Security Issues in Ubiquitous Computing*

Frank Stajano

1 Fundamental Concepts

The manifesto of ubiquitous computing is traditionally considered to be the justly famous 1991 visionary article written for *Scientific American* by the late Mark Weiser of Xerox PARC (Weiser, 1991). But the true birth date of this revolution, perhaps hard to pinpoint precisely, precedes that publication by at least a few years: Weiser himself first spoke of "ubiquitous computing" around 1988 and other researchers around the world had also been focusing their efforts in that direction during the late Eighties. Indeed, one of the images in Weiser's article depicts an Active Badge, an infrared-emitting tag worn by research scientists to locate their colleagues when they were not in their office (in the days before mobile phones were commonplace) and to enable audio and video phone call rerouting and other follow-me applications. That device was initially developed by Roy Want and Andy Hopper (Want, Hopper, Falcao, and Gibbons, 1992) at Olivetti Research Limited (ORL) in 1989. Soon afterwards other research institutions, including Xerox PARC themselves who later hired Want, acquired Active Badge installations and started to explore the wide horizons opened by the novel idea of letting your spatial position be an input that implicitly told the computer system what to do. I was fortunate enough to work at ORL as a research scientist throughout the Nineties, thereby acquiring first hand experience of these and other pioneering developments in ubiquitous computing[2]. Looking back and putting things into historical perspective, ubiquitous computing really was a revolution, one of these major events in computing that take place once

© Frank Stajano
University of Cambridge Computer Laboratory, Cambridge, United Kingdom.
e-mail: frank.stajano@cl.cam.ac.uk

* Revision 86 of 2009-01-29 18:00:05 +0000 (Thu, 29 Jan 2009).

[2] I later told that story in my book (Stajano, 2002) and the relevant 20+ page extract can be downloaded for free from my web site at http://www.cl.cam.ac.uk/~fms27/secubicomp/secubicomp-section-2-5.pdf.

every decade or so and transform society even for ordinary non-techie people, as happened with the personal computer and the World Wide Web.

Weiser opened his landmark paper with a powerful metaphor: writing is everywhere, but it is so deeply embedded in the fabric of society that it has become invisible, in the sense that we no longer notice it. His vision, that something similar would one day happen to computers, has undoubtedly come true: most people have no idea of the number of microprocessors they own and interact with on a daily basis; if asked, they may easily forget the ones hidden in their car, in their microwave oven and in their mobile phone—even though the latter probably offers more computing power than the 486 desktop computers that were top of the range when Weiser's article was published.

This revolution has transformed society in many ways. The one we shall explore together in this chapter is security. A modern computer is a sufficiently complex system that even an expert is incapable of entirely understanding it at all levels—even more so when we consider a system of many networked computers instead of a standalone machine, and even more so when such networking is invisible and spontaneous, as wireless technology allows it to be. Amongst all this complexity lurk bugs and unexpected interactions that cause unpredictable and probably undesirable behaviour. Worse, such bugs and interactions can be exploited maliciously by an adversary in order to cause intentional damage. It is easy to imagine that an adversary might render the system unusable ("A virus formatted my hard disk!") or at least cause data loss ("A virus deleted all my JPGs!"). However, as networked computer systems become more and more deeply embedded in the fabric of society, the type of damage that can be inflicted by malicious attacks on them become more varied and more pervasive. Nowadays, computer security can seriously affect even people who think they don't use computers—because they probably use them and depend on them without even realizing that they do.

This first section of the chapter will explore the scope of "ubiquitous computing", noting how much ground is covered, implicitly or explicitly, by this phrase and its many alternatives; it will give a brief overview of the core concepts of security; and it will then look at ubiquitous computing from the viewpoint of security, pointing out what could go wrong, what is at stake and what might require protection.

1.1 Ubiquitous (Pervasive, Sentient, Ambient. . .) Computing

From the late Eighties, when ubiquitous computing (or "ubicomp" for short) was the blue-sky research dream of a few visionaries, throughout the Nineties and the first decade of the new millennium, more and more research institutions have joined the exploration. Thanks in part to generous sponsorship from major long term government initiatives (especially in Korea and Japan), the ubicomp research field has become mainstream, and then quite crowded. Whether in academia or in industry, many groups have invented their own new name for the topics they explored—partly to highlight that their focus was just on a specific aspect of the phenomenon; but all too often, a cynic might say, simply for the same reason that cats leave scent marks. Be that as it may, we now have a variety of names for ubiquitous computing and each tells its own instructive little story.

"Ubiquitous computing" (Weiser, 1991), a locution starting with a sophisticated and at the time uncommon word that literally means "present everywhere"[3], places the emphasis on computers that, since they are everywhere, must be (by implication) cheap commodities that are plentiful and unobtrusive. In the Eighties the Microsoft slogan of "a computer on every desk" was still to some extent a dream, though one within reach; but "ubiquitous" pointed further down the line, to computers hidden inside everyday objects, in the same way as you have electric motors inside everyday objects, from vacuum cleaners to lawnmowers and hi-fi systems, and yet you don't say "oh, look, we have half a dozen electric motors in the living room". **"Pervasive computing"** (Ark and Selker, 1999) has similar connotations, particularly with respect to emphasizing embedded computers as opposed to standalone ones: computing then "pervades" the environment. Names such as **"Invisible computing"** (Norman, 1998) or **"Disappearing computing"** (Streitz, Kameas, and Mavrommati, 2007) also underline the distinction between the old computer as a keyboard-and-monitor machine and the new ones that are just embedded, each dedicated to its own special purpose within the object that embeds it, and not requiring one to sit in front of them or think about their existence. **"Sentient computing"** (Hopper, 2000) refers to computer systems that sense some aspects of their environment in order better to serve their users—for example by taking note of who is in the room and switching to their preferences in terms of lighting, temperature and user interface. **"Ambient intelligence"** (Ducatel, Bogdanowicz, Scapolo, Leijten, and Burgelman, 2001) is again similar, conjuring images of a world that automatically adapts to the users and their preferences; but this locution puts the accent, rather than on the sensing, on the deductive processing that takes place in the backend. **"Calm computing"** (Weiser and Seely Brown, 1997), which predates most of the others, describes essentially the same scenario, this time placing the emphasis on the fact that this new generation of computers is intended to work in the background, silently taking care of issues on our behalf as opposed to requiring constant attention and intervention from the user.

Perhaps the most significant thing to learn from this multitude of names, other than laughing at these manifestations of the "Not Invented Here" syndrome, is that ubiquitous computing means different things to different people, even within the research community, and that the global revolution that has changed the world even for non-techies, putting a powerful computer in the ordinary person's pocket in the form of a mobile phone that is also a camera, a music player and a web browser, is to some extent only fully described by the union of all of these disparate viewpoints.

[3] Amusingly, some authors have recently started to use the difficult word "ubiquitous" with no regard for its original meaning, using it instead as a synechdoche for "ubiquitous computing". We therefore hear of etymologically meaningless concepts such as "the ubiquitous society"...

1.2 Security

Security is another word that is frequently abused and overloaded to mean different things depending on the speaker. Since this is the only chapter of this volume specifically dedicated to security, I'll take you through a two-page crash course on what I consider to be the most fundamental security ideas. If you wish to read more on system security, my favourite books on the subject are *Beyond Fear* (Schneier, 2003), for an easy to read yet insightful and valuable introduction; *Security Engineering* (Anderson, 2008), for an authoritative and comprehensive reference suitable for both the researcher and the practicing engineer; and the older but still valuable *Fundamentals of Computer Security Technology* (Amoroso, 1994), for a rigorous formal description of the principles; not to mention of course my own *Security for Ubiquitous Computing* (Stajano, 2002) for a coherent and self-contained look at how all this applies specifically to ubicomp.

The viewpoint we shall adopt here, which I believe is the only one leading to robust system security engineering, is that *security is essentially risk management*. In the context of an adversarial situation, and from the viewpoint of the defender, we identify **assets** (things you want to protect, e.g. the collection of magazines under your bed), **threats** (bad things that might happen, e.g. someone stealing those magazines), **vulnerabilities** (weaknesses that might facilitate the occurrence of a threat, e.g. the fact that you rarely close the bedroom window when you go out), **attacks** (ways a threat can be made to happen, e.g. coming in through the open window and stealing the magazines—as well as, for good measure, that nice new four-wheel suitcase of yours to carry them away with) and **risks** (the expected loss caused by each attack, corresponding to the value of the asset involved times the probability that the attack will occur). Then we identify suitable **safeguards** (a priori defences, e.g. welding steel bars across the window to prevent break-ins) and **countermeasures** (a posteriori defences, e.g. welding steel bars to the window after a break-in has actually occurred[4], or calling the police). Finally, we implement the defences that are still worth implementing after evaluating their effectiveness and comparing their (certain) cost with the (uncertain) risk they mitigate[5].

[4] One justification for the a-posteriori approach, as clearly explained by Amoroso, is that it allows you to spend protection money only after you see it was actually necessary to do so; that way, you don't pay to protect against threats you only imagined. On the other hand, you can only hope to prevent reoccurrences, not the original threat, because by then you have already lost the asset; and in some cases (e.g. loss of life) this may be unacceptable. Another justification, emphasized by Schneier, is that you might not be able to safeguard against every possible attacker even if you wanted to; therefore, relying only on prevention will leave you exposed. Note that some authors, including Schneier, use the term "countermeasures" for both a-priori and a-posteriori defences.

[5] It is also worth noting in passing that, when managing risk, simply balancing the expected loss against the cost of safeguards and countermeasures is not the whole story: in many situations a potential victim prefers to pay out a known amount in advance, perhaps even every month but with no surprises, rather than not paying anything upfront but occasionally having to face a large unexpected loss—even if the second strategy means lower disbursements in the long run. This is what keeps insurers in business. Besides, insurers can spread a very costly but very unlikely-to-occur loss across their many premium-paying customers, whereas that kind of loss would probably

This choice is always a trade-off, and it is important to note that the amount of protection that it is reasonable to deploy is a function not simply of the possible *threats* (what bad things could happen) but crucially of the corresponding *risks* (how likely it is that those bad things would happen, and how much we'd lose if they did) which in turn depend on the value of the assets[6]. In the example above, if you were storing gold ingots under your bed instead of magazines, the threats and vulnerabilities and attacks would be the same but the risks would change dramatically. Steel bars at the window might be a proportionate response to protect gold ingots but they are probably a poor security choice to protect magazines, because their cost (sourcing and installation cost, but also inconvenience cost for the user of the bedroom) is disproportionate to the value of the assets they protect *and*, equally importantly, to the likelihood that such assets will ever be stolen. You'd be better off just by getting into the habit of closing the window (a much cheaper and weaker safeguard, but still an adequate one for that risk). Moreover, considering the relative likelihoods of burglaries and house fires, and comparing the loss of life by fire (increased by the impossibility of escaping through a window now blocked by steel bars) to that of material assets through burglary, for a residential installation the steel bars might turn out to be a poor security trade-off even for protecting gold ingots.

