Embedded Security in a Pervasive World

Christof Paar
Horst Görtz Institute for IT Security
Ruhr University Bochum, Germany
cpaar@crypto.rub.de

André Weimerskirch
escrypt – Embedded Security GmbH
Bochum, Germany
aweimerskirch@escrypt.com

Abstract

Embedded systems have become an integral part of our everyday life. Devices like vehicles, household appliances, and cell phones are already equipped with embedded microcontrollers. The networking of the myriads of embedded devices gives rise to the brave new world of pervasive computing. Pervasive computing offers enormous advantages and opportunities for users and businesses through new applications, increased comfort, and cost reduction. One often overlooked aspect of pervasive computing, however, are new security threats.

This article describes security issues in current and future pervasive security scenarios, ranging from privacy threats and unreliable products to loss of revenue. We also highlight the opportunities, such as new business models, which are enabled through strong embedded security solutions. Current research issues are also summarized. As case studies, we introduce security aspects in future automotive systems and in ad-hoc networks.

1 Introduction

Embedded systems have become a centrally important aspect in a wide variety of applications, ranging from hand-held devices to household appliances and RFID tags. Embedded controllers are said to have a market share of 98% or more of the global processor market, implying that less than 2% of all processors are employed in traditional computers [3]. The ever decreasing costs and form factor of digital "intelligence" together with increased capabilities has lead to a world of pervasive computing. A widely shared view is that pervasive computing is the next major evolutionary step in information technology, merging the notions of networks and computers with everyday devices [9]. One aspect of this evolution is the rise of new, and partially unique, security issues.

Unfortunately, many solutions developed for securing general IT systems, such as computer networks or data bases, are not applicable or not sufficient for embedded security. For instance, in many pervasive applications, communications must be kept to a minimum due to the mobile nature of applications, the target systems are often computationally extremely weak (8-bit processors are by far the most common embedded platform), an attacker often has physical control over the device, and there is a lack of IT infrastructure such as PKI. In addition to those technical boundary conditions, embedded applications tend to be extremely cost-sensitive because they are more often than not extremely high-volume devices in very competitive markets. It is important to note that pervasive security serves not only the purpose of assuring the smooth functioning of applications, but is also an enabling technology for new business models, such as fee-based feature activation in embedded systems

Most technologies needed for embedded security are currently under development in industry and academia, and embedded security is arguably one of the most active areas within applied security and cryptography. It should be noted that embedded security has only been recognized as a proper sub-discipline of security since the beginning of the decade. This article attempts to give an overview of the challenges but also of the opportunities which strong pervasive security solutions can offer.

In the following we will first motivate why pervasive security is needed. We will then present current areas of research and finally present case studies. During this overview we will often describe examples from the automotive world. Even though vehicles are only one of a many systems with perva-

sive security issues, cars are an excellent example as they are highly mobile, omnipresent, and equipped with a high number and wide variety of embedded nodes.

2 Why Do We Need Pervasive Security (or: Hard Disk Crash vs. Car Crash)

Ubiquitous embedded devices are the backbone of the pervasive computing world. Not surprisingly, the brave new pervasive world implies new security issues. It is important to note that there is not one single threat against pervasive computing systems. Rather, due to the extremely diverse nature of embedded applications, there is a wide range of damage that can be done through abuse in a pervasive world. The potential threats, ranging from privacy violation to financial loss or even bodily harm, will be discussed in this section. We argue that pervasive security is needed due to following developments: risk potential, financial aspects, new business models, privacy, reliabilty and legislation.

Risk Potential Due to the close coupling with the physical environment the risk involved in embedded systems can be much larger than the risk in conventional IT applications. For instance, hacking of an automotive brake system can have far more physical consequences than a hard disk destroyed by a computer virus. Less dramatic attacks, and perhaps less far-fetched, are manipulation of networked household appliances. It is easy to imagine many pranks (or attacks with more malicious intent) which involve washing machines, refrigerators or microwave ovens. Also, as an increasing number of embedded applications are involved in safety-critical applications — e.g., in ITS (intelligent transport systems such as automotive, railroad or airplane), military, or control systems — IT security is dramatically gaining importance. Generally speaking, the familiar threats of conventional IT systems (hacking, phishing, pharming, etc.) which, after all, target mainly abstract digital data are extended to include threats against our real physical environment. We would like to comment that this simple observation is hardly being discussed thus far.

