eurobits
European Competence Center for IT Security

Pervasive Computing und Informationssicherheit

11. Symposium on Privacy and Security
Rüschlikon, 13. September 2006

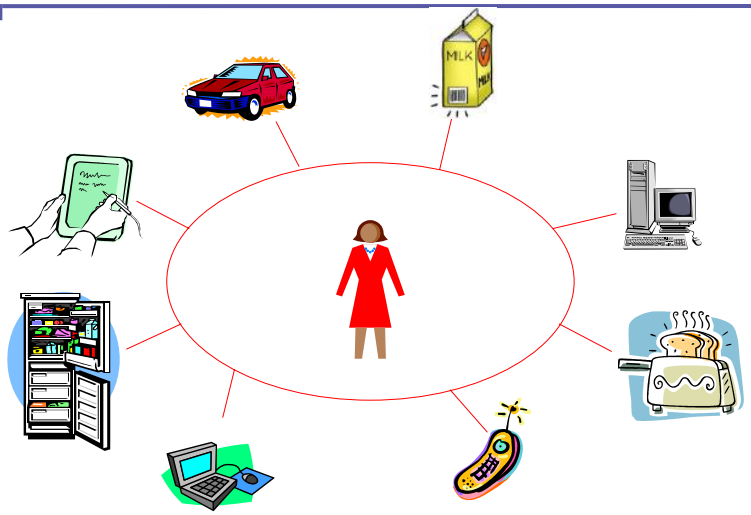
Prof. Christof Paar
European Competence Center for IT Security
www.crypto.rub.de

Contents

1. Pervasive Computing and Embedded Systems
2. Security in Pervasive Computing
3. Case Study
4. Technologies for Pervasive Security
5. Critical Infrastructures and Pervasive Computing
6. eurobits

Contents

Brave New Pervasive World



Pervasive Computing & Embedded Systems ?

- Pervasive Computing is **per definition** based on embedded systems.
- main difference: PvCo deals mainly with **networked** embedded devices
- Embedded systems perspective helps us to understand
  1. **importance of PvCo** and
  2. **security issues of PvCo**
- WRT security:
  embedded security ≈ pervasive computing security

What are Embedded Systems?


+

= Embedded System

- „A computer that doesn‘t look like a computer“, or
- „Processor hidden in a product“

Characteristics of Embedded Systems

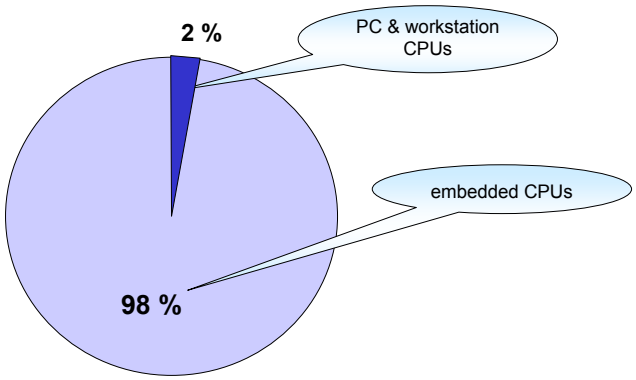
- Single purpose device



- Not general purpose like PC!
- Interacts with the world
- No (or primitive) user interface

Is this Really Important ?

CPU market 2000:



Remark: historically, most embedded applications did not have security issues

Contents

Security Concerns in Pervasive Computing

- Pervasive nature and **safety-critical** applications increase risk potential: *Hard disk crash vs. car crash*
- Often **wireless channels** ⇒ vulnerable
- **Contents protection** in many applications: iPod, XBox, navigation systems, …
- **Secure SW download**: cell phones, engine control, washing machine,…
- **Privacy issues:** geolocation, medical sensors, monitoring of home activities, etc.
- **Legislative requirements**: passports, road toll, data event recorders in machines, …
- **Stealing of services:** sensors, etc.
- …

General Thoughts about the Economics of Pervasive Security

- „One-user many-nodes“ paradigm (e.g. 10^2-10^3 processors per human)
- many new applications we don‘t know yet
- very high volume applications & very cost sensitive
- in some cases, the business model depends on security: iPod, XBox, software download, printer cartridges, …
- people won‘t be willing to pay for security per se
- but: people won‘t buy products without security

