. *IEEE Security and Privacy*, 2(6):34–41, 2004.

[6] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[7] EC Amazon. Instance types, 2014.

[8] Sandeesh Uppoor, Diala Naboulsi, and Marco Fiore. Vehicular mobility trace of the city of cologne, germany. *URL: http://kolntrace. project. citi-lab. fr*, 2011.

[9] Victoria Nebot and Rafael Berlanga. Building data warehouses with semantic web data. *Decision Support Systems*, 52(4):853–868, 2012.

[10] Antoine Zimmermann, Nuno Lopes, Axel Polleres, and Umberto Straccia. A general framework for representing, reasoning and querying with annotated semantic web data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 11:72–95, 2012.

[11] Martin Serrano, Hoan Nguyen Mau Quoc, Danh Le Phuoc, Manfred Hauswirth, John Soldatos, Nikos Kefalakis, Prem Prakash Jayaraman, and Arkady Zaslavsky. Defining the stack for service delivery models and interoperability in the internet of things: a practical case with openiot-vdk. *IEEE Journal on Selected Areas in Communications*, 33(4):676–689, 2015.

[12] Steffen Staab and Rudi Studer. *Handbook on ontologies*. Springer Science & Business Media, 2010.

[13] Mike Botts, George Percivall, Carl Reed, and John Davidson. Ogc® sensor web enablement: Overview and high level architecture. In *GeoSensor networks*, pages 175–190. Springer, 2008.

[14] Tim Bray, Jean Paoli, C Michael Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible markup language (xml). *World Wide Web Journal*, 2(4):27–66, 1997.

[15] Michael Compton, Payam Barnaghi, Luis Bermudez, RaúL GarcíA-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, et al. The ssn ontology of the w3c semantic sensor network incubator group. *Web semantics: science, services and agents on the World Wide Web*, 17:25–32, 2012.

[16] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *International symposium on handheld and ubiquitous computing*, pages 304–307. Springer, 1999.

[17] Albrecht Schmidt, Michael Beigl, and Hans-W Gellersen. There is more to context than location. *Computers & Graphics*, 23(6):893–901, 1999.

[18] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE communications surveys & tutorials*, 16(1):414–454, 2014.

[19] Judith S Bowman, Sandra L Emerson, and Marcy Darnovsky. *The practical SQL handbook: using structured query language*. Addison-Wesley Longman Publishing Co., Inc., 1996.

[20] Rick Cattell. Scalable sql and nosql data stores. *Acm Sigmod Record*, 39(4):12–27, 2011.

[21] World Wide Web Consortium et al. Rdf 1.1 concepts and abstract syntax. *World Wide Web Consortium*, 2014.

[22] Barbara Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26, 2004.

[23] Barbara Kitchenham, O Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology*, 51(1):7–15, 2009.

[24] Carlos Baladron, Javier M Aguiar, Belen Carro, Lorena Calavia, Alejandro Cadenas, and Antonio Sanchez-Esguevillas. Framework for intelligent service adaptation to user's context in next generation networks. *IEEE Communications Magazine*, 50(3), 2012.

[25] Ioanna Roussaki, Nikos Kalatzis, Nicolas Liampotis, Pavlos Kosmides, Miltiades Anagnostou, Kevin Doolin, Edel Jennings, Yiorgos Bouloudis, and Stavros Xynogalas. Context-awareness in wireless and mobile computing revisited to embrace social networking. *IEEE Communications Magazine*, 50(6), 2012.

[26] Youngki Lee, SS Iyengar, Chulhong Min, Younghyun Ju, Seungwoo Kang, Taiwoo Park, Jinwon Lee, Yunseok Rhee, and Junehwa Song. Mobicon: a mobile context-monitoring platform. *Communications of the ACM*, 55(3):54–65, 2012.

[27] George Okeyo, Liming Chen, and Hui Wang. Combining ontological and temporal formalisms for composite activity modelling and recognition in smart homes. *Future Generation Computer Systems*, 39:29–43, 2014.

[28] Elena Yndurain, Daniel Bernhardt, and Celeste Campo. Augmenting mobile search engines to leverage context awareness. *IEEE Internet Computing*, 16(2):17–25, 2012.