This risk management mindset is necessary if you wish to provide *effective* security rather than, as all too often happens, just an *appearance* of security in order to appease the customer or the regulator.

Traditionally, the threats to information systems are classified with reference to three fundamental security properties that they might violate: **confidentiality** (ensuring that only authorized principals can read the information), **integrity** (ensuring that only authorized principals can modify the information) and **availability** (ensuring that authorized principals can access and use the system as intended, without undue delays). These three categories do not necessarily exhaust all the desirable security properties of an information system but they are a good starting point to understand what's most important, at least in the context in which the system is used. You'll find that in many cases people automatically assume that the primary security concern is the protection of confidentiality ("unauthorized principals shouldn't be able to see the students' exam grades") but that, on second look, integrity is instead more significant ("unauthorized principals must not be able to edit the student's grades") and the primary concern by far, especially when the system integrates several different functions, is in fact availability ("the school's computer system cannot stop working for more than a couple of days", as it handles not only the students' grades but also the course timetable, the library catalogue and the staff payroll).

bankrupt individual victims if they had to bear it by themselves. Then, bearing the cost of the attack would be impossible, whereas paying the insurance premium might turn out to be cheaper and more convenient than implementing a safeguard.

[6] Note how attacker and defender may place very different values on the same assets. If a small-time crook enters the bank vault and steals a hundred dollars, to him the value of what he gains is a hundred dollars; but, to the bank, the value of what they lose is much greater—especially if it becomes generally known that a burglary in their vault occurred. Similarly, attacker and defender may have very different ideas on the likelihood of certain events happening. It is important to take the viewpoint of the correct actor into account when performing quantitative risk evaluations.

You may have noticed the recurrence of the phrase “authorized principals” in the previous paragraph. This appropriately suggests that the foundation upon which these three security properties rest is **authentication**, that is to say a means of distinguishing between authorized and unauthorized users of the system. To be more precise, this foundation usually consists of a sequence of three sub-actions: **identification** (understanding who the user claims to be), **verification** (checking the validity of this claim), **authorization** (granting that particular user the permission to perform certain specific actions within the system). Interestingly, in some cases there is a level of indirection such that you don’t identify a specific individual but just a role with associated capabilities—e.g. the librarian checks an authorization token to ensure that this customer is entitled to borrow up to four books, but doesn’t need to know her name and surname in order to lend her the volumes.

1.3 Security Issues in Ubiquitous Computing

The information security concepts presented in the previous section were developed for networked computer systems well before the advent of ubiquitous computing. They still represent a useful framework to reason about security in ubiquitous computing too. You should strive to recognize where the above-mentioned “boldface terms” are in any specific ubicomp scenario you may come to face, and then evaluate the corresponding security trade-offs. What are the assets? What are the risks? What safeguards and countermeasures are worth implementing to mitigate the risks?

A good starting point, so long as it is not mistaken for the whole security analysis, is to think about possible threats in the various scenarios of interest. As you imagine various specific ubicomp scenarios, try to understand what the assets are in each setting and what kind of damage they might incur.

Go beyond the obvious: if you are thinking about a mobile phone, it is probably clear that the handset itself is an asset (particularly if you paid for a sophisticated model rather than accepting the cheap default one offered with your subscription plan) and that so is the ability to make phone calls that will be paid for by the person *owning* the phone (you) rather than by the person *holding* it right now (the thief). But isn’t your phone directory also an asset? Your photos? Your short messages and emails? And is the damage primarily the fact that you no longer have them (availability), or the fact that someone else can now view them (confidentiality)? If you have a phone that lets you connect to the web, as will become more and more prevalent in the future, how about the credentials that your phone uses to connect to web sites that run financial transactions, such as online stores or home banking? You go from theft of the device to theft or fraud *enabled* by possession of the device. What about malicious software planted into your phone that silently accepts a call from a “trigger” number to turn on the microphone, acting as a stealth bug that you unknowingly but voluntarily carry into that confidential meeting? (Why are you so sure that such software isn’t already in your phone right now, by the way? How could you tell? A web search yields several companies ready to sell your enemies a copy[7].) Is your privacy also an asset, then? If you think so, in what other ways could it be violated? Would you also consider your current location, or the timestamped history of your past locations, as a privacy-sensitive piece of information? If so, surely you also see how it could be threatened simply by the fact that you carry a mobile phone.

Privacy is a complex issue and a very relevant one for ubiquitous computing. But, even for the public-spirited technologist whose aim is to design privacy-protecting ubicomp systems, the job of protecting privacy is made harder by two factors: one, that the potential victims do not appear to care much about it; two, that there are powerful economic interests pushing in the direction of privacy violations. These factors must be addressed, as they can be stronger than any simple-minded technical protection you might put in. We'll get back to this at the end of the chapter.

When looking for security issues in a given ubicomp setting, don't just think about assets and threats from the viewpoint of the potential victim. Often, some of the most interesting insights are obtained by thinking like the attacker. Which of the victim's assets would be most valuable to you *as an attacker*? Maybe these are things that the victim doesn't even *consider* as assets—at least until the attacker has had a go at them. And how would you go about attacking them? Adopting the mindset of the adversary is not easy and requires a great deal of lateral thinking; however, in most adversarial endeavours, from lockpicking to cryptography to tank armour development, the wisdom of experience says that the only people capable of building solid defences are the ones who are skilled at breaking them. You need the mindset of playing outside the rules, of violating conventions, of breaking out of neatly formalized abstraction levels: you will then, for example, defeat that unbreakable cipher by flying below the mathematics and attacking at the level of physics, electric currents, noisy signals and non-ideal power supplies.

2 Application Scenarios and Technical Security Contributions

The best way to get a feeling for the practical security issues relating to ubiquitous computing is to discuss a few significant scenarios. The most interesting question to ask is: "in what way does ubicomp make a difference to security in this setting"? Sometimes there is an obvious qualitative difference; other times there is "only" a quantitative difference and yet it becomes qualitative because of the scale of the change: when things become a thousand times smaller, or a thousand times faster, or a thousand times more numerous, the way we use them changes entirely; and the new usage patterns open up new vulnerabilities as well as new opportunities.

In this short book chapter it will certainly not be possible to examine in detail all the scenarios of interest, which would have to include at least smart homes, traffic management systems, smart cars, wireless sensor networks, ubicomp in sports and healthcare and so on, as well as specific wireless technologies such as Bluetooth,

[7] If you were paranoid you would already "know" that the spooks arm-twisted the manufacturers to add such functionality in the firmware anyway.

NFC, Wi-Fi, Zigbee and others. We shall arbitrarily pick a few significant examples and a few general security contributions that apply to several ubicomp settings.

2.1 Wearable Computing

As computers become smaller and cheaper and more powerful, they become devices we can easily carry around wherever we go. From the 10 kg "portables" of the Eighties to modern sub-kilogram netbooks and palm-sized PDAs (personal digital assistants), the truly ubiquitous instantiation of a computing device that even ordinary people carry around all the time is nowadays, as we have already remarked, the cellular phone. As more and more of these phones acquire the ability to connect to the Internet, to run external applications and to act as email clients, their architectural similarities with regular computers bring back all the traditional threats, vulnerabilities and attack vectors for which personal computers are notorious, starting with viruses and worms. It is therefore important for the ubicomp security practitioner to keep up to date about ordinary PC security. The fact that the computer changes shape to a telephone—or, perhaps tomorrow, to augmented-reality glasses that also accept voice commands—is irrelevant. Viruses, worms, spam, "Nigerian" scams and phishing will all migrate from desktop computers to mobile phones as soon as enough users starts making use of phone-based applications and web services. The transition from model-specific phone firmware to generic software platforms based on just two or three major phone OSes will help reach the critical mass level at which developing malware for phones becomes seriously lucrative.

"Augmented reality", by the way, refers to a type of display (usually implemented with special glasses) in which a computer generated image is superimposed onto the normal scene that is actually in front of the viewer—as opposed to "virtual reality" in which everything the viewer sees is computer-generated. One of the best known early adopters of augmented reality and wearable computing was Steve Mann, who for years went around with a wearable camera over his head and studied the technical and social consequences of doing so (Mann, 1997), from ways of stitching together composite images as the viewer's head moves (Mann and Picard, 1997) to issues of ownership of video footage containing images of other people (Mann, 1996), and to the relationship between the personal wearcam and the now-ubiquitous CCTV surveillance cameras (Mann, 1995).

Although wearable cameras are far from being a mainstream or ubiquitous item at the time of writing, they are a good catalyst for a discussion about the security and privacy implications of wearable computing in general. Mann's WearCam was at least in part meant as a provocation and therefore serves its purpose even if it does nothing more than making you think about such issues. A later project in a similar direction, the SenseCam[8] built by Lindsay Williams at Microsoft Research (Gemmell, Williams, Wood, Lueder, and Bell, 2004), integrates with the MyLifeBits

[8] The SenseCam is a low-resolution wide-angle still camera, worn on the chest with a necklace chain, that takes a photo every minute or so. The user never presses the shutter button or aims the

project (Gemmell, Bell, Lueder, Drucker, and Wong, 2002) to provide a visual diary of one's life and has been used as a memory prosthesis for Alzheimer patients. What are the consequences of transferring your memories from the wetware of your brain to some external hardware? For privacy-sensitive material, the change makes a big difference, especially in the face of the threat of coercion: it is much easier to extract your secrets by brute force out of your wearable camera than out of you, and the former carries much less of a moral stigma than the latter (Stajano, 2004).

You might not consider the disclosure of your visual diary as a major risk if you only think of wearable cameras[9], since you probably don't wear one; but you might reconsider your stance in light of the fact that most of the wearable (or at least "pocketable") technological gadgets you carry around, such as music players, digital cameras, voice recorders, USB sticks and so on, not to mention once again mobile phones, can hold multiple gigabytes of information—overall, a self-writing life diary even if you don't mean it. Even ignoring malicious intent for the moment, it is now fairly easy to lose several gigabytes in one go when that USB stick falls out of your pocket (think public transport, conference, cinema, gym...). This affects availability, unless you have up to date backups (ha ha)[10], and confidentiality, unless all your mass storage is protected by strong encryption (ha ha again). A variety of embarrassing high profile incidents (Collins, 2007; BBC, 2007; Heffernan and Kennedy, 2008) have demonstrated how easy it is, even for professionals and government officers entrusted with sensitive personal information, to lose gigabytes of it, and how unlikely it is that technical countermeasures (such as mass storage encryption) be actually used in practice[11].

