Security Engineering

Hardware security 1. Locks, alarms and seals. Hardware tamper resistance, differential fault analysis, differential power analysis.

Physical Security

- Locks, and walls, will be some part of your infrastructure at some level
- While the techniques are simpler than digital security, the weaknesses are often as subtle.
- Deter-detect-alarm-delay-respond

Who do we need to be secure against?

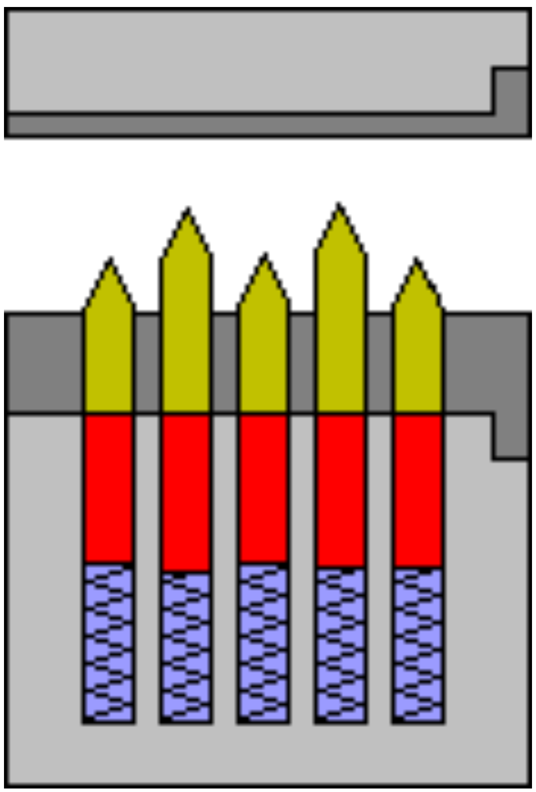
- Derek – 19-year old addict
- Charlie – 40-year old with 7 convictions
- Bruno – “gentleman criminal”
- Abdurrahman – head of a dozen agents

- Unskilled -> Skilled -> Highly Skilled with help -> Highly Skilled with resources

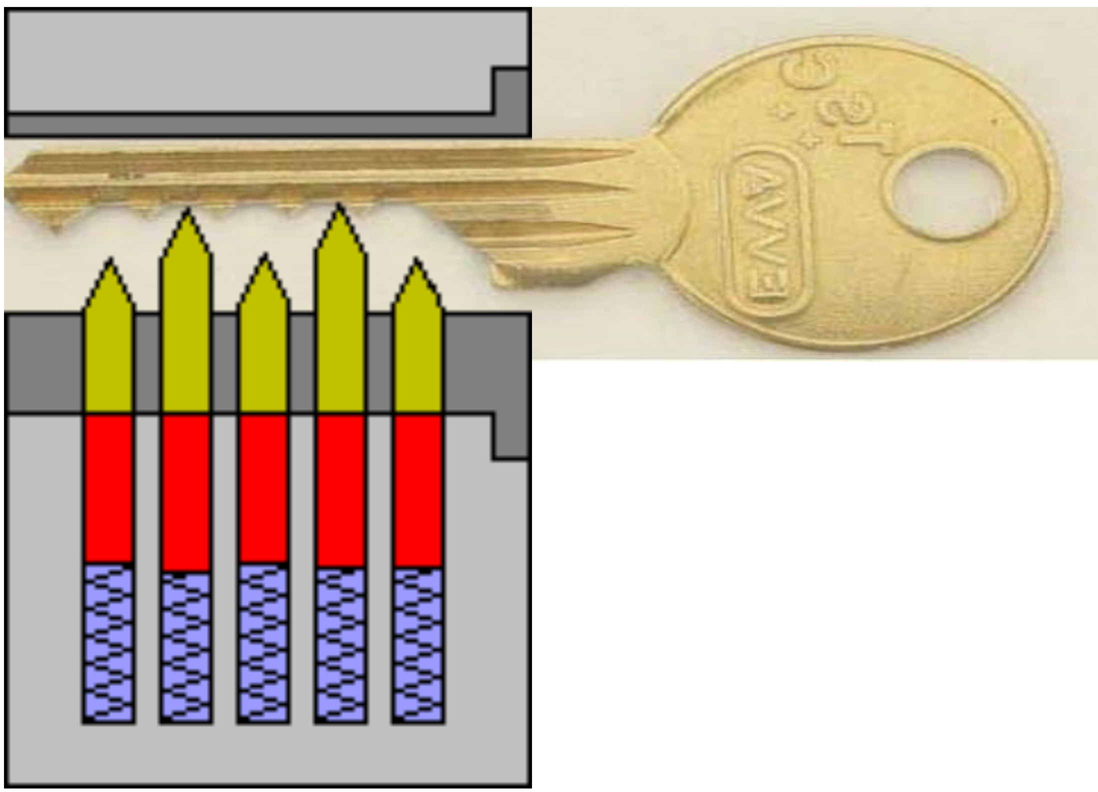
What are you trying to achieve?

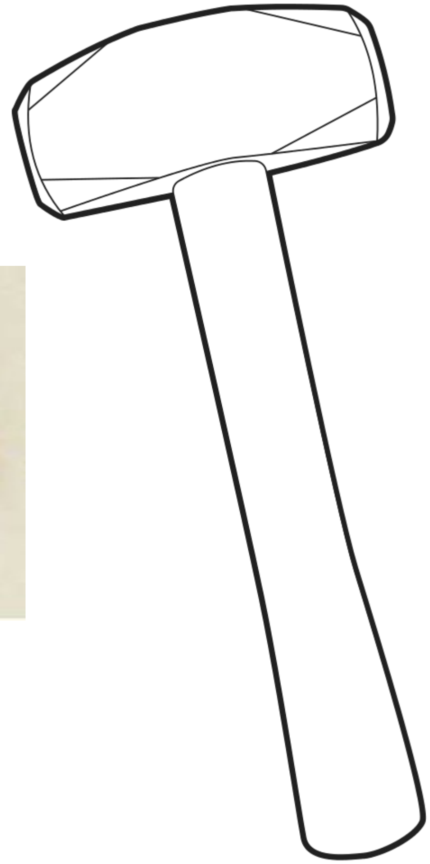
- Deterrence or just redistribution of crime?
- Are you really trying to protect your safe full of money, or your employees' lives?
- Don't just focus on the exciting threats
- Off-the-shelf product standards might use unusual/outdated assumptions