[29] Natalia Díaz Rodríguez, Manuel P Cuéllar, Johan Lilius, and Miguel Delgado Calvo-Flores. A fuzzy ontology for semantic modelling and recognition of human behaviour. *Knowledge-Based Systems*, 66:46–60, 2014.

[30] Nirmalya Roy, Sajal K Das, and Christine Julien. Resource-optimized quality-assured ambiguous context mediation framework in pervasive environments. *IEEE transactions on mobile computing*, 11(2):218–229, 2012.

[31] Chang-Woo Song, Daesung Lee, Kyung-Yong Chung, Kee-Wook Rim, and Jung-Hyun Lee. Interactive middleware architecture for lifelog based context awareness. *Multimedia Tools and Applications*, 71(2):813–826, 2014.

[32] Md Abdur Rahman, Heung-Nam Kim, Abdulmotaleb El Saddik, and Wail Gueaieb. A context-aware multimedia framework toward personal social network services. *Multimedia tools and applications*, 71(3):1717–1747, 2014.

[33] Henar Martín, Ana M Bernardos, Josué Iglesias, and José R Casar. Activity logging using lightweight classification techniques in mobile devices. *Personal and ubiquitous computing*, 17(4):675–695, 2013.

[34] Muhammad Ashad Kabir, Jun Han, Jian Yu, and Alan Colman. User-centric social context information management: an ontology-based approach and platform. *Personal and Ubiquitous Computing*, 18(5):1061–1083, 2014.

[35] Mirco Rossi, Sebastian Feese, Oliver Amft, Nils Braune, Sandro Martis, and Gerhard Tröster. Ambientsense: A real-time ambient sound recognition system for

smartphones. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pages 230–235. IEEE, 2013.

[36] Georgios Meditskos, Stamatia Dasiopoulou, Vasiliki Efstathiou, and Ioannis Kompatsiaris. Sp-act: A hybrid framework for complex activity recognition combining owl and sparql rules. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pages 25–30. IEEE, 2013.

[37] Elizabeth Papadopoulou, Sarah Gallacher, Nick K Taylor, and M Howard Williams. A personal smart space approach to realising ambient ecologies. *Pervasive and Mobile Computing*, 8(4):485–499, 2012.

[38] George Okeyo, Liming Chen, Hui Wang, and Roy Sterritt. Dynamic sensor data segmentation for real-time knowledge-driven activity recognition. *Pervasive and Mobile Computing*, 10:155–172, 2014.

[39] Edwin JY Wei and Alvin TS Chan. Campus: A middleware for automated context-aware adaptation decision making at run time. *Pervasive and Mobile Computing*, 9(1):35–56, 2013.

[40] Andrey Boytsov and Arkady Zaslavsky. Formal verification of context and situation models in pervasive computing. *Pervasive and Mobile Computing*, 9(1):98–117, 2013.

[41] Paulo Carreira, Sílvia Resendes, and André C Santos. Towards automatic conflict detection in home and building automation systems. *Pervasive and Mobile Computing*, 12:37–57, 2014.

[42] Bingchuan Yuan and John Herbert. Fuzzy cara-a fuzzy-based context reasoning system for pervasive healthcare. *Procedia Computer Science*, 10:357–365, 2012.

[43] Youngki Lee, Younghyun Ju, Chulhong Min, Seungwoo Kang, Inseok Hwang, and Junehwa Song. Comon: Cooperative ambience monitoring platform with continuity

and benefit awareness. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 43–56. ACM, 2012.

[44] Suman Nath. Ace: exploiting correlation for energy-efficient and continuous context sensing. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 29–42. ACM, 2012.

[45] Kiryong Ha, Zhuo Chen, Wenlu Hu, Wolfgang Richter, Padmanabhan Pillai, and Mahadev Satyanarayanan. Towards wearable cognitive assistance. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 68–81. ACM, 2014.

[46] Abhinav Mehrotra, Veljko Pejovic, and Mirco Musolesi. Sensocial: a middleware for integrating online social networks and mobile sensing data streams. In *Proceedings of the 15th International Middleware Conference*, pages 205–216. ACM, 2014.

[47] Boris Motik, Ian Horrocks, and Su Myeon Kim. Delta-reasoner: a semantic web reasoner for an intelligent mobile platform. In *Proceedings of the 21st International Conference on World Wide Web*, pages 63–72. ACM, 2012.

