Side Channel Attacks and Countermeasures, Countermeasures for Embedded Microcontrollers

Mohammad Tehranipoor

ECE4095/6095: Hardware Security & Test
University of Connecticut
ECE Department

Outline

- Introduction
- Side-Channel Emissions
- Attacks Using Side-Channel Information and Countermeasures
- Side-Channel Attacks on Microcontrollers and Countermeasures

Introduction

- Classic cryptography views the secure problems with mathematical abstractions
- The classic cryptanalysis has had a great success and promise
  - Analyzing and quantifying crypto algorithms' resilience against attacks
- Recently, many of the security protocols have been attacked through physical attacks
  - Exploit weaknesses in the cryptographic system hardware implementation aimed to recover the secret parameters

Side-Channel Emissions

- Side-Channel attacks aim at nonprime, side-channel inputs and outputs, bypassing the theoretical strength of cryptographic algorithms
- Five commonly exploited side-channel emissions:
  - Power Consumption
  - Electro-Magnetic
  - Optical
  - Timing and Delay
  - Acoustic

Side-Channel Emissions

- Power Consumption -- Logic circuits typically consume differing amounts of power based on their input data.
- Electro-Magnetic -- EM emissions, particularly via near-field inductive and capacitive coupling, can also modulate other signals on the die.
- Optical -- The optical properties of silicon can be modulated by altering the voltage or current in the silicon.
- Timing and Delay -- Timing attacks exploit data-dependent differences in calculation time in cryptographic algorithms.
- Acoustic -- The acoustic emissions are the result of the piezoelectric properties of ceramic capacitors for power supply filtering and AC to DC conversion.

Attacks Using Side-Channel Information

- Hardware Targets
- Attack Model
- Physical Attack Phases
- Attack Classification
- General Countermeasures
- Specific Attack Implementation and Corresponding Countermeasures

Hardware Targets

- Two common victims of hardware cryptanalysis are smart cards and FPGAs
  - Attacks on smart cards are applicable to any general purpose processor with a fixed bus architecture.

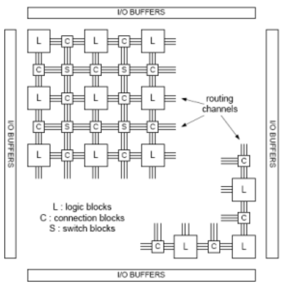
  - Attacks on FPGAs are also reported. FPGAs represent application specific devices with parallel computing opportunities.

Smart Cards

- Smart cards have a small processor (8bit in general) with ROM, EEPROM and a small RAM
- Eight wires connect the processor to the outside world
- Power supply: no internal batteries
- Clock: no internal clock
- Typically equipped with a shield that destroys the chip if a tampering happens

FPGAs

- FPGAs allow parallel computing
- Multiple programmable configuration bits



Attack Model

- Consider a device capable of implementing the cryptographic function
- The key is usually stored in the device and protected
- Modern cryptography is based on Kerckhoffs's assumption → all of the data required to operate a chip is entirely hidden in the key
- Attacker only needs to extract the key

Physical Attack Phases

- Physical attacks are usually composed of two phases:
  - **Interaction phase**: interact with the hardware system under attack and obtain the physical characteristics of the device
  - **Analysis phase**: analyze the gathered information to recover the key

Principle of divide-and-conquer attack

- The divide-and-conquer(D&C) attack attempt at recovering the key by parts
- The idea is that an observed characteristic can be correlated with a partial key
  - The partial key should be small enough to enable exhaustive search
- Once a partial key is validated, the process is repeated for finding the remaining keys
- D&C attacks may be iterative or independent

Attack Classification

- Invasive vs. noninvasive attacks
- Active vs. passive attacks
  - Active attacks exploit side-channel inputs
  - Passive attacks exploit side-channel outputs
- Simple vs. differential attacks
  - Simple side-channel attacks directly map the results from a small number of traces of the side-channel to the operation of DUA
  - Differential side-channel attacks exploit the correlation between the data values being processed and the side-channel leakage

