Authentication for Pervasive Computing

Sadie Creese[1], Michael Goldsmith[3,4], Bill Roscoe[2,3], and Irfan Zakiuddin[1]

[1] QinetiQ Trusted Information Management, Malvern, UK
{I.Zakiuddin,S.Creese}@eris.QinetiQ.com
[2] Oxford University Computing Laboratory
Bill.Roscoe@comlab.ox.ac.uk
[3] Formal Systems (Europe) Ltd.
{michael,awr}@fsel.com
http://www.fsel.com
[4] Worcester College, University of Oxford

Abstract. Key management is fundamental to communications security, and for security in pervasive computing sound key management is particularly difficult. However, sound key management itself depends critically on sound authentication. In this paper we review current notions of entity authentication and discuss why we believe these notions are unsuitable for the pervasive domain. We then present our views on how notions of authentication should be revised to address the challenges of the pervasive domain, and some of the new research problems that will arise. We end with some brief thoughts on how our revised notions may be implemented and some of the problems that may be encountered.

1 Introduction

1.1 Ambient Intelligence and Security

The Ambient Intelligence World. The pervasive computing paradigm fore- sees communicating and computational devices pervading all parts of our environment, from our physical selves, to our homes, offices, streets and so forth. Humans will be surrounded by intelligent and intuitive interfaces capable of providing information and communication facilities efficiently and effectively. Systems will recognise the presence of individuals, perhaps even their mood, in an unobtrusive manner, modifying their functionality according to changing needs. The huge numbers of communicating devices will provide and enable multiple dynamic networks at any one location. Users and their autonomous agents will be able to traverse these networks, passing seamlessly from one to another, coexisting in many at a single point in time, thus creating a truely ubiquitous intelligent computing environment.

In the future pervasive networking technologies will become commonplace within society and central to everyday life. Companies, organisations and individuals will increasingly depend on electronic means to store and exchange information in order to take advantage of ambient intelligence. Inevitably, many of these information transactions will be sensitive and critical.

D. Hutter et al. (Eds.): Security in Pervasive Computing 2003, LNCS 2802, pp. 116–129, 2004.

Appetite for Security. However, even in todays society information security is not taken as seriously as it should. There have been many 'sniffing' expeditions reported, aimed at locating and assessing the defences of wireless LAN networks[1]. Users frequently fail to initiate any form of security or information protection (not even well known encryption techniques). Citing such evidence, some security researchers have commented that it is not worth expecting users to care about security in the ambient intelligence world, since current attitudes are so sloppy.

But clearly people do care about both physical security (people protect their cars using locks) and information security (people protect their credit card numbers). There is building evidence that people are also becoming increasingly concerned about securing their privacy. A recent series by the U.K. newspaper The Guardian, 'Big Brother: Someone somewhere is watching you' [2], highlights the growing public debate surrounding personal privacy. One article details the results of a poll, conducted by ICM research, designed to measure people's attitudes to their privacy in an increasingly digital age. The results include the following: 58% of people in the U.K. don't trust the government to protect their privacy, 66% of people are worried about the security of their personal information travelling on the Internet, 72% of people would swap some functionality for security. Perhaps as computing becomes ubquitous, and pervasive technologies become as common as motor cars, then security and privacy will grow in importance in people's minds.

1.2 Security Requires Authentication

Security is commonly divided into four categories:

- *authenticity:* that a claim (especially of an identity) is valid
- *confidentiality:* that secrets are only shared between authorised principals
- *integrity:* that data cannot be altered in an unauthorised way, and
- *availability:* that secrets are always eventually made available to authorised principals.

To achieve security we must be able to ensure that we can correctly identify the authorised principals. Underpinning this is our ability to confirm (i.e. authenticate) that claims for authorisation are correct.

A fundamental building block of secure systems is sound key management, by which we mean the secure and correct generation, distribution, storage, use and revocation of cryptographic variables. As the wireless research community gains momentum [3, 4], key management for the wireless world is seen as an important research challenge. But key management itself depends critically on authentication. Without sound authentication, sound key management is infeasible.

In short, authentication is a basic building block of security. And for this reason this paper presents our thoughts on what authentication will mean in the world of ambient intelligence.