⇒ security must become “natural” part of product

⇒ security market is OEM-centered as opposed to end-user centered

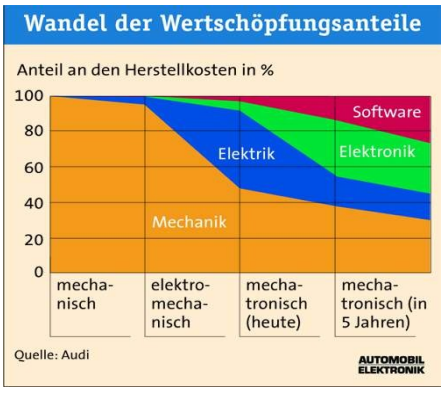
Contents

Examples of Security Needs in Pervasive Computing

- **Smart Cards** (phone, banking, …): Physical Attacks can cause huge financial damages
- **Pay-TV** (*Premiere* and such): Weakness in encryption create major financial damage
- **Portable entertainment** (*iPod* and such):
  - hacking of DRM damages content provider
  - usage as slurping device damages owner
- **Game consoles** (*XBox* and such): Copy Protection / DRM violations
- **Many, many other industries & applications** (household appliances, automation, medical, …)
- Let's look at **cars** as a case study…

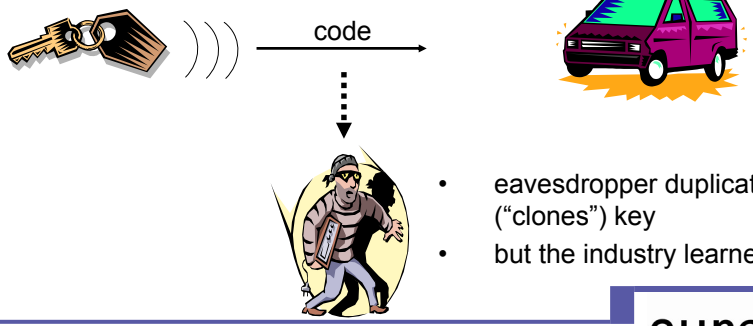
Basic Facts about IT and Cars

- up to 80 µP in high-end cars (talking about pervasive…)
- 50% of manufacturing costs related to electronics & software (5 year forecast)
- 90% of car innovations based on embedded IT
- > 100MB software in high-end cars



IT Security and Cars – Case Study

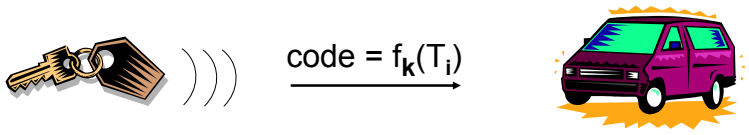
- early theft controls: unique code (password)



- eavesdropper duplicates ("clones") key
- but the industry learned…

IT Security and Cars – Case Study

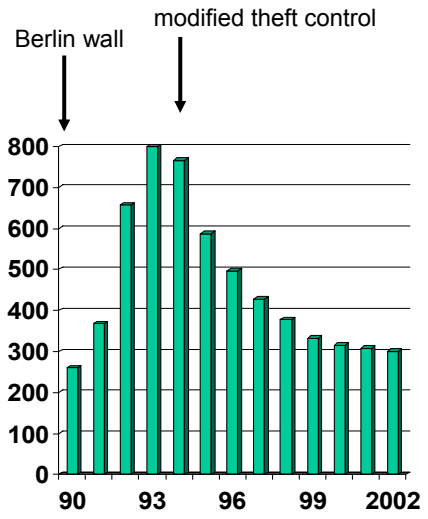
- advanced theft control: time-varying code



- $f_k()$ is a cryptographic one-way function

Fact: dramatic reduction in car thefts in Germany since mid-1990s

Car Theft in Germany: 1990-2002



Lesson learned: Sound security design can have major real-world benefits!

source: Gesamtverband Deutscher Versicherungsgesellschaften

IT Security and Cars – Case Study

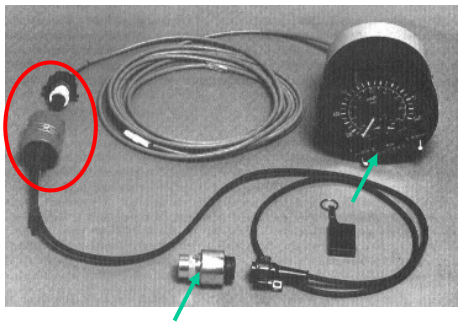
- "flashing" of embedded software

- customization of cars
- new products (SW tuning kits)
- new business models ("20 HP more for the weekend for €19.99")

