Technical Report

UCAM-CL-TR-612
ISSN 1476-2986

Number 612

UNIVERSITY OF
CAMBRIDGE

Computer Laboratory

Location privacy in ubiquitous computing

Alastair R. Beresford

January 2005

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted April 2004 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Robinson College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/TechReports/

ISSN 1476-2986

Abstract

The field of ubiquitous computing envisages an era when the average consumer owns hundreds or thousands of mobile and embedded computing devices. These devices will perform actions based on the context of their users, and therefore ubiquitous systems will gather, collate and distribute much more personal information about individuals than computers do today. Much of this personal information will be considered private, and therefore mechanisms which allow users to control the dissemination of these data are vital. Location information is a particularly useful form of context in ubiquitous computing, yet its unconditional distribution can be very invasive.

This dissertation develops novel methods for providing location privacy in ubiquitous computing. Much of the previous work in this area uses access control to enable location privacy. This dissertation takes a different approach and argues that many location-aware applications can function with anonymised location data and that, where this is possible, its use is preferable to that of access control.

Suitable anonymisation of location data is not a trivial task: under a realistic threat model simply removing explicit identifiers does not anonymise location information. This dissertation describes why this is the case and develops two quantitative security models for anonymising location data: the *mix zone model* and the *variable quality model*.

A trusted third-party can use one, or both, models to ensure that all location events given to untrusted applications are suitably anonymised. The mix zone model supports untrusted applications which require accurate location information about users in a set of disjoint physical locations. In contrast, the variable quality model reduces the temporal or spatial accuracy of location information to maintain user anonymity at every location.

Both models provide a quantitative measure of the level of anonymity achieved; therefore any given situation can be analysed to determine the amount of information an attacker can gain through analysis of the anonymised data. The suitability of both these models is demonstrated and the level of location privacy available to users of real location-aware applications is measured.

Contents

Acknowledgements

A lot of people have provided social and technical support throughout this project. Firstly, I would like to thank all the members of the Laboratory for Communication Engineering and especially Jonathan Davies, Kieran Mansley, Andy Rice, David Scott, Alastair Tse, and Jamie Walsh for their timely, skilled and diligent proof reading. I would like to thank my colleagues in the Computer Laboratory who have provided invaluable advice and direction; particular thanks goes to the security group for hosting many thought-provoking meetings, and the natural language and information processing group for use of their natural language tagging software [11]. I offer a special thank you to Catherine, Margaret and Owen for their love and support, particularly through the difficult “writing-up” period.

I am indebted to EPSRC and AT&T Labs – Cambridge for providing financial support and in particular to Andy Hopper who, first as supervisor and later as adviser, has given me the opportunity to take on this project and has provided continuing support and guidance.

Finally, and most importantly, I would like to thank my current supervisor, Frank Stajano. Frank has helped me clarify, justify and arrange my ideas for this dissertation; he has been, and I am sure will continue to be, an excellent mentor.

Chapter 1

Introduction

Most people no longer own just one computer, but a dozen; soon people will own a hundred or a thousand computational devices. For the most part, computer-enabled consumer products available today operate in isolation. However, since wireless network connectivity is becoming cheap and ubiquitous, devices will soon start to inter-operate.

The field of ubiquitous computing envisages an era where the average consumer owns a hundred or thousand inter-connected computing devices. Management of these devices will be impossible unless methods are developed to automate the vast majority of the tasks these devices are designed to perform, and reduce the cognitive load placed on users through improvements in human-computer interaction. In order to achieve these goals, computers need to collect and apply knowledge about the context of the user in the real world.

Providing computers with knowledge of the context of users means that computers will gather, collate and distribute much more personal information about individuals than they do today. Such personal information is often considered private. Therefore, there appears to be a fundamental clash between the needs of context-aware computing and a user's desire to retain control over the distribution and dissemination of private information.

In the field of ubiquitous computing, location information is one of the most common pieces of contextual data, and it is used to drive a wide variety of applications. Yet, when location systems track users automatically and continuously, an enormous amount of potentially sensitive information is generated. Users do not necessarily wish to stop *all* accesses to their location information, because some applications can use this information to provide useful services, but the user wants to be in control.

Technology is not privacy neutral: the precise design and deployment of a technology can have a dramatic effect on the level of privacy enjoyed by the users of the system. Much of the current research into the protection of location privacy for ubiquitous computing has concentrated on defining mechanisms that allow users to control access to their location information; however, explicit and detailed configuration of access parameters runs counter to the aims of ubiquitous computing (namely that of low cognitive load, and automation of tasks whenever possible).

In principle, privacy can be enabled by anonymising all data released to a third-party. Anonymisation has several advantages over access control: users may prefer to remain anonymous; configuration of access control parameters can be difficult and error-prone; and anonymised location data means location-aware applications do not have to be trusted, thus increasing confidence in the protection of location privacy.

This dissertation explores anonymisation as a method of achieving location privacy in ubiquitous comptuing. In particular, it examines whether location-aware applications can function with anonymised location data. In addition, the dissertation demonstrates that, under a realistic threat model, successful anonymisation of location events is often difficult, since removing explicit identifiers is usually not sufficient to protect privacy.

As with many issues in security, a thorough analysis of location privacy through anonymisation is only possible by examining the solutions from the perspective of both attack and defence. Therefore, providing a *quantifiable* metric of the level of location privacy in a particular situation is a prerequisite for analysing exactly how much information an attacker has learnt in a given situation.

Therefore, this dissertation develops two novel security models which provide location anonymity for two different classes of location-aware application. Both methods produce a quantifiable metric to estimate the level of anonymity available. These two security models are used to assess the suitability of enabling location privacy through anonymity for real-world applications deployed using the Active Bat system.

1.1 Dissertation outline

Chapter 2 describes the major challenges in the field of ubiquitous computing and outlines why location information concerning people, places and things is a primary source of context. The chapter provides an overview of the current methods used to capture, process and use location information in a ubiquitous computing environment.

Chapter 3 provides an introduction to privacy and the particular problems presented by the development of ubiquitous computing. The chapter goes on to describe how encryption, access control and anonymisation have been used to protect location privacy in the literature.

Chapter 4 describes several different architectures that support location-aware applications; each architecture represents a different trade-off between the type of applications it supports and the level of trust required for the user to be certain that their location privacy has not been invaded. The chapter concludes with an analysis of how location privacy through anonymity might work with the proposed architectures, and how applications may be adapted to function with anonymised location data.

Chapters 5 and 6 examine in detail the mix zone model and variable quality models respectively. These models anonymise location data for two different classes of location-aware application and provide a quantitative measure of the level of information an attacker can infer through analysis of the (anonymised) data. The algorithmic complexity of generating the quantifiable metrics is assessed, and heuristics are used to improve the performance of the mix zone model.

Chapter 7 describes how the mix zone model and variable quality model can be applied to measure the level of location privacy available in a real-world setting. In particular, the models are used to measure the level of location privacy available for location-aware applications deployed using the Active Bat system.

1.2 Goals

In summary, this dissertation aims to:

- describe why location information is important, and how it is used in ubiquitous computing applications;
- review previous techniques used to protect location privacy;
- describe how many location-aware applications can function with anonymised location information;
- develop two novel security models for anonymising location information which provide quantifiable metrics of location privacy; and
- apply these models to an example set of location-aware applications to assess the feasibility of the approach.

Chapter 2

Location-aware computing

"*Where you are and who you are with are closely correlated with what you are doing.*"
—Ulf Leonhardt and Jeff Magee, 1998. [76]

This chapter provides a survey of work from the field of location-aware computing. The chapter starts by exploring the possible effects small and cheap computing devices interconnected with ubiquitous wireless communication may have on future methods of human-computer interaction. As the cost of computing diminishes, users are going to own larger numbers of devices, so tasks need to be automated whenever possible. In this domain the use of sensor data is important in order to allow applications to model the context of the user and therefore adapt to their needs. In many cases location data is one of the most important pieces of contextual information; this chapter outlines the technologies developed to capture, analyse and distribute location information in an efficient manner.

2.1 Context-aware computing

Traditionally, computing focused on static home and office scenarios, but *mobile* and *pervasive computing*[1] allow us to move beyond these restricted domains. Weiser noticed these trends over ten years ago and used them to provide some predictions about human-computer interaction in the 21^{st} century. He coined the term *ubiquitous computing* [143] to describe a world in which computers "weave themselves into the fabric of everyday life until they are indistinguishable from it." Ubiquitous computing is driven by reductions in the cost, size and power requirements of computers together with the integration of wireless networking. Weiser claimed that "the most profound technologies are those that disappear." He noted that other technologies have done this before. For example, at the beginning of the 20^{th} century a factory contained

[1] The integration of computers into everyday objects is often referred to as *pervasive computing*, a term which does not have a consistent definition. In the Communications of the ACM, Lyytinen and Yoo define pervasive computing as embedded but static infrastructure, in comparison with ubiquitous computing which, in addition to pervasive computing, also supports mobile hosts and mobile code [79]. In contrast, the editor in chief of IEEE Pervasive Computing declares in the inaugural issue [114] that ubiquitous computing and pervasive computing are synonymous. The expression pervasive computing in this dissertation takes the former definition in preference to the latter.

a single motor which drove all the machines, but nowadays small and cheap electric motors give every tool its own motive force; many everyday objects have several electric motors and only a detailed examination of the object may reveal how many motors are installed, where and for what purpose, but the exercise would be pointless. Weiser envisioned computing becoming sufficiently embedded into our world that users would often interact with computers at a subconscious level, much like we do today with the written word on signs, billboards, books and magazines; "in essence . . . only when things disappear in this way are we freed to use them without thinking and so to focus beyond them on new goals" [143].

Humans use their five senses to interact with the environment; they share a rich common language with others and have an implicit understanding of how the world works. In contrast, traditional computer systems have a very restricted set of communication channels (i.e. screen, mouse and keyboard). *Context-aware computing* [115] or *sentient computing* [53] augments computers with sensors and actuators in order to achieve a better understanding of, and interaction with, the physical environment. Such systems collect sensor data, build a model of the environment or *world model* [3] and use the model to provide more useful and intuitive services to users by triggering actuators or automating tasks. Effective context-aware systems should therefore be easier to use than the traditional computing infrastructure commonplace today. In this way, context-aware computing provides a stepping stone toward the ultimate goal of ubiquitous computing.

Automating the collection and interpretation of contextual data relevant to a particular service is essential. Without such a system the user would have to explicitly provide the information—a burden we wish to avoid, since the emergence of mobile and wearable computing combined with pervasive computing often results in rapidly changing user context. Furthermore, users may have difficulty knowing what information is relevant or even know the required data.

A lot of ubiquitous computing research currently focuses on improving human-computer interaction. Tennenhouse argues that, since humans may soon own a hundred or a thousand computers each, more research effort should focus on *proactive computing* [133], taking humans "out of the loop" and reducing human-computer interaction to a supervisory role. Proactive computing aims to extend the application domain and reduce direct human involvement by connecting computers directly to the physical world via sensors and actuators. Data should be processed in real-time, using closed loop operation where possible (allowing the removal of the delay inherent with user confirmation) and user needs should be anticipated using statistical modelling to deal with uncertainty. Proactive computing has similar aims to IBM's *autonomic computing* [54]; Want *et al.* provide a discussion on the similarity between proactive and autonomic computing [138].

2.1.1 Context

Researchers have struggled to provide a precise definition of context. Context is often defined by example; López de Ipiña describes some of the common attributes, including user identity, location, time, features of the natural environment (temperature, light level, air quality), physiology (blood pressure, heart rate), activity (talking, reading, walking), social interaction (including who we are with) and nearby resources [23]. Dey and Abowd provide a thorough analysis of context and context-awareness [26] and provide a useful definition:

> Context is any information that can be used to characterise the situation of an entity.

> An entity is a person, place or object that is considered relevant to the interaction between the user and an application, including the user and the application themselves.

Dey and Abowd proposed[2] four *primary* context types for characterising the particular situation of an entity:

Identity: the identity of relevant entities.

Location: the geographical position of relevant entities.

Activity: the activity or activities being performed.

Time: the time period at which the entities perform the activity.

All other types of context information are declared *secondary* because they often require association with one or more pieces of primary context to be meaningful. For example, a phone number, address or date of birth is usually associated with an identity; similarly temperature information is only useful when combined with a location, time and possibly activity. It is usual for secondary context to require more than one piece of primary context in order to identify unambiguously a single piece of information; in database parlance, the primary contexts must be combined to form a composite key.

2.1.2 Context-awareness

Schilit *et al.* defined four types of context aware applications on two orthogonal axes [115]: whether the task gets *information* or executes a *command* and whether the task is executed *automatically* or *manually*. Four classes of application are then developed for each of these categories; Table 2.1 contains the classifications together with short examples.

Pascoe analysed context-awareness from the perspective of wearable computing and derived four types of context-awareness [96]:

Sensing: detecting states from the environment and presenting the data to the user; for example, presenting the current location of a user on a map.

Adaption: allowing wearable applications to alter their state in response to data from the environment; for example, turning on a back light to the wearable computer when it is dark.

Discovery: combining the context of the current user with information about the environment to determine resources which are available and accessible; for example, printing to the nearest printer or finding the closest open supermarket.

Augmentation: associating digital data with the real world; for example, a tourist guide may associate maps and details of historical artifacts with specific locations.

[2] Dey and Abowd claim another paper [108] by Ryan *et al.* defined these primary context types, but used the term *environment* rather than *activity*—Dey and Abowd correctly point out that *environment* is often used as a synonym for context and the word *activity* more directly describes what an entity is actually doing. The referenced Ryan *et al.* paper however does not provide a definition of four primary contexts types, so their derivation is attributed to Dey and Abowd here.

	Manual	**Automatic**
Information	*Proximate selection*: a user interface technique where objects nearby are made easier to select. *Contextual information:* direct provision of context, such as "Where is Alastair?".	*Automatic contextual reconfiguration*: adding, removing or altering connections between existing components; components are typically servers and connections are communication links, but components may also be loadable device drivers, mobile agents, graphical user interfaces, *etc.*
Command	*Contextual commands*: actions are altered by context, for example the print command always selects the nearest printer.	*Context-triggered actions*: for example, simple `if-then` rules automatically execute a command when the predicate is true.

Table 2.1: Schilit's definition of context-aware applications (in italics) together with short examples.

The definitions derived by Pascoe and Shilit *et al.* are very similar. Dey and Abowd claim a clear mapping between three of the four classes defined by Pascoe and Schilit *et al.*, namely, sensing ⇔ proximate selection, adaption ⇔ context-triggered actions, discovery ⇔ automatic contextual reconfiguration; in reality there is some confusion, particularly in the relation between adaption and context-triggered actions. Dey and Abowd derive their own classification of context aware features: (1) *presentation* of information and services to a user, (2) *automatic execution* of a service, and (3) *tagging* of context to information for later retrieval; they define context-aware as:

> A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.

2.1.3 Augmented reality

Augmented reality is a specific form of context-aware computing concerning the integration of *virtual reality* with thereal world. Augmented reality systems combine real and virtual components in an interactive, real-time fashion by superimposing virtual objects on the real world in three-dimensions. Many augmented reality systems use head-mounted displays to overlay text or graphics information to aid humans in everyday tasks; almost all augmented reality systems display visual information to the user, but it is conceivable that audible or olfactible augmentations could be used. Azuma describes some of the many application domains which have been explored [7], including systems for the medical, military, manufacturing, visualisation and the entertainment industries.

The aim of augmented reality is to enhance user perception of the real world; virtual objects are used to convey information which cannot be detected directly (or at least to a comparable resolution) by the five human senses. A lot of augmented reality research has focused on *registration*, or correctly aligning virtual objects with the real world and *sensing* or *motion tracking* [144] which detects the presence (*i.e.* relative location and orientation) of relevant entities. In order to achieve accurate registration, very accurate sensing of the environment is required.

Historically, context-aware systems have gathered location data from large areas with coarse granularity, whereas augmented reality systems have concentrated on fine-grained location systems with small coverage areas. The two research areas are heading for convergence: augmented reality is now pushing the boundaries of coverage, and context-aware applications are demanding better quality location information in order to permit more accurate, high-level inferences of activity.

2.1.4 Location-awareness

Traditionally, computers have only had methods of determining and sharing two of the four primitive context types: identity (*e.g.* authenticating individuals or computers via passwords, digital certificates, Kerberos tickets *etc.*) and time (*e.g.* network time protocol, Lamport time, *etc.*). Location awareness is something which has become important only recently. Knowledge of identity and time have well-defined applications within the traditional (fixed) computing paradigm, whereas location does not. Location information is most useful in combination with mobile and pervasive computing (made possible through the availability of small, portable and cheap hardware), where context information can provide more intuitive human-computer interaction and more aggressive levels of automation.

Welch and Foxlin categorise location systems as being either [144]: (1) "inside looking out" shortened to *inside-out*; or, (2) "outside looking in" shortened to *outside-in*. An inside-out location system calculates the location and orientation by measuring a physical property of the environment with sensors placed on the device which is attached or carried with the entity; in contrast an outside-in location technology calculates the location of entities from sensors placed in the environment. A *tagged* location system requires components be added to both the (moving) entities and the (static) environment; conversely a *tagless* location system needs only a sensor or sensors in the environment (outside-in) or entity (inside-out). For example GPS is an inside-out, tagged location system, since satellites are placed in the environment and transmit (via radio) an environmental property that each mobile device (the GPS receiver) can measure to determine location information. Conversely, Closed-Circuit Television (CCTV) systems combined with movement detection software form an outside-in tagless system since the network of static cameras placed in the environment detect the movements of entities centrally.

Hopper describes the three methods usually used to record location information [53]: (1) *co-ordinates*: a two- or three-dimensional vector of real numbers representing the distance[3] of an entity from a well-defined origin; (2) *proximity*: a real number (usually thresholded to a binary value) representing how close two or more entities are to one another; and, (3) *containment*: a ternary value representing the positive, negative or partial intersection of containers (usually circles or polygons) representing the interaction space of two or more entities (*e.g.* a laptop is inside a room).

2.2 Location technologies

Understanding the operation and accuracy of a location system is central to the thorough assessment of the risks such a system presents to location privacy, the subject of the next chapter.

[3] For small-scale location systems an Euclidean distance measure is common. Distance measurements on larger scales can often be more complex; see the World Geodetic System (`http://www.wgs84.com/`), an (approximately) spherical polar co-ordinate system.

Static	**Dynamic**
Spatial distortion: level of systematic and random error in measuring position over a defined spatial region.	*Latency:* amount of time taken to collect a reading from (possibly multiple) sensors and estimate the position of the entity.
Creep: long term change in spatial distortion due to changing environmental conditions or moving an object and returning it to the same pose.	*Update rate:* number of updates per second. Sometimes simultaneous or semi-simultaneous updates are possible when measuring multiple entities.
Orientation: level of systematic and random error in measuring orientation along three Cartesian axes.	*Dynamic spatial and orientation error:* location error not found in static spatial distortion, orientation or creep. Dynamic errors of this type include effects of Doppler shift and inaccuracies in any prediction algorithms employed by the system.

Table 2.2: Six accuracy properties of a location system.

Assessment of accuracy and function is also an essential element of providing a solution to some of the privacy problems presented by both location technologies and their corresponding location-aware applications.

In order to locate an entity we must use one or more physical properties of the environment to calculate its position. Many physical properties are amenable to providing location information, but no single technique is suitable for all purposes nor provides all the properties required for every application. Researchers have provided detailed reviews of location technologies in the literature: Hightower and Borriello provide a survey from the perspective of ubiquitous computing [50], Azuma analyses the state-of the-art in augmented reality [7] and Welch and Foxlin assess performance of location systems for motion tracking [144].

Table 2.2 outlines the basic accuracy metrics which can be used to measure the performance of a location system. Other important factors include size, weight, installation requirements, robustness to environment (including visual occlusion, heat, sound, magnetic and radio waves), freedom of movement (*e.g.* wireless operation), power, coverage area and cost.

Currently no location system performs well in all cases; application designers must therefore make a decision about what location system best suits their application domain. A survey of physical media suitable for providing location information to context-aware or augmented reality applications is described next.

2.2.1 Mechanical

Mechanical measurement is the oldest form of location; rulers and tape measures provide a simple method of locating one item with reference to another. More sophisticated mechanical techniques have been developed, including the construction of measurement arms with two or more rigid components interconnected with joints. Measurements of the angles of joints with potentiometers or shaft encoders combined with knowledge of the dimensions of the rigid components allow accurate calculation of the position of one end of the arm relative to the other.

FaroArm[4] and Boom 3C[5] are two examples of high-accuracy mechanical measurement arms. Ivan Sutherland was one of the first to develop head mounted displays and used mechanical measurement arms in early research [129].

AT&T Labs Cambridge developed a mechanical measurement system to survey the Active Bat ceiling receivers. The system employed three ASM WS19 one-dimensional measurement sensors[6] which contained retractable steel cables whose current extended length is measured with shaft encoders. The measurement sensors were mounted at known locations on a large rigid metal frame to triangulate the position of objects in three-dimensional space.

The Active Floor [1] measures the force placed on an array of load cells located under the adjoining corners of four floor tiles; data from the load cells is used to estimate the vertical component of the *ground reaction force* of humans walking on the floor. Headon describes how a Hidden Markov Model can be used to extract higher-level context information such as walking, running and jumping from raw sensor data [47]. Lee and Mase attempted to build an inside-out system by placing accelerometers on users rather than load sensors on the floor [75]; motion is detected and distance travelled from a known location is estimated by detecting the foot striking the floor.

2.2.2 Magnetic

Magnetic location systems measure either a static direct-current field using magnetometers or an alternating-current field using an electromagnetic coil. The natural magnetic field of the Earth can be used to estimate orientation, but accurate measurement of position is not possible. Active systems have been developed which require a source to generate magnetic fields along three orthogonal axis in quick succession. A tag is attached to the entity containing three orthogonal magnetic field sensors which measure the field vector from each of the three generated fields; position and orientation of the sensor with respect to the generator can then be calculated. Polhemus have produced several generations of magnetic tracking systems; their most recent is the Liberty Tracker.[7]

Magnetic tracking suffers from limited range and distortions caused by metal or other electromagnetic fields; Welch and Foxlin claim this effect can often be reduced by using direct-current magnetic fields as opposed to alternating-current systems [144], but this requires any sensor measurement to wait for the transient effects of a change in field strength to die down and careful calibration to remove the effect of any background field.

The Pinger [55] provides proximate location information from tags by detecting the presence of their near-field radio broadcast[8] at a set of receivers; the data transmitted encodes an 8-bit tag identifier using pulse-width modulation. Range is approximately 3 m (95% confidence interval) using an 8 MHz carrier and 60 mm × 90 mm antenna.

[4] `http://www.faro.com/`

[5] `http://www.fakespacelabs.com/`

[6] `http://www.asm-sensor.de/`

[7] `http://www.polhemus.com/LIBERTY.htm`

[8] Radio systems usually operate in the far-field of the antenna, however all antennae have a near-field region with an operating range which is much less than a wavelength; if a loop antenna is used then energy transfer between transmitter and receiver is via magnetic field component.

2.2.3 Radio and microwave

Radio and microwaves can be used to estimate the distance between a transmitter and a receiver either by measuring received signal strength or time-of-flight. Both methods are hindered by the complexity of the indoor and urban radio environment: signal strength estimation is hard to predict in environments containing attenuation due to water (*e.g.* humans) and solid structures such as buildings and moving vehicles; similarly, timing delay methods are impaired by multipath effects and line-of-sight difficulties.

One measurement is sufficient to provide an estimate of proximity or containment; three or more measurements are required to triangulate or multilaterate[9] an estimate of position. Orientation information can be derived either by placing multiple tags on a rigid entity or using past movement history to infer likely current orientation.

Signal strength techniques

Passive RFID tags contain a digital signal processing unit, radio transmitter and inductive loop to provide enough temporary power to transmit the tag identity over the air interface. The tags are queried by a battery or mains-powered tag reader, or *interrogator*, which powers the tag via induction and records the resultant tag identity. Since all power for the tag is provided by induction, the dimensions of the tag's inductive loop restrict the maximum distance between interrogator and tag. Texas Instruments TIRIS tags provide one example of small and low-cost passive RFID tags.[10]

Active RFID tags include a battery power source in order to extend the communications range between transmitter and receiver. 3D-iD [145] provides a scalable cell-based interrogator architecture designed to be deployed in hospitals and factories. A simple roll out would permit location containment measurements by placing interrogators on entry and exit to buildings or specific offices or work areas; additional interrogators can be added until three or more interrogators[11] can query the location of each tag.

There has been substantial interest in using existing wireless LAN networks to determine the location of laptops and PDAs. In order to model the attenuation due to fixed building infrastructure, RADAR [8] collects a set of training data to produce a *radio map* of transmitter signal strengths and signal-to-noise ratios for a regular grid of positions and across the coverage area. When tracking, several of the most recent samples are combined to produce a mean signal strength estimate. The signal strength estimate is compared against the training set and the location of the nearest match[12] determines the estimate of user location.

Smailagic and Kogan developed a set of approaches including inside-out and outside-in methods combined with pattern matching and multilateration techniques to estimate location of wireless LAN users [122]. Youssef *et al.* provide a good, up-to-date overview of existing techniques and present a novel and computationally-efficient statistically-based solution [150].

[9] Multilateration generalises triangulation by using measurements of distance from four or more known locations to provide an estimate of the location of a previously un-located entity. For example, Ward uses regression analysis to remove spurious distance measures and provide an estimate of location error [139].

[10] `http://www.ti-rfid.com/`

[11] Careful placement of just two readers may often be sufficient to determine a three-dimensional position of assets in cases where inventory is placed on shelves or moved down long thin corridors [145].

[12] If there are n installed base stations, a mobile device seeking a location estimate will have n signal strength readings. The estimated position of a mobile device is then calculated by finding the position in the radio map which best matches the n signal strengths recorded by the mobile device.

Timing techniques

Timing data can be combined with speed of signal propagation to estimate geographical distance between transmitter and receiver. Timing-based solutions traditionally suffer from difficulties in accurate timing and multi-path propagation effects in built-up and indoor environments. Direct sequence spread spectrum techniques can be used to mitigate the timing problems, and ultra-wide band systems, such as those from Ubisense,[13] have been proposed and test-systems demonstrated.

The GPS system [38] uses twenty-four satellites circling the earth twice a day to deliver highly accurate and synchronised time and satellite location information to outdoor users with GPS handsets via spread-spectrum radio. The handsets combine location and timing data from four or more satellites via multilateration to determine user location anywhere on the earth. In-building reception is very poor, but since the removal of selective availability, outdoor accuracy is usually better than fifteen metres.[14]

Radio bandwidth is a scarce resource, so mobile phone networks subdivide the service area into many *cells*, each containing a fixed transceiver called a *base station*; a frequency reuse pattern is used to ensure adjacent cells do not use the same transmission wavelength, while cells further apart reuse frequencies to reduce bandwidth requirements. Therefore, in order to receive incoming phone calls, the mobile network should forward an incoming call request to the correct cell.

Since the phones are mobile, users can move; therefore a *handover* mechanism is required to keep track of the mobile phone's current cell location. This is accomplished in GSM by a two level hierarchy. A mobile phone network provider has a central Home Location Register (HLR) which stores a record for each subscriber of the network, containing (among other details) an International Mobile Subscriber Identification (IMSI) number and the current local Mobile Switching Centre (MSC).

An MSC manages a number of base station controllers (which in turn manage several base transceiver stations) covering a contiguous geographical area. The MSC maintains a Visitor Location Register (VLR) containing each subscribers IMSI and HLR as well as the current Location Area (LA), a small number of cells, one of which contains the mobile phone. Movement between LAs controlled by the same MSC updates the location information in the VLR; movement between MSCs requires an update of the HLR to reflect a change in local MSC.

Mobile phone location systems are still very much in development; most location-based services available today can now provide cell-sized location information determined from the MSC VLR record.[15] Drane describes some of the more accurate sub-cell solutions [28] which rely on: (1) measuring the propagation delay via round trip timing from a base station to the phone and back (or vice-versa); (2) Measuring the time-difference-on-arrival of signals transmitted from synchronised base stations at mobile phones (or vice-versa); or, (3) measuring the angle of arrival of mobile phone signals at base station (or vice-versa). These techniques are often combined to provide enhanced performance.[16]

[13] `http://www.ubisense.net/`

[14] `http://www.garmin.com/aboutGPS/`

[15] *e.g.* `http://www.traceamobile.com/`

[16] See `http://www.appliedgenerics.com/`

2.2.4 Acoustic

Multi-path reflections affect sonic transmission in a similar fashion to radio; however, unlike radio, simple narrow-band pulses can be accurately timed to provide a measure of distance because the speed of sound in air is many orders of magnitude slower than radio transmission. Physically larger transducers are capable of radiating higher amplitude acoustic waves and therefore have a larger range when compared with smaller devices. However, a signal-strength distance-measurement approach prohibits long-range location sightings in sonic systems because a wide beam width is required for orientation invariance and this necessitates a small microphone and speaker. Sonic transmission in the range of human hearing is often avoided because of the unwanted distraction it causes,[17] so ultrasound is often used.

The Active Bat system [140] is a high-accuracy outside-in location system capable of tracking over a large coverage area. It uses radio transmission to schedule entities carrying active tags or *Bats* to emit conical-shaped uncoded narrow-band pulses of ultrasound in a forward and upward direction from the tag. Time synchronised receiver units positioned at known locations in the ceiling receive the ultrasound pulse from Bats; a multilateration algorithm is then used to calculate a position and error estimate for the Bat location. An estimate of orientation is achieved either through positioning multiple Bats on a rigid entity or analysing the constellation of visible receiver units with respect to the calculated Bat location. The Bats also contain two control buttons which allow simple control messages to be transmitted over the bidirectional radio link.

The Cricket system [99] is an inside-out location system; ceiling units transmit a narrow-band ultrasound pulse and radio transmission (containing the location of the receiver) simultaneously; receiving tags measure the time delay between radio and ultrasound arrival to determine their distance from the ceiling transmitter and multilateration can be used at the receiver to estimate location.

The Dolphin system uses a novel transducer capable of spread-spectrum ultrasound transmission; this enables data to be encoded over the ultrasound channel, eliminating the need for a radio link. The Dolphin system can work in both inside-out [45] and outside-in [46] modes.

2.2.5 Optical

Optical systems consist of light sources and optical sensors. Light sources may be entities which reflect ambient light or emit light directly (*e.g.* LEDs, lasers, fluorescent tubes or light bulbs). Optical sensors detect the presence or absence of light; both analogue and digital varieties exist and detect light level in one or two dimensions. Light sensors come in three overall varieties: (1) photo sensors, which detects the level of incident light on the sensor; (2) position sensing detectors (PSDs) are analog devices whose output(s) are proportional to the centroid of light intersecting a one- or two-dimensional sensor; and (3) charge-coupled devices (CCDs) which provide an array of pixel data representing a quantised image of the scene sampled over a short duration. A photo sensor has the advantage of being cheap and easy to integrate and PSDs can provide a much higher update rate than CCDs, which must transfer all the pixel data off the device before another sample image can be taken.

Lenses alter the field-of-view of the sensor and filters can be used to select only the wave-

[17] There are occasions when audible location systems are useful; for example, a siren helps people locate a police car.

lengths of light produced by the light sources (reducing background interference and thus simplifying detection). Optical systems require line-of-sight to operate and photo sensors and PSDs suffer from partial occlusion problems, where some of the incident light level is obscured, producing a plausible but inaccurate result.

Photo sensor systems

The Active Badge system [137] uses an infrared communications link to transmit a unique identifying code approximately every 10 seconds from a wearable badge to a network of room receivers connected to a central location service. The diffuse infrared broadcast initially offered room-scale location information, but this was refined to desk scale with the addition of a passive tuned circuit in the badge [44]. Radio transmitters are installed at desk locations and the transmit power and antenna design is controlled so that badges transmit when approximately 1 m from the transmitter. Badges retransmit the radio broadcast over the infrared channel so the system knows when a user is contained within a radio zone. Badges contain two buttons to allow basic interaction between the user and the environment. The PARCTAB system [136] enhanced the functionality of the Active Badge by incorporating a touch-sensitive display, three buttons and a speaker. The infrared system was augmented to provide data communication to allow PARCTABs to act as thin clients.

In contrast to the Active Badge's outside-in system, the Locust Swarm [124] uses small, solar powered nodes attached directly under office light fittings to announce their location via an infrared communications link to users equipped with wearable computers. Users can determine their own location and share this location information via the wearable computer's radio link.

PSD systems

Ward *et al.* developed an optical tracker for head-mounted displays [141]. The system detects light from 960 LEDs mounted in ceiling tiles via four analogue PSDs attached to a head-mounted unit. An LED manager turns on each LED in turn and measures the output of the sensors; previous sightings and careful grouping structure of LEDs is used to prevent a scan of the entire coverage area in most cases. Position and orientation of the head is reconstructed using the position of LEDs contained in the field of view of each PSD combined with the physical juxtaposition of the photo diodes on the head using a technique called *space resection by collinearity*.

CCD systems

Visual location estimation has traditionally been done with active or passive markers, often called *fiducials*, which are attached to the entities to be tracked and viewed with a CCD. A marker-based approach is attractive since fiducials can be designed to minimise computational overhead and ease mathematical analysis of their detection.

Infrared LEDs can be spotted quite easily with CCDs provided any infrared filter is removed; an infrared pass filter can be used to remove unwanted natural light. Ribo uses a pair of cameras mounted on a 2 m baseline to detect three retro-reflective markers attached to a head-mounted see-through display [104]. Knowledge of the relative position of the fiducials mounted on the display combined with the intrinsic[18] and extrinsic[19] parameters of the cameras allow the orientation and position of the display to be accurately calculated. Phicons [81] are small programmable devices with IR transceivers, several buttons and a small LCD; phicons communicate with a host PC via a camera, which obtains data via analysis of inter-frame presence or absence of an infrared spot from (possibly multiple) phicons in the image. The host PC can calculate the two-dimensional location of phicons in the field-of-view of the camera; the intended application is augmentation of a white-board to allow users print or email relevant areas. Arguably, Brightboard provides an enhanced system, detecting and interpreting handwritten commands drawn directly on a white-board [123].

Passive, paper-based markers have proved very popular with researchers. Rekimoto and Nagao describe one of the first paper-based fiducial systems [102] and developed a colour barcode system to annotate real-world objects with additional data which could be viewed with the aid of NAVICAM, a PDA equipped with a camera to capture images, a processing unit (to detect barcodes), and a display to annotate the processed image with additional context information. Rekimoto and Ayatsuka developed the barcode method further [101], producing CYBERCODE, a two-dimensional matrix marker containing 24 bits of data; the camera can determine the position of the marker relative to the camera by analysing the distortion of the fixed-sized fiducial.

The ARTOOLKIT system [64] also uses a square marker system, but allows arbitrary images to be contained within an outer-black frame; markers are differentiated by template matching fiducials against a set of previously configured images. A template system is useful for users interacting with markers directly, since fiducial images can provide intuitive meaning to users. Owen *et al.* extend ARTOOLKIT marker design with the use of orthogonal discrete cosine transform basis images [95]; markers were designed to minimise the probability of misidentification, particularly in noisy and partially occluded scenarios. ARTOOLKIT can extract the target pose (*i.e.* transform from camera to targets in the scene) and therefore is ideally suited for use in augmented reality applications.

Several systems use circular markers, taking advantage of the fact that circles appear as ellipses under any three-dimensional affine transformation. The TRIP system [24] uses circular markers with a solid (inner) ring for marker detection and pose extraction, combined with a series of code rings to provide a large address space of uniquely identifiable tags. The BBC developed the Free-D system [134] to ensure alignment of real and virtual components of a television studio set. The system uses an auxiliary camera pointing up to coded circular markers mounted on the ceiling.

Recently, inside-out scene analysis methods have been explored to locate humans and robots equipped with cameras connected to wearable computers. Starner *et al.* describe a method of using the mean light colour and luminance (as measured from head mounted cameras) combined with a Hidden Markov Model to provide an estimate of location [125]. Aoki *et al.* use colour histograms of a sequence of video frames to detect specific locations [5]. Both of these approaches attempt to sight landmarks in a video sequence and therefore only offer a notion of containment or proximity.

Dellaert *et al.* have developed a coordinate-based approach which uses a visual map of the ceiling of an indoor environment generated by aligning multiple images with a map-building algorithm [25]. To determine location information, a camera measures the mean light intensity

[18] Intrinsic parameters represent the field-of-view and optical distortion pattern of the camera.

[19] Extrinsic parameters represent the three-dimensional position and orientation of the cameras with respect to world coordinates.

of the ceiling from a vertically mounted camera; comparison of the current intensity value with the ceiling map results in a complex, multi-modal distribution; a condensation algorithm is combined with a Bayesian model and a sampling-based representation of possible locations of the entity to iteratively refine an estimation of location. Rungsarityotin and Starner take a similar approach but use a colour histogram and an L2 norm from an omni-directional camera to provide location estimation [107].

More complex analysis of inter-frame pixel movements in a stream of continuous video, often referred to as *optical flow*, has been extensively researched. Optical flow can be analysed from both an outside-in (*i.e.* static cameras and moving objects) as well as an inside-out (*i.e.* static scene and moving camera) perspective; the latter case is often referred to as *ego-motion*. General approaches taken include differential methods, correspondence methods and region-based matching in two or three dimensions. Irani *et al.* describe a robust ego-motion algorithm which relies on measuring the two-dimensional motion parameters of a static planar surface to recover the three-dimensional motion of the camera [56]. Neumann and You combine region detection and the differential method to provide an improved estimate of optical flow [89]. Rekleitis takes a novel approach, using motion blur as an estimate of dilation and translation of the video image [103]. Jiang and Neumann propose augmenting fiducial tracking with natural line detection methods to improve registration in augmented reality demonstrations [61].

The Pfinder [149] uses multi-class statistical model to detect and interpret the movement of people with a static colour camera. The system assumes a static background and single user in the scene in order to make the vision task tractable. In this limited domain, accuracy in tracking position and orientation is very high (< 3 pixels for location and < 5 degrees orientation error). Stillman *et al.* use two wide angle static cameras to calculate the location of individuals and locate the position of pan-tilt-zoom cameras on faces to enable face-recognition [127]. The EasyLiving [69] project detects the location of multiple users passing into and out of the field of view of three stereo cameras; depth information from the stereo cameras is used to help with scene analysis and solve (some) occlusion problems; colour histograms are used to maintain user identity.

2.2.6 Inertial

An inertial sensor contains three gyroscopes and three accelerometers. Traditionally, gyroscopes and accelerometers are mounted on a gimballed platform, with feedback from the gyroscopes used to ensure the the gimballed platform maintains a fixed orientation in all three axes whatever the motion of the entity; this ensures the accelerometers remain aligned to the world coordinate system whatever frame of reference the entity happens to be in. After removing the effect of gravity from the vertical accelerometer, data are then double-integrated to provide a measure of the offset between initialisation and current position. Traditional gimbal-based systems are large and cumbersome and are too heavy to mount on the human body, but these systems have been deployed in ships, submarines and planes.

More recent systems fix three gyroscopes and three accelerometers to a backplane which moves in the frame of reference of the entity. The accelerometer measurements are combined into an acceleration vector which is then rotated by the current entity-world transformation matrix (obtained from current gyroscope measurements). The resulting acceleration vector is then double-integrated and gravity compensated to provide a position in world coordinates. This design of inertial sensor combined with microelectronic mechanical systems (MEMS) integration have led to the development and use of small and very portable inertial sensors. The InertiaCube[20] product is one example popular with augmented reality projects and measures just 34 mm × 24 mm × 41 mm.

The major drawback of inertial systems is drift: minute bias errors in accelerometer measurements quickly lead to large positional errors because double-integration makes the errors cumulative. Inaccuracies in gyroscopes have a similar effect; any orientation error results in erroneous correction of gravitation from the accelerometers; again this results in cumulative positional error.

Inertial systems have been combined with other tracking systems to produce a hybrid location system which combines the high update rate of inertial sensors with the long-term stability of other sensor systems. Inertial sensors can be used to reduce the fiducial search space in a visual system by providing an estimate of movement since the last frame to bound the areas of image which are examined [126]. The VIS-tracker [84] provides wide-area tracking in a similar fashion, using the inertial sensor to provide a high update rate and estimate of the location of each fiducial. The tracker is wearable and can even automatically calculate the location of fiducials as long as four seed target positions are provided and initially visible in the field-of-view of the camera.[21]

2.2.7 Summary

Researchers have explored mechanical, radio, acoustic, optical and inertial methodologies. No single approach is suitable for all applications, although hybrid methods which combine fiducial or optical flow detection with inertial sensor systems are starting to show great promise, particularly for the augmented reality domain. Outside-in tagless tracking of humans is now becoming possible with CCD based systems and tagged tracking is available in the wide area with the near ubiquitous deployment and use of mobile phones.

2.3 Location-aware applications

Applications can be split into two broad categories: *personal* applications, which use only the location information of a single individual and *shared* applications which share location information with several individuals or other applications. Subdividing location-aware applications along any dimension is somewhat artificial, however the division used here clarifies discussion of the relevant privacy issues presented in the next chapter.