There are, however, more subtle privacy issues with wearable computing than just the relatively obvious one of losing media containing confidential information. We already hinted at the fact that the timestamped history of the locations you visited, which is more akin to metadata, can be equally sensitive. Let's open a separate subsection to examine that aspect in greater detail.

2.2 Location Privacy

One of the new buzzwords of ubicomp is "location-based services". At a bus stop, your location-aware mobile phone will not just bring up a timetable—it will also

camera, which works completely automatically. The user does not see what the camera records except at the end of the day when the camera is docked and its photos are downloaded.

[9] What if the camera also featured an always-on microphone?

[10] Some ubicomp devices, such as certain music players, PDAs and phones, automatically synchronize their contents with a desktop or laptop computer on connection, which provides a certain amount of redundancy and backup at little effort for the user. This is a good thing, but it is still far from a robust system solution: in the ubicomp world, the syncing desktop or laptop is not managed by a professional system administrator and may itself have its disk accidentally wiped at some point for a variety of reasons (malware, hardware failure, operator error etc).

[11] By the way: if and when encryption of secondary storage eventually becomes mainstream, expect many serious failures at the key management level.

open it at the page containing the schedule of the specific buses that actually stop there.

At the time of writing, location-based services are still mostly research visions rather than a commercial reality; but their chance will come once high resolution positioning is integrated into all mobile phones. This trend is already under way, led by the US requirement that phones placing emergency calls be locatable to 100 m or less. Researchers are busy imagining creative ways of putting this new capability to other uses.

An aspect of the question that has not yet received sufficient attention is security; in particular, the issue of protecting the location data of the individuals making use of such services. Without such provision, having one's movements tracked 24 hours a day will become an unacceptable invasion of one's privacy.

Think for example of the patient who visits a psychiatrist or a clinic specializing in sexually transmitted diseases: even if the test results and the discussions with the psychiatrist are never disclosed, the very fact that she attended might cause her great distress if others heard about it, and she might legitimately want to keep that to herself. Location privacy issues arise not just with mobile phones but to a certain extent with any mobile device that communicates with fixed infrastructure while on the move: the coffee shop, the library and the airport lounge will all have a record of the MAC address of your laptop in their logs if you use their wi-fi network. Crossing abstraction levels, your bank can track you around the country if you use your cash card in their ATM machines. So can the credit card company when you use their card in various shops around the world[12]. So can the subway or train company when you use their electronic season ticket to enter and exit each station.

Historically, the ORL Active Badges were among the first wearable computing devices and, because their explicit primary purpose was to help you find where your colleagues were, they were instrumental in early explorations of location privacy. In that context, Ian Jackson (Jackson, 1998) was one of the first to investigate location privacy in ubiquitous computing: he designed a system that, through mix-based anonymizers, would allow users to control the amount of information disclosed to applications and other users by the location tracking system. Alastair Beresford and I (Beresford and Stajano, 2003) explored fundamental location privacy principles within the experimental playground of the Active Badge and its successor the Active Bat; we introduced the *mix zone*, a spatial equivalent of David Chaum's mix (Chaum, 1981), and offered both a way to protect location privacy without entirely disallowing location-based applications, and a framework for measuring the amount of privacy (or, more precisely, anonymity) offered by a given system.

Jackson's primary way to protect location privacy is through access control, i.e. by giving you the power to grant or deny access to your location information based on the identity of the requester, as well as other features of the specific query being made. The strategy we adopt is instead to *anonymize* the information supplied to applications. So an application will know that *someone* is in that place, and will be able to provide service to that person, but won't know who it is.

[12] And it is in theory possible to distinguish from the logs whether the card was present during the transaction as opposed to having been read over the phone or internet.

To assess whether this protection is any good, we pretend we are a malicious application and attempt to find out the identity of the person whose location is being supplied anonymously. This is usually not hard: for each “home location” (i.e. each office, in our setting) we find out which pseudonym occupies that location more than any other; that’s usually the “owner” of that location—meaning we have already de-anonymized the pseudonym. Other simple heuristics can also be used if any ambiguity remains.

We then introduce a countermeasure, the “mix zone”, an unobservable spatial region in which we prevent the hostile application from tracking us. We hope that, on exiting the mix zone, the application will have mixed us up (i.e. confused us) with someone else. Once again we then take on the role of the hostile application and try to break the anonymity, attempting to correlate those who went into the mix zone with those who came out of it.

In the course of this process we develop quantitative measures of location privacy. These allow us to assess in an objective manner the effect of any privacy-protecting countermeasures we may adopt, such as degrading the spatial or temporal resolution of the data supplied to applications.

Marco Gruteser and Dirk Grunwald (Gruteser and Grunwald, 2003) propose an alternative safeguard, namely to degrade the spatial and temporal resolution of the answers from the location server to the applications in order to ensure a certain level of anonymity for the entities being located.

The most effective way of protecting location privacy, however, is to reverse the architecture and, instead of having mobile nodes (e.g. tags or cellphones) that transmit their location to the infrastructure, have the infrastructure tell the mobile nodes where they are (as happens in GPS) and, if appropriate, what location-based services are available nearby. For example, instead of the user asking the server which restaurants are nearby, the restaurants would broadcast messages with their location and the type of food they offer, which mobile nodes would pick up or discard based on a local filter set by the user’s query. Scalability can be achieved by adjusting the size of the broadcast cells (smaller, more local cells to accommodate more broadcasters and/or more frequent broadcasts). Note that in this reversed architecture the restaurants simply cannot know who receives their ads (hence absolute privacy protection), whereas in the former case the server knows and could tell them. This in itself may be an economic incentive *against* the deployment of such an architecture.

George Danezis and others (Danezis, Lewis, and Anderson, 2005; Cvrcek, Matyas, Kumpost, and Danezis, 2006) have studied the problem from an interestingly different perspective, with the aim of highlighting how much users actually value their privacy. Their studies use techniques from experimental psychology and economics in order to extract from each user a quantitative monetary value for one month’s worth of their location data (median answer: under 50 EUR), which they then compare across national groups, gender and technical awareness.

2.3 RFID

RFID (radio frequency identification) tags are a crucial development in ubicomp. For our purposes we consider them as low-cost, resource-constrained devices that communicate over a short-range radio interface and that are mainly used as machine-readable labels—to let a back-end system know that the real-world entity logically associated with the tag, be it a box of razor blades at a supermarket checkout or a passenger at a subway gate, is now in the vicinity of the reader. RFID chips can be made so small that they can be embedded in paper: possible applications include tickets, passports and even banknotes. Although tagging may look like an extremely basic application, RFID may well turn out to be the first *truly ubiquitous* computing technology. You might even call it “disposable computing”. . .

2.3.1 The Next Barcode

To a first approximation, once the price is right, there will be an RFID tag anywhere you now see a barcode. There are indeed many similarities between barcodes and the RFID tags poised to replace them—so many that it is easier just to focus on the differences. Conceptually, only two differences are significant: the *code space cardinality* and the *transmission mechanism*.

As for **code space cardinality**, the international EAN barcode standard only features 13 decimal digits and is therefore limited to a million manufacturers, each allowed to define up to 100,000 products. The Auto-ID Labs consortium, in contrast, defines a 96 bit code and a partitioning scheme that allows for 256 million manufacturers and 16 million products per manufacturer. More importantly, there are still enough bits left to provide 64 billion serial numbers for each individual product. While all the beer cans in the six-pack look the same to the barcode scanner, with RFID each razor blade cartridge in the whole warehouse will respond with its own distinct serial number.

As for **transmission mechanism**, optical acquisition means barcode and reader must be manually aligned, as anyone having ever queued at a supermarket checkout knows all too well. The RFID tag, instead, as its name indicates, uses radio frequency: it can be read without requiring line of sight.

From these two differences stem new opportunities. Because the code embeds a unique serial number, you can prove with your till receipt that *this* defective item was sold to you in *this* store; the manufacturer, in turn, when receiving the defective return from the retailer, knows exactly on what day and on which assembly line of which plant the item was made and can check for similar defects in any other items of the same batch. Because reading the code does not require a manual alignment operation, it may happen on a continuous basis rather than just at the checkout: smart shelves, both in the supermarket and in your own kitchen, can tally the products they contain and go online to reorder when they are running low.

2.3.2 Machine Vision—Without the Hard Bits

Andy Hopper's vision of Sentient Computing (Hopper, 2000) is one of systems with the ability to sense the world that surrounds them and to respond to stimuli in useful ways without the need for explicit prompting by their users. Despite phenomenal advances in machine vision in recent years, today's computer systems are not yet quite capable of "seeing" what surrounds them; but RFID, especially if universally deployed, makes a big difference here. Instead of fixing poor eyesight with expensive and technologically advanced solutions (contact lenses or laser surgery), we go for the cheap and cheerful alternative of wrapping a fluorescent jacket around each object to make it stand out. The myopic computer doesn't actually see the object: it just notices the fluorescent coat. Always remember this.

2.3.3 Technology

There are many varieties of tags using different working frequencies (implying different trade-offs between penetration, range and read rate), different power options (from passive tags powered by the radio wave of the reader to semi-passive and active tags that include batteries), different amounts of memory in the tag and different packaging options. They currently spread the cost spectrum from approximately 0.10 to 20 USD when bought in bulk. Proponents and manufacturers hope to bring down the cost of the cheapest tags by another order of magnitude; this won't be easy to accomplish but there is strong economic pressure towards that goal.

Much of the security discussion on RFID can be conducted at a relatively high level but there are a couple of technological points worth noting.

The process by which a reader device finds out which tags are within range must take into account the possibility of finding several such tags and must return the individual codes of all the tags in a reasonable time (imagine an automated supermarket checkout scenario). The process is known as **singulation** and can be implemented in several ways. An efficient one, in common use, is **binary tree walking**: for each bit of the code, the reader broadcasts a request, to which tags must respond with their corresponding bit. If there is no collision, the reader moves to the next bit; otherwise it visits the two subtrees separately, first asking the tags with a 1 in that position to be quiet and then vice versa.

