SOFTWARE: Challenges of the 21st Century
1999 AT&T Software Symposium

The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks

Frank Stajano[1,2] **and Ross Anderson**[2]
http://www.uk.research.att.com/~fms/,
http://www.cl.cam.ac.uk/~rja14/
[1]AT&T Laboratories Cambridge and [2]University of Cambridge Computer Laboratory

15th September 1999

Abstract

In the near future, many personal electronic devices will be able to communicate with each other over a short range wireless channel. We investigate the principal security issues for such an environment. Our discussion is based on the concrete example of a thermometer that makes its readings available to other nodes over the air. Some lessons learned from this example appear to be quite general to ad-hoc networks, and rather different from what we have come to expect in more conventional systems: denial of service, the goals of authentication, and the problems of naming all need re-examination. We present the *resurrecting duckling* security policy model, which describes secure transient association of a device with multiple serialised owners.

1 Introduction

The established trend in consumer electronics is to embed a microprocessor in everything—cellular phones, car stereos, televisions, VCRs, watches, GPS (Global Positioning System) receivers, digital cameras—to the point that most users have lost track of the number of items they own that contain one. In some specific environments such as avionics, electronic devices are already becoming networked; in others, work is underway: consumer electronics makers, for example, are promoting the Firewire standard [10] for PCs, stereos, TVs and DVD players to talk to each other.

We envisage that, in the near future, this networking will become much more general. After the microprocessor, a short range wireless transceiver will be embedded in everything, so that many gadgets will become more useful and effective by communicating and cooperating with each other. A camera, for example, might obtain the geographical position and exact time from a GPS unit every time a picture is taken, and record that information with the image. At present, if the photographer wants to record a voice note with the picture, the camera must incorporate digital audio hardware; in the future, the camera might let her speak into her audio recorder or cellphone. Each device, by becoming a network node, may take advantage of the services offered by other nearby devices instead of having to duplicate their functionality.

This vision of embeddable wireless connectivity has been in development for several years at AT&T Laboratories Cambridge in the context of the Piconet [3] project and is also being pursued, although with emphasis on different aspects, by several other groups including HomeRF [9, 14], IrDA [11] (which uses infrared instead of radio) and Bluetooth [5, 8].

Everyone—including potential users—understands that wireless networking is prone to passive eavesdropping attacks. But it would be highly misleading to take this as the only, or even the main, security concern.

In this paper we investigate the security issues of an environment characterised by the presence of many principals acting as network peers in intermittent contact with each other. To base the discussion on a concrete example we shall consider a wireless temperature sensor. Nearby nodes may be authorised to request the current temperature, or to register a "watch" that will cause the thermometer to send out a reading when the temperature enters a specific range. We wish to make our thermometer useful in the widest range of environments including environmental monitoring, industrial process control and medicine.

We will therefore consider how we can enable our thermometer to support all the security properties that might be required, including *confidentiality*, *integrity* (and its close relative *authenticity*) and *availability*. Contrary to academic tradition, however, we shall examine them in the opposite order, as this often (and certainly in our case) reflects their actual importance.

In our analysis we must not overlook the defining characteristics of this environment: portable wireless consumer electronics devices imply small size, low cost and battery-powered operation. From these constraints, and in particular from battery weight issues, follow limits on the computing speed of the processor and even on the total amount of computation and communication that can be performed. These constraints are new for the usual distributed systems scenario, and so are their effects on security.

2 Availability

Availability means ensuring that the service offered by the node will be available to its users when expected. In most non-military scenarios, this is the security property of greatest relevance for the user. All else counts little if the device cannot do what it should.

Apart from radio jamming attacks on the communication channel, the novel and interesting service denial threat concerns battery exhaustion.

A malicious user may interact with a node in an otherwise legitimate way, but for no other purpose than to consume its battery energy. Battery life is the critical parameter for many portable devices, and many techniques are used to maximise it; in Piconet, for example, nodes try to spend most of the time in a sleep mode in which they only turn on the radio receiver, or even the processor, once in a while (the period can be set from a few seconds to several minutes). In this environment, energy exhaustion attacks are a real threat, and are much more powerful than better known denial of service threats such as CPU exhaustion; once the battery runs out the attacker can stop and walk away, leaving the victim disabled. We call this new attack the **sleep deprivation torture**.

For any public access server, there is necessarily a tension between the contrasting goals of being useful to unknown users and not succumbing to vandals. Whereas some applications can restrict access to known principals, in others (such as web servers and name servers) this is infeasible since the very usefulness of the service comes from its being universally available.