2.3.1 Personal location-aware applications

Personal location-aware applications use location information (perhaps in conjunction with other pieces of context) to provide a service to a single user whose location is being tracked. Example services include "Where am I" applications which locate users on a map or convert coordinates into symbolic locations with semantic meaning. The MoBIC project [97] uses a GPS receiver to guide blind users on self-generated routes around Birmingham city centre. Audio

[20] `http://www.isense.com/products/prec/ic2/`

[21] A video of the tracker is available at `http://www.isense.com/support/downloads/vistracker.zip`

data provides direction and proximity information about places of interest along the route (*e.g.* post box, coffee shops *etc.*). The Touring Machine [33] uses GPS, a magnetometer and a two-axis inclinometer (detecting head pitch and roll) to provide location and orientation information of the user; relevant local information is superimposed on a see-through head-mounted display using OpenGL [148], and web pages containing local information are displayed on a hand-held computer. Tourist guides have been a popular location-aware application; the Lancaster GUIDE project is the arguably the most complete prototype system [16] using cell-level location information from 802.11 wireless LAN base stations to display relevant tourist information via web-based software on handheld computers.

An *electronic information lens* [34] uses the location and orientation of a handheld screen to augment the screen's display with extra information relevant to the current position. For example, pointing the electronic lens at a map allows additional information about particular landmarks to be displayed; the distance from the map can be used to zoom in on particular features. Displayed data could be weather information for the local region, traffic congestion, restaurants *etc.* This metaphor was later extended by Rekimoto and Nagao by using passive, colour barcodes to detect the location of objects in the environment and attach virtual information annotations to them (and therefore coping with moving objects as well as moving displays) [102]. Information aids represent a common application domain and prototype systems range from computer repair [51] to aeroplane production [88], allowing head-mounted overlay displays of useful information such as the name and function of computer components or the itemised list of screws to be attached to a particular set of holes in an aircraft fuselage.

Personal location-aware applications can be used to control or automate tasks. A simple example is desktop teleporting [105], where location information is used, in combination with a button press on an Active Badge or Active Bat, to initiate a transfer of the desktop of a user from one particular screen, mouse and keyboard to another using VNC [106]. The Reactive Room [17] was developed to automate and simplify the control of audio and visual presentation equipment; for example, the presence of documents on a tabletop is tracked by a video camera which automatically displays any document on an overhead projector and inserts a copy in the video stream presented to remote participants. An Active Poster [2] allows a Bat under coverage of the Bat system to control devices and set application state. For example, the "sentient" scanner consists of an ordinary scanner connected to a networked computer and printed instructions containing icons representing common controls for the scanner (*e.g.* colour/black & white, glass/sheet feeder *etc.*); the printed instructions are placed near the scanner and users interact with the scanner by placing their Bat within the icon area and clicking a button on the Bat. Scanner software records a series of selections and uses them to control the scan and even transmit the results back to the user via email.

Personal location-aware applications rely on the location of a single individual and as a consequence often use location systems built for inside-out operation. Other context relevant to operation is often fairly static and therefore can be stored on the locating device and operated without network connectivity. Some applications would benefit from updates, for example Fawcett and Robinson demonstrate that traffic navigation could be improved if up-to-date traffic congestion data are available [31].

2.3.2 Shared location-aware applications

Sharing location information of entities with applications such as "Where is Alastair?" introduces the need for distributing up-to-date location information. Sharing location data with others can also be used to infer some types of secondary context information.

Bulk location-aware applications use location data collected from all entities to infer *anonymous* secondary contextual information; in other words, context which is solely a function of location and time, and not of identity. Traffic congestion can be predicted and measured using bulk location data collected from the Trafficmaster system,[22] mobile phone network[23] or highway agency induction loops. Congestion information can then be used as a piece of secondary context in personal applications, for example, to reschedule meetings and adapt travel plans.

Location information from positioning systems which use multilateration techniques can be used to estimate environmental surroundings such as the position of desks, chairs and computer screens in an office. For example, Harle and Hopper model ultrasound transmissions from the Active Bat system as rays intersecting a set of cubes stacked at regular coordinates forming a three-dimensional lattice [42]; given an accurate location of the entity (determined by multilateration) reception of an ultrasound transmission implies there are no obstacles between Bat and ceiling receiver, and all intersected cubes along such a Bat-receiver path are marked as visible. Using ray-tracing methods to process a large collection of location sightings allows accurate updates of environmental surroundings to be performed automatically.

Location information can be used to infer *identity-based* secondary context information; in this scenario, a notion of identity is an essential component of the generated contextual information. One of the earliest examples is PEPYS [90], which generates a retrospective diary of meetings and movements of users from location data generated by an Active Badge network. The diary is designed to aid recall of *gatherings*[24] and meetings. More recently, manpower, fleet or asset scheduling and management tasks, as well as localised dating and chatting services have been proposed using location data from the mobile phone network.[25]

Shared location-aware applications favour an outside-in location system because this greatly simplifies the collection, processing and dissemination of location data and derived anonymous and identity-based secondary context information. Working, tagless, outside-in location systems maximise the recovery of location sightings from as many entities as possible, and as such could be seen as an ideal system design for delivering shared location-aware applications.

2.3.3 Providing user feedback

Users of location-aware applications must be provided with appropriate feedback. The use of sensors to enrich the context made available to humans can easily result in confusion on the part of the user when things go wrong. Simple examples of user feedback include the use of rising

[22] `http://www.trafficmaster.co.uk/`

[23] System trials have been conducted in Finland (`http://news.bbc.co.uk/1/hi/technology/2680561.stm`) and companies are now offering products; one example is Applied Generics (`http://appliedgenerics.com/`) RoDIN24 system which tracks user movement along highways and collates road traffic reports suitable for presentation to users via WAP.

[24] The original paper uses the term gathering in preference to meeting when referring to a casual collection of people; a variety of criteria are used to determine whether a gathering is a meeting, including number of people present, the location of the gathering, the speed at which people grouped and departed and the stability of attendance.

[25] `http://www.umtsworld.com/technology/lcs.htm`

and failing audio tones to signify the success or failure of configuration using Active Posters [2]. The usefulness of the "sentient" scanner is severely curtailed if, ten minutes after a supposedly successful scan, no scanned image appears in the user's email inbox.

2.3.4 Middleware

The overlap of functionality in many location-aware applications has not gone unnoticed. Researchers have developed middleware systems to integrate much of the common processing features of many location-aware applications into a centralised service. This approach reflects the dominance of outside-in approaches to location sensing for shared location-aware applications.

Common features of location-aware middleware include: (1) transformation between different (local and global) frames of reference and between co-ordinate and symbolic representations; (2) abstraction of location information from a distributed set of heterogeneous sensor systems; and (3) limiting communication to relevant changes in location information. Applications typically interact with location-aware middleware in one of two modes: (1) a proactive *query* for the current position of particular entities; and (2) an *event-based* interaction in which applications register interest in a particular geographical region and receive notification when an event of interest occurs. Often location-aware middleware is part of a more general context-aware system, where other primary and secondary forms of context are recorded, processed (to generate derivative forms of context, for example PEPYS) and delivered to applications.

The stick-e note [12] architecture allows applications to attach virtual documents to physical spaces, aiding the construction of personal applications (*e.g.* tourist guide) as well as collaborative services, such as Collaborage [82], which allows users to advertise pieces of context with each other such as an in-out board denoting their absence or presence in the office. Attaching virtual data or information to physical environments is also discussed as motivation for the development of the Locust Swarm [124], which has the ability to attach information labels to physical locations. Cooltown [67] develops a similar idea, connecting the physical and virtual worlds by embedding URLs and web-servers into people, places and things.

The Situated Computing Service [55] presents an event-driven model to program with space. Location-aware applications register interest in particular room-scale location predicates and receive a callback when a (previously false) predicate becomes true or vice-versa. The SPIRIT system [2] utilises the high-resolution of the Bat system and uses a quad-tree based indexing method to allow applications to register interest in positive or negative intersection of polyhedral containers attached either to entities or the environment. Fawcett develops a location service to manage location information from a heterogeneous collection of sensors, translate location information between multiple reference frames and allow applications to register interest in proximity and containment of entities [32].

The Situated Information Service [96] represents entities as objects and context information is attached to objects as member variables. State can either be read directly from sensors or *synthesised* from one or more sensor states; state information is drawn from sensors by a *monitor* and stored in the relevant objects' member variables. Finally, *relationships* are used to create entity dependant state; for example each human entity object has a relationship "nearby printers" which uses human location and a list of printers (and their locations) to determine a list of printers most proximate to the human.

The Context-aware Toolkit [109] is based on the graphical user interface event-driven model; widgets hide the complexity and heterogeneity of a set of sensors (providing a common API), abstract context information (allowing new, higher-level context data to be derived) and provide re-usable building blocks. For example an *IdentityPresence* widget has attributes for its location, time and identity of the last user detected; applications can either query the widget for information, or can register for a callback to notify the application of any state changes in *IdentityPresence* (*i.e.* a change in either the time or identity of the user last present). Widgets can be composed into a hierarchy where the state of one widget is dependant on another.

Katsiri and Mycroft have formalised the calculation and distribution of context in First-Order Logic (FOL) [65]. High-level context is derived from primitive context (*i.e.* context obtained directly from sensors) by expressing a series of implications where the LHS is a FOL formula and the RHS is the higher-order context of interest. For example, the mathmatical expression $\forall \mathsf{loc}(\exists \mathsf{user}\ \mathtt{UserAtLocation}(\mathsf{user}, \mathsf{loc}) \Rightarrow \mathtt{Occupied}(\mathsf{loc}))$ derives a higher form of context (namely, location loc is occupied) by using the universal quantifier to consider all locations and the existential quantifier to determine if any user exists at any particular location. Rules are instantiated into predicates, and all predicates which are true are stored in a knowledge base; a Rete network [36] is used to ensure the knowledge base is updated as new primitive context arrives from sensors. Applications can query the knowledge base directly or register for changes in state through an extended publish/subscribe protocol.

2.4 Summary

This chapter has introduced the field of ubiquitous computing and outlined why improved human-computer interaction and automation of tasks are important requirements in this new computing environment. In order to improve interaction and implement automation, applications of ubiquitous computing need knowledge of the context of the user. There are four primary types of context information (identity, time, location and activity); the traditional (fixed) computing paradigm only required information about identity and time, however a notion of location is a requirement in many ubiquitous computing applications.

A survey of location technologies demonstrated many physical properties of the environment can be used to calculate position, but no single technique is suitable for all application domains. Location information can be measured using a coordinate, container or proximity metrics and sensing technologies can be outside-in or inside-out. Location applications are either personal or shared and are often written with the aid of location-aware middleware which offers either query-based or event-based modes of interaction with location data.

Chapter 3

Privacy

> "*Instantaneous photographs and newspaper enterprise have invaded the sacred precincts of private and domestic life; and numerous mechanical devices threaten to make good the prediction that 'what is whispered in the closet shall be proclaimed from the house-tops.*'"
> —Samuel Warren and Louis Brandeis, 1890. [142]

Privacy has traditionally been viewed as the right to solitude [142] but, with the development of society and democracy, privacy is increasingly seen as *"a key value which underpins human dignity and other key values such as freedom of association and freedom of speech"* [68].

Almost all countries now recognise privacy as a fundamental right and have attempted to codify privacy in law. The first known piece of privacy legislation was the 1361 Justices of the Peace Act (England), which legislated for the arrest of eavesdroppers and stalkers. In 1890 Warren and Brandeis argued that citizens should have the right to prevent disclosure of personal information as a separate right to that of copyright law or of slander or libel [142]. The 1948 Universal Declaration of Human Rights [85, Article 12] states everyone has a right to privacy at home, with family and in correspondence. This declaration was reaffirmed in The European Convention on Human Rights [93, Article 8] and the International Covenant on Civil and Political Rights [86, Article 17].

Many other more recent pieces of legislation aim to protect citizens' privacy; some modern constitutions even go as far as expressing the specific rights of access and control of personal information. Lessig argues that we should go further than just protection and recognise (certain pieces of) information about individuals as "real property" in order to provide individuals with the ability to control the dissemination of their data [77, Chapter 11].

3.1 Defining threats to privacy

Providing a single definition of privacy is difficult. An extensive survey of personal privacy [74] was first carried out by Privacy International as part of The Global Internet Liberty Campaign. The original 1998 report is now revised and extended on a yearly basis by both Privacy International and the Electronic Privacy Information Center (for the 2003 report see [74]). The report identifies four broad personal privacy categories:

Information privacy: protection of data containing personally-identifiable information; examples of personal data include medical records, bank statements and governmental data.

Bodily privacy: protection of people from physical invasion; examples of bodily invasion include drug tests, cavity searches and genetic testing.

Privacy of communications: protection of all forms of communication from interception; examples of interception include monitoring telephone, email and written correspondence.

Territorial privacy: protection of domestic, work and public space from intrusion; examples of intrusion include search warrants, video surveillance and identity checks.

3.1.1 Technological threats to privacy

As society evolves, new threats to personal privacy continue to appear. The development and advancement of science in the 20^{th} century has empowered governments and companies with many ways of invading personal privacy. The goals of governments and companies do not always agree, but the technologies and techniques applied are the same.

As computer systems have developed, their capacity to collect, analyse and distribute personal data is increasing exponentially. At the same time advances in medicine, transportation, finance and communications have resulted in the recording of much larger amounts of sensitive personal information. High-speed networks allow the compilation of substantial personal dossiers of information from widely distributed data centres and geographical locations.

Globalisation has accelerated analysis and distribution of personal data through the development of a growing number of interoperability standards. Examples include the use of the Internet Protocol, facilitating ubiquitous network connectivity, and standardised data storage formats and meta-formats such as XML.

3.1.2 Reasons for privacy invasion

The reasons for privacy invasion are wide and varied. Governments, while wildly varying in political constitution, all demand information from citizens; examples include earnings, family make-up, religion and qualifications (everything from driving ability to medical training). The stated goal of data gathering is that of collective good for society; in many cases the benefit is clear (for example, the regulation of medical practitioners) but in some cases the societal benefits are less obvious, for example, the record of racial origin.[1]

Commercial organisations are primarily concerned with profit, yet companies are often much more invasive than their governmental counterparts. Traditionally, marketing, advertising or brand loyalty[2] have been suggested as the major motivation, but more recently *price dis-*

[1] Stanford students provide an excellent review of the use of computers to aid the the oppression of the black African majority in apartheid South Africa (1948-1994), see `http://www-cs-students.stanford.edu/~cale/cs201/`

[2] The Clubcard was the first major loyalty card introduced in the UK in 1995 (`http://www.guardian.co.uk/weekend/story/0,3605,999866,00.html`) and many major retail outlets have now followed suit. The cards record a history of all purchases and can be used to encourage loyalty to a particular company, allow analysis of customer spending and tempt customers with introductory offers on high-profit branded products which their previous spending pattern has shown they may like. In order to do this, the data collected is analysed to determine such factors as the number of householders (estimated by the amount of toilet roll used) whether you

crimination—the act of charging individuals a personalised price based on the amount they are prepared to pay—has been suggested as a strong motivating factor.

Odlyzko provides a good introduction to price discrimination and the Internet [92]. Discriminatory pricing works best in markets with large, fixed, up-front costs and low marginal costs. With the centralisation of services and reduced cost of communication and transportation, more and more industries fit this model. Price discrimination is not a new phenomenon: it was used extensively throughout the early development of the railways in the US and Britain until a customer backlash introduced extensive regulation. Traditionally its use has been limited by the lack of technology to perform detailed customer profiling.

More recent price discrimination examples include flights, computers and even DVDs.[3] The ability to automate the collection and analysis of consumer profiles (often referred to as *data mining* or *data profiling*) has greatly enhanced corporations ability to dynamically and differentially price products. Odlyzko uses economic theory to suggest that price discrimination combined with efficient customer profiling results in a more efficient market and thus without regulation its use will become widespread [92].

Data mining or profiling (whether for marketing, brand loyalty or price discrimination) will have a stark impact on consumer privacy. Not all users appear to be concerned with data collection and retention policies; provided a majority of users are content with such processing, industry-wide practise may force other less content consumers into participating either through lack of any choice or high discriminatory prices for those wishing to retain privacy.

3.2 Methods of privacy protection

Lessig describes the four basic methods of regulating the behaviour of individuals [77, Chapter 7]: architecture, social customs, market and law. Figure 3.1 provides a visual representation of the interaction between the individual and the four basic methods of regulation; each regulating constraint poses a distinct but interdependent cost on the individual. Constraints can be complementary or opposing, yet all can ultimately be governed by the rule of law. Architecture and the market regulate behaviour before an action can take place: a locked door prevents unwanted entry and the cost of cigarettes deters consumption. Social customs and law regulate behaviour after the event: diners may scorn if you smoke while eating and the police may arrest you for trespassing on private property. Most individuals feel the *threat* of social customs or law before the action even though social customs or law cannot actually prevent the act (unlike architecture or market forces).

Law is able to directly regulate behaviour through statute, and indirectly by regulating architecture, social customs and the market; examples of indirect regulation include building regulations, sex education in schools and cigarette taxation. Such indirect regulation does raise questions of transparency, particularly in the regulation of architecture. For example Lessig de-

have just had a baby, or even when you are about to go on holiday. Companies in different markets are joining forces (*e.g.* the Nectar card) to track consumer spending across sectors in order to offer good customers of one company discounts to become (hopefully good) customers of another.

[3] The region coding scheme on DVDs was (in part) motivated by the desire to enforce a price differential on discs sold in different areas of the globe. In September 2000, Amazon (`http://www.amazon.com/`) experimented with price discrimination of DVDs on a more personal level (`http://zdnet.com.com/2100-11-523742.html`): this practise was quickly discovered by Internet users and was widely reported in the press. The situation had the user community split—some thought the practise was fair, others abhorred it.

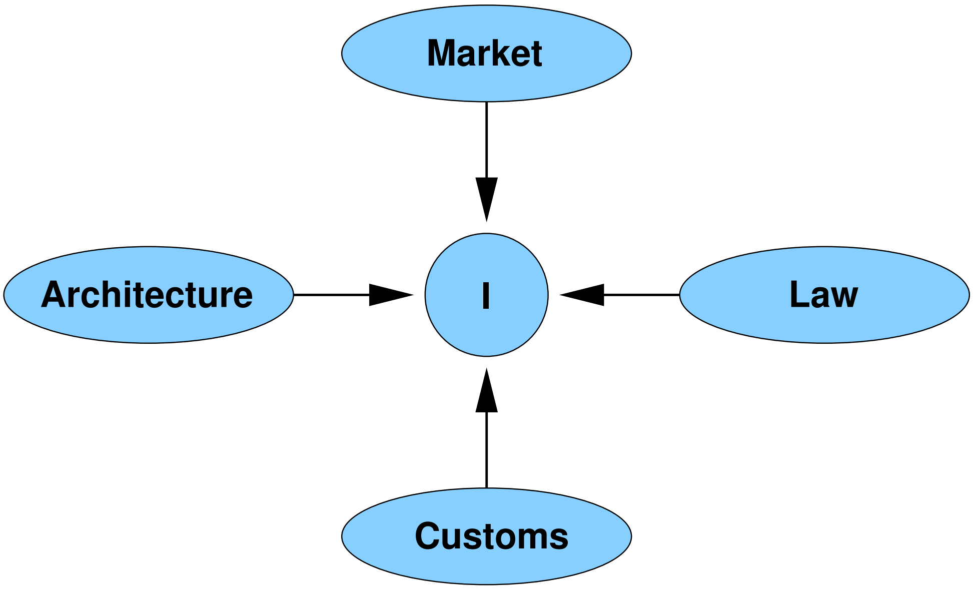


Figure 3.1: Four methods of regulation for the individual I. Each regulation represents a distinct but interdependent cost on the individual.

scribes how Robert Moses commissioned the bridges constructed on Long Island to prevent the passage of buses—the primary transportation of African Americans—preventing the progress of public transport to the beaches [77, Chapter 7 (p.92)].

There are occasions when breaking regulation may be morally justifiable (such as speeding to hospital with the seriously ill); law and social customs are good at adapting in this situation, but architecture and the market are not (a vehicle's speed limiter always limits). Therefore architectural and market forces are arguably more powerful but less adaptable regulators.

This distinction has important implications for the regulation of privacy. Law alone is not sufficient as it is not the most efficient method to prevent unwanted intrusion into the private lives of individuals because: (1) privacy is very subjective—what is acceptable to one person is unacceptable to another; (2) post-action retribution through law may not be able to provide any good remedy—"the cat may already be out of the bag." This does not mean law-makers provide no aid in protecting privacy; quite the opposite is true—law has a crucial role to play in controlling architecture, market forces and social customs to foster privacy.

3.2.1 Methods of regulating privacy

Techniques for protecting personal privacy differ markedly across the globe; four broad approaches have emerged:

Comprehensive: Europe, Australia, New Zealand and Canada have a comprehensive regulatory model with a public official in charge of enforcing data protection legislation.

Sectorial: the United States has not defined general data protection legislation, but has instead enacted a series of specific laws to deal with problems as they arise; specific regulation often lags behind technology, and can leave vast areas of commerce insufficiently regulated.

Self-regulation: various industries have attempted self-regulation through codes of practice, however many of these bodies suffer from a lack of effective enforcement and therefore have less impact than legislation; furthermore, industry-wide consensus does not necessarily result in good consumer privacy, although the threat of government intervention for poorly performing industry codes of practise may have some positive effect. The Press Complaints Commission[4] is an example of a self-regulatory body representing the British press.

Technological: technology can provide solutions as well as threats to personal privacy; Pretty Good Privacy (PGP) is a well known example of a technology enabling privacy of communications, and was so good at this, the United States government disliked its use and distribution immensely.[5]

3.2.2 The comprehensive model

The European Union provides the biggest example of comprehensive data protection legislation regulating collection of and access to *personally-identifiable* information. The 1995 Data Protection Directive [135] was designed to harmonise data protection law across all member states and embodies Westin's principles of privacy legislation [146], namely openness and transparency, reasonable security, accountability, collection and limitation, data quality, usage limitation and individual participation.

Accountability, openness and transparency is ensured through the *supervisory authority* [135, Article 28] which has the power to investigate, intervene, temporarily or permanently stop processing and ban the use of, or order the erasure of data. Further, the supervisory authority has the power to initiate legal proceedings in cases where the directive has been violated. All companies and individuals collecting and processing personal data must notify the supervisory authority [135, Article 18] of their intention to process data.

The directive demands that personally identifiable data are [135, Article 6]: (1) collected for explicit and legitimate purposes, and not further processed in a way incompatible with the original purposes; (2) accurate and kept up-to-date (inaccurate data must be either erased or rectified); (3) relevant and not excessive with respect to the purpose of collection; and (4) stored for as long as is necessary for the purposes for which the data was collected.

The directive also requires explicit consent [135, Article 7] or a demonstration of necessity of processing—this can include legal or contractual requirement where the data subject is a party, or for the protection of vital interests of the data subject (for example, emergency recovery of medical records in the event of an accident). Furthermore, data subjects are also entitled to access their personally identifiable information "without constraint, at reasonable intervals and without excessive delay or expense" [135, Article 12].

The directive requires any countries receiving data from within the EU to have enacted a similar level of data protection legislation [135, Article 25]. This last requirement has had far reaching implications for other countries wishing to do business with any European Union country. In particular, the United States has enacted the Safe Habor principles[6] to permit business to transfer personal data from the European Union.

[4] http://www.pcc.org.uk/

[5] http://www.cdt.org/crypto/current_legis/960626_Zimm_test.html

[6] http://www.export.gov/safeharbor/

The comprehensive model proposes an *architecture*, backed by legal punishment, to regulate and control data processing of personally identifiable information. The directive however does not dictate choice; as Langheinrich notes [72], notice and consent are of little good if consumers have no choice.

Privacy is a dynamic concept—with the introduction of new technologies, social customs and patterns of use develop around it and acceptable behaviour changes. Technology is not privacy neutral: the details of the design and implementation of computer systems have a profound effect on personal privacy. Technological (and in particular computer) developments have hugely increased the possibilities for invasions of privacy by governments, corporations and individuals.

3.3 Access control

Access control can be used to protect information privacy by building an architecture which allows one principal to impose restrictions on the ability of other principals to retrieve information. For example, the EU Data Protection Directive describes a legal architecture (which can be enforced through an appropriate software system) to protect the privacy of an individual's personally identifiable data. A wide variety of access control methods have been developed and many of these have been applied to control access to location information (with varying degrees of success).

Mandatory access control or multilevel security [4, Chapter 7] was initially developed by governments and military organisations to ensure confidentiality of data. The Bell-LaPadula model [9] defines a partially ordered set of security labels attached to each object or data file, denoting their level of *classification* (*e.g.* restricted $<$ classified $<$ secret $<$ topsecret); users are authorised to read or write to files up to a specific level of security label (*e.g.* $\leq$ secret), often called their *security clearance*.

The system then enforces two restrictions on an executing user process (which is invoked at a fixed security level the user has clearance for): (1) a process cannot read any objects or files with a higher security label than itself (no read-up); and, (2) data cannot be written to an object or file with a security label lower than the security level of the user process (no write-down).

An alternative to a fixed user process security level is the high-water principle: a process inherits the security level of the highest data file read since execution began. Notice that if a process' security level increases (*e.g.* because the process started at the classified level but then read a file with a secret security label) any file handles open to classified level files with write permissions must be closed (to prevent violation of the write-down property). The Biba model uses a similar system to enforce integrity, ensuring no data can be read from a data item with a lower label and information can only be written from a higher level to a lower one.

Discretionary access control has traditionally been used to control access to, or usage of, resources in distributed systems. Users are individuals typically authenticated using passwords or Kerberos [60] and their access to program or data files is mediated by an access matrix. An *access matrix* [71] has a row for each domain (or user) and a column for each object (*e.g.* file); each element in the matrix M_{ij} represents the access attributes (*e.g.* read, write, append) for a particular domain i and object j.

The access matrix is usually large and sparse, so the matrix is indexed by objects (access control lists) or by domains (capabilities) and the system checks access attributes when a domain (*e.g.* user process) attempts to access an object. Adding or removing users in the case of access

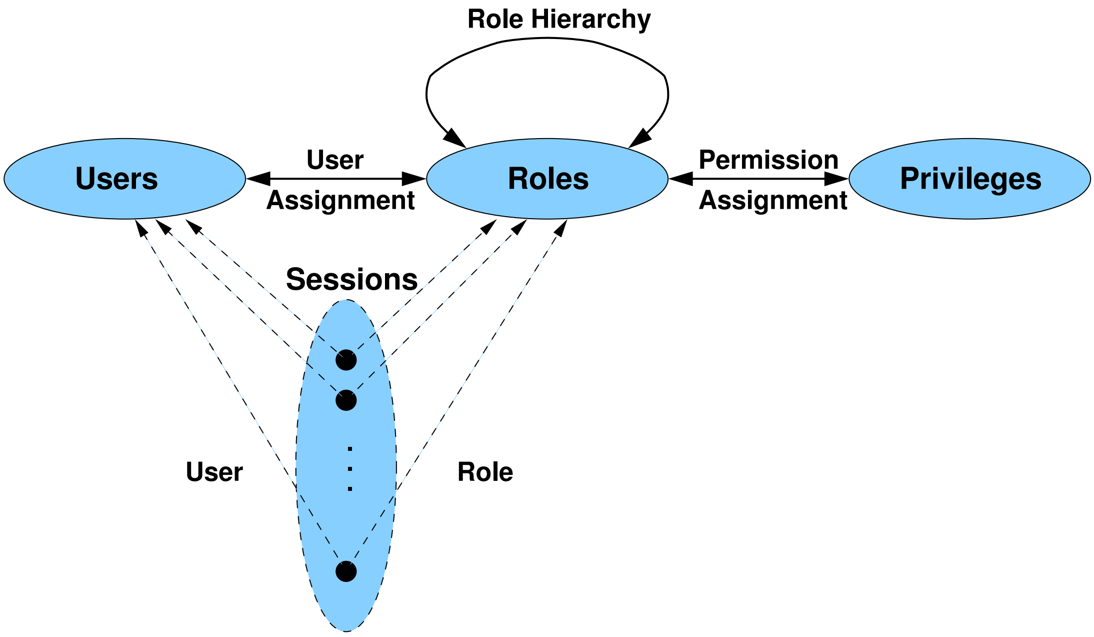


Figure 3.2: RBAC adds a layer of indirection between users and privileges.

control lists (or objects in the case of capabilities) requires each row (column) to be checked. For this reason, an access control matrix is not suitable for representing systems where both the objects and users change frequently.

Multilevel security is designed to allow centralised enforcement of secrecy or integrity. It is not suitable for protecting personal privacy because users of the system do not have ultimate control over the dissemination of their personal data. An access matrix (as represented by ACLs or capabilities) is also not suitable for context-aware computing because: (1) both the user community and availability of resources (objects) are highly dynamic and (2) the plethora of devices means that manually configuring access parameters for each individual entity would be impossible.

3.3.1 Role-based access control

Role-based Access Control (RBAC) allows both the user community and object permissions to be dynamic by introducing a layer of indirection between users and privileges; see Figure 3.2. At first glance roles appear to be merely groups of users, but there is a difference: groups contain a list of principals whereas roles are a set of permissions which one or more authorised users may assume for a period of time (for further information see [4, Chapter 4 (p.54)]). Furthermore, Osborn *et al.* demonstrated RBAC can be used to represent both discretionary and mandatory access control models, and as such generalises both schemes [94].

Sandhu *et al.* develop the NIST model of role-based access control [113], and divide current research and commercial systems into a hierarchy of four categories (each category includes the preceding category as part of its functionality):

Flat: user-role and permission-role assignment is a many-to-many relation and users can activate more than one role simultaneously; *user-role* review should be possible, in other words, an administrator (or the user themselves) should be able to determine both the roles available to a particular person, and the people permitted to activate any given role.

Hierarchical: roles are organised into either a restricted hierarchy (usually some form of tree) or a general hierarchy (*i.e.* graph); permissions associated with a role are inherited, reflecting the authority and responsibility structures of an organisation.

Constrained: the assignment of roles is restricted in order to ensure separation of duty; enforcement can be static (*e.g.* a user with the role Billing Clerk cannot also be assigned the role Accounts Received Clerk) or it can be dynamic, in other words, a user cannot activate the two roles simultaneously (*e.g.* one of two users with the role Billing Clerk can switch roles to Accounts Received Clerk to cover an absent colleague); to enable effective dynamic behaviour, users activate roles in a session.

Symmetric: the flat model required user-role review and symmetric RBAC extends this by requiring *role-permission* review, allowing an administrator to determine which roles can access any given object or vice-versa; effective role-permission review is difficult when role permissions are established over multiple administrative boundaries, requiring the cooperation of several administrators.

Covington *et al.* describe the generalisation of RBAC to allow *environmental roles* as opposed to the subject roles found in the traditional RBAC scheme described above [19]. Subject roles granted to a particular user are activated when a user presents the correct credentials required to authenticate their identity. In contrast environmental roles are activated when environmental conditions (*e.g.* the location or activity of particular users) are met and therefore contextual information about the environment must be collected from sensors in a secure fashion. Therefore environmental roles have the potential to be used as a method of enforcing privacy constraints based on the current state of the environment (*e.g.* "only users located in the same building as me can access my location information").

3.3.2 Multi-subject, multi-target policies

Section 2.3.4 described two dominant modes of access for location information: proactive queries and event-based callbacks. Two common types of query to a location-aware middleware are "Where is X?" and "Who is at location Y?". Leonhardt and Magee demonstrated that using an access matrix to control access to location information is not efficient, since representing the access matrix by capabilities makes access control by location inefficient [76]; similarly, representing the access matrix by ACLs makes specifying access control by user identity inefficient. Access control parameters for location information often require specifying both location-based and user-based access restrictions. Leonhardt and Magee's solution to this problem is to permit multi-target policies of the form:

$$\begin{array}{ccc} \mathsf{Alastair} & \{\mathsf{accessCoLocation}\} & \mathsf{Frank, WilliamGatesBuilding} \\ < \texttt{subject} > & < \texttt{action} > & < \texttt{target} >, < \texttt{target} >, \ldots \end{array}$$

(Alastair is given access to Frank's location when Frank is co-located within the William Gates Building.) The solution can also describe multi-subject policies; for example:

$$\mathsf{Margaret, Richard} \quad \{\mathsf{access}\} \quad \mathsf{Alastair}$$

(Mother (Margaret) and father (Richard) can *together* access the location of their son, Alastair). Policies with up to n multiple targets and m multiple objects are possible; note that in this generalisation, we end up with an $m \times n$ dimensional matrix to represent the access parameters. Therefore the policies are very expressive, but can be hard to efficiently check at run-time and difficult for users to understand. In an attempt to rationalise the complexity, the proposed system defines three layers of policy control (access to data is denied or modified by each layer in turn):

Access layer: all unauthorised requests are rejected here (*e.g.* $\mathsf{Alastair}\ \{\mathsf{accessCo{-}location}\}\ \mathsf{Frank}, \mathsf{WilliamGatesBuilding}$ rejects any requests Alastair makes for Frank's location except when Frank is located in the William Gate Building). Access requests meeting the access precondition may have visibility (accuracy) and identity constrained by visibility and anonymity levels.

Visibility layer: the resolution of location accuracy is reduced here (*e.g.* $\mathsf{Alastair}\ \{\mathsf{reduceResolution}\}\ \mathsf{Frank}, \mathsf{WilliamGatesBuilding/Floor2}$ states that if Frank is located within William Gates Building, Floor 2, his precise location should be replaced with "Floor 2").

Anonymity layer: the identity of subjects can be anonymised here (*e.g.* $\mathsf{Alastair}\ \{\mathsf{replaceID}\}\ \mathsf{Anon}, \mathsf{WilliamGatesBuilding}$ states that Alastair should receive anonymised location information concerning anybody located within the William Gates Building).

The paper says nothing about time-based constraints, although these may be introduced into the model at the visibility level. Similarly, the system does not mention roles (as opposed to explicit identities in the subject clause); adding a level of indirection provided by roles has the potential to reduce the number of rules needed significantly.

3.3.3 Encryption and digital certificates

To protect the confidentiality of messages sent over an untrusted channel, messages can be encrypted. Formally, an encryption function is a bijection between a set of plaintext messages and ciphertext messages. To encrypt a message the encryption function is applied to the plaintext, yielding the ciphertext; to decrypt, the inverse function of encryption—decryption—is applied to the ciphertext, returning the original plaintext.

Encryption and decryption functions are usually parameterised by a key, producing a set of encryption and decryption pairs. Kerckhoffs' Principle states that the security of a system should reside solely in the key, not in the obscurity of the system; this principle provides several benefits: (1) if the system is compromised, only a new key must be selected, rather than a new cryptography system, (2) placing all secrecy in the key enables an open peer-review of the cryptography system, which may improve the scheme by revealing faults or weaknesses; Anderson provides several examples of badly thought-out closed systems which may well have been improved if the design had been open to peer-review [4].

An attacker can mount a *brute-force attack* by attempting to decrypt a given piece of cipher text with all possible keys. A well designed cryptography system will ensure the *keyspace*, or the number of unique bijections (or encryption-decryption pairs) between plaintext and ciphertext, can be made large enough to make any brute-force attack computationally infeasible. Shannon demonstrated that only one cipher offered perfect secrecy [121] (every possible plain text of length n is equally likely to be the decrypted result of a given ciphertext of length n); this cipher is called the *one time pad* and works by applying an exclusive-or function on the plaintext and a key (a random number used once) to generate the ciphertext. Unfortunately, the one time pad has the cumbersome property of requiring as much key material as plaintext, making its use very expensive in key material (which is therefore difficult to distribute safely and economically). In his paper, Shannon also described two properties a more practical cipher algorithm must have: (1) confusion (each key bit influences as many ciphertext bits as possible); and (2) diffusion (each plaintext bit influences as many ciphertext bits as possible). These two properties make a *known-plaintext* attack (a cryptanalyst has access to a particular message in both plaintext and ciphertext form) or a *chosen-plaintext* attack (a cryptanalyst can choose his own plaintext message and read the resulting ciphertext) difficult.

Traditionally, ciphers have been symmetric—anyone in possession of the key can encrypt and decrypt messages. A cryptography system with a single secret or *shared key* requires users to distribute the key securely before they are able communicate confidentially; this is difficult if users have never met or the number of users is very large. Diffie and Hellman introduced the notion of *public key* cryptography to solve this problem [27]. Public key cryptography uses different keys for each pair of encryption and decryption functions with the property that given the encryption key it is infeasible to infer the (paired) decryption key.

The key used for encryption, or *public key*, can be distributed to everyone; conversely, the decryption key, or *private key*, must be kept secret. The system can be used in reverse (provided that the domain of plaintext messages and ciphertext messages is identical): the decryption function applied to a plaintext message yields a ciphertext message that anyone can decode using the corresponding encryption function, but can only have been created by the person with the private key; in other words the message has been *digitally signed*.

Diffie and Hellman proposed users publish their public keys in an on-line directory binding names to keys; users can then look up the public key for the person with whom they wish to communicate and encrypt a message to the person, safe in the knowledge that the only person who can decrypt the message is the one with the private key. Similarly, users who receive a digitally signed message can retrieve the public key of the user and check the message was indeed written by them.

The major problem with this scheme is that a single on-line public directory quickly becomes a performance bottleneck. Kohnfelder proposed digitally signed directory entries, or *certificates* to overcome this difficulty; well-known and trusted third parties, or *certificate authorities* (CAs) sign $(\mathsf{name}, \mathsf{key})$ pairs and these certificates are then distributed by untrusted servers. Provided the certificate authority is trusted and its public key well known, the public keys of other entities can be downloaded from untrusted sources and their integrity checked (by checking the certificate is indeed signed by a trusted certificate authority).

The ITU-T Recommendation X.500 describes a globally distributed method of defining unique names of principals (*e.g.* people, computers, companies, countries *etc.*) called *distinguished names*. Principals are organised into a tree structure, and a distinguished name is derived from the path between the root node and the principal. In order to make the scheme scalable, trust for managing the structure is distributed. The ITU-T Recommendation X.509 [58] describes a hierarchical certificate scheme originally developed to grant access to manage a particular sub-tree, however it is now more often used to bind a public key to a individual or company identified by a distinguished name. X.509 is most widely used in authenticating websites using Transport Layer Security (formally known as Secure Sockets Layer).

Pretty Good Privacy (PGP) favours a decentralised approach over the hierarchical trust built

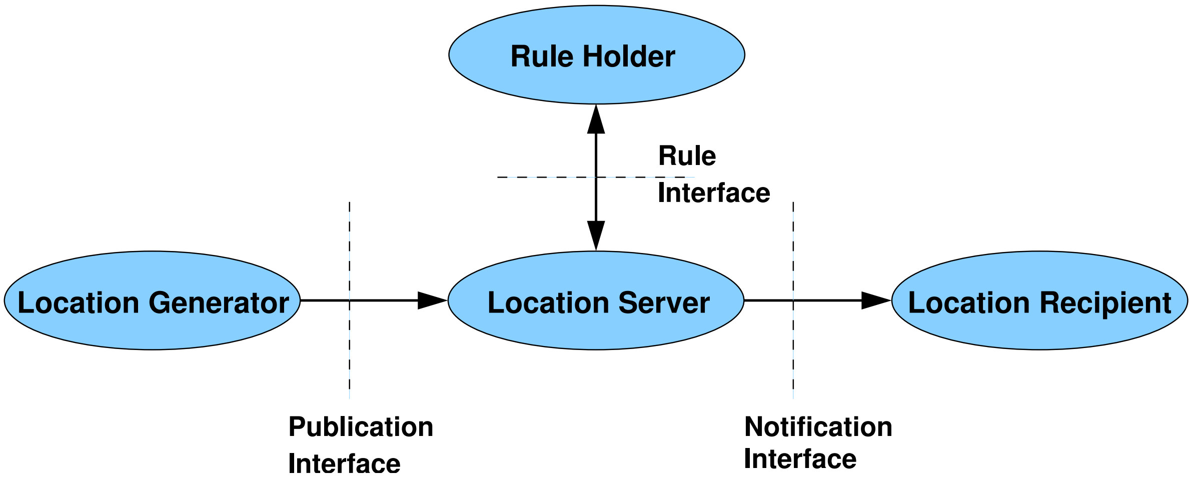


Figure 3.3: The four primary Geopriv components.

with X.509. Users authenticate with each other in a peer-to-peer fashion and a chain of trust is built from people you directly know to their peers, and peers of peers, and so on, to validate certified public keys from previously untrusted sources. User can assign a trust level to their friends (depending on how carefully they believe friends will have vetted their peers) in order to provide a quantitative measure of the amount of trust which should be placed in a particular chain. Trust can then be evaluated to decide whether to accept a certified key as valid or not (*e.g.* two independent verification certificates from well trusted sources may be sufficient, but five independent verifications may be required from less trusted informants).

The X.500 directory structure is unlikely to be ever fully realised; managing a single (albeit distributed) naming scheme is an extremely difficult thing to co-ordinate and manage. In the light of this difficultly, two projects were born: Rivest and Lampson developed the Simple Distributed Security Infrastructure (SDSI) at MIT and Ellison developed the Simple Public Key Infrastructure (SPKI) [30]. The commonality of the two approaches resulted in a combined solution: SPKI/SDSI described in RFC 2693 [29].

In a similar fashion to PGP, SPKI/SDSI does not have a global name-space. In the cases where a globally unique identifier is required, a hash of the public key of the principal is used. To aid human recollection (and to define authorisation polices, described later) a local name-space is defined for each principal. Principals can refer to another name space by qualifying it with a global identifier; the same name can bind to more than one key to form a group of subjects.

3.3.4 IETF Geopriv Working Group

The Geopriv Working Group[7] define a location services architecture designed to protect location privacy. The requirements draft [21] defines four (logically distinct) main components and two data objects (privacy rules and location objects); see Figure 3.3.

