Protecting Consumer Electronics

Benjamin Jun | Cryptography Research, Inc. | 8 April 2008 | HT1-108

RSACONFERENCE2008

RSACONFERENCE2008

Protecting consumer electronics

- Consumer electronics threats
- Learning from open-source

- Probing tools
- Information leakage
- Fault induction
- Boot attacks

Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

Consumer electronics threats

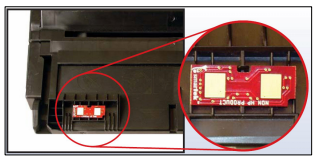
- Obtain “unauthorized” access to asset
  - Pirated digital content

RSACONFERENCE2008

Consumer electronics threats

- Obtain “unauthorized” access to asset
  - Pirated digital content
- Interact with “unauthorized” device/service
  - Use of non-authentic peripheral

“RAZR V3” Battery



Unauthorized toner chip

CRYPTOGRAPHY RESEARCH
Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

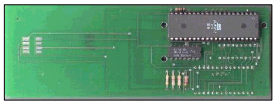
Consumer electronics threats

- Obtain “unauthorized” access to asset
  - Pirated digital content
- Interact with “unauthorized” device/service
  - Use of non-authentic peripheral
- Make device execute “unauthorized” code
  - Enable unauthorized features, linux

RSACONFERENCE2008

Consumer electronics threats

- Obtain “unauthorized” access to asset
  - Pirated digital content
- Interact with “unauthorized” device/service
  - Use of non-authentic peripheral
- Make device execute “unauthorized” code
  - Enable unauthorized features, linux
- Clone a device
  - Unauthorized copy



Pay TV smartcard emulator

Starting car w/ DST simulator

Source: “Security Analysis of a Cryptographically-Enabled RFID Device”, Bono, Green, Stubblefield, Juels, Rubin, Szydlo, 1/29/2005

CRYPTOGRAPHY RESEARCH
Copyright 2008 Cryptography Research, Inc.

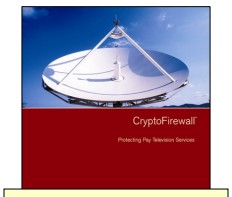
RSACONFERENCE2008

Who am I? What do I do?

- Cryptography Research, Inc.
  - Solve fraud, piracy problems
  - R&D emphasis on applied security issues
- Design and evaluation services
- License security technologies

CRYPTOGRAPHY RESEARCH

DPA:
Tamper Resistance



CryptoFirewall:
Pay-TV Security

SPDC / BD+:
Renewable Security
for Digital Content

RSACONFERENCE2008

Try this at home!

Learn from open-source embedded projects

CRYPTOGRAPHY RESEARCH
Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

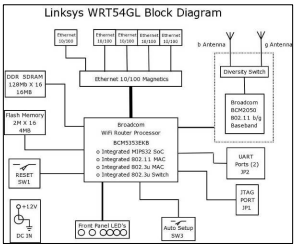
Looking to open-source embedded projects

- Not (really) a security compromise…
  - Learn about techniques to gain control of and use HW
  - Reverse engineering is a key part of project efforts
- Good engineering lessons
  - Fast way to explore deployed embedded environments
  - Good fun! Good return on $ + time.

RSACONFERENCE2008

Linksys WRT54GL

- $60 home wireless router
  - Linksys released open source SW (2003)
- Why modify?
  - Modify operating parameters (xmit power, etc.)
  - Add more firewall setttings, mesh networking, QoS, use metering, etc.
  - Linux on cheap, low power platform
- Thriving hobbyist community
  - DD-WRT (www.dd-wrt.com)
  - OpenWRT (www.openwrt.org)
  - Note: buy the WRT54GL



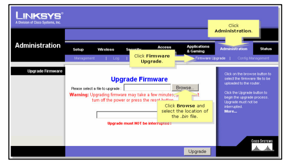
Linksys (Cisco Systems)
The Consolidated Hacking Guide For The Linksys WRT54GL, "ByteEnable", www.linuxelectrons.com

CRYPTOGRAPHY RESEARCH
Copyright 2008 Cryptography Research, Inc.

