The embedded security challenge: Protecting bits at rest

Patrick Schaumont
schaum@vt.edu

Acknowledgements: Eric Simpson, Pengyuan Yu

Secure Embedded Systems Group
ECE Department

Secret bits-at-rest

Hi-Res Digitized Signature

Car Unlock Code

AACS Keys (Blu-Ray DVD)
Classic security

- Protect bits 'in flight' using crypto

Embedded security

- Need more than crypto to protect bits 'at-rest'
Keeping secrets in software doesn't work

- Example - Fairplay encryption scheme of Apple

![Diagram of Fairplay encryption scheme]

- April 2007: Steve Jobs announces iTunes will sell DRM-free songs

Keeping secrets in hardware doesn't work

- Example: Aladdin eToken (2001) stores PIN in plaintext

![Image of Aladdin eToken]

- April 2007: Secustick ('self-destruct thumbdrive') is broken by means of a trivial hack

![Image of Secustick]

[tweakers.net/reviews/683/1]
We need a secure design methodology

- Methodology - series of steps that can be learned and repeated.

- **Zero-risk** security does not exist
  - *Zero-power* design does not exist either

- **Low-risk** security can be achieved
  - *Low-power* operation can be achieved

Objective of secure design methodology:
minimize spatial and temporal footprint of secret bits in embedded systems

The starting point: Root of Trust

- A secret in a box by itself is useless
  (useless like a key without a matching door-lock)

- So a secure system contains at least two parts

- **security policy**
  - authentication
  - integrity
  - confidentiality
  - non-repudiation
  - availability

- **root-of-trust**
  - The part that always works as expected (or so you think!)
  - Secret bits are inside of root-of-trust
Characteristics of attacks

- An 'attack' is an interface into the root-of-trust around the security policy

**abstraction level**
- logical
- side-channel
- physical

**attacker proximity**
- off-line
- connected
- close-range
- physical

**active/passive**
- invasive/non-invasive
  - side-channel
  - fault-based
  - spoof

Example: Cache timing attack

- Execution time dependency due to cache conflicts
- Software Solution: constant-time crypto (hard)
Example: Cache timing attack

- Execution time dependency due to cache conflicts
- Software Solution: constant-time crypto (hard)

Moving Sbox into Hardware
Moving Sbox into Hardware

- SBOX regs (16 byte + 4 byte rkey)
- sbox (AES32)
- 128
- 32
- 32
- ALU
- DCACHE
- ICACHE
- SBOX regs
- 128
- 32
- CPU
- instruction file
- register file
- instruction fetch/decode
- DCACHE
- ICACHE
- AES driver code
- RAM
- Hardware Sbox with ASIP interface
- Constant Execution Timing

- Timing Side-channel fixed, but others remain.

Design Step: Secure partitioning

- Secure design over multiple abstraction levels (protocol, software, hardware, circuits)
Applications

- Hardware Chain-of-Trust
  - Can we build a path in a device that is completely trusted, even as it extends into hardware?

- Side-channel resistant hardware in FPGA
  - Can we port side-channel resistant design styles for ASIC into FPGA while maintaining their properties?
Hardware Chain of Trust for DRM - Why?

- Download copy-righted multimedia into player
- Side-channel: Probe and extract DRM-free high-quality music

Hardware Chain of Trust for DRM - Principle

- Encrypt at Server Hard-drive
- Decrypt at D/A conversion in Player
- Challenge: How to build this chain-of-trust?
- Hardware-specific Key-exchange
Chain-of-trust for Video Messaging

[FPGA-unique encryption]

CF-card
- Display SW
- Trivium Key
- Message

SAM
- key
- Trivium Stream Cipher
- VGA

FPGA
- PPC
- Program Memory

Incorrect authentication
incomplete chain-of-trust

Correct authentication
complete chain-of-trust

[Eric Simpson, VT]
Chain-of-trust for Video Messaging

FPGA-unique encryption

CF-card

Display SW
Trivium Key
Message

Compact Flash Data

PPC Baseline Config

PPC bitstream decryption (keys)

SAM

key
Trivium Stream Cipher

VGA

Legend:

CF-card

Compact Flash Data

PPC Baseline Config

PPC bitstream decryption (keys)

SAM decryption (PUF)

Secure SW & Data Download

Configure VGA Display

SAM

key
Trivium Stream Cipher

VGA

Legend:

CF-card

Compact Flash Data

PPC Baseline Config

PPC bitstream decryption (keys)

SAM decryption (PUF)

Secure SW & Data Download

Configure VGA Display
Chain-of-trust for Video Messaging

Applications

- **Hardware Chain-of-Trust**
  - Can we build a path in a device that is completely trusted, even as it extends into hardware?

- **Side-channel resistant hardware in FPGA**
  - Can we port side-channel resistant design styles for ASIC into FPGA while maintaining their properties?
Side-channel resistant design

- Reduce Power Variation
- Constant-Power Design
- De-correlate Power Variation
- Masking

Wave Dynamic Differential Logic
Random Switching Logic

Reducing Power Variations: WDDL

- Wave Dynamic Differential Logic (Tiri 2003)
  Gates have exactly one 0→1 transition per clock cycle

- Differential output ensures each switch contains 0→1
- Precharge output guarantees switching each cycle
Exactly one switching event per gate per cycle

Asymmetry in C-load gives residual power leakage
Need symmetrical place-and-route technique, not easy in contemporary tools
Unbalanced WDDL easy to break

- Impact of imbalance of 1 part in 500 per WDDL net

WDDL in FPGA

- Use FPGA fabric regularity to build a better WDDL

- 'Copy' + 'Relocate' results in identical routing pattern
- 'Modify' LUT content creates complementary logic function
Mapping WDDL in FPGA

- Create complementary function starting from differential netlist by switching Q and Qbar

- 4X in area over single-ended

+ same advantages as WDDL with symmetrical routing

'DWDDL', Double Wave Dynamic Differential Logic

Test Setup considers 3 possible cases

Single-ended  WDDL  WDDL + Complementary

SE  WDDL  DWDDL

Test setup

Measurements
Conclusions

- In embedded systems, ZERO-risk security does NOT exist
- In embedded systems, LOW-risk security IS possible
- Methodology is essential
- Many open problems
  - How to quantify security? Number of measurements?
  - Cost versus Security trade-offs?
  - Can we build tools to automate analysis and secure design ?
Thanks!

Patrick Schaumont, Eric Simpson, Pengyuan Yu

Secure Embedded Systems Group
ECE Department