[48] Marcio EF Maia, Andre Fonteles, Benedito Neto, Romulo Gadelha, Windson Viana, and Rossana Andrade. Loccam-loosely coupled context acquisition middleware. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 534–541. ACM, 2013.

[49] OSGi Alliance. *Osgi service platform, release 3*. IOS Press, Inc., 2003.

[50] Zhi Xu, Kun Bai, and Sencun Zhu. Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*, pages 113–124. ACM, 2012.

[51] Qiang Wei and Zhi Jin. Service discovery for internet of things: a context-awareness perspective. In *Proceedings of the Fourth Asia-Pacific Symposium on Internetware*, page 25. ACM, 2012.

[52] James NK Liu, Yu-Lin He, Edward HY Lim, and Xi-Zhao Wang. A new method for knowledge and information management domain ontology graph model. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(1):115–127, 2013.

[53] Yohan Chon, Elmurod Talipov, and Hojung Cha. Autonomous management of everyday places for a personalized location provider. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):518–531, 2012.

[54] J Zhu, SK Ong, and AYC Nee. An authorable context-aware augmented reality system to assist the maintenance technicians. *The International Journal of Advanced Manufacturing Technology*, 66(9-12):1699–1714, 2013.

[55] Chang Choi, Junho Choi, and Pankoo Kim. Ontology-based access control model for security policy reasoning in cloud computing. *The Journal of Supercomputing*, 67(3):711–722, 2014.

[56] Daqiang Zhang, Min Chen, Hongyu Huang, and Minyi Guo. Decentralized checking of context inconsistency in pervasive computing environments. *The Journal of Supercomputing*, 64(2):256–273, 2013.

[57] George Okeyo, Liming Chen, Hui Wang, and Roy Sterritt. A hybrid ontological and temporal approach for composite activity modelling. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 1763–1770. IEEE, 2012.

[58] Hans Werner Guesgen. *Spatial reasoning based on Allen's temporal logic*. International Computer Science Institute Berkeley, 1989.

[59] Silvia Coradeschi, Amadeo Cesta, Gabriella Cortellessa, Luca Coraci, Javier Gonzalez, Lars Karlsson, Francesco Furfari, Amy Loutfi, Andrea Orlandini, Filippo Palumbo,

et al. Giraffplus: Combining social interaction and long term monitoring for promoting independent living. In *Human system interaction (HSI), 2013 the 6th international conference on*, pages 578–585. IEEE, 2013.

[60] Younghyun Ju, Youngki Lee, Jihyun Yu, Chulhong Min, Insik Shin, and Junehwa Song. Symphoney: A coordinated sensing flow execution engine for concurrent mobile sensing applications. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 211–224. ACM, 2012.

[61] Thuong Nguyen, Dinh Phung, Sunil Gupta, and Svetha Venkatesh. Extraction of latent patterns and contexts from social honest signals using hierarchical dirichlet processes. In *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 47–55. IEEE, 2013.

[62] Arkady Zaslavsky, Prem Prakash Jayaraman, and Shonali Krishnaswamy. Sharelikescrowd: Mobile analytics for participatory sensing and crowd-sourcing applications. In *Data Engineering Workshops (ICDEW), 2013 IEEE 29th International Conference on*, pages 128–135. IEEE, 2013.

[63] Kalla Madhu Sudhana, V Cyril Raj, and RM Suresh. An ontology-based framework for context-aware adaptive e-learning system. In *Computer Communication and Informatics (ICCCI), 2013 International Conference on*, pages 1–6. IEEE, 2013.

[64] Bo Liu, Keman Huang, Jianqiang Li, and MengChu Zhou. An incremental and distributed inference method for large-scale ontologies based on mapreduce paradigm. *IEEE Trans. Cybernetics*, 45(1):53–64, 2015.

[65] James McNaull, Juan Carlos Augusto, Maurice Mulvenna, and Paul McCullagh. Flexible context aware interface for ambient assisted living. *Human-Centric Computing and Information Sciences*, 4(1):1, 2014.

[66] Juan Carlos Augusto, William Carswell, Huiru Zheng, M Mulvenna, Suzanne Martin, P McCullagh, Haiying Wang, J Wallace, and P Jeffers. Nocturnal ambient assisted

living. In *International Joint Conference on Ambient Intelligence*, pages 350–354. Springer, 2011.

