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
WWW home page: http://www.formal.demon.co.uk
[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 important and challenging. 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 challenges 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 foresees communicating and computational devices embedded in all parts of our environment, from our physical selves, to our homes, offices, streets and so forth. Humans will be surrounded by intelligent, 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 the users changing needs. The prolific amount 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, creating a truely ubiquitous intelligent computing environment.

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

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, focused on locating and assessing the defences of wireless LAN networks[1]. Reports include numerous instances of either insecure configurations, or simply no attempt to initiate any form of information protection (including using well known techniques such as encryption). The authors have heard security researchers citing such evidence to conclude 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 security, both physical (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 Guardian newspaper '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 peoples attitudes to their privacy in an increasingly digital age. The results include the following: 58% of people 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 as common as motor cars, then security and privacy will grow in importance in the minds of the user.

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 principles,
- integrity, that those secrets cannot be altered in an unauthorised way, and
- availability, that secrets are always eventually made available to authorised principles.

To acheive security we must be able to ensure that we can correctly identify the *authorised* principles. 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 key 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.

Thus 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, without loss of generality, to devices with the wireless communications capability of a modern mobile phone and the computing resources of a modern laptop (quite compact devices will have such capabilities in a few years). Our PDAs are each assumed to have a unique user/owner, though one user may certainly have sevaral 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. Furthermore, solutions such as those of Corner, [5], can address the device-to-user authentication problem. Let us also note that in the world of ubiquitous computing, devices (and agents) may act with significant autonomy from their original masters.

When a set of users are involved in secure multi-party transactions, then we'll call the collection of user's PDAs the *legitimate* PDAs. Of course, legitimacy does not imply 'honesty'; it only refers to those PDAs that are owned by users who are themselves recognised to partake in the secure multi-party transaction. If a device is not legitimate, then it is (naturally) *illegitimate*.

1.4 Plan of the Paper

The rest of this paper starts with an overview of traditional notions of authentication and a critique of their suitability for securing ambient environments. We then consider how traditional notions need to be revised, and what the significant challenges to successful implementation are likely to be.

2 A Glance at Traditional Authentication

Authentication concerns proving, to a verifier, the validity of a claim. Our principal interest is in *entity* authentication, and with its deconstruction and revision, because entity authentication is the basis of key agreement, which is in turn the basis of securing interactions. When we talk about "traditional" authentication we simply mean entity authentication.

The basic idea of entity authentication is deceptively simple: you (and your interlocutors) want to make sure that you (and they) are talking to (and only to) the principal(s) that you think you are. And yet there are almost as many different definitions as there have been researchers studying the field. The subject of formalising mutual entity authentication is notoriously difficult. It is non-trivial to capture what is meant by "talking to" and "think you are", but the real depth comes from the distributed, multi-party nature of mutual entity authentication. Typically, the requirement is that the beliefs, actions and actual achievements of the parties match [6], [7] - despite malicious activity.

The objective of entity authentication is, clearly, to prove a claimed identity. Humans have exceptionally powerful capabilities to recognise images and sounds, but the electronic world works rather differently. A representative, but not exhaustive, list of electronic means for entity authentication includes:

- shared secrets, including passwords,
- public key cryptography schemes, including both PKIs and PGP,
- 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.

Authentication is, of course, a cornerstone of security. Nevertheless, entity authentication does not provide any security on its own; rather the security depends on *trust in the entity*.[2] 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, using any of the above schemes requires significant pre-existent trusted knowledge. Entity authentication requires a binding between a principal, or more precisely *the identity of a principal* and some information, and there must be assurance that this binding is correct. 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 who they share it with. The challenges of maintaining and managing this information are great. The sometimes rickety services provided by PKIs[3] is now perhaps the most widely discussed example of the difficulties of maintaining and 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 also constrains the type of assurances it delivers and the ways that it can be used. If there is a need to grade the levels of assurance required, then only oppurtunity to enforce varying grades of assurance is when the bindings between the principals and their trusted identifier is initialised.

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

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

3 Revising Authentication for the Pervasive Domain

3.1 What's Wrong with Traditional Authentication?

We will argue that in the pervasive world the focus on authenticating identities will be ever more misguided. Partly, because entity authentication needs to assume things that will increasingly improbable, and partly because the assurances achieved by entity authentication will be of diminishing value.

Some examples are probably the best way to elucidate our argument. 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 collection of PDAs bootstrapping a secure network. Both these examples have precedents in the literature, and we hope to foster more active debate on the fundamentals of key management and authentication by focusing on existing examples.