Financials There are an increasing number of pervasive applications that involve financial aspects, such as digital entertainment content in home and mobile devices, location-based services for hand-held devices, or smart cards with e-wallet functions. It is abundantly obvious that there is a high incentive to manipulate such systems for financial gain. Another class of applications which require security for revenue protection is identification of parts. A prime example are printer cartridges where the OEM attempts to enforce the use of original parts in order to increase its revenues.

New business models In addition to direct financial gain, there will be many pervasive applications where the *business model* relies on strong security functionality. In such systems, manipulation may lead to a loss of revenue. Pay-TV is one of the best established examples of an embedded system with high security requirements in order to protect a business model. Many future applications, e.g., time-limited feature activation in fielded products, will require even more sophisticated security solutions. More about this will be said under "Digital Rights Management" in the subsequent section.

Privacy Privacy is already a concern in conventional IT systems. In pervasive computing there is often an intimate link between human user and "computing" device. Privacy concerns include disclosure of a user's location or of his/her behavior, e.g., when location based services are accessed. Mobile devices such as smart phones and also vehicles come with privacy issues. If we look a bit ahead in time, it is easy to see that applications such as wearable computers will only aggravate the threat to privacy. Another domain with privacy concerns is the widely discussed field of RFID applications. RFID tags are one of the most intriguing embedded technologies. It is expected that they will be widely deployed in the near future, providing almost pure pervasiveness and ubiquitousness. When every item from our shopping cart is tagged with an RFID and can easily be tracked, privacy can obviously become a major concern. It should be noted that privacy is both a technical and organizational matter.

Reliability In many pervasive applications manipulations can harm the reliability of a prod-

uct. There is trend for certain classes of pervasive devices to allow (remote) software updates. Even though this function offers great opportunities to both users and manufacturers, unauthorized software updates can lead to sub-optimum products with reliability problems. An application domain where this issue is widely discussed is the so-called "chip tuning" in the automotive context.

Legislation Legislative requirement will force certain pervasive applications to provide strong security, e.g., road toll systems, e-voting systems, or mobile banking applications. For instance, tachographs for trucks in the EU must all be certified according to [2]. The recent discussions about electronic voting in Europe and the US shows impressively the perils which can rise from manipulation of embedded systems. This situation implies that the provider of such systems must be able to provide security solutions which gain government approval. Given the complexity of a security certification process à la Common Criteria or ITSEC, this is not an easy undertaking for industries without too much prior experience in security.

3 Opportunities Offered through Pervasive Security

In the following we'll describe pervasive systems in which strong security solutions will not only help to protect against abuse, but will also enable new functions and business models.

Digital Rights Management

Maybe the most exciting new applications in the pervasive world are driven by business models distributing digital data. Examples include software upgrades for household appliances; infotainment content for cars including navigation data and music or video and games for rear-seat passengers; or after market activation of features for cell phones. Today many embedded devices are already equipped with some wireless connection such as Bluetooth, WLAN, and Zigbee. Embedding reliable digital rights management (DRM) enables business models for usage-metered and on-demand utilization of digital content, software and even hardware beyond the classical lump-sum model. For instance, time-limited utilization can be provided in the same way as quantity-limited utilization. Furthermore, almost arbitrary combinations are possible. For instance, an afterwards activated enhanced comfort sensor for a vehicle (e.g., tire air pressure sensor) may be enabled for a trial period of four weeks. Business models using digital content that has usage or access restrictions are only possible with a secure and reliably implemented DRM system. As it could be seen in various DRM scenarios such as pay-TV, online music stores, or video game consoles, without such a secure anchor the business model will certainly fail.

