Embedded Systems Specification and Design

David Kendall

Northumbria University

Introduction

- Security of embedded systems
- Security goals
- Importance of secure protocols
- Modelling and analysis of protocol security

Security of embedded systems: example

Implantable Cardiac Defibrillator (ICD)

- *Pacing*
  Periodically send a small electrical stimulus to the heart
- *Defibrillation*
  Occasionally send a larger electrical charge to restore normal heart rhythm

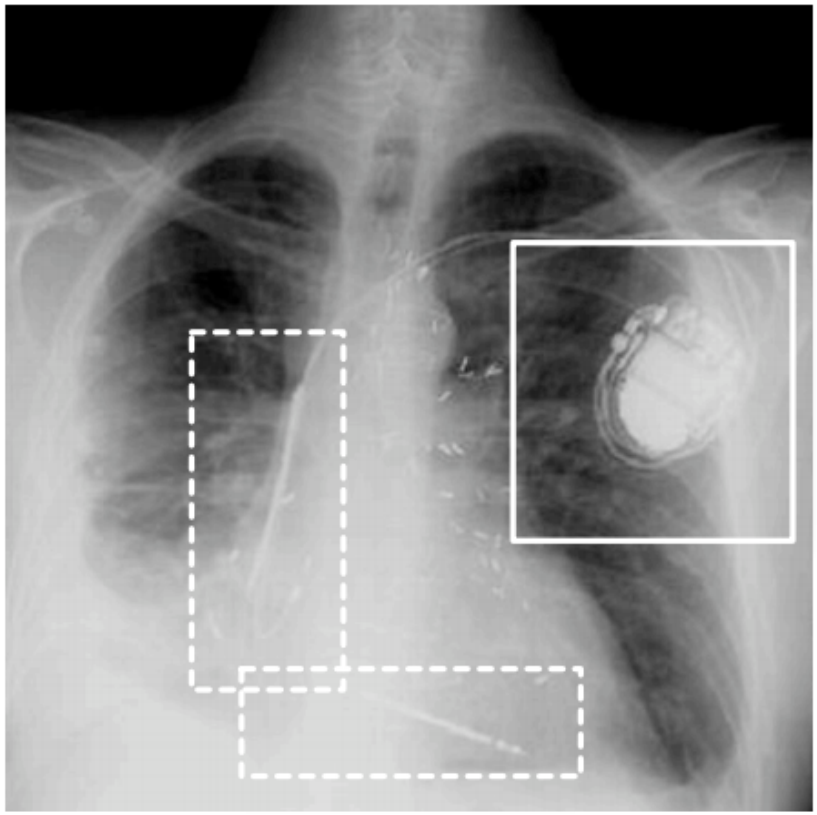

Fig. 1. Chest xray image of an implanted ICD (top right, near shoulder, solid outline) and electrical leads connected to heart chambers (center of rib cage, dotted outline).

ICD programmer

The “programmer” is a device intended to be used to:

- perform diagnostics
- read and write private (patient) data
- adjust therapy settings on the ICD.

Programmer communicates with ICD wirelessly.

- typically 175 kHz short-range communication

Security of the ICD device

- Halperin, D., Heydt-Benjamin, T.S., Ransford, B., Clark, S.S., Defend, B., Morgan, W., Fu, K., Kohno, T. and Maisel, W.H., *Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses*, IEEE Symposium on Security and Privacy, 129-142, May, 2008
- Considered attacks on ICD security by three classes of attackers:
  - Attacker possessing an ICD programmer
  - Attacker who simply eavesdrops on communications between an ICD and the programmer, using commodity software radio
  - Attacker who eavesdrops as well as generates arbitrary RF traffic to the ICD, possibly spoofing an ICD programmer.
- Demonstrated that successful attacks are possible under all three classes

Experimental results of attacks on ICD

	Commercial programmer	Software radio eavesdropper	Software radio programmer	Primary risk
Determine whether patient has an ICD	✔	✔	✔	Privacy
Determine what kind of ICD patient has	✔	✔	✔	Privacy
Determine ID (serial #) of ICD	✔	✔	✔	Privacy
Obtain private telemetry data from ICD	✔	✔	✔	Privacy
Obtain private information about patient history	✔	✔	✔	Privacy
Determine identity (name, etc.) of patient	✔	✔	✔	Privacy
Change device settings	✔		✔	Integrity
Change or disable therapies	✔		✔	Integrity
Deliver command shock	✔		✔	Integrity

TABLE I
RESULTS OF EXPERIMENTAL ATTACKS. A CHECK MARK INDICATES A SUCCESSFUL IN VITRO ATTACK.