If a server has a primary function (such as sending the outside temperature to the meteorological office every hour) and a distinct auxiliary function (such as sending the current temperature to anyone who requests it) then these functions can be prioritised; a reservation mechanism can ensure that the higher priority use receives a guaranteed share of the resource regardless of the number of requests generated by the lower priority uses. Even just turning on the receiver to listen to unexpected messages may become a rationed activity. The highest priority use of all may be battery management: if one can accurately estimate the amount of usable energy remaining, then the service can be monitored and managed—provided that the process does not itself consume too much of the resource it is intended to conserve.

It has been suggested that an attacker might starve the legitimate users by forcing the node to become unresponsive in self-defense, and that it would be better to identify and blacklist the source of the attack. However, this forces the node to use a lot of storage to perform traffic analysis, and in any case the attacker could mount his sleep deprivation attack via multiple paths. Authenticating other nodes is a possible solution, but has difficulties which we will discuss in the next section. The most general defence against sleep deprivation remains a resource reservation mechanism.

3 Authenticity

Authenticity is ensuring that the principals with whom one interacts are the expected ones.

In most applications where security matters, authenticity is an essential prerequisite. Granting resources to, obeying an order from, or sending confidential information to a principal of whose identity we are unsure is not the best strategy for protecting availability, integrity and confidentiality.

Many authentication protocols have been developed to identify authorized principals. However, the ad-hoc network environment introduces a fundamental new problem: the *absence of an online server*. When a new node comes within range, we cannot connect to an authentication server to obtain a Kerberos ticket or to check the validity of an exhibited certificate: suddenly, the traditional solutions no longer apply. Besides, the problem of greatest relevance in this new context is itself new—**secure transient association**, which we shall now describe.

If a householder owns a device, say a universal remote control, that lets her control various other devices in her home (such as hi-fi and television components, the heating system, lights, curtains and even the locks and burglar alarm) then she will need to ensure that a new device she buys from the shop will obey her commands, and not her neighbour's. She will want to be assured that a burglar cannot take over the heat sensing floodlight in the garden, or unlock the back door, just by sending it a command from a remote control bought in the same shop.

As well as being *secure* (whatever that means), the association between the controller and the peripheral must also be *transient*. When a householder resells or gives away her television set or hi-fi or fridge, the appliance will have to obey another controller; when her controller breaks down (or she decides to replace it or upgrade its operating system), she must be able to regain control of all the gadgets she already owns.

A central authentication service is possible for expensive consumer durables; most governments run such services for houses and cars. But there is no prospect that this will be extended to all durable consumer goods; the UK government abandoned dog licensing some years ago as uneconomic. In any case, there would be very grave civil liberties objections to the government maintaining lists of all PCs, hi-fis and DVD players in the country; the outcry over the Pentium III processor ID [7] indicates the likely level of political resistance. Even the existing registration services stop short of managing keys; the replacement of car keys is left to the motor trade, while house locks are completely uncontrolled. So it is desirable that key management be performed locally: the last thing we want is to impose an expensive and unpopular central solution. Yet it would be nice if we could still provide some means of making a stolen DVD player harder to resell.

Another insight comes from scenarios where we have a pool of identical devices, such as a bowl of disinfectant containing ten thermometers. The doctor does not really care which thermometer she gets when she picks one up, but she does care that the one her palmtop talks to is the same one she is holding and not any other one nearby.

A metaphor inspired by biology will help us describe the behaviour of a device that properly implements secure transient association. As Konrad Lorenz beautifully narrates [13], a duckling emerging from its egg will recognise as its mother the first moving object it sees that makes a sound, regardless of what it looks like: this phenomenon is called **imprinting**. Similarly, our device (whose egg is the shrink-wrapped box that encloses it as it comes out of the factory) will recognise as its owner the first entity that sends it a secret key. As soon as this "ignition key" is received, the device is no longer a newborn and will stay faithful to its owner for the rest of its life. If several entities are present at the device's birth, then the first one that sends it a key becomes the owner: to use another biological metaphor, only the first sperm gets to fertilise the egg.

We can view the hardware of the device as the body, and the software (particularly the state) as the soul. As long as the soul stays in the body, the duckling remains alive and bound to the same mother to which it was imprinted. But this bond is broken by death: thereupon, the soul dissolves and the body returns in its pre-birth state, with the resurrecting duckling ready for another imprinting that will start a new life with another soul. Death is the only event that returns a live device to the pre-birth state in which it will accept an imprinting. We call this process **reverse metempsychosis**. Metempsychosis refers to the transmigration of souls as proposed in a number of religions; our policy is the reverse of this as, rather than a single soul inhabiting a succession of bodies, we have

a single body inhabited by a succession of souls[1].

With some devices, death can be designed to follow an identifiable transaction: in medicine, a thermometer can be designed to die (and lose its memory of the current patient's temperature history) when returned to the bowl of disinfectant at the nursing station. With others, we can arrange a simple timeout, so that the duckling dies of old age. With other devices (and particularly those liable to be stolen) we will arrange that the duckling will only die when so instructed by its mother: thus only the currently authorised user may transfer control of the device. In order to enforce this, some level of tamper resistance will be required: assassinating the duckling without damaging its body should be made suitably difficult and expensive.