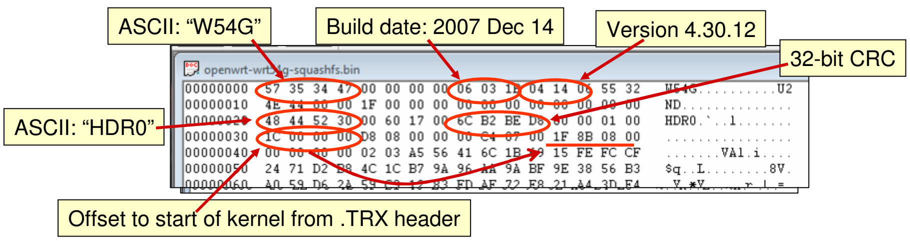
RSA CONFERENCE 2008

FW reflash of Linksys WRT54GL

- Example: Upgrade to OpenWRT firmware
  - Download OpenWRT whiterussian rc5, squash fs
    http://downloads.openwrt.org/whiterussian/rc5/bin/openwrt-wrt54g-squashfs.bin
  - Use (unsecured) firmware update mechanism
  - Router reboots, use www interface to set root pw
  - SSH in!



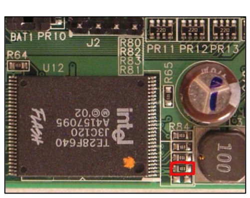
Stock Linksys FW: "4.30.12 1/10/2008" **OpenWRT FW: whiterussian rc5, squash FS**

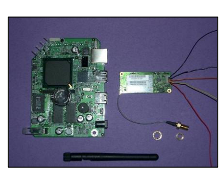


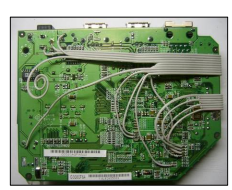
RSA CONFERENCE 2008

Try this: NSLU2

- Linksys NSLU2
  - Ethernet SMB (Windows) fileserver for USB disks ($70)
- NSLU-linux project (www.nslu2-linux.org)
  - Unslung distribution keeps stock Linksys functionality
  - Disk makes it easy to experiment with
- HW mods
  - Overclocking, serial port, RAM upgrade, debug ports, etc.







http://www.nslu2-linux.org/wiki/HowTo/AddAThirtyFourPinUniversalConnector http://www.nslu2-linux.org/wiki/HowTo/OverClockTheSlug David Hicks, http://www.nslu2-linux.org/wiki/HowTo/AddInternalWireless

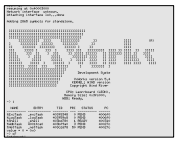
Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

No surprise to embedded developers…

- Nearly all products based on commodity cores, reference designs
  - WRT54GL: Broadcom BCM5352E
  - NSLU2: Intel IXP420 XScale (ARM)
- Small number of widely used embedded development environments
  - Linux, uC Linux, VxWorks
  - Good development, cross-compilation, debugging tools
- Hardware is incrementally "free"
  - Differentiating features governed by software
  - Software modifications have interesting results!

µClinux



RSACONFERENCE2008

Why follow these projects?

- Infrastructure type projects
  - "Boot linux on ____ platform."
  - Understand debugging, porting tools
- Patch type projects
  - "Get in, get system to do ____, get out."
  - SW reverse engineering
- Hardware extension projects
  - HW reverse engineering

Cryptography Research

Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

Probing

RSACONFERENCE2008

Probing goals

- Goal: Gather enough information for analysis
  - Reverse engineering is about making hypotheses
  - … and finding ways to confirm them !
- Static analysis
  - Use disassemblers and static analysis tools
- Probe system as it responds to stimuli
  - In normal processing
  - In abnormal processing

Cryptography Research

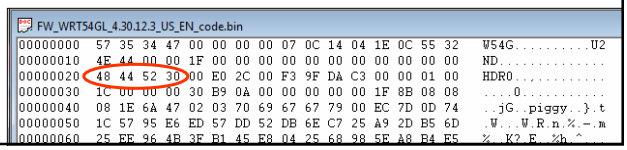
Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