Concerning transmission range: passive tags are energized by the radio wave carrying the reader's query and respond with a radio transmission of their own. This structure introduces a clear asymmetry: since the power of the transmission from the reader will necessarily be much higher than that from the tag, there will be a small spatial region near the tag in which both tag and reader can be overheard, but a much wider concentric region beyond that in which only the reader can be heard. This is significant for passive eavesdropping attacks.

There are significant technological constraints to the manufacture of tags meant to be batteryless and so cheap that they can be affixed to individual retail items without significantly increasing their cost. In practice the limits on gate count and power consumption mean that the tags' ability to compute cryptographic primitives is severely limited when not totally absent.

2.3.4 Security and Privacy

What risks are brought about by RFID systems? The first book to address this issue was probably mine (Stajano, 2002): as an attention-grabbing joke I suggested that sexual predators would take pleasure in reading the cup size, model and colour of the bra of their victims. Can you imagine any other risks? Katherine Albrecht has since become the most outspoken critic of the privacy threats introduced by RFID (Albrecht and McIntyre, 2005).

Apart from the numerous privacy risks for end-users, there are other kinds of concerns. What security problems can you anticipate for a retailer who tags all his products with RFID? Can you think of any frauds that the introduction of RFID tags makes easier?

RFID tags have been proposed as electronic price tags allowing automated check-out. It is not hard to imagine ways in which one might check out without paying, or paying for a cheaper product, by tampering with the price tags. Think in systems terms: even the glue (Johnston and Garcia, 1997) that sticks the tag to the object is part of the security envelope!

RFID tags might also become a vector for attacking the back-end systems: Melanie Rieback et al. (Rieback, Crispo, and Tanenbaum, 2006) showed how a maliciously reprogrammed tag could attack the back-end server through SQL injection or buffer overflow and even, in theory, propagate to infect further tags.

RFID tags have been proposed as anti-counterfeiting devices. How hard is it to clone them? (It depends to some extent on whether they have to look the same as the original to a visual inspection, or only at the radio interface.) See the later subsection on PUFs.

What about RFID tags used in access control situations, such as door entry cards and subway access cards? Do you expect the key card to be a passive or an active tag? What do you imagine the authentication protocol to look like? And how could a crook attack it to gain entry illegitimately? A replay attack (record a known-good response from a genuine user, then play it back later) is easy to mount if the key card always gives the same response. But, even if it doesn't, a *relay* attack (a specialized form of man-in-the-middle attack where the man-in-the-middle only forwards messages rather than attempting to alter or decrypt them) can open the door with the unsuspecting cooperation of a legitimate keyholder (Hancke, 2005). And the fact that regular users operate the turnstiles by dropping their handbag on the reader (rather than taking their wallet out of the bag and their card out of the wallet) makes it easier to conceal the attacking equipment—there is no need to make it resemble a genuine card.

2.3.5 Technical Safeguards

The first significant privacy-protecting solutions (hash-based access control, randomized access control and silent tree walking) were invented by Steve Weis et al. (Weis, Sarma, Rivest, and Engels, 2004) and will be examined next. Weis's master thesis (Weis, 2003) is a great source of good ideas. Ari Juels's extensive and authoritative survey (Juels, 2006) is an excellent reference, as is the book edited by Simson Garfinkel and Beth Rosenberg that collects interesting contributions from a wide spectrum of sources (Garfinkel and Rosenberg, 2005). Gildas Avoine currently maintains what is perhaps the web's most extensive bibliography on RFID security and privacy at `http://www.avoine.net/rfid/index.html`.

Killing the tag. A simple solution to many RFID privacy problems, and the most frequently cited by proponents of the tags, is to disable the tags permanently at checkout[13]. This was the standard mode of operation originally proposed by the AutoID Center, with tags responding to a "kill" command protected by an 8-bit "password". Discussion of denial of service attacks is left as an exercise for the reader.

This strategy prevents subsequent tracking but also disables any potential user benefits of the technology (e.g. fridge automatically reordering regular items about to run out, sorting refuse by type at the recycling centre. . .).

Hash-based access control. This scheme and the two that follow are due to Weis et al. (Weis, Sarma, Rivest, and Engels, 2004). To prevent unauthorized readers from accessing tag contents, each tag T is assigned a key k_T, which is hashed to yield a meta-ID $m_T = h(k_T)$. The tag just stores m_T, not k_T. Authorized readers are given the keys of the tags they can read. When a reader queries a tag, the tag responds with the non-invertible meta-ID m_T. The reader responds to that with the key k_T, found by reverse lookup in the reader's table of known tags. At that point the tag unlocks and sends its true ID.

Randomized access control. One problem with the previous scheme is that the meta-ID itself is a unique identifier, even though the real ID is unreadable. So the eavesdropper can't tell it's a black 16 GB iPod but it can tell it's the same item it saw yesterday and the day before. People can then be tracked by "constellations" of objects that move around with them, including their glasses, their watch, their wallet and their shoes.

In this alternative scheme, therefore, the tag responds to a reader query by generating a random number r and transmitting the pair $(r, h(ID||r))$. The reader performs a brute-force forward search on all the IDs it knows and eventually finds the one that matches. This solution ensures that the responses from the same tag are all different. Note that an attacker who does not know the tag's ID would have to brute-force an infeasibly large code space; however a legitimate user owning many tags will pay a high performance cost at every query, so this method is suitable for individuals but not for, say, supermarkets.

[13] An equivalent solution would be to affix the tag to the packaging rather than to the object.

Silent tree walking. Since the reader's messages can be overheard from very far away, a remote eavesdropper could learn the ID of a tag by observing which portions of the code tree are explored by the reader during singulation, even if the eavesdropper could not hear the replies of the tag itself. Weis et al. (Weis, Sarma, Rivest, and Engels, 2004) therefore modify the binary tree walking algorithm as follows.

When a reader asks tags in range to respond with the next bit of their code, if there is no collision then the bit common to all the tags that responded is a secret for the remote eavesdropper. When there is a collision, this shared secret can be used as a one-time pad to mask off the next bit to be transmitted. Therefore the reader splits the two sides of the tree not by asking for "who's got a 1 in the next bit" but rather "who's got a 1 when they xor the next bit with the secret prefix?" which leaves the remote eavesdropper in the dark[14].

Similarly, if the reader needs to transmit a value v confidentially to a tag, the tag can generate a one-time pad value p and send it to the reader, unheard by the remote eavesdropper. The reader will then transmit $v \oplus p$ to the tag.

The blocker tag. The blocker tag, proposed by Juels et al. (Juels, Rivest, and Szydlo, 2003), performs selective jamming of the singulation protocol. A subtree of the RFID code space (could just be, say, the half with the topmost bit set) is designated as a privacy zone and tags are moved into that subtree at checkout. Subsequently, any time a reader attempts to navigate the privacy protected subtree, a special RFID blocker tag, meant to be carried by the user in her shopping bag, responds to the singulation protocol pretending to be all the tags in that subtree at once. The reader is therefore stalled. In the user's home, though, the blocker tag no longer operates and the smart home can still recognize the tags as appropriate.

In a more polite variant, the blocker tag makes its presence known so that readers don't waste time attempting to navigate the privacy-protected subtree.

Anti-counterfeiting using PUFs. It has been suggested that RFID tags could be embedded in luxury goods (e.g. designer shoes) to prevent forgeries. The underlying assumption is that the technological capacity for manufacturing RFID chips is not widely available. This attitude is akin to a physical version of "security by obscurity". Can we do better?

A much stronger proposition is that of embedding a challenge-response black box inside an RFID chip (Tuyls and Batina, 2006) in such a way that the secret cannot be easily duplicated (because it is not merely a string of bits but a physical structure) and any attempt to open the chip to extract the secret by physical means will destroy the secret itself: basically, tamper-resistance at the RFID chip level. This is achieved by implementing the challenge-response box as a function

[14] There is a bootstrapping problem for this scheme: where does the secret come from in the first round? The authors handwave it away by stating that the population of tags being examined shares a common prefix, such as a manufacturer ID, which can be transmitted from tags to reader without being overheard by the distant eavesdropper. In practice, however, one should question the plausibility of both the common prefix assumption and of the implicit one that the eavesdropper wouldn't be able to *guess* that common prefix, if indeed it is something like a manufacturer's code.

keyed by some physical imperfections of the chip, such as parasitic electrical features of its coating or slight differences in the propagation delay of adjacent and nominally equal combinatory circuits. Opening up the chip to insert microprobes would alter those quantities and therefore change the "key". This kind of construction, indicated as as "Physical One-Way Function" by Ravikanth Pappu et al. (Pappu, Recht, Taylor, and Gershenfeld, 2002) who first proposed it with an implementation based on lasers and holograms, has also been called "Physical Random Function" and "Physical Unclonable Function" by Blaise Gassend et al. (Gassend, Clarke, van Dijk, and Devadas, 2002) who first described an integrated circuit implementation for it.

Suitable protocols have to be designed to cope with the fact that the device will *only probabilistically* respond in the same way to the same input (to some extent this can be remedied with error correction circuitry, but in the RFID environment gates are expensive) and with the fact that even the manufacturer doesn't actually know "the key"—he can only sample the device on a set of inputs before releasing it, and must remember the values somewhere.

Distance bounding protocols. Challenge-response protocols were designed to defeat replay attacks. However they are still subject to more subtle *relay* attacks. A relay attack, sometimes called a wormhole attack, is a specialized form of man-in-the-middle attack where the attacker does not alter the messages it overhears but just transports them to another location. A door might issue a challenge to verify the presence of an authorized entry card; but a relay attack would involve relaying that challenge to a legitimate entry card that is not currently near the door, and relaying the card's response to the door, which would find it genuine and grant access. Adding strong cryptography to the exchange cannot, by itself, fix the problem.

Distance-bounding protocols were introduced to defeat such relay attacks. The idea is to measure the response time and, from that and the speed of light (0.3 m/ns), infer that the responder cannot be any further than a certain distance. Then the door would ask the entry card to respond within a specified time, to be sure that the card is sufficiently close to the door.

The challenge-response protocol has to be specially designed in order to be used for distance bounding: just measuring a regular challenge-response exchange would provide timing information of insufficient precision for positioning purposes once we take into account the delays introduced by buffering, error correction, retransmission, cryptographic computations and so on. The fundamental technique introduced by Stefan Brands and David Chaum (Brands and Chaum, 1993) is to issue a single bit of challenge and expect an immediate single-bit response. This allows the verifier to measure just the round-trip time of one bit, eliminating most of the overheads. To prevent random guessing of the response by an adversary with 50% probability of success, this fast bit exchange is repeated for n rounds, reducing the probability of a correct guess to 2^{-n}.