Sometimes the legitimate user will lose the shared secret (e.g. when the password is forgotten or the remote control is broken beyond repair). To be able to regain control of the duckling, an easy solution is **escrowed seppuku**: someone other than the mother, such as the manufacturer, holds the role of Shōgun with a master password that can command the device to commit suicide. But this reintroduces centralised control: the Shōgun can always take over any device and, more worryingly, he is forced to keep a global database (built from warranty cards sent in by the owners as they purchase the ducklings) mapping serial numbers to legitimate owners, lest he give out the seppuku key to the thief of the DVD player. The experience of the motor industry shows that the manufacturer, whose profits come from sales, often has little incentive to make its products hard to steal. This makes us suspicious about how careful or scrupulous the Shōgun would be in managing such an escrow database. To keep key management local, a better solution is for the mother to backup the ignition key, and even split it into shares if necessary; this gives her better privacy guarantees, at the cost of transferring the key management burden from the Shōgun to her. A key recovery facility may be genuinely beneficial to the mother, as long as she is free to choose local escrow parties that she trusts (the neighbours holding a copy of her house keys) instead of global ones imposed from above. We prefer the decentralised end of this wide spectrum of possible key recovery solutions, though there might exist applications with little or no civil liberties implications for which a different trade-off between usability and privacy might prove more appropriate.

There are also applications in which only part of the duckling's soul should perish. Our thermometer might be calibrated every six months, and the calibration information must not be erased along with the patient data and user key when the device is disinfected, but only when it is plugged into a calibration station. So we may consider the device to be endowed with two souls—the calibration state and the user state—and a rule that the latter may not influence the former. So our resurrecting duckling security policy may be combined with multilevel security concepts (in fact, "multilevel secure souls" are a neat application of the Biba integrity policy model [4]).

During the imprinting phase, as we said, a shared secret is established between the duckling and the mother. One might think that this is easy to do: the mother generates a random secret and encrypts it under the public key of the duckling, from which it gets back a signed confirmation.

But many of our nodes, due to their peanut-sized CPU, lack the ability to do public key, and even if they did it would still not help much. Suppose that a doctor picks up a thermometer and tries to get his palmtop to do a Diffie-Hellman key exchange [6] with it over the air. How can he be sure

[1] Prior art on this technique includes Larry Niven's science fiction novel *A World Out of Time* (1977) in which convicted criminals have their personalities "wiped" and their bodies recycled.

that the key has been established with the right thermometer and not one of the others sitting in the bowl of disinfectant? If both devices have screens, then a hash of the key might be displayed and verified manually; but this is both tedious and error-prone, in an environment where we want neither. Besides, we do not want to give a screen to every device: sharing peripherals is one of the goals of ad-hoc networking.

In many applications there will only be one satisfactory solution, and we advocate its use generally as it is effective, cheap and simple: physical contact. When the device is in the pre-birth state, simply touching it with an electrical contact that transfers the bits of a shared secret constitutes the imprinting. No cryptography is involved, since the secret is transmitted in plaintext, and there is no ambiguity about which two entities are involved in the binding.

4 Integrity

Integrity means ensuring that the node has not been maliciously altered. The recipient wants to be sure that the measurements come from the genuine thermometer and not from a node that has been modified to send out incorrect temperature values (the application might not be meteorology or medicine, but a fire alarm warning system).

The threat model here assumes that the sensing node may be left unattended for long periods of time and that, sooner or later, an attacker will pick it up, mess around with its internals and put it back where it was. A legitimate node may then end up unknowingly transacting with a maliciously altered one.

This can in theory be avoided by making the nodes tamper-proof, but it is much easier to talk about this property than to implement it in practice [1], especially within the cost and form factor constraints of personal consumer electronics devices. Under the circumstances, it is not clear whether any extra assurance is given by furnishing our thermometer with the ability to do public key cryptography; such a device can have its private key read out just as a device with a certificate but without a private/public keypair can be forged.

In such environments it may often be more suitable to use tamper-evidence mechanisms (such as seals) rather than more expensive tamper-proofing ones (such as sensing switches that erase the memory). In this case, one must still design the device so that non-intrusive attacks (such as those based on protocol failure, power analysis and glitch attacks [2]) are not practical; it is also necessary to take into account the time that might pass before a broken seal is noticed, and the likelihood of successful attacks on the sealing mechanism [12].