A *location object* is used to represent location information (and possibly privacy rules) associated with a particular identity. A *location generator* determines the location of the *target* (a tag or object in the case of a tagless system) and constructs a location object to describe the

[7] http://www.ietf.org/html.charters/geopriv-charter.html

location of the tag.

The location generator then publishes the location object to one or more *location servers* which receives location objects from (possibly multiple) location generators; the location server may receive subscriptions for location information from location recipients.

The location server applies any privacy rules it learns from the rule holder to the location objects it receives and then notifies the location recipient of the location object as necessary. The *rule holder* object initially stores privacy rules on behalf of the *rule maker*. The rule maker is usually the owner of both the location device and the location information; there are exceptions however (*e.g.* parents may be rule makers of their children's location information and employees may be rule makers for corporate mobile telephones). A location server may query the rule holder for a set of rules or rules may be pushed from the rule holder to the location server. Rule delivery must be authenticated and encrypted since the contents of rules may be confidential.

The location object has a number of fields, including target identifier, recipient identity and credentials, location (a symbolic or co-ordinate based region), location type (format of location information), time taken, time-to-live,[8] privacy rule (either a URI to a rule, or an included rule), security headers and trailers (for confidentiality and authentication).

The Geopriv Working Group envisage the location object being used both to send and request location information (*e.g.* a location object with target identifier but no location information could be passed from the location recipient to location server to indicate a request for location information about the target identifier); currently the completed fields needed for a common set of requests and publications are not defined; the IETF draft declares "this is probably out[side] the scope of Geopriv."

The rule maker can amend any of the parameters in the location object to increase the privacy of the target; common adjustments may include (1) anonymising the target identifier by using pseudonyms; (2) reducing the spatial resolution (*e.g.* convert 15 Bridge Street, Cambridge to City Centre, Cambridge); and (3) reducing the temporal resolution. The use of a truth value (to flag whether the data is correct or falsified) is explicitly prohibited in the requirements document.

The architecture aims to support both request-response (location recipient asks for the current location of a target) or event notification (location recipient wants a callback when an event of interest occurs). In the latest policy rules draft [116] the working group anticipates that the location recipient can also provide additional rules to filter out events of no interest (*e.g.* write a rule which reduces the temporal accuracy to once an hour).

3.3.5 Certificate-based privacy protection

Hengartner and Steenkiste describe an architecture to protect personal location data based on digital certificates [49]. The system takes location data from a heterogeneous array of location sensors or estimators (GPS, WaveLAN and personal calendars) and combines these data feeds into a *people locator service* so any entity or *location seeker* can query location information of either a user or room by name; users specify user policies which restrict location data given out for user queries. Users can "own" rooms and therefore set room policies; alternatively, room policies can be set by central policy. Users can configure the system to modify the data given

[8] The length of time the information is valid, not the data retention period.

to location seekers by either reducing the granularity (for user queries) or by anonymising (for room queries). Time intervals can also be used to restrict queries to particular hours of the day.

Room policies and user policies can conflict and may require an arbiter to resolve any differences (the paper suggests several possible solutions, including providing preference to user or room policies, requiring approval from both policies, or attempting policy synchronisation). Scott *et al.* describe a similar problem present in managing the control of sentient mobile agents which can use local resources in conflicting ways [117]; for example, if Alice has a mobile agent which follows her and plays music out of the nearest available speakers, and Bob likes peace and quiet, so defines a policy to prevent music playing near him, what should happen when Bob and Alice meet?

Since location sensors, estimators and people locator systems may be run by different administrative domains, trust is required when transferring data between them. Location policy and therefore privacy is enforced as follows: (1) services which respond to a location seeker request must perform a location policy check to ensure the seeker has the required access rights; (2) services receiving requests from other services check the requesting service is trusted; and (3) services can delegate both location policy and service trust checks to other services (delegation can be used to eliminate redundant checks). Service trust and location policies are stated as SPKI/SDSI digital certificates and a service attempts to build a chain of certificates from itself to the location seeker (to demonstrate a valid location policy) or another service (to prove a service is trusted).

The Hengartner and Steenkiste system is a well thought-out system for controlling access to location information of users, directly by users. The authors do not discuss how their system deals with event-based applications or applications not under direct human control (*e.g.* PEPYS diary creation).

3.3.6 Notice and consent

Myles *et al.* describe a location privacy management system [83] based on LocServ (a Lancaster location server middleware system, currently under development). Users register their privacy preferences with relevant location servers (*e.g.* mobile phone company, local hospital location system *etc.*). Each user has a number of *validators* which are software components capable of making privacy preference judgements on behalf of the user. Example validators include basic manual user confirmation as well as automatic validators which use some form of user context (*e.g.* calendar data stating the user is in a meeting with at least three colleagues) to make an automated, yet informed decision about the likely desired privacy policy of the user. Validators may query other validators in order to reach a decision. Therefore a privacy preference policy for a particular user is a single validator, which may in turn call other validators for help in making access control decisions.

Applications present signed privacy statements detailing their proposed use of location data and the location privacy management system consults the relevant validators to determine whether the application can access the requested location information. Applications are trusted to adhere to the privacy preferences they provide in a similar way to P3P,[9] which trusts web sites to adhere to a privacy specification presented to web clients. The privacy specification is then backed up with legal sanctions (*e.g.* EU Data Protection Directive or contract law) to ensure application writers adhere to the privacy statement; this may be a suitable approach for

[9] See `http://www.w3.org/P3P/`

well known entities (such as applications run by large corporations), but does little to prevent applications written by obscure or unknown third parties.

Langheinrich developed a similar system called pawS [73] which aims to provide users of pervasive computing technologies with notification of any privacy invasion which may take place through sensors and computing infrastructure placed in the user's current environment. The pawS system requests user consent for the use of any personally identifiable data required for the operation of any installed pervasive systems.

3.4 Anonymisation

Anonymity is defined by the Oxford English Dictionary as "the state of being anonymous" which in turn is described as "nameless, having no name; of unknown name." Anonymity can be used in order to protect privacy by ensuring any information released to an untrusted party cannot be associated with a real-world entity; then, at least according to the EU Data Protection Directive, no personally identifiable information is released, and therefore information privacy is maintained. This dissertation uses the definition of anonymity common in the security community and promoted by Pfitzmann and Köhntopp [98]:

> "anonymity is the state of being not identifiable within a set of subjects, the anonymity set."

The anonymity set is associated with an action or role performed by the members of the set. For example, in anonymous communications the sender anonymity set is the set of people who could have written a message which was subsequently intercepted by an attacker; similarly, the receiver anonymity set is the set of people who could have received a message intercepted by an attacker. Measuring anonymity with an anonymity set is useful because it allows the definition of quantifiable metrics for anonymity.

Traditionally the cardinality of the anonymity set has been used as a measure of anonymity; in this dissertation the phrase a user is "k-anonymous" or "has k-anonymity" means that the user is one of at least k users within a specific anonymity set associated with a particular action. Recently, Serjantov and Danezis defined an information theoretic measure of anonymity [119]: each member a_i of the anonymity set A is assigned a probability equal to the (estimated) likelihood that member a_i performed the anonymous action. Shannon's entropy measure [120] can then be used to quantify the level of uncertainty or anonymity the members of the anonymity set achieve collectively with respect to their action.

Reiter and Rubin defined six different degrees of anonymity as part of their Crowds system [100] built to anonymise web transactions. The degrees of privacy protection are:

Absolute privacy: an attacker cannot distinguish between the occasions when a sender transmits a message and the occasions when they do not.

Beyond suspicion: from the attacker's viewpoint, the sender is no more likely to have sent a message than any other sender in the system.

Probable innocence: from the attacker's viewpoint, the sender no more likely to have transmitted the message than not.

Possible innocence: from the attacker's viewpoint, there is a non-trivial probability the sender did not transmit a message.

Exposed: from the attacker's viewpoint there is a only a trivial probability that the sender did not transmit a message.

Provably exposed: an attacker can prove the identity of the sender to others.

There are two principal difficulties with using anonymity to guarantee privacy in context-aware systems: (1) ensuring communication between the user and the application is anonymous, and (2) ensure the actual data provided does not reveal the identity of the user. The following two sub-sections describe the related work in these two research areas.

3.4.1 Anonymous communication

The majority of cryptography and protocol analysis assumes an *omnipresent attacker* who is capable of viewing all communication links; such an attacker is only constrained by cryptography which ensures perfect secrecy. Creese *et al.* point out that while it is sensible to adopt a threat model which includes the worst case scenario, an omnipresent attacker is not always realistic in the peer-to-peer and ad-hoc environment of ubiquitous computing [20]; decreasing the perceived power of an attacker's capabilities may be sensible in some scenarios, and can lead to the development of new security protocols which otherwise would not have been developed.

Anonymous communication systems have traditionally assumed an omnipresent attacker in the design and analysis of privacy-preserving networks. More recently, many researchers have questioned whether this threat model is realistic. The notion of anonymous digital communication was introduced by Chaum [15] in 1981; he proposed a system called a *mix network* to provide *unlinkability* of sender and receiver, which ensures that while an attacker can determine the sender and receiver are communicating, he cannot determine whom they are communicating with. A mix network is a store-and-forward network which contains normal message routing nodes alongside special *mix nodes*. The sender specifies the route of the message through one or more mix nodes using a protocol to ensure unlinkability.

A protocol based on public key cryptography is used to ensure the message cannot be tracked by an attacker as it passes through the network. Consider an analogy of sending a letter between Alice and Bob where public key encryption is equivalent to placing a message inside an envelope (the encryption function uses the public key of the addressee written on the front of the envelope). If Alice wishes to communicate with Bob, she first writes her message and then places it in an envelope with Bob's address on the front; Alice then places this envelope inside another envelope with the address of the final mix node on the front. Alice continues recursively packaging the envelope inside more envelopes until the address written on the front of the letter is that of the first node in the mix network. Alice then sends the envelope to the first mix node.

In its simplest form (called a *threshold mix*) a mix node waits until it collects n messages as input, removes the envelope to reveal the address of the next mix (or final destination) and reorders the envelopes by some metric (*e.g.* lexicographically based on the address) before forwarding them. Provided all messages are padded to be of equal length the system provides a measure of unlinkability between incoming and outgoing messages; even an omnipresent attacker cannot trace a message from its source to its destination without the collusion of the mix nodes.

Aside from unlinkability, two other anonymity properties are often required: (1) *sender anonymity* where an attacker cannot determine the identity of the sender; and (2) *receiver anonymity* where an attacker cannot determine the identity of the receiver. The Dining Cryptographers [14] protocol is one famous example of a method of providing sender anonymity. Receiver anonymity can be achieved by *anonymous broadcast* (*i.e.* sending a datagram addressed to an implicit address to a large number of recipients) and an attacker cannot tell which of the recipients decoded and read the message. Unfortunately the data must be broadcast to a large set of unintended recipients, possibly causing a communication overhead in order to increase the size of the anonymity set; this may be acceptable for wireless transmission from fixed infrastructure to densely-packed mobile nodes.

An *implicit address* (as opposed to an explicit address) is an identifier known only to the recipient. The implicit address can either be visible (*e.g.* available in the header of the datagram), in which case either successive messages to the same recipient can be linked, or the sender and recipient need to have as many addresses as messages. Alternatively, the implicit address can be invisible (*e.g.* encrypt the message data with a public key) in which case every host must consume resources in an attempt to read the message. Linkable visible implicit addressing can be implemented quite efficiently, but unlinkable visible addressing or invisible addressing may consume too much storage or processing resource.

3.4.2 Using implicit addressing to protect location privacy

A mobile phone network must keep track of the cell location of users in order to allow mobile phones to receive calls. The current GSM implementation keeps track of all mobile phones in centralised HLR and VLR databases. To increase capacity in GSM, cells are made smaller, maintaining a roughly constant number of users within the cell whatever the cell coverage area (a mobile operator wants each cell to contain a similar number of subscribers to maintain a constant load across the network). In UMTS an overlapping cell hierarchy is proposed to increase capacity (macro-, micro and pico-cells); the number of users in a cell will drop as users progress down the cell hierarchy (this also results in an increase in location accuracy). Thus the location information stored in a UMTS network (particularly one with lots of pico-cells) stores more accurate location data than a GSM network.

Replacing the explicit addressing of a mobile phone with implicit addressing would also result in UMTS providing less location privacy in comparison to a GSM system (where the number of users per cell—the anonymity set—is larger). When mobile phones transmit data, a more accurate location estimate can be determined using the sub-cell location methods described in the last chapter, thus reducing the anonymity set size further.

GSM and UMTS attempt to achieve a modicum of location privacy by preventing a mobile phone from transmitting its IMSI in clear text over the radio link. Instead the VLR issues the mobile phone with a *Temporary Mobile Subscriber Identity* (GSM) or *Temporary Mobile User Identity* (UMTS); this temporary address is then used to identify the mobile as it roams through the network. Unfortunately, in a GSM network a mobile phone can be forced to transmit its IMSI in clear text when first connecting to the network or when a software error occurs in the VLR. Devices have been built to exploit these loopholes in GSM and force a mobile phone to transmit its IMSI in clear text.

Kesdogan *et al.* proposed removing the HLR and VLR location databases from the operator and storing the current location of a mobile phone in a Home PC (HPC) [66]. Phone calls and location updates use a mix network to route data and thus prevent the network operator from determining the current location of its subscribers. To prevent an MSC from tracking the long-term location of a phone (which can be used to infer user identity, see Chapter 4), implicit addressing of the mobile is required. Kesdogan *et al.* also provide two pseudonym-based solutions to the location privacy problem presented by GSM. The first proposal uses temporary pseudonyms, which are changed at fixed time intervals; both the HPC and the mobile phone know the sequence of pseudonyms used (the paper suggests a stream cipher with a shared secret key) and the mobile terminal re-registers with the network under the new pseudonym after each update. The HPC is queried for the current pseudonym of the mobile by the operator when a mobile terminated call is placed. Location information of the subscriber is revealed when in-call; to prevent abuse of the HPC by the operator, limits on the look up rate or authentication of the caller could be required before the current pseudonym is released.

The second proposal removes the need for an HPC. The mobile provides the network with a hash [128, Chapter 7] of the IMSI, $\text{IMSI}' = h(R, \text{IMSI})$ where R is a random number. The hash function is designed to result in many phones sharing the same hash value, and thus building an anonymity set for each group of users under a common pseudonym. The LA of particular hash values is recorded in the VLR; similarly the presence of hash values at a particular VLR is recorded in the HLR.

To place a call to subscriber A, the caller provides IMSI'_A and invisible implicit address of the subscriber, $K_A(N, \text{IMSI}'_A)$ to the network.[10] The network pages all LAs which contain a subscriber matching IMSI'_A (determined from the HLR and VLR records). Each mobile handset with a matching IMSI' number attempts to decrypt the initialisation value, the succeeding one can then reveal his identity if he wishes to accept the call.

Jackson describes the development of a location anonymiser for the Active Badge system [59]. The system aims to place minimal trust in the network of badge sensor nodes, and achieves this by requiring users to transfer their badge identity through a mix network. An *address label* (anonymous mix message) is transmitted between mix nodes (the first mix node is the badge sensor); the mix node decrypts the outermost layer of the address label, determines the next hop to send the data to, encrypts the (possibly already encrypted) location of the badge with the key provided in the data portion of the outermost part of the address label, and sends the encrypted location data and the remains of the address label to the next hop. The process is repeated until the user's location server eventually receives the location result encrypted multiple times (by prior arrangement the user's badge and user's location server must agree on the keys to be used to encrypt the location information).

Jackson argues that the address label must be changed frequently to prevent a passive global hostile observer from monitoring all communication links and correlating the address label components with users. The assumption of unlinkability of successive address labels only exists for the coarse temporal and spatial granularity of the Active Badge data (and even then, only if there are many people sighted by the same Badge Sensor). Such a scheme would not work without granularity or coverage reduction for the Active Bat system.

[10] The authors do not elaborate on how callers know the IMSI, R and public key K_A; these could be pre-exchanged using a secure side-channel (*e.g.* exchange numbers in person in a similar way to swapping phone numbers).

3.4.3 Inference control

A wide variety of organisations collect personal information from individuals in the form of a survey or census. Inference Control or Statistical Disclosure Control (SDC) is the discipline concerned with the modification of personal confidential data collected from individuals in order to prevent third-parties using the data (*e.g.* government departments, academic researchers and pharmaceutical companies) from determining confidential information about those individuals.

Traditionally data was provided to third-parties in the form of aggregated tables of values, but increasingly individual data values, called *microdata*, are used. *Statistical disclosure* occurs when data provided for statistical purposes is misused. For example, a sociologist investigating urban deprivation notices data concerning an (anonymous) 40-50 year old vicar, knows there is only one such respondent in the geographic area and knows his or her identity; the sociologist can then misuse the data to disclose potentially sensitive personal data.

SDC techniques can be used to remove some of the information content in the released data in order to reduce the disclosure risk. Clearly removing all the data removes any disclosure risk but also prevents any use of data! Therefore there is a trade-off between maximisation of information content and minimisation of disclosure risk.

Microdata tables consist of one or more rows, representing data collected from different survey participants (or *records*), and one or more columns representing the value (or *variable*) of an answer given by the correspondent to a particular question. Formally, we define each column variable to have a domain (a set of values a variable can take). A *categorical variable* is a variable which can assume a finite set of values or categories; examples include sex, race and profession.

Some variables, such as age, income and profit can be considered as continuous, however these can be categorised by rounding values to the nearest year or one thousand pounds (in such cases we have potentially lost some information content, however survey respondents rarely know their own age to the second, or income to the nearest pound, so careful categorisation does not necessarily represent a large loss in information content). Top and bottom coding is sometimes used for variables without bounds; a variable whose value is greater or less than a threshold is placed in the top or bottom category (*e.g.* a profit/loss variable could have a top coding of $\geq$ \$5M and a bottom coding of $\leq$ -\$3M).

In some cases it is possible to define a *proximity graph* on a domain to express the closeness of particular categories. For example, age has a linear graph structure, profession is often a hierarchical tree-like graph structure and religion can be represented by a general graph. A proximity graph can be used to enable automatic *recoding* of the data, in other words, combining adjacent categories together to reduce data accuracy.

An attacker analysing microdata will first want to re-identify an individual, and then use that information to determine a *confidential variable* (a fact previously undisclosed to the attacker, but revealed in the microdata).[11] Therefore *direct identifiers* such as name, address, personal identity number and so on should be removed.

Once direct identifiers are removed, some of the remaining variables may be *indirect identifiers*, variables which an attacker can use to infer user identity. Determining which variables are potential identifiers depends on the *disclosure scenario*, in other words, the data an attacker has

[11] It could be argued that if microdata contains no confidential variables, then no disclosure has occurred, publicly available information is simply being re-distributed; counter arguments to this approach are (1) making data much more accessible may be considered privacy invasive; (2) the very act of taking part in a survey is often considered confidential.

access to. This could be public information, such as phone books, or more personal information such as date of birth or colleagues' desk locations in an office. To assess the disclosure risk properly we must take into account what an attacker may or may not know; in security parlance, a threat model.

An example of an indirect identifier are employee salaries in a small company. Company salaries are likely to be distinct for most workers, so an adversary armed with a list of employee earnings can use the salaries variable to re-identify individual records, and then use this information to associate other confidential variables with particular identities. For example an employee working in the payroll department may be able to determine the identities of workers in internally published performance microdata (and therefore determine colleagues' number of sick days, performance review ratings *etc.*).

Indirect identifiers are similar to keys in databases (the difference being that the keys do not have to be unique). Using indirect identifiers to determine values of confidential variables is called *predictive disclosure*; predictive disclosure can either be *deterministic*, by providing a precise value for the confidential variable, or *probabilistic*, by representing the uncertainty of the value of the confidential variable within a probabilistic framework.

Willenborg and de Waal describe several techniques which can be used to reduce the accuracy of variables [147]; the main categories are discussed next:

Global[12]recoding: two or more categories of a variable in a microdata file are combined into one. A proximity graph can be used to select suitable categories to combine.

Local suppression: replacing a variable value by a missing value indicator. Suppression can be done at a local level (unlike global recoding) so two records with the same variable value may be treated differently (*i.e.* one suppressed and the other retained).

Synthesis: rather than releasing actual data, synthetic data is generated from a model fitted to the real data. A less drastic approach is to replace only a subset of data with synthetic values; if there exist only a small number of confidential variables, these could be replaced with synthesised values.

Subsampling: releasing only a subset of records in a microdata set.

Perturbation: adding a random vector to a continuous variable. The random vectors are usually independent and drawn from a continuous probability distribution with a mean of zero (to prevent any bias of linear estimates drawn from the same data). Noise might not be readily apparent to the third-party, so rounding can be applied to make it more obvious.

Microaggregation: the microdata set is sorted with respect to a quantitative variable. Groups of consecutive variables are replaced by the group mean, preserving the grand total of the variable.

One or more of the disclosure protection techniques can be combined to provide a hybrid disclosure protection method. A measure of the loss of information would be particularly useful

[12] The term *global* is used here to distinguish it from *local recoding*, a method which combines categories on a record by record basis. Local recoding can result in combined records with the same (original) variable values being placed into different categories; this makes analysis by a third-party statistician hard, and is therefore usually avoided. Global recoding and local suppression are two common and more restrictive cases of local recoding, which have better defined semantics.

in this case in order to guide decisions about which combinations of techniques are preferable. Entropy can be a useful metric in this context.

Most SDC techniques assume *simple microdata*, in other words, that column variables are uncorrelated. There are cases where *complex microdata* exist; for example, consider microdata on records containing column variables on disease and sex; there are cases where correlations may allow us to undo recoding or suppression, *e.g.* $\mathsf{disease} = \mathsf{cancer\ of\ the\ womb} \Rightarrow \mathsf{sex} = \mathsf{female}$, so even if $\mathsf{sex} = \mathsf{missing}$, the value of the sex variable can be inferred. In such cases, general SDC methods fail to work properly, and a solution to the specific problem is often required [147, Section 1.3]; it turns out that location information forms complex microdata under certain disclosure scenarios.

In addition to SDC techniques, restricted data queries can be used if third-parties access the data remotely (*e.g.* from a trusted database server connected to the Internet) rather than being provided with the complete microdata set. The US census restrict queries using the "n-respondent, $k\%$-dominance rule" [4, p.175–176]: do not release a statistic when $k\%$ or more of the variable value is contributed by n or fewer records.

Statistical disclosure control for location data

Gruteser and Grunwald describe a method of reducing either the temporal or spatial accuracy of location information in order to anonymise location data [40]. Location information is recorded in container format. Spatial resolution reduction merges containers together so that there are at least k users inside every released container whereas temporal resolution reduction delays the release of any location events from a particular container until at least k users have visited the container. Chapter 6 begins with a more detailed analysis of this algorithm and presents a number of attacks on the original protocol.

Markkula describes a method of reducing the accuracy of location information primarily for use in anonymising static sets of location data of the type found in census data [80]. The algorithm converts the location data contained within each record to a container of a fixed size. Micro-aggregation is then performed on any container with fewer than a threshold of k users by replacing the actual number of users in the container with the average number of users within the set of containers present in the surrounding geographical area.

3.5 Summary

There are four methods of regulating the actions of an individual: market forces, architecture, social customs and law; furthermore, law can be used to influence the market, architecture and social customs. There is a long history concerning the legal regulation of privacy invasion, and different countries have adopted different regulatory methods. Privacy protection through legal means alone is not sufficient to ensure the privacy of citizens.

Technology is not privacy neutral—the architectural design can have a huge impact on the level of privacy offered to the users of the system. This chapter has analysed the two basic methods of designing privacy-aware systems: (1) access control; and (2) anonymisation. Previous work relevant to location privacy in these areas has been discussed and past solutions presented.

Chapter 4

Architecture

> *"...because the person has to give permission to be traced and the system is password protected, it's 100% safe!"*
> —TraceAMobile.com, 2003.

This chapter starts with a definition of the set of primitives which can be used to maintain location privacy. Section 4.2 describes four architectures for location-aware computing and provides an analysis of the privacy trade-offs inherent in each of the system designs. Section 4.3 describes some of the benefits of using anonymity over access control to protect location privacy and outlines why anonymisation is not sufficient for many location-aware applications. Section 4.4 describes a method of using dynamically changing pseudonyms to enable many location-aware applications to function and still protect user privacy through anonymity. Finally, Section 4.5 describes a threat model for using anonymisation to protect location privacy; this threat model is then used in the two security policy models presented in the subsequent two chapters.

4.1 Privacy primitives

There are a variety of factors which can be used to determine the level of location privacy available to users of a ubiquitous computing application. Some of these factors are used in traditional access control systems, others (such as restrictions by geographical area) are specialised to location-aware computing; a brief overview of the most pertinent factors are presented below.

Requester identity: the requester's identity may require *explicit authentication* where the identity given is globally unique and associated with a real human or limited company; alternatively *entity authentication* can be used where the identity given is not linkable to a real-world individual; examples of entity authentication include unlinkable pseudonyms and subject roles where the identity is known to be one of a set of individuals (but not which person within this set). Access to location data may be limited via the medium through which data are delivered and the machine the data are delivered to; for example, only serve data via a secure link to a particular PDA.

Requestee identity: the identity of the owner of the location data. The identity can be an explicit identifier and reveal the real-world identity represented by the location information;

alternatively, the identity may be a pseudonym, which allows communication between a location-aware application and the user, but prevents the application from linking the location data with a real-world entity.

Geographical area: access to location information can be restricted by the physical location of either the requester or requestee. In particular this restriction can be either absolute (*e.g.* "the requester can only see the requestee in the office") or relative (*e.g.* "the requestee can only see the requester when they are within 500 metres of each other"); access control based on the location of others requires some trust in the system providing their location information and therefore location information needs to be authenticated by a trusted party.

Time period: restrict the access to location information for particular time periods. For example, an office worker only allows his boss access to his location between 9am and 5pm.

Frequency of queries: only permit a limited number of queries per hour. This primitive may be useful in reducing the risk of pervasive tracking of all user movements. If the frequency of queries is restricted per requester identity, then collusion may take place between requesters to maximise the level of privacy invasion achievable. Alternatively, if the frequency of queries are globally restricted, then a denial-of-service attack is possible where a malicious requester deliberately issues the maximum number of location queries in order to prevent other potential requesters from accessing any location information.

Usage limitation: only permit access to data when interacting with particular services; for example, only reveal location information to the emergency services when calling the emergency number.

Historical access: information concerning past events can provide useful information to applications which wish to predict future movements of users. Unfortunately, uninhibited access provides the possibility of huge privacy invasion.

Reciprocity: Mutual exchange of location information and notification. For example, accessing location information of a colleague requires the release of the requester's location information as well (like the geographical area primitive, this methodology requires authenticated location information from a trusted party). An alternative reciprocal system may ensure notification to the requestee of an access request whenever a location query or event-based callback is made.

Spatial accuracy modification: Location resolution can be reduced to prevent intimate knowledge of the function or task at hand but still allow useful services; for example, accurate location of a user may allow others to infer the amount of time spent in the bathroom, whereas coarse location information only allows knowledge of "in the office". Fake location information may also be generated or location events in specific spatial areas may be removed.

Temporal accuracy modification: Temporal resolution can be reduced in a similar fashion to spatial accuracy above. The communication of location information may be deliberately delayed so as to reduce its value, or permit a user to prevent its publication even after the location event has occurred.

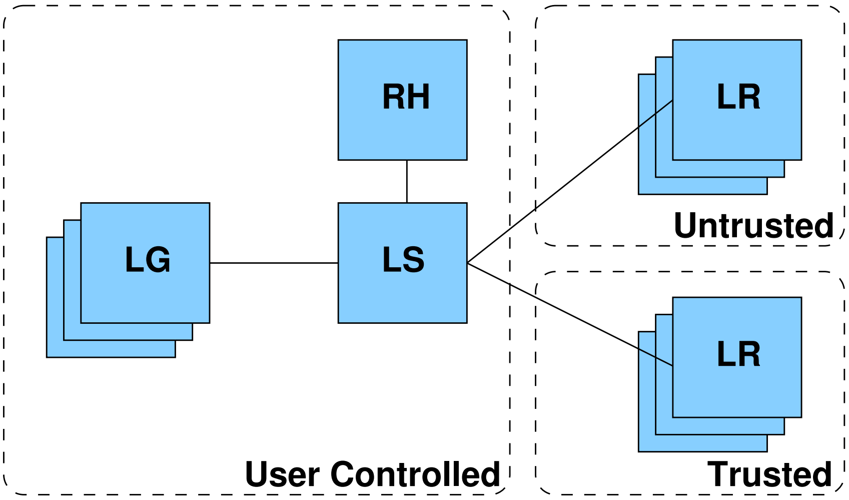


Figure 4.1: User-controlled model.

4.2 Architecture

Chapter 2 motivated the need for location-aware middleware, both to encourage code reuse and to save processor usage and network bandwidth by providing only the location information required for each application. The control, use and architectural design of the major components of the middleware can have a large impact on location privacy.

There are four main components in an architecture designed for location-aware computing. A *location generator* (LG) is a piece of hardware and related software which uses one of the physical properties of the environment (for example radio, acoustic or optical medium as described in Section 2.2) in order to produce location information represented by a co-ordinate, container or proximity event. A *location server* (LS) takes location information from a (possibly heterogeneous) set of location generators and performs transformations on the location data to convert location events between different location primitives (for example convert co-ordinates in one frame of reference into co-ordinates in another frame of reference, or translate from co-ordinates to containment events); in addition a location server provides access to location information to *location applications* (LAs) via either a query or event callback interface. Some architectures have a *privacy manager* or *rule holder* (RH) which contains information concerning the privacy preferences of the owners of the location events.

The manner in which each of these components are owned and trusted can affect the level of location privacy offered by the architecture to the users of the system. User ownership of all the components in a location architecture reduces the need to trust third-parties and implement (potentially complex) privacy restrictions. User ownership of all components of the location architecture is not always possible or desirable. The next four sub-sections outline four possible architectures in which a location component is either; (1) user-controlled; (2) trusted and run by a third-party; or (3) untrusted. Each of these architectural models are suited to different types of location-aware application and require different approaches to protect location privacy.

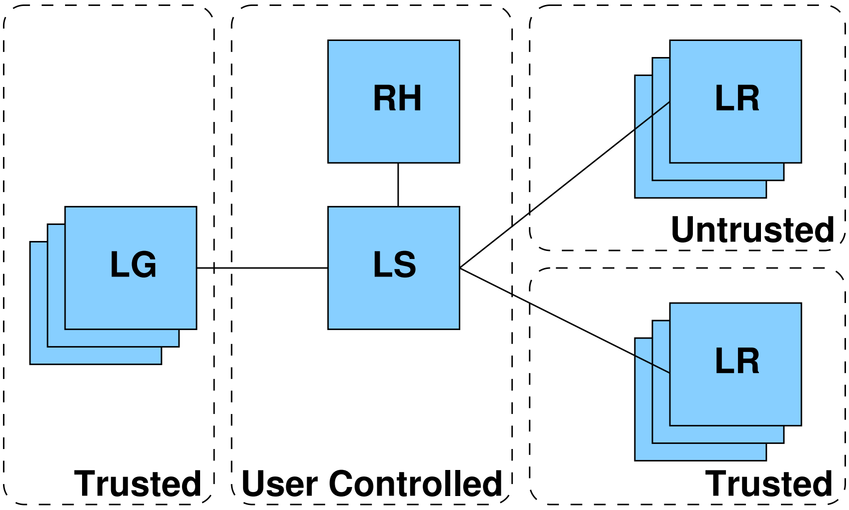


Figure 4.2: User-mediated model.

4.2.1 User-controlled model

The *user-controlled* model is depicted in Figure 4.1. In this model the location generators are inside-out location systems and therefore all location data initially reside solely on the tag (*e.g.* the Cricket system was designed to operate in this mode in order to minimise trust in third-parties). Since the user is in control of the tag, this places the user in direct control of any release of location information. In addition the location server and privacy manager are under the direct control of the user, and therefore the user can restrict access to location information to authenticated applications or (after anonymisation) to untrusted applications. In the case when location applications are also under user control, no trust is placed in third parties and therefore location privacy is assured if the software and hardware implementation is secure. Car navigation software driven from a local GPS receiver is one example of a user-controlled location service.

The user-controlled model is very suitable for personal location applications—the user is in direct control and, provided the hardware and software is well designed, nobody other than the user can possibly determine any personal location information. However this model has two major drawbacks: (1) users can (via applications) locate themselves quickly and efficiently, but determining their *context* (*e.g.* other users or objects nearby) requires the application to know about, be trusted with and contact every other user's location server (an overhead a centralised location server is designed to prevent); and (2) the model needs an inside-out location system and requires the user to carry around a device capable of deriving location information and executing the applications. The device may need significant resources if the device must store and process a lot of data. Therefore this model of location service is more suited to the field of wearable computing or augmented reality than ubiquitous computing.

4.2.2 User-mediated model

The *user-mediated* model is depicted in Figure 4.2; in this model the user does not control the location generators, which can therefore be inside-out or outside-in location systems, but

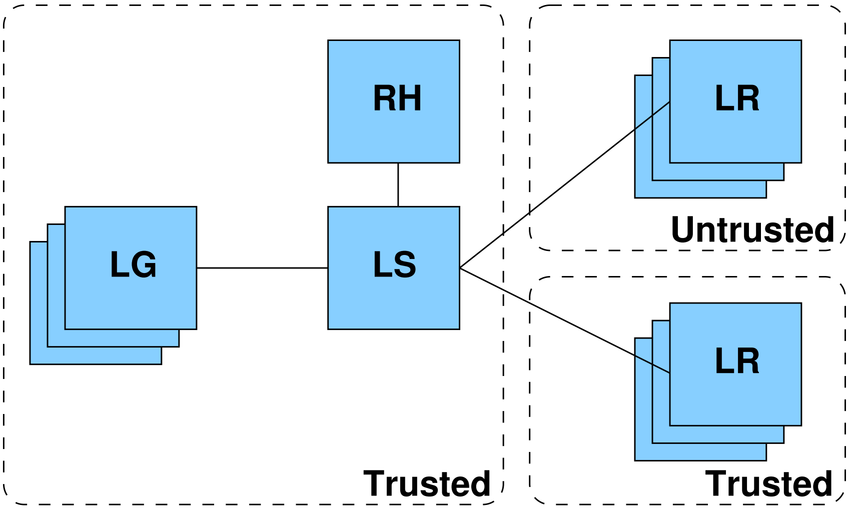


Figure 4.3: Third-party model.

instead the user owns and controls only the privacy manager and location server. The privacy manager and location server could reside on a roaming device carried with the user, or remain on a home PC connected to the Internet. The optimal position of the location server and privacy manager is dependant on the likely usage of the location information: a home PC is suited to location-aware applications used by people other than the owner of the location information (since its connectivity is likely to be better), whereas a PDA may be more convenient if location services are to be used by the roaming user.

This model does not support shared applications efficiently since a user's context cannot be determined easily. The model does have the advantage of enabling the user to reduce the amount of resources present on the device which must be transported with him; in case of a tagless outside-in system, where location information is stored on a home PC and the application actuators are placed in the environment, the user does not have to carry any device at all. If implicit addressing is used to transfer location information from the location generators to the location server (for example, by using the methods developed by Kesdogan *et al.* and Jackson and described in Section 3.4.2) then trust does not need to be placed in any particular mix node or location generator (at least one uncompromised mix node needs to be in operation in order to provide any level of anonymity).

4.2.3 Third-party model

The *third-party* model is depicted in Figure 4.3. All the main components of the architecture are controlled by one or more trusted entities on behalf of the user. The location server and privacy manager may be run by a collective of friends and family, an independent service provider (*e.g.* mobile phone operator) or by an employer. In this model shared applications can be built to use contextual information based on location information from one or more other users of the location server; personal applications continue to function efficiently. If the location generators are outside-in tagless systems and the application actuators are placed in the environment, the user does not need to carry any special hardware with them. As described in the user-mediated model above, implicit addressing can be used in order to reduce the amount of trust which must

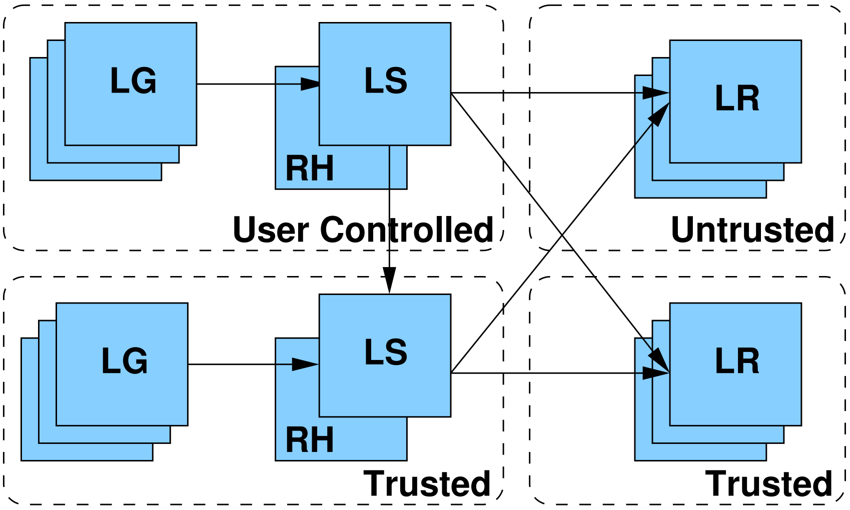


Figure 4.4: Hybrid model.

be placed in the location generators.

In this model the user must trust the service provider to correctly configure the required software and hardware for the privacy manager and location server. At the most fundamental level the user therefore relies on contract or privacy laws in order to ensure the correct level of location privacy is provided. Relying on legal protection ultimately requires a reliance on the government; in other words, if the government grant themselves (and possibly others) exemption from privacy legislation then the user may have to accept invasions in location privacy.

4.2.4 Hybrid model

In some situations a *hybrid* model is a likely outcome where different location technologies are combined in an iterative fashion to build a hierarchy of location servers; see Figure 4.4. For example, some mobile phone users will soon have handsets with built-in GPS and WiFi and in this case GPS and WiFi location services may act as inside-out location generators; the user can distribute the resulting location information via the data interface of the mobile phone either directly to their friends and family (by running a location server and privacy manager on the phone) or via their own user-controlled remote location server.

Irrespective of the distribution of location information determined from the GPS and WiFi location generators, the mobile operator can still determine (probably less accurate) location information from the phone using outside-in GSM location techniques (and feed this information into the mobile operator's location server). The user may also have the ability to provide a subset of location information from the (more accurate) GPS and WiFi location generators either directly to the mobile operator or via their own user-controlled location server. In this model the distribution of location information can be viewed as a multilevel secure system where location servers form a trust hierarchy and distribute location information adjusted by privacy primitives to applications or to other location servers with less security clearance.

4.3 Location privacy through anonymity

The last chapter discussed the two broad approaches available to protect information privacy: (1) access control, and (2) anonymisation. Access control has been the dominant mode of enabling location privacy for ubiquitous computing in the research present in the literature today. There are several reasons why access control has been suggested as a suitable mechanism for enabling location privacy; these include:

- Access control can be used to protect location privacy whatever the type of application. In contrast, anonymity techniques are not suitable for all location-aware applications since some location-based queries inherently require knowledge of underlying user identities. For example, providing the current location of "Alastair Beresford, born on the 22nd December 1977 in Loughborough, UK." will always require the location service to identify an individual. This form of location query is important, and must be enabled through access control.

- Location-aware computing is currently in its infancy and often locating a named individual *is* the application. Therefore the first requirement for ensuring privacy in such systems is to protect location-based queries, thus necessitating the development of access controls.

- A lot of research effort in computer science and engineering has been expended on access control methodologies (for example, consider the access matrix, role-based access control, policy, and digital certificate technologies discussed in the previous chapter). Anonymisation as a means of protecting communication or information privacy has so far attracted a much smaller research community.

If access control can be used to facilitate location privacy it is important to ask the question: why bother with anonymity? There are a combination of factors which make anonymity-based techniques more attractive than traditional access control methodologies. First, users may accept anonymised location-aware services but shy away from services which require concrete identity. Typical real-world examples include the use of cash as an anonymous payment method for sports betting and the acceptance of false details as the norm when visiting a sexual health clinic.

Second, configuration of access rights is a difficult task for a skilled computer scientist; the situation may be nearly impossible for an average user. Role-based access control and well configured defaults may mitigate the worst privacy worries, but these solutions will only partially solve the problem. A primary aim of ubiquitous computing is to make interaction with computational devices as easy as possible through the automation of tasks and minimisation of the cognitive load of human-computer interaction. Anonymity side-steps the configuration problem by denying an application access to any personally identifiable data, and therefore is more suited to ubiquitous computing applications than access control.

Third, if applications can be made to work with anonymous location sightings, then the applications themselves need not be trusted. Any reduction in the trusted computing base of a system increases our confidence in its correctness; in location-aware computing, removing applications from the trusted domain will increase our confidence in user privacy. Concerns about programmatic correctness alone (*e.g.* buffer overflow exploits) would justify this reduction in the trusted computing base since the vast majority of code will reside in the application layer.

(Making applications untrusted means that an attacker cannot use, say, a buffer overflow exploit in a location-aware application to determine the location of a particular person).