[67] Rolf H Weber. Internet of things–new security and privacy challenges. *Computer Law & Security Review*, 26(1):23–30, 2010.

[68] Jim Hahn. Security and privacy for location services and the internet of things. *Library Technology Reports*, 53(1):23–28, 2017.

[69] Aafaf Ouaddah, Anas Abou Elkalam, and Abdellah Ait Ouahman. Towards a novel privacy-preserving access control model based on blockchain technology in iot. In *Europe and MENA Cooperation Advances in Information and Communication Technologies*, pages 523–533. Springer, 2017.

[70] Leonardo A Martucci, Simone Fischer-Hübner, Mark Hartswood, and Marina Jirotka. Privacy and social values in smart cities. In *Designing, Developing, and Facilitating Smart Cities*, pages 89–107. Springer, 2017.

[71] Elias Tragos, Alexandros Fragkiadakis, Vangelis Angelakis, and Henrich C Pöhls. Designing secure iot architectures for smart city applications. In *Designing, Developing, and Facilitating Smart Cities*, pages 63–87. Springer, 2017.

[72] Anupam Bera, Anirban Kundu, Nivedita Ray De Sarkar, and De Mou. Experimental analysis on big data in iot-based architecture. In *Int. Conf. on Data Engineering and Communication Technology*, pages 1–9. Springer, 2017.

[73] Ammar Rayes and Salam Samer. *Internet of Things from Hype to Reality: The Road to Digitization*. Springer, 2016.

[74] Youcef Ould-Yahia, Soumya Banerjee, Samia Bouzefrane, and Hanifa Boucheneb. Exploring formal strategy framework for the security in iot towards e-health context using computational intelligence. In *Internet of Things and Big Data Technologies for Next Generation Healthcare*, pages 63–90. Springer, 2017.

[75] Elisa Bertino and Ravi Sandhu. Database security-concepts, approaches, and challenges. *IEEE Trans. on Dependable and secure computing*, 2(1):2–19, 2005.

[76] Michael J Covington, Wende Long, Srividhya Srinivasan, Anind K Dev, Ahamad Mustaque, and Gregory D Abowd. Securing context-aware applications using environment roles. In *Proc. of the 6-th ACM symposium on Access control models and technologies*, pages 10–20. ACM, 2001.

[77] Junzhe Hu and Alfred C Weaver. A dynamic, context-aware security infrastructure for distributed health-care applications. In *Proc. of the first workshop on pervasive privacy security, privacy, and trust*, pages 1–8. Citeseer, 2004.

[78] Charu C Aggarwal and S Yu Philip. A general survey of privacy-preserving data mining models and algorithms. In *Privacy-preserving data mining*, pages 11–52. Springer, 2008.

[79] Kato Mivule. Utilizing noise addition for data privacy, an overview. *arXiv preprint arXiv:1309.3958*, 2013.

[80] Sheetal Kalra and Sandeep K Sood. Secure authentication scheme for iot and cloud servers. *Pervasive and Mobile Computing*, 24:210–223, 2015.

[81] B Clifford Neuman and Theodore Ts' O. Kerberos: An authentication service for computer networks. *Communications Magazine, IEEE*, 32(9):33–38, 1994.

[82] Ronald L Rivest, Adi Shamir, and Len Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[83] Laurent Eschenauer and Virgil D Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47. ACM, 2002.

[84] Krishna Kumar Venkatasubramanian, Adrish Banerjee, Sandeep KS Gupta, et al. Ekg-based key agreement in body sensor networks. In *INFOCOM Workshops 2008, IEEE*, pages 1–6. IEEE, 2008.

[85] Rodrigo Roman, Cristina Alcaraz, Javier Lopez, and Nicolas Sklavos. Key management systems for sensor networks in the context of the internet of things. *Computers & Electrical Engineering*, 37(2):147–159, 2011.

[86] Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. *ACM*, pages 42–51, 2003.

[87] Seyit A Camtepe and Bülent Yener. Combinatorial design of key distribution mechanisms for wireless sensor networks. *Computer Security–ESORICS 2004*, pages 293–308, 2004.

[88] Donggang Liu and Peng Ning. Establishing pairwise keys in distributed sensor networks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security*, CCS '03, pages 52–61, New York, NY, USA, 2003. ACM.