Static reading of flash / ROM

- For discrete parts, this is easy…
  - Use ROM reader
  - Tap data bus, drive address bus
- On-die ROM
  - Get software to read it!
- With a code dump…
  - Recognize processor type
  - Disassemble, probe for implementation weakness
  - Search for keys (high entropy)

Conitec GALEP-4 progammer/reader



RSACONFERENCE2008

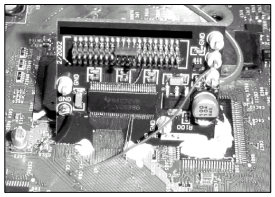
Probing tools

IC test clip (PLCC)

Emulation Technology Part 5281

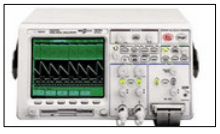
SCSI Bus Analyzer

Cryptography Research



Custom FPGA Logic Analyzer

Bunnie Huang, http://www.xenatera.com/bunnie/proj/anatak/xboxmod.html



GPIB Oscilloscope + Logic Analyzer

Agilent Technologies, 1672G data page

Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

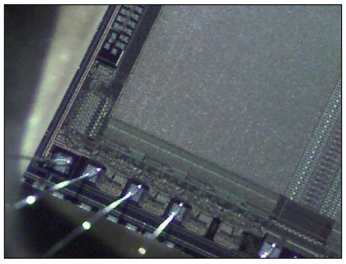
Countermeaures to probing

- Bury vias
  - "Production" mods harder, one-time probing still doable…
- Use "encrypted" memory
  - Use global or unique key
  - If keys managed by SW, this is essential
  - … but difficult to rely on with many copies of identical data
- Hardcode keys + algorithms + processing
  - Integrity + privacy benefits
  - Good: In SoC ROM
  - Better: In HW gates

RSACONFERENCE2008

Die imaging: ROM

- Imaging on-die ROM
  - Optical microscope / FIB: Use automated tool to image ROM
  - Active probing: Tap data bus, drive address bus
  - Passive probing: Tap bus lines
- Responses
  - Shield metal layers
  - "Encrypted" memory
  - HW-based obfuscation



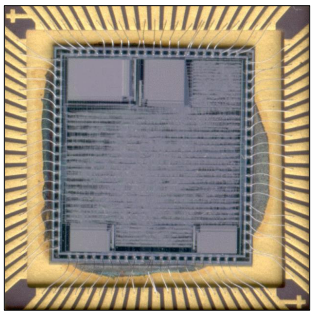
Memory image: Hector Vega
FIB: University of Cambridge Department of Materials Science Device Materials Group

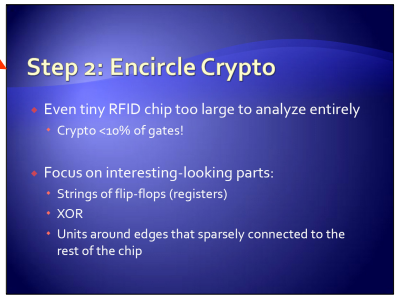
CRYPTOGRAPHY RESEARCH
Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

Die imaging: Gates

- Imaging attacks on gates
  - Automated tool to image + recover netlist
  - Easier if crypto area small, design has good structure
  - One reference: Nohl, Starbug, Plotz, *Mifare Security*, CCC 2007
- Responses
  - Shield layers
  - Camouflage libraries
  - If imaging not 100% accurate…



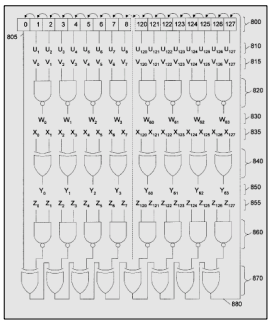


http://www.ece.cmu.edu/~koopman/stack_computers/binar_chip.jpg
Nohl, Starbug, Plotz, Mifare Security, CCC 2007

