Security & Privacy Issues in Embedded Systems

Bart Jacobs

Institute for Computing and Information Sciences, Radboud University Nijmegen
www.cs.ru.nl/B.Jacobs

Bits&Chips Embedded Systems, Eindhoven, 12/11/2009

Outline

Who is this guy?

- Computer security professor at Nijmegen (0.0 also at Eindhoven)
- Focus on protection & abuse of ICT
- Involved in ***e-passport***, ***e-voting***, ***road pricing***, ***smart meters***, ***OV-chip*** 
- Occasional role in media
- Author of online book *De Menselijke Maat in ICT*, see www.cs.ru.nl/B.Jacobs/MM

Computer Security

- Regulating access to sensitive digital assets, such as:
  - military or industrial (stock/product) data
  - privacy-sensitive data: health, finance, communication, *etc.*
- Requires proper mix between technical, organisational, and legal measures
- Standard security goals:
  - Confidentiality: only authorised parties can read
  - Integrity: only authorised parties can write
  - Availability: services are up & running when needed
  - Non-repudiation: un-deniability / authenticated commitment

Security protocols I

- Protocols are recipes for achieving security goals, in presence of malicious attacker
- They run on top of network protocols: unintentional transmission errors are excluded.
- Mathematical basis in cryptography, notably:
  - encryption $\{m\}_K$ of message m with key K
  - hashing/fingerprinting $h(m)$ of message m
  - signing, esp. with public key crypto, *etc.*

Security protocols II

- Typical Alice-Bob one-way authentication protocol with shared key K_{AB}:

$$\begin{array}{lcll} A & \longrightarrow & B & : \text{“Hi, I'm Alice!”} \\ B & \longrightarrow & A & : \text{“Really? Tell me what is this:”}\ \{N\}_{K_{AB}} \\ A & \longrightarrow & B & : \text{“It's me indeed:”}\ N \end{array}$$

- N is random “nonce” (number used once).
- Security protocols are difficult. Roger Needham:

> security protocols are three-line programs that people still manage to get wrong

Difficulty lies in capturing what an attacker can do.

Computer security problems

- Security problems are structural (see *e.g.* BugTraq, `datalossdb.org`)
- Perverse market incentives:
  - fast market share most important (for consumer lock-in)
  - attitude is: “ship now, fix in verion 5”
- Lack of:
  - liability, wrt. software errors
  - transparancy, wrt. data management/loss
- Security mechanisms only increase price, but add nothing.
  - on the contrary: security mechanisms restrict functionality
  - best, but non-existent attitude of manufacturer:

> My product is very secure; it can do almost nothing!

Users are:

- familiar with physical security

  When you ask a stranger in the street for the key to his house, you probably will not get it

- ignorant / gullible in IT-security

  When you ask a stranger to type his credit card number (plus ccv-code) in a webform to check if it has been stolen, you may very well succeed.

Comparison with car security

- In the 1960s,
  - road casualties were much higher
  - many car security techniques were already available
  - they were not demanded, by drivers or authorities
- Now many security mechanisms are built-in.
  - Car sales person does not advise you to get your brakes separately, on the way home
  - However, for computer security it still works this way.

Security gets embedded

Bruce Schneier:

- Security becomes embedded in the IT-products we buy
- Trend: security companies are now being bought by other companies
  - his own (Counterpane) by British Telecom
  - many more examples
- Relevant for embedded systems community!

IT-architecture

- Mitchell Kapor, co-founder of the *Electronic Freedom Foundation*:

> Architecture is Politics

- IT-professionals are architects,
  - not just of the digital world, . . .
  - but also of the social world
- Information is power, and information architecture determines power relations between people.

Road pricing example

- Foreseen in NL, since many years
- Cars get a special box, called **OBU**, for “on-board unit”
- . . . which can at least:
  - determine its own position, via GPS or Galileo
  - communicate with backoffice, via GSM, GPRS, Wifi, . . .
  - calculate & store data
- Tariff map needed for fee calculation on basis of “trajectory parts”

Big Question

- Where to store trajectory information?
  - in the back office of the authorities
    (who use it to calculate bills)
  - in the vehicle, i.e. in the OBU
    (so OBU contains tariff map to calculate its own bill)
- This is an architectural decision about information flow
- But also about division of power in society
  (balance citizen – state)
- Underlying theme:

  Centralised versus decentralised architectures

Architectures: general characteristics

- Centralised
  - Data outside user control: privacy depends heavily on organisational measures
  - Easier abuse (e.g. by insiders) or loss (accidentally, or via hacking)
  - Convenience for user
  - Easier maintance & policy enforcement
  - Informational control leads to societal control (profiling/datamining)
- Decentralised
  - Privacy-friendly, in-context storage of data
  - More responsibility/activity on user side required
  - Fraud resistance possibly more difficult

Who decides on architecture?

