Information Technology

German language title goes here

Mobile security and trusted computing

Professor Chris Mitchell: Royal Holloway, University of London, TW20 0EX Egham, Surrey, United Kingdom
Tel: +44-1784-443423, Fax: +44-1784-430766, E-Mail: c.mitchell@rhul.ac.uk

Keywords: security, mobility, ubiquitous computing, trusted computing

Schlagworte: German keywords go here

MS-ID: Heft: 46/3 (2004)	c.mitchell@rhul.ac.uk	August 30, 2006

Abstract

Some of the most significant security issues arising in the context of ubiquitous mobile computing are reviewed. Emerging technologies which may be able to help overcome these security problems are also described; in particular we consider methods for secure 'imprinting' of mobile devices, techniques proposed for establishing trust between devices with no prior relationship, and finally the relevance of trusted computing technology to mobile security issues.

Zusammenfassung

German abstract goes here

Widmung

To the memory of Professor Doctor Thomas Beth

1 Mobility and security

Over the last twenty years, mobile telecommunications has grown from being a small niche technology to a massive industry. Mobile telephones are now ubiquitous, and the distinctions between PCs, PDAs, mobile phones and other mobile devices are becoming increasingly blurred.

At the same time, personal computing devices are becoming increasingly ubiquitous, mobile, and connected. Of particular significance in this context is the growing use of wireless communications (typically using variants of the IEEE 802.11 protocol) to enable mobile devices to connect to the Internet anywhere, anytime.

Against this background, the security of information handled by these devices, and of the mobile devices themselves, becomes ever more important, not least because of our growing dependence on these devices. It is rather frightening to consider what might happen if these computing devices suddenly lost the capability to communicate wirelessly.

The main purpose of this paper is to review some of the most serious of the security issues arising in the context of ubiquitous mobile computing. We also consider some emerging technologies which may be able to help overcome these security problems. In particular we consider methods for secure 'imprinting' of mobile devices, techniques proposed for establishing trust between devices with no prior relationship, and finally the relevance of trusted computing technology to mobile security issues.

2 Security issues

2.1 Security threats

Most of the security issues arising for networks of mobile devices connected using wireless means are the same as those arising in more conventional wired networks; these concerns are documented in many standard security textbooks (see, for example, [4, 9]). That is, threats exist to the confidentiality, integrity and availability of information handled by these devices; there is also a need to protect the resources of the devices (including data, processing power, etc.) against unauthorised use.

It is, of course, true that some of these security issues may be exacerbated by the ease with which wireless communications can be intercepted. For example, it is often claimed that mobile telephone calls are more readily intercepted than calls made using the wired network; it is certainly true that the ease with which wireless transmissions can be overheard has caused embarrassment on more than one occasion in the past. Whilst this latter problem is no longer a serious threat in practice because of the widespread use of over-the-air encryption of telephone calls in GSM and third generation systems, the threat of interference with wireless communications remains.

As a result, consideration of the threats to wireless communications between mobile devices leads to a familiar list of requirements for security services, most importantly for:

- *Entity authentication*: devices need a means to verify the correctness of a claimed identity for a remote device, or, more generally, to verify the claimed properties of a device, e.g. membership of an authorised group;
- *Authorisation*: measures need to be established to ensure that all access to resources is authorised; this typically requires prior authentication of the requesting entity;
- *Accountability*: this is a service closely related to authorisation, meaning that entities must be held accountable for all resource-related actions;
- *Confidentiality*: sensitive data needs protecting against being divulged to unauthorised parties;
- *Integrity*: data must be protected against unauthorised modification (including reordering, deletion or replication).

Apart from these security requirements, there may be related (but distinct) privacy requirements. For example, a user of mobile devices may not wish his or her identity to be revealed to wireless eavesdroppers. One reason to distinguish between security and privacy is that in some cases security and privacy requirements may actually conflict; for example, accountability often requires identifying the author of each action, which could conflict with a desire by an end user to remain anonymous.