RSACONFERENCE2008

Example: Entropic Array

- State-of-the-art reverse engineering techniques are error-prone
  - Attacks involve human interpretation + correction
  - For normal circuits, attackers do fine w/buggy output
  - Example: AES with a few errors is obviously AES
- Goal: make imperfect reverse engineering results useless
  - Exact circuit required to perform crypto correctly
    - Attackers can't infer where they made mistakes
  - EA designed to be difficult to interpret
    - "Grown" from design rules and random seed
    - Internal entropy / lack of structure



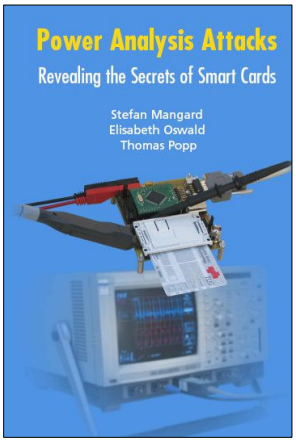
Example EA

US patent 6,640,305

CRYPTOGRAPHY RESEARCH
Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

Information Leakage Attacks

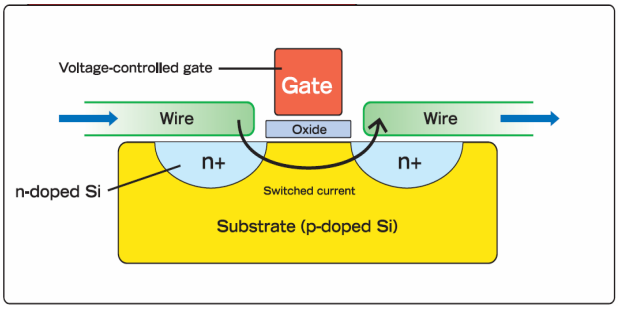


Mangard, Oswald, Popp
www.dpabook.org

RSACONFERENCE2008

Information leakage

Integrated circuits consume power as they operate.

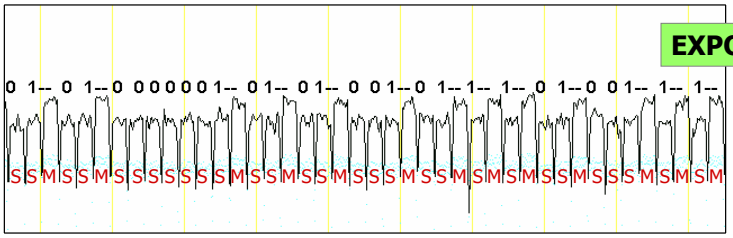


Typical MOS Transistor

Cryptography Research
Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

Simple Power Analysis (SPA)




```
Algorithm MODULAR_EXPONENTIATION (exponent e, base b, modulus n)
      Let A = 1
      Let X = b mod n
      Let k be the number of bits in e.  Then,
      For i = k downto 0, do
            A = (A * A) mod n
            Let e_i be the i'th bit of exponent e        ; bit 0 is LSB
            if (e_i == 1), do
                  A = (A * X) mod n
            Done
      Done
      Return A
End Algorithm
```

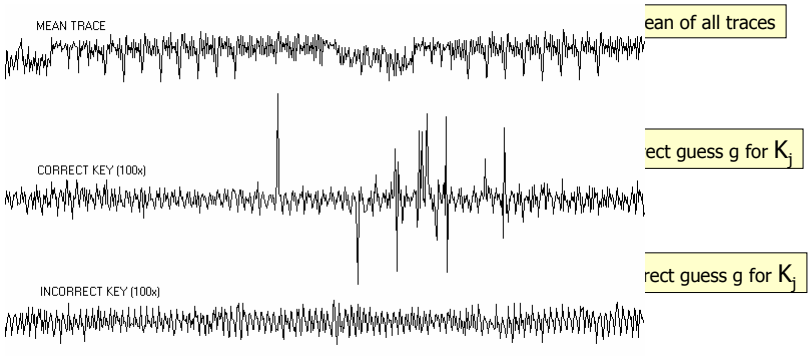
LOOP

SQUARE

CONDITIONAL MULT

RSACONFERENCE2008

Differential Power Analysis (DPA)



- **To read more about DPA**
  - **www.cryptography.com/dpa**
  - **www.dpabook.org**