Security goals

- Secrecy
  - Confidential data is not leaked by the system to those not authorised to have it
- Authentication of origin
  - Outputs appearing to come from the system are actually generated by the system
- Integrity
  - System cannot be modified by attacker
- Access and availability
  - System is always able to provide its service (denial of service not possible)

What can we learn from this example?

- Why attacks succeed
  - Messages sent in plaintext
  - No attempt to confirm that messages are received from / sent to an authorised ICD programmer
- Power matters
  - Some attacks simply cause ICD to keep transmitting - depletes its battery
  - Ideally defences should use zero power
- Secure protocols are needed

Importance of secure protocols

- Secure protocols needed to
  - Allow agents to authenticate each other
  - Establish session keys for confidential communication
  - Ensure integrity of data and services
  - Prevent unauthorised access to data and services
- Other components in a secure system include
  - Good cryptographic algorithms
  - Systems security measures for access control
- Security protocols are
  - Often apparently simple
  - Often flawed and vulnerable to unexpected attacks
- Careful design and analysis of security protocols is important
- Following slides are based on, and should be read in conjunction with, chapter 0 of:
  - Ryan, P., Schneider, S., Goldsmith, M. and Lowe, G., *Modelling and analysis of security protocols: the CSP approach*, Addison Wesley, 2000 [pdf]

Notation: Messages

- The steps in a protocol can be represented using the following notation:

$$n. \quad A \rightarrow B : data$$

- This is intended to mean that in the nth step of the protocol agent A dispatches a message containing *data* to agent B.
- N_A is used to represent a *nonce* generated by agent A.
- Compound terms are formed by *concatenation* and *encryption*:
  - X, Y denotes the value X followed by the value Y
  - $\{data\}_K$ denotes the value *data* encrypted using the key K
- Example

$$1. \quad A \rightarrow B : \{N_A, A\}_{K_B}$$

indicates that in the first step of the protocol, agent A dispatches a message to agent B. The message comprises a nonce generated by A, followed by A's name, and is encrypted using B's public key.

Notation: Keys

- Let A and B be communicating agents.
- K_{AB} represents a symmetric key, shared by A and B
- K_A represents an asymmetric public key for A
- $\overline{K}_A$ represents an asymmetric private key for A
- Given K_A it is infeasible to compute $\overline{K}_A$
- $\{\{M\}_{K_A}\}_{\overline{K}_A} = M = \{\{M\}_{\overline{K}_A}\}_{K_A}$

See Chapter 5 in Ross Anderson's book if you need more explanation about symmetric and asymmetric keys, and public key and shared key cryptography.
Anderson, R. *Security Engineering: A guide to building dependable distributed systems* (2nd edition), John Wiley, 2008 [local copy of Chapter 5]

Example: Needham-Schroeder Secret Key Protocol (NSSK)

- The protocol

1. $A \rightarrow S : \{A, B, N_A\}$
2. $S \rightarrow A : \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{SB}}\}_{K_{SA}}$
3. $A \rightarrow B : \{K_{AB}, A\}_{K_{SB}}$
4. $B \rightarrow A : \{N_B\}_{K_{AB}}$
5. $A \rightarrow B : \{N_B - 1\}_{K_{AB}}$

- Notes
  - This is a protocol for authenticated key exchange
  - S represents the key server and K_{SA} and K_{SB} are long term secret keys shared between the server and agents A and B, respectively.
  - K_{AB} is the key generated by the key server for agents A and B to use for communication between each other.
  - At the completion of the protocol, A and B share the secret key K_{AB} and each knows that the other knows the key!

Man-in-the-middle attack (outsider)

- Consider this protocol:

$$\begin{array}{ll} 1. & A \rightarrow B : \{X\}_{K_A} \\ 2. & B \rightarrow A : \{\{X\}_{K_A}\}_{K_B} \\ 3. & A \rightarrow B : \{X\}_{K_B} \end{array}$$

- Seems like a neat protocol to allow A to send a secret message to B without needing to know B's public key
- Relies on the property that $\{\{X\}_{K_A}\}_{K_B} = \{\{X\}_{K_B}\}_{K_A}$
- But can be attacked by an intruder, Y, who can intercept and insert messages:

$$\begin{array}{ll} 1. & A \rightarrow Y(B) : \{X\}_{K_A} \\ 2. & Y(B) \rightarrow A : \{\{X\}_{K_A}\}_{K_Y} \\ 3. & A \rightarrow Y(B) : \{X\}_{K_Y} \end{array}$$

- Problem arises because of lack of authentication: A does not know who she is talking to.

