CS 161: Computer Security

Lecture 18

November 17, 2015

The Joy of Tech

by Nitrozac & Snaggy


ACCIDENT
ON
MOTORWAY


ACCIDENT
ON
MOTORWAY


CAUTION!
ZOMBIES!
AHEAD!!!
RUN FOR
COLD
CLIMATES
THE END
IS NEAR!
!!!!!!!!
NAZI
ZOMBIES!
RUN!!!


RAPTORS
AHEAD
CAUTION


TRAPPED
IN SIGN
FACTORY


SEND
HELP!

The dream – secure hardware

- FIPS 140
  - “*Security Requirements for Cryptographic Modules*”
- Specifies hardware standards for security

Tamper Resistant Devices

- Locks, guards, alarms- massive physical protection, often inconvenient
- Portable tamper resistant devices- secure distributed apps on device free of physical attack
- Case studies:
  - IBM 4758
  - iButton
  - Dallas 5002
  - Clipper chip
  - Smartcards

Advantages

- Control information processing by linking to a single physical token
- Assurance that data are destroyed at definite and verifiable time
- Reduce need to trust human operators
- Control value counters- prepaid cards with limited value

Tamper Resistance vs Tamper Evident

- Tamper resistant- key cannot be extracted, very costly, not available for mass market
- Tamper evident- key extraction obvious, only physically invasive attacks possible
- Possible to make smartcards tamper-evident

In “Da Vinci Code”, a cryptex used to store secret info. Have to know the code and align dials properly. If force opened a vial with vinegar breaks dissolving the papyrus paper with the secret info.

Evolution

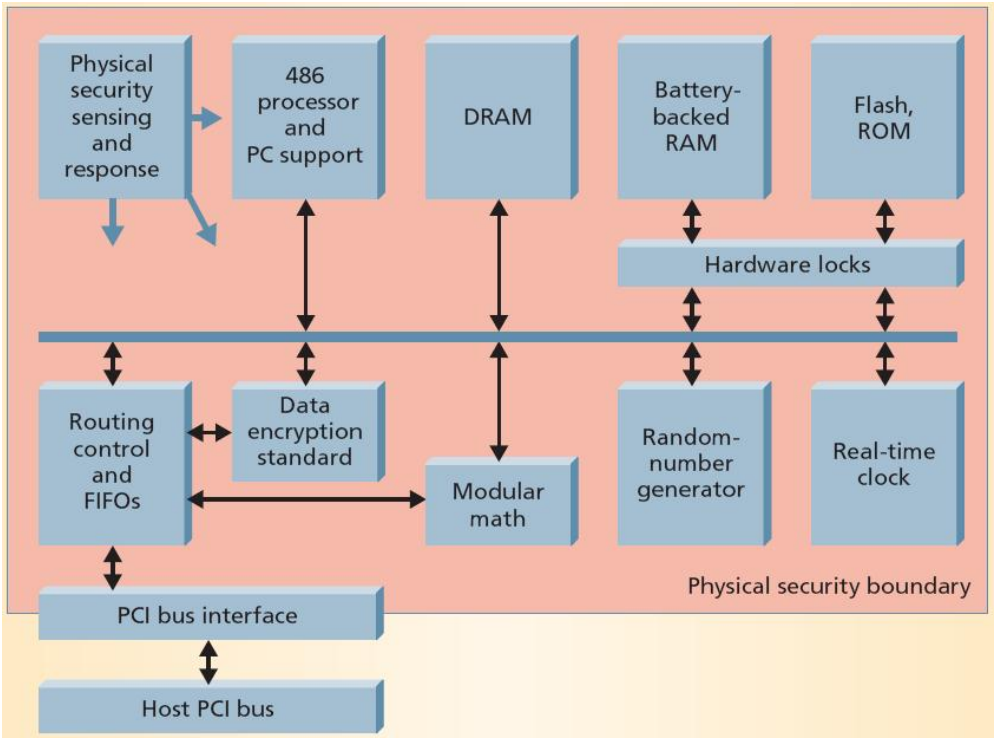
- Operator dependent:
  - Weighted code books
  - Water-soluble ink
  - Cellulose nitrate printing, self-destruct thermite charges
- Tamper-resistant or tamper-evident devices:
  - Tattle-tale container, a tamper-evident containers for paper keys
  - “Fill gun” portable device for controlled loading of crypto keys

IBM 4758

- PCI Cryptographic Coprocessor
- Available for IBM servers, costs ~$2000
- High security, certified under FIPS at level 3 & 4
- Blocks knowledgeable insiders and funded organizations
- Secure key storage and secure processing of sensitive apps

IBM 4758

- Electronics, microprocessor, memory, random generator encased in metal enclosures
- Tamper-sensing barrier, aluminum shielding, potted in block of epoxy
- Self-initialization: root certificate generated at factory
- Outbound authentication: device & content can be identified externally



IBM 4758

- Physically secure, no known attacks
- Export controlled, protected protocol
- Broken by Mike Bond, Cambridge student. Stole 3DES key from CCA software, the API protocol
- IBM taxonomy of attackers
  - Class 1: clever outsiders
  - Class 2: knowledgeable insiders
  - Class 3: funded organizations

iButton

- Medium-security processor, FIPS level 3.5
- Small, self-contained crypto processor; costs $10-$20
- Used as access token for secure laptops, parking, mass transit
- Contains 8051 µprocessor, exp circuit, clock, tamper sensors, static RAM for software and keys encased in steel can
- Lithium battery, lid switch, tamper-sensing mesh

Dallas 5002

- DS5002 microcontroller
- Medium-grade security device from Dallas Semiconductor
- Used in point-of-sale terminals
- Holds keys for encrypting PINs
- Bus encryption-memory addresses and content on the fly, can use external memory
- Tamper-sensing mesh

Clipper, Capstone

- Tamper-resistant chips
- Escrow Encryption Standard
- Use a government owned key to encrypt user supplied key, LEAF
- Compute 16-bit checksum of LEAF with “family key” shared by all Clipper chips
- Capstone used in government service, PCMCIA card to encrypt secret data
- Program withdrawn after broad criticism