Our privacy worries extend beyond concerns about program correctness. As discussed in the Section 2.2.3, several mobile phone operators have already deployed systems which permit third-parties to connect to their networks and gain location information of users. Therefore trusted applications require trusted third-parties. This requires contractual agreements between mobile phone operators and third-party application providers. In addition, third-parties may have to adhere to restrictive legal regulations (*e.g.* EU Data Protection Directive) since they are handling personally identifiable information.

Distinct, separate identities are common in the real world. Humans are often said to have separate public and private lives; there is minimal interaction between our General Practitioner, employer and supermarket—people have a different sense of identity when interacting with each of these parties. Pseudonyms are a concept citizens are already familiar with, use and guard closely, and therefore a privacy-enabled system based on pseudonymity should be understandable and acceptable.

4.3.1 Some applications function anonymously

Whilst anonymisation is attractive from the perspective of the user, it can make writing location-aware applications harder. Some location-aware applications will function on anonymised location data (*i.e.* data with all direct identifiers removed). Example uses of anonymised location sightings include service planning (*e.g.* how many users of bus route 12 are there on an average Monday morning in winter), environmental discovery (*e.g.* Harle and Hopper detect the layout of the physical environment based on historical location event data [42]) and control of shared resources (*e.g.* automatically opening public doors).

The main characteristic of a truly anonymous application is the one-directional flow of data from user to application; information flow in the other direction is difficult because the application does not know which user to communicate with. One approach is to broadcast reverse path messages to all users, but this technique will not scale to large anonymity sets. Applications which interact with users via devices which have an inherent user identity associated with them cannot provide an identifier for return messages without sacrificing privacy; for example a mobile telephone or home server cannot provide a telephone number or IP address because these are considered direct identifiers under many threat models. Therefore many interactive applications require an *unlinkable* pseudonym in order to communicate with the user. For the rest of this dissertation all pseudonyms are designed to be unlinkable except where explicitly stated otherwise.

4.3.2 Pseudonyms required for user interaction

To demonstrate the need for pseudonyms in location-aware applications, a very simple location-aware application is introduced:

> **Coffee Shop Alert**
> *A computer scientist with a love of coffee configures his mobile phone to receive an alert containing the price and brand of coffee of any passing coffee shop. The application accepts a possibly empty set of configuration parameters to control*

when the alerts occur (e.g. only provide alerts for certain brands of coffee during work hours and then only if the coffee is cheap).

The coffee shop alert application requires a mechanism for contacting the underlying user in order to provide an alert concerning the presence of a coffee shop. Since the device (in this case the mobile phone) is strongly correlated with the user's identity, the user will effectively reveal his identity if he reveals his mobile phone number.

One solution to allow the coffee shop alert application to communicate with users requires a trusted component to replace any direct identifier with a pseudonym before forwarding location data to the application (and communication data in the other direction). In other words, the trusted component acts as an *anonymising proxy*, forwarding relevant location events to applications, and supporting data communication between user and application.[1] If the threat model anticipates an attacker being able to perform traffic analysis on the communication network, then a mix network is required.

4.3.3 Static pseudonyms do not protect privacy

Fine-grained location information of the type required for many location-aware services provides a very rich data set of user movement. Users often have one or more regions of space and time they predominantly occupy; such regions are called *home locations*. We define the term *simple home location* to refer to any home location which can be identified with a single location event (*i.e.* a single co-ordinate, containment or proximity measurement associated with a timestamp); conversely, a *complex home location* requires two or more location events to enable an attacker to identify an individual. Consider the following examples: the set of people sitting at a particular office desk for the majority of the working day; people found entering or exiting a family home at a particular time in the morning; the route (time, speed and direction) taken by someone to work.

Simply replacing user identity with a single pseudonym is not sufficient to protect privacy. If an attacker has unrestricted access to fine-grained location information, it is often possible to determine which underlying identity is represented by any given pseudonym since publicly available data can be used to correlate pseudonyms with underlying real-world identities. Large amounts of correlating data are available; examples include the electoral roll, company and university websites,[2] phone books and poorly protected databases connected directly to the Internet.[3]

A case study: the Active Bat system

To demonstrate that static pseudonyms do not protect privacy, two weeks of employee location sightings from the Bat System installed at AT&T Laboratories Cambridge Ltd were recorded. The raw data set contained just over 3 million location sightings and the data were filtered to

[1] In a third-party architecture, the location server can act as the anonymising proxy between applications and client devices.

[2] For example, several departments in the University of Cambridge provide public, web-based access to office location and phone numbers of staff and graduate students.

[3] Many web-based applications connect to large databases containing sensitive data. Scott and Sharp analyse some of the security threats created by connecting databases to the Internet via web services [118]. Even large vendors such as Oracle have been vulnerable in the recent past; for example Litchfield discovered multiple flaws in Oracle 9i [78], leading to CERT Advisory CA-2002-08.

remove location events from visitors and employees who were away for part of the measurement period, leaving just over 1.5 million location sightings. It is assumed that the attacker can collate the following information:

- list of company employees,
- access to pseudonymised location information,
- knowledge of the topology of the building, and
- position of researchers' desks.

The aim of the attacker is to correctly match each employee with his or her pseudonymised location information. While the location data set of most employee movements may present something slightly embarrassing (*e.g.* number of hours worked or visits to the toilet) it is conceivable a more personal matter could be uncovered by an interested third-party (*e.g.* the data could provide evidence of a late night love tryst between two employees[4]).

Harle and Hopper demonstrate that with raw location data, a fairly accurate topology of the building can be generated automatically [42]. At the time of the study, some employee office location information was publicly available on the company website, however relative desk positions within an office containing multiple employees was not; a knowledgeable insider or other external data (such as likely employee arrival and departure times from the office, or email address headers received from lab machines in known positions) are needed to resolve ambiguities between researchers sharing an office.

Measuring the total time each pseudonym spent at every floor-plan position reveals that most employees of a research lab spend the majority of time in a small contiguous area centred around their desk. These data can be extracted from the location information and correlated with known user desk positions to determine, with high certainty, the underlying user of any Bat identifier; Figure 4.5 provides one such example.

Many researchers have more than one location which is predominantly used by them; for example, researchers are sometimes allocated laboratory space as well as an office. Further evidence of user identity can be provided by correlating user pseudonyms against the set of spaces a user predominately occupies. Let $T_{i,l}$ define the number of hours during which employee Bat i is at sighted at location l. We mark position p as a home location of employee Bat e when ratio of time spent at p by e compared with all the other employees Bats is greater than a threshold value, τ; or equivalently, when Equation 4.1 holds. Figure 4.6 demonstrates this marking process for $0.5 < \tau < 1$.

$$\frac{T_{e,p}}{\sum_{j=1}^{j=n} T_{j,p}} > \tau \qquad (4.1)$$

All pseudonyms in this study were correctly correlated with their underlying user identities using these relatively simple techniques. While the user community is small, the ease with which correlation is possible demonstrates the potential pitfalls of using a single, fixed pseudonym to ensure user privacy.

[4] The likelihood of lovers wearing Bats in such circumstances is perhaps small, but a tagless outside-in system such as EasyLiving could be easier to forget about.

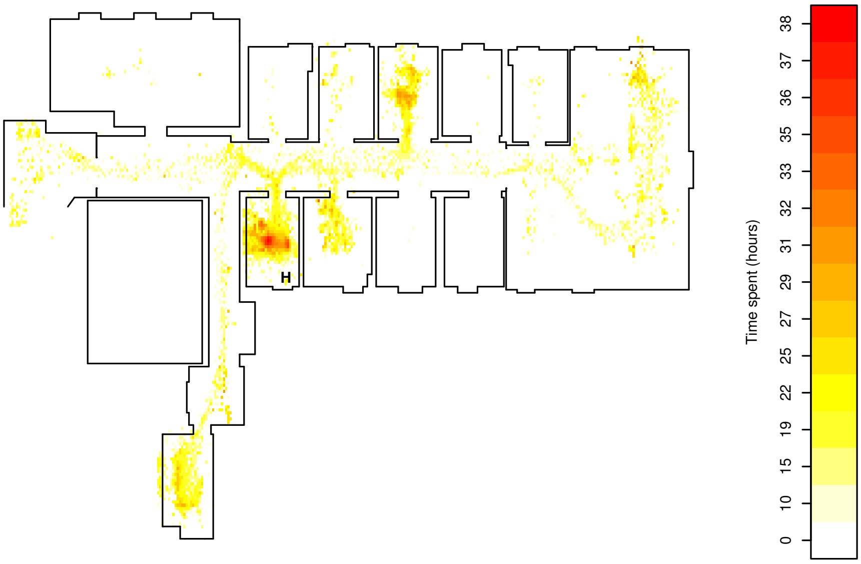


Figure 4.5: Knowledge of a user's primary desk location can be discovered by identifying the place where the Bat resides for the longest period of time; 'H' marks the user's home office.

4.3.4 Changing pseudonyms

Replacing user identity with a single, static pseudonym does not guarantee the user any level of anonymity (and therefore little privacy) if the location data associated with the pseudonym is of high temporal and spatial resolution. In this case, simply changing pseudonyms frequently may not be sufficient either: an attacker may be able to "follow the footsteps" of the underlying users simply by matching a disappearing pseudonym with the nearest (in both the spatial and temporal sense) newly appearing pseudonym.

Location-aware applications often require much less spatial and temporal resolution and coverage area than the underlying location system provides (this statement will be justified further in Section 4.4). For example the coffee shop alert application requires only positive containment notification in areas adjacent to known coffee shops. This, of course, is the exact reason *why* location service platforms provide a reduction in network bandwidth and processing requirements when compared with providing location information of all users directly to applications.

In order to prevent an attacker from correctly linking two pseudonyms to the same underlying user, the spatial and temporal resolution of the location data must be reduced in order to introduce confusion about the possible mappings of pseudonyms (*i.e.* prevent the linking of pseudonymous "footsteps"). Resolution reduction can be applied globally by reducing the spatial and temporal accuracy of all the location data provided by the location server to the location applications. For applications that require high levels of accuracy but reduced coverage area, it may be possible to constrain the resolution reduction methods to regions which are of little interest to the application.[5]

In order to guarantee user privacy through pseudonymity we need to limit the amount of

[5] As we shall see later, in most cases where applications can be constrained by coverage area and require high accuracy location information, location information outside the coverage area can be removed completely.

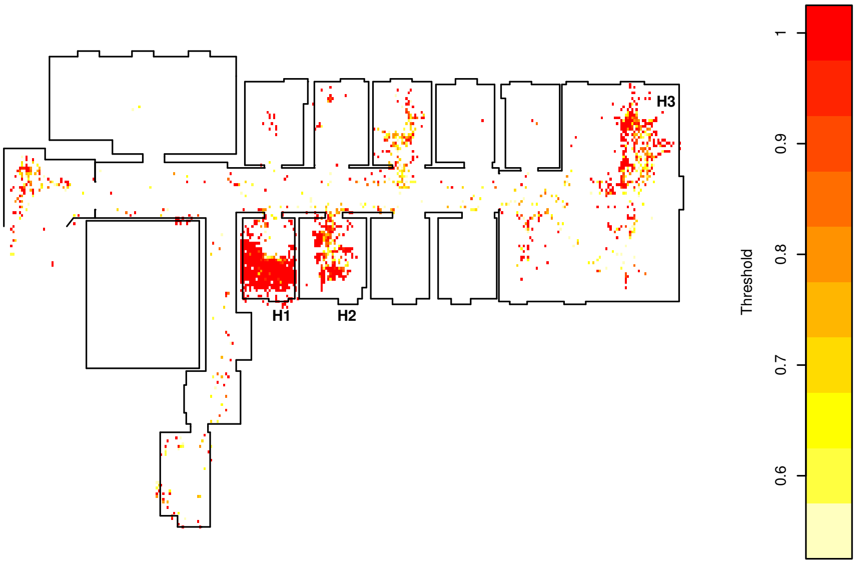


Figure 4.6: This particular researcher appears to have three home locations: office H1, office H2 and laboratory seat H3. Comparison against other employee thresholds rules out office H2, since other users uniquely occupy other parts of this office as well; this anomaly occurs because the regular user of office H2 was away during data analysis.

information presented to applications to ensure: (1) the information distributed with any given pseudonym is not sufficient to identify the underlying real-world entity with sufficient certainty, and (2) different pseudonyms cannot be combined together to form a single pseudonym with sufficient information to contravene (1). Provided these two statements hold, we can argue, at least from the Data Protection Directive viewpoint (see Section 3.2.2), that no personally identifiable information has been made available to the application and thus privacy is assured. Requirement (1) is perhaps a little too strict—it prevents *any* application from functioning in a home location. In some cases this is a necessary restriction since a user have many home locations and does not wish to reveal which one; in other cases the location of users can be inferred with high probability without a location system at all (consider the case of researchers at AT&T Laboratories Cambridge Ltd: users were very likely to be located at their desk during office hours). In this latter case, applications may be allowed to function at the home location, and instead applications should be prevented from determining the whereabouts of researchers when away from their desk.

4.4 Making applications work

It is important to assess which applications may continue to function with reduced temporal or spatial resolution, or with reduced coverage area. Applications may also need to be altered to cope with changing pseudonyms (traditionally most location applications assume a static user identity). In order to make this assessment, applications first described in Section 2.3 are revisited here to determine whether pseudonymity techniques are likely to be possible.

4.4.1 Managing changing pseudonyms

Some applications are *stateless* and are unaffected by a change of pseudonym for each location event transferred between the location server and the location application. Other applications require temporary or *session state* to identify a user for a short period of time in order to distinguish between separate location events from different people. Finally, some applications require *fixed state* to operate, requiring the same pseudonym to be provided on each occasion the user activates the application.[6]

Often applications can provide basic functionality in a stateless fashion, but require session or fixed state to provide more enhanced features. For example, the coffee shop alert application is stateless if the computer scientist is always alerted to the presence of the coffee shop (no matter what the brand, price or hour of the day). If alerts are delivered based on personal preference, either the mobile terminal must filter the alerts, or a fixed state pseudonym is required to permit the application to associate configuration parameters with a particular user.

Using a fixed state pseudonym with temporal and spatial resolution reduction alone may violate users' location privacy unless the level of reduction is quite severe. If even a single home location exists in the data set, an identity can be attached to the pseudonym, and any location privacy is therefore broken. Since it is often difficult to derive the exact threat model of an attacker (more on this in Section 4.5) it is difficult to assess the likelihood of the existence of a home location after global recoding. For these reasons, a fixed state pseudonym should be avoided with all but very aggressive data modification.

Conversely, for a restricted coverage area, a fixed state pseudonym can be less concerning, provided that the pseudonym remains unlinkable with any other pseudonym and the coverage area is a public location which is unlikely to possess any home location; the maximum privacy invasion is then limited to the coverage area required for the application to function. The mix zone model addresses these concerns in Chapter 5.

4.4.2 Configuration parameters

Other contextual data may need to be associated with temporary pseudonyms so that applications continue to function; the coffee shop alert configuration parameters provide a simple example. Such configuration parameters may last longer in duration than the temporary identifier itself; for example, tastes in coffee generally change at a much slower pace than the passing of coffee shops! If configuration parameters are retained for longer periods of time than pseudonyms, then configuration parameters must not uniquely identify individuals (*e.g.* by using techniques from SDC to anonymise the data, see Section 3.4.3).

A set of configuration parameters can become quite valuable to users (consider the configuration file for your favourite text editor as an example); deleting portions of this data or preventing updates in order to maintain privacy (as required by SDC) is not without cost to the user. Inference control can be applied to configuration parameters, provided a large enough data set is given; however application designers need to be aware of the possibilities of configuration modification.[7] Therefore, permitting client-side event filtering may be a more appropriate so-

[6] Therefore the pseudonym is an example of a *pure name* [87]: the pseudonym can only be used to test for equality with other pseudonyms and provides no further information. This is in contrast to the name `http://www-lce.eng.cam.ac.uk/` which, if it is valid, provides additional information as to the protocol used to retrieve data, a hint concerning the geographical location of the server and affiliation of the server owner.

[7] Application designers often attempt to "lock-in" their users by using custom file formats for user data and

lution (*e.g.* permit applications to install sand-boxed filter code on to the user's mobile phone); however this uses extra bandwidth and processing power of the user-controlled device.

4.4.3 A taxonomy of location applications

In order to assess how well location applications function with pseudonymised location data, categorisation of applications is required. The important criteria for assessing the likely performance of a location application with pseudonyms include:

concrete identity *vs.* pseudonym Some applications require concrete identities; examples include "Where is <person>"? Pseudonyms will simply not function for these type of applications. Other applications only require a method of communicating with the user or discriminating among several different users; the appropriate (stateless, session state or fixed state) pseudonyms are suitable for these applications.

event-driven *vs.* polled Some location applications poll the location server for location events whenever data is required (sometimes this includes requesting a stream of location events), other applications are event-driven and only receive location events when a containment or proximity event occurs.

personal *vs.* shared This criterion, first derived in Section 2.3, separates applications which only require location information of a single individual entity compared with applications which inherently need to locate many individuals. For personal location applications, a user-controlled location service model may be most appropriate to preserve location privacy. This is not always the case—an application may require access to extensive (and possibly dynamic) databases which are too large to replicate on mobile devices.

reduced coverage Some applications only require containment events or high-accuracy location information inside a small coverage area; see Chapter 5 for more detailed discussion.

reduced accuracy Some applications will function with reduced location accuracy. The scale of reduction is relative to the density of the population: for resolution reduction to be effective in preserving location privacy, then the detail of location information offered to an application must be limited to representing more than one possible user for any particular location event; see Chapter 6 for further information.

Table 4.1 provides an assessment of many of the applications first presented in Chapter 2 against the above criteria. For example, some applications will function with pseudonyms and reduced coverage in an event driven fashion. The ability of an application to function successfully with sufficient reduced accuracy to guarantee anonymity is dependent on the population density; for example, location services in a desert with an outside-in location system may require huge (and unacceptable) resolution reduction methods, whereas applications executed in urban environments may accept temporal and spatial reduction methods readily. Nevertheless there are many existing location-aware applications which are amenable to either coverage reduction of granularity reduction. Chapter 7 provides some real and simulated anonymity measurements to quantify the level of resolution reduction required for indoor and outdoor environments.

configuration information. Therefore allowing users to alter, or otherwise interact with application configuration data may be undesirable from the application writer's perspective.

Application	**Description**	**Identity** *vs.*	**Pseudonymity**	**Event-Driven** *vs.*	**Polled**	**Personal** *vs.*	**Shared**	**Reduced Accuracy**	**Reduced Coverage**
Fieldwork aid [96]	Augment notes on a field-trip	✓			✓	✓		✓	✗
PEPYS [90]	Automatic retrospective diary creation	✓		✓			✓	✗	✗
Where is <person>?	Locate a person (*e.g.* xab, smap)	✓			✓		✓	✗	✗
GUIDE [16]	Tour guide		✓	✓		✓		✓	✓
Reactive Room [17]	Intelligent audio-visual control		✓	✓		✓		✓	✓
Teleport [106]	Relocate desktop		✓	✓		✓		✗	✓
Stick-e note [12]	Personal reminders		✓	✓		✓		✓	✓
	Collaborative documents		✓	✓			✓	✓	✓
Driving Apps. [40]	Driving Conditions Monitoring		✓	✓			✓	✓	✗
	Road Hazard Detection		✓	✓			✓	✓	✗
Navigation [97]	Provide user directions		✓		✓	✓		✓	✗
Where is the nearest?	Find nearest printer, cash machine *etc.*		✓		✓	✓		✓	✗
How busy is room <X>?	Is the meeting room free?		✓		✓		✓	✓	✓
Office Layout [42]	Estimate layout of office furniture		✓		✓		✓	✓	✗

Table 4.1: A taxonomy of location applications: which applications are suitable for providing location privacy through pseudonymity? The criteria used in this table are described in detail in Section 4.4.3.

4.5 The location anonymity threat model

Traditional models of access control have typically empowered either a central authority (*e.g.* Bell-LaPadula) or users of the system (*e.g.* discretionary access control). Location privacy is about empowering the owners of location data with the ability to control the dissemination of their identity. One instructive example from the field of medicine is the British Medical Association (BMA) model [4, Chapter 8, p.166-172], which aims to protect private medical data from careless or corrupt staff. A threat model for location anonymity which is applicable for both resolution and coverage reduction pseudonymity techniques is discussed next.

The aim of anonymisation is to remove end-user applications from the trusted computing base. Therefore no assumptions can be made about the behaviour of location applications, which may collude, and therefore all applications must be viewed as one *global hostile observer*. Attackers can also own location objects in the location system, and therefore, in the case of third-party and hybrid location service models, multilateral security is required and precautions should be taken to protect individuals from each other.

The location service platform provides an architecture which encourages application programmers to request smaller coverage area(s) and lower temporal and spatial resolution than the underlying system can provide. The location server is ideally placed to coordinate the location privacy requirements of location event owners, and therefore the two security mechanisms presented here are executed from the perspective of the location server. Having said this, applications designed to function without any location privacy constraints may fail to work well with either access control or anonymity methods; instead programmers should take privacy constraints into account when writing programs (and strive for efficient operation with pseudonyms whenever possible). Chapter 7 analyses data from the Active Bat system to demonstrate the amount of resolution reduction required to meet particular anonymity constraints in an indoor office environment.

The aim of the attacker is to request location information from the location service in such a way as to recover the underlying identities associated with the location events. The aim of the location service is to prevent this from happening by producing a safe subset of location updates for a given set of location-aware applications.

In mix networks, dummy traffic is introduced to increase the amount of communication traffic flowing over the network and therefore improve the size of the anonymity set. Dummy location events can, in theory, be introduced in location-aware computing, however there are several drawbacks: (1) a series of location sightings are more complex in structure than encrypted dummy traffic and observant attackers may be able to differentiate between real and dummy movements (see Chapter 7 contains a more detailed discussion on the difficulty of generating accurate location data); (2) the cost associated with dummy traffic is fixed at the cost of communication, whereas location events can be used to purchase services, activate machinery or indicate resource usage and therefore can have both cost and correctness problems, for example:

- A “sentient” scanner receiving dummy location events will scan an empty scanner bed (producing a blank page scan which is easily detectable by an attacker), wear out the scanning mechanism more quickly and deny legitimate users the ability to control the device.

- The cost associated with using dummy users with a tourist guide application may be prohibitive if the guide service charges per hour, or per information request.

- A location application which monitors the status of a meeting room may incorrectly mark the room busy if dummy location events are included in the data feed.

4.6 Summary

Some applications will function with anonymised location data, but many more require pseudonyms to be associated with location information to enable interaction between location-aware applications and (anonymised) end users. Associating a single, static pseudonym with each user is not sufficient to guarantee anonymity when the location data is of relatively high spatial and temporal accuracy. This is because users have many simple and complex home locations within the data which allow an attacker to correlate publicly available information with location events and associate a concrete identity with a pseudonym. Different applications (and even different application functionalities) require different guarantees concerning the persistence of the association between pseudonyms and users; this chapter has outlined three useful variants, namely fixed, session and stateless pseudonyms.

A taxonomy of common location-aware applications was developed to demonstrate the suitability of this approach. This chapter identified two typical types of location-aware application: (1) applications which are only interested in one or more restricted coverage areas, and (2) applications which do not require accurate temporal *and* spatial accuracy but only one or the other (or neither). Methods to restrict coverage area and reduce the spatio-temporal accuracy of location data whilst preserving location privacy are described in the next two chapters.

Chapter 5

The mix zone model

> "*Every program and every user of the system should operate using the least set of privileges necessary to complete the job.*"
> —J.H. Saltzer and M.D. Schroeder, 1978. [110]

Chapter 4 described the principal benefits of anonymising location information over using access control to enable location privacy. The aim of the mix zone model is to prevent tracking of long-term user movements, but still permit coverage restricted location applications to function. This chapter describes how the mix zone model works, outlines an algorithm for calculating a quantifiable measure of anonymity (and therefore location privacy) and describes how the computational complexity associated with providing a quantitative measure of location privacy can be minimised. The chapter starts with a definition of the security policy for the mix zone model:

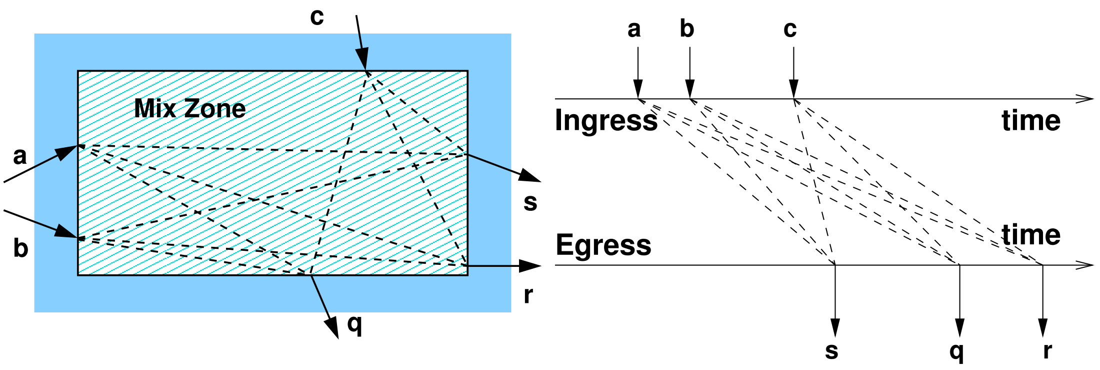
Security policy

1. *Application Registration:* Applications register interest in one or more coverage areas or *application zones*. The location server may refuse to register or modify the registration of an application zone.

2. *User Registration:* Users must register with the location server and inform the location server of any applications they wish to use.

3. *Location Service Guarantee:* The location server aims to provide the location application with all relevant location events inside the specified coverage area. The location server *must* reduce the coverage area to the level requested by the application; the location server *may* reduce the coverage area further than requested in order to protect the location privacy of its users. The location server will attach a pseudonym to each location event to enable communication between the user and the application.

4. *Quantitative Measurement:* The location server *may* provide a quantitative measure of the level of location privacy available to the user and allow the user to specify what actions to take when a (user defined) minimum level of location privacy is not met.



(a) plan view of the mix zone.

(b) Timeline of movements.

Figure 5.1: Example movement of three people through a simple mix zone. Who went where?

In order to provide location privacy, each user has one or more unregistered geographical regions where no application can trace user movements; such areas are called *mix zones*, because once a user enters such a zone, user identity is mixed with all other users in the mix zone. Since each user may register for a different set of location-aware applications, a mix zone is represented by the geographical regions not part of an application zone registered by one or more users. A *boundary line* is defined as the border between a mix zone and an application zone.[1]

Stateless pseudonymity applications receive a new (unused) pseudonym associated with every location event. Session state pseudonyms associate the same pseudonym with each location event for the duration of the visit to the application zone (a new pseudonym is associated with the same user for successive visits to the application zone). For a fixed state pseudonym, the user retains the same pseudonym on successive visits to the same application zone. The location server ensures a different pseudonym is associated with the same underlying user for every application zone the user has registered with.

The aim of the attacker is to link together pseudonyms and therefore track long-term user movements. An application observing a sequence of stateless pseudonyms may be able to "follow the footsteps" of the underlying user and link together pseudonyms, effectively converting stateless pseudonyms into session state pseudonyms. An application zone may contain a home location for one or more users and users may notice and refuse applications which attempt to register coverage areas near obvious (*e.g.* simple) home locations. Nevertheless, more subtle home locations may still persist and therefore an attacker may be able to associate an identity with a pseudonym for the duration of a visit to the application zone. A more concerning invasion of location privacy occurs when an attacker can link together movements between application zones since this increases the chance that an attacker can find a complex home location; furthermore, if a home location is found then the attacker is able to track user movements over a much larger area.

An *ingress* event occurs whenever a user enters a mix zone, and an *egress* event occurs whenever a user exits a mix zone. How well can an attacker correlate ingress movements with egress movements? In other words what measure should be used to quantify location privacy

[1] Location is a two-dimensional position in this dissertation, but a more complex model could be developed by moving to three dimensions and considering boundary surfaces rather than lines.

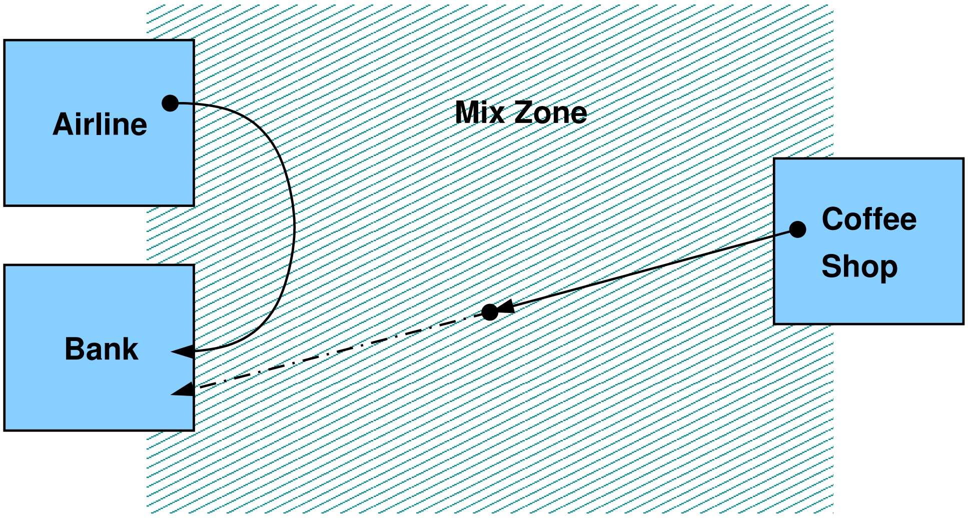


Figure 5.2: A sample mix zone arrangement with three application zones. The airline agency (A) is much closer to the Bank (B) than the Coffee Shop (C); users leaving A and C at the same time may be distinguishable on arrival at B.

in this model? Figure 5.1 provides a simple example scenario. Provided mix zones guarantee sufficient mixing, the invasion of location privacy is (at worst) restricted to the coverage area of the application zone.

5.1 Quantitative measure of anonymity

A strong analogy can be drawn between mix zones and mix nodes in a mix network: users, like messages arriving at a mix node, enter the mix zone, change identity, and then exit. Therefore the anonymity set (*i.e.* the number of users in the mix zone at a given time) appears to be a likely candidate for providing a measure of the level of mixing in the mix zone. Unfortunately the anonymity set alone is not a very robust measure; the anonymity set measure of a mix zone is likely to overestimate the level of anonymity.

Timing-based attacks Mix nodes can (in theory at least) wait for an indefinite period of time in order to collect enough messages to meet a minimum anonymity set size. Mix zones on the other hand must deliver location events in near real-time in order for many applications to be useful.[2] With large mix zones, there is a non-negligible minimum period of time taken by a user to move from one application zone to another application zone; an attacker can use this knowledge to subdivide an anonymity set in two: those ingress pseudonyms who could have entered the mix zone with enough time to egress and those ingress pseudonyms who could not. For example, Figure 5.2 provides a plan view of a single mix zone with three application zones around the edge: an airline agency (A), a bank (B) and a coffee shop (C). A is much closer to B than C, so if two users leave A and C at the same time and a user reaches B within a short period, an observer will suspect with high probability that the user emerging from B is not the one who entered the

[2] There are exceptions to this: location information can be useful as an off-line resource. For example, updating an environmental model concerning the whereabouts of office furniture.

mix zone at C; furthermore, if nobody else was in the mix zone at the time, the user can only be the one from A. The amount to which users are anonymised by the mix zone will therefore be smaller than one might believe by looking at the size of the anonymity set.

Location-based attacks The location of the ingress point of a pseudonym has an effect on the most probable egress point. For example, most people walking down a corridor or driving down a street continue in the same direction rather than perform a U-turn. This situation was analysed by Beresford and Stajano for a corridor at AT&T Labs Cambridge Ltd where the Active Bat system was installed [10]. User movements over a two week period were analysed and users were found to do a U-turn with probability 0.1%, or continue in a straight line with probability 99.9%.

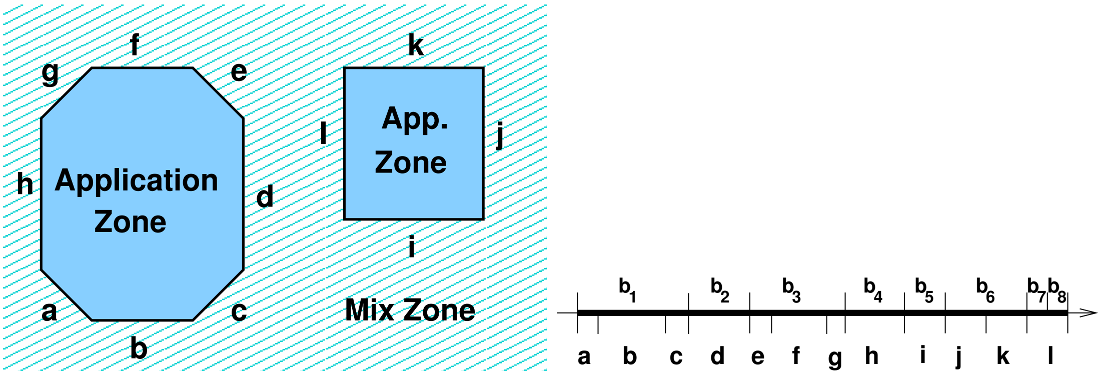
Intuitively, the larger the mix zone, the less the likelihood that ingress and egress points are strongly correlated (since there are likely to be many more paths through the mix zone, as well as more destinations of interest). Conversely, as the mix zone gets larger, timing-based attacks become more likely. A useful model is required to take both of these methods into account.

5.2 The mix zone security mechanism

An application zone is represented in the model as a closed polygon. The polygon has three or more straight boundary lines which separate the application zone from the mix zone. In order to model user movement, a method of naming the precise ingress and egress positions is required. It is important to record not just the ingress and egress boundary line, but also the crossing point on the line since a boundary line can be arbitrarily long and the movement model may change considerably along its length. A one-dimensional coordinate for the crossing point can be generated by arranging all the boundary lines lengthwise along the real axis. A precise crossing point for the spatial domain can then be recorded as a single real value. A *boundary section* is defined as a range between two real values; the range of each boundary section does not overlap with any other, and the combined range of all boundary sections is identical to the range of all boundary lines. Figure 5.3 provides an example transform between boundary lines and boundary sections.

The location server can calculate, and the attacker can estimate, a *movement matrix*[3] whose indexes are ingress boundary section i, egress boundary section e and time t to move from i to e (the actual time taken t' is quantised into t which represents the nearest multiple of the pre-defined update period τ). The cells of a movement matrix represent the frequency of users moving across the mix zone from i to e in t seconds. A mix zone may have more than one movement matrix associated with it to represent variations of movement patterns at different times of the day or week. For example, a pedestrianised city centre may have one movement model for shopping hours (when almost all people are walking) and one to model evening movements (when cars are permitted in the city centre).

[3] Strictly, a movement matrix is not actually a matrix but a 3^{rd} rank tensor (mathematically, the term matrix should be used to refer to two-dimensional data-structure). In general an nth rank tensor has n indexes and is a generalisation of a scalar (which has no indexes), a vector (which represents one dimensional data with a single index), and a matrix (which represents two-dimensional data with two indexes). A more accurate description would therefore be “movement tensor”.



(a) Plan view of two application zones.

(b) Projection of boundary lines and sections onto real line.

Figure 5.3: Example layout of two application zones and the projection of their boundary lines onto the real line. The lengths of the boundary lines are preserved in the projection. The eight boundary sections are defined by the ranges $b_1, \ldots, b_8$.

In general, making the movement matrix larger (*i.e.* reducing the size of the boundary sections or quanta of time τ) increases the accuracy of the movement model; however there are fundamental limits which prevent the movement matrix from becoming arbitrarily large:

Location system accuracy The time period τ cannot go below the update period of the underlying location system and the length of a discrete section cannot go below the spatial accuracy of the underlying location system.

Sampling accuracy Intuitively, the greater the number of samples for each cell in the movement matrix, the more accurate the estimate; therefore if the discretisation is too small, our estimates of user movement are likely to be inaccurate.[4]

Computational cost There is a computational cost associated with increasing the accuracy of the movement matrix; this is discussed further in the next section.

The attacker can observe the times, coordinates and pseudonyms of all these ingress and egress events and can use analytical movement models or historical data from nearby application zones to infer likely user movement across the mix zone. The location server can use historical location data of user movement across the *actual* mix zone in order to provide a more accurate estimate of the movement matrix than the attacker can generate. The attacker's goal is to reconstruct the correct mapping between all the ingress events and the egress events. This is equivalent to discovering the mapping between new and old pseudonyms. An efficient method to compute the most likely mapping and provide a measure of confidence in the best mapping is developed in the next section. A practical implementation and quantitative analysis of the level of mixing obtained in indoor environments is presented in Chapter 7.

[4] More precisely, the central limit theorem states that, as the size of a random sample n increases, the distribution of the sample mean $\bar{X}$ tends toward $N(\mu, \sigma/\sqrt{n})$, where μ and σ are the mean and variance of the underlying population. Therefore increasing the number of samples, n, reduces the variance on our random sample mean, $\bar{X}$, and therefore $\bar{X}$ becomes a more accurate predictor of the underlying population mean.

5.3 Computational complexity

This section describes how an attacker can use the movement matrix described in the last section to estimate the most likely mapping between ingress and egress pseudonyms. It then develops an information-theoretic measure of how certain the attacker can be in his guess. Calculating this measure of uncertainty turns out to be computationally intractable in the general case; therefore a method to calculate a lower-bound on the measure of uncertainty is developed.

In the last section it was observed that an attacker receives the times and coordinates of pseudonyms entering and exiting the mix zone and the aim of the attacker is to discover the correct mapping between ingress and egress pseudonyms. How many plausible mappings are there? During the period of observation, assume there are i ingress events and e egress events. The attacker observes i old pseudonyms going in, and e new pseudonyms coming out, often with some interleaving. Without loss of generality, assume $i \geq e$, then there is a total of $i^{\underline{e}} = i!/(i-e)! = i(i-1)\ldots(i-e+1)$ mappings. Many of the mappings can be ruled out because:

- a user cannot exit a mix zone before they enter it,
- users cannot move between two non-connected mix zones without passing though an application zone (and therefore being sighted), and
- portions of boundary lines containing walls or other impassable objects prevent users entering or exiting at these locations.

These temporal and spatial restrictions are represented in the movement matrix as zero value cells and therefore the movement matrix is likely to be sparse.

The mapping problem faced by the attacker can be viewed as a weighted bipartite graph $B = (X \cup Y, E)$, with ingress vertexes X and egress vertexes Y such that all edges $E = \{(x,y) | x \in X \wedge y \in Y\}$. A bipartite graph is *balanced* if $|X| = |Y|$, otherwise it is *unbalanced*. Weights are associated with each edge via a cost function $c(x,y)$ and represent the probability two distinct pseudonyms describe the same underlying person. The cost function can be estimated from the movement matrix by normalising the frequency count in each cell with the total number of movement sightings; edges with zero weight are removed from the graph. The *maximum-cost maximal match*[5] of this bipartite graph represents the most probable mapping of incoming pseudonyms to outgoing ones, assuming, of course, that the behaviour of any user is independent of any other, and the likelihood of their movement can be accurately represented by the movement matrix.

Galil surveys the past work [37] in finding maximal matchings of bipartite and general graphs, of which the Hopcroft and Karp algorithm [52] for finding a maximal matching in an unweighted graph is perhaps the most famous. Finding the maximal-cost perfect match of a weighted bipartite graph is an example of the *linear assignment problem*, in which n items (*e.g.* jobs) are assigned to n machines in an optimal way (*e.g.* the match which minimises the sum total cost of getting the jobs done). Burkard and Çela provide an extensive survey of the linear assignment problem and solutions presented in the literature [13].

[5] A *maximal match* in a bipartite graph occurs when the maximal number of vertexes are each connected by at most one edge; a match in which every vertex is connected by precisely one edge is called a *perfect match*. The maximal match with the highest summation of edge weights is the *maximum-cost maximal match*.

Kuhn developed the first well-known polynomial time algorithm to the linear assignment problem [70]. More recently, Jonker and Volgenant surveyed the existing solutions and presented the LAPJV algorithm [62], which has a uniformly lower computation time than the best implementation of other algorithms. A C implementation of this algorithm is adapted by the author and built into a Python-C module for use in this analysis. The LAPJV algorithm finds the minimum-cost perfect match. A maximal-cost perfect match can be found by negating the edge weights. Finding the maximum-cost maximal match when $m \neq n$ represents the case when more users have entered the mix zone than those who have exited. One method of adding the appropriate vertexes and edges to the graph in order to determine who is most likely to remain within the mix zone is outlined next.

5.3.1 Maximal matching in an unbalanced bipartite graph

First, some standard graph theory definitions. A *path* is a sequence of vertexes connected by an edge and an *elementary path* is a path which visits each vertex at most once. A *Hamiltonian path* is an elementary path which visits every vertex exactly once. An *elementary circuit* is an elementary path with the exception that it starts and ends at the same vertex and a *Hamiltonian circuit* is an elementary circuit which visits every vertex exactly once. A *match* in a bipartite graph is a set of edges where each vertex is connected to at most one edge.

An *augmenting path* is a path in a bipartite graph with respect to a match and contains an odd number of edges with the property that every even numbered edge is assigned in the match whereas every odd-numbered edge is unassigned in the match. By replacing the even edges of the augmenting path with the odd edges, the number of edges in the match is augmented by one. An *alternating path* in a bipartite graph with respect to a match is an elementary path with an even number of edges where every odd (or even) edge is assigned in the match and every even (odd) edge is not assigned in the match; swapping edges in the match with edges not in the match results in a new match of with the same number of edges. An *alternating circuit* is an alternating path with the exception that it starts and ends at the same vertex.

