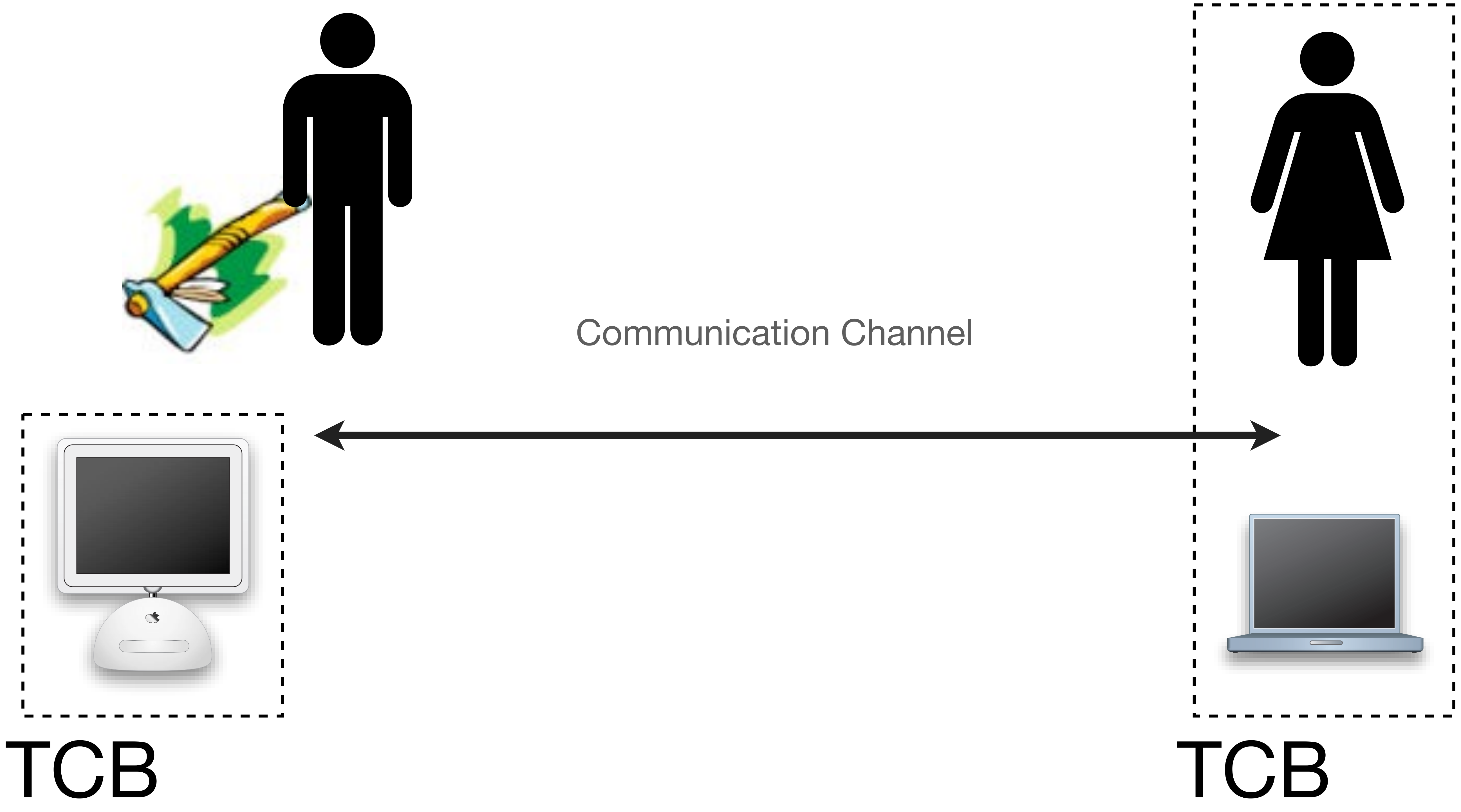
Practical Cryptographic Systems

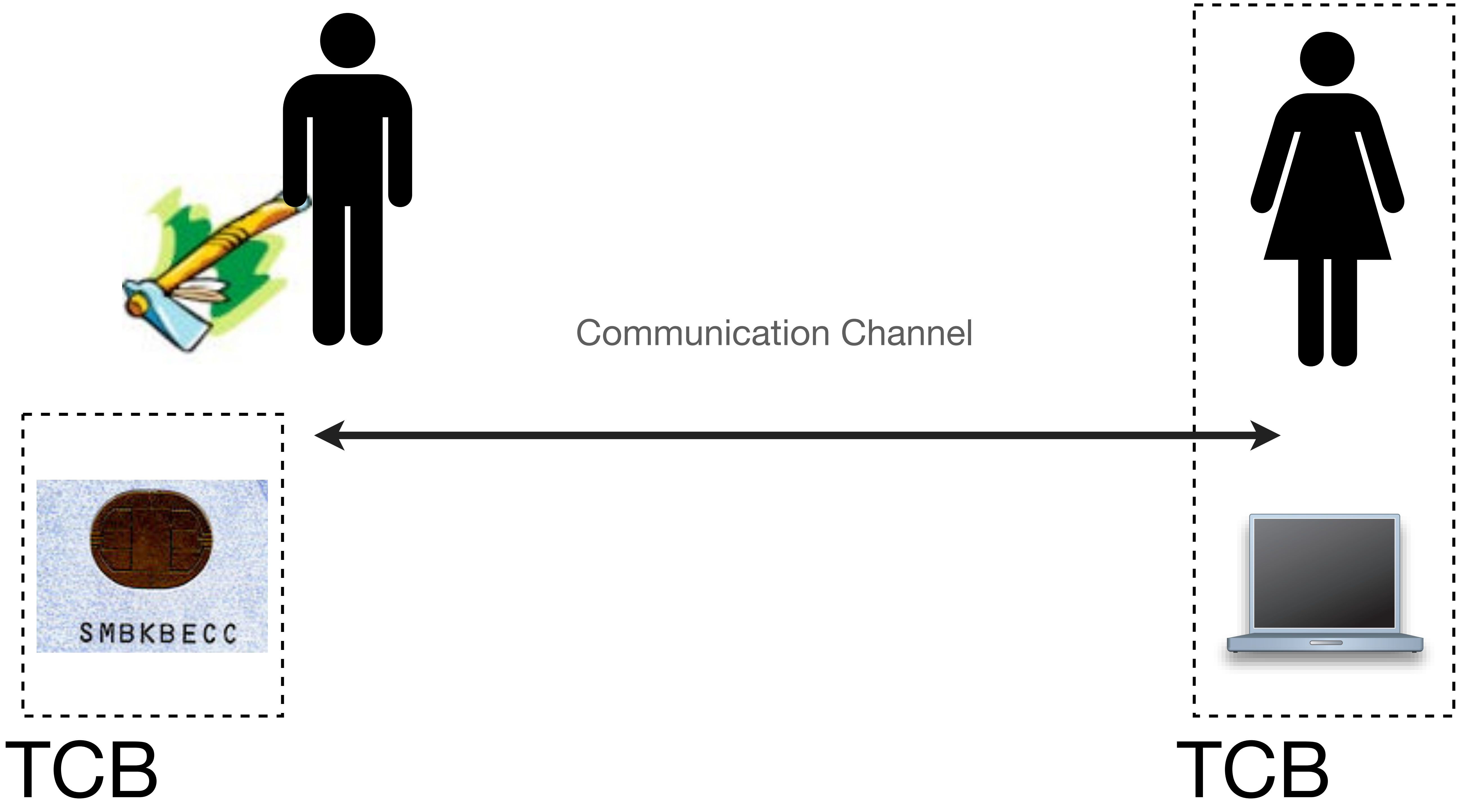
Hardware Security & Tamper Resistance*

Matthew Green

Today



Today



Physical Security

- Devices in a hostile environment
  - Military cryptoprocessors
  - Financial services
- ATM machines
- Credit-card smartchips
  - Digital Rights Management
- Trusted Computing

Threat Models

- Various levels of sophistication
  - Class 1: Clever Outsiders
    Moderate knowledge of system, will usually attack via a known weakness

  - Class 2: Knowledgeable Insiders
    Specialized technical education & experience
    Access to the system & tools to exploit it
  - Class 3: Funded Organizations
    Team of attackers backed by significant funding,
    access to experts and Class 2 attackers