Smartcards

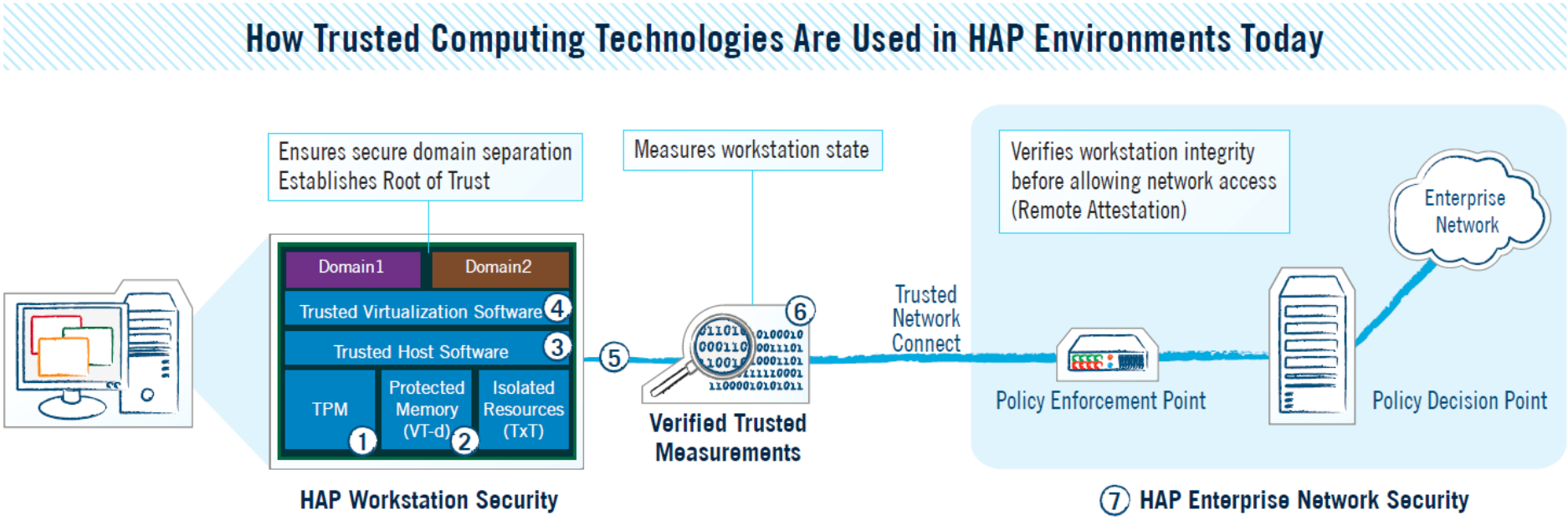
- Most common secure processor, cheap, costs ~$1
- Authentication, security function for more expensive electronics: cell phones, settop boxes, hotel keys, next generation US credit cards and ATM/debit cards

Smartcards

- 8-bit microprocessor, memory, serial I/O circuit
- Often packaged in plastic card
- ROM: program and immutable data (16 KB)
- EEPROM: customer-specific data (16KB)
- RAM: transient data (256 B)
- External connections: power, reset, clock, & serial port
- Usually has protective surface mesh or coating

Attestation

- “Trusted computing”
  - Controversial, but Intel continues pushing forward



Attestation

Trusted Platform Module (TPM)

Embedded security chip that secures keys and data. HAP uses the TPM to attest to the machine's identity and the integrity of the software running on it.

Embedded Hardware Virtualization Security

Intel®'s VT-d and TxT hardware technologies protect execution space and memory, and directly pair I/O devices with domains. HAP uses these technologies to protect resources in one domain from unauthorized access by hardware or software in another domain.

Trusted Operating System

A rigorously tested operating system that is measured and granted privileges to access critical security data and resources. HAP uses a trusted operating system as a host operating system for the secure virtualization software and to tightly restrict the ability of one process to compromise the security of another process.

Secure Virtualization Software

A specially tested commercial hypervisor manages concurrent operation of multiple guest operating systems. HAP uses enhanced secure virtualization software to ensure secure domain separation.

Trusted Boot

The trusted boot process measures the software that is running on a machine each time it boots. HAP securely reports or "attests to" those measurements when required.

Remote Attestation

Remote attestation provides "verification" of software state on a client and verifies that proof on a remote machine. HAP uses remote attestation to manage network access control and ensure that only trusted machines in the proper state are allowed on HAP-protected networks.

Trusted Network Access Control

Manages which devices are allowed to access network resources. Trusted network access control utilizes a cryptographic proof or "attestation" that the software on the client is trusted. HAP includes two remote components, a Policy Decision Point (PDP) and Policy Enforcement Point (PEP), to securely manage the validation of the client attestation and control access to the network.

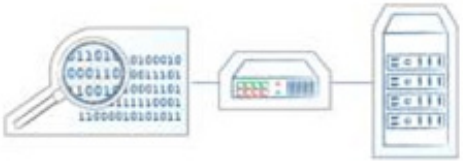
Attestation

Attestation

1) Hardware-based Root of Trust: HAP relies on the Trusted Platform Module (TPM), an implicitly trusted hardware component, to store encryption keys and system measurements and protect against software-based attacks.

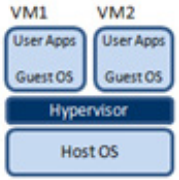
2) Device Measurement: The identity and integrity of each hardware and software system component are measured and verified before passing control.

3) Measurement Monitoring: Verifiable reports of a device's identity and current configuration are transmitted to the network, where decisions are made governing network access and device disposition. No unknown or noncompliant devices are allowed on the network.