Cryptography Research

Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

Hybrid attacks

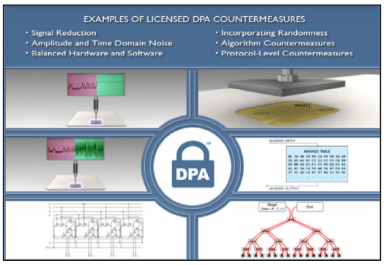
- Use power consumption as an output channel

  - Code insertion
    - Buffer overflow loads + executes target code
  - For each bit of data
    - Perform instruction A or instruction B 256 times
    - Dump memory via SPA activity

RSACONFERENCE2008

Defenses against power analysis

- Categories
  - Obfuscation
  - Leak Reduction
  - Balanced HW / SW
  - Amplitude & Temporal Noise
  - Incorporating Randomness
  - Protocol Level CM
- Certifications / Requirements
  - FIPS 140-3 draft
  - Common Criteria
  - CAC, E-Passport, HSPD-12
- Complicated!
  - Naïve CM can make things worse
  - Device evaluation is important



Cryptography Research

Use of these countermeasures requires a license from Cryptography Research and is protected under US patents 6,278,783, 6,298,442, 6,304,658, 6,327,661, 6,510,518, 6,539,092, 6,654,884 and other patents issued and pending in the US and worldwide.

CRYPTOGRAPHY RESEARCH

Copyright 2008 Cryptography Research, Inc.

RSA CONFERENCE 2008

Fault Induction Attacks

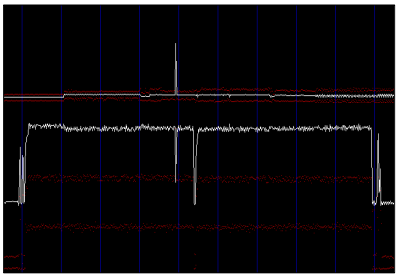
Van de Graff (left)

US Smithsonian Institution

RSA CONFERENCE 2008

Why induce faults?

- Security code (generally) assumes that platform is reliable
  - What if it isn't?
- Why do this?
  - Induce a computational fault
  - Change value being read/stored
  - Break into/out of an execution loop
  - Change program counter (PC)
- Surprises
  - More repeatable than you think…



Glitch impulse & power trace during successfully-glitched RSA CRT

Cryptography Research, Inc.

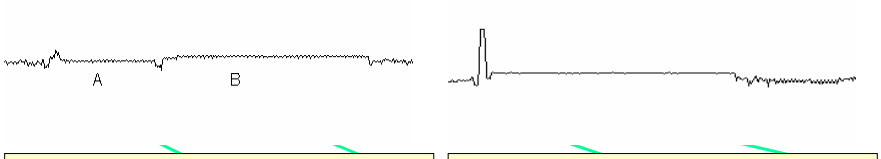
Cryptography Research

Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

Tearing nonvolatile memory writes

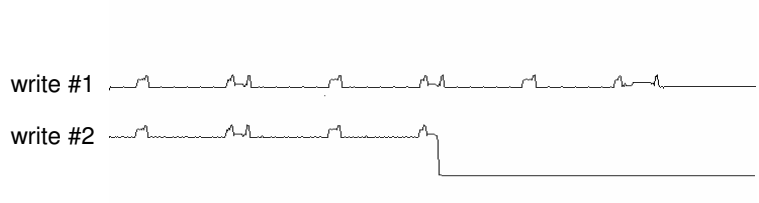
- Typical EEPROM write/clear cycles



Device 1: Bit Clearing Phase (A) and Bit Setting Phase (B)

Device 2: Bit Setting Phase (A) and Bit Clearing Phase (B)

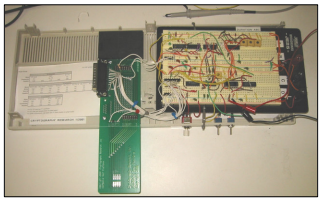
- Tearing example

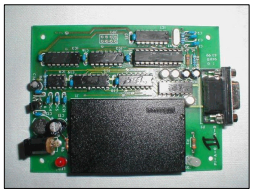


Cryptography Research, Inc.