[1] For example "Winning The New Wireless War", in [1].

1.3 Some Conventions

The range of types of device that will interact in the world of ubiquitous computing is very large (from laptops to pacemakers). So to simplify our terminology we'll use the term 'PDA' to refer to arbitrary devices with the wireless communications capability of a modern mobile phone and the computing resources of a modern laptop[2]. Our PDAs are each assumed to have a unique user/owner, though one user may certainly have several PDAs.

The problem of ensuring that a PDA is in use only by its rightful owner will be elided in the following discussion. This is a major simplification, but it does make it easier to think about authentication requirements and foster debate, which is our aim[3]. Let us also note that in the world of ubiquitous computing, devices (and agents) may act with significant autonomy from their human masters, and this partly justifies the simplification.

When some people are involved in secure multi-party transactions, then we'll call the collection of their PDAs the legitimate PDAs. Of course, legitimacy does not imply 'trustworthiness'; it only refers to the PDAs owned by the people taking part in the secure multi-party transaction. If a device is not legitimate, then it is (naturally) illegitimate.

1.4 Plan of the Paper

Section 2 provides a brief overview of traditional notions of authentication, setting the context for their subsequent deconstruction and revision. Section 3 provides two "case studies" of pervasive computing security, which highlight why traditional notions of authentication are inadequate (and indeed inappropriate). This is followed by our revision of authentication in, Section 4. The paper is primarily about requirements, but the penultimate section presents a few thoughts about how future notions of authentication may be implemented, and the security issues that we see as arising there.

2 A Glance at Traditional Authentication

Authentication concerns proving, to a verifier, the validity of a claim. Our principal interest is in entity authentication, which concerns proving the validity of a claimed identity. We concentrate on entity authentication because it is the basis of sound key agreement and authorisation. When we talk subsequently about "traditional" authentication we simply mean entity authentication.

The basic idea of entity authentication is deceptively simple: one principal wants to be sure that it is talking to (and only to) whomever it intends. However, the subject of formalising mutual entity authentication, where two or more principals authenticate themselves to each other, is notoriously difficult. The depth and difficulty come from

[2] Quite compact devices will have such capabilities in a few years, and devices may well have much greater capabilities, further in the future. However, a modern mobile phone and laptop form a sufficient basis for our discussion.

[3] Solutions such as those of Corner, [5], can address the device-to-user authentication problem.

the distributed, multi-party nature of mutual entity authentication. Typically, the requirement is that the beliefs, actions and actual achievements of all principals match [6, 7] despite malicious activity.

An identity is usually recognised by confirming that some known attribute of the subject is present. Humans have exceptionally powerful capabilities to recognise images and sounds, but in the electronic world identities are recognised by matching digital information. A representative, but not exhaustive, list of electronic means for entity authentication includes:

- shared secrets, including passwords,
- public key cryptography schemes, including both Public Key Infrastructures and Pretty Good Privacy[8],
- tokenisation,
- biometrics.

While these schemes appear quite diverse, for our discussion we can note that they share some basic assumptions and features. These will be the basis for our arguments against entity authentication, so we summarise them here.

Firstly, while authentication is a cornerstone of security, it does not provide any security on its own; rather, the security depends on *trust in the entity*[4]. Passwords or biometrics might correctly grant a user access to confidential data, but that data is secure only if the user is trustworthy. The same holds true for mutual entity authentication by crypto-protocols. I may have perfectly justified confidence that I have a keyed link with Alice, but I will only communicate sensitive data to Alice if I trust her.

Secondly, we noted identities are electronically confirmed by matching digital information. To implement entity authentication there must be a binding between a principal, or more precisely the identity of a principal and some recognising information (a password, PIN, and so forth), and there must be assurance that this binding is correct. An immediate consequence of this fact is that the recognising digital information and the binding has to be initialised, stored and managed securely. In general, any entity authentication scheme requires significant pre-existent trusted knowledge and infrastructure[5]. For instance, to use a password, or shared secret, all agents (human and electronic) must secretly agree the password, and of course be confident of whom they share it with. The challenges of maintaining and managing such information are great. The sometimes rickety services provided by PKIs[6] are now perhaps the most widely discussed example of the difficulties of managing the trusted information for entity authentication.