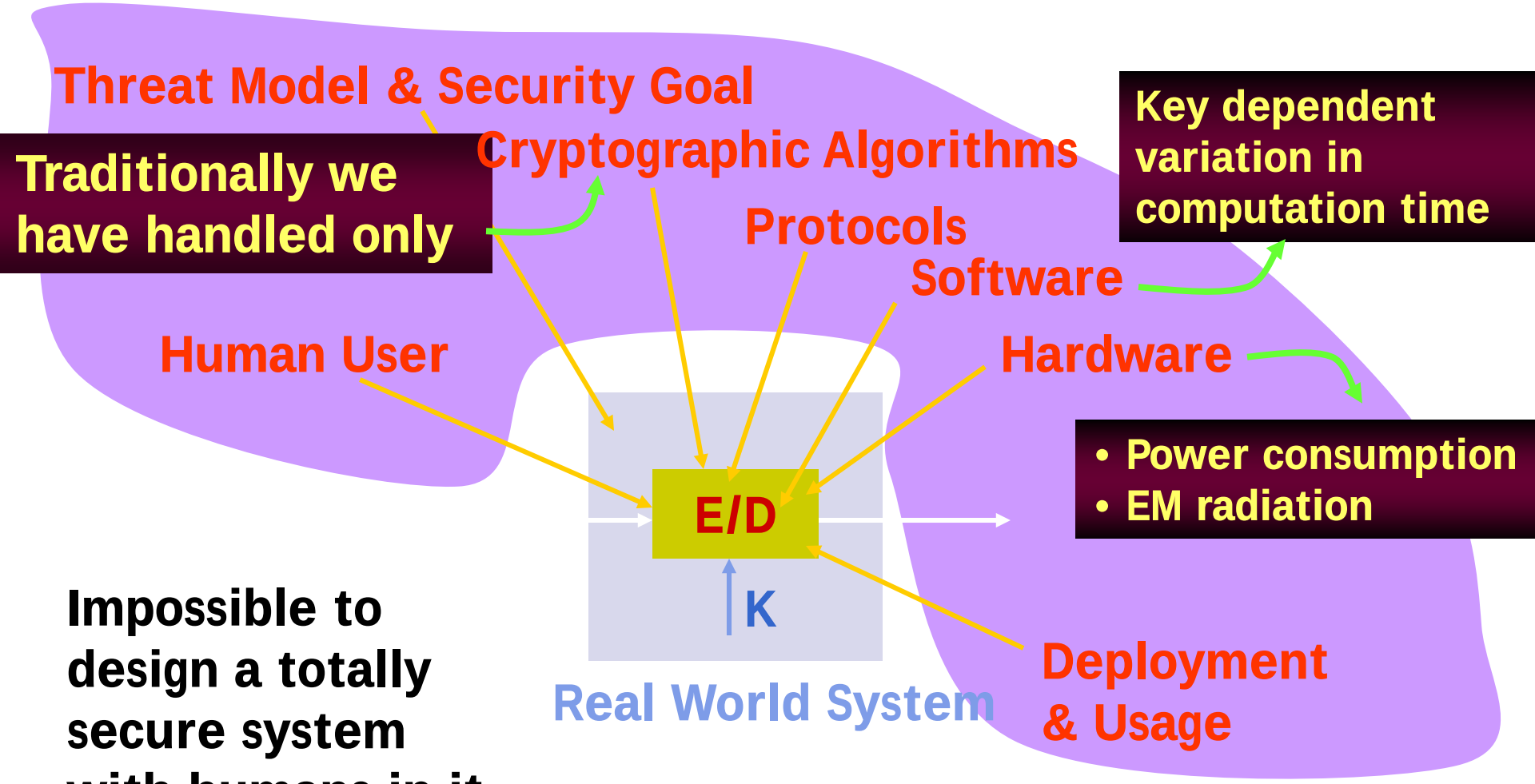
4) Long Term Protected Storage: Hardware-based full disk encryption ensures that data is secure, even if drives are removed from workstations.

5) Process Separation: HAP uses hardware- and software-secured virtualization to separate user processes from supervisor processes. Secure domain separation enables multiple security domains to be hosted on a common computing platform base with no unintended interaction.



6) Program Isolation: HAP uses guest partitions like virtualization or separation kernels to separate applications from one another. Code, Data and Resources associated with Process A are unavailable to Process B.

Side Channel Sources

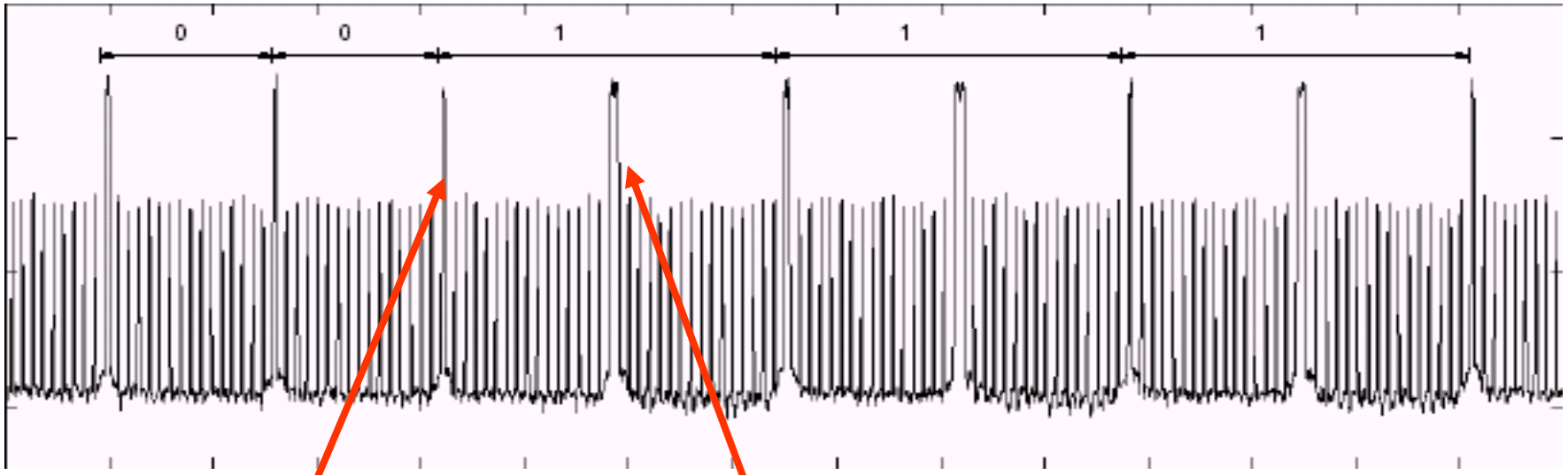


Impossible to design a totally secure system with humans in it

Power Analysis Attack

Idea: During switching CMOS gates draw spiked current

Trace of Current drawn - RSA Secret Key Computation



Reported Results : Every smartcard in the market broken

Possible side channels

- Power
- Time
- Faults
- Electromagnetic radiation
- Sound
- and many more ….