It must also be realised that the tampering may not be limited to the onboard code and keys: a very effective attack on the unattended thermometer is to simply replace its analogue sensing element with a bad one. This attack highlights that even enclosing the entire processor, memory and backup battery in a high-grade tamper resistant enclosure, with only a ribbon connector to interface with the outside world, would still leave us vulnerable to direct attacks on its “peripherals”. Bringing the sensor itself within the tamper resistant enclosure may make manufacturing too expensive (the computing and communication core willl no longer be a modular building block) and may even interfere with the proper working of the sensor. So the transducer may be an Achilles’ heel,

and it may not be worth spending large sums on tamper-proofing the core if the sensor cannot economically be protected.

5 Confidentiality

We find that we have little to say about confidentiality other than remarking that it is pointless to attempt to protect the secrecy of a communication without first ensuring that one is talking to the right principal. Authenticity is where the real issues are and, once these are solved, protecting confidentiality is simply a matter of encrypting the session with whatever key material is made available by the authentication process.

6 Conclusions

We examined the main security issues that arise in an ad-hoc wireless network of mobile devices. The design space of this environment is constrained by tight bounds on power budget and CPU cycles, and by the intermittent nature of communication. This combination makes much of the conventional wisdom about authentication, naming and service denial irrelevant; even tamper resistance is not completely straightforward.

There are interesting new attacks, such as the sleep deprivation torture, and limitations on the acceptable primitives for cryptographic protocols. However, there are also new opportunities opened up by the model of secure transient association, which we believe may become increasingly important in real networking applications.

The contribution of this paper was to spell out the new problems and opportunities, and to offer a new way of thinking about the solution space—the resurrecting duckling security policy model.

7 Acknowledgements

Thanks to Alan Jones for suggesting the wireless thermometer, a prototype of which had just been built in the context of Piconet, as a minimal but still meaningful example. Thanks also to Jonathan Smith and his group at the University of Philadelphia for insightful criticisms when this research was presented there in June 1999.

This is an abridged and updated version of the paper by the same name [15] presented in April 1999 at the 7th Security Protocols workshop, Cambridge, UK.

References

[1] Ross Anderson and Markus Kuhn. Tamper resistance—a cautionary note. In *Proc.* 2nd *USENIX Workshop on Electronic Commerce*, 1996. `http://www.cl.cam.ac.uk/~mgk25/tamper.pdf`.

[2] Ross Anderson and Markus Kuhn. Low cost attacks on tamper resistant devices. In Mark Lomas, editor, *Security protocols: 5th international workshop, Paris, France, April 7–9, 1997: proceedings*, volume 1361 of *Lecture Notes in Computer Science*, pages 125–136. Springer-Verlag, 1998. `http://www.cl.cam.ac.uk/~mgk25/tamper2.pdf`.

[3] Frazer Bennett, David Clarke, Joseph B. Evans, Andy Hopper, Alan Jones, and David Leask. Piconet: Embedded mobile networking. *IEEE Personal Communications*, 4(5):8–15, October 1997. `http://www.uk.research.att.com/abstracts.html#79`.

[4] Kenneth J. Biba. Integrity considerations for secure computer systems. Technical Report MTR-3153, MITRE Corporation, April 1975.

[5] Bluetooth SIG. `http://www.bluetooth.com/`.

[6] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT-22(6):644–654, November 1976.

[7] Electronic Privacy Information Center, JunkBusters, and Privacy International. `http://www.bigbrotherinside.org/`.

[8] Jaap Haartsen, Mahmoud Naghshineh, Jon Inouye, Olaf J. Joeressen, and Warren Allen. Bluetooth: Visions, goals, and architecture. *ACM Mobile Computing and Communications Review*, 2(4):38–45, October 1998.

[9] HomeRF Working Group. `http://www.homerf.org/`.

[10] IEEE. IEEE standard for a high performance serial bus. IEEE Standard 1394, 1995.

[11] Infrared Data Association. `http://www.irda.org/`.

[12] Roger G. Johnston and Anthony R.E. Garcia. Vulnerability assessment of security seals. *Journal of Security Administration*, 20(1):15–27, June 1997. `http://lib-www.lanl.gov/la-pubs/00418796.pdf`.

[13] Konrad Lorenz. *Er redete mit dem Vieh, den Vögeln und den Fischen* (King Solomon's ring). Borotha-Schoeler, Wien, 1949.

[14] Kevin J. Negus, John Waters, Jean Tourrilhes, Chris Romans, Jim Lansford, and Stephen Hui. HomeRF and SWAP: Wireless networking for the connected home. *ACM Mobile Computing and Communications Review*, 2(4):28–37, October 1998.

[15] Frank Stajano and Ross Anderson. The resurrecting duckling: Security issues in ad-hoc wireless networks. 1999. To appear in B. Christianson, B. Crispo and M. Roe (Eds.). *Security Protocols, 7th International Workshop Proceedings*, Lecture Notes in Computer Science, Springer-Verlag. `http://www.cl.cam.ac.uk/~fms27/duckling/`.