Let $G = (X \cup Y, E)$ be a bipartite graph and let the cost function $c(x, y)$ represent the cost of traversing the edge (x, y). We are interested in calculating the maximum-cost maximal match when $|X| \neq |Y|$; assume without loss of generality that $|X| > |Y|$. Let $Y' = \{v_1, \ldots, v_{|X|-|Y|}\}$ (*i.e.* create $|X| - |Y|$ new vertexes) and $G' = (X \cup (Y \cup Y'), E')$ where $E' = E \cup \{(x, y')|x \in X \wedge y' \in Y'\}$ and the cost function c' is defined as:

$$c'(x, y) = \begin{cases} 0 & if\, y \in Y' \\ c(x, y) & otherwise \end{cases}$$

In other words, G' is a balanced version of the unbalanced bipartite graph G with the necessary extra vertexes Y' connected to every vertex in X with zero weight edges.

THEOREM 1 *Let M represent the maximum-cost maximal match of G and M^+ represent the maximum-cost maximal match of G'. Let the function $cost(E) = \sum_{(x,y)\in E} c(x, y)$; then $cost(M) = cost(M^+)$.*

PROOF Let $M' = M^+ \setminus \{(x, y)|x \in X \wedge y' \in Y'\}$. If $M = M'$ then $cost(M) = cost(M^+)$ since $cost(M^+ \setminus M') = 0$. If $M \neq M'$ then consider the graph $D = (X \cup Y, M \,\Delta\, M')$;[6]

[6] $A \,\Delta\, B = (A \setminus B) \cup (B \setminus A)$ [symmetric difference]

$M \, \Delta \, M'$ must contain at least one edge. Each vertex in D can have a degree of at most two (one edge from M and one edge from M'). Therefore edges in $M \, \Delta \, M'$ form elementary paths of one or more edges. Along each path, edges are alternately members of M then M' (otherwise a vertex must have been assigned to two edges in either M or M'). Each path must be of even length (since otherwise the path is an augmenting path and one of the matches is not maximal). For all paths in $M \, \Delta \, M'$ the first edge must belong to M (or M') and even numbered edges, which are in M' (M), can be swapped with the odd edges which are in M (M') to form a new match. Any new match cannot cost more than the original match in M (M') since otherwise the original match was not a maximum-cost maximal match. If, starting from M, *all* paths in $M \, \Delta \, M'$ are swapped, we end up with match M' (because $M' = M \, \Delta \, (M \, \Delta \, M')$) and therefore $cost(M) \geq cost(M')$. Similarly, starting from M', and swapping all paths in $M \, \Delta \, M'$ ensures $cost(M') \geq cost(M)$. Therefore $cost(M) = cost(M') = cost(M^+)$, since $cost(M^+ \setminus M') = 0$.

■

Therefore an unbalanced bipartite graph can be transformed into balanced bipartite graph by adding the appropriate new vertexes and edges; THEOREM 1 guarantees that LAPJV will still return a maximum-cost maximal match.

5.3.2 Measure of confidence

The maximum-cost maximal match represents the most likely de-anonymisation of the underlying users passing through the mix zone. The value of the maximum-cost maximal match represents the absolute probability of users progressing through the mix zone in the manner described by the match. This value is not as useful as it might sound: instead the attacker needs a measure of *confidence* in the quality of this result. Consider an example mix zone event with three possible matchings $M = \{m_1, m_2, m_3\}$ with the following probabilities $\{\frac{1}{100}, \frac{1}{150}, \frac{1}{150}\}$ respectively. Knowing the most likely event $P(m_1) = \frac{1}{100}$ is not sufficient; what is really required is knowledge of how much more likely this match is when compared with the rest; *one* of these matches must have occurred because these are the only matches which explain this pattern of ingress and egress pseudonyms, at least according to the model. This conditional probability can be calculated as:

$$P(m_j|M) \overset{def}{=} \frac{P(m_j \wedge M)}{P(M)} = \frac{P(m_j)}{\sum_j P(m_j)} \tag{5.1}$$

in this case because $m_j \in M$.

The level of uncertainty in the set of possible matches $m_j \in M$ can then be measured by using Shannon's classic entropy measure [120]:

$$h = -\sum_j P(m_j|M) log P(m_j|M) \tag{5.2}$$

If every match is equally likely, entropy is maximal and the level of uncertainty is $\log i^{\underline{e}}$. The level of uncertainty for user movements across the mix zone is likely to be much smaller than this upper bound since the movement matrix is likely to be sparse for the reasons outlined at the beginning of this section.

The problem with this technique comes in calculating $P(m_j|M)$: the probabilities of all of the possible matches must be calculated, and this is not computationally tractable because there are $i^{\underline{e}}$ of them. Instead of calculating $P(M) = \sum_j P(m_j)$ directly, lazy evaluation can yield a lower bound $P_l(M) \leq P(M)$ by iterating through only maximal matchings in the bipartite graph (an efficient algorithm to do this is discussed in Section 5.3.3).

The number of maximal matches cannot be pre-computed, and there is no guaranteed polynomial bound on computation time before the last match is found (since there could be as many as $i^{\underline{e}}$ matches). Therefore the lazy evaluation of $P_l(M)$ is a lower bound; its value cannot diminish because every new maximal match adds one term to denominator $\sum_j P(m_j)$ and probabilities are always in the range $[0, 1]$. Each time a new match is found the entropy or level of anonymity offered by the mix zone can be recalculated and can only go up. Proof of the guaranteed increase in entropy is non-trivial and is therefore given in the Appendix. As lazy evaluation of $P_l(M)$ progresses one of three outcomes can occur:

- The level of anonymity in the mix zone rises to meet the minimum level specified by all the users as part of the security policy. The level of privacy requested by the user has been met and the algorithm is terminated.

- The lazy evaluation terminates (*i.e.* all possible matches have been found), so $P_l(M) = P(M)$. If the level of anonymity in the mix zone is still not sufficient, the identities of the users could be compromised by an attacker.

- Computation time runs out (*i.e.* computation has gone on as long as practicable), therefore $P_l(M) \leq P(M)$. If the level of anonymity offered by the mix zone is still not sufficient it is unknown whether a sufficient level of anonymity will ever be reached for this mix zone (but, given similar computing power, the attacker is uncertain of the quality of his guess as well).

5.3.3 Algorithm design and optimisation

This subsection describes an algorithm to perform the lazy evaluation of the level of anonymity available in a mix zone. This work is based on the work of Itai *et al.* who developed an algorithm for lazy evaluation of matches in an unweighted bipartite graph [57]. The algorithm is extended here to deal with weighted bipartite graphs representing movements across a mix zone. The algorithm developed uses a heuristic to maximise the measured level of anonymity in the mix zone given a fixed amount of computational power. All theorems and algorithms which are the work of others are attributed.

An algorithm is required to lazily iterate through all possible maximal matchings of a bipartite graph $B = (X \cup Y, E)$. A function to find the maximum-cost maximal match, $\mathit{find\_maxmatch}()$, can be implemented using the LAPJV algorithm introduced in Section 5.3. Given one match, subsequent new maximal matchings can be found by searching for alternating paths or alternating circuits in the graph with respect to the match.

LEMMA 1 *A perfect match M is unique if and only if there exists no alternating circuit. [by Itai et al.[57]]*

PROOF Let M and M' denote two distinct perfect matchings in a bipartite graph $B = (X \cup Y, E)$. Consider the graph $D = (X \cup Y, M \,\Delta\, M')$. Since $M \neq M'$, there must be at least one edge. Each vertex in D can have a positive degree of at most two (one edge from M and one edge from M'). There cannot exist a vertex v with positive degree one (since otherwise there is

Algorithm 5.1 Find a circuit in a directed graph [derived from Tarjan [130]]

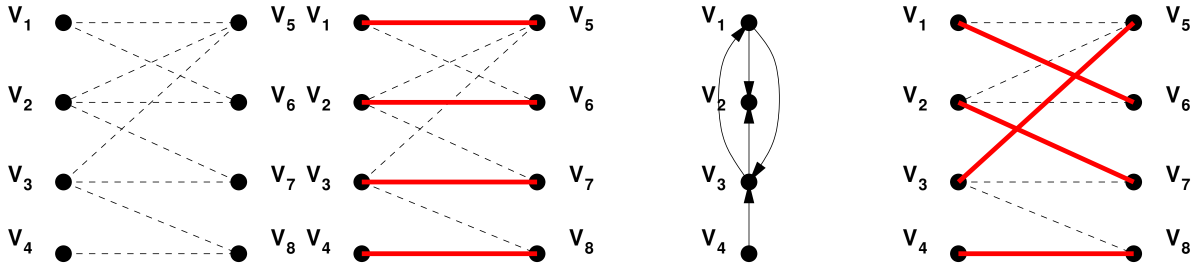
```
Require: $start$: set of vertexes on which exploration has begun
Require: $end$: set of vertexes on which exploration has ended
Require: $path$: path of vertexes from start vertex
proc $find\_ac(G, M)$
    $V, E \leftarrow build\_auxgraph(G, M)$
    for all $v \in V$ do
        if $v \notin start$ then
            $p \leftarrow visit(0, 0, v)$ //assume vertexes numbered from 1
            if $p \neq \varnothing$ then
                return $p$
    return $\varnothing$
proc $visit(c, v_1, v_2)$
    $start \leftarrow start \cup \{v_2\}$
    $path \leftarrow path \cup \{(c, v_1, v_2)\}$
    for all $\{v'|(v_2, v') \in E\}$ do
        $c' \leftarrow cost(v_2, v')$
        if $v' \notin start$ then
            $visit(c', v_2, v')$
        if $v' \notin finish$ then
            return $v', path \cup \{(c', v_2, v'\}$ //circuit in path edge list starting from $v'$
    $finish \leftarrow finish \cup \{v_2\}$
    $path \leftarrow path \setminus \{(c, v_1, v_2)\}$
    return $\varnothing$
```

only one edge in $M \bigtriangleup M'$ which is incident on v; let this be in M; then it does not exist in M', and therefore M' does not have an edge incident on v and it is not a perfect match.) Since every vertex of positive degree must be of degree two, $M \bigtriangleup M'$ must consist of one or more disjoint alternating circuits.

■

LEMMA 1 states that given a graph and a perfect match, another perfect match exists if and only if there exists an alternating circuit (no alternating paths can exist). An algorithm for finding alternating circuits in a graph and match can be simplified by constructing an auxiliary graph $A = (X, F)$ for a given bipartite graph $B = (X \cup Y, E)$ and perfect match M where $F = \{(x, x')|\exists y.(x, y) \in M \wedge (x', y) \in E \setminus M\}$. The resulting auxiliary graph is a directed graph where every elementary circuit in this graph represents an alternating circuit in the bipartite graph B with respect to the match M. A function $find\_ac()$ to find a circuit in the auxiliary graph (and therefore find an alternating circuit in the bipartite graph) is shown in Algorithm 5.1. (The algorithm shown keeps track of the edge weights, a detail which can be safely ignored for the moment; its use becomes important later.) The algorithm is based on a method developed by Tarjan [130] to find all *strongly connected components*[7] in a directed graph. The algorithm

[7] A strongly connected component is a subset of vertexes in a directed graph with the property that from any one vertex in the subset, all other nodes can be reached by traversing one or more edges. Tarjan later extended the strongly connected components algorithm to determine all elementary circuits in a directed graph [131]. More re-



(a) Graph (b) Perfect match (c) Aux. graph (d) Perfect match

Figure 5.4: Figures (b) and (d) show two possible perfect matchings for the bipartite graph shown in (a). Note the existence of an alternating circuit between v_1, v_5, v_3, v_7, v_2 and v_6 which can also be found as a circuit in the directed auxiliary graph derived from (b) and shown in (c).

presented here is adapted to return the first circuit found in the graph, rather than using the existence of circuits to determine all the strongly connected components.

An example bipartite graph and a perfect match are shown in Figures 5.4(a) and (b) respectively. Figure 5.4(c) shows the auxiliary graph derived from the bipartite graph and perfect match; a circuit exists in the auxiliary graph and this can be used to derive another matching which is shown in Figure 5.4(d). Itai *et al.* developed a method of finding alternating circuits in a graph with respect to a perfect match in an organised fashion in order to lazily iterate through all the possible perfect matchings in a bipartite graph [57]. The function $next\_ac()$ in Algorithm 5.2 performs this operation. It takes as input a bipartite graph B, a match M and an alternating circuit C; the initial value of R is the empty set $\varnothing$. First the algorithm returns the next available match by applying the alternating circuit C to the match M to yield a new match. The algorithm then selects an arbitrary edge which exists in both the match M and the alternating circuit C and explores two possibilities (by recursive function call): (1) the selected edge is not present in any more matches and is excluded from the graph, and (2) the selected edge is included in all future matches (and is passed as a required edge in the variable R). Itai *et al.* provide a proof of correctness for this approach.

If a matching M is maximal but not perfect, then there exists at least one exposed vertex z where no $m \in M$ is incident upon it and therefore has degree zero. Let (z, y) be an edge in the bipartite graph B. There must exist an edge (x, y) in M (since M is a maximal match); such a pair of edges is called the M-transposition path (x, y, z). Given a match and an M-transposition (x, y, z), a new match can be generated as $(M \cup \{(z, y)\}) \setminus \{(x, y)\}$.

Lemma 2 *A maximum match M is unique iff there exists no alternating circuit and no M-transposition. [by Itai et al.[57]]*

Proof If there exists no M-transposition, then Lemma *1* proves a maximal match M is unique iff there exists no alternating circuit. (If an M-transposition (x, y, z) exists, then an alternate maximal match $(M \cup \{(z, y)\}) \setminus \{(x, y)\}$ exists.)

■

cently Nuutila and Soisalon-Soininen improved Tarjan's algorithm to handle sparse graphs and trivial components (containing only one node) in an efficient manner [91].

Algorithm 5.2 Find another match & search for more acs [by Itai *et al.*[57]]

```
proc $next\_ac(B, M, C, R)$
    $M_{next} \leftarrow M \,\Delta\, C$
    output $M_{next} \cup R$
    $(x, y) \in M \cap C$ //select an arbitrary set member
    $B_{ex} \leftarrow B \setminus \{(x, y)\}$
    $C_{ex} \leftarrow find\_ac(B_{ex}, M_{next})$ //returns $\varnothing$ if there are no alt-cs left
    $M_{in} \leftarrow M \setminus \{(x, y)\}$
    $B_{in} \leftarrow B_{ex} - \{x, y\}$ //remove vertexes $x$ and $y$ from the graph
    $C_{in} \leftarrow find\_ac(B_{in}, M_{in})$
    if $C_{ex} \neq \varnothing$ then
        $next\_ac(B_{ex}, M_{next}, C_{ex}, R)$
    if $C_{in} \neq \varnothing$ then
        $next\_ac(B_{in}, M_{in}, C_{in}, R \cup \{(x, y)\})$
```

Algorithm 5.3 Find maximum-cost M-transposition

```
Require: $B = (X \cup Y, E)$
proc $find\_mtrans(X, Y, E, M)$
    $mtrans \leftarrow \varnothing$
    $c \leftarrow \infty$
    for all $x \in X$ do
        if $unmatched(x)$ then
            for all $y \in \{u|(x, u) \in E\}$ do
                $z \leftarrow v|(v, y) \in M$ //since $M$ is a match, only one $v$ exists
                $c_{new} \leftarrow cost(x, y) - cost(z, y)$
                if $c_{new} < c$ then
                    $c \leftarrow c_{new}$
                    $mtrans \leftarrow (x, y, z)$
    return $mtrans$
```

Algorithm 5.4 Iterate through all maximal matches [by Itai *et al.* [57]]

```
proc $All\_Solutions(B)$
    $M \leftarrow find\_maxmatch(B)$ //using LAPJV
    output $M$
    if $(T \leftarrow find\_mtrans(B, M)) \neq \varnothing$ then
        $next\_mtrans(B, M, T, \varnothing)$ //using Algorithm 5.5
    else
        if $(C \leftarrow find\_ac(B, M)) \neq \varnothing$ then
            $next\_ac(B, M, C, \varnothing)$ //using Algorithm 5.2
```

Using LEMMA 2 and given a maximal match M in the bipartite graph B another match can be found by searching for an alternating circuit C as before with the perfect match case (*i.e.* by finding an elementary circuit in the auxiliary directed graph). In addition a new match can also be found by searching for an M-transposition (x, y, z); the algorithm to do this is shown in Algorithm 5.3 and works by iterating through all exposed vertexes $x \in X$. (The algorithm actually finds the maximum-cost M-transposition, a detail which becomes important later). If the algorithm finds an M-transposition, then a new match can then be generated as $(M \cup \{(z, y)\}) \setminus \{(x, y)\}$.

If no alternating circuit or M-transposition is found, then the match is unique. Itai *et al.* presented an algorithm for iterating through all maximal matches in an unweighted bipartite graph one at a time by first lazily iterating through all M-transpositions of a match with respect to a graph and when no more M-transpositions exist search for any remaining alternating circuits [57].

Algorithm 5.4 describes the overall approach and relies on two sub-routines $next\_mtrans()$ and $next\_ac()$. The function $next\_mtrans()$ is shown in Algorithm 5.5 and takes as input the bipartite graph B, match M, an M-transposition (as calculated by $find\_mtrans()$) and a variable R initially set to $\varnothing$. The algorithm first returns the new match made possible with the M-transposition, and then explores two possibilities (by recursive function call): (1) the edge (x, y) is not present in any more matches and is excluded from the graph, and (2) the edge (x, y) is included in all future matches (and is passed as a required edge in the variable R).

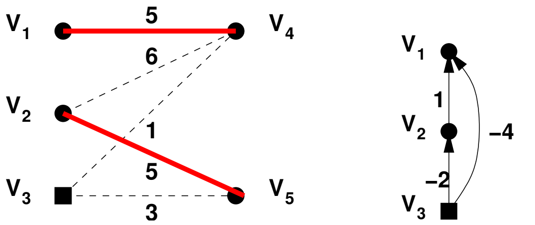
Maximum-cost maximal matches

The lazy evaluation of matches developed by Itai *et al.* was designed to function on unweighted bipartite graphs, and, as such, can work unmodified on weighted graphs. Recall that edge weights in the bipartite graph $B = (X \cup Y, E)$ correspond to the probability that the ingress pseudonym (a vertex in X) and egress pseudonym (a vertex in Y) represent the same underlying user. Ideally the iteration through matches in a weighted graph would proceed in order of match cost from the maximum-cost maximal match (corresponding to the most likely mapping between ingress pseudonyms and egress pseudonyms) to the minimum-cost maximal match (the least likely mapping between ingress pseudonyms and egress pseudonyms). This ordering is desirable in order to ensure that, given a fixed computation time, our estimate of $P_l(M)$ is maximised (since there could be as many as $i^{\underline{e}}$ mappings there may not be time to compute all of them). To achieve this goal, algorithms for $find\_maxmatch()$, $find\_mtrans()$ and $find\_ac()$ must be defined to recursively select the next best match.

The maximum-cost maximal match can be found with the LAPJV algorithm as discussed

Algorithm 5.5 Find another match and search for more M-Transpositions

```
proc $next\_mtrans(B, M, (x, y, z), R)$
  $M_{next} \leftarrow M \cup \{(z, y)\} \setminus \{(x, y)\}$
  output $M_{next} \cup R$
  $B_{ex} \leftarrow B \setminus \{(x, y)\}$
  $T_{ex} \leftarrow find\_mtrans(B_{ex}, M_{next})$ //returns $\varnothing$ if there are no $M$-trans left
  $M_{in} \leftarrow M \setminus \{(x, y)\}$
  $B_{in} \leftarrow B - \{x, y\}$ //remove vertexes $x$ and $y$ from the graph
  $T_{in} \leftarrow find\_mtrans(B_{in}, M_{in})$
  if $T_{ex} \neq \varnothing$ then
    $next\_mtrans(B_{ex}, M_{next}, T_{ex}, R)$
  else
    if $C_{ex} \leftarrow find\_ac(B_{ex}, M_{next})$ then
      $next\_ac(B_{ex}, M_{next}, C_{ex}, R)$
  if $T_{in} \neq \varnothing$ then
    $next\_mtrans(B_{in}, M_{in}, T_{in}, R \cup \{(x, y)\})$
  else
    if $C_{in} \leftarrow find\_ac(B_{in}, M_{in})$ then
      $next\_ac(B_{in}, M_{in}, C_{in}, R)$
```



(a) Maximum-cost maximal match

(b) Aux. graph

Figure 5.5: Figure (a) provides an example of a maximum-cost maximal match; two M-transpositions are available: (v_1, v_4, v_3) and (v_2, v_5, v_3). Neither of these M-transpositions leads to the next highest-cost maximal match, which is $\{(v_2, v_4), (v_3, v_5)\}$. This fact can be confirmed by searching for the maximal-cost path starting from an exposed vertex (in this example v_3) in the auxiliary graph depicted in Figure (b).

earlier. Unfortunately strict ordering of matches from maximal- to minimal-cost within the function $\mathit{find\_mtrans}()$ cannot be achieved; Figure 5.5(a) provides a simple counter example. In the counter example, two M-transpositions are required to get to the second most costly maximal match. The M-transpositions available for a given bipartite graph $B = (X \cup Y, E)$ and a match M can be displayed in an auxiliary graph $D = (X, F)$ where $F = \{(x, x') | \exists y.(x, y) \in E \setminus M \wedge (x', y) \in M\}$ and the cost function $c(x, x') = c(x', y) - c(x, y)$. Figure 5.5(b) provides an example auxiliary graph for the bipartite graph and match shown in Figure 5.5(a). Using the auxiliary graph, the next best combination of M-transpositions can be found by searching for the maximum-cost path which starts from any exposed vertex.

THEOREM 2 *Finding the maximum-cost path in a weighted graph is NP-Complete.*[8]

PROOF Determining the existence of a Hamiltonian path in a graph is NP-Complete.[9] Let $G = (V, E)$ represent an instance of the Hamiltonian path problem. An instance of the maximum-cost path problem can be constructed as follows: let $G' = (V, E)$ and assign a weight of one to each edge in E. If the maximum-cost path has cost $c = |V| - 1$ then we have found a Hamiltonian path in G; otherwise $c < |V| - 1$ and there exists no Hamiltonian path in G.

■

Therefore it is not computationally tractable to iterate in order of cost from the maximal-cost M-transposition to the minimum-cost M-transposition in the bipartite graph. A similar problem occurs when attempting to find the alternating circuit which returns the next best maximum-cost maximal match. The alternating circuit which provides the next best-cost match is represented by the maximum-cost circuit in the auxiliary graph A described in the last Section and used to simplify the search procedure used by $\mathit{find\_ac}()$ in Algorithm 5.1. Calculating the maximum-cost circuit in a weighted graph is also NP-Complete[10] (this immediately follows from THEOREM 2).

Given that it is not computationally tractable to iterate from the maximum-cost maximal match to the minimum-cost maximal match in strict cost order a good heuristic is required. One solution to this problem is to replace the recursive function calls at lines 11 and 13 in Algorithm 5.2 and lines 10, 13, 15 and 18 in Algorithm 5.5 with insertion operations onto a priority queue. The queue is ordered by the cost of next match made available by the M-transposition or alternating circuit which was found (if no further M-transposition or alternating circuit is found, no element is added to the queue). The head of the queue (containing the largest-cost maximal match) is explored next (and any new M-transpositions or alternating circuits it generates are inserted back into the queue); this is an example of a greedy algorithm since the queue ensures the next best candidate match available is explored.

[8] Thanks go to Anuj Dawar for providing the key insight in this proof.

[9] Cormen *et al.* prove that determining the existence of a Hamiltonian circuit in a graph is NP-Complete [18]. Let $G = (V, E)$ represent an instance of the Hamiltonian circuit problem. An instance of the Hamiltonian path problem can be constructed as follows: let v represent an arbitrary vertex $v \in V$ and define $G' = ((V \setminus v) \cup \{v_1, v_2\}, E')$; for every $(u, v) \in E$ then $(u, v_1) \in E'$ and $(u, v_2) \in E'$ (similarly for $(v, u) \in E$) and if $(i, j) \in E$ and $i \neq v$ and $j \neq v$ then $(i, j) \in E'$. A Hamiltonian path exists in G' iff there is a Hamiltonian circuit in G. Therefore determining the existence of a Hamiltonian path in a graph is NP-Complete.

[10] Interestingly, the cost of the *mean* maximum-cost circuit, (in other words the circuit which, on average, costs the most per edge node in the circuit) can be computed in polynomial time. Karp describes the first polynomial time solution to this problem [63]; Dasdan and Gupta provide a recent survey of work in this field and present some incremental improvements [22]. Unfortunately, the mean maximum-cost circuit solution is not applicable to the problem at hand since the mean maximum-cost circuit does not necessarily equate to the maximum-cost circuit.

	n	s	e	w
n	$\frac{1}{32}$	$\frac{3}{64}$	$\frac{1}{64}$	$\frac{1}{32}$
s	$\frac{3}{64}$	$\frac{1}{64}$	$\frac{1}{32}$	$\frac{1}{32}$
e	$\frac{1}{64}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{3}{64}$
w	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{3}{64}$	$\frac{1}{32}$

(a) $t = \tau + 1$

	n	s	e	w
n	$\frac{1}{64}$	$\frac{3}{64}$	$\frac{1}{32}$	$\frac{1}{32}$
s	$\frac{3}{64}$	$\frac{1}{64}$	$\frac{1}{32}$	$\frac{1}{32}$
e	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{3}{64}$
w	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{3}{64}$	$\frac{1}{32}$

(b) $t = \tau + 2$

Table 5.1: Movement matrix for $\tau + 1$ and $\tau + 2$.

Since this algorithm is only a heuristic, there will be cases where this results in a non-optimal ordering of matches, however in many cases the order matches are discovered in are close to optimal. Providing a definitive assessment of the performance of a heuristic is often difficult. The performance of the heuristic was assessed on 10000 bipartite graphs with a uniform distribution of the number of vertexes in the range $5 \leq |X| \leq 10$ and $|X| \leq |Y| \leq 10$. Edge weights are selected to have a bimodal distribution in one of two ranges (1) range (1000,10000) is selected with probability 0.1, and (2) range (1,100) is selected with probability 0.9.

A bimodal distribution of edge weights increases the disparity between the optimal and sub-optimal ordering of matchings; bimodal or multi-modal distributions are likely to exist in the dataset since the are often strong correlations between ingress and egress positions in the mix zone. The worst heuristic performance found during the test is shown in Figure 5.6; the performance of all the test cases is shown in Figure 5.7. Overall the priority queue performs well, selecting near optimal ordering of matchings in the majority of cases.

Using a queue to prioritise the search transforms the algorithm from a depth-first search to a breadth-first search. The memory requirements for the breath-first algorithm are higher than the depth first-search since the depth of the search tree is at most $O(|E|)$ whereas in the worst case the breadth of the tree grows linearly with the number of iterations of the algorithm. Since it is anticipated the algorithm will run out of computation time in the case when there are $i^{\underline{e}}$ mappings this limitation does not constitute a major problem.

5.4 Real-time anonymity measurements

Interestingly, the level of anonymity received by a group of users moving through the mix zone can change even *after* some users have left the mix zone. Only when every user has left is the exact value of the level of anonymity in the mix zone known.

To illustrate this, consider the following simple scenario: a mix zone with four boundary lines, north (n), south (s), east (e) and west (w) is illustrated in Figure 5.8. To simplify the example, boundary lines are quantised coarsely so that each boundary line is represented by a single boundary section in the movement matrix; furthermore any user entering the mix zone is guaranteed to have left after either one or two time periods. A movement matrix for this example is given in Table 5.1.

Consider the movements of two users u_1 and u_2 who enter the mix zone at the same time τ, one from n and one from e respectively. If u_1 exits at time $\tau + 1$ through s and u_2 remains in

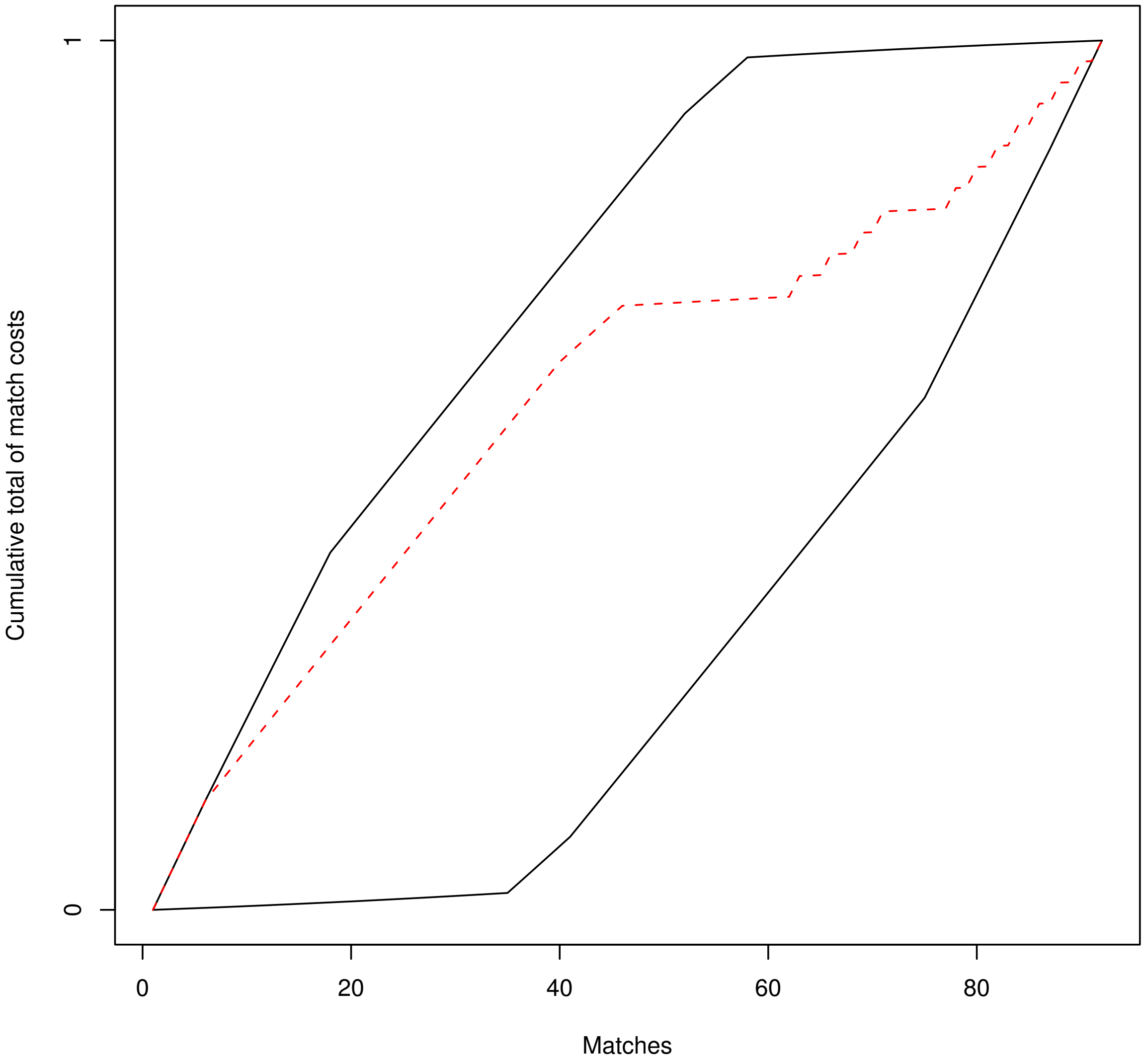


Figure 5.6: The worst performance of the heuristic in all of the runs of the heuristic algorithm; the solid black lines indicate the best and worst possible ordering of searches for matchings and the dashed coloured line indicates the order of matchings chosen by the heuristic.

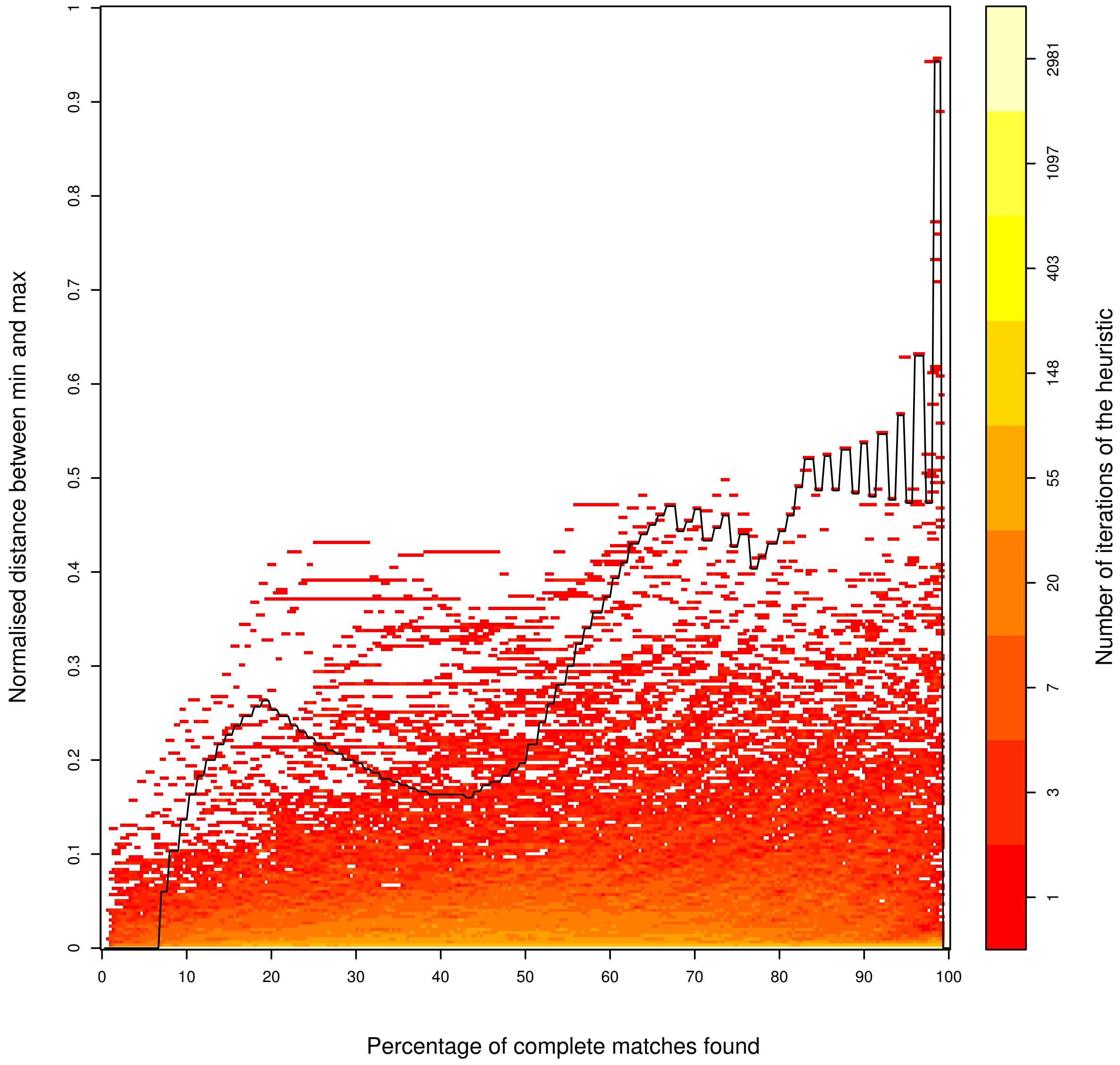


Figure 5.7: Results of all the tests, normalised in both the horizontal axis (percentage of matches completed) and vertical axis (indicating the relative distance between the best and worst ordering of matches); colour intensity indicates the number of runs of the heuristic which produced a match for that point in the graph and the black line shows the worst performance of the heuristic in all of the runs of the algorithm.

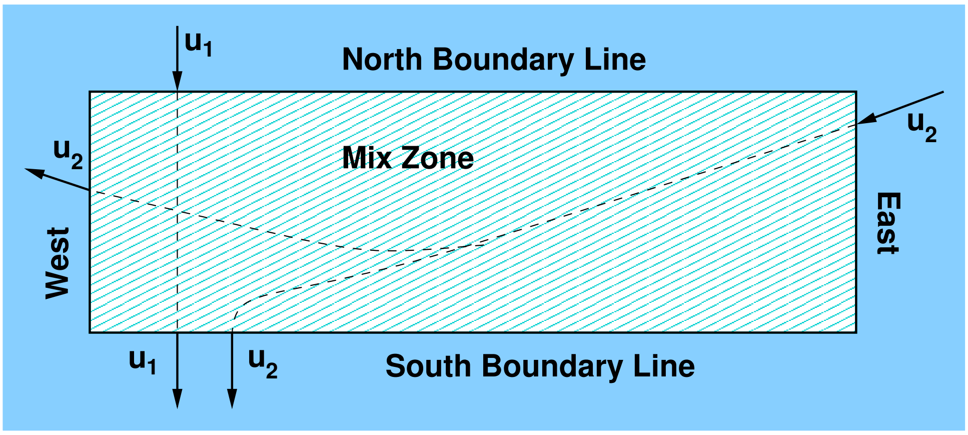


Figure 5.8: Entropy of the mix zone is dependent on the exit taken by u_2.

the mix zone, what level of mixing has occurred?

The measure of uncertainty is dependent on the exit taken by u_2 in the following time period $\tau + 2$. If u_2 goes west, the possible matches are $\{n \rightarrow s, e \rightarrow w\}$ and $\{e \rightarrow s, n \rightarrow w\}$. Of these, according to the probabilities encoded in the movement matrix, the first is much more likely. If, on the other hand u_2 goes south, then the possible matches become $\{n \rightarrow s, e \rightarrow s\}$ and $\{e \rightarrow s, n \rightarrow s\}$, whose probabilities are identical. So, when u_2 goes south the attacker is much less certain about what happened and u_1 is much more anonymous (1 bit) than if u_2 had exited through west (0.47 bits).

Given a movement matrix for a mix zone, the location server can calculate a lower bound on the level of mixing for a particular user u (who is still in the mix zone) by assuming all the other current users leave the mix zone in the most probable manner. A lower bound on the level of mixing u experiences can then be calculated for each possible exit from the mix zone.

5.5 Individual anonymity

The entropy measure used up until now calculates the level of uncertainty present in the attacker determining the *complete* mapping between ingress pseudonyms and egress pseudonyms. If an attacker is interested in tracking the movement of a particular user or subset of users, the entropy measure described in this chapter so far provides an over-estimate in the level uncertainty. For example, consider users $u_1, \ldots, u_n$ whose occupancy in the mix zone overlap temporally. There may be many mappings between ingress pseudonyms $i_1, \ldots, i_n$ and egress pseudonyms $e_1, \ldots, e_m$, and yet every mapping connects pseudonym i_1 to pseudonym e_1. In this scenario, the level of mixing experienced *collectively* may be very good, however the level of mixing experienced by the user represented by pseudonym i_1 is poor (since the attacker can determine with a high degree of certainty that user $i_1 \equiv e_1$).

Therefore a model is required to take into account the variation in uncertainty for each ingress pseudonym and egress pseudonym. A crude measure can be constructed by calculating the anonymity set for each egress pseudonym e_j; in other words, the set of ingress pseudonyms connected by an edge to the egress pseudonym e_j in at least one of the matches $m \in M$.

A better model should take into account the likelihood of the occurrence of each of the matchings $P(m|M)$. One way of achieving this is to estimate the probability of the mapping between i_k to e_j for a particular match $m \in M$ as:

$$p^m_{i_k \to e_j} = \begin{cases} P(m|M) & \mathit{if}(i_k, e_j) \in m \\ 0 & \mathit{otherwise} \end{cases} \tag{5.3}$$

Therefore each egress pseudonym e_j has a probability distribution defined over the set of possible ingress pseudonyms i_k as:

$$p_{i_k \to e_j} = \sum_{m_j \in M} p^{m_j}_{i_k \to e_j} \tag{5.4}$$

Entropy can then be used to measure the uncertainty in determining which ingress pseudonym matches a particular egress pseudonym. For example, an estimate of the level of uncertainty in the previous identity of egress pseudonym e_j can be calculated as:

$$H_{e_j} = -\sum_k p_{i_k \to e_j} \log p_{i_k \to e_j}$$

Intuitively, the entropy metric H_{e_j} is related to the number of possible ingress pseudonyms which could have generated the egress pseudonym e; in other words, if entropy is measured as *b bits* then one of 2^b ingress pseudonyms could have generated the egress pseudonym.

This model of likely user movements across a mix zone is not the only way of providing a quantitative metric of anonymity; however it does:

- take into account the likely movement of all the users in the mix zone to provide an estimate of the *collective* movements of all users;

- use the collective movement model to infer the likely individual movements within the context of the movements of others.

Intuitive user feedback is now possible, allowing the user to decide whether to suspend certain location-aware applications or take a detour if the level of privacy offered is too low. For example, the level of anonymity gained in a mix could be displayed as an “anonymity strength” readout on the location device (*e.g.* mobile phone). A more futuristic approach might use virtual reality goggles or a micro-optical display to annotate the user’s view of the surroundings with details of the level of mixing obtained through exiting the mix zone in different directions.