Side Channel Analysis: Simple

- Simple Side Channel Analysis
  - Use characteristics directly visible in single measurement trace
  - Key has simple, exploitable relationship with the operations that visible in trace.
  - Typically, vulnerable implementations include key dependent branching.

Side Channel Analysis: Differential

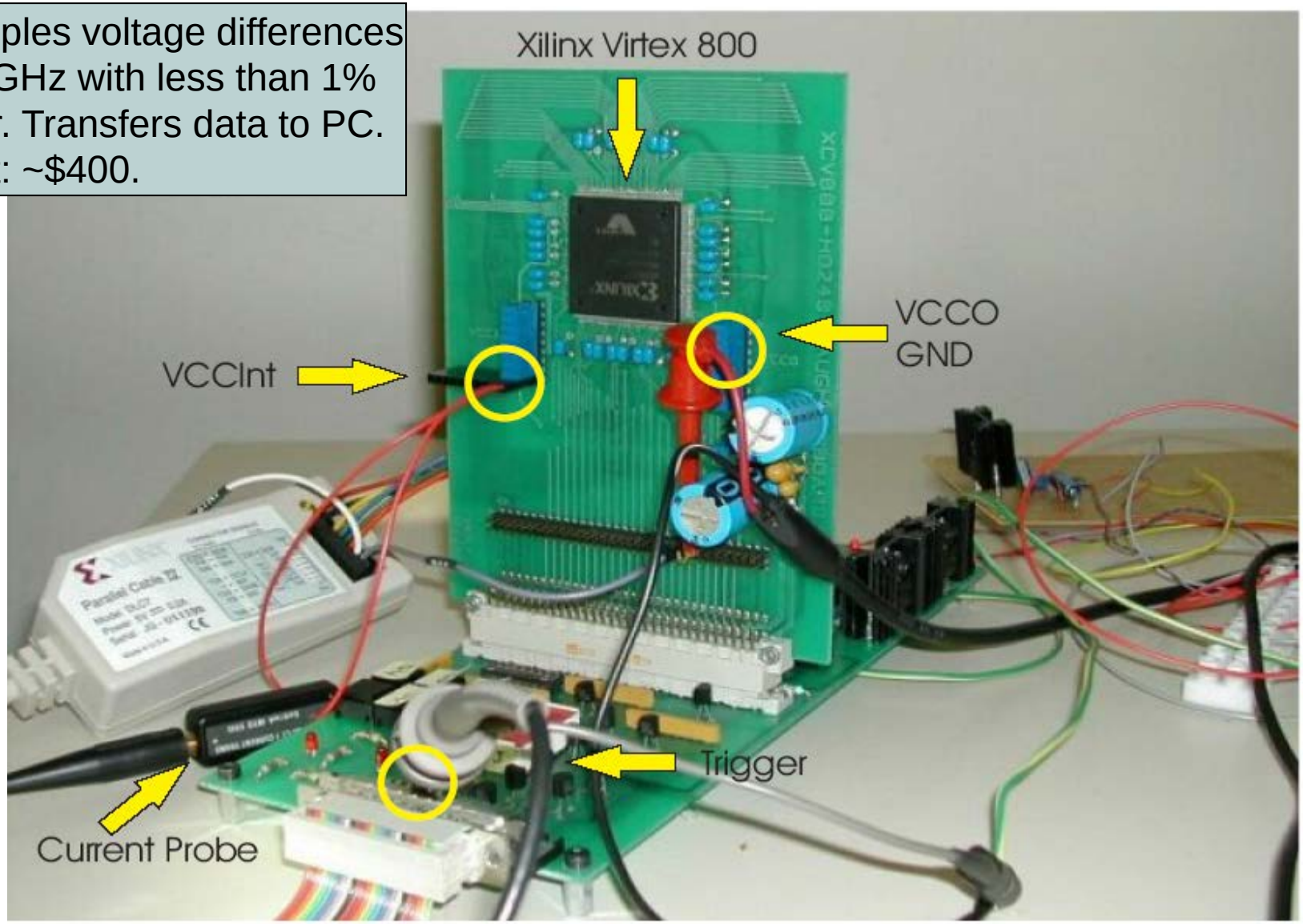
- Differential Side Channel Analysis
  - Requires multiple traces, use statistical methods
  - Targets specific intermediate result in a specific part of the measurement traces
  - Choose selection function (intermediate result)
  - Selection function depends on known input/output data and a small number of hypotheses on key value

Power Attacks (PA)

- Differential Power Attacks (DPA)
- Dynamic current consumption of chip is correlated to gate activity
  - intermediate results of the algorithm

Equipment for power analysis

Samples voltage differences At 1GHz with less than 1% error. Transfers data to PC. Cost: ~$400.



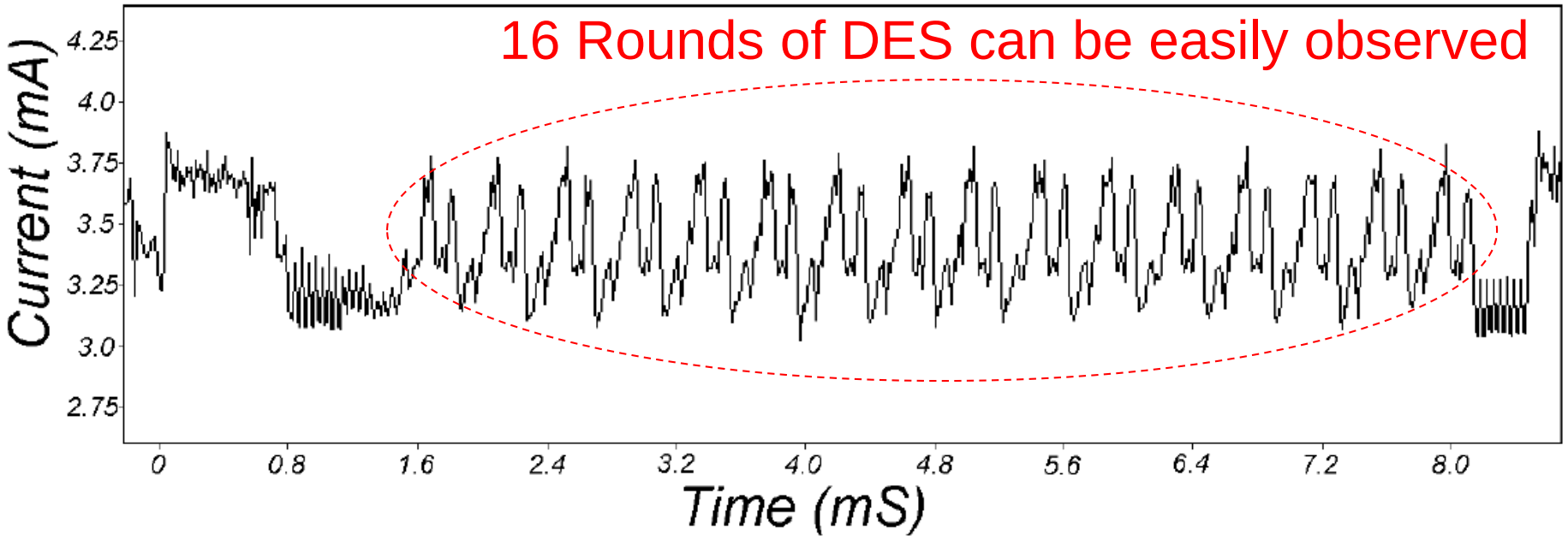
Simple Power Analysis (SPA)