Secure Access: Remote Maintenance

It is useful to allow a remote access to embedded devices, mainly for maintenance purposes. For instance, there is a diagnostic access in vehicles for workshops. New business models already thought of go far beyond traditional mechanisms. For instance, household appliances could have a remote access via Internet that allows a technician to check the appliance in case of a failure before actually driving to the household and checking it in the real world (and realizing he/she did not bring the needed spare part). If the remote maintenance is properly designed, this might save the technician from driving twice to the household, which in the end saves cost for the owner. Clearly, there is a wide variety of applications where remote maintenance is already used or will be soon used, such as for remote installation of industrial network appliances (e.g., VPN and Firewall for automation industry).

Secure access can be implemented relatively easy by a challenge and response scheme (CR) as depicted in Figure 1. Here, the remote device Alice sends a random number c as challenge which the technician Bob encrypts using the shared secret K and sends back. Alice can then check if Bob knew the secret K by decrypting the response. The crucial part in an embedded device is again the storage of the secret K. Note that secure access schemes can also be based on asymmetric cryptography and on time-stamps. Even though CR protocols are well established, the challenge at hand is to integrate them in a secure fashion in pervasive applications.

Software Integrity: Secure Software Download

Embedded devices usually contain a microcontroller as well as a software program that determines the

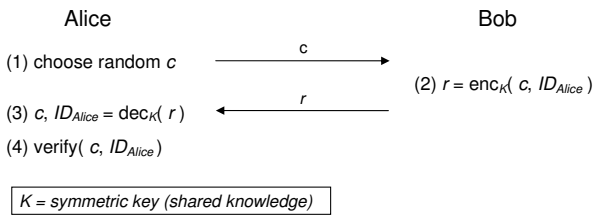

Figure 1: Challenge-and-response protocol.

behavior of the embedded device. An increasing number of embedded applications allow downloading of updated program code and data. Applications are as varied as the control unit of heavy machinery or a printer. Such software might be a mere firmware update, or a bug fix, or an update enabling additional functionality. The first case is also called software download or simply flashing (since flash memory is updated). The download might be performed over a diagnostic channel or another available communication channel such as Bluetooth or GSM. Once such communication channels of embedded devices are opened to the outside world for downloading software, its integrity must be ensured. An example for a malicious software download is the replacement of a firmware by an unauthorized party, e.g., as done on a large scale through chip-tuning in vehicles.

In order to control software updates, digital signatures play a central role. Here, a digital signature is attached to the new firmware. The embedded device is then able to verify authenticity of the new firmware. Only if the verification is successful is the new firmware is actually run by the device. This is also shown in Figure 2. A proper signature verification algorithm in this case is RSA with short exponent that runs in a few milliseconds on an ARM-class CPU. Again, provision of a digital signature itself is often not the main problem, but its integration in the pervasive device is.

Certainly a secure software download is only useful if there are neither hidden access points to the firmware available nor if a flawed implementation allows illegal access. Hence, in addition to the secure software download a very careful design and implementation phase has to be performed to make sure that only the defined access path is given.

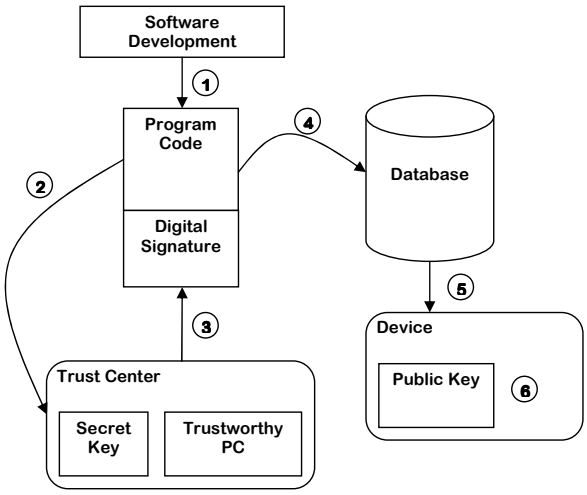

Figure 2: Secure software download.