5.6 Partial evaluation of occupancy

The number of users currently inside a mix zone is not necessarily known by an attacker; furthermore, busy mix zones may never empty of people, and therefore the bipartite graph modelling the ingress and egress of pseudonyms continues to grow indefinitely as more users move though the mix zone. In many situations the number of users staying within the mix zone for long periods is low (for example, see the analysis of Active Bat data in Chapter 7) and therefore mixing can be estimated by assuming all users leave within a certain time period. This assumption allows the bipartite graph to be pruned in order to remove pseudonyms who are very likely to have exited the mix zone.

Figure 5.9 provides a graphical representation of the pruning technique. The bipartite graph is truncated to remove all vertexes representing egress pseudonyms who exited the mix zone

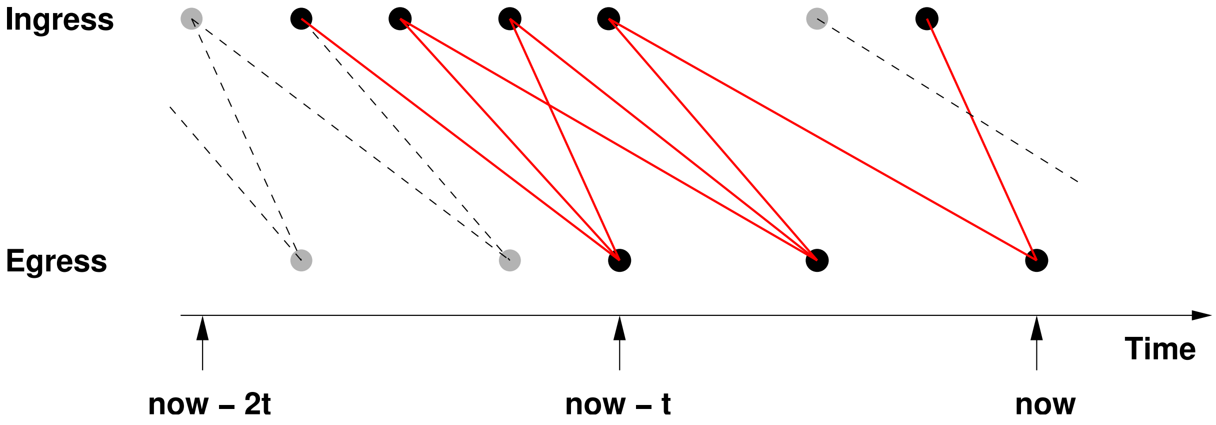


Figure 5.9: The solid black vertexes remain after pruning, whereas the dashed grey vertexes are removed. The solid red lines depict the edges remaining in the bipartite graph for estimating the mapping between ingress pseudonyms and egress pseudonyms; the dashed grey lines represent edges removed from the graph.

more than t seconds ago; all vertexes representing ingress pseudonyms which are not connected by an edge to an egress pseudonym vertex are also removed. Since all users are assumed to leave the mix zone within time t, then the oldest possible ingress pseudonym considered must have entered the mix zone at most $2t$ seconds ago.

5.7 Summary

This chapter introduced the mix zone model to enable location privacy through anonymity. The model functions by reducing the coverage area(s) of location information presented to untrusted applications and therefore prevents malicious applications from tracking the long-term movement of users. Limiting the tracking of user movements reduces the chance an attacker can correlate a pseudonym with a home location and thus infer the pseudonym's real-world identity.

Calculation of the level of anonymity received in the mix zone model is difficult and in the worst case (*i.e.* fully connected mapping between i ingress and e egress users) requires $O(i^{\underline{e}})$ calculations. A method of calculating a lower bound on the level of mixing has been developed; ensuring an optimal lower bound for a given computation time was shown to be NP-Complete and a heuristic-based approach was developed. Methods of providing individuals with feedback concerning the level of mixing within the mix zone was discussed.

Chapter 6

Variable quality model

> "*Every axiom is a security weakness.*"
> —Fred B. Schneider,[1] 2003.

Chapter 4 described the principal benefits of anonymising location information over using access control to enable location privacy. The aim of the variable quality model is to reduce the spatio-temporal quality of location information in order to guarantee a minimum level of anonymity (and therefore location privacy). This chapter starts with a description of one method of using variable quality reduction to increase location privacy. Section 6.2 describes some failure cases of the Gruteser and Grunwald granularity reduction method and presents a new algorithm to provide safe spatially and temporally reduced location information.

Temporal and spatial granularity reduction achieves location privacy by ensuring multiple location events overlap in both spatial and temporal domains. Therefore granularity reduction introduces uncertainty about the identity of users. Let us consider a simple application which is amenable to granularity reduction:

> **Find a Coffee Shop**
> *A computer scientist with a love of coffee is in desperate need of a cappuccino. Therefore he requests a list of nearby coffee shops from his mobile phone together with a map of the local area. The application accepts a possibly empty set of configuration parameters to control which coffee shops are suitable (e.g. only select open shops with certain brands of coffee costing less than a specified amount).*

The "Find a Coffee Shop" application will readily accept a reduction in spatial quality, however the application needs to be responsive, and therefore only a very short degradation in temporal quality is tolerable. For applications which are not interactive, reductions in temporal quality are possible. For example, Gruteser and Grunwald describe a road hazard detection application which monitors vehicle braking sensors to detect near-accident situations [40]. The application requires high spatial accuracy (to accurately locate areas of the highway which are accident-prone) however exact temporal data is not required (just enough to determine whether it was day or night time and associate the accident statistic with relevant weather data).

[1] Part of Fred Scheider's presentation at an Advanced Summer School on Mobile Computing (`http://www.mobilecomputing.list.it/`) in Pisa, Italy which the author attended in September 2003.

Provided the location information returned to the application is spread over a large enough temporal or spatial domain to conceivably originate from many users, an attacker will have difficulty in determining user identity. There are two overall approaches to reducing the quality of location information offered to applications in order to protect privacy:

Constant reduction: reduce the level of accuracy over a particular geographical region or time domain by a constant factor. This has the advantage of knowing *a priori* what accuracy can be offered to applications; however users may be exposed in certain locations (although suitable and relatively infrequent granularity adjustment should prevent long-term tracking). Assessing the level of privacy protection received is difficult.

Variable reduction: dynamically adjust the resolution provided to always ensure some lower-bound on anonymity is met for each location event released. This approach is potentially safer for the users; however applications must cope with varying levels of location accuracy.

This chapter concentrates on the latter technique, taking a similar approach to Gruteser and Grunwald. Their work is extended here to permit session state pseudonymous applications to function and allow multiple applications to request location information with both spatial and temporal granularity reductions without compromising location privacy (even in the presence of collusion). The analysis in the next section discusses a flaw in the spatial reduction algorithm employed by Gruteser and Grunwald; an enhanced algorithm which enables these extensions and corrects the flaw is also presented in the next section. Discussion begins with the security policy:

Security policy

1. *Application Registration:* Applications must register with the location server and express their preferences with regard to spatial and/or temporal granularity reduction.

2. *User Registration:* Users must register with the location server and inform the location server of any applications they wish to use.

3. *Location Service Guarantee:* The location server aims to provide the location application with all the relevant location events. The location server *must* reduce the temporal and spatial resolution to ensure the minimum level of anonymity requested by a user is attained; the location server *may* reduce the resolution further than requested in order to protect the location privacy constraints of other users. The location server will attach a pseudonym to each location event to enable communication between the user and the application.

6.1 Quantitative measure of anonymity

In the variable quality model, a location event is represented by the triple:

$$\langle pseudonym, location\_container, time\_period\rangle.$$

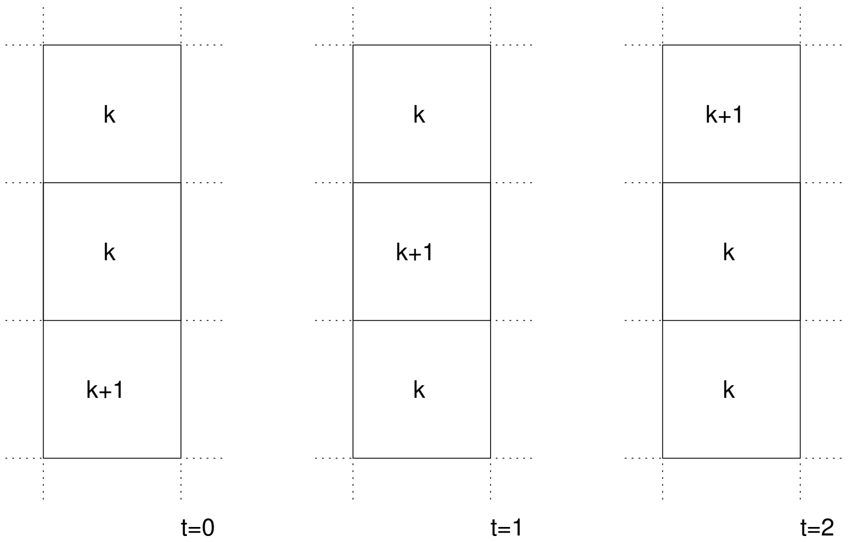


Figure 6.1: Three spatial areas at different time offsets t are marked with their anonymity set size (either k or $k + 1$). One interpretation of the changing set sizes is that a single user is walking from bottom to top and all other users are stationary. The likelihood of this prediction depends on the relative probabilities of users moving as opposed to staying stationary.

The location server can adjust the size of the $location\_container$ and the $time\_period$ of the location event in order to increase location privacy; such adjustments may result in a degradation in performance of the location-aware application (*e.g.* the "Find a Coffee Shop" application may have to list a greater number of shops than would be necessary if a precise location of the computer scientist is known).

In the mix zone model, the attacker gained precise knowledge of user movement within the application zone; this knowledge could then be applied to improve estimates of possible user movements across the mix zone. For this reason the anonymity set is not a good measure for the level of anonymity gained within the mix zone model. In the variable quality model the location server can enlarge the $location\_container$ and increase the $time\_period$ associated with location events to ensure every user is k-anonymous; in other words, ensure that at least k distinct users are represented by location event triples with identical values for both $location\_container$ and $time\_period$. Provided the user pseudonym changes on every location update, an attacker viewing the location event triples released by the location server cannot directly measure the movement patterns of users from the data. Therefore the anonymity set measure is a suitable quantitative metric of location privacy for the variable quality model.

Since each individual location event is indistinguishable from at least $k - 1$ others there can exist no simple home locations; however it is still possible that a complex home location (*e.g.* route to work) could reveal the identity of a user. For example, there remain scenarios where an attacker, together with a model of possible user movement may be able to link pseudonyms together (a prerequisite for a complex home location attack); for example see Figure 6.1. The value of k can be increased to make a complex home location attack progressively more difficult.

Session state applications do not work if the pseudonym representing a particular user must change with every new location event. If the pseudonym does remain constant, the user can be tracked and anonymity (or more accurately pseudonymity) relies solely on there existing no complex home locations in the data set. If an attacker can determine the true identity behind a pseudonym, tracking is limited to the length of the session. Pseudonyms kept constant for session state application interaction should be excluded from the anonymity measure since otherwise an attacker has access to movement models of some of the users in the anonymity set.

The location server can choose how to reduce the spatial and temporal quality of the location events in order to ensure the k-anonymity property holds. Since some applications may prefer spatial quality reduction whereas other applications may prefer temporal reduction both these approaches are explored next. These techniques can be considered as independent security mechanisms which enforce the security properties of the variable quality model.

Section 6.2 analyses possible attacks on these two security mechanisms and examines how a location server may safely provide temporally reduced location data to one application, while revealing spatially reduced location events to another. This is non-trivial since, under the threat model (see Section 4.5), applications are untrusted and therefore may collude and share information.

6.1.1 Spatial resolution reduction

Location data is normally indexed by coordinates, containment or proximity (see Section 2.1.4). In order to apply spatial quality reduction, coordinate and proximity primitives are transformed into containers.[2] It is not sensible to centre containment on the user's true location since averaging the container extent will reveal a more accurate estimate of user location; therefore container boundaries must be constrained to boundary lines common to all users.

In order to achieve variable reduction in spatial quality, a hierarchy of successively less accurate containers is needed. A tree data structure can be used to describe the containment relation, where a node represents a location container and the boundary lines of any child nodes are within the boundary line of their parent; in other words containers should be arranged so that the boundary line of any given node in the tree is larger than, or equal to, the convex hull of all its child node boundary lines. Ideally, containers should tessellate to ensure each location coordinate lies within exactly one container at every level in the tree; this ensures that any location event can be represented at every level of spatial quality. For a two-dimensional Cartesian coordinate system, an R-tree structure of rectangles is applicable. (Section 6.2.4 describes the benefits of the R-tree data structure in greater detail.)

At a particular time, spatial quality reduction can be achieved for the current set of location events by placing the location events at the nodes in the tree representing their spatial containment. Initially each location event is likely to be located at a leaf node in the tree since leaf nodes represent the most accurate spatial containers for each location event. We can ensure k-anonymity holds for all location events by moving each of the location events progressively up the tree (and therefore reducing their spatial accuracy) until every location event has at least $k - 1$ other location events sharing the same node in the tree. An efficient algorithm to do this is discussed in the next section.

[2] Container information can be transformed into proximity or coordinate formats after resolution reduction has been achieved if applications prefer to receive events in these formats.

6.1.2 Temporal resolution reduction

Temporal reduction of location events is achieved by increasing the time period over which the location event could have occurred. In order to ensure users are k-anonymous, the location server buffers consecutive location events received for each spatial container. When k location events are recorded from k distinct users, the time period of all the location events are adjusted to span from the time the first location event occurred to the time the last location event occurred. The location server can then attach a unique pseudonym to each location event and release the data to the application. Since none of the location information for a particular spatial region is released until at least k distinct users have visited the spatial region, an attacker cannot determine any details of the path taken by a user through the containers.

Buffering of k location events for a particular spatial container will often take more than one location update period of the underlying location system. (Of course, if there are at least k users simultaneously located within one spatial container, only one update period is needed.) Therefore a user may pass through several containers while the location server is buffering enough location events to ensure the k-anonymity property holds. It is important to ensure that a user, who may visit a particular spatial region more than once before the k-anonymity property is met, is only counted once in each spatial region.

6.2 Ensuring correctness

The last section described two security mechanisms to enforce the variable quality model through either reduction in the temporal quality or reduction in the spatial quality of a location event. Some applications may prefer spatial quality reduction whereas other applications may prefer temporal reduction. This section describes a method which enables a location server to offer applications both spatially reduced and temporally reduced location data whilst maintaining the location privacy of its users. The section begins with a description of three attacks which are made possible by allowing applications to choose whether to receive location events with either spatial or temporal reductions in quality. This section then outlines a solution to prevent these attacks from occurring.

In this section a basic quantum of space-time is represented by a cuboid c whose x and y axes measure x_1 to x_2 and y_1 to y_2 and represent spatial coordinates; similarly, the z axis represents time from t_1 to t_2. For example, if a location event occurs at location (x,y) and time t, it is contained within cuboid c if $x_1 \leq x < x_2 \wedge y_1 \leq y < y_2 \wedge t_1 \leq t < t_2$.

Spatial quality reduction is represented by grouping together two or more cuboids which are spatially adjacent (as defined by the containment relation in Section 6.1.1) and are constrained within one particular quanta of time z_i to z_{i+1}. Similarly temporal reduction is achieved by grouping together two or more cuboids adjacent in time along the z axis and are all part of a particular basic quanta of space x_j to x_{j+1} and y_k to y_{k+1}.

When two or more cuboids are combined together by the location server to form a larger container, the attacker does not know the number of users within each cuboid; instead he only knows the total number, m, of users within a given set of cuboids. Let $c_1, \ldots, c_n$ represent the number of users within each cuboid numbered $1, \ldots, n$. If all n cuboids are combined by the location server in order to ensure each user is at least k-anonymous then the location server is providing the attacker with the following information, (which is in the form of a linear

Algorithm 6.1 Gruteser and Grunwald's algorithm for spatial reduction of p_u

```
Require: $P$: Set of people (initially all people located in the world)
Require: $|P| \geq k$
  $q \leftarrow quadtree\_root$
  $q_{prev} \leftarrow quadtree\_root$
  while $|P| > k$ do
    $q_{prev} \leftarrow q$
    $q \leftarrow get\_sub\_quad(p_u, q)$ //select the sub-tree of $q$ which contains $p_u$
    $P \leftarrow members\_of(q)$
  output $q_{qprev}$
```

equation):

$$c_1 + \ldots + c_n = m \geq k. \tag{6.1}$$

Note that each equation variable represents (an unknown) number of users within a cuboid, and the equation defines a composition of cuboids which forms a container representing the location events given to the attacker. According to the threat model, an attacker is a global hostile observer and will therefore receive a set of linear equations describing all the location events which the location server has made available. Since the location server may provide both spatial reductions in quality and temporal reductions in quality to different applications, the number of users in some of the cuboids c_i may occur in two linear equations (once for temporal reduction and once for spatial reduction). Under the assumptions of the variable quality model, location privacy is broken if the attacker is able to conclude:

$$c_1^{'} + \ldots + c_n^{'} < k \tag{6.2}$$

for any combination of cuboids $c_1^{'} + \ldots + c_n^{'}$.

6.2.1 Worked example: the Gruteser and Grunwald algorithm

Analysis is conducted on a location privacy algorithm by Gruteser and Grunwald, where location data is anonymised by reducing either the spatial or temporal accuracy of location events [41]. The algorithm uses the size of the anonymity set as its quantifiable metric of location privacy. The spatial reduction algorithm uses a quad-tree data structure [111] to represent the containment relation of a hierarchy of progressively larger spatial containers. The algorithm presented in the paper is described in Algorithm 6.1 and takes a single user's position p_u and the location of other users $p_1 \ldots p_n \in P$ as input and returns the minimum spatial container for user p_u which still meets the anonymity set size k; it does this by keeping track of the current container q and the parent container q_{prev} in a quad-tree representing successively more accurate spatial containers.

The original attack presented in the Gruteser and Grunwald paper is now described in terms of cuboids and linear equations. Figure 6.2 provides a graphical representation of the location of users within the cuboids. Using Algorithm 6.1 on this data with $k = 3$ returns the following location event triples:

$$\begin{array}{lll} \langle i_1, a, t\rangle & \langle i_2, a, t\rangle & \langle i_3, a, t\rangle \\ \langle i_4, c, t\rangle & \langle i_5, c, t\rangle & \langle i_6, c, t\rangle \\ \langle i_7, d, t\rangle & \langle i_8, d, t\rangle & \langle i_9, d, t\rangle \\ \langle j, a{+}b{+}c{+}d, t\rangle & & \end{array}$$

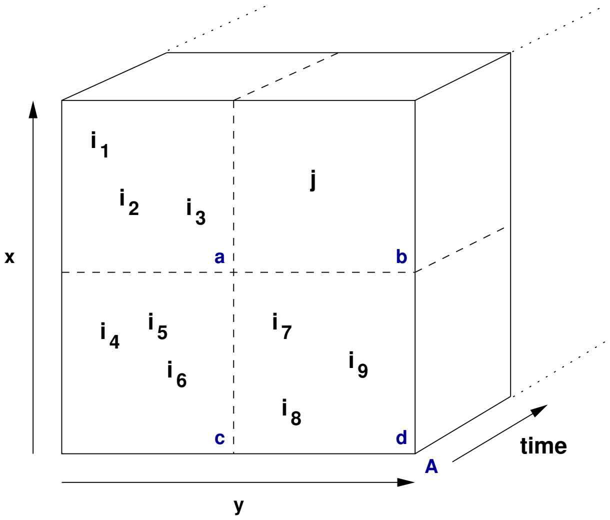


Figure 6.2: Example erroneous case of Gruteser and Grunwald algorithm with $k = 3$ as presented in the original paper. The algorithm returns quad-tree areas a, c, d and $A = a+b+c+d$, leaving the user j exposed in b; an attacker can deduce that a single user occupies b because he receives three sightings from a, b and c and one from the larger area A.

The location event triples represent the following set of linear equations:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 3 \\ 10 \end{bmatrix}$$

which have four variables and four non-collinear equations. The attacker can break the anonymity introduced by Algorithm 6.1 through Gaussian elimination and infer a single user with pseudonym j is located in the spatio-temporal region b (region b may be a home location, and thus the privacy offered through pseudonymity to user j is potentially violated).

The anonymity methods proposed by Gruteser and Grunwald suffer from several problems:

- in certain cases the spatial reduction method described in Algorithm 6.1 does not ensure the set of released location events preserve location privacy (since the k-anonymity requirement does not always hold);

- the quad-tree hierarchy which allows progressive reduction in spatial quality does not take into account physical constraints of the environment (*e.g.* walls) which affects which cuboids should be combined together to form larger containers;

- the methods proposed by Gruteser and Grunwald do not consider the potential for an attacker to combine temporally reduced data with spatially reduced data in order to break the location privacy offered to users; and

- shared location-aware applications require knowledge of the movement of other users of the location server, and therefore efficiency can be improved by calculating the spatial reduction of all the location events in one iteration through the quad-tree.

6.2.2 Possible attacks on released location events

Section 6.1 demonstrated that the anonymity offered by a security mechanism which enforces the variable quality model is broken if an attacker can produce an equation of the form shown in Equation 6.2. Therefore the location server must ensure that any set of spatial containers (equations) built from cuboids (variables) do not allow the attacker to infer a container (equation) which contains less than k location events.

Given a set of linear equations describing the released set of location events, any combination of subtraction or addition of equations is permissible (representing addition or subtraction of cuboids which form different containers). Three possible ways of combining linear equations which lead to possible attacks are described next.

Subtraction attack

If all the cuboids in container A are also cuboids present in container B then, in the case where $B \neq A$, $B \setminus A$ contains at least one cuboid. Subtracting the linear equation representing the container A from the linear equation representing container B may result in a new container (represented by a linear equation) with less than k location events. For example, the data released by the Gruteser and Grunwald algorithm for the scenario shown in Figure 6.3 is:

$$\begin{array}{lll} \langle i_1, a, t\rangle & \langle i_2, a, t\rangle & \langle i_3, a, t\rangle \\ \langle j, a{+}b{+}c{+}d, t\rangle & & \end{array}$$

which is represented by the linear equations:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

In this case the attacker can infer that j is alone and contained within the cuboids $b{+}c{+}d$, since subtracting the first linear equation from the second produces the new linear equation $b{+}c{+}d = 1$; this violates the constraint required in Equation 6.2 to ensure the variable quality model provides k-anonymity.

Negation attack

An attacker can also use the absence of any location information to infer that someone is *not* in a home location. For example, if a user has a simple home location covering a spatial region which is absent from the released dataset then an attacker can infer the user must be present elsewhere—the lack of any information from a spatial region allows an attacker to infer information concerning the location of individuals; this type of attack is especially important if a user was *supposed* to be at a particular location (*e.g.* office desk) and an attacker (*e.g.* employer) can demonstrate they were not. An example is shown in Figure 6.4. The data released by the Gruteser and Grunwald algorithm for this example is:

$$\begin{array}{lll} \langle i_1, a, t\rangle & \langle i_2, a, t\rangle & \langle i_3, a, t\rangle \\ \langle i_7, b, t\rangle & \langle i_8, b, t\rangle & \langle i_9, b, t\rangle \\ \langle i_4, c, t\rangle & \langle i_5, c, t\rangle & \langle i_6, c, t\rangle \end{array}$$

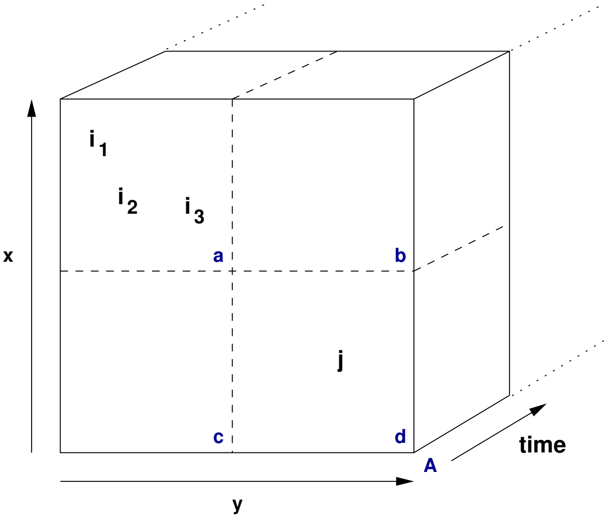


Figure 6.3: The Gruteser and Grunwald algorithm with $k = 3$ returns quad-tree areas a and $A = a+b+c+d$, leaving the user j exposed in d; an attacker can deduce that a single user occupies either $b+c+d$ because he receives three sightings from a and one from A.

and this results in the the following set of linear equations (the last equation is implicitly derived from the lack of any information for the cuboid d):

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 3 \\ 9 \end{bmatrix}$$

and from this the attacker can infer that $d = 0$.

Linear inference attack

If there are at least as many variables (cuboids) as equations (sets of cuboids released as containers) then an attacker can deduce the number of users within each cuboid which may potentially result in pseudonyms being less than k-anonymous. If the scenario depicted in Figure 6.5 results in four containers $a+b$, $c+d$, $a+c$ and $b+d$ and the following dataset (not possible in the Gruteser and Grunwald algorithm, but may occur with a more generalised algorithm):

$$\begin{array}{ccc} \langle i_1, a+b, t\rangle & \langle i_2, a+b, t\rangle & \langle i_3, a+b, t\rangle \\ \langle i'_1, a+c, t\rangle & \langle i'_2, a+c, t\rangle & \langle i'_4, a+c, t\rangle \\ \langle i_4, c+d, t\rangle & \langle i_5, c+d, t\rangle & \langle i_6, c+d, t\rangle \\ \langle i'_3, b+d, t\rangle & \langle i'_5, b+d, t\rangle & \langle i'_6, b+d, t\rangle \end{array}$$

then, when viewed as a set of linear equations, the attacker can use Gaussian elimination to deduce the number of users in each cuboid. Note that the released datasets overlap—this may occur as a result of multiple applications having different preferences for the type of spatio-temporal granularity reduction employed. (Recall that under the location anonymity threat model (see Section 4.5) all applications are viewed as one global hostile observer).

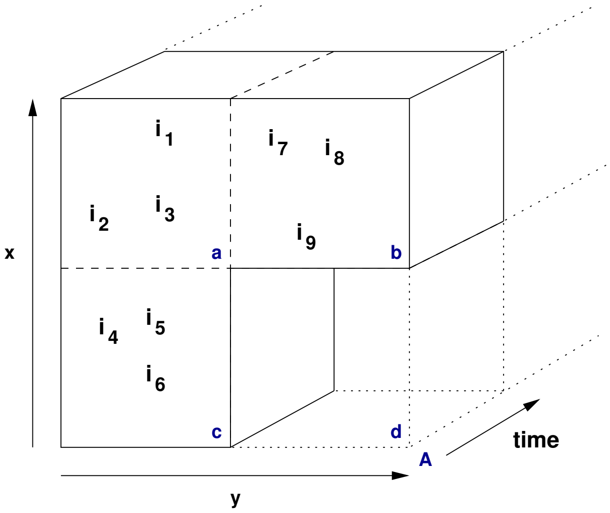


Figure 6.4: The Gruteser and Grunwald algorithm with $k = 3$ returns quad-tree areas a, b and c; if d represents a simple home location, an attacker can deduce that the user who predominantly occupies d is currently located elsewhere.

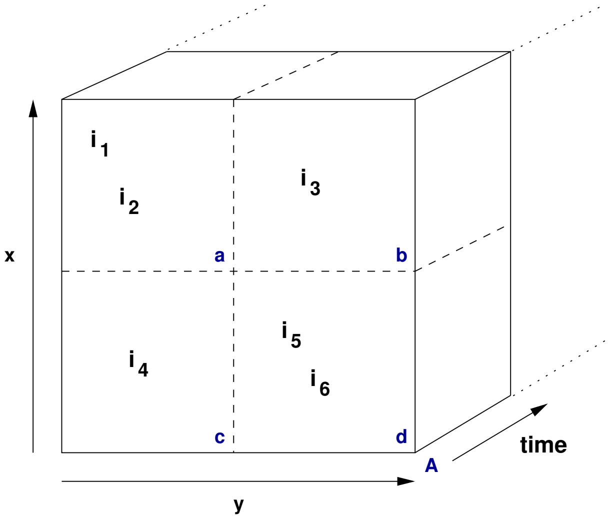


Figure 6.5: If four containers $a{+}b$, $c{+}d$, $a{+}c$ and $b{+}d$ are released in an attempt to meet the anonymity constraint $k = 3$ then Gaussian elimination can be used to infer the number of users in each cell. For all meaningful values of k ($k \geq 2$) then this scenario represents a successful attack.

Viewing the formation of cuboids into containers as linear equations allows careful analysis of the possible attack space; three methods have been presented which allow an attacker to combine location events in different ways in order to infer a new container which contravenes the location privacy criteria expressed in Equation 6.2. Since Equation 6.2 is a necessary precondition for breaking the k-anonymity property, a security mechanism which prevents these three types of attack enforces the variable quality model and therefore protects location privacy.

6.2.3 Container constraints

The linear inference, subtraction and negation attacks can be avoided by constraining the shape of containers the location server generates. A simple and effective measure is to force each variable to appear exactly once in any linear equation; in other words every cuboid is contained within exactly one container, so no containers can overlap either spatially or temporally with each other. This is difficult if two applications require access to the same underlying location data, with one preferring temporal quality reduction and the other preferring spatial quality reduction.

In order to permit multiple applications which have different spatial and temporal accuracy requirements to function, multiple views on the same dataset can be constructed. A *view* is simply a set of containers which make use of each cuboid at most once. If every view uses a different set of pseudonyms to represent the same underlying users the following constraints on any released linear equations prevent an attacker from applying one of the attacks presented in the last section:

1. no linear equation shall contain only the variables of another (no subtraction attack),
2. the set of linear equations is under-constrained (no linear inference), and
3. the set of linear equations for each view of the dataset contains every variable (no negation attack).

These three rules translate into spatio-temporal restrictions on containers:

1. all containers must be at least partially disjoint from one another,
2. the number of containers positioned over any spatio-temporal region must be less than the number of cuboids used to construct the containers, and
3. for every view, each cuboid must be present in exactly one container.

One convenient and safe arrangement of containers built on top of cuboids is to permit only two types of resolution reduction:

Spatial reduction: a set of non-overlapping containers which (1) contain at least three cuboids, and (2) every cuboid in a container is in the same time period $(t_i, t_i + 1)$;

Temporal reduction: a set of non-overlapping containers which (1) contain at least two cuboids, and (2) every cuboid within a container is located in the same spatial region $(x_j, x_{j+1}), (y_k, y_{k+1})$.

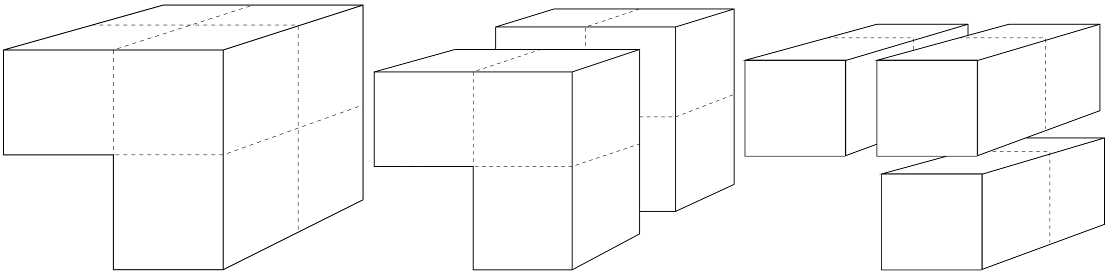

(a) Space-time region (b) Spatial containers (c) Temporal containers

Figure 6.6: One valid arrangement of spatial and temporal container views. Figures (b) and (c) provide two views on the same underlying data contained within the space-time segment shown in (a).

Note that spatially reduced containers cannot overlap with other spatially reduced containers (nor can temporally reduced containers overlap with each other); however temporally reduced containers may intersect spatially reduced containers. Provided the spatially reduced containers and temporal reduced containers both cover the entirety of space-time, then the following container constraints meet the three rules described above:

1. *All containers must be at least partially disjoint from one another.* Intersection is only possible between temporal resolution reduction containers and spatial resolution reduction containers. This intersection is only ever partial, since temporal containers must be at least two units of time long, and spatial containers can only ever be one unit of time long.

2. *The number of containers positioned over any spatio-temporal region must be less than the number of cuboids used to construct the containers.* The smallest size temporally reduced and spatially reduced containers can be optimally packed as shown in Figure 6.6 which contains two spatially reduced containers and three temporally reduced containers placed on top of the same underlying user data. In this configuration there are six unknown variables (cuboids) and five containers (linear equations), so the problem is under-constrained. Decreasing the spatial resolution (*i.e.* adding one or more cubes in the same time period to a spatial container) or temporal resolution (*i.e.* adding one or more cubes in the same spatial area to a temporal container) introduces more unknown variables (cuboids) than linear equations (containers), so a linear inference attack is not possible for larger-sized containers.

3. *For every view, each cuboid must be present in at least one container.* Both spatially reduced containers and temporally reduced containers cover all of space-time (at least all of space-time which is representable in the model), so this rule is satisfied.

The spatial reduction and temporal reduction views meet the three spatio-temporal restriction rules outlined earlier and therefore ensure the three attacks presented earlier are not possible. However whilst the spatial and temporal views do not lead to an immediate attack, it is possible for an attacker to position an observer within one or more cuboids at carefully chosen times and use the information gained about underlying user positions to provide enough data to mount a substraction attack, negation attack or possibly a linear inference attack. Therefore giving away more than one view on a location dataset increases the power carefully placed hostile observers may pose to the system. There are two approaches to reducing this risk: (1) increase the minimum number of cuboids which make up the spatial or temporal containers and therefore decrease the ratio of linear equations to variables; and (2) increase the value of k in order to reduce the likelihood of an attack leading to an actual invasion of location privacy (*i.e.* inferring the real-world identity of a pseudonym).

6.2.4 Modelling environmental constraints

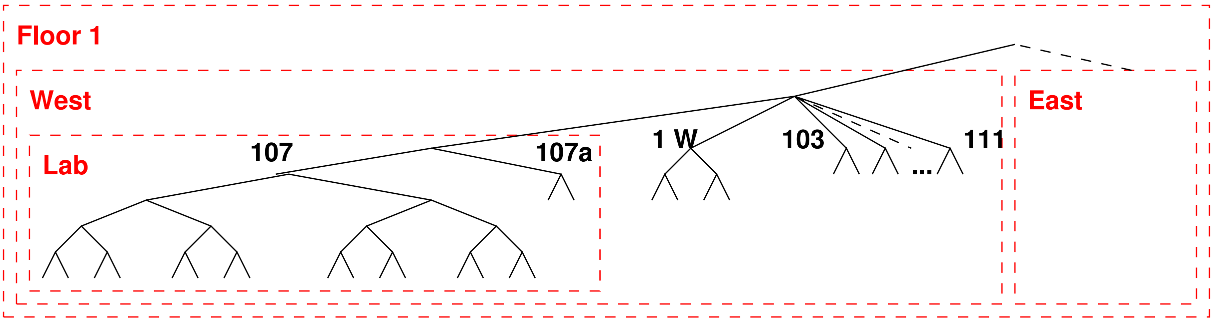
Defining spatial reduction is more difficult in built-up areas (*e.g.* inside offices or streets) than open spaces because the environment restricts the movement of users due to the existence of obstructions. Progressive reductions in spatial accuracy must combine areas which are logically adjacent rather than necessarily physically closest. For example, combining location events based on Cartesian coordinates alone is not sensible if users in portions of two different rooms are combined together in preference to all users of one room. For this reason a simple quad-tree data structure representing successive refinement of spatial accuracy is not appropriate in environments with obstructions. Instead, containment should take into account the logical connectivity of spatial regions.

Samet describes in detail several spatial data structures [112], including the R-tree data structure (a generalisation of the B-tree data structure where every node in the tree has at most two children). An R-tree of the order (m, M) contains between $m \leq \lceil M/2 \rceil$ and M entries for each node in the tree except for the root node which has at least two entries (unless it is a leaf node). Each node in the tree represents a rectangle which is a tight spatial bound on all the containers within its subtree.

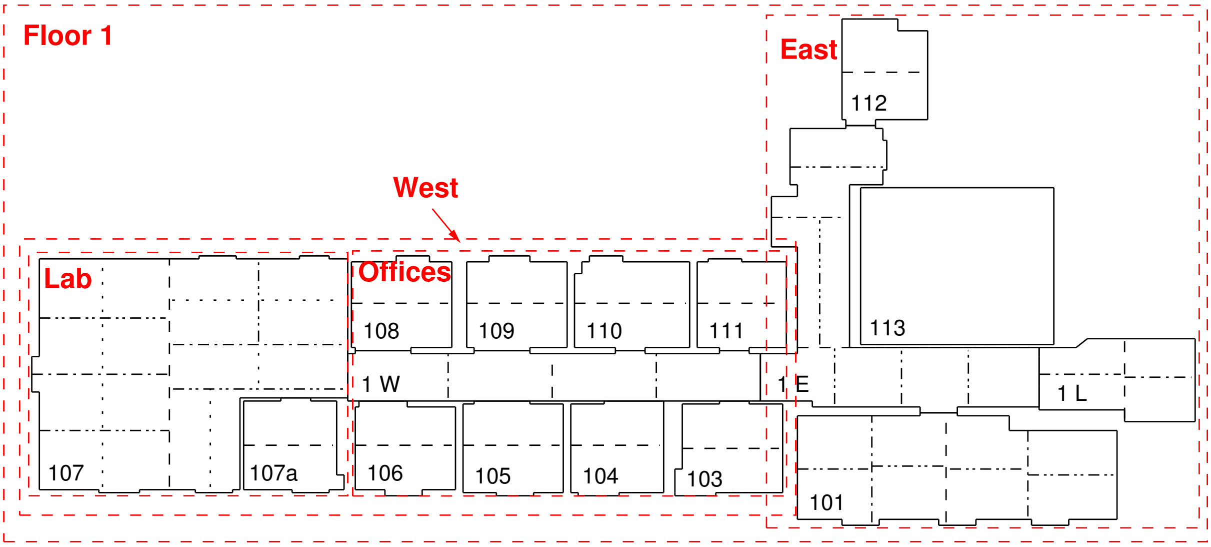
An example $(2, 10)$ R-tree for the first floor of AT&T Labs Cambridge is shown in Figure 6.7; note that rooms undergo adaptive subdivision in order to increase the spatial accuracy within a room; the leaves of the tree contain a list of cuboids which define the outline of the relevant portion of the room (not drawn on the diagram to aid clarity of the rest of the figure). The R-tree has been constructed to demonstrate one interpretation of logical adjacency of rooms on the first floor.

Samet describes several methods of performing insertion and deletion of spatial containers into an R-tree as well as methods of determining the set of containers any given location position is a member of. Since rooms and groups of rooms are not always rectangular, the bounding rectangles represented by nodes in the R-tree may overlap. A room, or portion of a room can only appear in one subtree. So, when we wish to determine the hierarchy of containers a room (or area within a room) is in,the search may involve following more than one subtree. Therefore, if there are n nodes in the R-tree, the worst case search cost is $O(n)$. Foley *et al.* describe a method of providing a unique address to each leaf node, based on the path from root to leaf in a quad-tree [35]. This technique can be extended to an (m, M) R-tree with n nodes and a maximum tree depth of N by assigning n addresses in the address space M^N. For example, room 107a in the graph shown in Figure 6.7 is addressed as 001; the first digit determines which edge to choose from the root node to a secondary node; the second digit determines the edge from the secondary node to the tertiary node, and so on.

A location event described in Cartesian coordinates can be converted into a cuboid by thresh-



(a) R-Tree of rooms and their sub-divisions



(b) Map of the first floor of AT&T Labs Cambridge

Figure 6.7: Figure (a) gives a direct view of the R-tree representing the spatial containment relation of the rooms on the first floor of AT&T Labs Cambridge, which is shown in Figure (b).