General Countermeasures

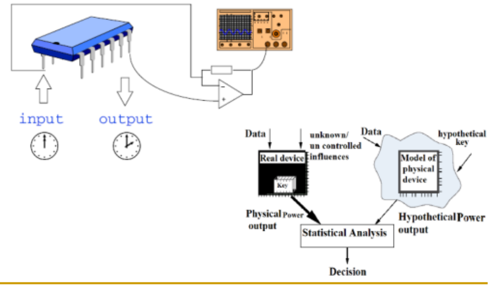
- Hiding -- reduce the SNR by either increasing the noise or reducing the signal
  - Noise Generators, Balanced Logic Styles, Asynchronous Logic, Low Power Design and Shielding
- Masking/Blinding -- remove the correlation between the input data and the side-channel emissions from intermediate nodes in the functional block
- Design Partitioning -- separate regions of the chip that operate on plaintext from regions that operate on ciphertext
- Physical Security and Anti-Tamper -- denial of proximity, access, and possession

Specific Attack Implementation & Countermeasures

- Power Attacks
- Timing Attacks
- EMA Attacks

Power attacks

- Measure the circuit's processing time and current consumption to infer what is going on inside it.



Measuring Phase

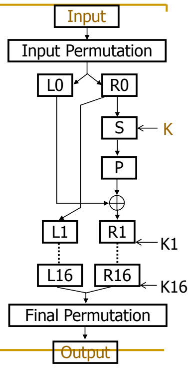
- The task is usually straightforward
  - Easy for smart cards: the energy is provided by the terminal and the current can be read
- Relatively inexpensive (<$1000) equipment can digitally sample voltage differences at high rates (1GHz++) with less than 1% error
- Device's power consumption depends on many things, including its structure and data being processed

Simple Power Analysis (SPA)

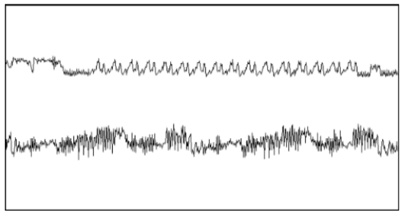
- Originally proposed by Paul Kocher, 1996
- Monitor the device's power consumption to deduce information about data and operation
- Example: SPA on DES – smart cards
  - The internal structure is shown on the next slide
- Summary of DES – a block cipher
  - a product cipher
  - 16 rounds iterations
    - substitutions (for confusion)
    - permutations (for diffusion)
  - Each round has a *round key*
    - Generated from the user-supplied key

DES Basic Structure

- Input: 64 bits (a block)
- Li/Ri– left/right half (32 bits) of the input block for iteration i– subject to substitution S and permutation P
- K - user-supplied key
- Ki - round key:
  - 56 bits used +8 unused
    (unused for encryption but often used for error checking)
- Output: 64 bits (a block)
- Note: Ri becomes L(i+1)
- All basic op's are simple logical ops
  - Left shift / XOR



SPA on DES (cont'd)



- The upper trace – entire encryption, including the initial phase, 16 DES rounds, and the initial permutation
- The lower trace – detailed view of the second and third rounds
- The power trace can reveal the instruction sequence

SPA on DES (cont'd)

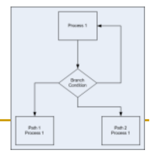
- SPA can be used to break cryptographic implementations (execution path, instruction, key change, etc.)
  - **DES key schedule:** Involves rotating 28-bit key registers
  - **DES permutation:** involves conditional branching
  - **Comparison:** Involves string and memory comparison operations performing a conditional branch when a mismatch is found
  - **Multipliers:** Involves modular multiplication – The leakage function depends on the multiplier design but strongly correlated to operand values and Hamming weights
  - **Exponentiators:** Involves squaring operation and multiplication
- SPA Countermeasure:
  - Avoid procedures that use secret intermediates or keys for conditional branching operation

SPA on DES (cont'd)

- The DES structure and 16 rounds are known
- Instruction flow depends on data → power signature
- Example: Modular exponentiation in DES is often implemented by square and multiply algorithm
- Typically the square operation is implemented differently compared with the multiply (for speed purposes)
- Then, the power trace of the exponentiation can directly yields the corresponding value
- All programs involving conditional branching based on the key values are at risk!