- Directly interprets the power consumption of the device
- Trace: A set of power consumptions across a cryptographic process
- 1 millisecond operation sampled at 5MHz yields a trace with 5000 points

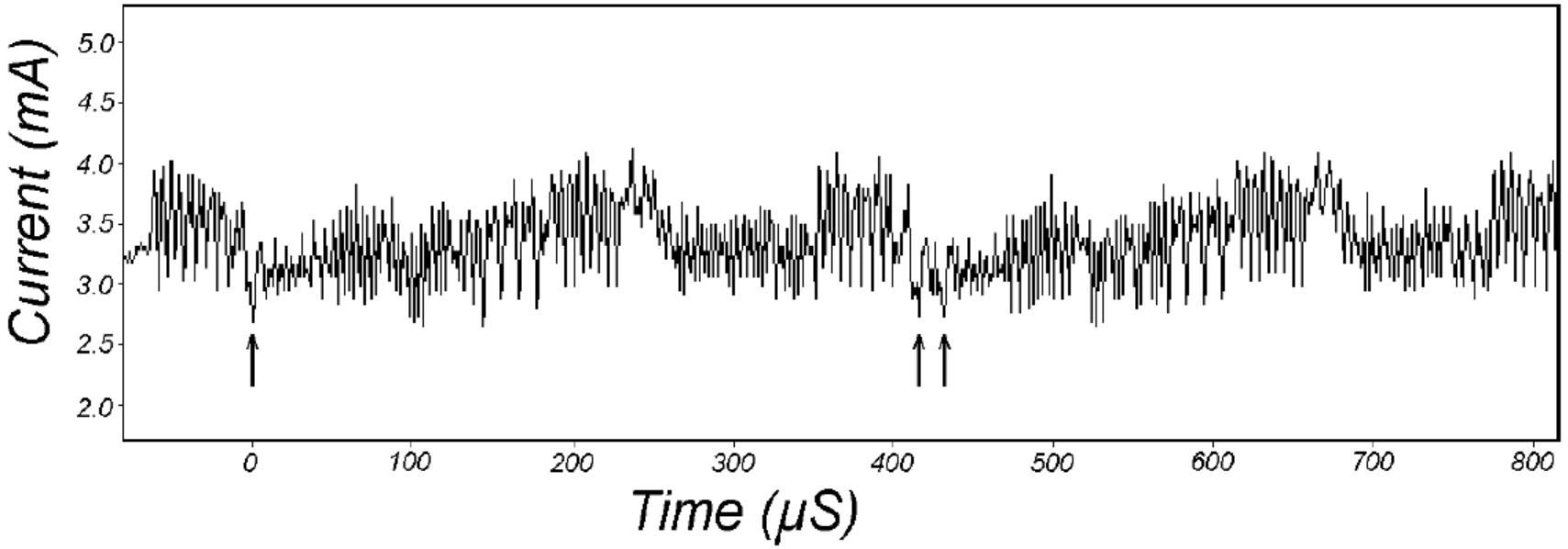
DES review

- DES is a block cipher
- 64 bit block length
- 56 bit key length
- 16 rounds
- 48 bits of key used each round (subkey)
- Each round is simple (for a block cipher)
- Security depends primarily on “S-boxes”
- Each S-boxes maps 6 bits to 4 bits
- Each S-box has a share of 6 bits of the key

