

# Chapter 1

## Introduction

Ubiquitous computing is the vision of a world in which computing power and digital communications are extremely inexpensive commodities, so cheap that they are embedded in all the everyday objects that surround us. This book examines the security issues of such a scenario.

In this chapter we briefly introduce ubiquitous computing (more on this in the next chapter), we define some basic terminology and we point out the principal security concerns that we shall be facing.

### 1.1 Scenario

The established trend in consumer electronics is to embed a microprocessor in everything—cellphones, car stereos, televisions, VCRs, watches, GPS (Global Positioning System) receivers, digital cameras. In some specific environments such as avionics, electronic devices are already becoming networked; in others, work is underway. Medical device manufacturers want instruments such as thermometers, heart monitors and blood oxygen meters to report to a nursing station; consumer electronics makers are promoting the Firewire standard for PCs, stereos, TVs and DVD players to talk to each other; and kitchen appliance vendors envisage a future in which the oven will talk to the fridge, which will reorder food over the net.

It is to be expected that, in the near future, this networking will become much more general. The next step is to embed a short range wireless transceiver into everything; then many gadgets can become more useful and effective by communicating and cooperating with each other. A camera, for example, might obtain the geographical position and exact time from a nearby GPS unit every time a picture is taken, and record that information with the image. At present, if the photographer wants to record a voice note with the picture, the camera must incorporate digital audio hardware; in the future, the camera might instead let the photographer

speak into her digital audio recorder or cellphone. Even better, the audio data might optionally take a detour through the user’s powerful laptop, where a speech recognition engine could transcribe the utterance, so as to annotate the photograph with searchable text rather than just with audio samples—and of course this could be done at any time that the camera detects the proximity and availability of the laptop with the speech recognition service. In this scenario each device, by becoming a network node, may take advantage of the services offered by other nearby devices instead of having to duplicate their functionality.

This vision, as we shall see in chapter 2, was first put forward by Mark Weiser of Xerox PARC [259], who coined the locution “ubiquitous computing” in 1988. Between then and now, many research organizations have started projects to explore various facets of this vision, and some of this research is now materializing into consumer products. In 2001, the most visible commercial incarnations of this idea were two open standards for wireless radio networking: Bluetooth [40, 126], originally thought of as a “serial cable replacement” for small computer peripherals, and 802.11, originally developed as a wireless LAN system for laptops. Estrin, Govindan and Heidemann [102] present a future scenario of ubiquitous embedded networking that encompasses this and much more.

## 1.2 Essential terminology

Computer people generate neologisms at an alarming rate. The inflation of trendy buzzwords and acronyms is all too often a dubious marketing gimmick to cover the lack of contents, but there are cases in which a new term genuinely is the best way to describe a new technology or a new way of doing things. I leave it to the reader to decide whether my use of new terms in this book falls in the first or the second category, but it seems in any case a good idea to define the most relevant ones in advance.

The focus of this work shall be the examination of **security issues for ubiquitous computing and ad hoc networking**. The *Oxford English Dictionary* [203] (henceforth “the *OED*”) defines “ubiquitous” as

Present or appearing everywhere; omnipresent.

With **ubiquitous computing** we refer to a scenario in which computing is omnipresent, and particularly in which devices that do not look like computers are endowed with computing capabilities. “A computer on every desk” does not qualify as ubiquitous computing; having data processing power inside light switches, door locks, fridges and shoes, instead, does.

As we saw in section 1.1, we envisage a situation in which all those devices are not only capable of computing but also of communicating, because their synergy

then makes the whole worth more than the sum of the parts. We do not however expect a fixed networking infrastructure to be in place—certainly not one based on cables. It would be less than practical to run data cables between switches, locks and fridges—not to mention shoes. A wireless network infrastructure looks more plausible: as happens with mobile telephones, a base station could cover a cell, and a network of suitably positioned base stations could cover a larger area. But we are interested in a broader picture, in which even this arrangement may not always be possible or practical: think of a photographer taking pictures in the desert and whose camera wants to ask the GPS unit what coordinates and timestamp to associate with the picture. The computing and the communications may be ubiquitous, but the network infrastructure might not be. In such cases the devices will have to communicate as peers and form a local network as needed when they recognize each other’s presence. This is what we mean by **ad hoc networking**. The *OED* defines “ad hoc” as

Devoted, appointed, etc., to or for some particular purpose.

The wireless network formed by the camera and the GPS receiver is ad hoc in the sense that it was established just for that specific situation instead of being a permanent infrastructural fixture.

Finally, it would perhaps be desirable to define **security**, not because the term is new or unfamiliar, but because it is overloaded, and may be interpreted differently by different readers.