Locks

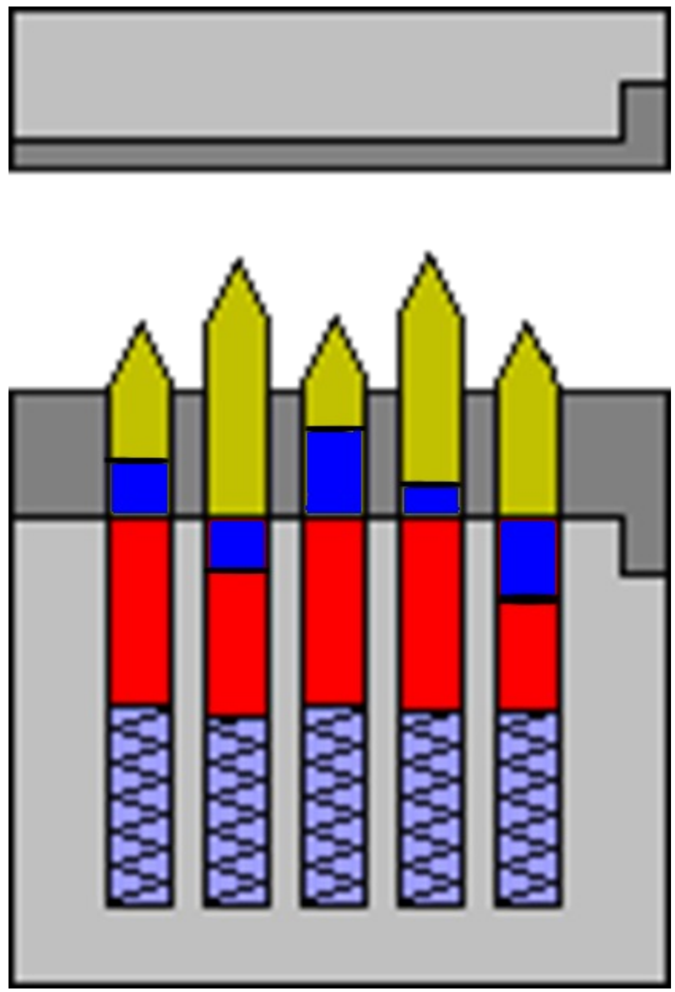


Bumping

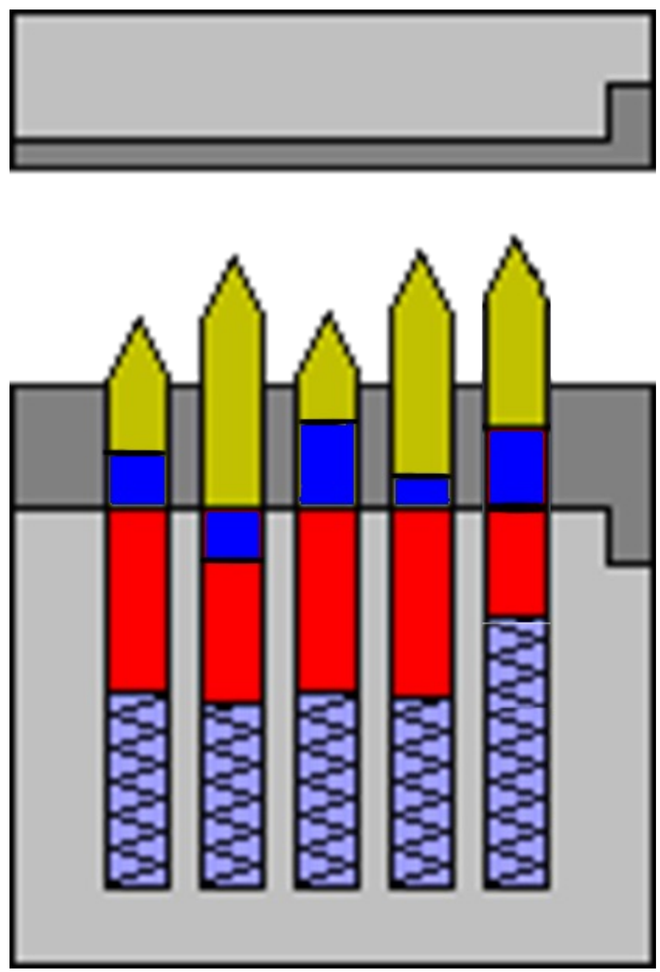




Master-Key Attacks



Master-Key Attacks



Electronic Locks

- E.g. wireless smart cards and card readers using challenge-response protocols
- Mifare Classic: Vulnerable but still widely deployed!
- All the usual crypto issues apply: weak ciphers, bad random number generators, short keys…

Alarms

- Deter-detect-alarm-delay-respond.
- Timeliness very important: if your criminal can get away before the security arrive, there’s no point!
- Don’t get blinded by the “Titanic Effect”

Types of Sensor

- Vibration: fences, footsteps
- Switches: doors/windows
- Infrared heat detection
- Motion detectors e.g. ultrasonics
- Movement sensors e.g. optical cables
- Invisible barriers of light beams
- Pressure pads
- Video cameras, possibly triggered by above

Alarms: Challenges

- False positives: Hurricane? Thunderstorm? Loud lorry?
- Denial of service attacks: keep triggering the alarm till the guards stop listening!
- Choosing a good combination of sensors is key
- Deter-detect-alarm-delay-respond.
- Feature interactions are difficult: if your fire alarm goes off, should you ignore your infrared heat intrusion detectors?