Finally, an obvious feature of entity authentication, is the essentially static and binary nature of the assurances it delivers. Clearly, a principal's claim to an identity is either true or false, a password either matches or it does not, and so forth. But this logical feature of entity authentication constrains the ways it can be used. If there is a need to grade levels of assurnace, then the facilities to do that (setting the minimum size of a password, for example) are rather limited.

[4] Thanks to Peter Ryan and Dieter Gollman for emphasising this point.

[5] Colin Boyd has a theorem, in [9], that entity authentication is impossible without pre-existent shared secrets. Maurer discusses and develops these ideas in [10].

[6] For instance, in March 2001, Microsoft announced that 2 digital certificates that were issued in it's name, by a highly trusted third party, were false.

3 What's Wrong with Traditional Authentication?

We will argue that for the pervasive world the current focus on authenticating identities will be misguided. The failings of entity authentication are partly because what it needs to assume will not be available, and partly because the assurances achieved by entity authentication will be of diminishing value.

The best way to elucidate our arguments is by some examples. The very nature of pervasive computing means that there are a lot of scenarios, but we will focus on two: using a public printer from a wireless device and a collection of PDAs bootstrapping a secure network. Both these examples have precedents in the literature. We hope to foster more active debate on the fundamentals of key management and authentication by focusing on these existing examples.

3.1 Using Public Printers, via a Wireless Link

For the first example imagine a user in a public place, like an airport, with a PDA. Suppose the PDA contains confidential data which the user wants to print out, and assume that the airport has a number of printers that can potentially service the user's needs. Ideally, the user wants to use a wireless link to send the confidential data securely to a chosen printer[7].

Balfanz *et al.*'s paper [11] is centred on this problem. They briefly discuss security aims (where we intend to dwell) and then they concentrate on a solution. Their proposal to secure the PDA to printer wireless link is based on the 'Resurrecting Duckling' of Stajano, [12]. Essentially, a keyed link is created by physical contact between the device and the printer. No pre-existent authentication mechanisms, like certificates, are needed; but securing the link does require users to touch their chosen printer with their PDA, on an appropriate physical interface. As such Balfanz *et al.*'s solution does 'bootstrap' a degree of security.

Nevertheless, the problem bears re-examination to understand better the security requirements, not least because it is a good example of wireless access to public utilities. In this example the security assurances that a user is likely to want are:

1. The confidential data on the user's PDA goes to the specific printer chosen by the user and to no other devices.
2. The printer treats the confidential data in a 'trustworthy' manner. Where trustworthy captures properties like guaranteeing that no other party has access to the data while it is resident on the printer, and (most likely) that the confidential data is deleted immediately after being printed.

Can traditional entity authentication meet these two requirements? For the first requirement the user might use entity authentication to ensure that the data only goes to the chosen printer. But this would mean that the user would need to have the printer's public key. And the user can only get the printer's public key after reliably determining

[7] Whilst this example is not particularly futuristic, the basic service model and security requirements will remain relevant in future pervasive environments.

the printer's name and then accessing a certification authority that serves the printer and whose certificates the user's PDA can recognise.

Reliable name resolution and access to useful certification information are both major assumptions about the world, which, as Balfanz et al. point out, are not likely to hold in the ubiquitous computing future. Obtaining the printer's name reliably will at best be impractical: will user's have to type in IP addresses of utilities into their PDA before using them? And it will be difficult to ensure the integrity of a claimed name of an arbitrary device. Even if the name could be determined reliably and easily, finding a useful public key for an arbitrary device, in an unspecified location, would require that every single device in the world is served by a small collection of PKIs. Furthermore, with wireless communications it may not be feasible to assume that a certification authority will be accessible.

Thus, for the first security requirement – namely that confidential data is received by the chosen device and no other – it appears that the *outcome* of traditional authentication could serve, but that it assumes things (viz. name resolution and certification) that will be increasingly difficult to provide. What about the second requirement, of ensuring that the printer treats the data in a trustworthy manner? We noted, in section 2, that for entity authentication to give security we need to trust the entity. In the pervasive domain, simply having a keyed link to an effectively arbitrary named device will give us no assurance about that device's trustworthiness. In effect the user still has to trust a random printer in an unknown place. Thus, entity authentication appears to be of very little value in delivering assurance about how the printer will behave. In fact, this type of assurance seems to fall outside the ambit of entity authentication.