- *But*: Unauthorized flashing poses major risk for safety and profits
- *Lessons learned*: Cryptographic protection (digital signatures of code) is enabling technology for new business models

IT Security and Cars – Case Study

- Truck driver control via digital tachograph: sensor & meter
- Sophisticated manipulation device allows cheating



from: R. Anderson "Security Engineering", Wiley, 2001

Lessons learned:

1. Never underestimate the attacker
2. Be aware of interaction economics ↔ security
3. Standard crypto mechanisms can prevent attack

Contents

Pervasive Security Technologies

1. Efficient cryptography
2. Side channel resistance
3. Tamper resistance
4. Trusted Computing
5. System security

Remark: Generally speaking, embedded security deals with the interaction of **security and engineering** issues.

Pervasive Security Technologies (1): Efficient Cryptography

Facts:

1. Cryptography is a key tool for security
2. Crypto algorithms tend to be computationally expensive

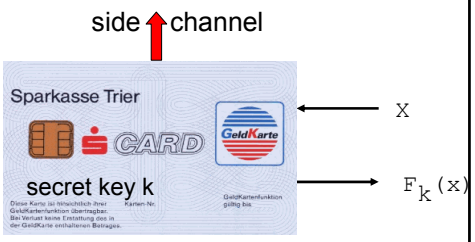
Embedded realities (in contrast to PC-ish applications)

1. Wide variety of CPUs: 32bit @ 400MHz … 8bit @ 3 MHz
2. Memory is often the most costly resource
3. Common problems:
   - Digital signature in less than 100msec with 25MHz CPU
   - AES with less than 2kB of ROM
   - Block ciphers with 1000 gates

Pervasive Security Technologies (2): Side Channel Resistance

Classical security assumption

1. Attacker knows x, $F_k(x)$
2. Attacker deduces k (Enigma break etc.)



Embedded realities

- Adversary has access to security module
- Physics often provide side channel to algorithm
  - Timing behavior
  - Power trace, EM trace
  - Cache access

Pervasive Security Technologies (3): Tamper Resistance

Underlying embedded problem:

- Adversary has access and full physical control over target device
- Security hinges on secure storage of key (+ other parameters)

Desirables

- Security design that is robust against tamper attacks

or

- prevent reading of internal & external Flash, ROM, …

Central (annoying) side condition: In almost all cases there is no money for secure hardware!

Pervasive Security Technologies (4): Trusted Computing

Many embedded applications require

1. TC for DRM applications
2. TC for high assurance applications
3. TPM for physical security reasons

Integration of TPM and secure OS (e.g., micro kernel) in embedded systems is highly attractive

Pervasive Security Technologies (5): System Design

In summary:

- \- Adversary has physical access (bad)
- \- Computation/memory/power constraints (bad)
- \- No money for secure hardware (bad)
- \+ Reverse engineering often difficult (good)
- \+ Security requirements greatly vary (good at times)

Hence, a careful system design, including

- threat analysis, and
- cost/security trade-offs

is of central importance.

Contents

Critical Infrastructure Perspective

Security in PvCo:

- Attacks against **single/few devices**: **User problem**
- Attacks against **domain of devices: Infrastructure problem**

Not so desirable scenarios:

- Emergency break signal sent to 1% of cars (i.e.,1.5 tons of steal all over German autobahns …)
- Internet-connected sensors in chemical/pharma/petro/… plants are attacked
- GPS signals are corrupted ⇒ logistics, agriculture etc. are affected

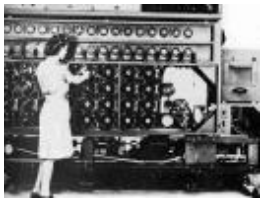
Contents

eurobits
Pervasive Computing Research & Applications

Upcoming Workshops

CHES – Cryptographic Hardware and Embedded Systems
October, 2006, Yokohama

escar – Embedded Security in Cars
November, 2006, Berlin



SHARCS (Special-purpose Hardware for Attacking Cryptographic Systems)
April, 2006, Köln

… and a related book

- Embedded technologies in general
- Security issues in cars