Given that the security requirements for these emerging wireless mobile ubiquitous systems are much the same as those for more familiar systems, then what is the problem? That is, why do we need to consider these systems any further? Despite the fact that the security requirements are the same, there are nevertheless important differences in this new setting, which do give rise to major new problems. Most significantly there are differences

in the approach necessary to manage security, and in the necessary underlying technologies.

The first major difference we note is the breakdown of the model traditionally used when protecting closed corporate networks. Such networks are typically protected by a combination of physical measures (building access control, etc.) and the use of firewalls to protect all traffic sent between the *inside* and the *outside* of the corporate network. Unfortunately, the use of mobile wireless devices by employees outside of the closed and protected environment means that the notions of 'inside' and 'outside' no longer apply. In this context we should mention the work of the Jericho Forum[1]; this international collaboration of IT customer and vendor organisations is dedicated to the development of open standards to enable secure and boundaryless information flows across organisations.

Secondly, we must contrast the managed security environments of corporations, where all devices are owned and managed by the organisation, with the world of multiple interactions between personal devices owned by many disparate individuals and organisations. Providing security in a managed environment, where we can rely on the existence of a security infrastructure, is essentially a solved problem (albeit that there may be practical issues with deploying the technology). In a managed setting it is relatively straightforward, at least in principle, to establish and manage a security infrastructure, e.g. the key management systems established to support the provision of security services for the GSM and 3rd generation networks (see, for example, [7]).

However, managing security for ad hoc collections of communicating devices is a problem for which there are no simple solutions. Devices which may never have interacted before, and whose owners have no contractual or other relationship, may nevertheless need to interact in a secure way. For example, it may be mutually beneficial for a collection of devices to collaborate to form an *ad hoc* network, providing network connectivity to all participants, where no such connectivity would exist without cooperation. Indeed, such an idea is the basis of *peer-to-peer* (p2p) computing, a subject of rapidly growing practical importance. Establishing a basis for security in such an environment is problematic indeed.

2.2 Security requirements

We have already noted the need for the provision of authentication, authorisation, accountability, confidentiality and integrity services. We have also noted the potential difficulty of providing these in an unmanaged environment. We now briefly justify this claim, by examining what is necessary to provide some of the security services we have already identified.

We first observe that our *primary security requirements*, i.e. authentication, authorisation, confidentiality, etc., together with any privacy requirements, give rise to what we refer to here as *secondary requirements*. That is, if we assume that our primary requirements will be met using cryptographic techniques, then use of these techniques imposes its own set of requirements. In particular, use of cryptography requires the existence of a key management infrastructure, which must either provide shared secret keys for use with symmetric cryptographic techniques, or reliable copies of public keys with use with asymmetric techniques. Moreover, each pair of communicating entities will typically need to establish a security context, including not only any shared keys, but also information about the level of trust and authorisation capabilities of the respective parties.

Providing key management and content establishment services is therefore a fundamental requirement. However, major issues exist in establishing *ad hoc* security relationships between mobile devices, in the absence of a pre-existing security infrastructure. Initial trust setting is one such major issue.

Moreover, not only is there no obvious way to establish security contexts and perform key management in many mobile networks, but most users of mobile devices are also not likely to be security aware, and hence cannot be relied upon to perform complex security management functions. Therefore there is need for a minimal security configuration overhead on the user — (almost) automatic security initialisation of devices is required.

In the remainder of this paper we look in more detail at some of these secondary security requirements, and consider possible approaches to addressing these problems in specific scenarios. However, we really only scratch the surface of the problem, and this area remains both a problematic one for users, and also a focus of much research activity.

[1] www.opengroup.org/jericho/

3 Two fundamental security issues

3.1 Security context establishment

We start by considering a basic problem encountered when providing a security context for a pair of mobile devices, when they are owned or managed by the same person. Of course, the problem becomes more difficult when the devices are owned by different individuals, and we consider this problem later.

That is, to consider a simple scenario, suppose a user has just purchased a new mobile device, and wishes to enable it to communicate securely with another device (via a wireless link). At the same time, the user is not a security expert, and will not wish to perform a lengthy and complex initialisation procedure. This is, of course, a problem faced by every purchaser of a new Bluetooth enabled wireless device, e.g. a wireless headset for an MP3 player.