Alarms: Challenges (II)

- Spoofing of "liveness" signals
- Fix: Bury your cables in concrete, or use cryptography?
- Denial of service (II): cut your rivals' phone lines, then wait for the police to come and go again?
- Even if your own infrastructure is buried in concrete, what about the kerbside box your network goes through?

Who watches the watchmen?

- Bribery and corruption of your guards is often an issue.
- Which is worth more: your treasures or your guards' lives?
- Will dual controls help? Yes for bribery, less so for coercion
- An extreme case: prisons. *"How would I do this differently if half my staff were convicts on day release?"*
- Who might you have to contend with: just thieves, or also angry customers, spouses, ex-employees? Shooters?

Lessons

- Locks can be defeated, so alarms matter
- DoS is hard and important.
- Integrate detect-alarm-delay-respond
- Defence in depth
- Perimeter is least reliable and most important.
- Hard to keep guards alert under false alarms.
- Don't design for Charlie to keep about Bruno!
- You'll need specialist subcontractors, but can't leave everything to them, due to integration failures.

Seals and Tamper Resistance

Inspection

- Primary: untrained, possibly negative motivation
- Secondary: competent and motivated, performed in the field
- Tertiary: Full lab with experts

- Standards: FIPS 140 levels 1-4 (V1,2,3), ISO 19790

Security Printing

- Simultan presses, intaglio, letterpress, embossing, watermarks, microprinting, metal threads…
- Primary vs Secondary vs Tertiary inspectors
- Race against the forgers – add new features before your secondary inspectors get fooled

Seals

Tamper Resistance

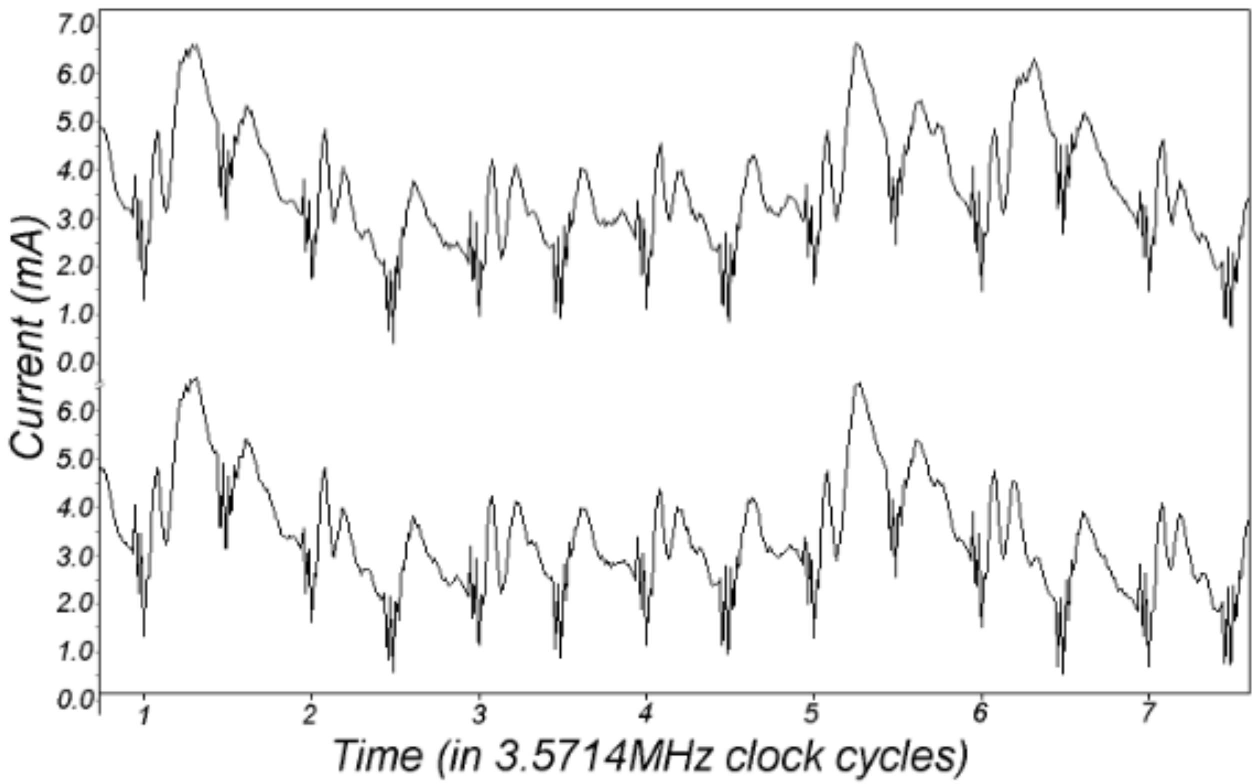
- Will your users (or anyone who can get hold of your device) be motivated to attack your device, and if so, can they attack your ecosystem?
- What are you protecting: authentication, service control, trusted execution, accessory control, manufacturing control?