[89] Chunye Hu, Jie Zhang, and Qiaoyan Wen. An identity-based personal location system with protected privacy in iot. In *Broadband Network and Multimedia Technology (IC-BNMT), 2011 4th IEEE International Conference on*, pages 192–195. IEEE, 2011.

[90] Jing Liu, Yang Xiao, and CL Philip Chen. Internet of things' authentication and access control. *International Journal of Security and Networks*, 7(4):228–241, 2012.

[91] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Advances in Cryptology–EUROCRYPT 2005*, pages 457–473. Springer, 2005.

[92] Shucheng Yu, Kui Ren, and Wenjing Lou. Fdac: Toward fine-grained distributed data access control in wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 22(4):673–686, 2011.

[93] Pablo Picazo-Sanchez, Juan E Tapiador, Pedro Peris-Lopez, and Guillermo Suarez-Tangil. Secure publish-subscribe protocols for heterogeneous medical wireless body area networks. *Sensors*, 14(12):22619–22642, 2014.

[94] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *Advances in CryptologyCRYPTO 2001*, pages 213–229. Springer, 2001.

[95] Petros Belsis and Grammati Pantziou. A k-anonymity privacy-preserving approach in wireless medical monitoring environments. *Personal and ubiquitous computing*, 18(1):61–74, 2014.

[96] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[97] Internationalized Resource Identifiers. Network working group m. duerst request for comments: 3987 w3c category: Standards track m. suignard. Technical report, Microsoft Corporation January, 2005.

[98] Mike Botts and Alexandre Robin. Opengis sensor model language (sensorml) implementation specification. *OpenGIS Implementation Specification OGC*, 7(000), 2007.

[99] Nicholas J Beeching, Manuel Fenech, and Catherine F Houlihan. Ebola virus disease. *Bmj*, 349:g7348, 2014.

[100] Derek Gatherer and Alain Kohl. Zika virus: a previously slow pandemic spreads rapidly through the americas. *Journal of General Virology*, 97(2):269–273, 2016.

[101] Federico Montori, Prem Prakash Jayaraman, Ali Yavari, Alireza Hassani, and Dimitrios Georgakopoulos. The curse of sensing: Survey of techniques and challenges to cope with sparse and dense data in mobile crowd sensing for internet of things. *Pervasive and Mobile Computing*, 2018.

[102] Apache Jena. semantic web framework for java, 2007.

[103] Bastian Quilitz and Ulf Leser. Querying distributed rdf data sources with sparql. In *European Semantic Web Conference*, pages 524–538. Springer, 2008.

[104] Elena Polycarpou, Lambros Lambrinos, and Eftychios Protopapadakis. Smart parking solutions for urban areas. In *2013 IEEE 14th International Symposium on*, pages 1–6. IEEE, 2013.

[105] Paolo Neirotti, Alberto De Marco, Anna Corinna Cagliano, Giulio Mangano, and Francesco Scorrano. Current trends in smart city initiatives: Some stylised facts. *Cities*, 38:25–36, 2014.

[106] Rosamaria Elisa Barone, Tullio Giuffrè, Sabato Marco Siniscalchi, Maria Antonietta Morgano, and Giovanni Tesoriere. Architecture for parking management in smart cities. *IET Intelligent Transport Systems*, 8(5):445–452, 2013.

[107] Rongxing Lu, Xiaodong Lin, Haojin Zhu, and Xuemin Shen. Spark: a new vanet-based smart parking scheme for large parking lots. In *INFOCOM 2009, IEEE*, pages 1413–1421. IEEE, 2009.

[108] Robin Grodi, Danda B Rawat, and Fernando Rios-Gutierrez. Smart parking: Parking occupancy monitoring and visualization system for smart cities. In *SoutheastCon, 2016*, pages 1–5. IEEE, 2016.

[109] Yanxu Zheng, Sutharshan Rajasegarar, and Christopher Leckie. Parking availability prediction for sensor-enabled car parks in smart cities. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015 IEEE Tenth International Conference on*, pages 1–6. IEEE, 2015.

[110] Jim Cherian, Jun Luo, Hongliang Guo, Shen-Shyang Ho, and Richard Wisbrun. Poster: Parkgauge: Gauging the congestion level of parking garages with crowdsensed parking characteristics. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 395–396. ACM, 2015.