Finally, it should be noted that a user may have varying degrees of concern for the information that needs to be sent to the printer. We noted the essentially binary nature of entity authentication, but a more useful capability will enable a user to grade the assurance provided.

The above reasons, for traditional authentication having limited value in the case of the PDA and printer, hold true more generally. In a pervasive computing environment these failings will become more acute.

3.2 Mesh Networks

For our second example imagine a set of people meeting in some place and wanting to work together securely. Of course, they will have their PDAs and so they will want these PDAs to form a 'secure' network. The users' PDAs are automatically legitimate (according to our original definition), but it is quite conceivable that the users will want to network securely with other devices in their vicinity, and these nominated devices must also be treated as legitimate. We'll take secure to mean that it is infeasible for any illegitimate device to decipher communication between the legitimate PDAs and peripherals. It may be possible to assume that the users' PDAs will be 'trustworthy'. However, when the users want to use other devices then their trustworthiness will probably need to be validated.

The problem is: how do the legitimate PDAs and other legitimate devices form a secure network (in the sense just mentioned) with minimal pre-existent trusted knowledge (such as valid certificates)? When the users are in an arbitrary place and want to

use various devices from their environment, then this problem generalises the previous problem - of secure use of a public printer.

In the literature the simplest form of this problem is discussed by Asokan and Ginzboorg [13]. They discuss the problem of users with PDAs, in a closed meeting room, wanting to create a secure network across their PDAs. Their solution is based on protocols that use weak encryption to agree a strong encryption key. The basic idea is that the users agree a password and then they type that password into their PDAs. The legitimate PDAs are by definition those that have had the password input. This password makes the weak encryption key, which is, nevertheless, sufficient for the legitimate PDAs to agree a strong key - using the protocols that they present. Thus they provide a solution for bootstrapping security that requires no pre-existent electonic trust. In effect they have manual initialisation of trust, since the users are (implicitly) responsible for controlling the exposure of the password. It is also clear that their solution can be extended to include any device with a keyboard.

Asokan and Ginzboorg preface their solution with a brief discussion of the security requirement. They note that maintaining confidentiality with respect to identities, which can be achieved by authenticated key agreement, is not what is required here. Instead they propose that the security requirement (they use the term "prior context") is defined by location. To quote:

Only people present in this meeting room can read the messages that I send.

In other words, the legitimate PDAs are only those owned by people in the meeting room and only these should be able to decrypt the messages.

In this problem, trust stems from the fact that people in the room can see each other, and already know that they wish to share secrets. Security, in this location-centric context, is still predicated on trust in the PDAs and, in the more general case, in the nominated peripherals. However, imagine the room containing people who are less trusted than the others (perhaps some people know each other, but some are strangers). Whilst it remains true that principals only wish to share their secrets with principals in the room, they may also wish to authenticate the strangers. In this case the strangers will need to provide credentials, about themselves and their devices, to admit them to the secure multi-party session. Trusted Third Parties may be the means for obtaining added assurance, though what constitutes an acceptable credential is likely to vary.

To summarise the light this example sheds on entity authentication, trust is based largely on location and human contact, not on identity. Effective solutions to this example should be able to translate human trust into electronic trust quickly and easily (this is what the proposed solution, by Asokan and Ginzboorg does). If assurance is required concerning the behaviour of peripherals, then this example simply iterates the failing of entity authentication from the previous example. Finally, the inflexible binary assurance of entity authentication does not match the spectrum of assurances (and means of assurance) that may be needed in the future.

4 Revising Authentication for the Pervasive Domain

4.1 What Should We Authenticate?

The examples in the previous section tried to describe why, in the world of ubiquitous computing, entity authentication will neither be easy to achieve nor give the desired

assurances. It will be impractical because device names will usually be indeterminate and it is unlikely that infrastructures to support certifying names of the huge numbers of devices will be available. Furthermore, entity authentication will not give the desired assurances, because critical to entity authentication is trust in the entity. When desiring secure interaction with an unknown entity, assurance of its identity is far less relevant than assurance that it is 'trustworthy', where the interpretation of trustworthy varies considerably according to the application.