Hardware Security Modules (HSMs)

Side Channels in HSMs

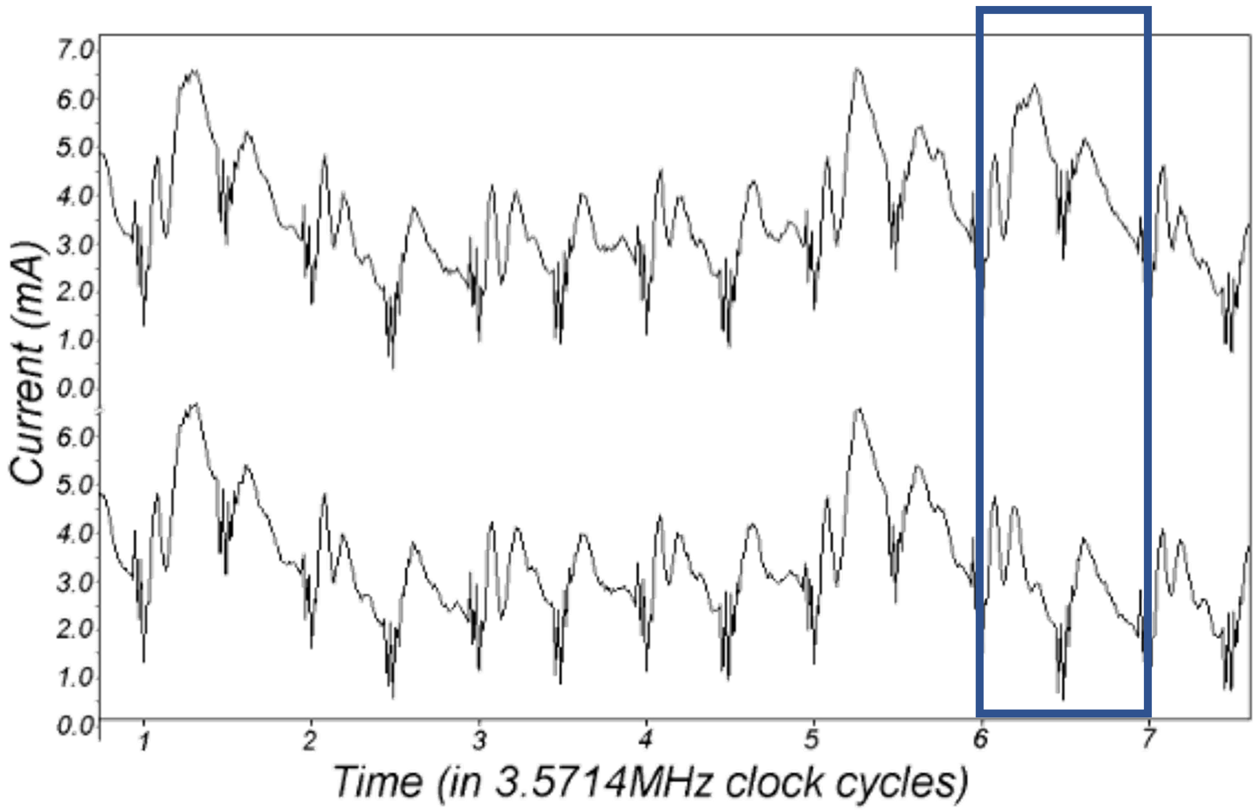
- Can we recover the key even if the device has been switched off – is the wiping mechanism reliable?
- Yes! Memory Remanence – they key will leave an imprint on the SRAM cells!
- Also, the SRAM won't wipe straight away if the power is cut – Cold Temperatures, and Cold Boot Attacks

Side Channels in Smart Cards: Power Analysis



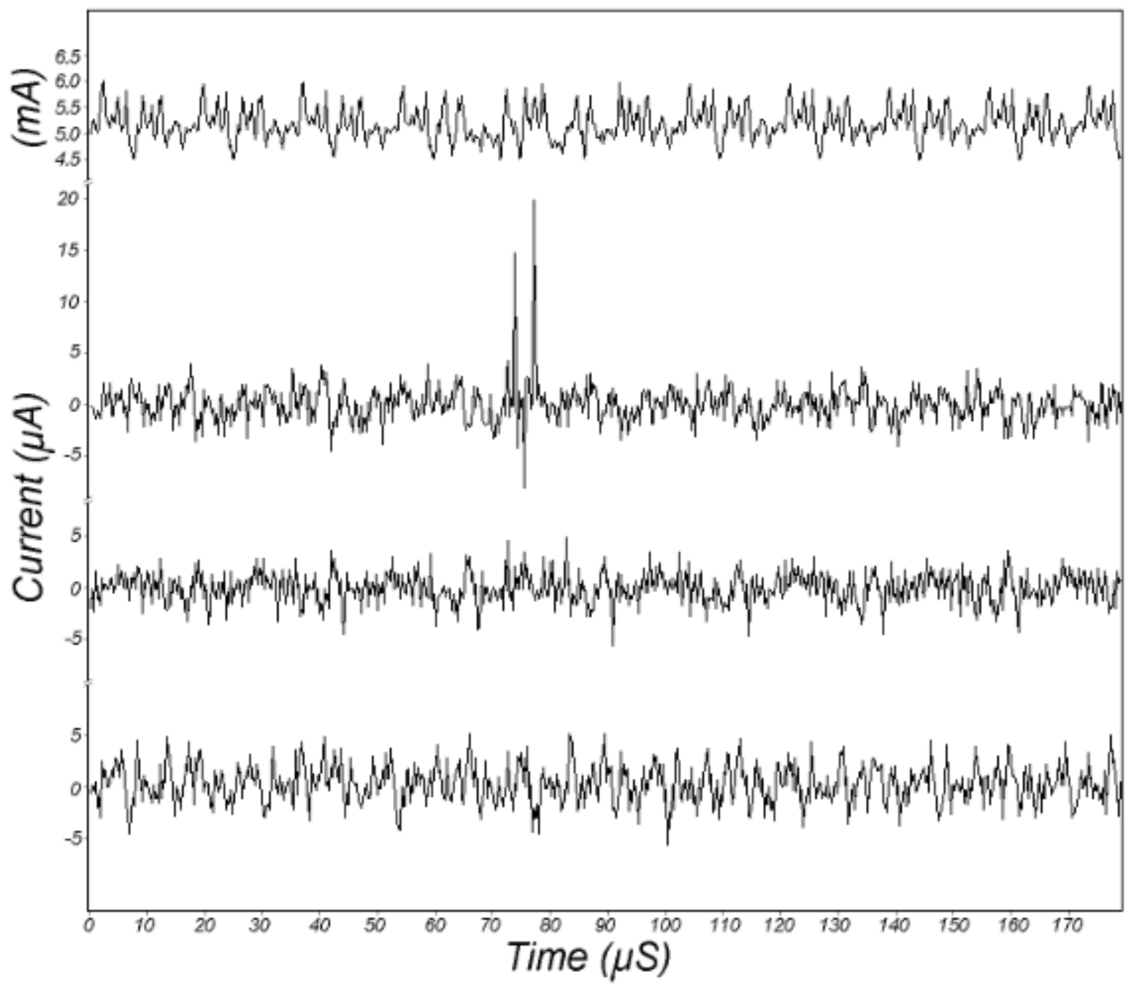
From Differential Power Analysis, Kocher, Jaffe and Jun, CRYPTO ‘99