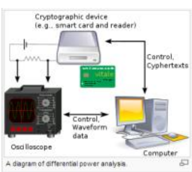
square and multiply algorithm

```
exp1(M, e, N)
{  R = M
   for (i = n-2 down to 0)
   {  R = R^2 mod N
      if (ith bit of e is a 1)
         R = R·M mod N }
   return R }
```



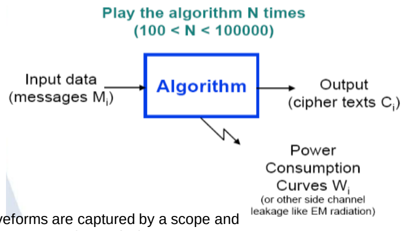
Differential power analysis (DPA)

- SPA targets variable instruction flow
- DPA targets data-dependence
  - Different operands presents different power
- Difference between smart cards and FPGAs
  - In smart cards, one operation running at a time
    - → Simple power tracing is possible
  - In FPGAs, typically parallel computations prevent visual SPA inspection → DPA



DPA

- DPA can be performed on any algorithm that has the operation $\beta=S(\alpha\oplus K)$,
  - α is known and K is the segment key



The waveforms are captured by a scope and sent to a computer for analysis

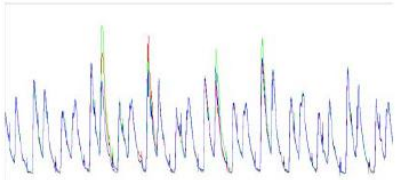
Assumption: Either Plaintext or Cipher is known

What is available after acquisition?

- After data collection, what is available ?
  - N plain and/or cipher random texts

00	B688EE57BB63E03E
01	185D04D77509F36F
02	C031A0392DC881E6 ...

  - N corresponding power consumption waveforms



DPA (cont'd)

- Assume the data are processed by a known deterministic function f (transfer, permutation…)
- Knowing the data, one can re-compute off line its image through f

M_i → **f** → $M'_i = f\,[M_i]$

- Now select a single bit among M' bits (in M' buffer)
- One can predict the true story of its variations

i	Message	bit
0	B688EE57BB63E03E	1
1	185D04D77509F36F	0
2	C031A0392DC881E6	1 ....

The bit will classify the wave w_i

- Hypothesis 1: bit is zero
- Hypothesis 2: bit is one
- A differential trace will be calculated for each bit!

Assumption: Attacker knows the algorithm well

DPA (cont'd)

- Partition the data and related curves into two packs, according to the selection bit value…

M_i → **f** → *bit (M_i') = 0* / *bit (M_i') = 1*

- … and assign -1 to pack 0 and +1 to pack 1

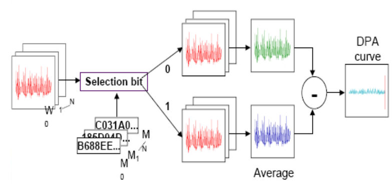
0	B688EE57BB63E03E	1	+1
1	185D04D77509F36F	0	-1
2	C031A0392DC881E6	1	+1 ...

- Sum the signed consumption curves and normalise
- <=> Difference of averages

$(N_0 + N_1 = N)$

$$DPA = \frac{\sum W_1}{N_1} - \frac{\sum W_0}{N_0}$$

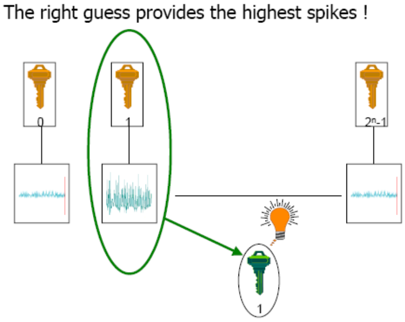
DPA (cont'd)



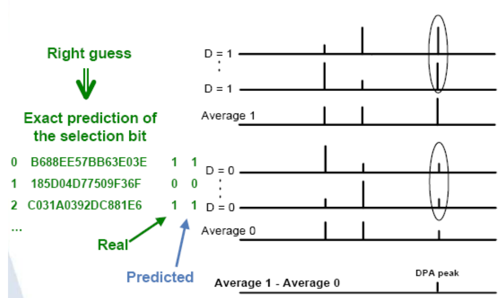
$$\Delta_n = \frac{\sum_{w_i \in S_0} w_i}{|S_0|} - \frac{\sum_{w_i \in S_1} w_i}{|S_1|}$$

DPA -- testing

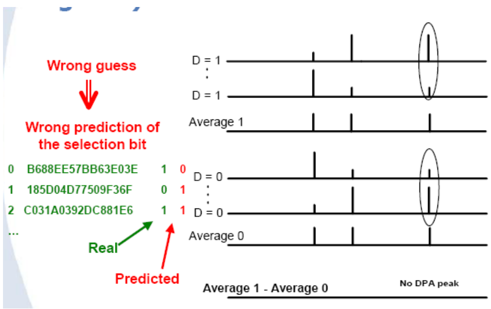
- The right guess provides the highest spikes !