Gerhard Hancke and Markus Kuhn (Hancke and Kuhn, 2005) developed a distance-bounding protocol optimized for the RFID environment: the tag, acting as prover, does not have to compute any expensive operations and the protocol, unlike others, works even on a channel subject to bit errors. The protocol also addresses the issue that passive RFID devices do not have an internal clock (they derive it from the radio wave of the reader) and therefore an attacker might attempt to supply the RFID token with a faster clock in order to make it respond more quickly, which would in turn allow the attacker to pretend to be closer to the verifier than he is.

Multi-factor access control in e-passports Modern "e-passports" incorporate an RFID chip containing a digitized photograph of the bearer and optionally other biometrics such as fingerprints and iris scans. To prevent unauthorized access to such data while the passport is in the owner's pocket or purse, the RFID chip will only release that information after a successful challenge-response in which the reader demonstrates knowledge of the data printed on the OCR strip of the main page of the passport. The intention is to ensure that a reader can only acquire the RFID chip data when the passport is open and has been optically scanned. Ari Juels, David Molnar and David Wagner (Juels, Molnar, and Wagner, 2005) offer a detailed description of this and related mechanisms. Markus Kuhn (Kuhn, 2003) suggested a simpler way to achieve the same effect: incorporate an RF shield in the front cover of the passport.

2.4 Authentication and Device Pairing

Most authentication methods ultimately rely on one of "what you know, what you have or what you are" (e.g. a password, a lock key, a fingerprint). But across a network, all you get are bits, and the bits of the crook smell the same as the bits of the honest user.

A variety of protocols have been devised for authentication in distributed systems, the most influential being probably Needham-Schroeder (Needham and Schroeder, 1978) which later gave rise to Kerberos (Kohl and Neuman, 1993) and ultimately to the authentication system of Windows 2000 and its successors. That family of protocols relies on a central authority that knows all the principals in its security domain and can act as introducer to any two entities wishing to communicate with each other. To use such a protocol, it is necessary for the central authority to be reachable.

Authentication with public key certificates may appear not to require connectivity, but it does if we take into account the need to check for *revoked* certificates.

In the ad-hoc wireless networks that are commonplace in ubiquitous computing, where online connectivity to a global authority cannot be guaranteed, we may therefore need a more local authentication solution.

In the ubicomp context, authentication is often really just an interaction between two devices: one which offers services and one which requests them. The latter device (client) is in some way a controller for the former (server), in that it causes the server to perform certain actions: one entity wants the other to "do something". Under this alternative viewpoint, authentication morphs into a kind of master-slave pairing. The server (or “slave” or “verifier”) authenticates the client (or “master”or “prover”) and, if the authentication succeeds, the server accepts the client as its master for the duration of that interaction session.

2.4.1 Big Stick

In more cases than you might at first think, the Big Stick principle (Stajano, 2002) is an appropriate access control policy:

> Whoever has physical control of the device is allowed to take it over.

Your fridge usually works that way. So does your DVD player. This is a strong, sound security policy because it closely matches the constraints already imposed by reality (when your made-up rules conflict with those of reality, the latter usually win). However it may not be appropriate for devices you must leave unattended outside your control, and it does nothing to deter theft.

2.4.2 Resurrecting Duckling

The Resurrecting Duckling security policy model (Stajano and Anderson, 1999) was created to solve the above problem and in particular to implement *secure transient association*: you want to bind a slave device (your new flat screen TV) to a master device (your cellphone used as a universal remote controller) in a **secure** way, so that your neighbour can’t turn your TV on and off by mistake (or to annoy you) and so that the stolen TV is useless because it doesn’t respond to any other remote controller; but also in a **transient** way, so that you can resell it without also having to give the buyer your cellphone.

It is based on four principles (Stajano, 2002):

1 - Two State principle. The entity that the policy protects, called the *duckling*, can be in one of two states: *imprintable* or *imprinted*. In the imprintable state, anyone can take it over. In the imprinted state, it only obeys its mother duck (q.v.).

2 - Imprinting principle. The transition from imprintable to imprinted, known as *imprinting*, happens when a principal, from then on known as the *mother duck*, sends an *imprinting key* to the duckling. This must be done using a channel whose confidentiality and integrity are adequately protected (physical contact is recommended). As part of the transaction, the mother duck must also create an appropriate backup of the imprinting key.

3 - Death principle. The transition from imprinted to imprintable is known as *death*. It may only occur under a very specific circumstance, defined by the particular variant of the Resurrecting Duckling policy model that one has chosen. Allowable circumstances, each corresponding to a different variant of the policy, include the following.

- Death by order of the mother duck (default).
- Death by old age after a predefined time interval.
- Death on completion of a specific transaction.

4 - Assassination principle. The duckling must be built in such a way that it will be uneconomical for an attacker to *assassinate* it, i.e. to cause the duckling's death artificially in circumstances other than the one prescribed by the Death principle of the policy.

Note that the Assassination principle implies that a duckling-compliant device must be endowed with some appropriate amount of tamper resistance.

The Resurrecting Duckling policy has very general applicability. It is not patented and therefore it has been applied in a variety of commercial situations.

It has been extended (Stajano, 2001) to allow the mother duck to delegate some or all of its powers to another designated master.

2.4.3 Multi-Channel Protocols

Security protocol descriptions usually consist of sequences of messages exchanged between two or more parties. It may however be significant to explore the properties inherently provided by different channels. When you receive a radio message, for example, it is hard to figure out for sure who sent it to you: there may be a name in the header, but it might be a fake. But if you receive a message over a visual channel (e.g. you scan a barcode) then the source is obvious, and is quite difficult for a middleperson to forge; so the visual channel implicitly provides *data origin authenticity*.

Consider other channels commonly used in ubiquitous computing and examine their useful properties: fingers pressing buttons on a keyboard; voice; infrared; direct electrical contact; optical fibres; quantum communication; and so on. Look at them in terms of confidentiality, integrity, source or destination authenticity, capacity, cost per bit, usability, whether they are point-to-point or broadcast and so on.

Sometimes, the best approach is to use different channels at different stages in the protocol, sending (say) message 3 over radio but message 4 over a visual channel. This applies particularly to the ad-hoc wireless networks that are commonly found in ubiquitous computing environments.

When two principals Alice and Bob wish to establish a shared secret although they have never met before, if the channel over which they communicate does not offer confidentiality, then an eavesdropper who overhears their exchange will also learn the secret they establish[15]. To overcome this problem, they may use a Diffie-Hellman key exchange[16]: Alice thinks of a secret a and sends g^a, Bob thinks of b and sends g^b, and both of them can compute g^{ab} (respectively from a and g^b or from

[15] Bluetooth pairing used to be affected by this problem (Jakobsson and Wetzel, 2001; Wong, Stajano, and Clulow, 2005).

[16] We omit all the "mod n" for brevity.

b and g^a) while Eve the eavesdropper can't (because she only has g^a and g^b and the discrete log problem is computationally hard).

This defeats the eavesdropper, but the next problem is the man-in-the-middle: an active attacker sits between the two parties and pretends to be Bob to Alice and to be Alice to Bob. The problem now is that Alice, when she establishes a shared secret using Diffie-Hellman *over radio*, can't say for sure whether she is establishing it with Bob or with middleperson Mallory. It would be secure, although very laborious, for them to exchange their g^a and g^b on pieces of paper, because then Alice would know for sure that g^b comes from the intended Bob, and vice versa; but radio waves, instead, might have come from anywhere. Unfortunately the data-origin-authentic channels (writing a note on a piece of paper, copying a string of hex digits from a display into a keypad, acquiring a visual code with a camera…) have usability constraints that limit their capacity to a small number of bits per message, insufficient to carry a full g^a. The idea of multi-channel protocols (Balfanz, Smetters, Stewart, and Wong, 2002; McCune, Perrig, and Reiter, 2005; Wong and Stajano, 2005) is to use a convenient, user-friendly and high capacity channel such as radio for the "long" messages, and a different, perhaps less convenient but data-origin-authentic, channel for short messages that somehow validate the ones exchanged over the other channel.

2.5 Beyond Passwords

Fifteen years ago, many computer systems did not even allow you to set a password longer than 8 characters. Nowadays, 8-character passwords can be trivially cracked by brute force, because computers are now fast enough to try all possible combinations. Many longer passwords can also be cracked by trying dictionary words, song lyrics, girl names, swear words, movie lines and a number of "`r&latively 0bv1ous Var1a$h0ns`". We have reached the stage where many of the passwords that a non-geek can plausibly remember are within reach of a brute-force search.

The problem is compounded by the proliferation of systems that require a password: nowadays most of us own and use several computers (an uncommon circumstance fifteen years ago) but, besides that, we are requested to use login passwords in many other computer-mediated interactions. Each online shopping web site, each blog, each email server, each ISP account, each online banking account requires a password, adding to the cognitive load. And, of course, reusing the same password for all your accounts makes you globally vulnerable to a malicious insider at any of the corresponding sites—or, more generally, reduces the security of all your accounts to that of the most insecure of them all.

In the context of ubiquitous computing, where each person is expected to interact daily with tens or hundreds of devices, it should be clear that depending on passwords for authentication is not going to be a long term solution. What are the alternatives? We already mentioned the well-known taxonomy of authenticators whose categories are “something you know, something you have or something you are”[17]. If “something you know” is no longer suitable, let’s then explore the other two options in greater detail.

2.5.1 Tokens: Something You Have

In token-based authentication, you are authenticated based on the fact that you own a physical artefact. Outside computers, this has been a common practice for centuries: rings, seals, medallions, passports and indeed (mechanical) keys are examples. An obvious problem we mentioned is that the procedure verifies the presence of the token, not of the person carrying it, and as such it fails if the token is forgotten, lost or stolen and presented by someone else.

Methods for binding the token to the intended person have been devised, such as the wrist or ankle straps used for attaching a radio tag to prisoners released on probation (if the strap is opened or cut in order to remove the tag, the radio signal stops, triggering an alarm at the back-end monitoring system). Their use on non-prisoners carries unpleasant overtones. Moreover, if the token gives access to anything valuable (e.g. a bank vault), the fact that it stops working when the strap is opened may be a perverse incentive to kidnap the wearer or chop off the limb[18].

Physical tokens like door keys can be cloned by someone with physical access to them. More worryingly, electronic tokens that communicate via a wireless signal are subject to replay attacks, where the signal is recorded and later replayed. For that reason, all but the very worst wireless tokens use a challenge-response mechanism, where each challenge is different and an attacker is not supposed to be able to predict the response given the challenge.