Hardware Integrity: Component Identification

The secure identification of devices – a seemingly specific problem – is a major concern for a large number of applications. Counterfeiting of all kinds of products (ranging from textiles over consumer electronics to bank notes) and parts (from printer cartridges to spare parts for heavy machinery) is one area with urgent need for strong and secure device identification. A facile way of providing a basic protection against hardware manipulation can be achieved by mechanical countermeasures deploying special component constructions. This could be proprietary constructions that fit only into a system of a single manufacturer or subsystems that require proprietary (not publicly available) tools. An example is car radios. However, that solution is uncomfortable and provides only minimal hardware security.

More reliable approaches [8] for detection of non-original components are based on pervasive computing. The basic idea is to use small computing tags attached to each crucial component, in order to logically and physically identify components of a (technical) system. Such component identification schemes rely on the tamper-evidence of the computing tags that are tightly (non-removable) integrated into critical components and which can communicate with each other. RFID can play an important role here, but is certainly not the only solution. More advanced technologies such as PUF (physical unclonable functions) can offer interesting solutions too [7].

4 Research Areas in Pervasive Security

Embedded security has only recently evolved into a field of research. Until the late 1990s, research results were scattered across the technical literature. However, since roughly the beginning of this decade, the need for securing pervasive applications has become obvious. Today, embedded security is arguably one of the most active sub-disciplines within the field of security and cryptography. The CHES (Cryptographic Hardware and Embedded Systems) conference series is a good example of the growing importance of this topic. Some of the security issues in embedded systems are also addressed in the excellent book by Anderson [1]. In the following, we describe current and future research areas within embedded security.

Lightweight Cryptography

Embedded applications are commonly resource constrained with respect to processing power, gate count, power dissipation and monetary costs. Providing security in such environments is a major research challenge, especially since the underlying cryptographic algorithms are notoriously computationally intensive. The ever increasing scope of applications — ranging from sensor networks, RFID tags to electronic component identification in consumer electronics — with security needs in strictly constrained environments, requires the development of highly optimized software algorithms and hardware architecture. What is needed are cryptographic algorithms for extreme low-power ad-hoc networks, novel cryptographic schemes with low memory and processing requirements, and efficient arithmetic algorithms tailored for today's modern processors.

Physical Security

In contrast to many attack scenarios in traditional networks (Internet, LAN), an attacker often has physical access to the target device in pervasive computing systems. Examples include set-top boxes for digital content control, RFID, or wireless smart cards for financial transactions. Modern security solutions almost always hinge on the fact that the device is equipped with some type of cryptographic secret. In the real world, methods for extracting such keys from hardware devices by physical means, e.g., by observing the electromagnetic radiation or the power trace of an embedded security processor, has become one of the most dangerous attacks in recent years. Interestingly, physical security has become a major concern for the trusted computing community which needs to build digital rights management systems (e.g., digital content distribution via the Internet), and companies like IBM, Intel and Philips are currently looking intensively at this problem. In the PC world there was recently a so called Trusted Platform Module (TPM) introduced that was specified by the Trusted Computing Group (TCG) [6]. In the embedded world there are first approaches such as ARM's TrustZone. These approaches basically embed cryptographic functions as well as a secure key storage into the controller. For high-security applications the controller should also be tamper-resistant as is the case for smart-card controllers.

Ad-Hoc Network Security

A common assumption is that the majority of pervasive applications will communicate wirelessly and some devices will even be passive, relying on an external magnetic field (e.g., passive RFID) or energy harvesting. Providing security for communication with RFID tags or sensor networks will need radically different solutions from those used in existing applications such as the mobile phone world. Not always but often, power and therefore bandwidth is limited, the connection time is short, and computation is limited. Hence, there is need to develop new security protocols which are both lightweight with respect to communication and cryptographic computations.

Furthermore, the pervasive network security is a highly researched area. Secure routing as well as fair cooperation schemes that detect cheating nodes is widely discussed. Providing privacy in mobile networks is of interest to any user and crucial for the acceptance of pervasive networks. Finally, new and flexible solutions for establishing ad-hoc network infrastructures that replace traditional fixed infrastructures such as PKI are required.

