Understanding Hardware Security

Black Hat Japan 2004 Briefings

Joe Grand
Grand Idea Studio, Inc.
joe@grandideastudio.com
Goals

• Learn the concepts of designing secure hardware
• Become familiar with types of attacks and attackers
General Security Concepts

• Nothing is ever 100% secure
  – Given enough time, resources, and motivation, an attacker can break any system

• Secure your product against a specific threat
  – What needs to be protected
  – Why it is being protected
  – Who you are protecting against (define the enemy)
And this house is even more secure! The front door is four feet thick and made of solid titanium...
Security During Product Development

- Establish a security policy as the "foundation" for design
- Treat security as an integral part of your product's development
- Minimize the elements you need to secure
- Reduce risk to an acceptable level
  - Elimination of all risk is not cost-effective
Security During Product Development 2

- Implement layered security
- Do not implement unnecessary security mechanisms
  - Each mechanism should support a defined goal
- Costs of a successful attack should outweigh potential rewards
Types of Attack

- **Insider Attack**
  - Significant percentage of breaches
  - Ex.: Run-on fraud, disgruntled employees
- **Lunchtime Attack**
  - Take place during a small window of opportunity
  - Ex.: During a lunch or coffee break
- **Focused Attack**
  - Time, money, and resources not an issue
Types of Attackers

- **Clever Outsiders**
  - Intelligent, but have limited knowledge of the product
  - Usually take advantage of a known weakness
  - Ex.: Curious kids, college students

- **Knowledgeable Insiders**
  - Substantial specialized technical experience
  - Highly sophisticated tools and instruments
  - Ex.: Professional engineers
Types of Attackers 2

- Funded Organizations
  - Specialists backed by great funding resources
  - In-depth analysis, sophisticated attacks, most advanced analysis tools
  - Ex.: Government, organized crime
Accessing the Product

- **Purchase**
  - Attacker buys the product from a retail store

- **Evaluation**
  - Attacker rents or borrows the product

- **Active**
  - Product is in operation, not owned by attacker

- **Remote Access**
  - No physical access to product
  - Attacks launched remotely
Threat Vectors

- **Interception (or Eavesdropping)**
  - Gain access to information without opening the product
- **Interruption (or Fault Generation)**
  - Preventing the product from functioning normally
- **Modification**
  - Invasive tampering of the product
- **Fabrication**
  - Creating counterfeit data in a product
Goals of an Attack

- **Competition (or Cloning)**
  - Specific theft to gain marketplace advantage
- **Theft-of-Service**
  - Obtaining a service for free that normally costs money
- **User Authentication (or Spoofing)**
  - Forging a user's identity to gain system access
- **Privilege Escalation (or Feature Unlocking)**
  - Gaining increased command of a system or unlocking hidden/undocumented features
Anti-Tamper Mechanisms

- Primary area of physical security for embedded systems
- Attempts to prevent unauthorized physical or electronic tampering against the product
- Most effectively used in layers
- Possibly bypassed with knowledge of method
  - Attackers may intentionally destroy a device to determine its security mechanisms
Anti-Tamper Mechanisms 2

• Tamper Resistance
  – Specialized materials used to make tampering difficult
  – Ex.: One-way screws, epoxy encapsulation

• Tamper Evidence
  – Ensure that there is visible evidence left behind by tampering
  – Only successful if a process is in place to check for deformity
  – Ex.: Passive detectors (seals, tapes, glues), special enclosure finishes ("bleeding paint")
Anti-Tamper Mechanisms 3

- Tamper Detection
  - Enable the hardware device to be aware of tampering
  - Switches: Detect the opening of a device or breach of security boundary
  - Sensors: Detect an operational or environmental change (ex.: temperature, voltage, radiation)
  - Circuitry: Detect a puncture, break, or attempted modification of a defined security envelope (ex.: nichrome wire, W.L. Gore's D3 enclosure)
Anti-Tamper Mechanisms 4

- Tamper Response
  - Countermeasures taken upon the detection of tampering
  - Ex.: Erase memory, shutdown/disable device, enable logging