An important advantage of the wireless token-checking arrangement is that the verifier can interrogate the token many times, for example once a minute, to check whether the authenticated party is still there[19]. With passwords, instead, if the session lasts for three hours, the verifier only knows that the authenticated principal was there at the start of the three hours but has no real guarantees that she continued to be there throughout.

A more subtle problem is that of relay (not replay) attacks, already mentioned in the section on RFID: if the token is, for instance, a wireless tag that opens a door when placed on the door’s pad, an attacker will place a fake tag near the door pad and a fake door pad (actually just its electronics) near the tag and create a *wormhole* between them.

[17] Sometimes, for added security at the cost of a little extra hassle for the user, the verifier might check more than one such item, preferably from different categories (for example ownership of a card and knowledge of its PIN): we then speak of “multi-factor authentication”.

[18] The same comment applies to biometrics, examined next.

[19] Imagine the case of wanting to keep a laptop locked, with all files encrypted, unless in presence of its owner (Corner and Noble, 2002).

2.5.2 Biometrics: Something You Are

A biometric feature which has been studied and used long before computers is the fingerprint. Some laptops and cellular phones now include basic fingerprint readers. Other biometrics used for recognition of individuals include hand geometry, voice, face (geometry of facial features), retina and iris. An unavoidable problem is that the response from the recognition system cannot be perfect: there is always a certain amount of misclassification giving rise to false positives (verifier recognizes user when it shouldn't; this allows fraud) and false negatives (verifier doesn't recognize user when it should; this insults the legitimate user). It is easy to reduce one at the expense of the other by moving the recognition threshold but it is difficult to reduce both. Technically, the system offering the best performance nowadays is iris recognition, used in airport security in several countries around the world (Daugman, 2004).

A problem of biometrics as opposed to other authentication methods is that your biometric features are yours for life and therefore, barring surgery, they can't be revoked. If your thumbprint opens your front door and someone clones it and raids your apartment, when you install a replacement door lock you can't go to the locksmith and get a new thumbprint.

Another problem, this time from the privacy perspective, is that using the same biometric feature for authenticating to a multitude of different services makes you traceable across these various services should they share and compare such authentication data. And, on the subject of reuse of authentication credentials, if a thumbprint reader is installed not only at your front door but also at your company and on your safety deposit box at your bank, does this mean that a malicious insider at your company now has what it takes to make a gummy finger (Matsumoto, Matsumoto, Yamada, and Hoshino, 2002) and raid your house and safety deposit box? For that matter, is it rational to use fingerprints and iris codes as keys that protect anything valuable now that, under US-VISIT, you must give them to border security officers every time you travel to the United States?

A common fallacy is to assume that recognizing a valid biometric feature (e.g. recognizing that what's being presented at the reader is Joe's fingerprint or Joe's iris scan) is authentication, since at the same time you say who you are and you prove it, under the assumption that no one else has that biometric. But it's not quite *authentication*: it's actually closer to just *identification*. Although there is admittedly an element of verification in the process of checking whether what's being presented at the reader matches with Joe's fingerprint or iris as previously recorded in the verifier's database, to compete the verification the verifier would also need to check that the feature is in fact genuinely "attached" to the person who was born with it, and is not the result of applying a gummy finger or a printed contact lens. This is hard to ensure and almost impossible in unattended settings where the scanning is not supervised by a human officer. For this reason, locking a laptop or a PDA with a fingerprint does not offer strong protection—particularly when you consider that the owner may have even left fingerprints on the object itself, from which a cloned finger could be produced. It's relatively OK to assume that the biometric is unique to an individual, but it is wrong to assume that it is a secret.

One may argue that something similar happens with a token. Indeed so, and it is important to remember that what is being authenticated is the presence[20] of the token, from which the presence of its intended owner does not necessarily follow. The difference, perhaps more psychological than theoretical, is that it is normal to think of the biometric as being closely associated with its owner ("something you are"), as opposed to the token ("something you have") where it is more natural to assume that the two may at some point become separated.

3 Going Up a Level: the Rest of the Story

We could go on for much longer looking for threats and discussing technical security solutions for many other plausible ubicomp scenarios: how about, say, hordes of Internet-enabled home appliances being remotely "0wned" by organized crime and recruited into evil botnets in order to send spam or perform DDOS attacks? However, to provide effective security, once we have a general feeling for the issues that challenge us and the technical solutions we may use to address them, it is also important to step back and look at the field from a higher viewpoint. Often the true root of the problem is not strictly technical and therefore a technical solution will only act as a temporary and ineffective fix.

3.1 Security and Usability

Usability is probably one of the most difficult challenges for security today (Cranor and Garfinkel, 2005). Even though we have a pretty good understanding of low level security mechanisms and protocols (for instance, we are able to build efficient and robust ciphers, such as AES, that even major spy agencies cannot crack by brute force), we are not very good yet at building security systems that don't get in the way of users. The problem is that, if a security mechanism is too difficult or too inconvenient to use, it will just not be used. The user will take whatever steps are necessary to bypass the security in order to get her job done. Introducing strong security mechanisms without concern for their repercussions in terms of human factors is not a good recipe for building a secure system.

As we made clear in one of the initial sections, security is not a feature or a component that can be added later: it is instead a global property of an entire system, just like sound quality in a hi-fi system. Engineering a system for security is similar to engineering it for robustness; but, while robustness is about protecting the

[20] But recall the earlier discussion about relay attacks: unless a distance bounding protocol is employed, not even the presence of the token may be guaranteed.

system from accidental damage, security is about protecting it from damage that is purposefully caused by malicious attackers.

Usability, just like security, is not a feature or a component but a system property. It is not about "building a user interface" but about understanding how the user interacts with the system and understanding where the system fails because it is counterintuitive, confusing or frustrating. It is about designing and refining the system so that the user's intention can easily be translated into action without ambiguity, difficulty or stress. It is about viewing the system with someone else's eyes and realising that what is obvious, intuitive and easy for the designer may not be so for the user.

In the interaction between security and usability there is often a tension: essentially, security is about limiting the user's actions whereas usability is about facilitating them. Fortunately, the situation is not as contradictory as it may sound from the above comment, because security is about limiting *bad* actions while usability is about facilitating *good* actions. However it is undeniable that there is a tension and that security mechanisms, in their attempts to impede the bad actions of bad users, often get in the way of the honest regular users who just want to pursue their honest activities as simply as possible: witness for example wireless LANs left at their default insecure settings[21], Bluetooth pairing PINs (and cellphone PINs and any other kind of factory-supplied PINs or passwords) left at their easily-guessable default value, security-related dialog boxes that ask for a confirmation from the user but are only perceived as "Here's a hard question you won't understand anyway, but press OK if you want to keep working otherwise the computer will stop"; and so on. (Security has been described as a "tax on the honest" caused by the existence of the dishonest.) As we said, when security and usability fight, it's usability that wins, in the sense that the user will probably prefer to switch off the security entirely. In a constrained situation—such as, for example, that of a military environment—it may be possible for the security designer to impose a security mechanism on the user regardless of how inconvenient it is to operate; but, in the civilian context of, say, the consumer electronics market, building a system that is too inconvenient to operate because of its poorly designed security features will only lead customers to vote with their wallets and choose a competing product. Especially since very few customers choose a product based on its security rather than its features or performance.

3.2 Understanding People

It is important to be able to view one's design through someone else's eyes. For security, we must view our systems through the eyes of an *attacker*.

- Where are the weaknesses?

[21] And sold in that configuration because "insecure by default, but just works" is much better for the manufacturer than having the customer return their wireless access point and buy a competitor's because they were incapable of setting up WPA.

- Where are the valuable assets?
- How can I misuse the system in order to break it?
- What implicit assumptions of the designer can I violate in order to make the system behave in an unexpected way?
- Where can I cause maximum damage?
- What are my motivations in attacking the system?
- What do I stand to gain from it?
- What attacks can I deploy on an industrial scale, replicating them with little effort on all the devices of this model?

It is not easy to acquire this mindset; but, if we cannot ask these questions ourselves, we'll leave it to the real attackers to do it for us once the product is fielded.

For usability, too, we need to adopt someone else's viewpoint; but this time we must view the system through the eyes of a *user*.

- How can I make the product do what I want? (I know what I want, but not how to cause this machine to do it.)
- Why doesn't what I do (which seems natural to me) work?
- I find this activity too hard.
- I find this other activity not difficult, but annoying and frustrating—it insults my intelligence.
- I am not enjoying this.
- I forget how to do this.
- It's not obvious.
- The display is hard to see.
- It's too small to read at my age.
- I can't figure out what the icon represents: is it a cloud, a splatted tomato or a devil?
- This other icon I can see, but I can't understand the metaphor—what does a picture of a toilet roll mean in this context?
- I can't understand the response of the system.
- It didn't do what I thought it would.

And so forth. It takes much humility for a designer to listen to and accept all the criticism implicit in such comments. But they are like gold dust. Learning to listen to the user, to understand the user's viewpoint and to think like the user is an essential step for progress towards an intuitive and pleasant to use system that reduces the possibility of accidental mistakes.

In that spirit, the well-thought-out security specification of the One Laptop Per Child project (Krstić, 2007) makes for instructive and fascinating reading. Of particular interest are the "principles and goals" which demonstrate a laudable explicit concern towards making security usable even for very young children: for example,

> Security cannot depend upon the user's ability to read a message from the computer and act in an informed and sensible manner. [...] a machine must be secure out of the factory if given to a user who cannot yet read.

3.3 Motivations and Incentives

Most systems are actually marketed and sold on the basis of their features rather than of their security. This provides little incentive for vendors to compete on providing the best security. Often, security-related components are totally absent, or are added as an afterthought to count as extra "features" or to satisfy a regulatory requirement. Needless to say, this ad hoc approach cannot deliver a great deal of proper security at the system level. To make matters worse, customers do not usually understand security at the system level—they only worry about specific incidents—and are not usually in a position to evaluate whether the system on offer would or would not be secure for the intended purpose, meaning that there is no feedback loop to close in order to improve security at the next iteration, except if independent evaluation is provided by competent experts.