Secure Operating Systems

As mentioned above, security controllers are used to securely store key data as well as to perform security functions. The operating system that provides

the basis for the controller's programs has to be secure though. The operating system controls the resources by providing input and output, store and restore functions, and provides the ability to run customized programs. The operating system has to be secure in the sense that a well defined interface is provided that does not leak any information. In particular there must be no security weaknesses and implementation flaws that allow recovery of secret data. Secure operating systems are required in a wide variety of applications. They are required for smart card 8-bit controllers, for ARM class 32-bit controllers as well as for PC class systems. Clearly, implementing a secure operating system is costly. A single implementation flaw endangers the security and might allow an exploit that reads out secret data. The more interface methods the operating system offers the higher is the probability of implementation flaws. Hence, where secure operating systems for 8-bit smart cards seem to be possible at high cost, secure operating systems for PCs are impossible to achieve with today's approaches. An alternative are secure micro-kernel architectures such as EMSCB (European Multilaterally Secure Computing Base) [4]. Secure micro-kernels consist of a few thousand lines of code only, and base their security on a security hardware anchor such as a TPM. The small size of the micro-kernel allows verification of its correct implementation. The secure operating system then provides a secure boot that detects any manipulation at the operating system, secure resource control of memory and storage and secure remote attestation. Here, an external party can verify whether the device runs in a trustworthy state. Currently, secure operating systems are developed for embedded devices such as cell phones, e.g., as part of the EMSCB project.

Future Application Areas

One of the exciting aspects of embedded security is that methods from the security community need to be applied in non-conventional application areas. "Non-conventional" is to be understood here in relation to "security": RFID-enabled goods are well known examples of security needs in unconventional settings. From a security research perspective, the challenge is (1) to understand the technical, economical and often also the social context of a given application area, in order to (2) develop security solutions which are appropriate and acceptable. Examples include digital rights management and trusted computing platforms for embedded applications such as home networks or car navigation systems. Automotive is one embedded area for which security solutions need to be developed. Applications as diverse as home networks, automotive, clothes, or hand-held devices will also need to be equipped with security.

5 Case Studies

In the following we describe two typical scenarios for pervasive security as case studies. We believe that automotive IT security is an excellent example for the typical problems of embedded security whereas ad-hoc network security is an inherent problem of basically all pervasive networked devices.

5.1 Case Study I: Automotive Security

Information technology — which we broadly define as being systems based on digital hardware and software — has gained central importance for many new automotive applications and services. The costs for software and electronics are estimated to approach the 50% margin in car manufacturing in the years to come. Perhaps more importantly, there are estimates that already today more than 90% of all vehicle innovations are centered around IT software and hardware. These applications are realized as embedded systems and range from simple control units to infotainment systems equipped with high-end processors whose computing power approaches that of current PCs. In high-end cars one can find around 80 processors that are connected by several separate buses and up to several hundred megabytes of embedded code.

Not suprisingly, many classical IT and software technologies are already well established within the automotive industry, for instance hardware-software co-design, software engineering, software component re-use, and software safety. However, one aspect of modern IT-systems has little attention in the context of automotive applications: IT-security. Security is concerned with protection against malicious manipulation of IT-systems. The difference between IT-safety and IT-security is depicted in Figure 3.

However, there are today niche applications in the automotive domain (e.g., immobilizers) that particularly rely on IT-security technologies. Nevertheless,

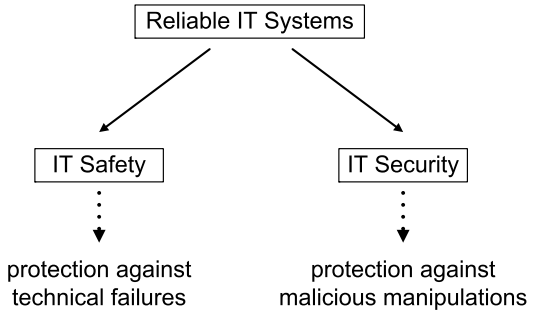

Figure 3: The relationship between IT-safety and IT-security.

the majority of software and hardware systems in current cars are *not* protected against manipulation. The reason being that past car IT-systems did not need security functions because there was only little incentive for malicious manipulation. Secondly, security tends to be an afterthought in any IT-system, because achieving the core function is often the main focus when designing a system. As can be seen for instance by the Internet development implementing IT-security afterwards, is a non-trivial undertaking.