DPA -- testing

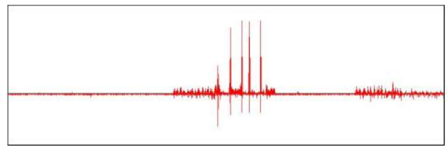


DPA – the wrong guess



DPA (cont'd)

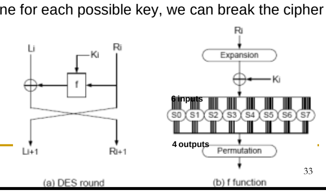
- The DPA waveform with the highest peak will validate the hypothesis



Example: DPA on DES

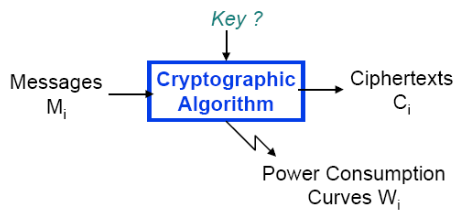
- Assumption: Attacker presumes detailed knowledge of the DES
- Divide-and-conquer strategy, comparing powers for different inputs
  - Record large number of inputs and record the corresponding power consumption
  - Start with round 15 -- We have access to R_{15}, that entered the last round operation, since it is equal to L_{16}
  - Take this output bit (called M'_i) at the last round and classify the curves based on the bit
    - 6 specific bits of R_{15} will be XOR'd with 6 bits of the key, before entering the S-box
    - By guessing the 6-bit key value, we can predict the bit b, or an arbitrary output bit of an arbitrary S-box output
  - Thus, with 16 partitions, one for each possible key, we can break the cipher much faster

A closer look at HW Implementation Of DES



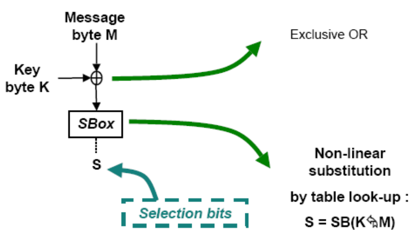
Attacking a secret key algorithm

- DPA works thanks to the perfect prediction of the selection bit
- How to break a key ?



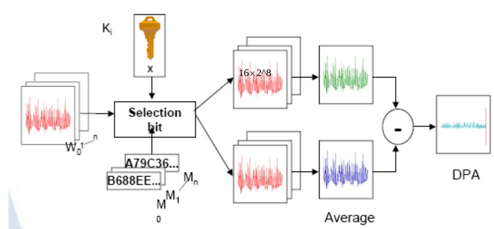
Typical DPA Target

- Basic mechanism in Secret Key algorithms (AES, DES...)



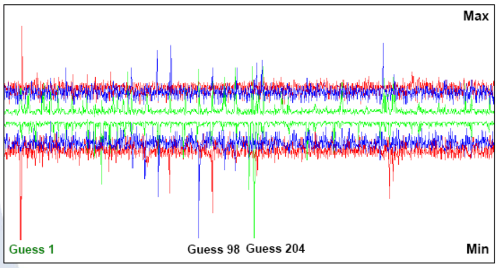
Example – DPA on AES

- Example : AES 128 bits key = 16 bytes K_i (i = 1 to 16)
  - Test 256 guesses per K_i with 256 DPA
  - 128 key bits disclosed with 16 x 256 = 4096 DPA (<< 2^{128} !)