[111] Zhanlin Ji, Ivan Ganchev, Máirtín O'Droma, Li Zhao, and Xueji Zhang. A cloud-based car parking middleware for iot-based smart cities: Design and implementation. *Sensors*, 14(12):22372–22393, 2014.

[112] Paulo RL De Almeida, Luiz S Oliveira, Alceu S Britto Jr, Eunelson J Silva Jr, and Alessandro L Koerich. Pklot–a robust dataset for parking lot classification. *Expert Systems with Applications*, 42(11):4937–4949, 2015.

[113] Andrew Koster, Allysson Oliveira, Orlando Volpato, Viviane Delvequio, and Fernando Koch. Recognition and recommendation of parking places. In *Ibero-American Conference on Artificial Intelligence*, pages 675–685. Springer, 2014.

[114] Yanxu Zheng, Sutharshan Rajasegarar, Christopher Leckie, and Marimuthu Palaniswami. Smart car parking: temporal clustering and anomaly detection in urban car parking. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6. IEEE, 2014.

[115] Juan Rico, Juan Sancho, Bruno Cendon, and Miguel Camus. Parking easier by using context information of a smart city: Enabling fast search and management of parking resources. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, pages 1380–1385. IEEE, 2013.

[116] Elham Akhavan-Rezai, Mostafa F Shaaban, Ehab F El-Saadany, and Fakhri Karray. Online intelligent demand management of plug-in electric vehicles in future smart parking lots. *IEEE Systems Journal*, 10(2):483–494, 2016.

[117] Yanfeng Geng and Christos G Cassandras. New" smart parking" system based on resource allocation and reservations. *IEEE Trans. Intelligent Transportation Systems*, 14(3):1129–1139, 2013.

[118] Luca Baroffio, Luca Bondi, Matteo Cesana, Alessandro Enrico Redondi, and Marco Tagliasacchi. A visual sensor network for parking lot occupancy detection in smart cities. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pages 745–750. IEEE, 2015.

[119] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital watermarking and steganography*. Morgan Kaufmann, 2007.

[120] Elisa Bertino, Beng Chin Ooi, Yanjiang Yang, and Robert H Deng. Privacy and ownership preserving of outsourced medical data. In *21st Int. Conf. on Data Engineering*, pages 521–532. IEEE, 2005.

[121] Jiexing Li, Yufei Tao, and Xiaokui Xiao. Preservation of proximity privacy in publishing numerical sensitive data. In *Proc. of the 2008 ACM SIGMOD Int. Conf. on Management of data*, pages 473–486. ACM, 2008.

[122] Dave Beckett and Art Barstow. N-triples. *W3C RDF Core WG Internal Working Draft*, 2001.

[123] Arezou Soltani Panah, Ron Van Schyndel, Timos Sellis, and Elisa Bertino. On the properties of non-media digital watermarking: a review of state of the art techniques. *IEEE Access*, 4:2670–2704, 2016.

[124] Scott Miller and Donald Childers. *Probability and random processes: With applications to signal processing and communications*. Academic Press, 2012.

[125] Anatolii Leukhin and Andrew Tirkel. Ensembles of sequences and arrays. In *2015 Int. Workshop on Signal Design and its Applications in Communications*, pages 5–9. IEEE, 2015.

[126] Arezou Soltani Panah, Ron van Schyndel, Timos Sellis, and Elisa Bertino. In the shadows we trust: a secure aggregation tolerant watermark for data streams. In *IEEE 16th Int. Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–9. IEEE, 2015.

[127] Maurice Maes, Ton Kalker, Jaap Haitsma, and Geert Depovere. Exploiting shift invariance to obtain a high payload in digital image watermarking. In *IEEE Int. Conf. on Multimedia Computing and Systems*, volume 1, pages 7–12. IEEE, 1999.

[128] Tejal Shah, Ali Yavari, Karan Mitra, Saguna Saguna, Prem Prakash Jayaraman, Fethi Rabhi, and Rajiv Ranjan. Remote health care cyber-physical system: quality of service (qos) challenges and opportunities. *IET Cyber-Physical Systems: Theory & Applications*, 1(1):40–48, 2016.

[129] Ali Yavari, Prem Jayaraman, Dimitrios Georgakopoulos, and Surya Nepal. Contaas: An approach to internet-scale contextualization for developing efficient internet of things applications. In *Proc. of the 50th Annual Hawaii Int. Conf. on System Sciences*. IEEE, 2017.