Reflecting on the examples from Section 3 does yield some clues about how we might revise traditional notions of authentication. Firstly, a name is an attribute of an object, but only one of many. Entity authentication is predicated on the belief that the name is sufficient to infer the appropriate property of trustworthiness. In the world of ubiquitous computing few trustworthiness properties will follow simply from the name of an object.

But what about other attributes of objects? In both the examples in Section 3 it is clear that location is an important attribute. In the mesh network example physical location was fundamental to specifying the legitimate PDAs and devices, and indeed for the decision to trust the people taking part. Also, when the users want to connect to various peripherals in their locality, they might confirm the legitimacy of the choosen devices by seeking assurance of the type of device. Thus a printer might be asked to prove that it is in a specific location and that it is a printer. By authenticating these sorts of attributes we might have justifiable confidence in legitimacy.

But as we have noted, legitimacy does not necessarily imply any useful property of trustworthiness; usually it gives little assurance about what the device will do. People make decisions about who or what to trust based on experience, on various conventions (like referal) and even on 'instinct'. We can transfer the same trust model into the digital domain. I trust my computer to work as I expect because I have bought it from a trusted vendor or manufacturer. The software on board is from a trusted software developer. Continuing the transfer, if a user can authenticate the printer's manufacturer and the user trusts the manufacturer, then that might be a basis to trust the printer. Thus the attribute of manufacturer may also be important.

Certificates may be a viable means to prove a reliable manufacturer to a user. But it should be noted that even if it is decided to use this as a basis for trust it may be necessary to confirm that the printer retains its original dependability despite being in a public place for a protracted period. Thus there may be a need to exhibit trusted maintenence and tamper freedom.

Tamper freedom is part of the 'state' of a device (and the state of a device is also one of its attributes) and aspects of state may imply appropriate trustworthiness properties. Another important aspect of state might be that the device is not running another concurrent session, with someone else.

In general, by authenticating various attributes we would aim to confirm precisely which devices are legitimate. Having sufficient evidence of legitimacy we would then need a basis to trust what those devices are doing, or will do.

4.2 How Much Should We Authenticate?

Authenticating different attributes of an entity will give varying levels of assurance. These assurance levels will fluctuate depending upon the environment. For example, you can be more sure when you are in an environment where you can confirm the results of an authentication attempt. Consider the printer example described above. Verifying the location of the printer using GPS will be a much stronger form of authentication when you can see it and you know the coordindates of yor own location, because you can then make a judgement about your distance apart.

Clearly, there are many different levels of assurance possible. By authenticating sets of attributes (hereafter referred to as authentication sets) we are likely to gain more assurance than by authenticating only one. But assurance usually brings a cost. This cost may be in terms of financial price of implementing a particular authentication (high assurance could cost more to engineer), or the cost may be the time it takes a user to achieve assurance (this is likely to be a major factor). In some cases there may be no price too high, in others speed may be of the essence, or budget. Other sensible decision criteria could also be applied.

While it is clear that some sort of context aware, dynamic security policy will be required in the pervasive paradigm, it is not so clear how to define and implement one. We first need to devise metrics for comparing authentication sets, and find decision criteria reflecting our priorities (for instance, cost in time or money).

Consider Figure 1 below:

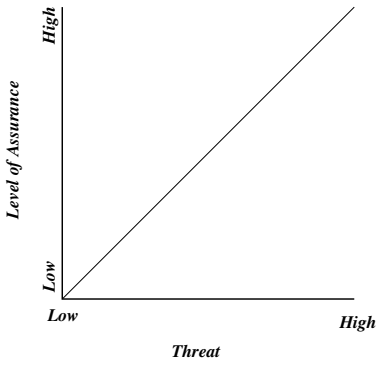

Fig. 1. A notional optimal level of assurance

This graph represents a notional optimal level of assurance, with respect to environmental threat. It is 'notional' in that it is proposed only as a starting point to discuss requirements. It is 'optimal' in the sense that the line defines the level of assurance that is just sufficient. Assuming that cost is related to assurance, the line captures the minimum costs that is likely to be incurred.