This disillusioned and somewhat cynical description applies across the board but it is certainly no less true for ubiquitous computing than for the general case. For example, when we worked with civil engineers to monitor bridges and tunnels for signs of deterioration using wireless sensor networks, we performed penetration testing on the WSN hardware that our team had chosen and we found a variety of security holes, from subtle ones to others that were quite blatant (Stajano, Cvrcek, and Lewis, 2008). We duly reported our findings to the manufacturer, months ahead of publication, but it became clear from their responses that, at least for the time being, security was not a commercial priority. Similar examples come from the worlds of Wi-Fi and Bluetooth where products shipped with weak and easy to break authentication procedures until researchers exposed the vulnerabilities (Borisov, Goldberg, and Wagner, 2001; Jakobsson and Wetzel, 2001), forcing new standards to be promulgated and adopted. The same also happened with RFID tags and cards (Bono, Green, Stubblefield, Juels, Rubin, and Szydlo, 2005; Garcia, de Koning Gans, Ruben Muijrers, Verdult, Schreur, and Jacobs, 2008) and in the latter case NXP, maker of the MIFARE Classic card, sued Radboud University in an attempt to block the above publication describing the break (Mills, 2008).

The blame doesn't rest exclusively with vendors, though: customers vote with their wallet, and largely get what they ask for. The generic lack of security is therefore also due to the fact that customers don't appear to place a high value on it, as shown by their unwillingness to pay[22] any substantial extra premium to obtain it (Acquisti, 2004). You may design strong security and privacy protection features only to see that your users don't actually care and just leave them all disabled for simplicity.

As far as privacy goes, there are in fact several powerful entities with strong incentives to intrude on the privacy of ordinary citizens. A brilliant paper by Andrew Odlyzko (Odlyzko, 2003) provides the counterintuitive but fundamental insight that knowing private information about you (and in particular about your ability and

[22] Though, some argue, there will also be cynical users who are unwilling to pay not because they don't value privacy but because they don't believe that what's on offer at a premium will provide it.

willingness to pay for specific goods and services) is economically valuable because it allows vendors to engage in *price discrimination*, selling the same goods at a different price to each buyer in order to extract the maximum price that each buyer is willing to pay—in an extreme version of what airlines have already been doing for years. Similarly, insurance companies have an obvious incentive to find out as much as possible about their prospective customers in order to steer away, or charge more, the prospects that are more likely to come up with expensive claims. Governments cite national security as their main reason for invading the privacy of citizens and travellers, though one should seriously question whether the response is proportionate to the actual risk (Schneier, 2003). A report by civil liberties watchdog Statewatch (Bunyan, 2008) quoted a paper from the EU's "Future Group" (Future Group, 2007) as saying:

> Every object the individual uses, every transaction they make and almost everywhere they go will create a detailed digital record. This will generate a wealth of information for public security organisations, and create huge opportunities for more effective and productive public security efforts.

The report commented:

> When traffic data including internet usage is combined with other data held by the state or gathered from non-state sources (tax, employment, bank details, credit card usage, biometrics, criminal record, health record, use of e-government services, travel history etc) a frightening detailed picture of each individual's everyday life and habits can be accessed at the click of a button.

None of this would have been technically possible before the age of ubiquitous computing and the associated ubiquitous back-end database servers.

Ubiquitous computing is unquestionably changing society and affecting the lives of all of us. The pervasiveness of invisible computer systems in modern society has the potential for making many hard things simpler and for making many interesting things possible; but at the same time it has the potential for exposing us to many new risks to our safety and privacy that did not previously exist and which we, as individuals, are not well equipped to address.

As responsible architects of the new world of ubicomp we have a duty to protect the unwary from threats they cannot face on their own, particularly as they originate in innovations that they did not request or ever fully understand. But this is a battle that public-spirited system designers cannot fight alone: systems will never be secure unless users value and demand that they be. The first step is therefore to educate the public about security and risks, in order that they can use security as one of the criteria upon which to base their choice of which systems to support or buy. The second step is to understand, by carefully and humbly listening to the users' own viewpoint, what are the assets to be protected, and the extent to which it is reasonable to go through inconvenience and expense in order to protect them. This trade-off is at the core of security and, in the world of ubiquitous computing, it is essential that everyone learn to find the appropriate balance.

Acknowledgements This chapter is based on copyrighted material from my book *Security for Ubiquitous Computing* (both from the first edition published by Wiley in 2002 and from the draft of a forthcoming second edition). Some of this material started as studies and reports I wrote for my consultancy clients, to whom I am grateful for the opportunity of rooting my research in real world problems. I also thank Stuart Wray, Bruce Christianson and Bernd Eckenfels for detailed and insightful feedback on an earlier draft, as well as Nick Towner, Pete Austin and a blog reader signing off as “Lil-Wayne Quotes” for shorter but equally valuable comments.

About the Author Frank Stajano, PhD, is a University Senior Lecturer (≈ associate professor) at the Computer Laboratory of the University of Cambridge, UK. His research interests cover security, ubiquitous computing and human factors, all linked by the common thread of protecting security and privacy in the modern electronic society. Before his academic appointment he worked as a research scientist in the corporate research and development laboratories of Toshiba, AT&T, Oracle and Olivetti. He was elected a Toshiba Fellow in 2000. He continues to consult for industry on a variety of topics from security to strategic research planning. Since writing *Security for Ubiquitous Computing* he has given about 30 keynote and invited talks in three continents. He is also a licensed martial arts instructor and runs the kendo dojo of the University of Cambridge. Visit his web site by typing his name into your favourite search engine.

References

Acquisti A (2004) Privacy in electronic commerce and the economics of immediate gratification. In: Proceedings of the ACM Electronic Commerce Conference (ACM EC), ACM, URL http://www.heinz.cmu.edu/~acquisti/papers/privacy-gratification.pdf

Albrecht K, McIntyre L (2005) Spychips: How Major Corporations and Government Plan to Track Your Every Move with RFID. Thomas Nelson

Amoroso E (1994) Fundamentals of Computer Security Technology. Prentice Hall

Anderson R (2008) Security Engineering (Second Edition). Wiley

Ark WS, Selker T (1999) A look at human interaction with pervasive computers. IBM Syst J 38(4):504–507

Balfanz D, Smetters DK, Stewart P, Wong HC (2002) Talking to strangers: Authentication in ad-hoc wireless networks. In: Proceedings of Network and Distributed System Security Symposium (NDSS 2002), The Internet Society, URL http://www.isoc.org/isoc/conferences/ndss/02/papers/balfan.pdf

BBC (2007) Thousands of driver details lost. URL http://news.bbc.co.uk/1/hi/northern_ireland/7138408.stm

Beresford A, Stajano F (2003) Location privacy in pervasive computing. IEEE Pervasive Computing 2(1):46–55, URL http://www.cl.cam.ac.uk/~fms27/papers/2003-BeresfordSta-location.pdf

Bono SC, Green M, Stubblefield A, Juels A, Rubin AD, Szydlo M (2005) Security analysis of a cryptographically-enabled rfid device. In: McDaniel P (ed) Proceedings of the 14th USENIX Security Symposium, USENIX Association, pp 1–16, URL http://www.usenix.org/events/sec05/tech/bono/bono.pdf

Borisov N, Goldberg I, Wagner D (2001) Intercepting mobile communications: the insecurity of 802.11. In: MobiCom ’01: Proceedings of the 7th annual international conference on Mobile computing and networking, ACM, New York, NY, USA, pp 180–189, DOI http://doi.acm.org/10.1145/381677.381695

Brands S, Chaum D (1993) Distance-bounding protocols (extended abstract). In: Helleseth T (ed) Advances in Cryptology—EUROCRYPT 93, Springer-Verlag, 1994, Lecture Notes in Computer Science, vol 765, pp 344–359

Bunyan T (2008) The shape of things to come. Tech. rep., Statewatch, URL http://www.statewatch.org/analyses/the-shape-of-things-to-come.pdf

Chaum DL (1981) Untraceable electronic mail, return addresses, and digital pseudonyms. Commun ACM 24(2):84–88, DOI http://doi.acm.org/10.1145/358549.358563

Collins T (2007) Loss of 1.3 million sensitive medical files in the us. URL http://www.computerweekly.com/blogs/tony_collins/2007/07/loss-of-13-million-sensitive-m.html

Corner MD, Noble BD (2002) Zero-interaction authentication. In: MobiCom ’02: Proceedings of the 8th annual international conference on Mobile computing and networking, ACM, New York, NY, USA, pp 1–11, DOI http://doi.acm.org/10.1145/570645.570647, URL http://www.sigmobile.org/awards/mobicom2002-student.pdf

Cranor LF, Garfinkel S (2005) Security and Usability: Designing Secure Systems that People Can Use. O’Reilly

Cvrcek D, Matyas V, Kumpost M, Danezis G (2006) A study on the value of location privacy. In: Proc. Fifth ACM Workshop on Privacy in the Electronic Society (WPES), ACM, pp 109–118, URL http://www.fi.muni.cz/usr/matyas/PriceOfLocationPrivacy_proceedings.pdf

Danezis G, Lewis S, Anderson R (2005) How much is location privacy worth? In: Proceedings of Workshop on Economics of Information Security (WEIS), URL http://infosecon.net/workshop/pdf/location-privacy.pdf

Daugman J (2004) How iris recognition works. IEEE Transactions on Circuits and Systems for Video Technology 14(1), URL http://www.cl.cam.ac.uk/users/jgd1000/csvt.pdf

Ducatel K, Bogdanowicz M, Scapolo F, Leijten J, Burgelman JC (2001) Scenarios for ambient intelligence in 2010. Tech. rep., European Commission, Information Society Directorate-General, URL ftp://ftp.cordis.europa.eu/pub/ist/docs/istagscenarios2010.pdf

Future Group (2007) Public security, privacy and technology in europe: Moving forward. Tech. rep., European Commission Informal High Level Advisory Group on the Future of European Home Affairs Policy, URL http://www.statewatch.org/news/2008/jul/eu-futures-dec-sec-privacy-2007.pdf

Garcia FD, de Koning Gans G, Ruben Muijrers PvR, Verdult R, Schreur RW, Jacobs B (2008) Dismantling mifare classic. In: Jajodia S, Lopez J (eds) 13th European Symposium on Research in Computer Security (ESORICS 2008), Springer, Lecture Notes in Computer Science, vol 5283, pp 97–114

Garfinkel S, Rosenberg B (eds) (2005) RFID : Applications, Security, and Privacy. Addison-Wesley

Gassend B, Clarke D, van Dijk M, Devadas S (2002) Controlled physical random functions. In: ACSAC '02: Proceedings of the 18th Annual Computer Security Applications Conference, IEEE Computer Society, Washington, DC, USA, p 149

Gemmell J, Bell G, Lueder R, Drucker S, Wong C (2002) Mylifebits: fulfilling the memex vision. In: MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia, ACM, New York, NY, USA, pp 235–238, DOI http://doi.acm.org/10.1145/641007.641053