A common mistake is to identify security with cryptology, the art of building and breaking ciphers (*cryptography* and *cryptanalysis* respectively). While it’s true that cryptology gives computer security many of its technical weapons, to identify the two is to miss the big picture and to expose oneself to less glamorous but probably more effective attacks. As demonstrated by Anderson [11, 8] with a wealth of case studies, what fails in real life is rarely the crypto.

In a nutshell, security is really risk management. Security is assessing **threats** (bad things that may happen, e.g. your money getting stolen), **vulnerabilities** (weaknesses in your defences, e.g. your front door being made of thin wood and glass) and **attacks** (ways in which the threats may be actualized, e.g. a thief breaking through your weak front door while you and the neighbours are on holiday), estimating costs for the threats, estimating probabilities for the attacks given the vulnerabilities, developing appropriate **safeguards** (*a priori* vaccines) and **countermeasures** (*a posteriori* remedies), and implementing the ones for which the certain price of the defence is worth spending compared to the uncertain loss that a potential threat implies.

In this context it is apparent that cryptology is only one of many tools, not the discipline itself. Amoroso [7], whose clear terminology we adopted in the

previous paragraph, offers a rigorous overview of this process. Schneier, author of an extremely popular cryptography textbook [227], candidly admits in a later book [228] to having previously missed the forest for the trees.

Having clarified this, I shall give an overview of computer security mechanisms for the uninitiated reader in chapter 3.

## 1.3 Problems

Ubiquitous computing imposes peculiar constraints, for example in terms of connectivity, computational power and energy budget, which make this case significantly different from those contemplated by the canonical doctrine of security in distributed systems.

A well-established taxonomy subdivides computer security threats into three categories, according to whether they threaten confidentiality, integrity or availability. Let us review these three fundamental security properties given the preconditions of ubiquitous computing.

**Confidentiality** is the property that is violated whenever information is disclosed to unauthorized principals<sup>1</sup>. Everyone realizes that wireless networking is more vulnerable to passive eavesdropping attacks than a solution based on cables: by construction, information is radiated to anyone within range. It is natural to expect that the security requirements of a wireless system will include addressing this concern.

**Integrity** is violated whenever information is altered in an unauthorized way. This applies both to information within a host and to information in transit between hosts. Imagine a wireless temperature sensor on your roof that relays its measurements to a display inside your house (at ORL we built a prototype of such a device for Piconet in 1998, as part of a playground of simple communicating devices which also included fans, displays, logging nodes and so on (see section 2.5.5); but a much nicer, if less versatile, commercial version could probably be bought at Radio Shack even then). If an attacker modifies either the sensor's firmware or the transmitted messages so that the displayed temperature is off by 10 degrees then, if you are sufficiently gullible, you may be cheated into wearing the wrong type of clothes for that day's weather. If this does not look like a terribly dramatic security violation, imagine instead that the sensor is monitoring a patient's temperature in a clinic or, even better, that it is part of an alarm system for a nuclear power plant. As happens with confidentiality, the wireless nature of communications increases the vulnerability of the system to integrity violations: if the receiver listens to the

---

<sup>1</sup>We call *principals* the entities that can perform actions; this general and somewhat ambiguous term encompasses without distinction humans, machines that act as representatives for humans, and machines that don't.

strongest signal that “looks right”, an attacker wishing to substitute forged messages for the original ones only needs to shout loudly enough, without having to splice any cables. As for the integrity of hosts, as opposed to that of messages in transit, the ubiquitous computing vision of *unattended* devices ready to communicate with whoever comes in range clearly makes it likely that an attacker will sooner or later tamper with such unattended devices if this can bring her any benefits.

**Availability** is the property of a system which always honours any legitimate requests by authorized principals. It is violated when an attacker succeeds in *denying service* to legitimate users, typically by using up all the available resources. As we remarked about integrity, the fact that ubiquitous computing implies unattended devices opens the door to many abuses. If we envisage that these ubiquitous hosts might accept mobile code that roams from one of them to another, then denial of service might also be caused by malicious programs that lock up the host device.

While illustrating the three fundamental security properties of confidentiality, integrity and availability we have repeatedly referred to “authorized principals”. It follows that a fundamental prerequisite of a secure system is the ability to establish whether any given principal is or is not authorized to perform the action it is requesting. To define “who is authorized to do what” is the duty of the *security policy*, a concise specification of the security goals of the system. In order to ascertain whether the policy authorizes a principal to perform an action, there is also a need for *identification* (finding out who the principal claims to be) and particularly *authentication*<sup>2</sup> (establishing the validity of this claim). **Authentication** is one of the foundations of security: it is easy to come up with examples that demonstrate that, in its absence, the three fundamental properties can be trivially violated. (Looking for example at confidentiality, even if your communications are protected with military-grade encryption, you are still liable to suffer from a disclosure threat if you have unknowingly established your encrypted channel with a recipient other than the one you intended.) Since authentication is such a central issue, we shall examine how various existing systems deal with it and then turn to the peculiar problems encountered in performing authentication in ad hoc networking, where the absence of infrastructure makes the traditional approaches impracticable.