The situation has changed dramatically. More and more vehicular systems need security functionality in order to protect the driver, the manufacturer and the component supplier. Secure software update of electronic control units (ECU), preventing chip tuning, preventing the unauthorized change of the mileage, or assembling non-original parts are only some examples. Future cars will become even more dependent on IT-security due to the following developments:

- An increasing number of ECUs will be reprogrammable and have to be protected.
- Vehicles will communicate with the environment in a wireless fashion that requires protected car to infrastructure communication.
- New business models (e.g., time-limited car functions or pay-per-use infotainment content) will be established but will only be successful if abuse can be prevented.
- An increasing number of legislative requirements (e.g., secure emergency call functions)
- Increasing networking of cars enables car to car communication that has to be protected against abuse and violation of privacy.
- Electronic anti-theft measures will go beyond current immobilizers, e.g., by protecting individual components.

IT-security will play an important role for several future automotive technologies and will even be an enabling technology for some future applications. The target platforms within cars which incorporate security functions are embedded systems, rather than classical PC-style computers. Some obvious differences in comparison to common PC-based environments are listed below.

- Embedded devices have small processors (often 8-bit or 16-bit micro-controllers) which are limited with respect to computational capabilities, memory, and power consumption. Hence, the usage of cryptographic primitives and protocols is limited.
- Embedded devices mostly have only limited possibilities and limited bandwidth for external communication. Hence, the extent and frequency of external communication, e.g., for internal updates, is limited.
- Attackers of embedded systems have often physical access to the target device itself.
- Embedded systems are often relatively cheap and cost sensitive because they often involve high-volume products. Thus, adding complex and costly security solutions is not acceptable.
- It is costly to establish the necessary organizational aspects for security products, e.g., one needs to adopt the production and life-cycle chain.

Hence, the technologies needed for securing vehicular applications mainly belong to the field of embedded security that differs from general IT-security. Detailed solution approaches in all related areas for the automotive area can be found in [5].

5.2 Case Study II: Ad-hoc Networks

In a pervasive computing world electronic devices such as cellular phones, televisions, and video recorders exist side by side with smart devices embedded in clothing, eyeglasses, buildings, and barcodes. Once these devices are equipped with a wireless radio, they form an extremely widespread network that connects many devices of our environ-

ment. This all-embracing network of mobile and static devices must be self-organized and should neither rely on a fixed infrastructure, nor a centralized administration as devices may be introduced to and removed from the network in a highly dynamic fashion. In fact, in the general case it is assumed that each node relies on its neighboring nodes to keep the network connected, e.g., each node routes data packets for its neighbors. Furthermore, each node might take advantage of services offered by other nodes. This type of network is called an ad-hoc network. It is particularly useful where a reliable fixed or mobile infrastructure is not available or too expensive. If the network consists of very small computing devices that are able to sense their environment we call such a network a sensor network. Here, all sensors collaborate in order to gather information based on their sensing capabilities. The sensor devices are very low-cost and thus extremely resource constrained. However, due to the low cost of these sensors, they can be deployed in the hundreds or thousands in a small area. Sensor networks are in most cases static, but mobile sensors are also conceivable. As for ad-hoc networks, the sensors run self-organized without any external guidance once they are deployed.

As ad-hoc and sensor networks become a growing part of our everyday life, they could become a threat if security is not considered carefully before deployment. For instance, consider a possible scenario of the future road traffic. There will be communication between cars, and between cars and roads. Cars will form a wireless network in an ad-hoc manner. They might communicate to each other in order to exchange information about free parking spaces or to warn about road threats. As an example, if there is an obstacle on the highway, a car could warn all following cars. Obviously, this information must be trustworthy and authenticated, and this process has to be done efficiently in real-time. All cars might have an electronic license plate embedded that identifies each car uniquely. If the electronic license plate broadcasts a unique identifier while the car is running, it is possible to identify the car and thus the driver of this car in case of a hit-and-run accident. Hence, it must not be possible to forge an electronic license plate in order to impersonate another driver and car, and it must not be possible to manipulate the electronic license plate of a car. There are several security goals in ad-hoc networks. The provision of authentication is a core requirement for secure and trustworthy communication in ad-hoc networks. The messages of the car warning all following drivers must be authenticated as well as the broadcast signal of the electronic license plate, and the electronic license plate must be inseparably bound to the car.