Gemmell J, Williams L, Wood K, Lueder R, Bell G (2004) Passive capture and ensuing issues for a personal lifetime store. In: CARPE'04: Proceedings of the the 1st ACM workshop on Continuous archival and retrieval of personal experiences, ACM, New York, NY, USA, pp 48–55, DOI http://doi.acm.org/10.1145/1026653.1026660

Gruteser M, Grunwald D (2003) Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of MobiSys 2003, The Usenix Association, San Francisco, CA, USA, pp 31–42, URL http://www.winlab.rutgers.edu/~gruteser/papers/gruteser_anonymous_lbs.pdf

Hancke G (2005) A practical relay attack on iso 14443 proximity cards. URL http://www.cl.cam.ac.uk/~gh275/relay.pdf

Hancke GP, Kuhn MG (2005) An rfid distance bounding protocol. In: SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks, IEEE Computer Society, Washington, DC, USA, pp 67–73, DOI http://dx.doi.org/10.1109/SECURECOMM.2005.56, URL http://www.cl.cam.ac.uk/~mgk25/sc2005-distance.pdf

Heffernan B, Kennedy E (2008) Alert as 170,000 blood donor files are stolen. URL http://www.independent.ie/national-news/alert-as-170000-blood-donor-files-are-stolen-1294079.html

Hopper A (2000) The clifford paterson lecture, 1999: Sentient computing. Phil Trans R Soc Lond A 358(1773):2349–2358, URL http://www.cl.cam.ac.uk/Research/DTG/lce-pub/public/files/tr.1999.12.pdf

Jackson IW (1998) Who goes here? confidentiality of location through anonymity. PhD thesis, University of Cambridge, URL http://www.chiark.greenend.org.uk/~ijackson/thesis/

Jakobsson M, Wetzel S (2001) Security weaknesses in bluetooth. In: Naccache D (ed) CT-RSA, Springer, Lecture Notes in Computer Science, vol 2020, pp 176–191, URL http://link.springer.de/link/service/series/0558/bibs/2020/20200176.htm

Johnston RG, Garcia AR (1997) Vulnerability assessment of security seals. Journal of Security Administration 20(1):15–27, URL http://lib-www.lanl.gov/la-pubs/00418796.pdf

Juels A (2006) Rfid security and privacy: A research survey. IEEE Journal on Selected Areas in Communication 24(2), URL http://www.rsasecurity.com/rsalabs/staff/bios/ajuels/publications/pdfs/rfid_survey_28_09_05.pdf, the draft version on the author's web site is more detailed than the one eventually published in J-SAC.

Juels A, Rivest RL, Szydlo M (2003) The blocker tag: Selective blocking of rfid tags for consumer privacy. In: Atluri V (ed) Proc. 8th ACM Conference on Computer and Communications Security, ACM Press, pp 103–111, URL http://www.rsasecurity.com/rsalabs/staff/bios/ajuels/publications/blocker/blocker.pdf

Juels A, Molnar D, Wagner D (2005) Security and privacy issues in e-passports. In: Proceedings of International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm 2005), IEEE Computer Society, Los Alamitos, CA, USA, pp 74–88, DOI http://doi.ieeecomputersociety.org/10.1109/SECURECOMM.2005.59

Kohl J, Neuman C (1993) The kerberos network authentication service (v5). RFC 1510, IETF, URL http://www.ietf.org/rfc/rfc1510.txt

Krstić I (2007) Bitfrost. URL http://wiki.laptop.org/go/Bitfrost, draft 19 release 1, timestamped 2007-02-07, but last edited on 2009-01-04.

Kuhn M (2003) Rfid friend and foe, with a note on biometric passports. The RISKS Digest 22, URL http://catless.ncl.ac.uk/Risks/22.98.html@subj7#subj7

Mann S (1995) Privacy issues of wearable cameras versus surveillance cameras. URL http://wearcam.org/netcam_privacy_issues.html

Mann S (1996) 'smart clothing': Wearable multimedia and 'personal imaging' to restore the balance between people and their intelligent environments. In: Proceedings, ACM Multimedia 96, Boston, MA, pp 163–174, URL http://wearcam.org/acm-mm96.htm

Mann S (1997) Wearable computing: A first step toward personal imaging. Computer 30(2):25–32, URL http://www.wearcam.org/ieeecomputer/r2025.htm

Mann S, Picard RW (1997) Video orbits of the projective group: A simple approach to featureless estimation of parameters. IEEE Transactions on Image Processing 6:1281–1295

Matsumoto T, Matsumoto H, Yamada K, Hoshino S (2002) Impact of artificial gummy fingers on fingerprint systems. In: Proceedings of SPIE, vol 4677, Optical Security and Counterfeit Deterrence Techniques IV, URL http://cryptome.org/gummy.htm

McCune JM, Perrig A, Reiter MK (2005) Seeing-is-believing: Using camera phones for human-verifiable authentication. In: SP '05: Proceedings of the 2005 IEEE Symposium on Security and Privacy, IEEE Computer Society, Washington, DC, USA, pp 110–124, DOI http://dx.doi.org/10.1109/SP.2005.19

Mills E (2008) Dutch chipmaker sues to silence security researchers. URL http://news.cnet.com/8301-10784_3-9985886-7.html

Needham RM, Schroeder MD (1978) Using encryption for authentication in large networks of computers. Communications of the ACM 21(12):993–999

Norman DA (1998) The Invisible Computer: Why Good Products Can Fail, the Personal Computer Is So Complex, and Information Appliances Are the Solution. MIT Press

Odlyzko AM (2003) Privacy, economics, and price discrimination on the internet. In: Sadeh N (ed) ICEC2003: Fifth International Conference on Electronic Commerce, ACM, pp 355–366, URL http://www.dtc.umn.edu/~odlyzko/doc/privacy.economics.pdf

Pappu R, Recht B, Taylor J, Gershenfeld N (2002) Physical one-way functions. Science 297(5589):2026–2030, URL http://www.sciencemag.org/cgi/reprint/297/5589/2026.pdf

Rieback MR, Crispo B, Tanenbaum AS (2006) Is your cat infected with a computer virus? In: PERCOM '06: Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications, IEEE Computer Society, Washington, DC, USA, pp 169–179, DOI http://dx.doi.org/10.1109/PERCOM.2006.32

Schneier B (2003) Beyond Fear. Copernicus (Springer)

Stajano F (2001) The resurrecting duckling—what next? In: Christianson B, Crispo B, Malcolm JA, Roe M (eds) Security Protocols, 8^{th} International Workshop, Revised Papers, Springer-Verlag, Lecture Notes in Computer Science, vol 2133, pp 204–214, URL http://www.cl.cam.ac.uk/~fms27/papers/duckling-what-next.pdf

Stajano F (2002) Security for Ubiquitous Computing. Wiley, URL http://www.cl.cam.ac.uk/~fms27/secubicomp/

Stajano F (2004) Will your digital butlers betray you? In: Syverson P, di Vimercati SDC (eds) Proceedings of the 2004 Workshop on Privacy in the Electronic Society, ACM, Washington, DC, USA, pp 37–38, URL http://www.cl.cam.ac.uk/~fms27/papers/2004-Stajano-butlers.pdf

Stajano F, Anderson R (1999) The resurrecting duckling: Security issues in ad-hoc wireless networks. In: Christianson B, Crispo B, Malcolm JA, Roe M (eds) Security Protocols, 7^{th} International Workshop, Proceedings, Springer, Lecture Notes in Computer Science, vol 1796, pp 172–182, URL http://www.cl.cam.ac.uk/~fms27/papers/1999-StajanoAnd-duckling.pdf

Stajano F, Cvrcek D, Lewis M (2008) Steel, cast iron and concrete: Security engineering for real world wireless sensor networks. In: Bellovin SM, Gennaro R, Keromytis AD, Yung M (eds) Proceedings of 6th Applied Cryptography and Network Security Conference (ACNS 2008), Lecture Notes in Computer Science, vol 5037, pp 460–478, DOI 10.1007/978-3-540-68914-0_28, URL http://www.cl.cam.ac.uk/~fms27/papers/2008-StajanoCvrLew-steel.pdf

Streitz NA, Kameas A, Mavrommati I (eds) (2007) The Disappearing Computer, Interaction Design, System Infrastructures and Applications for Smart Environments, Lecture Notes in Computer Science, vol 4500, Springer

Tuyls P, Batina L (2006) Rfid-tags for anti-counterfeiting. In: Pointcheval D (ed) Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2006, Proceedings, Springer, Lecture Notes in Computer Science, vol 3860, pp 115–131

Want R, Hopper A, Falcao V, Gibbons J (1992) The active badge location system. ACM Transactions on Information Systems 10(1):91–102, DOI http://doi.acm.org/10.1145/128756.128759, URL http://www.cl.cam.ac.uk/Research/DTG/publications/public/files/tr.92.1.pdf

Weis SA (2003) Security and Privacy in Radio-Frequency Identification Devices. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA 02139

Weis SA, Sarma SE, Rivest RL, Engels DW (2004) Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In: Security in Pervasive Computing, Lecture Notes in Computer Science, vol 2802, pp 201–212

Weiser M (1991) The computer for the twenty-first century. Scientific American 265(3):94–104, URL http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html

Weiser M, Seely Brown J (1997) The coming age of calm technology. In: Denning PJ, Metcalfe RM (eds) Beyond Calculation: The Next Fifty Years of Computing, Springer-Verlag, pp 75–85, URL http://www.ubiq.com/hypertext/weiser/acmfuture2endnote.htm, previously appeared as "Designing Calm Technology" in *PowerGrid Journal*, v 1.01, July 1996, http://powergrid.electriciti.com/1.01.

Wong FL, Stajano F (2005) Multi-channel protocols. In: Christianson B, Crispo B, Malcolm JA (eds) Security Protocols, 13^{th} International Workshop Proceedings, Springer-Verlag, Lecture Notes in Computer Science, vol 4631, pp 112–127, URL http://www.cl.cam.ac.uk/~fms27/papers/2005-WongSta-multichannel.pdf, see also updated version in *IEEE Pervasive Computing* 6(4):31–39 (2007) at http://www.cl.cam.ac.uk/~fms27/papers/2007-WongSta-multichannel.pdf

Wong FL, Stajano F, Clulow J (2005) Repairing the bluetooth pairing protocol. In: Christianson B, Crispo B, Malcolm JA (eds) Security Protocols, 13^{th} International Workshop Proceedings, Springer-Verlag, Lecture Notes in Computer Science, vol 4631, pp 31–45, URL http://www.cl.cam.ac.uk/~fms27/papers/2005-WongStaClu-bluetooth.pdf