Power Trace of DES

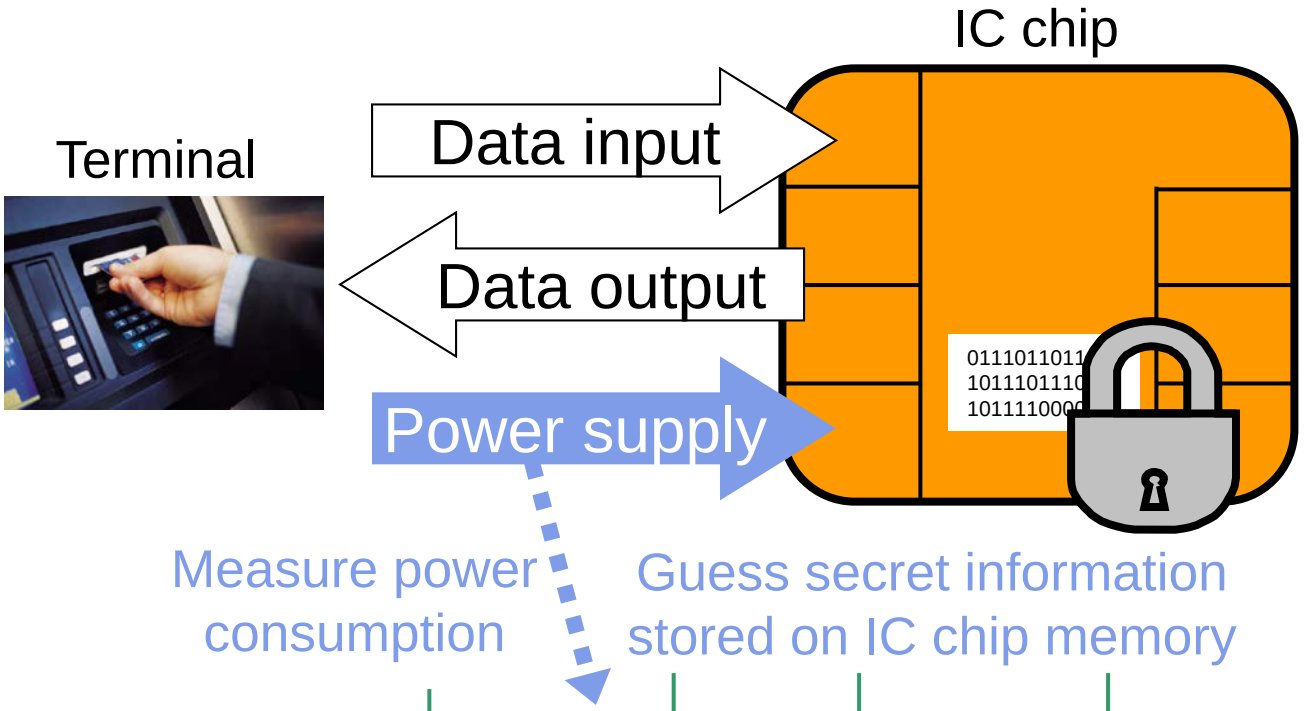


Power Traces for DES



28 bit key registers C and D are rotated once in round 2; twice in round 3. These conditional branches depending on the key bits leak critical information.