We shall also look more closely at a peculiar aspect of confidentiality that is not quite mainstream: **anonymity**. Most of the attention devoted to confidentiality concentrates on how to prevent disclosure of the *contents* of messages, which leads naturally to cryptology. Sometimes, however, the really sensitive information is not in the body but in the header. Given the same number of pages, a detective or a spy will generally find an itemized phone bill for his target much more revealing than the transcript of any individual phone call. This sort of attack is called *traffic*

---

<sup>2</sup>Here we refer in particular to *identity* authentication, but it is reasonable to discuss the authenticity of other kinds of information. The term has therefore wider applicability in the general case.

*analysis*. The danger is not limited to the world of secret agents: credit cards and loyalty cards record your spending patterns, cash machine transactions and cellular telephone calls timestamp your whereabouts, and the fusion of all these logs can be used to build disturbingly detailed and intrusive dossiers on private individuals. As we design the technology that will enable ubiquitous computing, we have a duty to protect future users (ourselves included) from what could otherwise turn by default into an Orwellian ubiquitous surveillance.

We shall examine each of these problems in turn: I have dedicated one chapter to each of the boldface terms in this section. Finally, an appendix offers a brief survey of deployed network security solutions.

## 1.4 Notation

Existing notations for encryption are many and varied. To some extent, each author seems to come up with his or her own preferred flavour. I shall not break with this tradition: in the interest of explicitness, I shall adopt my own personal variation that will allow us to mention the cipher explicitly where this is useful, and to identify the function being performed without relying on implicit inferences from the key in use. We shall use the function names  $E$ ,  $D$ ,  $S$ ,  $V$ ,  $h$  and  $MAC$  respectively for encryption, decryption, signature, verification, hash and message authentication code (see chapter 3 for definitions of these terms), with optional subscript and superscript to indicate key and algorithm. So

$$E(m), E_K(m), E_K^{\text{AES}}(m)$$

respectively indicate the encryption of message  $m$ , the encryption of message  $m$  under key  $K$  and the encryption of message  $m$  with AES under key  $K$ .

There is much less disagreement over the rest of the notation. I shall, like most authors, indicate a symmetric key shared between  $A$  and  $B$  as  $K_{AB}$ , whereas for public key cryptography I shall use keys that mention only one principal in the subscript:  $K_A$  will be  $A$ 's public key and its inverse  $K_A^{-1}$  will be  $A$ 's private key.

I shall place a delta (for “definition”) over the equal sign (like this:  $\triangleq$ ) when I mean “is hereby defined to be equal to” as opposed to simply “is equal to”.

As you probably guessed, numbers enclosed in square brackets (like this: [92]) are references to the annotated bibliography at the end of the book.

For protocols, I shall adopt the classical notation whereby

$$A \rightarrow B : m$$

indicates that principal  $A$  sends message  $m$  to principal  $B$ —with a notable exception in section 8.2. This should not be confused with the superficially similar notation

used in appendix A to define the domain  $A$  and range  $B$  of a relation  $R$ , as in

$$R : A \rightarrow B.$$

For dates, I shall use the ISO standard notation of year-month-day; among its many advantages, its monotonic big-endianness<sup>3</sup> means that a simple string sort also works as a chronological sort. (This rule does not apply to the publication dates in the bibliography, though: there, the inscrutable  $\text{BIB}\text{T}_\text{E}\text{X}$  does whatever it likes.)

For money, as an engineer I refuse to write “ten thousand dollars” as “\$10k”, which is just as nonsensical as writing “s10 $\mu$ ” for “ten microseconds”. The scaling prefix “k”, being indeed a scaling *prefix* and not a cute abbreviation for the string “,000”, is placed *before* its fundamental unit to form a new derived unit; and the correct SI order is “value, space, unit”, not “unit, value”. I shall therefore write “10 k\$”.

Some authors draw fine distinctions between the related terms of “confidentiality”, “secrecy” and “privacy”. The definitions vary subtly from author to author, often in contradictory ways; the only common ground seems to be that “privacy” describes personal secrets as opposed to organizational ones (for which some authors use “secrecy” and others “confidentiality”). I feel that there is little clarity to gain in officially assigning arbitrary nuances to these almost synonymical terms. In this book I shall normally use “confidentiality” (whose meaning I define in section 3.1), and my occasional use of “secrecy” and “privacy” shall not imply a purposeful technical distinction.

---

<sup>3</sup>Any given digit weighs more than any digit on its right.