Side Channels in Smart Cards: Power Analysis



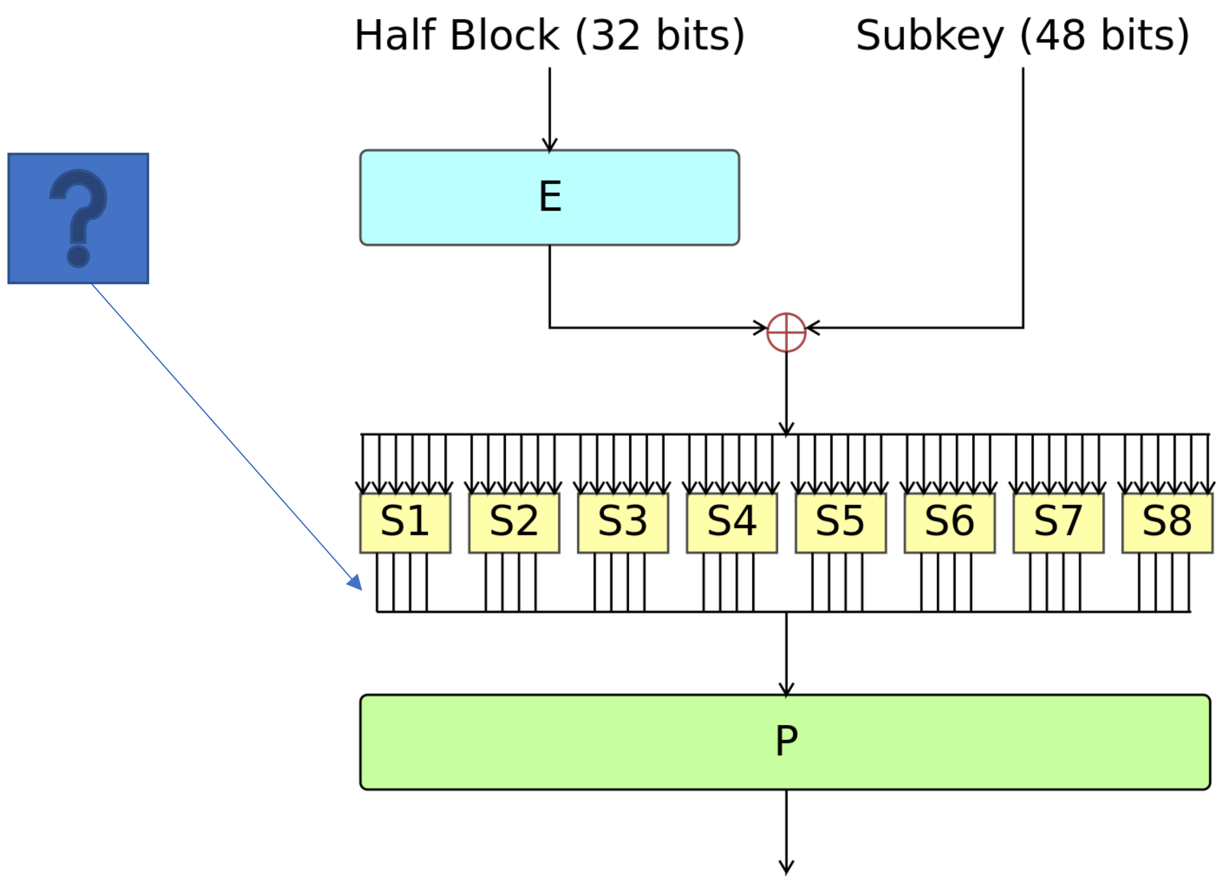
From Differential Power Analysis, Kocher, Jaffe and Jun, CRYPTO '99

Differential Power Analysis



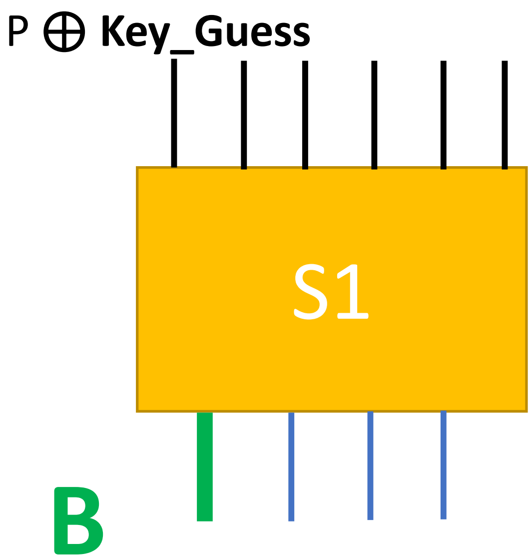
From Differential Power Analysis, Kocher, Jaffe and Jun, CRYPTO '99