This leads to the notion of 'imprinting', as introduced by Stajano and Anderson (see, for example, [10]). In this context imprinting simply refers to the process where a security association is established between a newly initialised device and some other device.

Bluetooth does incorporate a PIN-based initialisation (imprinting) procedure, with the goal of setting up a shared secret key between two devices. However, the Bluetooth solution is, unfortunately, seriously flawed, as has been documented widely on the web and in the research literature (see, for example, section 9.5.2.1 of [7]). However, better solutions have been devised, and some of these are included in a recent international standard [5].

We conclude this discussion by giving a simple example of a mechanism from ISO/IEC 9798-6. This mechanism is designed for the case where one device (A say) has only a simple input interface (e.g. a button used to signal successful completion of the imprinting process) but has a more sophisticated output interface (e.g. a small screen), and the other device (B say) has a simple output interface (e.g. a pair of lights) but has a more sophisticated input interface (e.g. a numeric keypad). Other mechanisms exist to cover the cases where both devices have a simple input interface, and both devices have a simple output interface. The mechanism enables the two devices to be certain that they both have the same data string D, which we assume has been sent across the wireless interface (and could have been modified or spoofed by a malicious interceptor); whilst D is assumed to be public, it can be used with a mechanism such as Diffie-Hellman key agreement to establish a shared secret key between the two devices (as described in ISO/IEC 9798-6).

Both devices output a signal to acknowledge that they have received data D and that they are ready for the mechanism to commence. On observing that both devices are ready, the user enters a signal into device A to tell it to start. Device A generates a random secret key K, where K is suitable for use with a check-value function shared by the two devices. Using K, device A computes a short check-value as a function of D. The check-value and the key K are then output to the user by device A. The user reads the check-value and K from the output interface, and enter them both into device B using its input interface. Device B then uses the key K to re-compute the check-value as a function of its stored version of D. If the two check-values agree, then device B outputs a success signal to the user via its simple output interface; otherwise it shall give a failure signal. Finally, the user enters the result output by device B, i.e. success or failure, into device A via its simple input interface. Both the key and the check sum can be very short, e.g. 16–20 bits, meaning that the user will not be required to type in a large number of digits, and yet the mechanism will be secure, as long as the check-function is chosen carefully.

3.2 Trust and reputation

We next examine the problem of security context establishment in the case where devices belong to different, potentially unrelated users. On the face of it this seems an almost insoluble problem. Yet it is a problem which not only must be solved, but in the real world it is a problem we routinely solve in practice in our social interactions. One of the ways in which we as human beings manage interactions with strangers is via recommendations of friends and official bodies. We use these recommendations as input (along with a host of visual and other cues) into our assessment of the degree of trust we can place in the authenticity of a stranger.

This has led to a desire to try and emulate these notions of trust assessment and recommendation in the electronic world. Electronic analogues of trust already exist in systems designed to enable groups of users to assess each others' trustworthiness, e.g. in reputation systems used by online auction sites. A huge amount of recent research effort has thus been expended on designing methods for quantifying trust, and on developing techniques for computing a trust value. These techniques are

often completely *ad hoc*, and without any firm theoretical basis.

Interestingly, the basis for much of this recent research on quantifying trust was established a dozen years ago by Beth et al. [3]. This 1994 paper described a means of computing a trust value for an entity A based on trust values for A held by third parties for whom trust values are already known. The method proposed is shown to possess some nice mathematical properties. Indeed, despite the recent explosion of interest in the area, the work in [3] is still of significance.

Sadly, more recent work is often far less rigorous in its approach than that of Beth et al. Often ad hoc methods for trust computation are presented with no attempt made at a mathematical analysis. Most seriously, what appears to be lacking is any serious attempt to understand how robust various measures of trust are against deliberate attempts by malicious entities to manipulate the trust computations. In the study of cryptography we regard the use of cryptanalytic techniques as essential to try to judge the security of a cryptographic primitive. However, in contrast, general approaches to understanding the robustness of distributed trust computation methods appear to be lacking.