- *Physical Security Devices for Computer Subsystems* [1] provides comprehensive attacks and countermeasures
Enclosure & Mechanical

- Product Housing
- External Interfaces
Product Housing

- Attack goal of opening the product is to get access to internal circuitry
- Usually as easy as loosening some screws or prying open the device
- Designers should prevent easy access to product internals
Product Housing 2
Product Housing 3

- Sealed or molded housing
  - Use a high-melting point glue
  - Use ultrasonic welding to create a one-piece outer shell
  - Will require destruction of device to open it
  - Consider service issues (if a legitimate user can open device, so can attacker)
Product Housing 4

- Security bits and one-way screws
  - Prevents housing from being easily opened
  - Ex.: 3.8mm, 4.5mm, and Tri-Wing screw for Nintendo and Sega cartridges/consoles
  - Beware: Attackers can purchase many of these special bits online
External Interfaces

- Usually connects a product to the outside world
  - Manufacturing tests, field programming/upgrading, peripheral connections
  - Ex.: RS232, USB, Firewire, Ethernet, JTAG (IEEE 1149.1)
External Interfaces 2

- Will likely be probed or monitored by attacker
- Only publicly known information should be passed
- Encrypt secret or critical components
  - If they must be sent at all...
  - Ex.: Palm OS system password decoding attack [2]
External Interfaces 3

- Don't just hide the interface
  - Will easily be discovered by an attacker
  - Ex.: Proprietary connector types, hidden access doors or holes, stickers
- Protect against malformed, bad packets
  - Intentionally sent by attacker to cause fault
External Interfaces 4

- Physically remove all diagnostic, debug, and backdoor interfaces from production units
  - Even if they are undocumented
  - Difficult to do
  - Do not just cut traces or remove resistors (which could be repaired by an attacker)
  - Ex.: Intel NetStructure crypto accelerator administrator access [3], Palm OS debug mode [4]
External Interfaces 5

• Field programmability
  – Only allow new versions of firmware to be loaded into product (so attacker can not make use of old, known security flaws)
  – Do not release firmware on your Web site (could be disassembled and analyzed by attacker)
  – If you must, use code signing (DSA) or hashes (SHA-1, MD5) to verify integrity
  – Even better, encrypt firmware images
Circuit Board

- Physical Access to Components
- EMI/ESD/RF Interference
- PCB Design and Routing
- Memory and Programmable Logic
- Power Supply
- Cryptographic Processors and Algorithms
Access to Components

- Giving an attacker easy access to components aids in reverse engineering of the product
- Make sensitive components difficult to access
  - Ex.: Microprocessor, ROM, RAM, ASICs, FPGAs
- Remove identifiers and markings from ICs
  - Use stainless steel brush, small sander, micro-bead blast, laser etcher, or third-party
  - Easy for attacker to find data sheets online
Access to Components 2

• Use advanced package types
  – Difficult to probe using standard tools
  – Ex.: BGA, Chip-on-Board (COB), Chip-in-Board (CIB)

• Use proprietary or customized ICs
Access to Components 3

- Cover critical components with epoxy or urethane encapsulation
  - Usually used to protect circuitry from moisture, dust, mold, corrosion, or arcing
  - Difficult, but not impossible, to remove with chemicals or tools
EMI/ESD/RF Interference

- All devices generate electromagnetic interference (EMI)
- Can be monitored and used by attacker to determine secret information
  - Ex.: Data on a computer monitor [5], cryptographic key from a smartcard [6]
- Devices may also be susceptible to RF or electrostatic discharge (ESD)
  - Intentionally injected to cause failure
EMI/ESD/RF Interference 2

• Install EMI shielding
  – Decrease emissions and increase immunity
  – Ex.: Coatings, tapes, sprays, housings
  – Be aware of changes in thermal characteristics that shielding may introduce (heating)

• Prevent against ESD on exposed I/O lines
  – Clamping diodes or Transient Voltage Suppressors
  – Ex.: Keypads, buttons, switches, display