DPA on DES



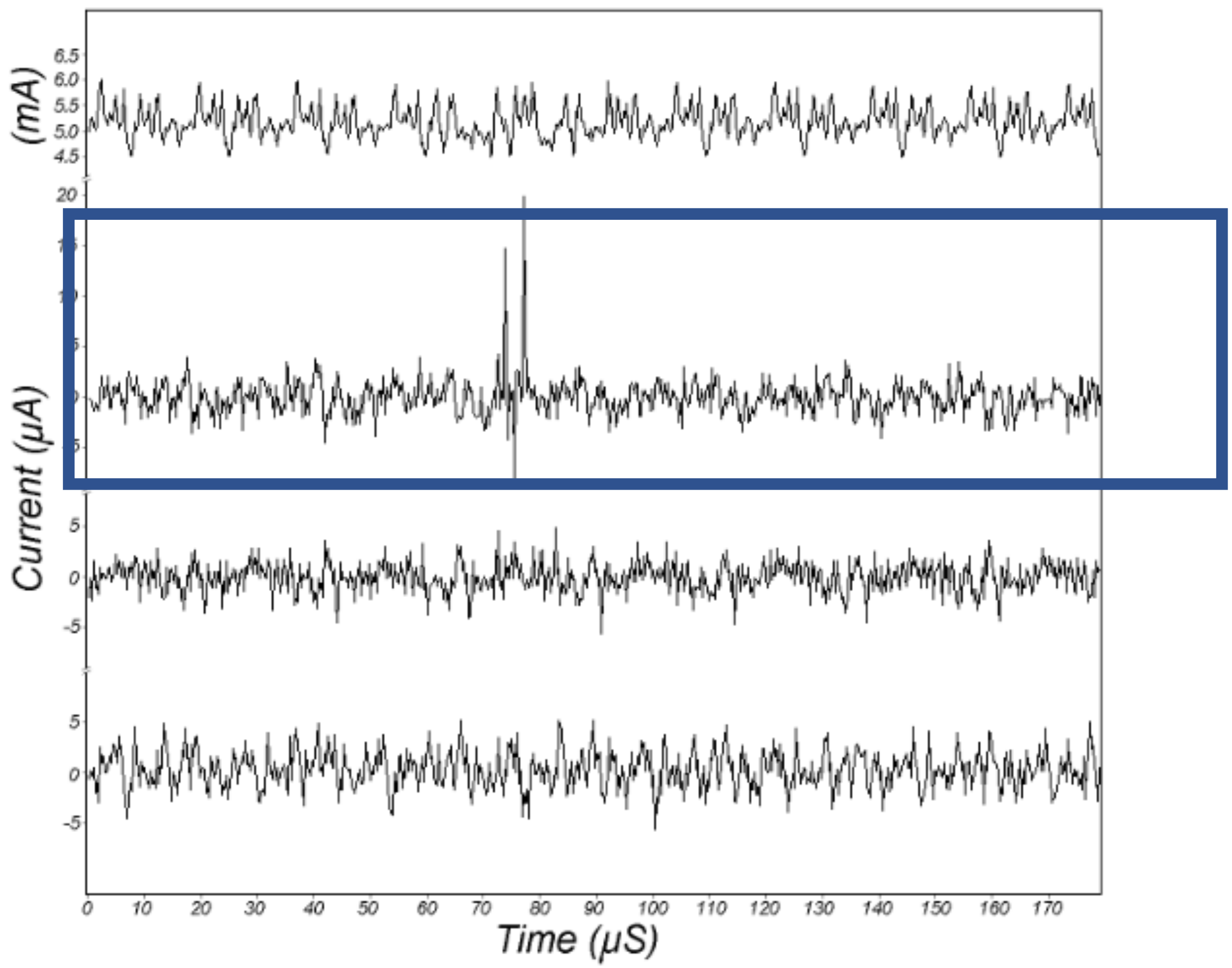
DPA on DES

Guessed Key Input xor Inferred Half Block



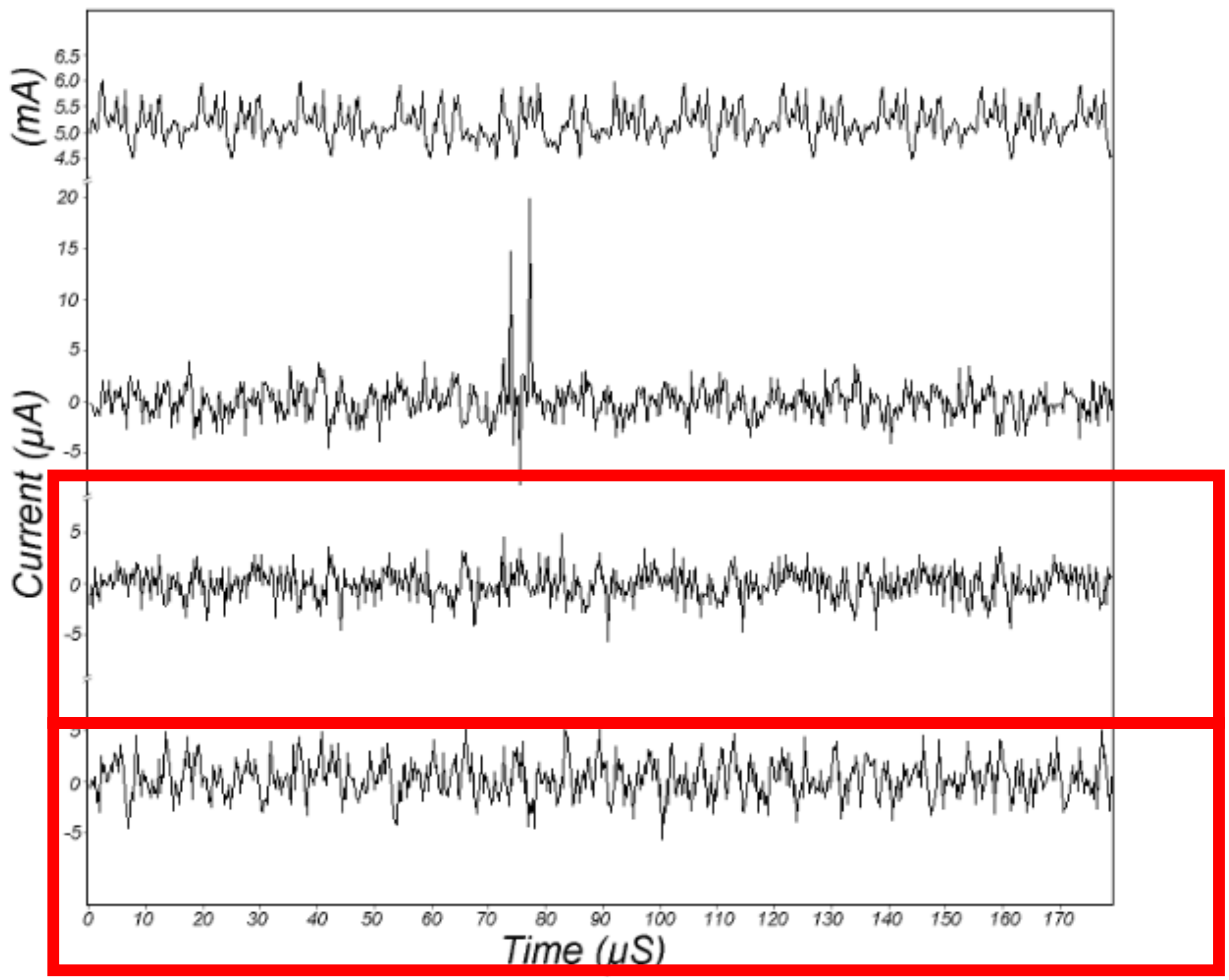
Plaintext	Trace
0x12345678…	
0x898979AB…	
0xDE424567…	
0XA0003341…	