Even if we can come up with completely satisfactory and robust means for computing measures for the trustworthiness of entities, the problem remains of reliably identifying other network entities. That is, a means is needed for securely associating a name (or address) with a single network entity. Without such a means, a malicious entity could assume the identity of a trusted party.

One approach to resolve this issue has been devised by Aura, and is known as *Cryptographically Generated Addresses* [2]. In essence, a network device is required to generate a key pair for a signature scheme, and then to generate its address as a function of the public key using a one-way hash-function (see, for example, [6]). The entity can then authenticate itself against the derived address by using the private signature key. As long as the address space is sufficiently large (and the cryptographic primitives in use are secure), no third party can find a key pair that matches the address, and hence no third party can claim ownership of the address.

It should be clear that cryptographically generated addresses prevent one user from claiming to own another user's address. However, it does not prevent a user from claiming many addresses simultaneously. Such an attack, known as a Sybil attack, poses a threat in some p2p scenarios.

4 Trusted computing

We now consider one technology which appears to have the potential to solve a wide variety of the problems associated with mobile device interactions. This technology is known as *Trusted Computing*.

It is possible that the existence of trusted functionality on future mobile platforms will help to solve many apparently intractable security and privacy problems in the mobile world. Some degree of trusted functionality is already present in most new mobile phones, to support security services required by network operators, such as International Mobile Equipment Identifier (IMEI) protection (to protect against reuse of stolen phones), and SIMlock (to prevent re-use of subsidised phones with different SIMs). However, whilst current trusted functionality is manufacturer specific, there is a move to use functionality conformant with the industry standard Trusted Computing Group (TCG) specifications[2].

In general terms, Trusted Computing refers to hardware-based functionality in a computing platform that: (a) provides a hardware (cryptographic) basis of trust for system boot, integrity verification of applications, etc., and (b) provides a means for an external third party to verify the current state of a platform. Considerable resources are currently being devoted to both hardware and software that supports trusted computing technology. A number of vendors have produced TPMs (Trusted Platform Modules) conformant with the TCG specifications. PCs incorporating TPMs are now widely available. Meanwhile operating system vendors and providers, including Microsoft and the open source community, are working on operating systems exploiting this functionality (note, in particular, the OpenTC project[3]).

Of particular relevance here are two main properties of a TCG compliant platform (or *trusted platform*).

- First, any such platform possesses a secure environment containing a platform-specific asymmetric key pair, the means to compute cryptographic functions, and the means to cryptographically prove to

[2] www.trustedcomputinggroup.org
[3] www.opentc.net

a third party its right to own one of a (potentially large) number of externally generated aliases. This in itself can provide the means to help establish security contexts with other devices.

- Second, a trusted platform is able to securely demonstrate to any other party precisely which software it is running. In the context of a p2p network, or a collaborative *ad hoc* network, this will be potentially hugely valuable in proving that an entity is cooperating as required (and is not behaving in a 'selfish' fashion).

We conclude this discussion of trusted computing by considering ways in which TCG functionality might help solve some of the most pressing security and privacy problems arising in mobile networks.

4.1 Stable identities

As we have already mentioned, in scenarios lacking a security infrastructure, the Sybil problem (where a single rogue entity claims multiple addresses) is a major challenge. Examples of where this presents a problem include p2p and *ad hoc* network settings.

More specifically, in an *ad hoc* network, newly admitted devices will typically need to be assigned a network address (or addresses). In the absence of a fixed infrastructure this is problematic. Many currently proposed solutions for address assignment can easily be subverted to lead to a denial of service, if Sybil attacks can be launched.

Trusted computing may be able to help, as discussed by Balfe *et al.* in chapter 10 of [7]. We now very briefly indicate how this might be achieved, in a way which nevertheless preserves user privacy.

Version 1.2 of the TCG specifications include a protocol known as *Direct Anonymous Attestation* (DAA) (see, for example, chapter 3 of [7]). This protocol enables a trusted platform to prove to a third party that it provides a reliable computing environment without revealing its identity. Balfe *et al.* describe a way of using the DAA protocol which enables all the actions of a particular platform to be linked, while not revealing the true identity of that platform. This provides a *stable identity* for a platform, both helping to prevent Sybil attacks and still providing a degree of privacy for an end user.