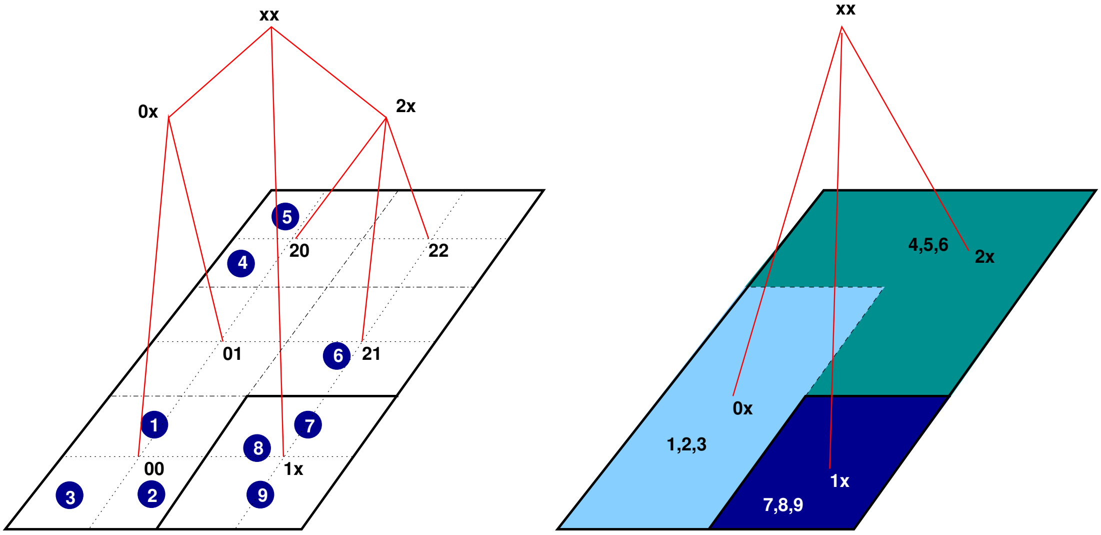
Algorithm 6.2 Spatial reduction

```
Require: $k = anonymity\ size$
Require: $L = set\ of\ location\ sightings\ as\ (pseudonym, address)\ pairs$
  proc $process\_region(address, subtree)$
    if $leafnode(subtree)$ then
      $found \leftarrow \varnothing$
      while $address\_of(head(L)) = address$ do
        $found \leftarrow found \cup \{pop(L)\}$
      return $found$
    else
      $regions \leftarrow \varnothing$
      $k\_anon \leftarrow true$ //remains true iff all subregions have $\geq k$ pseudonyms
      for all $c \in children(subtree)$ do
        $r \leftarrow process\_region(address\_of(c), c)$
        $regions \leftarrow regions \cup \{r\}$
        if $|r| < k$ then
          $k\_anon \leftarrow false$
      if $k\_anon$ then
        return $regions$
      else
        $reduce \leftarrow \varnothing$
        for all $(pseudonym, old\_address) \in regions$ do
          $reduce \leftarrow reduce \cup \{(pseudonym, address)\}$
        return $reduce$
  proc $Reduce\_Granularity()$
    $r \leftarrow process\_region(\varnothing, R{-}tree)$
    if $|r| < k$ then
      return $\varnothing$
    else
      return $r$
```

olding. A matrix representing the mapping from cuboids to the correct position in tree (represented by the address procedure described above) can be used to find the position of the location event in the R-tree in $O(1)$. Using the matrix data structure increases the performance of determining a cuboid's position in the R-tree at the expense of memory usage (which scales linearly with coverage area).

6.2.5 Combining container constraints and spatial models

Efficient computation of temporal and spatial container regions is much easier for the variable quality model than the mix zone model. Applications can express interest in location data with either a fixed spatial resolution or fixed temporal resolution. Temporal reduction can be achieved by combining cuboids in the temporal domain to build a new container which contains at least k users and meets the minimum number of cuboids required to build a container. Spatial reduction can be done by combining multiple containers together using the R-tree as a guide for adjacency. In order to ensure the minimum number of cuboids exist in any spatial container



(a) Location of nine users

(b) Spatial reduction of location

Figure 6.8: Figure (a) shows the raw location data of the users in the world together with the R-tree which describes spatial proximity. Figure (b) shows the result of running the reduction algorithm on the raw location data.

a leaf node in the R-tree should contain the minimum number of cuboids.

The matrix mapping scheme can determine the position of l location events in the R-tree with n nodes with complexity $O(l)$. The location events can then be sorted (ordered by unique address) in $O(l \log l)$ time. Algorithm 6.2 processes an ordered list of location events to produce a list of spatially reduced location events in $O(l + n)$ time. The algorithm works by processing the R-tree describing the physical environment in a top-down fashion. At node d, the algorithm recursively visits all its children to calculate the number of location events in each container. If the visited node d is a leaf node, then the location events contained within its spatial region are at the head of the list of location events (provided that the ordering for sorting the list and recursing the R-tree is the same); these location events are removed from the list and returned to the calling function. If the visited node d is not a leaf node and each child node contains at least k users, then the container addresses of the location events are returned to the calling function unaltered; otherwise the container addresses are updated to the address assigned to the current node d.

To aid understanding of the algorithm, consider the simple example shown in Figure 6.8(a) where $k = 3$ and the minimum number of cuboids within a spatial container is 4. For this example the elements of the sorted list are:

$$L = \begin{array}{lllll} [\langle 1, 00\rangle, & \langle 2, 00\rangle, & \langle 3, 00\rangle, & \langle 7, 1\mathrm{x}\rangle, & \\ \langle 8, 1\mathrm{x}\rangle, & \langle 9, 1\mathrm{x}\rangle, & \langle 4, 20\rangle, & \langle 5, 20\rangle, & \langle 6, 21\rangle] \end{array}$$

The algorithm starts at the root node (labelled xx) and recurses to node 0x followed by 00 which (since it is a leaf node) returns three the location events $[\langle 1, 00\rangle, \langle 2, 00\rangle, \langle 3, 00\rangle]$ from the head of the list L. The function then recurses to node 01 and returns no location events. The location events received from its children at node 0x are now processed; one child (namely 01) does not provide k-anonymity (in this case 01 provides no location events) so the address labels of all the location events within this subtree are relabelled with 0x before being returned to the calling function. Processing of the remaining two branches from the node xx continues in a similar fashion, resulting in the spatial reductions depicted in Figure 6.8(b) and shown below:

$$L' = \begin{array}{l} [\langle 1, \mathsf{0x}\rangle, \quad \langle 2, \mathsf{0x}\rangle, \quad \langle 3, \mathsf{0x}\rangle, \quad \langle 7, \mathsf{1x}\rangle, \\ \ \langle 8, \mathsf{1x}\rangle, \quad \langle 9, \mathsf{1x}\rangle, \quad \langle 4, \mathsf{2x}\rangle, \quad \langle 5, \mathsf{2x}\rangle, \quad \langle 6, \mathsf{2x}\rangle] \end{array}$$

6.3 Summary

This chapter introduced the variable quality model to enable location privacy through anonymity. The model works by reducing the spatio-temporal accuracy of location information in order to ensure users are at least k-anonymous. This technique prevents an attacker from determining home locations associated with the data and therefore prohibits malicious application writers (who may collude) from correlating a pseudonym with a real-world identity.

The major flaws of the spatio-temporal reduction algorithms of Gruteser and Grunwald were discussed. The attack strategy was formalised using a set of linear equations in order to demonstrate how the flaws could be successfully removed. A method of performing (correct) spatio-temporal granularity reduction which scales $O(l \log l + n)$ in complexity with the number of users l and number of containers n in an R-tree hierarchy was developed.

Chapter 7

Implementation

> "*[W]hile technology infrastructures tend to persist for generations, laws and policies can change overnight. . . . If we do nothing, new technologies will give the government new automatic surveillance capabilities that Stalin could never have dreamed of.*"
> —Phil Zimmermann,[1] 1996.

Chapter 5 and Chapter 6 described two different security mechanisms for enforcing location privacy through anonymity. This chapter demonstrates how these techniques can be applied to real data gathered from the Active Bat system. The chapter starts with a discussion on the quality of location data available and then analyses the performance of the mix zone model and variable quality model using these data.

7.1 Available location data

An empirical analysis of the performance of the mix zone model and variable quality model requires access to a large body of location data concerning the movement of individuals in their environment. It is expected that the behaviour of both schemes will depend on the movement patterns and density of the user population. Detailed movement data are available from the Active Bat system and this can be used to assess the levels of location privacy available in a fairly constrained indoor setting. Unfortunately information concerning the movement of individuals in wide area outdoor scenarios is scarce. This is partially because the dominant source for fine-grained outdoor location data is currently GPS, which is an inside-out location technology, and therefore there is no central repository of movement data of all GPS users.[2] Location data from mobile phone network operators is still not of sufficient quality to offer fine-grained location-based services, although this will undoubtedly improve as more advanced techniques to increase the accuracy are deployed (see the discussion in Section 2.2.3).

An alternative approach is to use simulation of user movement to estimate the effectiveness of the mix zone model and variable quality model. Micro-models of the movement of crowds within buildings, stadia and city centres has received attention from researchers investigating the movements of people during emergency situations. Research often focuses on modelling

[1] `http://www.cdt.org/crypto/current_legis/960626_Zimm_test.html`

[2] From a privacy perspective one might regard this as a good thing.

how people react to a fire within a confined space; these models are then used to improve the design of emergency exits. Helbing *et al.* describe a method of simulating the movement of pedestrians as a many particle system [48] where each individual pedestrian is represented by an individual particle (often called *microscopic* simulation). Pedestrians are assigned a desired velocity and sociological and physical constraints (*e.g.* walls) are modelled as forces which affect the movement of the individual. The movement of users is described by a set of coupled differential equations where the velocity of the individual is modified by the interaction forces of other individuals and physical constraints. Gloor used this method to produce a simulator called PedSim[3] which models the movement of hikers through Alpine terrain [39].

Teknomo provides a detailed description of the various microscopic pedestrian models present in the literature [132] and develops a computer-aided method of measuring the movement of people across the field of view of a camera mounted vertically above a street. The collected movement data was then used to validate several movement models. Teknomo has published the pedestrian simulator developed as part of his PhD work on the Internet.[4]

The current generation of microscopic simulators assume pedestrians have simple goals; for example, simulating panic and attempts by individuals to exit from a burning building. More complex models representing the likely long-term goals of shoppers and tourists present in a city centre are needed in order to ensure realistic movement models can be generated. Accurate long-term goal models (*e.g.* "visit shop X to buy book, stop for coffee at cafe Y, go to bus stop") are a prerequisite for using simulated data to estimate the likely level of location privacy in the mix zone model and predict the levels of accuracy reduction in the variable quality model. Therefore this dissertation uses data from the Active Bat system to enable analysis of the mix zone model and variable quality model.

7.1.1 Active Bat data

The data used in this chapter were collected from an installation of the Active Bat system at AT&T Labs Cambridge during January and February 2002. This dataset was chosen in preference to more recent data gathering activities within the Laboratory for Communication Engineering because: (1) the coverage area at AT&T was much larger; and (2) a greater number (and larger percentage) of employees at AT&T wore their bat consistently whilst in the office.

The raw dataset contains over ten million location events. Several filters were applied to remove location events which are: (1) from Bats not attached to personnel; (2) recorded outside weekdays between 9am and 5pm; and (3) outside the constraints of the first floor of the building. A plan view of the building can be seen in Figure 7.1. The figure also depicts three types of spatial containers which are used to create four event-based location services:

Teleport desktops: when a user enters a computer zone and clicks a button on their Bat, their current desktop is moved (via VNC) to the computer associated with the computer zone.

Gym card reminder: when a user collects a (physical) corporate gym card the user clicks on the gym card reminder service button positioned in hallway 1W. The reminder service then commands the Active Bat to play an audible beep (similar to a mobile phone ring

[3] `http://pedsim.silmaril.org/`

[4] `http://people.revoledu.com/kardi/`

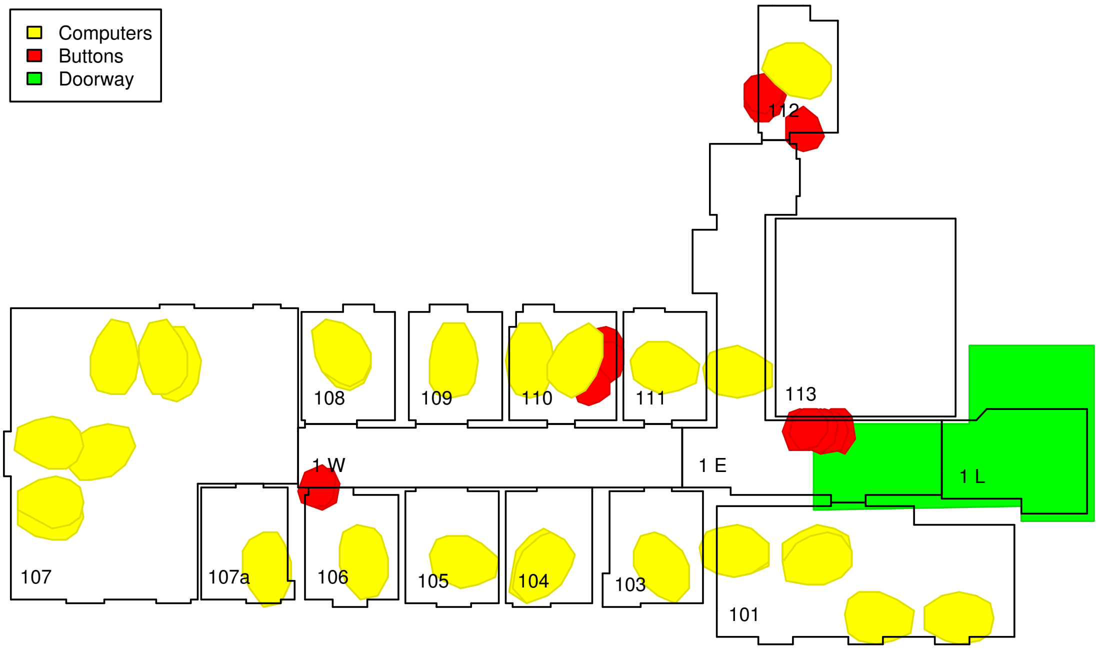


Figure 7.1: Layout of the mix zone. Each coloured area represents an application zone; all other areas represent the mix zone.

tone) every thirty minutes until the token is returned. The user will not notice the audible signal whilst at the gym,[5] but will be reminded on return.

Sentient scanner: a user can click one of several buttons positioned in hallway 1E to control a "sentient" scanner. Controls allow selection of sheet feeder and image format; the scanned data are then delivered to the user via email.

Automatic door opening: when a user approaches the (locked) internal door between the stairwell and main corridor, the automatic door application detects their presence in the doorway zone and unlocks the door.

7.2 Mix zone model performance

By using the location server as an anonymising proxy each of the event-based applications described in Section 7.1.1 can be modified to function without knowledge of the true identity of the user generating the location information. Some applications require session state to function. For example, the sentient scanner is controlled by several buttons in hallway 1E and it requires session state pseudonymity over the region of all these buttons (the application zone) in order to correlate successive button clicks of the user. If the user wishes to retain the same scanner

[5] Bat audio feedback is controlled centrally via a radio interface from the controlling system to the Bat. Therefore when the Bat is not locatable, or when placed in a specially designated "quiet zone" it will not provide any audio alerts.

settings between visits, then a fixed state pseudonym is required. When a user interested in using the sentient scanner enters the scanner's application zone, the location server assigns a pseudonym to the user's location events and passes these events via an event-based callback mechanism to the application. The application processes these location events (effecting the appropriate controls on the scanner hardware) and emails the user any scanned images using the location server as an anonymising proxy to forward the email addressed to the pseudonym to the correct underlying user identity.

Using pseudonyms to anonymise location information prevents applications from directly discovering user identity. However other information (particularly in the content of scanned documents or computer desktops) will provide further information which may enable an attacker to link a pseudonym with an underlying user (this issue was first discussed in Section 4.4.2); some of the information contained within the stream of data transferred between these applications and anonymising proxy might be *more* privacy invasive than the release of the current mapping between user identity and pseudonym (consider the scenario where a VNC desktop is currently displaying a user's hospital appointment or on-line banking details). Notwithstanding that, the choice of applications here may not be ideal, but the availability of the data at least permits a demonstration of how the mix zone model can be applied to a real dataset.

Some of the computer zones shown in Figure 7.1 are also simple home locations. For example, the location of the computer zone in Room 111 contains the home location shown in Figure 4.6. These home locations are preserved in the location privacy analysis which follows since user movements are affected by the presence or absence of location-aware applications; removing the application zones does not remove their effect on the underlying data patterns.

An untrusted application is able to determine (with a high degree of certainty) the mapping between a pseudonym and employee if the pseudonym is positioned within a home location. A more advanced location server could use the techniques presented in Section 4.3.3 to prevent application zones from being registered in home locations. However, complex home locations may span multiple application zones, and therefore removal of all home locations is a very difficult task. The number of complex home locations in the dataset may be reduced due to the spatial restrictions imposed by the requirement for the registration of (relatively small) application zones. Nevertheless some complex home locations may remain, and detecting them in an automated fashion is difficult. Thankfully, restricting the coverage area of an untrusted application limits the scope of the privacy invasion to the geographical limits of the application zone, provided that an attacker cannot link together movements of users between application zones.

Chapter 5 argued that the anonymity set is an inadequate measure of the anonymity in the mix zone model (and therefore a poor measure of location privacy). This deficiency can be demonstrated using the AT&T February 2002 data. The update period of the bat system was adjusted by post-processing the movement data to assess user location at 1, 10 and 60 second intervals. Figure 7.2 provides a bar chart of the cardinality of the anonymity set for the mix zone (all areas which are not application zones in Figure 7.1) for the measurement period. The size of the anonymity set suggests fairly low levels of anonymity for users passing through the mix zone. The event-based location-aware applications presented in the previous section would not function with a location update period of 10 or 60 seconds, however it does demonstrate that quite drastic decreases in update rate do not significantly increase the level of anonymity offered to the user; this is because the mix zone is quite large and there is a small population of users present on the first floor of the laboratory.

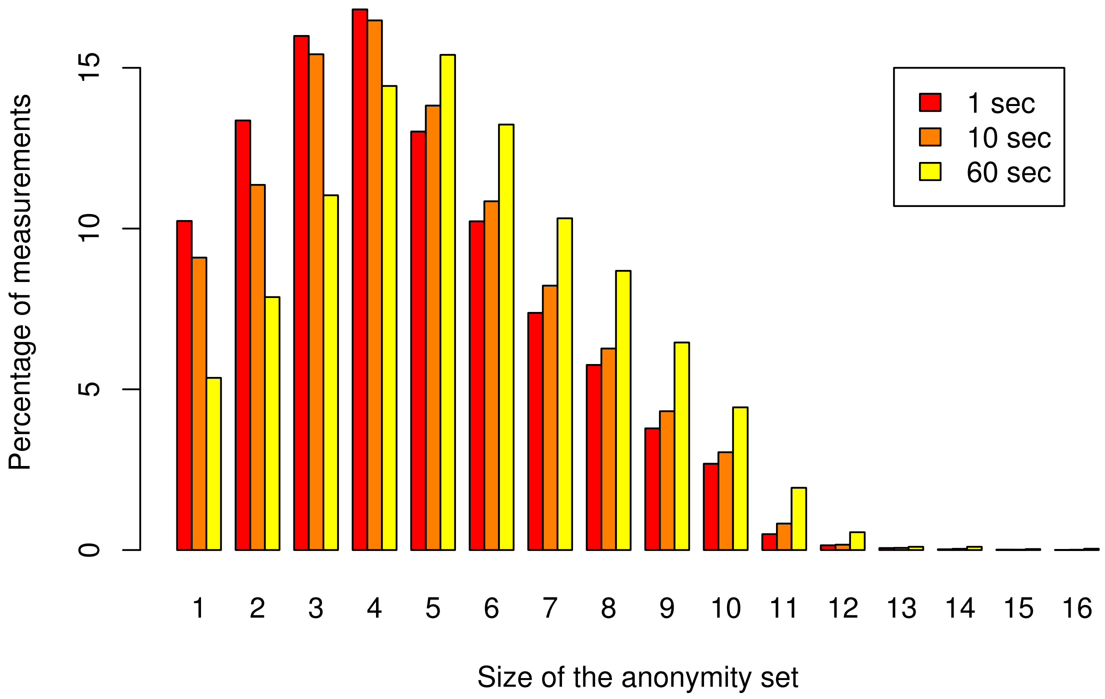


Figure 7.2: Size of anonymity set for different update periods of the location system. The mix zone for this dataset consists of the areas not defined as application zones in Figure 7.1.

The modelling techniques described in Chapter 5 can be applied to these data. Figure 7.3 describes the distribution of the time periods spent inside the mix zone by AT&T researchers during January 2002. A significant number of users spend less than two seconds in the mix zone; given the geographical layout of the application zones, it was initially expected that few sightings would result in time periods of less than five seconds. The cause of these movements can be attributed to: (1) user movement near the edge of computer zones where researchers regularly move from typing at their keyboard to reading or writing at their desk; (2) location sensor error in which the direct line of sight between a Bat and the matrix of ceiling receivers is obscured—instead a location event is calculated based on a reflection of the ultrasonic pulse from a flat surface in the environment (a computer monitor is a common reflector); and (3) since the Bat transmits an unencoded ultrasound pulse, occasionally ultrasonic transmissions present naturally in the environment are detected instead of an (obscured) Bat ultrasound transmission.

Analysis of the movement data for the mix zone reveals strong correlations between incoming and outgoing movements, particularly for short time spans; this is as one might expect from a mix zone with an appreciable distance across it. Figure 7.4 provides a view of the movement matrix for several shorter time periods constructed from the movements across the mix zone; note the strong correlation between the ingress and egress events from the same computer zone.

Since strong correlations are present in the data the movement matrix provides a tighter bound on the level of anonymity experienced by the users moving through the mix zone

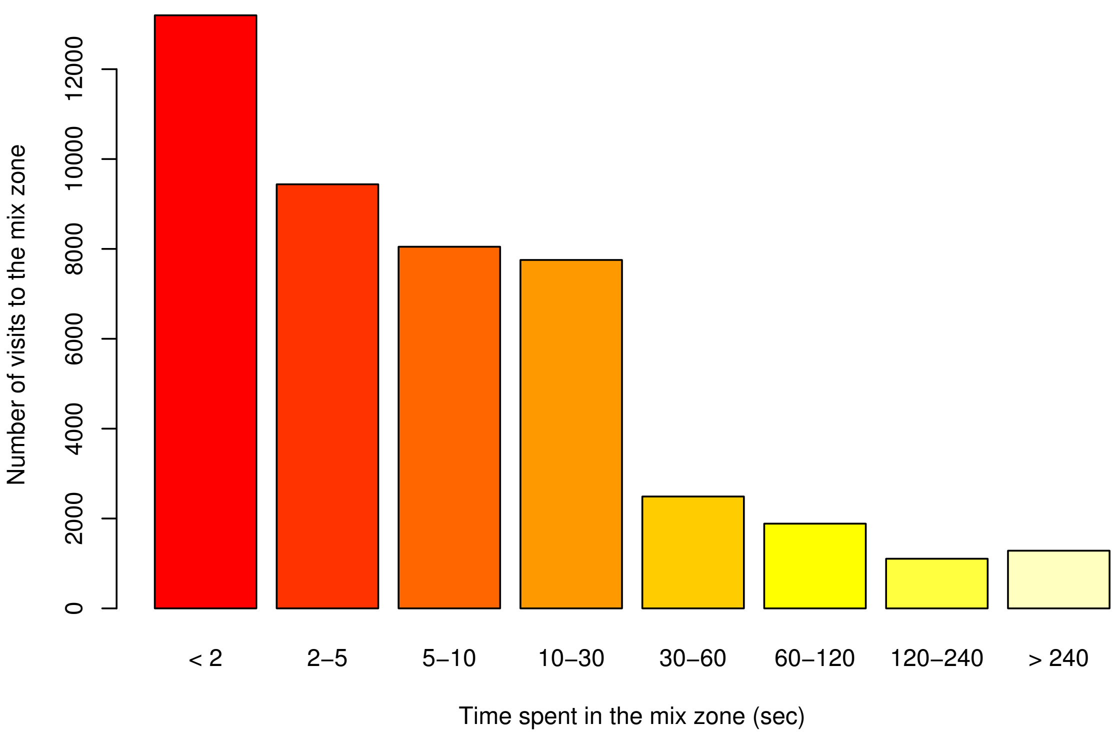


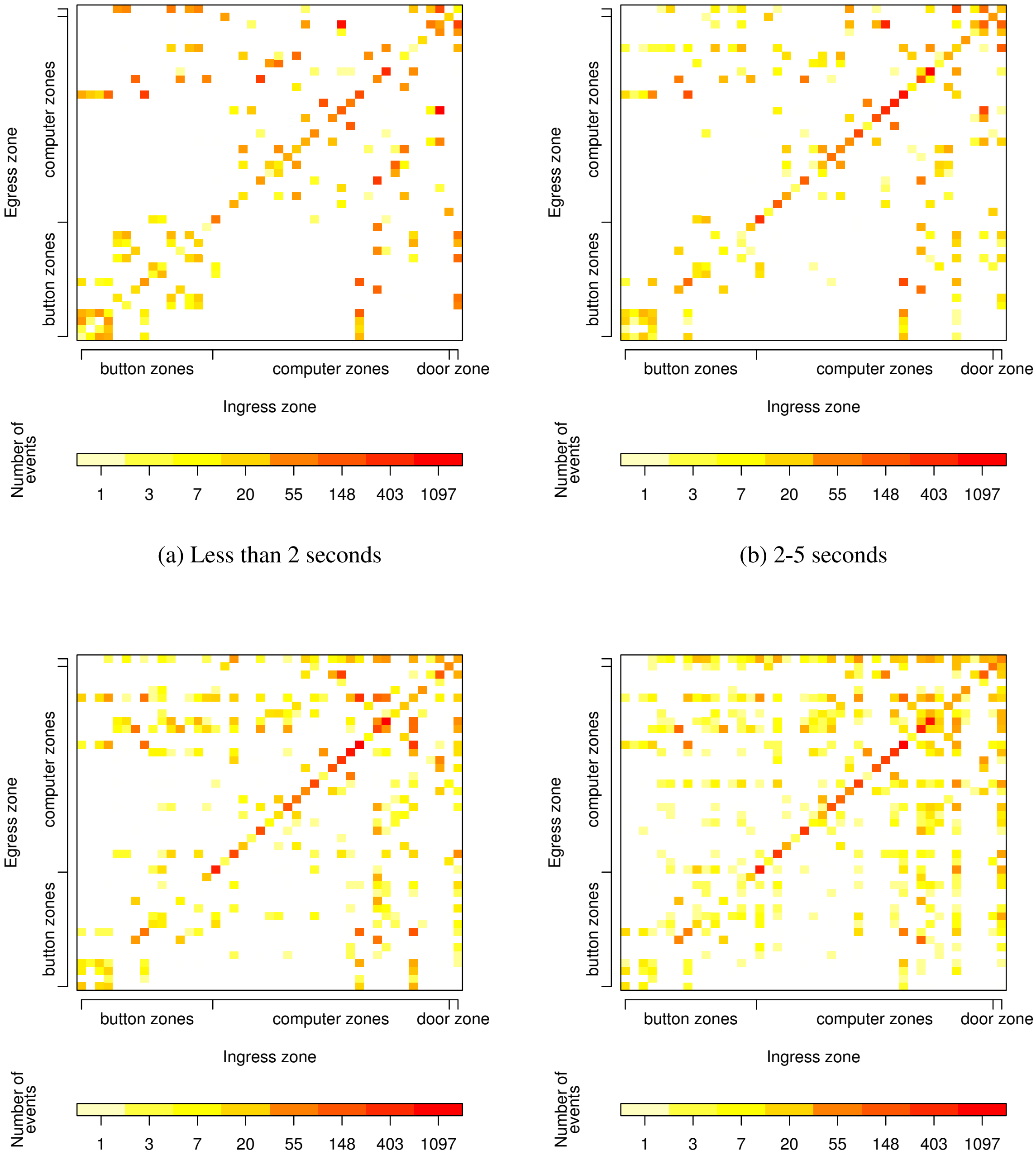
Figure 7.3: Distribution of times taken by users to cross the mix zone.

than is possible with the anonymity set measure. The February 2002 dataset can be analysed with the mix zone model to measure the level of anonymity available in the mix zone. For this comparison a movement matrix was generated using data from January 2002. The movement matrix consists of ingress and egress positions represented by the application zones depicted in Figure 7.1; the temporal components are grouped into five ranges $[0,2), [2,5), [5,10), [10,30), [30,\infty]$.

Since the mix zone is quite large, it rarely empties completely, and therefore the bipartite graph representing the movements of users into and out of the mix zone is partially evaluated by assuming all users leave within four minutes of entry (see Section 5.6 for more details). Location movement data were replayed so that the number of bipartite matches found (using the heuristic described in Algorithm 5.4) was limited as if location events had occurred in real-time.

One week of the February 2002 dataset was analysed with the mix zone model using a movement matrix constructed from data recorded in January 2002. More than 83% of movements across this mix zone were successfully followed by selecting the most likely ingress pseudonym for any given egress pseudonym using the methods described in Section 5.5. On many occasions more than one successive movement by the same underlying user could be successfully predicted. A bar chart of the number of (non-zero) successive correct correlations of ingress pseudonym with egress pseudonym is shown in Figure 7.5.

The level of location privacy afforded by using the mix zone model with the application



(a) Less than 2 seconds

(b) 2-5 seconds

(c) 5-10 seconds

(d) 10-30 seconds

Figure 7.4: Movement matrix for the first floor of AT&T Labs Cambridge using data collected during February 2002.

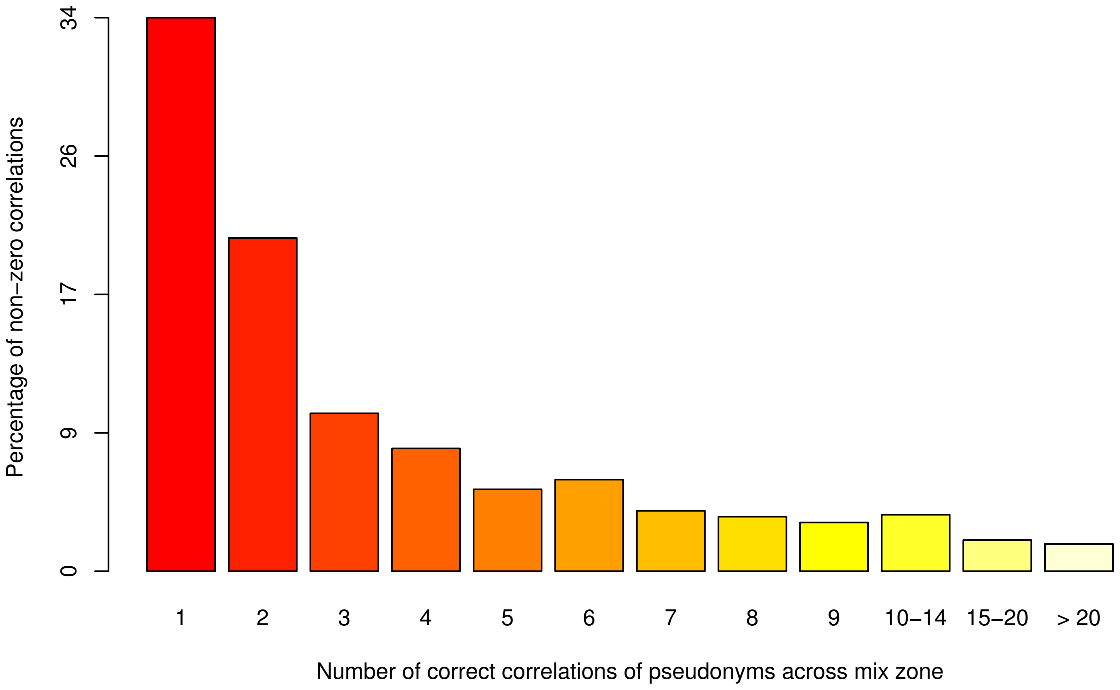


Figure 7.5: Bar chart of the number of successive ingress and egress pseudonyms successfully correlated.

zones outlined in Figure 7.1 is poor. This is because the user population is low and fairly static, and therefore making predictions concerning the movement of individuals is relatively easy. Large numbers of successive correlations of ingress pseudonyms with egress pseudonyms increase the likelihood that an attacker can determine a complex home location for the user being tracked.

7.3 Variable quality model performance

In this section the performance of the variable quality model is assessed using location data available from the Active Bat system for February 2002. In order to apply the variable quality model to these data the coordinate data provided by the Active Bat must be transformed into container-based location information. Therefore the Bat data were preprocessed to reduce the update period of the location events to 1 second with spatial containers of 10 cm × 10 cm. Therefore our basic quanta of space-time, or cuboid, measures 10 cm × 10 cm × 1 second.

Spatial accuracy reduction is performed as described in Chapter 6 using Algorithm 6.2. Since the user population is quite small, only the very upper levels of the R-tree appear in the released dataset. Figure 7.6 shows the portions of the R-tree which appeared in the spatially reduced dataset together with a bar chart which depicts the level of spatial reduction applied to meet the anonymity set size constraint of $k \in \{2, 3, 4, 5\}$. (A very small fraction of location events were released under the containers Room 112, Hall 1E, Room 101 and Hall 1L. These do not do not feature in the bar chart in order improve clarity of presentation.) In a small number of cases there were less than k people on the entire floor and therefore a reading of “no data” was recorded.

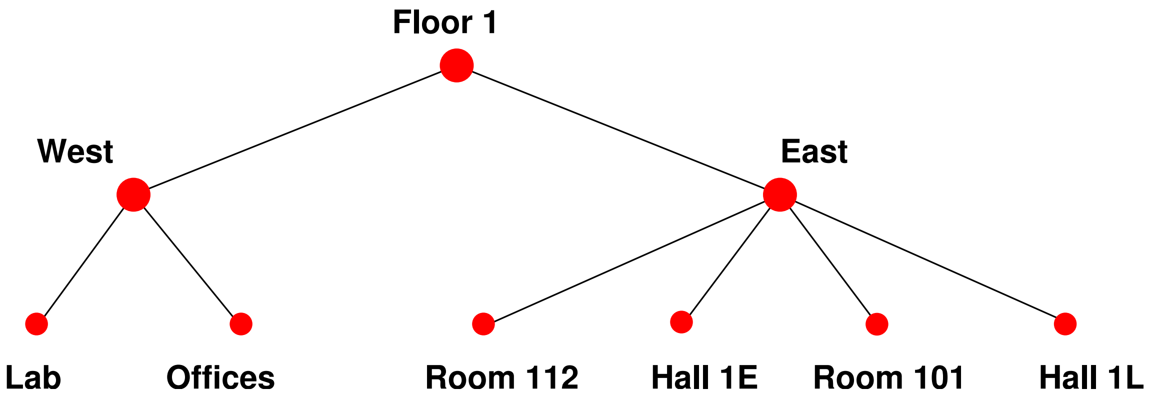
The level of accuracy reduction required to achieve k-anonymity is quite severe. Some applications, such as “Where is the nearest working printer?” still work, however other applications may not function. Such severe spatial reduction occurs because the user population is sparse; in particular, the prevention of a negation attack (see Section 6.2.2) requires large spatial reductions even when several people are in close proximity.

Temporal reduction was performed on the data for anonymity set sizes of $k \in \{2, 3, 4, 5\}$. Figure 7.7 shows the mean value of temporal reduction required to meet the minimum level of k-anonymity for different spatial regions. The minimum level of temporal reduction was just under two hours when $k = 2$; the maximum time taken for temporal reduction was 19 days. Some portions of the environment do not receive any sightings and therefore there is no coverage in those regions whatsoever.

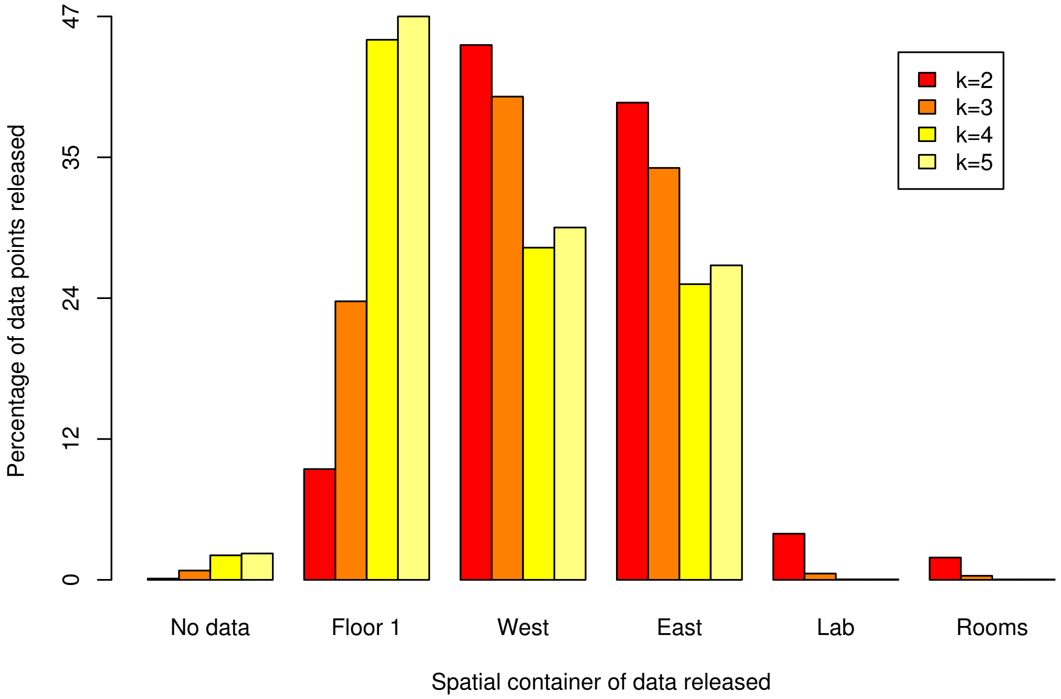
The temporally reduced data can be used to provide some useful applications. For example, the data might be used to maintain a model of the position of obstacles in the environment. In the office environment, furniture is not stationary; chairs, tables and cupboards are occasionally moved by occupants. Therefore the world model which catalogues the position of these objects needs maintenance. Harle and Hopper demonstrate techniques to detect the position of office furniture [42]; data subject to moderate temporal reduction still support this technique. The availability of particular routes or pathways through the building can also be detected. Harle and Hopper describe [43] how a Voronoi diagram [6] can be constructed using movement data from Bat location data. Such a diagram could be used to aid the movement of autonomous robots around the building or deliver fire escape directions to visitors in the case of an emergency.

7.4 Summary

This chapter has applied the variable quality model and the mix zone model to location data from the Active Bat system. The level of location privacy available through the mix zone model is low; this is because the population of users within the first floor of AT&T was small and the movement of users is low, since researchers spent the majority of time in front of their desks. For this reason the level of spatial and temporal reduction in the variable quality model is also high.

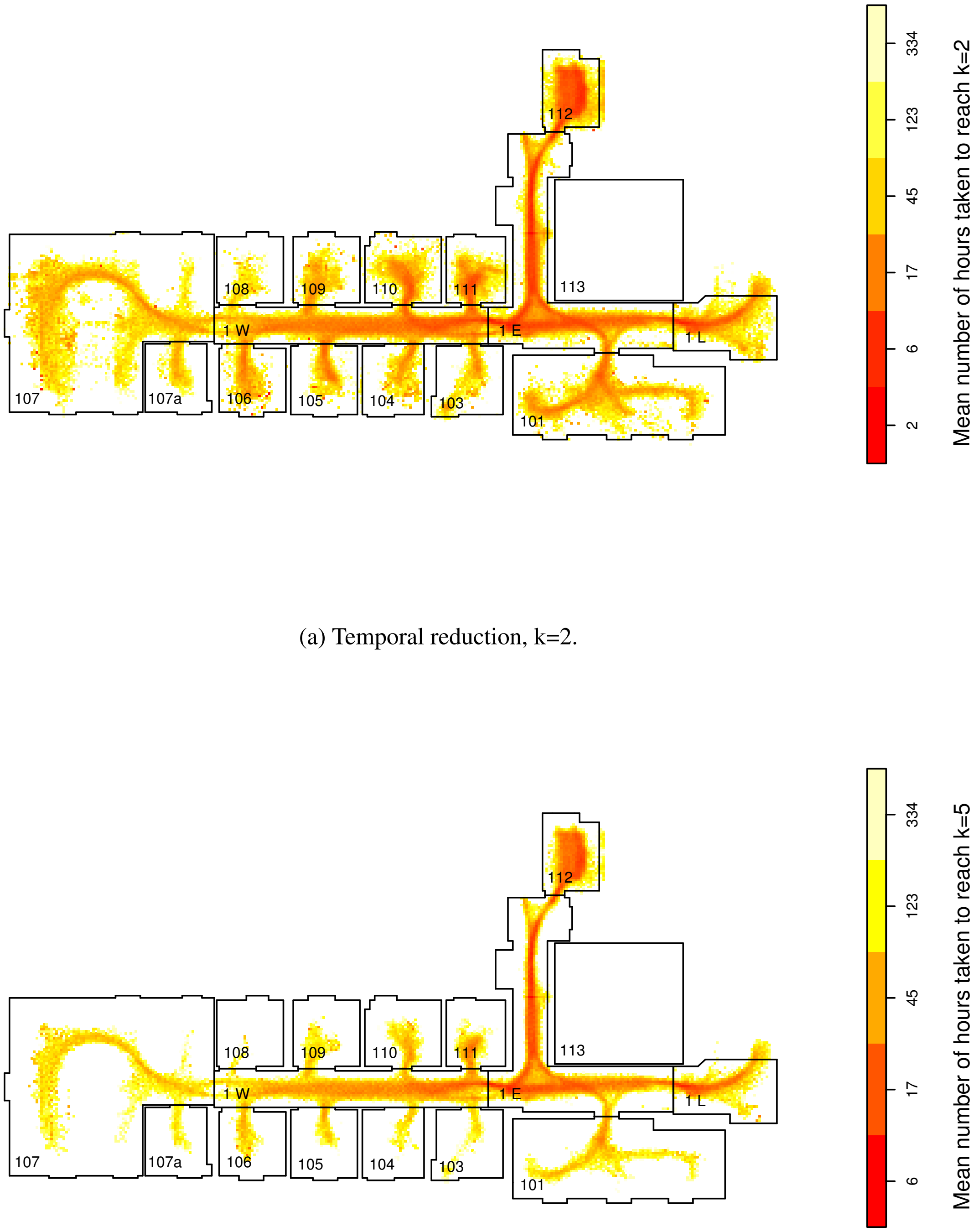


(a) Upper portion of the R-tree representing AT&T



(b) Spatial reduction of location

Figure 7.6: The upper portion of the R-tree used to represent the hierarchy of spatial regions for the first floor of AT&T. The bar chart of the level of spatial reduction applied to location events is shown in Figure (b).