The horizontal axis represents a function of the threats that we want to guard against. In pervasive computing threats will vary from juvenile hacking, to corporate espionage, right up to cyber terrorism and government surveillance. The vertical axis represents

the level of assurance that is required, this will be determined by a number of factors including:

- the criticality of the security service that we want to protect,
- the type of association that is being made.

This last will vary from a transient link (as is the case with a public printer), to on-going associations (with devices in the home or office), to long term or lifetime associations (for instance, a pacemaker). A further dimension to the type of association is the number of principals involved; group key management is significantly more complex than the two party case, and we can similarly expect multiparty associations to exacerbate the complexity of the required levels of assurance.

When developing or evaluating a security technology, or a security policy, all that really matters is that you possess at least as much assurance as is required. This simply means that the level of assurance achieved must be above the optimal line. In our naive representation this is just the shaded region in Figure 2.

Note that Stajano's work, on the 'Resurrecting Duckling' [12], is an avowedly low assurance solution, designed for low cost applications. As such, an evaluation of the Resurrecting Duckling should find it on the bottom left hand corner of the shaded region, on the graph in Figure 2. Given that researchers are already investigating low assurance solutions, it seems that an understanding of gradeable levels of assurance is necessary.

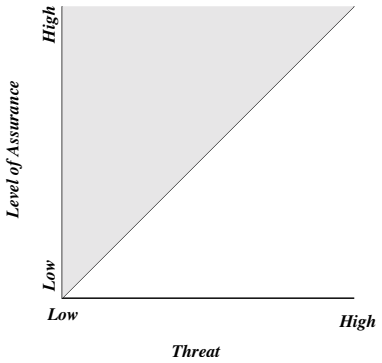

Fig. 2. The region of safe assurance levels

In summary to enable the flexibility that will be required the research community should aim to:

1. Establish metrics for comparing and quantifying the assurance levels gained from differing authentication sets.
2. Understand better what the optimal graph actually looks like.
3. Understand better the cost drivers and their impact, for each context.

A sound understanding of these subjects will form a basis for the flexible and dynamic security policies that the ambient intelligence world will need.

5 Thoughts on Future Implementations

The primary aim of this paper is to re-examine traditional notions of authentication and to suggest alternatives. Nevertheless, it is also worth some noting issues and problems regarding implementation.

5.1 Using Certificates

We noted above, in Section 4, that if it were possible to certify attributes such as the manufacturer of a device, then this may, in appropriate circumstances, provide assurances about how that device will behave. More generally, while the examples discussed above point away from entity authentication, that does not imply abandoning entity authentication. For instance, in the example of mesh networks the security requirement is defined with respect to locality, but there may be a need for the local mesh to connect securely to an infrastructure. Alternatively, the degree of trust that users have in the local group may vary and some users may need to supply additional credentials of trust, like certificates. Thus, when considering implementation attention first needs to be paid to the use of traditional authentication mechanisms in the pervasive domain.

Debate about the practicality of PKIs is an on-going subject. Questions of scalability are paramount: will PKIs simply not scale, or will it be possible to address scalability by making PKIs work together [14]? The debate should be extended to the use of certificates, and the value they add, in the pervasive domain. Wireless communications are fundamental to pervasive computing, and that means variable connectivity, so what use is a Certification Authority, if it is not always accessible? What trust will certificates carry, when their timely revocation will be even more problematic? How will certificates be used? Will certificates have to be issued per device, for its lifetime? It may be the case that some attributes can only be authenticated by certificates (static ones like manufacturer spring to mind). These questions must be explored.

5.2 Agent Based Solutions

Are there more direct ways of achieving confidence in what a device will do, other than relying on its manufacturer (and perhaps evidence of tamper freedom)? The mobile code community has invested effort into studying techniques for self-certifying code, where a software agent carries evidence that it will behave in a trustworthy manner [15]. Security for pervasive computing could use similar concepts (although it isn't clear how much will simply 'port'). For instance, hardware could be configured to send a hash of its configuration to devices it wishes to communicate with. Then policies may be implemented where devices will only interact with other self-certifying devices, whose hash is acceptable. If the assurance at the hardware level was sufficiently high, then this could be used to mitigate the assurance demanded at the network level.