The security issues for ad-hoc and sensor networks are different than the ones for fixed traditional networks such as local area networks (LANs) and wide area networks (WANs). While the security requirements are the same, namely availability, confidentiality, integrity, authentication, and non-repudiation, their provision must be approached differently for ad-hoc and sensor networks. This is due to system constraints of mobile devices, frequent topology changes in the network, a missing fixed infrastructure, and the weak physical security of low-power devices. Furthermore, the main security targets differ in ad-hoc networks. For instance, secure routing and secure stimulation of cooperation are crucial issues in ad-hoc networks. The provision of authentication is required to implement secure protocols in ad-hoc networks such that authentication might provide the basis for a secure routing or stimulation scheme. Privacy is a main issue in ad-hoc networks as is availability. The later is hard to provide though. It seems the best one can do is to detect non-availability but not to avoid it – physical denial of service attacks such as channel jamming are always possible when a wireless channel is used.

We believe that ad-hoc networks will emerge in very simple topologies such as today's Bluetooth connection between two devices. By the time there will be more exciting applications such as vehicle-to-vehicle communication networks that will probably be established in Europe and USA by 2015. These will still use today's security architectures such as PKIs. The military will probably establish solutions that are based on more costly communication channels such as satellites but can be deployed extremely quickly worldwide. Consumer electronics is then to offer cost-efficient solutions using new innovative approaches.

6 Conclusions

We are already surrounded by embedded devices. A typical household already has dozens of them in cell phones, home entertainment, printers, household appliances, cars, etc. Once all these devices are equipped with a wireless communication channel,

we've arrived in the area of pervasive computing.

Pervasive computing will introduce new security threats, ranging from a loss of privacy, over reduced revenues, to bodily injuries. Some of the new security threats are well-known from conventional IT systems, whereas others are unique to the pervasiveness of the devices. At the same time, strong security in pervasive applications, e.g., fee-based feature activation in products, offers new opportunities for businesses and users.

Pervasive security is an emerging discipline and there is an active academic and industrial community working on strong security solutions.

References

[1] ANDERSON, R. *Security Engineering: A Guide to Building Dependable Distributed Systems*. John Wiley and Sons, 2001.

[2] EC. Joint Interpretation Library (JIL): Security Evaluation and Certification of Digital Tachographs, JIL Interpretation of the Security Certification according to Commission Regulation (EC) 1360/2002, Annex 1B, Version 1.12, June 2003.

[3] ESTRIN, D., GOVINDAN, R., AND HEIDEMANN, J. Embedding the Internet. *Communications of the ACM 43*, 5 (May 2000), 39–41.

[4] EUROPEAN MULTILATERALLY SECURE COMPUTING BASE. Towards Trustworthy Systems with Open Standards and Trusted Computing. URL `www.emscb.org`.

[5] LEMKE, K., PAAR, C., AND WOLF, M. *Embedded Security in Cars*. Springer-Verlag, 2006.

[6] TRUSTED COMPUTING GROUP. URL `www.trustedcomputinggroup.org`.

[7] TUYLS, P., SCHRIJEN, G. J., SKORIC, B., VAN GELOVEN, J., VERHAEGH, N., AND WOLTERS, R. Read-proof hardware from protective coatings. In *CHES* (2006), L. Goubin and M. Matsui, Eds., vol. 4249 of *Lecture Notes in Computer Science*, Springer, pp. 369–383.

[8] WEIMERSKIRCH, A., PAAR, C., AND WOLF, M. Cryptographic component identification: Enabler for secure vehicles. In *IEEE 62nd Vehicular Technology Conference* (Dallas, USA, 2005).

[9] WEISER, M. The computer for the 21st century. *Scientific American* (September 1991), 66–75.