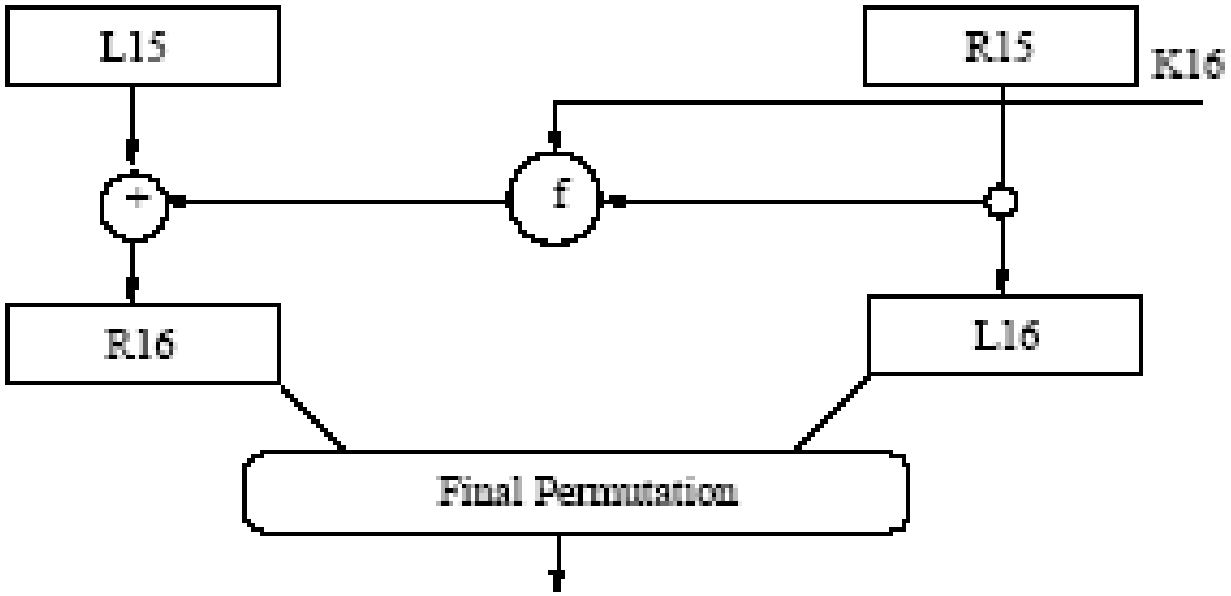
Simple Power Analysis

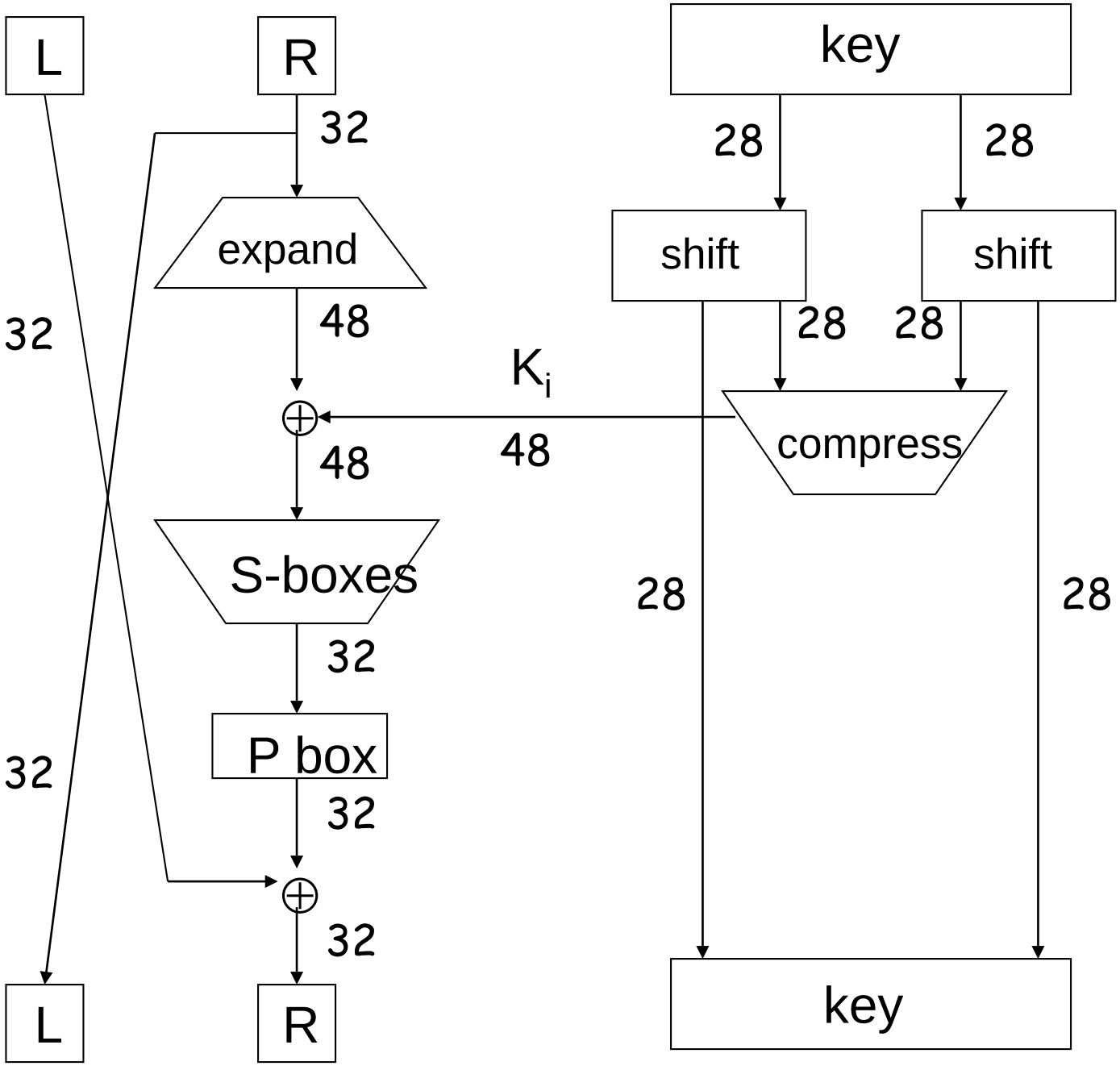


Differential Power Analysis

- More powerful and difficult to prevent than simple power analysis
- Note: different power consumption for different states (var = 0 or 1)
- Data collection phase data analysis phase
  - Gather many power consumption curves
  - Assume a key value
  - Divide data into two groups (0 and 1 for chosen bit)
  - Calculate mean value curve of each group

Last Round of DES





One Round of DES

DPA for DES

1. Make current consumption measurement of about 1000 DES operations (100,000 data points/curve)
2. Assume a key for an S-box of the last round
3. Calculate first S-box first bit output for each plaintext using the assumed key
4. Divide the measurement into two groups (0 & 1)
5. Calculate average curve of each group
6. Calculate difference of two average curves
7. Correct key →spikes in differential curve
8. Repeat 2-7 for other S-boxes

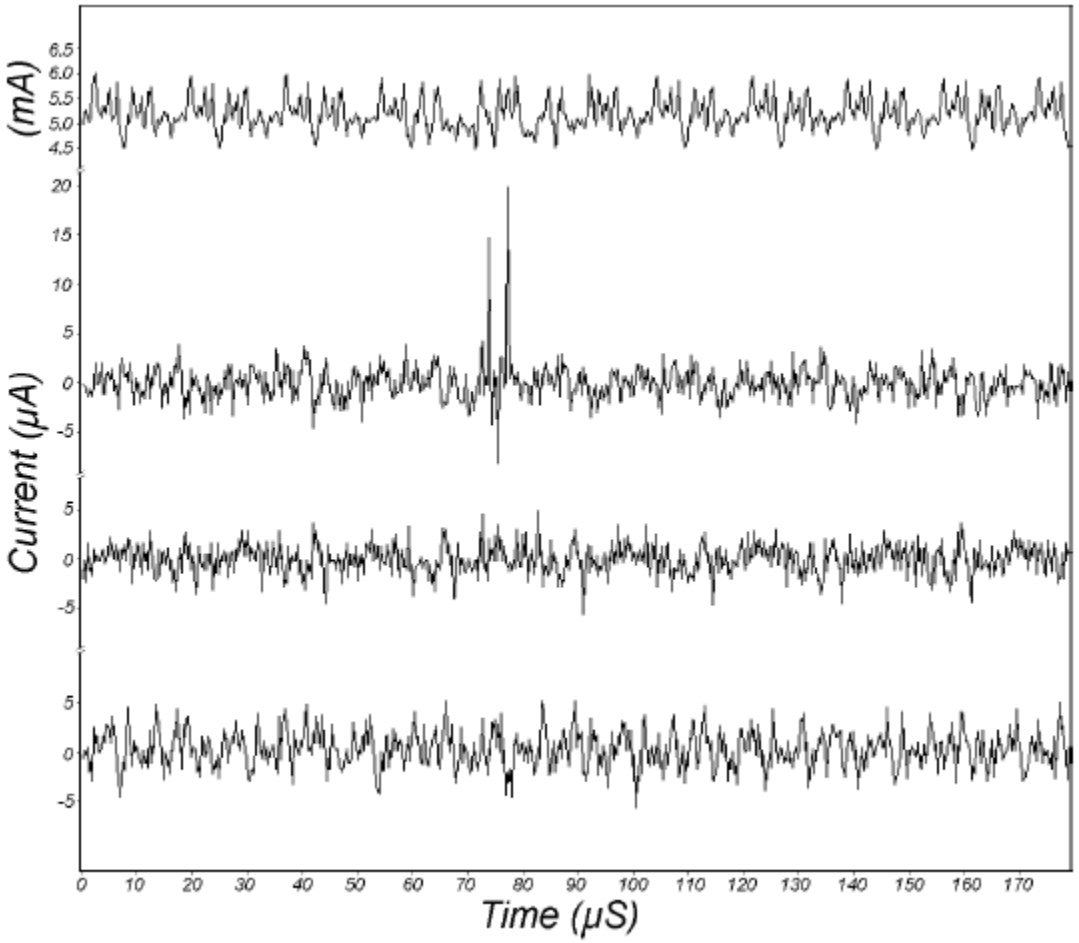
DPA for DES example

Average Power Consumption

Power Consumption Differential Curve With Correct Key Guess

Power Consumption Differential Curve With Incorrect Key Guess

Power Consumption Differential Curve With (different) Incorrect Key Guess



Countering DPA

- First approach
  - Make power consumption of device independent of the data processed
    - Detached power supplies
    - Logic styles with a data independent power consumption
    - Noise generators
    - Insertion of random delays
  - Methods costly, not in tune with normal CAD methods