FIPS Levels

- FIPS 140-2: Cryptographic Modules
  - FIPS 140, Level 1

- Software only, no physical security
  - FIPS 140, Level 2

- Tamper evidence or pick-resistant locks
  - FIPS 140, Level 3

- Tamper resistance
  - FIPS 140, Level 4

- Strong physical security around device

System Examples

- Classical ATM machine:
  - Armor, temperature sensors, tilt sensors
  - Attacker has to drill through the shell, defeat any tamper sensors, avoid setting off alarms
  - Might lose $$, but can often protect cryptographic secrets

System Examples

- Modern ATM machine:
  - No armor, limited physical security
  - Owner/operator might be untrustworthy
  - Limited $$ amounts, still worry about loss of key material
  - Interface based on commodity OS (Windows CE?)

System Examples

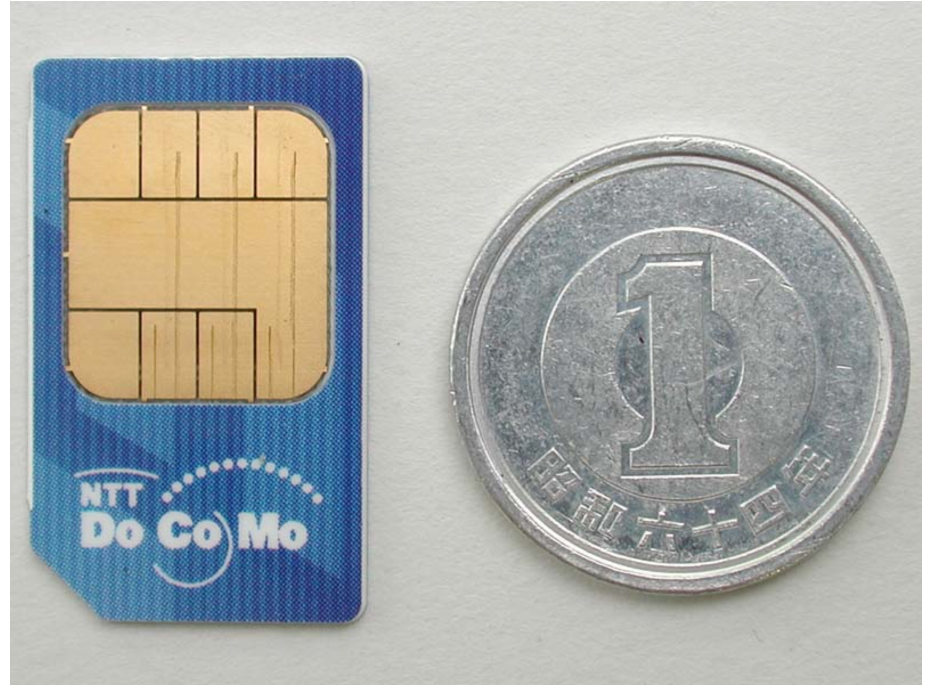
- Back-end server:
  - Strong physical security <u>around</u> server room
  - Performance is critical
  - Insider attacks a major concern
  - Large potential $$ losses if key material com
  - Linux/Windows/Legacy

System Examples

- Client-side smartcard:
  - User/owner is potentially hostile
  - Worse if card is stolen
  - Attacker-su [illegible] ver

Two approaches

- Tamper-evidence
  - Make tampering obvious
  - Allow for recovery/renewability
    (e.g., re-generate cryptographic keys)
- Tamper-resistance:
  - Keep attackers out
  - Wipe cryptographic key material



A General Approach

- Minimize the size of the “T” in our TCB
  - I.e., putting an ATM in a safe is <u>expensive</u>
  - Anchor trust in this area, build a secure system from there
  - Might involve untrusted storage/RAM/ROM/processing, all connected to one trusted component

A General Approach

- Might also be necessary for practical reasons:
  - Support multiple “untrusted” manufacturers
  - E.g., SIM chips
  - E.g., CableCARD

Protecting Cryptoprocessors

- Metal
  - Can be cut/drilled
- Epoxy
  - Can be scraped off, penetrated with a logic analyzer probe

IBM 4758

IBM 4758

- High-security Cryptoprocessor
  - RAM, CPU, cryptographic circuitry
  - Dedicated SRAM for key memory
  - Battery powered
  - All wrapped in:

•Aluminum EM shielding

•Tamper-sensing mesh

•Potting material

IBM 4758

- Tamper sensing mesh:
  - Thousands of small wires wrapped around device
  - Interrupting any circuit (i.e., drilling) wipes the contents of key SRAM

•v1: copper wires

•v2: printed circuits

IBM 4758

- Memory remanence attacks:
  - Key “burn in”
  - After storing keys for too long, they may become permanent default states of RAM
  - NSA <u>Forest Green Book</u> has guidelines for this
  - Solution: “RAM savers”

IBM 4758

- Memory remanence attacks:
  - Attacker might try to freeze the device
  - Thus keys would survive wipe
  - Bursts of X-Rays can “lock” RAM in
  - Protections: temperature sensor, radiation sen

•Gets too cold, keys go away

•Though what if we ship UPS!

Capstone/Clipper

- Proposed national voice encryption device
  - Designed as a compromise between need for strong crypto, and US's need to eavesdrop
  - Contained a secret cipher (Skipjack) w/80-bit key
  - Tamper-resistance:

•Difficult to extract embedded secret keys

•Difficult to R.E. Skipjack design

•Difficult to alter operation

  - Currently used in Fortezza card (US gov't SECRET)

Capstone/Clipper

- Threat model & design considerations
  - Extremely hostile environment
  - Range of well-funded adversaries (probably non-military)
  - Protecting secrets & design & operation
  - Very small device

Clipper/Capstone

- Tamper-resistance:
  - Extremely sophisticated
  - Metal/epoxy top
  - Vialink Read-Only Memory (VROM)

•Bits set by blowing antifuses using electrical charges

Clipper/Capstone

- Attacking Clipper & QuickLogic:
  - Remove upper metal layer (difficult)
  - Use an electron microscope to read VROM (expensive & difficult, requires extra analysis)
  - Monitor circuit while chip is in use (more promising)

Capstone/Clipper

- Operation:
  - Each ciphertext accompanied by LEAF (Law Enforcement Access Field)
  - Essentially, key escrow under gov't key
  - LEAF contains a "checksum" (MAC) to prevent tampering/removal
- Break:
  - Matt Blaze - found that LEAF bound to message w/ 16-bit checksum

Smartcards

- Very widely used
  - Contact/Contactless varieties
  - Small microprocessor/RAM/Serial bus
  - Became major targets of attack due to:

•Satellite TV

•GSM phones

•Lately: Payment Cards

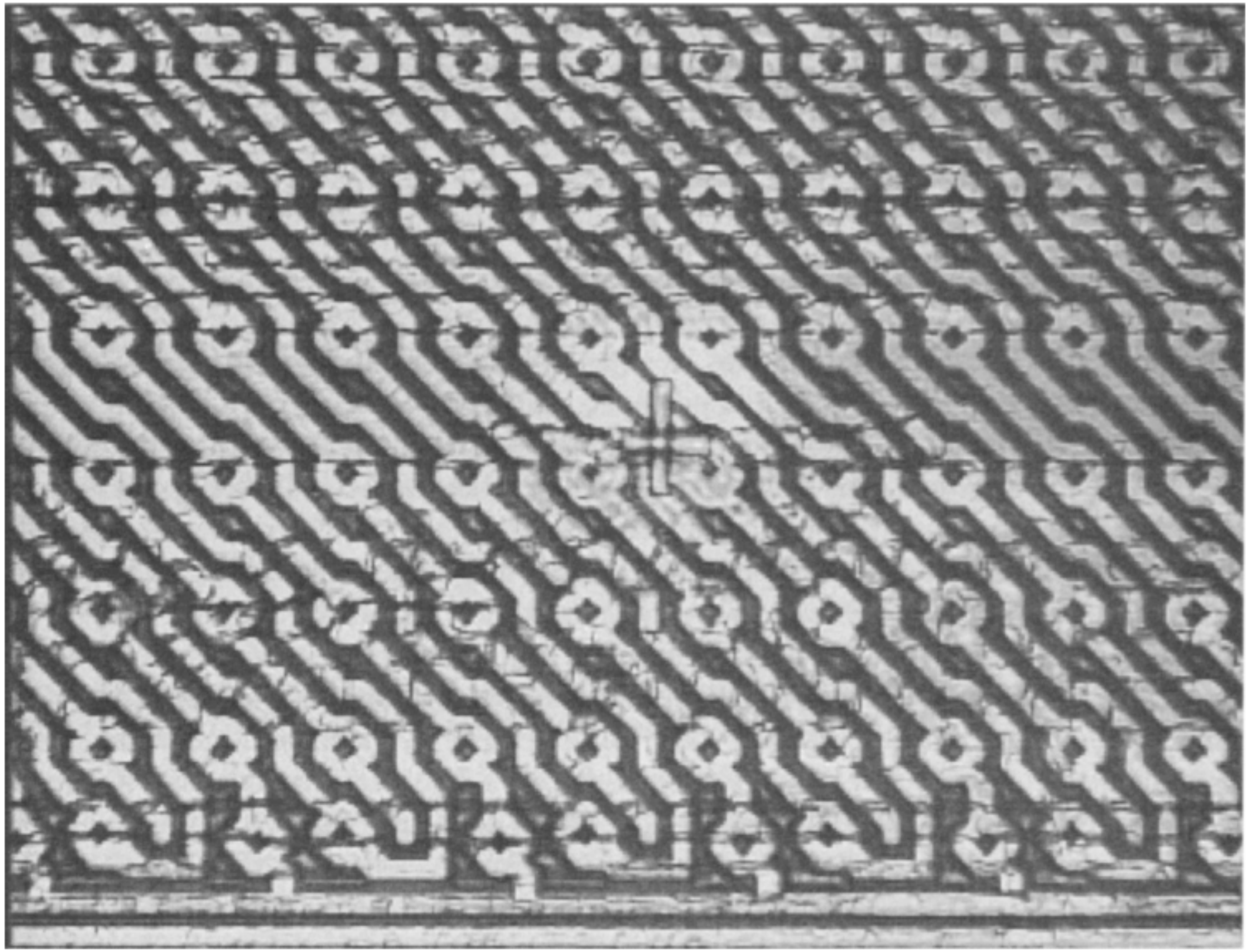
Attacking Smartcards

- Protocol-level attacks
  - Intercept messages, control access
- Side-channel attacks
  - Attacker controls the power source
  - DPA countermeasures introduces in 1990s

Attacking Smartcards

- Attacks on voltage supply
  - Memory read/write attacks
  - Fault injection
- Attacks on hardware
  - Take it apart, use an electron microscope
  - Same countermeasures, though:

•protective mesh, potting



Tamper-evidence

- Seals/locks
  - Make sure you can detect and renew security after an attack occurs
- Can even be implemented in software (under certain assumptions)