RSACONFERENCE2008

Glitching tools





Cryptography Research

Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

Glitching examples

- Glitch instruction to extend output loop
  - Example: DJNZ instruction

```
; This routine is used to output a 4-byte result
; DPTR points to first byte of data
result_out:  MOV R0, 3              ; set loop counter
result_loop: CALL #putch            ; call UART routine
             INC DPTR               ; increment pointer
             DJNZ R0, #result_loop  ; conditional loop
             RET
```

- Glitching program counter
  - "Serpent's tail": NOP sled followed by target code

RSACONFERENCE2008

Example glitch defenses (SW)

- Check your work!
  - Verify private key computations
- Preincrement error flags

Before

```
// check password
result = secureMemCmp(passwordLen, userPin, userInput);
if (result)
  nvmErrorCount = 0;
else if (++nvmErrorCount > errorMax)
  lockOut();
return (result);
```

After

```
if (nvmErrorCount++ > errorMax)
  lockOut();
result = secureMemCmp(passwordLen, userPin, userInput);
if (!result)
  nvmErrorCount = 0;
return (result);
```

CRYPTOGRAPHY RESEARCH
Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

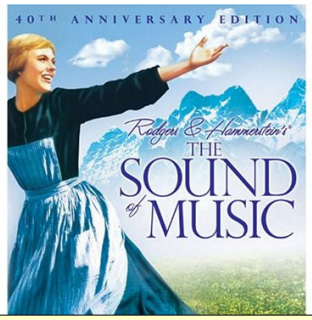
Example glitch defenses (HW)

Canaries used to detect CO gas in mines
(Hollinger Mine, Ontario, 1928)

- Environmental sensors
  - Monitor clock, voltage
  - Not available in all silicon processes
- Independent votes
  - Independent modules vote before enabling output, unlock
- Canary logic (CryptoFirewall)
  - Cumulative hash of system control state on every clock
  - Hardware design element designed to be on critical path
  - Example: Gate output unless canary logic permits operation

RSACONFERENCE2008

Boot Attacks



"Let's start at the very beginning
A very good place to start
When you read you begin with A-B-C
When you sing you begin with do-re-mi"
– Maria

Rodgers, Hammerstein, The Sound of Music (1965)

CRYPTOGRAPHY RESEARCH
Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

Taking over a device

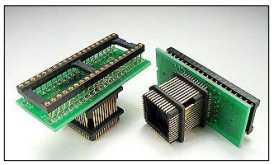
- Step 1: Understand how device operates
- Step 2: Get (malicious) code on the device
- Step 3: Get control of program counter (PC)

RSACONFERENCE2008

Getting to execution – ROM/flash replacement

- Socketed flash/ROM

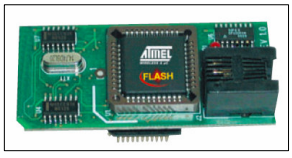
BIOS socket (X-Box)
Bunnie Huang, xenatera.com



DIP to PLCC adaptor
Logical systems, www.logicalsys.com

- ROM / flash emulator

ROM emulator
Transtronics Pocket ROMulator



Flash emulator
MSC AT51Flash-Emulator

Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

Getting to execution – debug tools

- JTAG analyzer
  - Full debug interface
  - Boot code replacement
  - In-target flash/ROM reading + programming

EJTAG In-target analyzer

First Silicon Solutions, ISA-MIPS

ARM JTAG debugger interface

Olimex Ltd., ARM-USB-OCD

RSACONFERENCE2008

Getting to execution – taking over control

- Take over control directly
  - Buffer overflow, glitch to executable code, protocol errors, …
  - Suspend/resume with memory image change
- Bypass authentication in bootloader
  - Bypass authentication process (CRC, signature)
    - Common problem: bad authentication protocol
  - Ex: “Savegame” buffer overflow in “007” Xbox game (2003)
    - Modifies public key, makes modulus divisible by 3
    - Signature forgery, clean start sequence!

```
BE00000080          mov esi, 80000000
AD                  lodsd                       ; 0x00000032
81EE03000000        sub esi, 00000003
3DBD1B4BA4          cmp eax, A44B1BBD
75F2                jne 00000032                ; scan RAM for public key
8176FFD68BD72D xor dword ptr [esi-0], 2DD78BD6  ; modify last 4 bytes of public key
```

