Security in Embedded Systems

Mohammad Tehranipoor

ECE 4095: Hardware Security & Trust
University of Connecticut
ECE Department

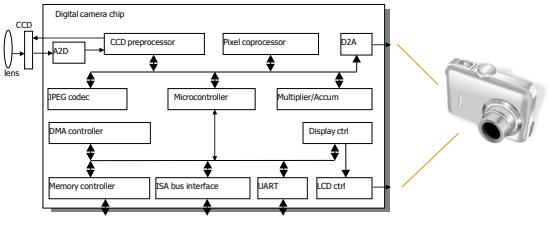
What is an Embedded System?

It is a microprocessor that is used as a component in a device and is designed for a specific control function within that device

Examples of Embedded Systems

- Embedded systems are used in many consumer electronics such as:
  - Cell Phones
  - PDAs
  - MP3 Players
  - Household Appliances
  - Digital Cameras
  - MANY MORE!!

Example of Digital Camera



Characteristics of Embedded Systems

- Constrained
  - Low cost
  - Fast
  - Hardware
  - Software
- Size
  - Small
- Reliability
- Reactive and real-time

Attacks on Embedded Systems

Attacks on Embedded Systems

- Types of attacks on embedded systems
  - Embedded Software Attacks
  - Physical Attacks
  - Logical Attacks
  - Timing Analysis
  - Power Analysis
  - Fault Induction
  - Electromagnetic Analysis

Embedded Software Attacks

- The software in an embedded system is a source of security vulnerability.
- Three factors which make security risks a challenge in software
  - Complexity
  - Extensibility
  - Connectivity

Embedded Software Attacks

- Complexity
  - Software is complicated
  - More lines of code
    - Increases possibility of bugs and security vulnerabilities
  - Unsafe programming languages being used
    - C and C++ are most common

Embedded Software Attacks

- Extensibility
  - Modern software systems are designed to be extended
    - Updates
    - Extensions
  - Loadable device drivers and modules

Embedded Software Attacks

- Connectivity
  - Embedded systems are being connected to the Internet
  - Possible for small failures to occur leading to security breaches
  - Attacker no longer needs physically access to system
    - Use a series of automated attacks

Physical Attacks

- Eavesdropping
  - Use of probes to eavesdrop on inter-component communications
- Micro-probing
  - Use normal communication interface and abuse security vulnerabilities

Physical Attacks

- De-packaging is done by using fuming acid

Physical Attacks

- Once de-packaged, the next step is layout reconstruction
  - During reconstruction internals of chip and be inferred
- Micro-probing can be used to observe values on buses

Logical Attacks

- Send messages to a device and observe the response
- Goal is to trick device into revealing the key
- Often exploit design flaws

Timing Analysis

- Keys can be determined by analyzing small variations in the time required to perform cryptographic computations

Power Analysis

- The operating current drawn by a hardware is correlated to computations it is performing
- In most ICs, logic gates and losses due to the parasitic capacitance are the major contributors to power consumption
- Two types of power analysis attacks
  - Single power analysis (SPA)
  - Differential power analysis (DPA)

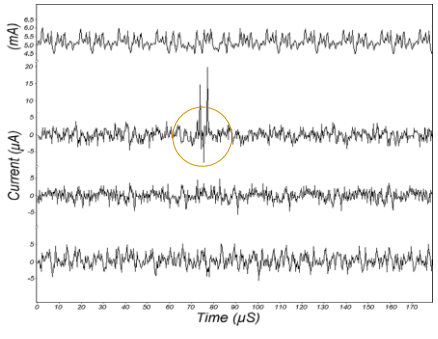
Power Analysis

Incorrect

Correct

Incorrect

Incorrect



Fault Induction

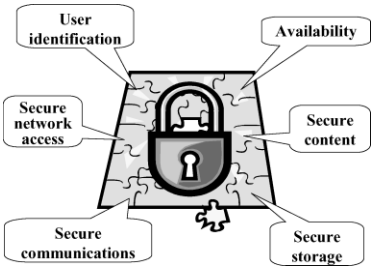
- Security doesn't just depend on software
  - Security can be compromised if hardware fails to make proper computations
- RSA implementation can be compromised if there is any computation errors