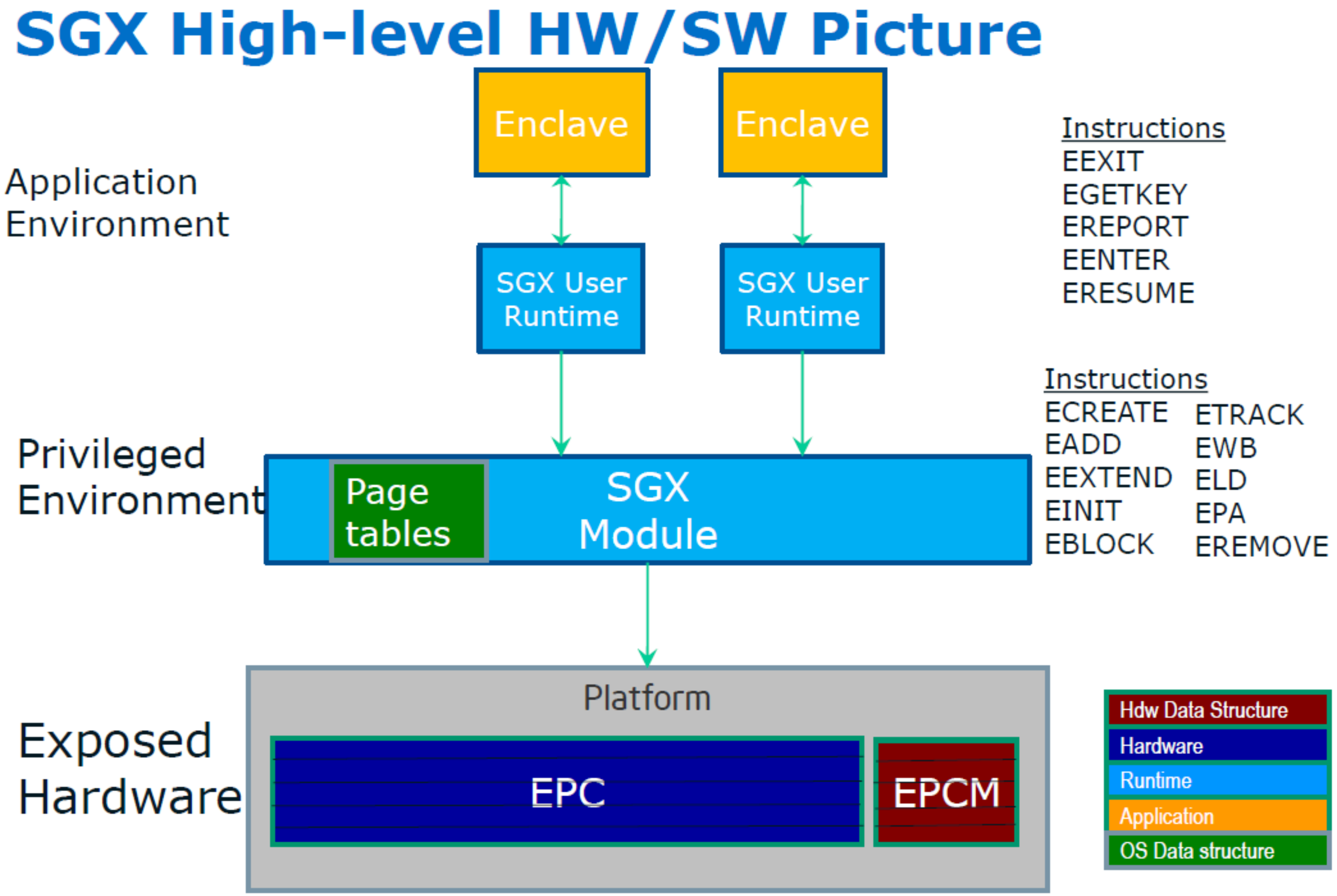
Trusted Computing

- Effort to put a tamper-resistant security processor into every PC
  - Useful to verify software integrity
  - Key management for access control & DRM
  - Minimize co-processor footprint by bootstrapping secure software
- Newest incarnation: Intel SGX

Trusted Computing

- Effort to put a tamper-resistant security processor into every PC
  - Useful to verify software integrity
  - Key management for access control & DRM
  - Minimize co-processor footprint by <u>bootstrapping</u> secure software
- Newest incarnation: Intel SGX