• Keep circuit traces as short as possible
EMI/ESD/RF Interference 3

- Use properly designed power and ground planes
- Power supply circuitry as physically close as possible to power input
- Remove unnecessary test points
  - Use filled pad as opposed to through-hole, if necessary
- Unused I/O pins and modules should be disabled or set to fixed state
PCB Design and Routing

- Confuse trace paths to prevent easy reverse engineering
  - Hide critical traces on inner board layers
  - Be aware of data being transferred across exposed and/or accessible address, data, and control buses

- Use buried vias whenever possible
  - Connects between two or more inner layers but no outer layer
  - Cannot be seen from either side of the board
PCB Design and Routing 2

• Ex.: Tap board used to intercept data transfer over Xbox's HyperTransport bus [7]
Memory and Programmable Logic

- Most memory is insecure
  - Can be read with standard device programmer
  - Serial EEPROMs can be read in-circuit, usually SPI or I²C bus (ex.: USB authentication token [8])

- Difficult to securely and totally erase data from RAM and non-volatile memory [9]
  - Remnants may exist and be retrievable from devices long after power is removed
Memory and Programmable Logic 2

- SRAM-based FPGAs most vulnerable to attack
  - Must load configuration from external memory
  - Bit stream can be monitored to retrieve data
- Protect against I/O scan attacks
  - Attacker cycles through all possible combinations of inputs to determine outputs
  - Use unused pins to detect probing
Memory and Programmable Logic 3

- Security fuses and boot-block protection
  - Enabled for "write-once" access to a memory area or to prevent full read back
  - Implement if available
  - Ex.: PIC16C84 attack in which security bit is removed by increasing VCC during repeated write accesses [10]
Memory and Programmable Logic 4

- Limit the amount of time that critical data is stored in the same region of memory
  - Periodically flip the stored bits
- If using state machine, ensure all conditions and defaults are covered
- Add digital "watermarks"
  - Features or attributes in design that can be uniquely identified as being rightfully yours
Memory and Programmable Logic 5

- Chip Decapping and Die Analysis attacks
  - Attacker can visually recreate contents or modify die (Ex.: to obtain crypto key or remove security bit)
  - Tools: Chip Decappers, Scanning Electron Microscope, Voltage Contrast Microscopy, Focused Ion Beam
Power Supply

- Define minimum and maximum operating limits
  - Ex.: Comparators, watchdogs, supervisory circuits
- Do not rely on end user to supply a voltage within recommended operating conditions
  - Implement linear regulator or DC-DC converter
Power Supply 2

- Simple Power Analysis (SPA)
  - Attacker directly observes power consumption
  - Varies based on microprocessor operation
  - Easy to identify intensive functions (ex.: cryptographic)

- Differential Power Analysis (DPA)
  - Advanced mathematical methods to determine secret information on a device
Cryptographic Processors and Algorithms

- Strength of cryptography relies on secrecy of key, not the algorithm
  - Do not create your own crypto algorithms
- It is not safe to assume that large key size will guarantee security
- If algorithm implemented improperly, can be broken or bypassed by attacker
  - Test implementations in laboratory first!
Cryptographic Processors and Algorithms 2

- Move cryptographic processes out of firmware and into FPGA
  - Harder to probe than ROM devices
  - Increased performance (more efficient)
- Or, use secure cryptographic coprocessor
  - Self-contained, hardware tamper response, authentication, general-purpose processor
  - Ex.: Philips VMS747, IBM 4758
In Conclusion

- Determine what to protect, why you are protecting it, and who you are protecting against
  - No one solution fits everyone
- Do not release product with a plan to implement security later
  - It usually never happens...
- Nothing is 100% secure
In Conclusion 2

• Be aware of latest attack methodologies & trends
• As design is in progress, allocate time to analyze and break product
• Learn from mistakes
  – Study history and previous attacks
References


Thank You!

ジョセフ B. グランド, 電気工学理学士
社長・最高経営責任者
株式会社 グランド アイディア スタジオ

Joe Grand
Grand Idea Studio, Inc.
http://www.grandideastudio.com
joe@grandideastudio.com