- General rule: those in power tend not to choose architectures that decrease their control. On the contrary!
- There are very few incentives to choose for decentralised architectures. Possibly from:
  - Consumer/privacy interest groups
  - Government, if it ever remembers its constitutional task to protect citizen privacy.

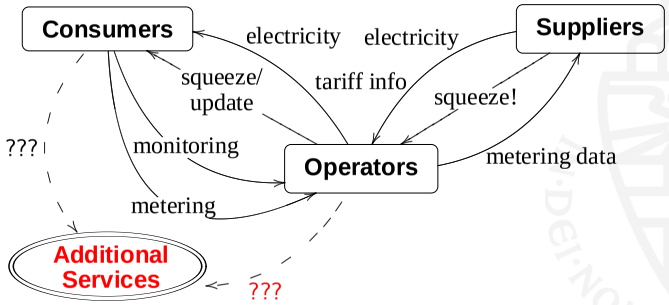
Metering context

- Traditional utility metering in protected hardware, with decentralised (local, in house) storage of data.
- New paradigm: smart meters, with remote meter reading and centralised storage of metering data
- Naive (political) motivation: consumption reduction if people can see their usage via the web.
  - Nonsensical detour: why not give local access?
    (and do bi-monthly read-out)
- Motivations from the utility sector:
  - cost reduction (remote reading/maintenance)
  - better grid management (granularity level unclear)
  - user profiling & additional commercial services

Timeline

1. **Summer 2008**
   New utility law adopted by Parliament (Second Chamber)
   - making smart meters compulsory,
   - meter recording *every 15 minutes*
     (daily read-out, with opt-in for every 15 min.)
   - remote squeeze/disconnect possible
   - clients can also supply energy (solar/wind/. . .)
2. **Spring 2009**
   Senate (First Chamber) removes compulsary character
   - serious privacy/security concerns
   - positive impact: sector finally wakes-up
3. **Currently**
   Awaiting update of law (*novelle*)
   - obligation to accept meter will disappear; details pending

Flow schema essentials



Sensitive issues

How much/often metering / monitoring / control / services info

Privacy concerns: Pamphlets (in Dutch only. . .)

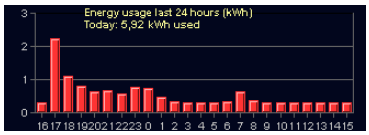
SLIMME METERS

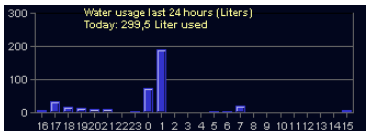
MIJN BROERTJE GAAT LANGER DOUCHEN IN DE HOOP DAT DE CONTROLEURS DENKEN DAT HIJ EEN VRIENDINNETJE HEEFT

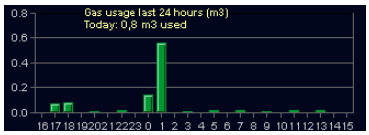
Loesje

Postbus 1045
6801 BA Arnhem www.loesje.nl

Privacy concerns: example readings (bwired.nl)







Privacy concerns & personal security

With 15 minute & daily meter reading . . .

- Operator/producer employees see when I'm at home or not
- Useful info for burglars
  (can use blackmail/bribery/infiltration/hacking to get such info)
- Why am I exposed to this new vulnerability?
- Privacy is important for personal security!

With remote squeeze option . . .

- DOS risk, see *Delta Lloyd Hackman* Video
- National security risk, exploitable by blackmailers/terrorists/. . .
- Why do we introduce these vulnerabilities?

Challenges for embedded systems sector

- Security must be built into devices / components
  - E.g. encapsulated power measurement with encrypted output signal
  - Key management issue of its own
- Explicit security policies & designs needed
- Architectures become even more complicated
- Security & functionality don't mix well
- Enable software/security update? If so how?
  - Embedded/distributed hardware hard to replace/update
  - See Mifare/OV-chip debacle

Main points

- Security will become more important in embedded systems.
- Architecture is politics.
- Think big!
  - Many societal (privacy) issues involved
  - . . . that can make or break your project.
- Slides available at `www.cs.ru.nl/~bart/TALKS`