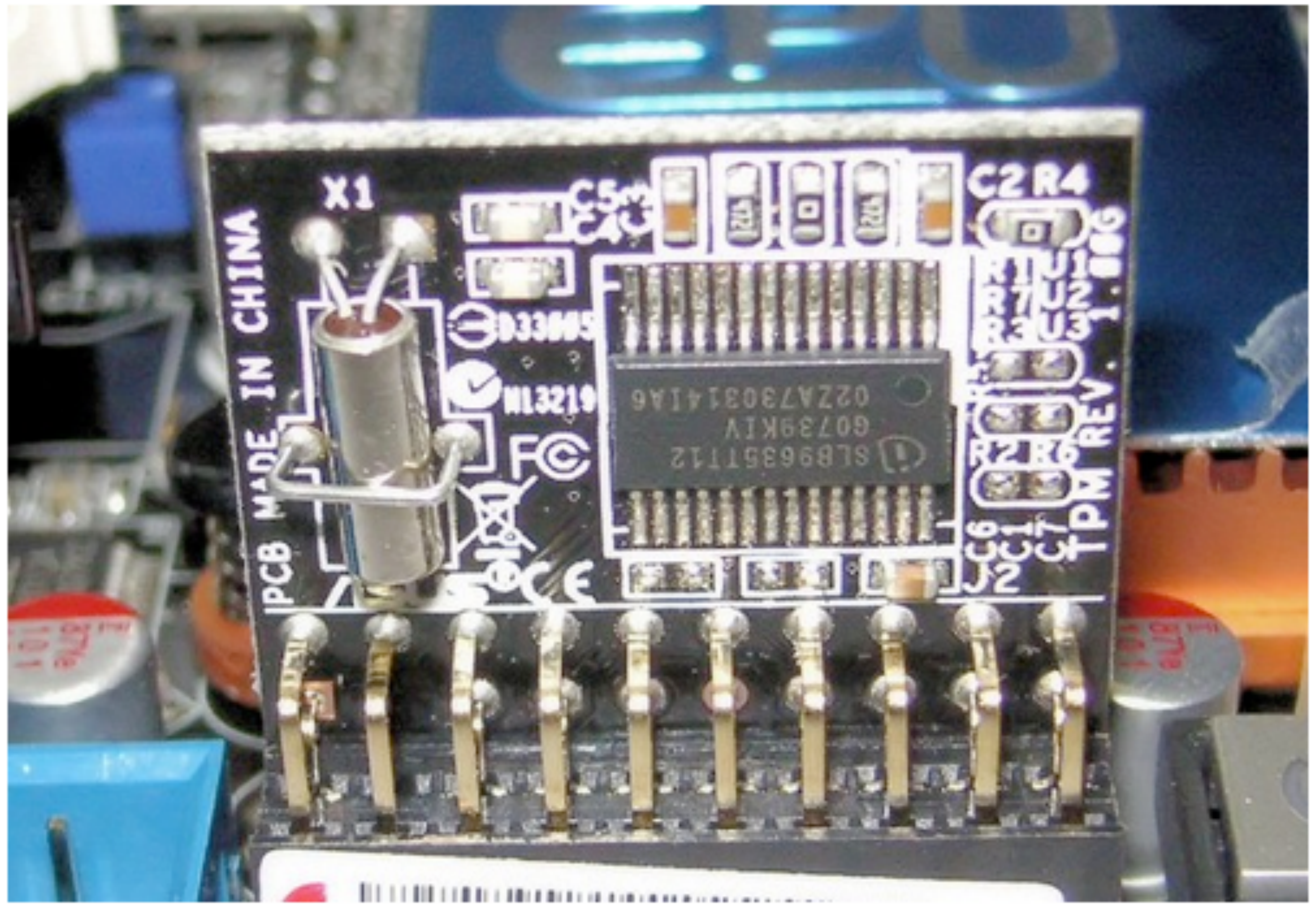
SGX


SGX High-level HW/SW Picture
Enclave
Enclave
Application Environment
SGX User Runtime
SGX User Runtime
Instructions
EEXIT
EGETKEY
EREPORT
EENTER
ERESUME
Instructions
ECREATE
EADD
EEXTEND
EINIT
EBLOCK
ETRACK
EWB
ELD
EPA
EREMOVE
Privileged Environment
Page tables
SGX Module
Platform
Exposed Hardware
EPC
EPCM
Hdw Data Structure
Hardware
Runtime
Application
OS Data structure

FILED UNDER *Desktops*, *Laptops*

Christopher Tarnovsky hacks Infineon's 'unhackable' chip, we prepare for false-advertising litigation

By Tim Stevens posted Feb 12th 2010 10:31AM



As it turns out, Infineon may have been a little bit... *optimistic* when it said its SLE66 CL PE was "unhackable" -- but only a little. The company should have put an asterisk next to the word, pointing to a disclaimer indicating something to the effect of: "Unless you have an electron microscope, small conductive needles to intercept the chip's internal circuitry, and the acid necessary to expose it." Those are some of the tools available to researcher Christopher Tarnovsky, who perpetrated the hack and presented his findings at the Black Hat DC Conference earlier this month. Initially, Infineon claimed what he'd done was impossible, but now has taken a step back and said "the risk is manageable, and you are just attacking one computer." We would tend to agree in this case, but Tarnovsky still deserves serious respect for this one. Nice work, Big Gun.

Trusted Computing

- Implications
  - Same chip used in the XBox 360
    (allegedly, Tarnovsky offered $100k to hack it)
  - Illustrates the most important lesson
    of tamper resistant systems:

- Individual devices can and will be hacked
- Must ensure that single-device hack
(or easily replicable attack)
does not lead to system-wide compromise!