4.2 Identity management

A range of different *Single Sign-On* (SSO) technologies exist. In an SSO system, a user authenticates once to an *Identity Service Provider* (ISP), and this ISP then vouches for the identity of the user to multiple *Service Providers* (SPs). Clearly, the SP must trust the ISP to tell the truth about who has been authenticated and how. Typically this means that the ISP must be a networked entity remote to the user.

This may be inconvenient in a mobile setting, where the opportunity may not arise to set up the necessary third party relationships. As described by Pashalidis in chapter 6 of [7], trusted computing technology can he used to enable the ISP to be implemented on the user platform, in such a way that the SP can verify its trustworthiness.

4.3 Personal information management

A growing number of possibilities now exist for Internet SPs to offer services tailored to end users. However, this possibility also represents a privacy threat, since the SP will typically need to know potentially privacy-breaching information about an end user in order to provide the tailored service. One key example (relevant in a mobile context) is the use of location information. How does the subject of location information prevent it being disseminated and used in unauthorised ways?

It is possible that trusted computing can help with this problem. The holder of location information, and associated policy information (e.g. defining user preferences), can use trusted computing functionality to check the platform requesting this information before sending it. This check could involve verifying the type of recipient platform and the identity of the receiving application.

4.4 Limitations of trusted computing

Before concluding, it is important to note that trusted computing technology has not been greeted with universal enthusiasm; see, for example, [1]. In particular, suggestions have been made that trusted computing is both a potential threat to user privacy and a threat to the ability of the owner of a PC to use it as he or she sees fit. However, some of the more outspoken criticisms seem to have arisen from misconceptions about the likely applications of the technology.

Of course, until the technology is in wide use, it is difficult to know whether trusted computing will be a force for good or ill. Whilst the hardware is becoming increasingly ubiquitous, at least on the desktop, the software necessary to support application of the hardware is taking much longer to appear. The appearance of such software does

not appear imminent, particularly since the ambitions of Microsoft's Vista now seem rather more limited than was the case a year or so ago. Indeed, the open source community, as exemplified by the work of the European Open Trusted Computing project, seems likely to produce software capable of taking full advantage of the technology long before Microsoft.

5 Conclusions

In this paper we have attempted to convey some of the major security issues presented by the growing reality of ubiquitous networks of mobile computing devices. We have sketched some of the possible approaches to addressing these security problems, and have also highlighted areas where further research is required. The main finding of this paper is that trusted computing technology could prove to be of major benefit in solving some of the most fundamental security problems that beset mobile and ubiquitous computing. These problems include that of providing stable identities for devices, providing single sign-on services, and controlling the dissemination of personal information.

References

[1] B. Arbaugh: *Improving the TCPA specification. IEEE Computer*, 35(8):77–79, August 2002.

[2] T. Aura: *Cryptographically generated addresses.* RFC 3972, The Internet Society, 2005.

[3] T. Beth, M. Borcherding, B. Klein: Valuation of trust in open networks. In: *Proc. ESORICS 94*, 1994, pp. 3–18.

[4] D. Gollmann: *Computer Security.* John Wiley and Sons, 2005 (2nd edition).

[5] ISO/IEC 9798-6: *Information technology — Security techniques - Entity authentication - Part 6: Mechanisms using manual data transfer.* International Organisation for Standardization, 2005.

[6] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone: *Handbook of Applied Cryptography.* CRC Press, Boca Raton, NY, 1997.

[7] C. J. Mitchell (editor): *Security for Mobility.* IEE, London, 2004.

[8] C. J. Mitchell (editor): *Trusted Computing.* IEE, London, 2005.

[9] C. P. Pfleeger, S. L. Pfleeger: *Security in Computing.* Prentice Hall, Upper Saddle Rover, NJ, 2002 (3rd edition).

[10] F. Stajano, *Security for Ubiquitous Computing.* John Wiley and Sons, 2002.