Electromagnetic Analysis

- Have been documented since the 80's
- Measures electromagnetic radiation emitted by device to reveal sensitive information
- Successful deployment would require knowledge of chip layout
- Two classes of EMA attacks
  - Simple EMA (SEMA)
  - Differential EMA (DEMA)

Security Requirements and Design Challenges

Common Security Requirements



Secure Embedded System Design Challenges

- Processing Gap
- Battery Gap
- Flexibility
- Tamper Resistance
- Assurance Gap
- Cost

Processing Gap

- Some embedded systems are not capable of keeping up with computational demands of security processing
  - Increase data rates and complexity of security protocols
- Processing gap is obvious in systems which need process very high data rates
  - Network routers, firewalls, and web servers

Battery Gap

- Battery capacity increases at an average of 5% to 8% per year.
- Security processing energy requirements outpace the increase in battery capacitance
  - This leads to a battery gap

Flexibility

- Embedded systems often required to execute multiple and diverse security protocols
- Need to be able to support
  - Multiple security objectives
  - Interoperability in different environments
  - Security processing in different layers of the network protocol stack

Tamper Resistance

- Attacks due to malicious software
  - Most common:
    - Viruses
    - Trojan Horses
  - Can exploit OS vulnerabilities
  - Disrupt normal functioning

Assurance Gap

- Truly reliable systems are much more difficult to build
- Reliable systems must be able to handle wide range of situations
- Secure systems should be able to operate despite attacks
- Increase in complexity makes it more difficult to realize if something was overlooked

Cost

- Fundamental factor that influences the security architecture
- Increase in security leads to an increase in cost
- Designer's responsibility to balance security and cost

Levels of Security

- Level 1
  - Requires minimal physical protection
- Level 2
  - Requires the addition of tamper-evident mechanisms
    - Seal or enclosure
- Level 3
  - Stronger detection and response mechanisms
- Level 4
  - Mandates environmental failure protection and testing

Countermeasures

Embedded Software Security

- Best approach is to think about security early on in the software design life cycle (SDLC)

Embedded Software Security

- Software security should be applied at various levels
  - Requirements level
  - Design and Architecture level
  - Code level

Physical Attacks

- Hard to use because of the chip size, the smaller the better
- Expensive, relative to other types of attacks

Logical Attacks

- Logical attack countermeasures are designed with the following considerations:
  - Ensure privacy and integrity of sensitive code and data
  - Determine that it is safe from a security standpoint to execute a program
  - Identify and remove software bugs and design flaws

Timing Analysis

- Obvious countermeasures do not work
  - Quantizing the total time
  - Adding random delays
- Message blinding can be used with RSA
- Make all computations the exact same time
  - Only a few programs operate in exactly the same constant time
- Other public-key cryptosystems

Power Analysis

- Run other circuits simultaneously
  - Does not prevent attack but attacker needs more samples
- Effective countermeasures are mathematically rigorous and non-intuitive
- Effective countermeasures remain expensive and challenging

Fault Induction

- RSA implementations can check their answers by performing a public-key operation
- Many cryptographic devices include an assortment of glitch sensors
  - Detect conditions likely to cause computation errors

Questions?

Work Cited

- ACM Transactions on Embedded Computing Systems, Vol. 3, No. 3, August 2004, Pages 461–491
- M. Tehranipoor and C. Wang, Introduction to Hardware Security and Trust, Springer, 2011
- Security as a New Dimension in Embedded System Design by Paul Kocher, Ruby Lee, Gary McGraw, Anand Raghunathan, and Srivaths Ravi