Differential Power Analysis



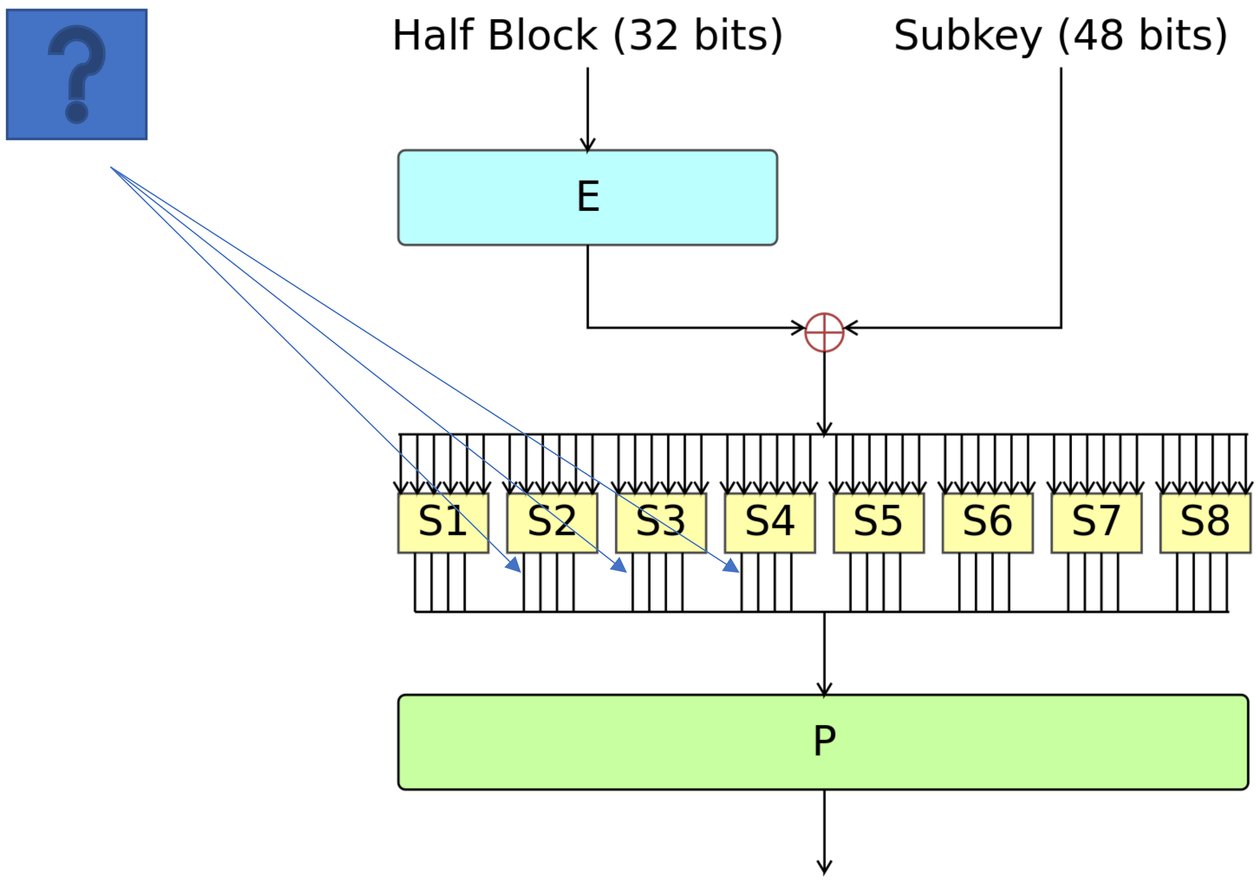
From Differential Power Analysis, Kocher, Jaffe and Jun, CRYPTO '99

Differential Power Analysis



From Differential Power Analysis, Kocher, Jaffe and Jun, CRYPTO '99

DPA on DES



Fault Analysis

- Computers are really analog devices that behave mostly digitally.
- What about with an attacker able to control voltage / with a laser?
- You can cause faults, and security vulnerabilities too…

Fault Analysis on RSA Signatures

Sig = Msg^d (mod n)

n = public key = p * q, two (secret) prime numbers

d = private key (a function of p and q)

Fault Analysis on RSA Signatures

Sig = Msg^{d} (mod n)

n = public key = p * q, two (secret) prime numbers

d = private key (a function of p and q)

Faster to calculate by combining:

Sig1 = Msg^{dp} (mod p)

Sig2 = Msg^{dq} (mod q)

Fault Analysis on RSA Signatures

What if we inject an error in the second one?