Man-in-the-middle attack (insider)

- Consider this protocol (Needham-Schroeder Public Key Protocol):

$$
\begin{aligned}
1.&\quad A \rightarrow B : \{A, N_A\}_{K_B} \\
2.&\quad B \rightarrow A : \{N_A, N_B\}_{K_A} \\
3.&\quad A \rightarrow B : \{N_B\}_{K_B}
\end{aligned}
$$

- A and B think this protocol ensures that
  - They have been interacting with each other
  - They agree on the values of N_A and N_B
  - No one else knows N_A and N_B
- But consider this attack by insider Y

$$
\begin{aligned}
1(a).&\quad A \rightarrow Y : \{A, N_A\}_{K_Y} \\
1(b).&\quad Y(A) \rightarrow B : \{A, N_A\}_{K_B} \\
2(a).&\quad B \rightarrow Y(A) : \{N_A, N_B\}_{K_A} \\
2(b).&\quad Y \rightarrow A : \{N_A, N_B\}_{K_A} \\
3(a).&\quad A \rightarrow Y : \{N_B\}_{K_Y} \\
3(b).&\quad Y(A) \rightarrow B : \{N_B\}_{K_B}
\end{aligned}
$$

Man-in-the-middle attack (insider)

- This very famous attack was discovered by Gavin Lowe, using a model-checker, and published in 1996, 18 years after the original publication of the protocol.
- It relies on a dishonest insider, Y, interleaving two runs of the protocol.
- At the end of the attack, A believes that she shares knowledge of N_A and N_B exclusively with Y. While B believes that he shares knowledge of N_A and N_B exclusively with A. Neither belief is correct.
- Notice that the attack does not involve breaking any cryptography.
- It is possible just by the nature of the protocol.

Modelling and analysis of protocol security

- Lowe's use of a model-checker to discover the attack on NSPK protocol stimulated much research in this area.
- Now many protocols have been analysed with general-purpose and custom-built model-checkers.
- The SPIN model-checker can be used for such analysis
- For example, see the analysis of the NSPK protocol, using SPIN, by Maggi and Sisto:
  - Maggi, P. and Sisto, R., *Using SPIN to verify security properties of cryptographic protocols*, Proceedings of SPIN Workshop 2002, LNCS 2318, 187-204, Springer Verlag, 2002 [pdf]

Modelling and analysis of NSPK protocol with SPIN (Maggi and Sisto 2002)

- Abstract all data values, e.g all nonces generated by A represented by `mtype = Na`.
- Cryptography assumed to be perfect: only way to decrypt a message is by knowing the key.
- Use a different channel for each different message structure, e.g. `chan ca = [0] of mtype, mtype, mtype, mtype;`, for a message such as $\{N_A, N_B\}_{K_A}$
- Assume all communication goes through the intruder to model the capability of the intruder to intercept, lose, forward, duplicate and insert messages between any agents.

Modelling and analysis of NSPK protocol with SPIN (Maggi and Sisto 2002)

- Model the behaviour of each of the agents using a Promela `proctype` for each, in the usual way.
- Model a highly non-deterministic intruder using a Promela `proctype`, i.e. the intruder should be able to send and receive any type of message to/from any of the other agents at any time.
- Use the verifier to check LTL properties expressing security goals and examine any counter-examples to identify possible attacks.

Acknowledgements

- Koopman, P., *Embedded system security*, IEEE Computer, 37(7): 95-97, July 2004 [pdf]
- Halperin, D., Heydt-Benjamin, T.S., Ransford, B., Clark, S.S., Defend, B., Morgan, W., Fu, K., Kohno, T. and Maisel, W.H., *Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses*, IEEE Symposium on Security and Privacy, 129-142, May, 2008 [pdf]
- Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, and Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage, *Experimental Security Analysis of a Modern Automobile*, IEEE Symposium on Security and Privacy 2010, 447-462, 2010 [pdf]
- Maggi, P. and Sisto, R., *Using SPIN to verify security properties of cryptographic protocols*, Proceedings of SPIN Workshop 2002, LNCS 2318, 187-204, Springer Verlag, 2002 [pdf]