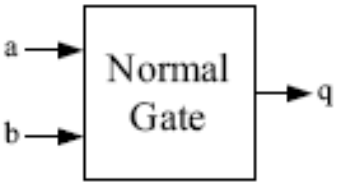
Countering DPA

- Second approach
  - Randomize intermediate results
  - Idea: power consumption of randomized data uncorrelated to actual intermediate results
  - Masking: applied at algorithm level or gate level

Gate Level Masking

- No wires store values correlated to intermediate algorithm result
- Converting unmasked digital circuit to a masked version can be automated

Why are normal gates susceptible to DPA?



a	b	q	Energy
$0 \to 0$	$0 \to 0$	$0 \to 0$	$E_{0\to0}$
$0 \to 0$	$0 \to 1$	$0 \to 0$	$E_{0\to0}$
$0 \to 0$	$1 \to 0$	$0 \to 0$	$E_{0\to0}$
$0 \to 0$	$1 \to 1$	$0 \to 0$	$E_{0\to0}$
$0 \to 1$	$0 \to 0$	$0 \to 0$	$E_{0\to0}$
$0 \to 1$	$0 \to 1$	$0 \to 1$	$E_{0\to1}$
$0 \to 1$	$1 \to 0$	$0 \to 0$	$E_{0\to0}$
$0 \to 1$	$1 \to 1$	$0 \to 1$	$E_{0\to1}$

a	b	q	Energy
$1 \to 0$	$0 \to 0$	$0 \to 0$	$E_{0\to0}$
$1 \to 0$	$0 \to 1$	$0 \to 0$	$E_{0\to0}$
$1 \to 0$	$1 \to 0$	$1 \to 0$	$E_{1\to0}$
$1 \to 0$	$1 \to 1$	$1 \to 0$	$E_{1\to0}$
$1 \to 1$	$0 \to 0$	$0 \to 0$	$E_{0\to0}$
$1 \to 1$	$0 \to 1$	$0 \to 1$	$E_{0\to1}$
$1 \to 1$	$1 \to 0$	$1 \to 0$	$E_{1\to0}$
$1 \to 1$	$1 \to 1$	$1 \to 1$	$E_{1\to1}$

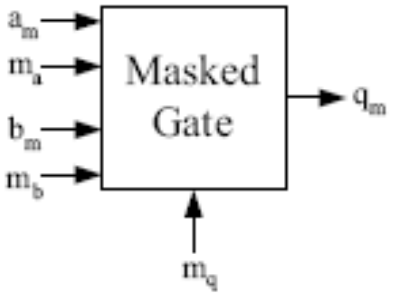
1. Attacker measures large number of power traces
2. Splits traces into two groups when $q = 0$ and when $q = 1$ at end of clock cycles
3. Expected means are not in general equal, leading to DPA attacks (spikes in the differential trace)
4. Here, means of the energies of the groups are:

$$E(q=0) = \frac{3E_{1\to0} + 9E_{0\to0}}{12}$$

$$E(q=1) \quad \frac{(3E_{0\to1} + E_{1\to1})}{4}$$

Since, $E(q=0) \neq E(q=1)$, DPA attack is possible

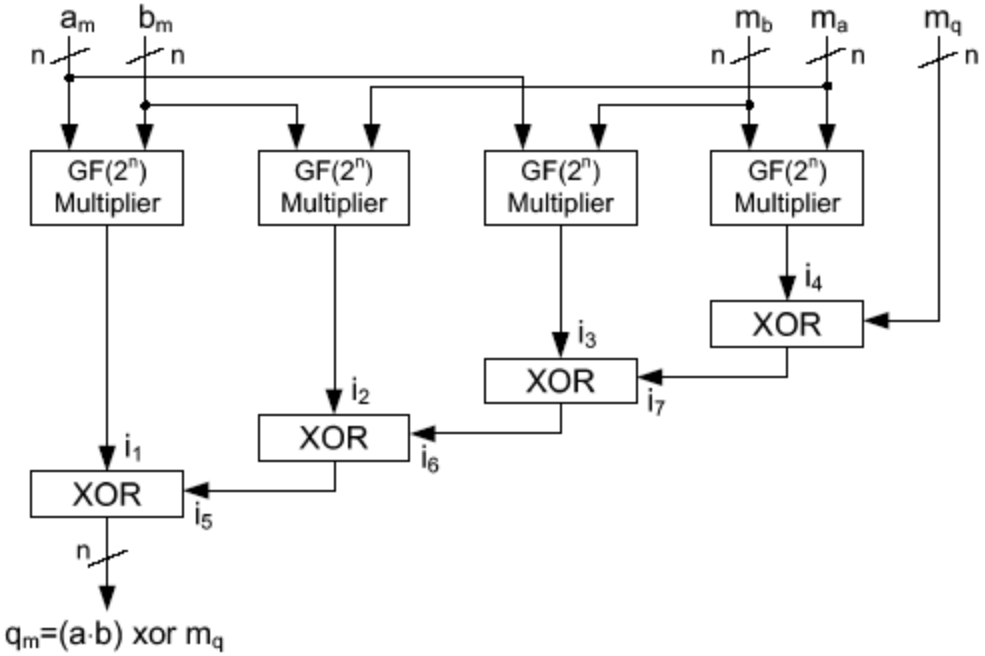
Masked AND Gate



$$a_m = a \oplus m_a$$
$$b_m = b \oplus m_b$$
$$q_m = q \oplus m_q$$
$$q = f(a,b)$$
$$q_m = \hat{f}(a_m, m_a, b_m, m_b, m_q)$$

1. There are $4^5 = 1024$ possible input transmissions that can occur
2. It turns out that the expected value of the energy required for the processing of $q = 0$ and $q = 1$ are identical
3. Thus protected against DPA, under the assumption that the CMOS gates switch only once in one clock cycles
4. (But we know there are glitches, and so the output of gates swing a number of times before reaching a steady state. Hence... the argument continues.)

Masked Multiplier



Masking is not perfect

- Xor gates can leak information about unmasked values
- Xor gates do not change output when both the inputs change value simultaneously or within a small time
- Amperage of xor gates depend on signal time arrival
- Time delays are related to the unmasked values
- (Masked circuits may still be vulnerable to DPA, because of signal delay.)