Continuing to draw inspiration from the mobile code community, user friendly security could be enforced by having a mobile agent act on the user's behalf. A user's agent might certify that other devices are fit for interaction, in the senses that we have discussed. Such approaches would have the obvious drawback of needing to ensure that the agent runs correctly on the correct device. It may be observed that the 'which' and

the 'what', that we mentioned in Section 4.1 have re-appeared, in this new context. Nevertheless, an agent approach may broaden the range of techniques that can be deployed, as well as yielding user friendly solutions.

5.3 Man-in-the-Middle Vulnerabilities

We have argued forcefully for a paradigm shift from authenticating names to a much broader and flexible notion of attribute authentication. However, it is worth giving thought to whether such a change of orientation will create fresh security vulnerabilities.

Our main concern has been that when a connection is established by something other than name there is a greater than usual danger of man-in-the-middle attacks. Here an attacker sits "in the middle" of a channel between two agents, passing the information on, but not giving any evidence that he is there. For instance, in the example of using the printer (from Section 3.1), if a PDA's wireless link to the printer, P , has a device-in-the-middle, D_M, then the PDA might be communicating with D_M, instead of P . To mask the interception D_M would still forward the user's messages onto P, and indeed pass P 's response back to the PDA. If the PDA doesn't know P's identity, then it has no way of knowing that it is talking to D_M instead of P. In the case of the mesh networks it is possible for a set of n nodes to suffer a 'men-in-the-middle' attack, where each of the n nodes ends up connected to an attacking network of $n - 1$ nodes.

Entity authentication precludes man-in-the-middle attacks because each agent has proof of the identity of his interlocutor. If we pronounce entity authentication obsolete or impractical, then we need to think about the added risk of man-in-the-middle attacks, and how these risks may be averted.

An inevitable feature of a man-in-the-middle attack is that it adds a hop to each communication, and this gives a clue to avoidance. If we can introduce some feature in the protocol in which there is an inevitable loss of some resource or entropy from messages as they are passed around, then this would be a basis for avoidance. This could be the passage of time (for instance ensuring each authentication takes some measurable time in a way that is noticeable in the particular circumstance) or some use of watermarking or cryptographic hashing. Another might be each participant announcing in some way a sufficient piece of information about the agent or set of agents to which it is connected. So, in the case of the printer, we might get the printer to print out a banner page with the serial number of the device with which it is operating the protocol. Provided that the man-in-the-middle is using a public key certificate other than that of the originator, it will not be the man-in-the-middle's device number - assuming the protocol is properly designed). In the case of the PDAs, each node might be told a hash of the serial numbers of all the nodes in the network: they might then check that these all agree.

A further possibility is for some unambiguous description of a printer, say, to be conveyed by the protocol to the potential user. This description might be the printer's position, as certified by some especially precise form of GPS or some signed description by an authority the user trusts.

Note that all of the methods assume some way of passing information between nodes other than through the wireless network. The given examples are: reading a printout, listening to one's colleagues at a meeting, watching timed behaviour or even look-

ing where a printer is (or, indeed, checking the integrity of its tamperresistant seal). In the work of Balfanz, et al. [11], physical contact is the means for a non-wireless exchange of information, and indeed this is another possibility. Given the breadth of the problem domain it seems impossible to state fixed methods. Nevertheless, the avoidance of man-in-the-middle (and indeed men-in- the-middle) attacks should be an important item on the future research agenda.

6 Conclusions

Enabling security will be critical to realising the exciting future of Ambient Intelligence. But sound key management will be critical to securing transactions in the world of ubiquitous computing. In the wireless security research community key management is seen as one of the fundamental problems and an area of active research [3, 4]. But key management itself depends fundamentally on sound authentication. Based on the past experiences of the security community (particularly in the area of key management) we claim that understanding and implementing authentication are among the most important challenges facing pervasive computing security.

Traditional authentication has concentrated on the notion of entity authentication, which provides assurance of who is the subject of a secure interaction. We have argued that the requirements to implement entity authentication are unlikely to be practical in the pervasive domain. Further, we have argued that the assurances delivered by entity authentication will be of limited value.