(a) Temporal reduction, k=2.

(b) Temporal reduction, k=5.

Figure 7.7: Spatial distribution of the level of temporal reduction applied to the location data. Figure (a) shows the level of temporal reduction when $k = 2$; Figure (b) shows the level of temporal reduction for $k = 5$.

Chapter 8

Conclusions

> "*You have zero privacy anyway—get over it.*"
> —Scott McNealy,[1] 1999.

This dissertation has explored the feasibility of using anonymisation to enable location privacy in ubiquitous computing. Using anonymisation to protect location privacy has been shown to possess a number of potential advantages compared with access control: (1) users may prefer to remain anonymous when interacting with certain services; (2) configuration of access control parameters can be difficult and error prone and thus run counter to the aims of ubiquitous computing (namely that of exhibiting low cognitive load and automation of tasks whenever possible); and (3) anonymising location places applications outside the TCB and therefore increases our confidence in the protection of location privacy.

Not all applications can be written to use anonymised (or indeed pseudonymised) location data; therefore access control methods are still required in particular instances. However anonymised location data can enable a large class of location-aware applications. A significant subset of this class of applications can also provide services to the user with reduced temporal or spatial accuracy of data. Two typical types of location-aware application which are suited to location privacy protection through anonymisation were identified: (1) applications which are only interested in one or more restricted coverage areas, and (2) applications which do not require accurate temporal and spatial accuracy but only one or the other.

The mix zone model was developed to enable location privacy through anonymity for applications which function with restricted coverage areas. A metric based on information theory was developed to measure the likelihood an attacker could track the movement of users between application zones and therefore invade the location privacy of users.

The variable quality model was developed to enable location privacy through anonymity for applications which continue to function with either temporally or spatially reduced accuracy. Location data are released under the condition that every user shares the temporal and spatial parameters of the location event with at least $k - 1$ other users. Therefore the value of k, the size of the anonymity set, is used as the metric of location privacy.

The mix zone model and variable quality model were applied to location-aware applications using the Active Bat system. The level of location privacy obtained was low, however the analysis demonstrated that the mix zone model and variable quality model can be applied to real-world data.

[1] `http://www.wired.com/news/politics/0,1283,44255,00.html`

The difficultly with using anonymisation to protect location privacy is in the very act of data anonymisation itself. In short, anonymisation is hard. This is, in part, because a model of the knowledge of an attacker is required in order to anonymise information successfully, yet providing an accurate model is often difficult. Therefore it is necessary to overestimate the power of an attacker in order to provide a reasonable assurance of safety. In addition, it can be hard to remove subtle clues as to the underlying user identity represented by high-accuracy location data; for example, removing complex home locations from a dataset is very problematic in the general case. Finally, even relatively simple security models can quickly become computationally intractable, and therefore reliance is necessarily placed on heuristics and lower-bound estimates.

8.1 Future work

Location privacy through anonymity has the potential to be more successful for users in outdoor regions than indoor areas since the user population is larger and more mobile outdoors than in an indoor environment. Currently, quantifying this difference is difficult—access to fine-grained location data of users moving through a large outdoor region is not yet generally available. This situation will change. Fine-grained location information is likely to be available to operators of mobile phone networks in the near future; it would be interesting to apply the techniques presented in this dissertation to this dataset.

Providing a quantitative measure of anonymity for the mix zone model or variable quality model currently requires a trusted location server which receives location events for a specific group of users; a user's anonymity is then measured with reference to the location events of all other users in the group. In an user-controlled or user-mediated architecture, location data concerning the movements of other users in the near vicinity may not be available to the location server. However, some people may not wish to place trust in any one third-party entity and may want to use a user-controlled or user-mediated architecture. A method of enabling a quantitative measure of location privacy in this scenario is desirable.

Enhancements to the mix zone model could enable a location server to use historical movement data to estimate the likely change in the level of mixing if the registration of a new application is accepted; similar techniques could be used to provide feedback to users when registering new application zones (for example, if a user wishes to register for a new application then the location server can use data detailing their past movement records to tell the user what the likely levels of mixing obtained between their existing applications and the new application are going to be).

8.2 Outlook

Computing systems of the future are going to gather an ever increasing amount of personal information in order to automate tasks for people and reduce the cognitive load placed on users. Much of the personal information required is considered by many to be private and therefore users will not want it to be made publicly available, or at least not publicly attributable to them. Therefore there appears to be a clash between the need for personal data in order to manage the growing complexity inherent in ubiquitous computing and the need to preserve the privacy of users of ubiquitous computing technologies.

Technology is not privacy neutral—the design of the architecture to support ubiquitous computing applications will have a large impact on the privacy available to users. Privacy, like security, is hard to retrofit to a design, and therefore in order to preserve user privacy it is critical that the design of the architecture to support ubiquitous computing is carried out carefully.

Appendix A

Entropy for collective mixing always increases

Measurement of the level of collective mixing in the mix zone model may consider as many as $i^{\underline{e}}$ possible matches between i ingress pseudonyms and e egress pseudonyms. The algorithm presented in Section 5.3.2 iteratively determines all possible matchings $m_0, m_1, \ldots, m_{n-1}$. Let p_k represent the probability of the kth match, $P(m_k)$. The algorithm presented in Section 5.3.2 guarantees that p_0 is the most probable match (since the LAPJV algorithm always returns the most likely match) and all matches found have a non-zero probability, therefore:

$$p_0 > p_i > 0 \quad where \quad i = 1, \ldots, n-1. \tag{A.1}$$

Since the probability of the kth match represents the absolute probability of e ingress pseudonyms exiting as e egress pseudonyms with respect to match m_k then:

$$\sum_{i=0}^{n-1} p_i \leq 1 \tag{A.2}$$

On the kth iteration of the lazy matching algorithm, an upper-bound estimate of the probability of the mapping between ingress and egress pseudonyms is calculated as:

$$q_j^{(k)} \overset{def}{=} \frac{p_j}{\sum_{i=0}^{k} p_i} \tag{A.3}$$

Therefore the entropy of the mix zone on finding the kth match is:

$$\begin{aligned} H_k &= -\sum_j \frac{p_j}{\sum_i p_i} \log\left(\frac{p_j}{\sum_i p_i}\right) \\ &= -\sum_j q_j^{(k)} \log q_j^{(k)} \end{aligned} \tag{A.4}$$

THEOREM 3 *For all values of k in the range* $0, \ldots, n-1$, $H_k < H_{k+1}$.

PROOF The theorem can be proved by induction. First, start with the base case:

$$\begin{aligned}
H_0 &< H_1 \\
-\frac{p_0}{p_0}\log\left(\frac{p_0}{p_0}\right) &< -\sum_{j=0}^{1}\frac{p_j}{p_0+p_1}\log\left(\frac{p_j}{p_0+p_1}\right) \\
0 &< -\sum_{j=0}^{1}\frac{p_j}{p_0+p_1}\log\left(\frac{p_j}{p_0+p_1}\right)
\end{aligned}$$

which is true since the constraints in Equations A.1 and A.2 ensure p_0 and p_1 are in the range $(0, 1)$. Step case:

$$\begin{aligned}
H_k &< H_{k+1} \\
-\sum_{j=0}^{k} q_j^{(k)}\log q_j^{(k)} &< -\sum_{j=0}^{k+1} q_j^{(k+1)}\log q_j^{(k+1)} \\
&< -q_{k+1}^{(k+1)}\log q_{k+1}^{(k+1)} - \sum_{j=0}^{k} q_j^{(k+1)}\log q_j^{(k+1)}
\end{aligned}$$

Therefore in order to demonstrate $H_k < H_{k+1}$ it is sufficient to show:

$$-q_j^{(k)}\log q_j^{(k)} \leq -q_j^{(k+1)}\log q_j^{(k+1)} \tag{A.5}$$

for all j in range $0, \ldots, k$. Note that the less than operator becomes less than or equals because there is an additional (positive) term $-q_{k+1}^{(k+1)}\log q_{k+1}^{(k+1)}$ in H_{k+1} not included in the term by term comparison in Equation A.5.

We can prove $H_k < H_{k+1}$ by rearranging Equation A.5:

$$\begin{aligned}
-q_j^{(k)}\log q_j^{(k)} &\leq -q_j^{(k+1)}\log q_j^{(k+1)} \\
q_j^{(k)}\log q_j^{(k)} &\geq q_j^{(k+1)}\log q_j^{(k+1)} \\
q_j^{(k)}\log\frac{p_j}{\sum_{i=0}^{k} p_i} &\geq q_j^{(k+1)}\log\frac{p_j}{p_{k+1}+\sum_{i=0}^{k} p_i} \\
q_j^{(k+1)}\log(p_{k+1}+\sum_{i=0}^{k} p_i) - q_j^{(k)}\log(\sum_{i=0}^{k} p_i) &\geq \left(q_j^{(k+1)} - q_j^{(k)}\right)\log p_j
\end{aligned}$$

Since $\log(p_{k+1}+\sum_{i=0}^{k} p_i) > \log(\sum_{i=0}^{k} p_i)$:

$$\begin{aligned}
\log(\sum_{i=0}^{k} p_i) &\geq \log p_j \\
\sum_{i=0}^{k} p_i &\geq p_j
\end{aligned}$$

which is true since, due to the constraint in Equation A.2, $p_0 > p_j$ for all j in $1 \ldots k$.

■

Bibliography

[1] M. D. Addlesee, A. H. Jones, F. Livesey, and F. S. Samaria. ORL active floor. *IEEE Personal Communications*, 4(5):35–41, October 1997. http://ieeexplore.ieee.org/iel4/98/13641/00626980.pdf. (Ref: p. 19.)

[2] Mike Addlesee, Rupert Curwen, Steve Hodges, Joe Newman, Pete Steggles, Andy Ward, and Andy Hopper. Implementing a sentient computing system. *Computer*, 34(8):50–56, 2001. http://dx.doi.org/10.1109/2.940013. (Ref: p. 27, 29.)

[3] Noha Adly, Pete Steggles, and Andy Harter. SPIRIT: A resource database for mobile users. In *Proceedings of ACM CHI'97 Workshop on Ubiquitous Computing*, March 1997. (Ref: p. 14.)

[4] Ross Anderson. *Security Engineering: A guide to building dependable systems*. Wiley, 2001. (Ref: p. 36, 37, 39, 50, 66.)

[5] Hisashi Aoki, Bernt Schiele, and Alex Pentland. Realtime personal positioning system for wearable computers. In *Proceedings of the Third IEEE International Symposium on Wearable Computers*, pages 37–43. IEEE Computer Society, 1999. (Ref: p. 24.)

[6] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, September 1991. http://doi.acm.org/10.1145/116873.116880. (Ref: p. 117.)

[7] Ronald T. Azuma. A survey of augmented reality. *Presence*, 6(4):355–385, August 1997. http://mitpress.mit.edu/catalog/item/default.asp?tid=825&ttype=6. (Ref: p. 16, 18.)

[8] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*, pages 775–784, March 2000. http://ieeexplore.ieee.org/iel5/6725/18009/00832252.pdf. (Ref: p. 20.)

[9] D. Elliot Bell and Leonard J. LaPadula. Secure computing systems: Mathematical foundations and model. Technical Report ESD-TR-73-278, Mitre Corporation, April 1974. (Ref: p. 36.)

[10] Alastair R. Beresford and Frank Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 3(1):46–55, 2003. http://ieeexplore.ieee.org/iel5/7756/26614/01186726.pdf. (Ref: p. 72.)

[11] E. Briscoe and J. Carroll. Robust accurate statistical annotation of general text. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*, pages 1499–1504, Las Palmas, Canary Islands, May 2002. (Ref: p. 8.)

[12] P. J. Brown. The stick-e document: A framework for creating context-aware applications. *Electronic Publisher*, 8(2):259–272, January 1996. http://www.cs.ukc.ac.uk/pubs/1996/396. (Ref: p. 29, 65.)

[13] Rainer E. Burkard and Eranda Çela. *Handbook of Combinatorial Optimization*. Kluwer, 1999. (Ref: p. 74.)

[14] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):66–75, 1988. (Ref: p. 46.)

[15] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981. http://doi.acm.org/10.1145/358549.358563. (Ref: p. 45.)

[16] Keith Cheverst, Nigel Davies, Keith Mitchell, and Adrian Friday. Experiences of developing and deploying a context-aware tourist guide: The guide project. In *Proceedings of the Sixth Annual International Conference on Mobile Compting and Networking*, pages 20–31. ACM Press, 2000. http://doi.acm.org/10.1145/345910.345916. (Ref: p. 27, 65.)

[17] Jeremy R. Cooperstock, Sidney S. Fels, William Buxton, and Kenneth C. Smith. Reactive environments. *Communications of the ACM*, 40(9):65–73, 1997. http://doi.acm.org/10.1145/260750.260774. (Ref: p. 27, 65.)

[18] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The MIT Press, 1990. (Ref: p. 83.)

[19] Michael J. Covington, Wende Long, Srividhya Srinivasan, Anind K. Dey, Mustaque Ahamad, and Gregory D. Abowd. Securing context-aware applications using environment roles. In *Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies*, pages 10–20. ACM Press, 2001. http://doi.acm.org/10.1145/373256.373258. (Ref: p. 38.)

[20] S.J. Creese, M. Goldsmith, A.W. Roscoe, and I. Zakiuddin. The attacker in ubiquitous computing environments: Formalising the threat model. In *Proceedings of the Workshop on Formal Aspects in Security and Trust*, Pisa, September 2003. http://web.comlab.ox.ac.uk/oucl/research/areas/security/papers/fast.pdf. (Ref: p. 45.)

[21] Jorge R. Cuellar, John B. Morris, Deirdre K. Mulligan, Jon Peterson, and James Polk. Geopriv requirements, October 2003. http://www.ietf.org/internet-drafts/draft-ietf-geopriv-reqs-01.txt. Internet Draft, work in progress. (Ref: p. 41.)

[22] A. Dasdan and R.K. Gupta. Faster maximum and minimum mean cycle algorithms for system-performance analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(10):889–899, October 1998. http://ieeexplore.ieee.org/iel4/43/15708/00728912.pdf. (Ref: p. 83.)

[23] Diego López de Ipiña. *Visual Sensing and Middleware Support for Sentient Computing*. PhD thesis, University of Cambridge, January 2003. (Ref: p. 14.)

[24] Diego López de Ipiña, Paulo Mendonça, and Andy Hopper. TRIP: a low-cost vision-based location system for ubiquitous computing. *Personal and Ubiquitous Computing*, 6(3):206–219, May 2002. `http://dx.doi.org/10.1007/s007790200020`. (Ref: p. 24.)

[25] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. In *Computer Vision and Pattern Recognition*, 1999. (Ref: p. 24.)

[26] Anind K. Dey and Gregory D.Abowd. Towards a better understanding of context and context-awareness. In *Proceedings of the CHI 2000 Workshop on "The What, Who, Where, When, Why and How of Context-Awareness"*, 2000. (Ref: p. 14.)

[27] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. `http://ieeexplore.ieee.org/iel5/18/22693/01055638.pdf`. (Ref: p. 40.)

[28] C. Drane, M. Macnaughtan, and C. Scott. Positioning GSM telephones. *IEEE Communications Magazine*, 36(4):46–54, April 1998. `http://ieeexplore.ieee.org/iel5/35/14678/00667413.pdf`. (Ref: p. 21.)

[29] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. SPKI certificate theory. Technical Report 2693, IETF, September 1999. `http://www.ietf.org/rfc/rfc2693.txt`. (Ref: p. 41.)

[30] Carl M. Ellison. The nature of a useable PKI. *Computer Networks*, 31(8):823–830, 1999. (Ref: p. 41.)

[31] John Fawcett and Peter Robinson. Adaptive routing for road traffic. *IEEE Computer Graphics Applications*, 20(3):46–53, May 2000. `http://dx.doi.org/10.1109/38.844372`. (Ref: p. 27.)

[32] John K. Fawcett. *Sentient Computing: A Universal Framework for Spatial Data*. PhD thesis, University of Cambridge, 2004. (Ref: p. 29.)

[33] Steven Feiner, Blair MacIntyre, Tobias Hollerer, and Anthony Webster. A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. In *Proceedings of the First IEEE International Symposium on Wearable Computers*, page 74. IEEE Computer Society, 1997. (Ref: p. 27.)

[34] George W. Fitzmaurice. Situated information spaces and spatially aware palmtop computers. *Communications of the ACM*, 36(7):39–49, July 1993. `http://doi.acm.org/10.1145/159544.159566`. (Ref: p. 27.)

[35] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Huges. *Computer Graphics—Principles and Practice*. Addison-Wesley, 1996. (Ref: p. 103.)

[36] C. L. Forgy. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19:17–37, 1982. (Ref: p. 30.)

[37] Zvi Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys*, 18(1):23–38, March 1986. http://doi.acm.org/10.1145/6462.6502. (Ref: p. 74.)

[38] Ivan Getting. The Global Positioning System. *IEEE Spectrum*, 30(12):36–47, December 1993. http://ieeexplore.ieee.org/iel3/6/6726/00272176.pdf. (Ref: p. 21.)

[39] Christian Gloor, Pascal Stucki, and Kai Nagel. Hybrid techniques for pedestrian simulations. In *Proceedings of the Swiss Transportation Research Conference*, 2004. http://www.strc.ch/pdf_2004/Gloor_Stucki_Nagel_HybridTechniquesPedestrianSimulations_STR.pdf. (Ref: p. 110.)

[40] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the ACM/USENIX International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2003. (Ref: p. 50, 65, 91.)

[41] Marco Gruteser, Graham Schelle, Ashish Jain, Rick Han, and Dirk Grunwald. Privacy-aware location sensor networks. In *Proceedings of HotOS*, 2003. (Ref: p. 96.)

[42] Robert K. Harle and Andy Hopper. Building world models by ray-tracing within ceiling-mounted positioning systems. In *Proceedings of the Fifth International Conference on Ubiquitous Computing (UbiComp)*, pages 1–17, October 2003. http://www.springerlink.com/index/DJEN3T4NB5GXCVRN.pdf. (Ref: p. 28, 58, 60, 65, 117.)

[43] Robert K. Harle and Andy Hopper. Using personnel movements for indoor autonomous environment discovery. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 125–132, March 2003. http://ieeexplore.ieee.org/iel5/8487/26747/01192734.pdf. (Ref: p. 117.)

[44] A. Harter and A. Hopper. A distributed location system for the active office. *IEEE Network*, 8(1):62–70, January 1994. http://ieeexplore.ieee.org/iel3/65/6572/00260080.pdf. (Ref: p. 23.)

[45] M. Hazas and A. Ward. A high performance privacy-oriented location system. In *Proceedings of the IEEE International Conference Pervasive Computing and Communications (PerCom)*, pages 216–223, 2003. http://csdl.computer.org/dl/proceedings/percom/2003/1893/00/18930216.pdf. (Ref: p. 22.)

[46] Mike Hazas and Andy Ward. A novel broadband ultrasonic location system. In *Proceedings of the Fourth International Conference on Ubiquitous Computing (UbiComp)*, volume 2498, pages 264–280. Springer-Verlag, September 2002. (Ref: p. 22.)

[47] R Headon. Movement awareness for a sentient environment. In *Proceedings of the IEEE International Conference Pervasive Computing and Communications (PerCom)*,

pages 99–106, 2003. http://csdl.computer.org/dl/proceedings/percom/2003/1893/00/18930099.pdf. (Ref: p. 19.)

[48] Dirk Helbing, Illes Farkas, and Tamas Vicsek. Simulating dynamical features of escape panic. *Nature*, 407:487–490, September 2000. http://dx.doi.org/10.1038/35035023. (Ref: p. 110.)

[49] Urs Hengartner and Peter Steenkiste. Protecting access to people location information. In *Proceedings of the First International Conference on Security in Pervasive Computing*, March 2003. (Ref: p. 42.)

[50] Jeffery Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001. http://dx.doi.org/10.1109/2.940014. (Ref: p. 18.)

[51] William A. Hoff, Khoi Nguyen, and Torsten Lyon. Computer vision-based registration techniques for augmented reality. *Intelligent Robots and Computer Vision XV*, 2904:538–548, November 1996. (Ref: p. 27.)

[52] John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal of Computing*, 2(4):225–231, December 1973. (Ref: p. 74.)

[53] Andy Hopper. 1999 Sentient Computing. *Phil. Trans. R. Soc. Lond.*, 358(1):2349–2358, 2000. (Ref: p. 14, 17.)

[54] Paul Horn. Autonomic computing. http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf. (Ref: p. 14.)

[55] Richard Hull, Philip Neaves, and James Bedford-Roberts. Towards situated computing. In *Proceedings of the First IEEE International Symposium on Wearable Computers*, pages 146–153. IEEE Computer Society Press, IEEE Computer Society, October 1997. (Ref: p. 19, 29.)

[56] Michal Irani, Benny Rousso, and Shmuel Peleg. Robust recovery of ego-motion. In *Computer Analysis of Images and Patterns*, pages 371–378, 1993. (Ref: p. 25.)

[57] Alon Itai, Michael Rodeh, and Steven Tanimoto. Some matching problems for bipartite graphs. *Journal of the Association for Computing Machinery*, 25(4):517–525, October 1978. http://doi.acm.org/10.1145/322092.322093. (Ref: p. 77, 79, 80, 81.)

[58] ITU-T. X.509 data networks and open system communications directory, February 2001. (Ref: p. 40.)

[59] Ian. W. Jackson. *Who goes here? Confidentiality of location through anonymity*. PhD thesis, Cambridge University, 1998. (Ref: p. 47.)

[60] Clifford Neuman Jennifer Steiner and Jeffrey I. Schiller. Kerberos: An authentication service for open network systems. In *Proceedings of the Winter Usenix Conference*, pages 191–201, February 1988. (Ref: p. 36.)

[61] Bolan Jiang and Ulrich Neumann. Extendible tracking by line auto-calibration. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR)*, pages 97–105, October 2001. (Ref: p. 25.)

[62] Roy Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Journal of Computing*, 38:325–340, 1987. (Ref: p. 75.)

[63] Richard M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, 23:309–311, 1978. (Ref: p. 83.)

[64] Hirokazu Kato and Mark Billinghurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proceedings of the International Workshop on Augmented Reality (IWAR)*, pages 85–94, October 1999. (Ref: p. 24.)

[65] Eleftheria Katsiri, Jean Bacon, and Alan Mycroft. An extended publish/subscribe protocol for transparent subscriptions to distributed abstract state in sensor-driven systems using abstract events. In *Proceedings of Distributed Event-Based Systems*, 2004. (Ref: p. 30.)

[66] Dogan Kesdogan, Hannes Federrath, Anja Jerichow, and Andreas Pfitzmann. Location management strategies increasing privacy in mobile communication systems. *Information systems security: facing the information society of the 21st century*, pages 39–48, May 1996. (Ref: p. 46.)

[67] Tim Kindberg, John Barton, Jeff Morgan, Gene Becker, Debbie Caswell, Philippe Debaty, Gita Gopal, Marcos Frid, Venky Krishnan, Howard Morris, John Schettino, Bill Serra, and Mirjana Spasojevic. People, places, things: Web presence for the real world. Technical Report HPL-2001-279, Hewlett Packard, 2001. http://www.hpl.hp.com/techreports/2001/HPL-2001-279.html. (Ref: p. 29.)

[68] Michael Kirby. Australian privacy charter. http://privacy.org.au/About/PrivacyCharter.html. (Ref: p. 31.)

[69] John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, and Steve Shafer. Multi-camera multi-person tracking for easyliving. In *Proceedings of the Third IEEE International Workshop on Visual Surveillance*, pages 3–10. IEEE Computer Society, 2000. (Ref: p. 25.)

[70] H. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955. (Ref: p. 75.)

[71] Butler W. Lampson. Protection. In *Proceedings of the Fifth Princeton Symposium on Information Sciences and Systems*, pages 437–443. Princeton University, March 1971. Reprinted in ACM SIGOPS Operating Systems Review, 8(1):18–24, January 1974. (Ref: p. 36.)

[72] Marc Langheinrich. Privacy by design—principles of privacy-aware ubiquitous systems. In *Proceedings of the Third International Conference on Ubiquitous Computing (UbiComp)*, pages 273–291. Springer-Verlag, 2001. (Ref: p. 36.)

[73] Marc Langheinrich. A privacy awareness system for ubiquitous computing environments. In *Proceedings of the Fourth International Conference on Ubiquitous Computing (UbiComp)*, pages 237–245. Springer-Verlag, 2002. (Ref: p. 44.)

[74] Cedric Laurant. Privacy and human rights 2003. http://www.privacyinternational.org/survey/phr2003/. (Ref: p. 31.)

[75] Seon-Woo Lee and Kenji Mase. Incremental motion-based location recognition. In *Proceedings of The Fifth International Symposium on Wearable Computers (ISWC)*, October 2001. (Ref: p. 19.)

[76] Ulf Leonhardt and Jeff Magee. Security considerations for a distributed location service. *Journal of Network and Systems Management*, 6(1):51–70, March 1998. http://dx.doi.org/10.1023/A:1018777802208. (Ref: p. 13, 38.)

[77] Lawrence Lessig. *Code and other laws of cyberspace*. Basic Books, 1999. (Ref: p. 31, 33, 34.)

[78] David Litchfield. Hackproofing oracle application server. Technical report, NGSSoftware, 2002. (Ref: p. 59.)

[79] Kalle Lyytinen and Youngjin Yoo. Issues and challenges in ubiquitous computing. *Communications of the ACM*, 45(12):62–65, December 2002. (Ref: p. 13.)

[80] Jouni Markkula. Statistical disclosure control of small area statistics using local restricted imputation. In *Bulletin of the International Statistical Institute (52nd Session)*, Finland, 1999. http://www.stat.fi/isi99/proceedings/arkisto/varasto/mark0358.pdf. (Ref: p. 50.)

[81] Darnell J. Moore, Roy Want, Beverly L. Harrison, Anuj Gujar, and Ken Fishkin. Implementing phicons: combining computer vision with infrared technology for interactive physical icons. In *Proceedings of the Twelfth Annual ACM Symposium on User Interface Software and Technology*, pages 67–68. ACM Press, 1999. http://doi.acm.org/10.1145/320719.322585. (Ref: p. 24.)

[82] Thomas P. Moran, Eric Saund, William Van Melle, Anuj U. Gujar, Kenneth P. Fishkin, and Beverly L. Harrison. Design and technology for collaborage: collaborative collages of information on physical walls. In *Proceedings of the Twelfth Annual ACM Symposium on User Interface Software and Technology*, pages 197–206. ACM Press, November 1999. http://doi.acm.org/10.1145/320719.322602. (Ref: p. 29.)

[83] Ginger Myles, Adrian Friday, and Nigel Davies. Preserving privacy in environments with location-based applications. *IEEE Pervasive Computing*, 2(1):56–64, March 2003. http://dx.doi.org/10.1109/MPRV.2003.1186726. (Ref: p. 43.)

[84] Leonid Naimark and Eric Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self tracker. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2002. http://csdl.computer.org/dl/proceedings/ismar/2002/1781/00/17810027.pdf. (Ref: p. 26.)

[85] United Nations. Universal declaration of human rights: General assembly resolution 217a (iii), December 1948. http://www.un.org/Overview/rights.html. (Ref: p. 31.)

[86] United Nations. International covenant on civil and political rights, December 1966. http://www.unhchr.ch/html/menu3/b/a_ccpr.htm. (Ref: p. 31.)

[87] Roger M. Needham. *Distributed Systems*, chapter "Names", pages 315–327. Addison-Wesley, 1993. (Ref: p. 63.)

[88] U. Neumann and Y. Cho. A self-tracking augmented reality system. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 109–115, July 1996. (Ref: p. 27.)

[89] Ulrich Neumann and Suya You. Integration of region tracking and optical flow for image motion estimation. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, October 1998. http://csdl.computer.org/dl/proceedings/icip/1998/8821/03/882130658.pdf. (Ref: p. 25.)

[90] William M. Newman, Margery A. Eldridge, and Michael G. Lamming. Pepys: Generating autobiographies by automatic tracking. In *Proceedings of the Second European Conference on Computer-Supported Co-operative Work (CSCW)*, pages 175–188, September 1991. (Ref: p. 28, 65.)

[91] Esko Nuutila and Eljas Soisalon-Soininen. On finding the strongly connected components in a directed graph. *Information Processing Letters*, 49(1):9–14, 1994. http://dx.doi.org/10.1016/0020-0190(94)90047-7. (Ref: p. 79.)

[92] Andrew M. Odlyzko. Privacy, economics, and price discrimination on the internet. In *Proceedings of the Fifth International Conference on Electronic Commerce*, pages 355–366. ACM, ACM Press, 2003. http://doi.acm.org/10.1145/948005.948051. (Ref: p. 33.)

[93] Council of Europe. The european convention on human rights, November 1950. http://conventions.coe.int/treaty/en/Treaties/Html/005.htm. (Ref: p. 31.)

[94] Sylvia Osborn, Ravi Sandhu, and Qamar Munawer. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Transactions on Information and System Security*, 3(2):85–106, 2000. http://doi.acm.org/10.1145/354876.354878. (Ref: p. 37.)

[95] Charles B. Owen, Fan Xiao, and Paul Middlin. What is the best fiducial? In *The First IEEE International Augmented Reality Toolkit Workshop*, pages 98–105, September 2002. (Ref: p. 24.)

[96] Jason Pascoe. Adding generic contextual capabilities to wearable computers. In *Proceedings of the Second IEEE International Symposium on Wearable Computers*, pages 92–100. IEEE Computer Society, October 1998. http://ieeexplore.ieee.org/iel4/5898/15725/00729534.pdf. (Ref: p. 15, 29, 65.)

[97] Helen Petrie, Valerie Johnson, Thomas Strothotte, Steffi Fritz, Rainer Michel, and Andreas Raab. Mobic: Designing a travel aid for blind and elderly people. *Journal of Navigation*, 49(1):45–52, 1996. (Ref: p. 26, 65.)

[98] Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability and pseudonymity—a proposal for terminology. In *Designing Privacy Enhancing Technologies: Proceedings of the International Workshop on the Design Issues in Anonymity and Observability*, volume 2009 of *Lecture Notes in Computer Science*, pages 1–9, 2000. http://link.springer.de/link/service/series/0558/papers/2009/20090001.pdf. (Ref: p. 44.)

[99] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, pages 32–43. ACM Press, August 2000. http://doi.acm.org/10.1145/345910.345917. (Ref: p. 22.)

[100] Michael K. Reiter and Aviel D. Rubin. Crowds: anonymity for web transactions. *ACM Transactions Information Systems Security*, 1(1):66–92, 1998. http://doi.acm.org/10.1145/290163.290168. (Ref: p. 44.)

[101] Jun Rekimoto and Yuji Ayatsuka. Cybercode: designing augmented reality environments with visual tags. In *Proceedings of the Workshop on Designing Augmented Reality Environments*, pages 1–10. ACM Press, April 2000. http://doi.acm.org/10.1145/354666.354667. (Ref: p. 24.)

[102] Jun Rekimoto and Katashi Nagao. The world through the computer: computer augmented interaction with real world environments. In *Proceedings of the Eighth Annual ACM Symposium on User Interface and Software Technology*, pages 29–36. ACM Press, 1995. http://doi.acm.org/10.1145/215585.215639. (Ref: p. 24, 27.)

[103] Ioannis M. Rekleitis. Optical flow recognition from the power spectrum of a single blurred image. In *Proceedings of the International Conference on Image Processing*, Lausanne, Switzerland, September 1996. IEEE Signal Processing Society. (Ref: p. 25.)

[104] Miguel Ribo, Axel Pinz, and Anton L. Fuhrmann. A new optical tracking system for virtual and augmented reality applications. In *Proceedings of the IEEE Instrumentation and Measurement Technology Conference*, May 2001. (Ref: p. 23.)

[105] Tristan Richardson. Teleporting: Mobile X sessions. *The X Resource*, 13(1):133–140, 1995. (Ref: p. 27.)

[106] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood, and Andy Hopper. Virtual network computing. *IEEE Internet Computing*, 2(1):33–38, January 1998. http://dx.doi.org/10.1109/4236.656066. (Ref: p. 27, 65.)

[107] Wasinee Rungsarityotin and Thad E. Starner. Finding location using omnidirectional video on a wearable computing platform. In *Proceedings of the Fourth International Symposium on Wearable Computers*, pages 61–68. IEEE Computer Society, October 2000. http://ieeexplore.ieee.org/iel5/7125/19202/00888466.pdf. (Ref: p. 25.)

[108] Nick Ryan, Jason Pascoe, and David Morse. Enhanced reality fieldwork: the context aware archaeological assistant. In *Proceedings of Computer Applications in Archaeology*, 1997. (Ref: p. 15.)

[109] Daniel Salber, Anind K. Dey, and Gregory D. Abowd. The context toolkit: aiding the development of context-enabled applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 434–441. ACM Press, 1999. http://doi.acm.org/10.1145/302979.303126. (Ref: p. 29.)

[110] J.H. Saltzer and M. D. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9), September 1975. (Ref: p. 69.)

[111] Hanan Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys*, 16(2):187–260, 1984. http://doi.acm.org/10.1145/356924.356930. (Ref: p. 96.)

[112] Hanan Samet. *The design and analysis of spatial data structures*. Addison-Wesley, 1990. (Ref: p. 103.)

[113] Ravi Sandhu, David Ferraiolo, and Richard Kuhn. The NIST model for role-based access control: Towards a unified standard. In *Proceedings of the Fifth ACM Workshop on Role-based Access Control*, pages 47–63. ACM Press, 2000. http://doi.acm.org/10.1145/344287.344301. (Ref: p. 37.)

[114] M Satyanarayanan. A catalyst for mobile and ubiquitous computing. *Pervasive Computing*, 1(1):2–5, January 2002. http://ieeexplore.ieee.org/iel5/7756/21413/00993138.pdf. (Ref: p. 13.)

[115] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Proceedings of the Workshop on Mobile Computing Systems and Applications.*, pages 85–90, September 1993. (Ref: p. 14, 15.)

[116] H. Schulzrinne, John B. Morris, H. Tschofenig, Jorge R. Cuellar, and James Polk. Geopriv policy rules for disclosure and modification of geographic information, October 2003. http://www.ietf.org/internet-drafts/draft-ietf-geopriv-policy-00.txt. Internet Draft, work in progress. (Ref: p. 42.)

[117] David Scott, Alastair Beresford, and Alan Mycroft. Spatial security policies for mobile agents in a sentient computing environment. In *Proceedings of the Conference on Fundamental Approaches to Software Engineering (FASE)*, volume 2621 of *Lecture Notes in Computer Science*, pages 102–117. Springer-Verlag, April 2003. (Ref: p. 43.)

[118] David Scott and Richard Sharp. Developing secure web applications. *IEEE Internet Computing*, 6(6):38–45, November 2002. http://ieeexplore.ieee.org/iel5/4236/22924/01067735.pdf. (Ref: p. 59.)

[119] Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In *Proceedings of the Workshop on Privacy Enhancing Technologies (PET)*, 2002. (Ref: p. 44.)

[120] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, July 1948. http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf. (Ref: p. 44, 76.)

[121] Claude Elwood Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656–715, 1949. (Ref: p. 39.)

[122] Asim Smailagic and David Kogan. Location sensing and privacy in a context-aware computing environment. *IEEE Wireless Communications*, 9(5):10–17, October 2002. http://ieeexplore.ieee.org/iel5/7742/22372/01043849.pdf. (Ref: p. 20.)

[123] Quentin Stafford-Fraser and Peter Robinson. Brightboard: a video-augmented environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 134–141. ACM Press, 1996. http://doi.acm.org/10.1145/238386.238457. (Ref: p. 24.)

[124] T. Starner, D. Kirsh, S. Assefa, and L. Swarm. An environmentally-powered networkless location and messaging system. In *Proceedings of the International Symposium on Wearable Computing*, pages 169–170, October 1997. http://csdl.computer.org/comp/proceedings/iswc/1997/8192/00/81920169.pdf. (Ref: p. 23, 29.)

[125] Thad Starner, Bernt Schiele, and Alex Pentland. Visual contextual awareness in wearable computing. In *Proceedings of the Second IEEE International Symposium on Wearable Computers*, pages 50–57. IEEE Computer Society, 1998. http://csdl.computer.org/dl/proceedings/iswc/1998/9074/00/90740050.pdf. (Ref: p. 24.)

[126] Andrei State, Gentaro Hirota, David T. Chen, William F. Garrett, and Mark A. Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 429–438. ACM Press, 1996. http://doi.acm.org/10.1145/237170.237282. (Ref: p. 26.)

[127] Scott Stillman, Rawesak Tanawongsuwan, and Irfran Essa. A system for tracking and recognizing multiple people with multiple cameras. Technical Report GIT-GVU-98-25, Georgia Institute of Technology, 1998. (Ref: p. 25.)

[128] Douglas R. Stinson. *Cryptography: theory and practice*. CRC Press, Inc., 1995. (Ref: p. 47.)

[129] Ivan Sutherland. A head-mounted three-dimensional display. In *AFIPS*, volume 33, pages 757–764, 1968. (Ref: p. 19.)

[130] Robert Tarjan. Depth first search and linear graph algorithms. *SIAM Journal of Computing*, 1(2):146–160, June 1972. (Ref: p. 78.)

[131] Robert Tarjan. Enumeration of the elementary circuits of a directed graph. *SIAM Journal of Computing*, 2(3):211–216, September 1973. (Ref: p. 78.)

[132] Kardi Teknomo. *Microscopic Pedestrian Flow Characteristics: Development of an Image Processing Data Collection and Simulation Model*. PhD thesis, Tohoku University,

2002. http://people.revoledu.com/kardi/publication/Dissertation.pdf. (Ref: p. 110.)

[133] David Tennenhouse. Proactive computing. *Communications of the ACM*, 43(5):43–50, 2000. http://doi.acm.org/10.1145/332833.332837. (Ref: p. 14.)

[134] G. A. Thomas, J. Jin, T. Niblett, and C. Urquhart. A versatile camera position measurement system for virtual reality tv production. In *Proceedings of the International Broadcasting Convention (IBC)*, pages 284–289, September 1997. (Ref: p. 24.)

[135] European Union. Data protection directive (95/46/ec). *Official Journal of the European Communities*, L. 281:31, November 1995. http://www.cdt.org/privacy/eudirective/EU_Directive_.html. (Ref: p. 35.)

[136] R. Want, B.N. Schilit, N.I. Adams, R. Gold, K. Petersen, D. Goldberg, J.R. Ellis, and M. Weiser. An overview of the PARCTAB ubiquitous computing experiment. *IEEE Personal Communications*, 2(6):28–43, December 1995. http://ieeexplore.ieee.org/iel4/98/10192/00475986.pdf. (Ref: p. 23.)

[137] Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992. http://doi.acm.org/10.1145/128756.128759. (Ref: p. 23.)

[138] Roy Want, Trevor Pering, and David Tennenhouse. Comparing autonomic and proactive computing. *IBM Systems Journal*, 42(1):129–135, 2003. http://www.research.ibm.com/journal/sj/421/want.pdf. (Ref: p. 14.)

[139] A. Ward. *Sensor-Driven Computing*. PhD thesis, Cambridge University, August 1998. (Ref: p. 20.)

[140] Andy Ward, Alan Jones, and Andy Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, October 1997. http://ieeexplore.ieee.org/iel4/98/13641/00626982.pdf. (Ref: p. 22.)

[141] Mark Ward, Ronald Azuma, Robert Bennett, Stefan Gottschalk, and Henry Fuchs. A demonstrated optical tracker with scalable work area for head-mounted display systems. In *Proceedings of the Symposium on Interactive 3D Graphics*, pages 43–52, March 1992. http://doi.acm.org/10.1145/147156.147162. (Ref: p. 23.)

[142] Samuel Warren and Louis Brandeis. The right to privacy. *Harvard Law Review 4*, pages 193–200, 1890. (Ref: p. 31.)

[143] M. Weiser. The computer for the 21st century. *Scientific American*, 365(3):94–104, September 1991. (Ref: p. 13, 14.)

[144] Greg Welch and Eric Foxlin. Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Computer Graphics and Applications*, 22(6), November 2002. http://ieeexplore.ieee.org/iel5/38/22427/01046626.pdf. (Ref: p. 16, 17, 18, 19.)

[145] Jay Werb and Colin Lanzi. Designing a positioning system for finding things and people indoors. *IEEE Spectrum*, pages 71–78, September 1998. http://ieeexplore.ieee.org/iel4/6/15512/00715187.pdf. (Ref: p. 20.)

[146] Alan F. Westin. *Privacy and Freedom*. Bodley Head, London, 1970. (Ref: p. 35.)

[147] Leon Willenborg and Ton de Waal. *Statistical Disclosure Control*. Springer, 2001. (Ref: p. 49, 50.)

[148] Mason Woo, Jackie Neider, and Tom Davis. *OpenGL Programming Guide*. Addison Wesley Developers Press, 1997. (Ref: p. 27.)

[149] Christopher Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997. http://dx.doi.org/10.1109/34.598236. (Ref: p. 25.)

[150] Moustafa A. Youssef, Ashok Agrawala, and Udaya Shankar. WLAN location determination via clustering and probability distributions. In *Proceedings of the IEEE International Conference Pervasive Computing and Communications (PerCom)*, 2003. http://csdl.computer.org/dl/proceedings/percom/2003/1893/00/18930143.pdf. (Ref: p. 20.)