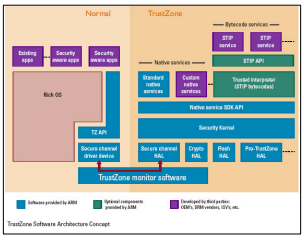
http://xbox-linux.sourceforge.net/docs/007analysis.html

Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

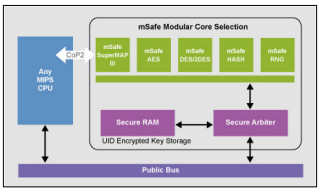
HW tools for boot protection

- ARM TrustZone
  - Lightweight hardware extensions for security
    - S-Bit added to CP15 register
    - SMI (Secure monitor interrupt) instruction
  - Only privileged OS can issue SMI
  - Non-secure processes have limited privileges



ARM TrustZone Architecture

- MIPS Safe-SOC
  - Peripheral on MIPS coprocessor bus
  - Crypto accelerator with access to memory space



MIPS Safe-SOC Architecture

- Cautions
  - Monitor and secure kernel should be small
  - Watch out for complexity!

ARM, MIPS product literature

RSACONFERENCE2008

Example: Bootloader with code signatures

- Boot ROM with code authentication
  - Reset vector jumps to boot ROM
  - Verify RSA cert (hash of PK can be hard-coded)
  - Hash and verify payload, FAIL on error
  - Decrypt + decompress payload to RAM
  - Clear all other RAM
  - Jump to payload

- HW requirements
  - Force jump to internal boot ROM on reset
  - JTAG and other execution vectors should be off on boot
  - Development/test: skip bootloader if fuse unblown

CRYPTOGRAPHY RESEARCH
Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

Example: Continuous code checking

- Continuous verification
  - Boot ROM starts watchdog timer
  - Background code scanning process
    - Steps through program RAM & computes hash
    - Restart timer if code check passes
  - Code scanning is interruptible
    - Halt & restart completely if interrupted
    - Can work on smaller chunks of code
      - Verify one branch of a hash tree at a time

- HW requirements
  - Watchdog that can only be started by code in boot ROM area
  - Requires protected RAM or dirty bit that is set on interrupt

RSACONFERENCE2008

Bootloader challenges

- Challenges
  - Enabling development environment
  - Support for powerdown / sleep modes
  - MMU behavior as system boots

- "Cat and mouse"
  - Example: smartcard code hashing and emulators

Checkpoint Charlie (1986)

Cryptography Research

Copyright 2008 Cryptography Research, Inc.

RSACONFERENCE2008

Where do we go from here?

RSACONFERENCE2008

Conclusion

- Most consumer electronics devices:
  - Are in physical possession by attacker
  - Have flat memory architectures
  - Are used offline
- Developer tools == attacker tools
  - Most systems use standard platforms + tools
  - Good debuggers and debugging interfaces
- HW and SW can't do this alone!
  - SW can enable functionality, handle updates
  - HW can enforce small set of rigid rules

www.spirit-o-gro.com

Cryptography Research

Copyright 2008 Cryptography Research, Inc.

RSA CONFERENCE 2008

Contact Information

For more information, or to discuss how Cryptography Research can help with a security problem:

Benjamin Jun
ben@cryptography.com
415.397.0123
www.cryptography.com

We're hiring!

If you are technically strong and want to work on challenging crypto and security problems, please send a resume!

© 1998-2008 Cryptography Research, Inc. (CRI) Portions may be protected under issued and/or pending US and/or international patents. A separate license from CRI is required for the CryptoFirewall™, DPA Countermeasures, and Self-Protecting Digital Content™. All trademarks are the property of their respective owners. The information contained in this presentation is provided for illustrative purposes only, and is provided without any guarantee or warranty whatsoever, and does not necessarily represent official opinions of CRI or its partners.

Copyright 2008 Cryptography Research, Inc.