Instead of entity authentication we have argued that assurances are required of which devices are the subject of interaction and what those devices will do. These assurances may be achieved by 'authenticating' (or providing tamperresilient confirmation of) a much broader class of device attributes than name. For the pervasive domain it is clear that location is an important attribute, but many more attributes are likely to be required, including origin, aspects of current state, retention of original integrity, and more. Making this shift is likely to introduce many new challenges and create new security vulnerabilities, the increased likelihood of man-in-the-middle attacks being an important example. It is also clear that the requirements of which attributes to authenticate will vary from context to context.

A further concern is the rigid, binary nature of entity authentication. The much broader range of interaction will need more flexible security policies with richer gradations of assurance. Deciding upon the appropriate security policy will be crucial, and devising metrics to facilitate such a decision will be an important research topic.

The subject of security for ad-hoc and wireless networks is relatively new but the area is growing very fast [3], with key management being one of the major challenges. However, much of the research in this area focuses on engineering traditional approaches, based on certification (for example, [16]) or tokenisation (for example, [5]) to solve specific problems within the domain. Above we have discussed in detail the work of Balfanz et al. [11] and Asokan and Ginzboorg [13]. Both these papers concentrate on engineering solutions. Thus far we have seen little work that thinks specifically of how authentication needs to be deconstructed and revised. We hope it is clear that the concepts we have discussed underpin the on-going solution-oriented research. More im-

portantly, we hope that this work will help to foster debate on the new security concepts for the Ambient Intelligence World.

Acknowledgements

The authors would like to thank give special thanks to Gavin Lowe for stimulating discussions, as well as to Peter Ryan, Dieter Gollman, Colin Boyd, Colin OHalloran and Nick Moffat.

References

1. *Information Security Management.* June 2002, published by Penton.
2. *The Guardian Newspaper.* 7th Spetember, 2002
3. http://www.crhc.uiuc.edu/ nhv/wise/
4. http://www.pampas.eu.org/
5. Corner, M. D. and B. D. Noble.Zero-interaction authentication. The *8th ACM Conference on Mobile Computing and Networking*, September 2002, Atlanta, GA.
6. Diffie, W., P.C.van Oorschot and M.J.Wiener, Authentication and Authenticated Key Exchange.Design, *Codes and Cryptography*, 2 (1992), pp 107-125.
7. Roscoe, A.W. Intensional Specifications of Security Protocols. *Proceedings of the 1996 IEEE Computer Security Foundations Workshop*, IEEE Computer Society Press, 1996.
8. Zimmerman, P. *The Official PGP Users Guide*. The MIT Press, 1995.
9. Boyd, C. Security Architectures Using Formal Methods. *IEEE Journal on Selected Areas in Communications*, Vol. 11, No. 5, 1993, pp. 694-701.
10. Maurer, Ueli and Pierre Schmid. A Calculus for Security Bootstrapping in Distributed Systems. *Journal of Computer Security*, vol. 4, no. 1, pp. 55-80, 1996.
11. Balfanz, Dirk, D. K. Smetters, P. Stewart and H. Chi Wong. Trusting Strangers: Authentication in Ad-hoc Wireless Networks. *Network and Distributed Systems Security Symposium*, 2002. Available from: http://www.isoc.org/isoc/conferences/ndss/02/proceedings/index.html
12. Stajano, F. and R. J. Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. 7th Security Protocols Workshop, LNCS vol. 1796, Cambridge, UK.
13. Asokan, N. and P. Ginzboorg. Key Agreement in Ad-hoc Networks. Computer Communication Review, 2000. Available from: http://www.semper.org/sirene/people/asokan/research/index.html
14. http://www.dti-mi.org.uk/newweb/fiducia.htm
15. Vigna, G. Mobile Agents and Security. LNCS, July 1998.
16. Kong, J., P. Zerfos, H. Luo, S. Lu, L. Zhang. Providing Robust and Ubiquitous Security Support for Mobile Ad-hoc Networks. Proceedings of 9th International Conference on Network Protocols. IEEE Computer Society Press, 2001.