Using Public Printers, via a Wireless Link. For the first example imagine a user in a public place, like an airport, with a PDA. The PDA contains confidential data which the user wants to print out, we'll assume that the airport has a number of printers that can potentially service the users needs. Ideally the user wants to use a wireless link to send the confidential data securely to a chosen printer.[4]

Balfanz *et al.*'s paper [8] 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 printer to PDA wireless link is based on the 'Resurrecting Duckling' of Stajano, [9]. 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 only to the specific printer choosen 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 of the above the user might use entity authentication to ensure that the data only goes to the specific choosen printer. But this would mean that the user

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

would need the printer's public key. And the user can only get the printer's public key after reliably determining the printer's name and then accessing a PKI 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 ubquitous computing future. Obtaining the printer's name reliably will at least 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. Then even if the name could be reliably and easily determined, finding a useful public key for an arbitrary device, in an unspecified location, is equivalent to requiring 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 only 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 unlikely. What about the second requirement, of ensuring that printer treats the data in a trustworthy manner? We noted (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 even less 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 reasons traditional authentication is of limited value in the case of the PDA and printer holds true more generally. In a pervasive computing environment these problems will become more acute.

Mesh Networks. For our second example imagine a set of people meeting 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 (like 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 [10]. 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-existing 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, as would 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 this meeting room and only these should be able to decrypt messages.

In this problem trust starts 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 strangers 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. TTPs maybe the means for them providing added assurance, but 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. 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.

3.2 What Should We Authenticate?

The previous examples tried to describe why, in the world of ubiquitous computing, entity authentication will not be easy to achieve nor will it necessarily 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 googleplex numbers of devices . 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, the 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 3.1 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 was 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 3.1 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. Again for the mesh network, 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 a printer and it is in a specific location. By authenticating these sorts of attributes we might (justifiably) have confidence in legitimacy.

But as we have noted legitimacy does not necessarily imply any 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, various conventions (like referal) and even 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*.

3.3 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 actually see it, because you will be able to check your own location, and make a judgement as to the distance you are apart.

Clearly, there are many different levels of assurance possible. By authenticating sets of attributes (here after referred to as *authentication sets*) we are likely to gain more assurance than by authenticating only one. But assurance brings with it 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 in 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. And other sensible decision criteria could also be applied.

Whilst 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 a metric for comparing authentication sets, and find decision criteria reflecting our priorities (cost in time or money). Consider Figure 1 below:

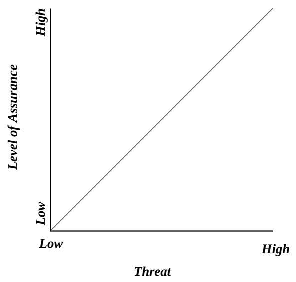

Fig. 1. *The* **optimal** *level of assurance*

This graph represents the 'optimal' level of assurance that should be used, with respects to the environmental threat. It is optimal because the line defines the level of assurance that is just sufficient. It captures the lowest cost acceptable level of assurance. The x-axis represents the threats that we want to guard against and the criticality of the security service that we want to protect. The threats will vary from juvenile hacking, to corporate espionage, right up to cyber terrorism and government surveillance. The y-axis will be the type of association that is being made, this 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). Just as group key management is significantly more complex than the two party case, multiparty associations will exacerbate the complexity of security requirements. When there is no threat there is no requirement for assurance. When the threat is at its maximum, then correspondingly you require maximum levels of assurance. In the ideal world maximum levels of assurance would come with no penalty. This would then make such high assurance useable under any threat, and result in the best scenario which is represented by the dashed line in Figure 2.

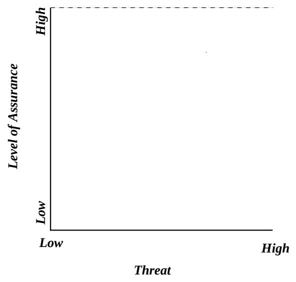

Fig. 2. *The* **best** *level of assurance.*

Actually, all that really matters is that you possess at least as much assurance as is required. Which means that a policy which dictates levels of assurance above the *optimal* will be sufficient. If we use our naive representation of *optimal* then this means policies in the shaded region in Figure 3.

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.