Sig1 = Msg^{dp} (mod p)

Sig2’ = Msg^{dq} (mod q)

Sig’ = CRT(Sig1,Sig2’)

Fault Analysis on RSA Signatures

What if we inject an error in the second one?

Sig1 = Msg^{dp} (mod p)

Sig2' = Msg^{dq} (mod q)

Sig' = CRT(Sig1,Sig2')

Msg = Sig'^{e} (mod p)

Msg != Sig'^{e} (mod q)

(e is public exponent)

Fault Analysis on RSA Signatures

What if we inject an error in the second one?

Msg = Sig'^{e} (mod p)

Msg != Sig'^{e} (mod q)

(Sig'^{e} – Msg) is divisible by p

(Sig'^{e} – Msg) is not divisible by q

Fault Analysis on RSA Signatures

What if we inject an error in the second one?

$(Sig'^e - Msg)$ is divisible by p

$(Sig'^e - Msg)$ is not divisible by q

So, $p = GCD(Sig'^e - Msg, n)$

– much simpler than prime_factor(n)

Fault Analysis on RSA Signatures

What if we inject an error in the second one?

(Sig'^{e} – Msg) is divisible by p

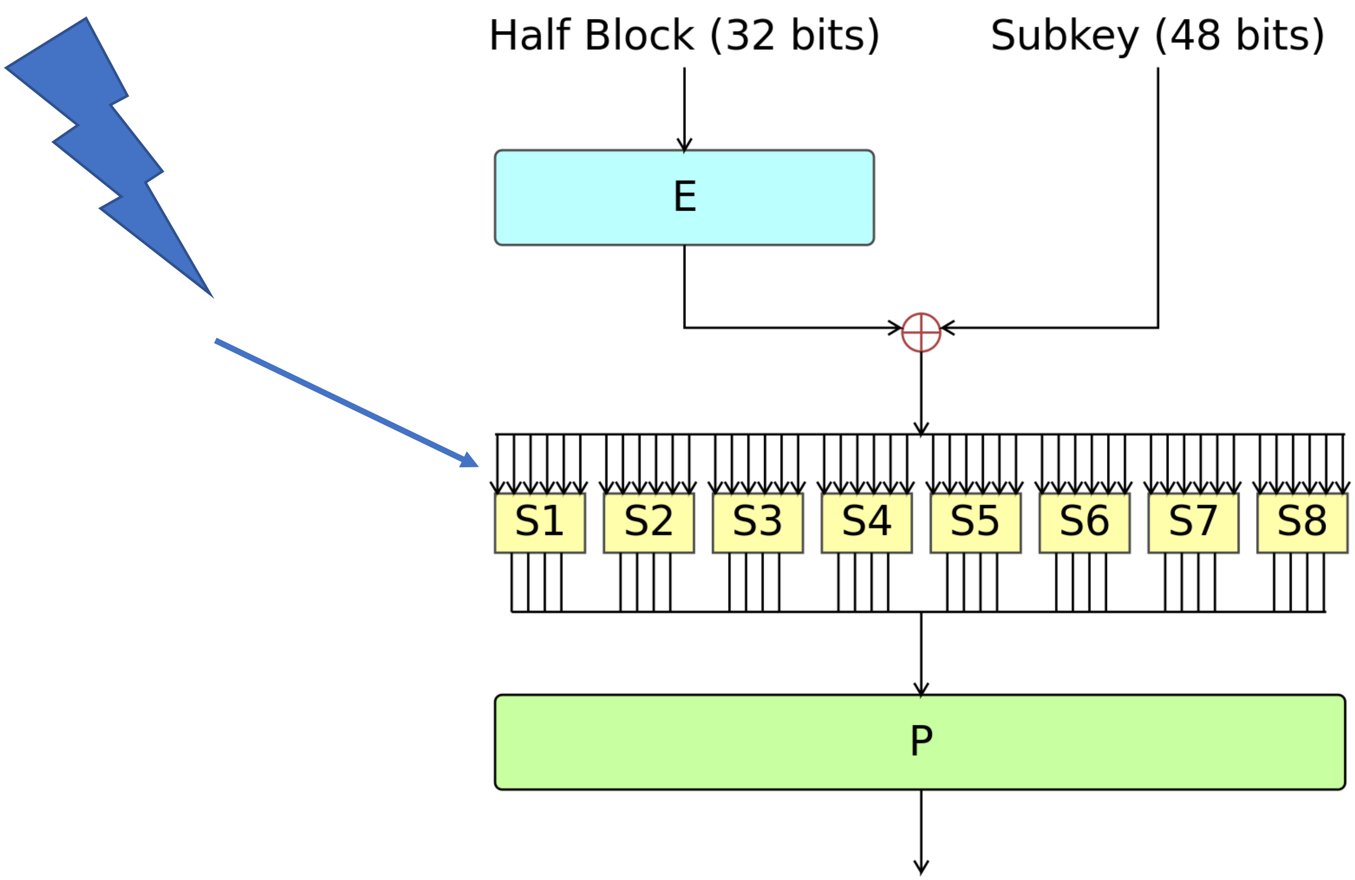
(Sig'^{e} – Msg) is not divisible by q

So, p = GCD(Sig'^{e} – Msg, n)

n = p*q

So q = n/p

Differential Fault analysis on AES



Tamper Resistance: The Moral

- If someone can benefit by physically subverting your system, and that attack can scale, you need to pay attention to physical device properties
- Standards are out-of-date, and manufacturer incentives often misaligned
- You need to know enough about these attacks to work out whether they are valid for your threat model.

Further Reading

- Security Engineering Chapter 13: Locks and Alarms
- Security Engineering Chapter 18: Tamper Resistance
- Security Engineering Chapter 19: Side Channels