[130] Oscar Moreno and Andrew Tirkel. New optimal low correlation sequences for wireless communications. In *Int. Conf. on Sequences and Their Applications*, pages 212–223. Springer, 2012.

[131] Jonathan K Su and Bernd Girod. Power-spectrum condition for energy-efficient watermarking. *IEEE Trans. on Multimedia*, 4(4):551–560, 2002.

[132] Rodrigo Roman, Jianying Zhou, and Javier Lopez. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10):2266–2279, 2013.

[133] Elisa Bertino. Data security and privacy in the iot. In *EDBT*, volume 2016, pages 1–3, 2016.

[134] Craig Gentry. Toward basing fully homomorphic encryption on worst-case hardness. In *Annual Cryptology Conference*, pages 116–137. Springer, 2010.

[135] Elisa Bertino, Elena Camossi, and Michela Bertolotto. Multi-granular spatio-temporal object models: concepts and research directions. In *International Conference on Object Databases*, pages 132–148. Springer, 2009.

[136] Lorrie Cranor, Marc Langheinrich, Massimo Marchiori, Martin Presler-Marshall, and Joseph Reagle. The platform for privacy preferences 1.0 (p3p1. 0) specification. *W3C recommendation*, 16, 2002.

[137] Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, et al. Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21:79, 2004.

[138] Ali Yavari, Arezou Soltani Panah, Dimitrios Georgakopoulos, Prem Prakash Jayaraman, and Ron van Schyndel. Scalable role-based data disclosure control for the internet of things. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 2226–2233. IEEE, 2017.

[139] Elisa Bertino, Lorenzo Martino, Federica Paci, and Anna Squicciarini. *Security for web services and service-oriented architectures*. Springer Science & Business Media, 2009.

[140] Dimitrios Georgakopoulos, Ali Yavari, Prem Prakash Jayaraman, and Rajiv Ranjan. Towards a risc framework for efficient contextualisation in the iot. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 1993–1996. IEEE, 2017.

[141] John Greenough. The connected car report. http://www.businessinsider.com/connected-car-forecasts-top-manufacturers-leading-car-makers-2016-4-29, 2016.

[142] Matt Duckham. Moving forward: location privacy and location awareness. In *Proc. of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, pages 1–3. ACM, 2010.

[143] Ren-Hung Hwang, Yu-Ling Hsueh, and Hao-Wei Chung. A novel time-obfuscated algorithm for trajectory privacy protection. *IEEE Trans. on Services Computing*, 7(2):126–139, 2014.

[144] Raju Halder, Shantanu Pal, and Agostino Cortesi. Watermarking techniques for relational databases: Survey, classification and comparison. *J. UCS*, 16(21):3164–3190, 2010.

[145] Arezou Soltani Panah, Ron van Schyndel, Timos Sellis, and Elisa Bertino. On the properties of non-media digital watermarking: A review of state of the art techniques. *Special Section on Latest Advances and Emerging Application of Data Hiding,IEEE Access*, 4:2670–2704, 2016.

[146] Solomon W Golomb and Guang Gong. *Signal design for good correlation: for wireless communication, cryptography, and radar*. Cambridge University Press, 2005.

[147] Alberto Peinado, Jorge Munilla, and Amparo Fúster-Sabater. Improving the period and linear span of the sequences generated by dlfsrs. In *Inter. Joint Conf. SOCO14-CISIS14-ICEUTE14*, pages 397–406. Springer, 2014.

[148] Arezou Soltani Panah, Ron van Schyndel, and Timos Sellis. Towards an asynchronous aggregation-capable watermark for end-to-end protection of big data streams. *Future Generation Computer Systems*, 72:288–304, 2017.

[149] Elaine B Barker and John Michael Kelsey. *Recommendation for random number generation using deterministic random bit generators (revised)*. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, Computer Security Division, Information Technology Laboratory, 2007.

[150] Pramod Bhatotia, Alexander Wieder, Rodrigo Rodrigues, Umut A Acar, and Rafael Pasquin. Incoop: Mapreduce for incremental computations. In *Proceedings of the 2nd ACM Symposium on Cloud Computing*, page 7. ACM, 2011.