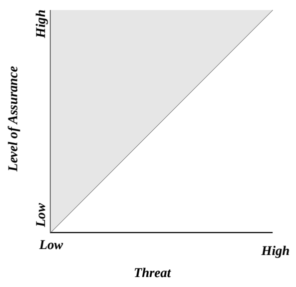

Fig. 3. *The region of safe assurance levels.*

4 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 noting issues and problems regarding implementation.

4.1 Using Certificates

We noted above 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, *e.g.* certificates. Thus in considering implementation attention first needs to be paid to the use of traditional authentication mechanisms in the pervasive domain.

Debate about the practicality of PKI's is an on-going subject. Questions of scalability are paramount, will PKI's simply not scale, or will it be possible to address scalability by making PKI's work together [11]? 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? In this case what trust will certificates carry, since 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), thus these questions will have to be understood.

4.2 Agent Based Solutions

Are there more direct ways of achieving confidence in what a device will do, than relying on its manufacturer (and perhaps evidence of tamper freedom)? The mobile code community has studied extensive techniques for *self-certifying* code, where a software agent carries evidence of that it will behave in a trustworthy manner [12]. 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 users behalf. A user's agent might certify that other devices are fit for interaction, in the sorts of sense that we have discussed. Such approaches would have the obvious drawback of needing to ensure that the agent was running correctly on the correct device. It may be observed that the 'which' and the 'what', that we mentioned in 3.2 have simply been shifted, but an agent approach may broaden the range of techniques that can be deployed, as well as yielding user friendly solutions.

4.3 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 engender fresh security vulnerabilitites.

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 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 attacking network of $n - 1$ nodes.

Entity authentication precludes man-in-the-middle 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, 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 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 it is connected to. 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 (and provided that the man in the middle is using a public key certificate other than that of the originator, it will not be his if the protocol is properly designed). In the case of the PDA's each node might be told a hash of the serial numbers of all the nodes in the network: they might then check that these are all the same.

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

Note that all of the methods assume some way of information passing between nodes which is not through the wireless network. This is reading a print-out, listening to one's colleagues at a meeting, watching timed behaviour or even looking where a printer is (or, indeed, checking the integrity of its tamper-resistant seal). In the work of Balfanz, *et al.*, the 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 attacks should be an important item on the future research agenda.

5 Conclusions

Enabling security will be critical to realising the exciting future of Ambient Intelligence. But critical to securing transactions in the world of ubiquitous computing will be sound key management. In the wireless security [3], [4] research community key management is seen as one of the fundamental problems and an area of active research. 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'd claim understanding and implementing authentication is one of the first and most important challenges facing pervasive computing security.

Traditional authentication concentrated on the notion of entity authentication, and this provided assurance of *who* was the subject of a secure interaction. We have argued that the requirments to implement entity authentication are not likely to be practical in the pervasive domain. Further, we have argued that, for the most part, 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 assurance may be achieved by 'authenticating' (or providing tamper resilient 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 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 engender new security vulnerabilities, the increased likelihood of man-in-the-middle being an important example. It is also clear is 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.

With regards to related work, 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, [13]) or tokenisation (for example, [5]) to solve specific problems within the domain. Above we have discussed in detail the work of Balfanz *et al.* [8] and Asokan and Ginzboorg [10]. Both of these paper 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 that we have discussed underpin the on-going solution oriented work. More importantly we hope that this work will help to foster debate on the new security concepts for the Ambient Intelligence World.

6 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 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.* In 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.* In 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. Balfanz, Dirk, D. K. Smetters, P. Stewart and H. Chi Wong. *Trusting Strangers: Authentication in Ad-hoc Wireless Networks.* In Network and Distributed Systems Security Symposium, 2002.
   Available at: `http://www.isoc.org/isoc/conferences/ndss/02/proceedings/index.html`
9. Stajano, F. and R. J. Anderson. *The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks.* In 7th Security Protocols Workshop, LNCS vol. 1796, Cambridge, UK.
10. Asokan, N. and P. Ginzboorg. *Key Agreement in Ad-hoc Networks.* In Computer Communication Review, 2000. Available from:
    `http://www.semper.org/sirene/people/asokan/research/index.html`
11. `http://www.dti-mi.org.uk/newweb/fiducia.htm`
12. Vigna, G. *Mobile Agents and Security.* LNCS, July 1998.
13. Kong, J., P. Zerfos, H. Luo, S. Lu, L. Zhang. *Providing Robust and Ubiquitous Security Support for Mobile Ad-hoc Networks.* In Proceedings of 9th International Conference on Network Protocols. IEEE Computer Society Press, 2001.