Example – hypothesis testing

DPA on AES : 1^{st} round and 1^{st} byte (right guess = 1)

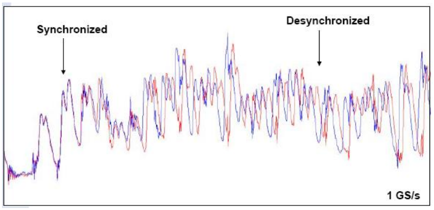


Anti-DPA countermeasures

- Applicative counter-measures : make message free randomization impossible !
  - Fix some message bytes
  - Constrain the variable bytes (ex : transaction counter)
- Decorrelate power curves from data
  - by hardware : current scramblers (additive noise)
  - by software : data whitening
- Desynchronise the N traces (curves misalignment)
  - software random delays
  - software random orders (ex : SBoxes in random order)
  - hardware wait states (dummy cycles randomly added by the CPU)
  - hardware unstable internal clock (phase shift)
- DPA is powerful, generic (to many algorithms) and robust (to model errors)...
- ... but there are counter-measures !

Anti-DPA

- Internal clock phase shift



Timing attacks

- Running time of a crypto processor can be used as an information channel
- The idea was proposed by Kocher, Crypto'96

- You put \$28 in one of the pots and \$10 in the other:

- Question: Compute
  - Blue * 10 + Red *7
  - Tell me if the result is odd or even.
- Is your answer enough to reveal what's in each pot?

Timing attacks (cont'd)

- Well, normally not :

28 * 7 + 10 * 10 = 296 is an even number

and

10 * 7 + 28 * 10 = 350 is also even...

- However, just by monitoring the time it takes to give the answer one can tell where each amount is!

RSA Cryptosystem

- Key generation:
  - Generate large (say, 2048-bit) primes p, q
  - Compute n=pq and $\varphi(n)=(p-1)(q-1)$
  - Choose small e, relatively prime to $\varphi(n)$
    - Typically, e=3 (may be vulnerable) or $e=2^{16}+1=65537$ (why?)
  - Compute unique d such that ed = 1 mod $\varphi(n)$
  - Public key = (e, n); private key = (d, n)
    - Security relies on the assumption that it is difficult to factor n into p and q
- Encryption of m: $c = m^e \bmod n$
- Decryption of c: $c^d \bmod n = (m^e)^d \bmod n = m$

How Does RSA Decryption Work?

- RSA decryption: compute y^x mod n
  - This is a modular exponentiation operation
- Naive algorithm: square and multiply

Let $s_0 = 1$.
For $k = 0$ upto $w - 1$:
  If (bit k of x) is 1 then
    Let $R_k = (s_k \cdot y) \bmod n$.
  Else
    Let $R_k = s_k$.
  Let $s_{k+1} = R_k^2 \bmod n$.
EndFor.
Return (R_{w-1}).

Kocher's Observation

Let $s_0 = 1$.
For $k = 0$ upto $w - 1$:
  If (bit k of x) is 1 then
    Let $R_k = (s_k \cdot y) \bmod n$.
  Else
    Let $R_k = s_k$.
  Let $s_{k+1} = R_k^2 \bmod n$.
EndFor.
Return (R_{w-1}).

Whether iteration takes a long time depends on the k^{th} bit of secret exponent

This takes a while to compute

This is instantaneous

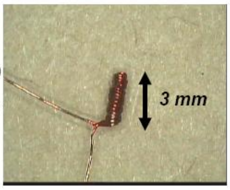
Outline of Kocher's Attack

- Idea: guess some bits of the exponent and predict how long decryption will take
- If guess is correct, we will observe correlation; if incorrect, then prediction will look random
  - This is a signal detection problem, where signal is timing variation due to guessed exponent bits
  - The more bits you already know, the stronger the signal, thus easier to detect (error-correction property)
- Start by guessing a few top bits, look at correlations for each guess, pick the most promising candidate and continue

Electromagnetic Power Analysis

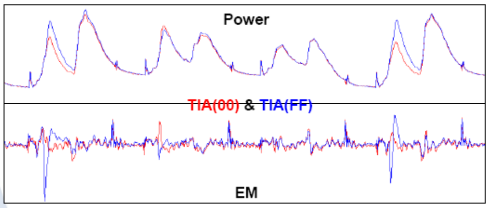
EMA – probe design

- Hamming distance model for information leakage
  Correlated to the number of flipping bits (CMOS, VLSI)
- Electrical transitions disturb EM near field (and its flow ϕ)
- Captation by inductive probe
  - Handmade solenoid $V = -\frac{d\phi}{dt}$
    (Diameter = 150 to 500 µm)
  - Difficult to calibrate
    (Bandwidth > 100 MHz, low voltage, parasitic effects)
  - Good acquisition chain required, but no Faraday cage
    (Sampling at 1GHz)



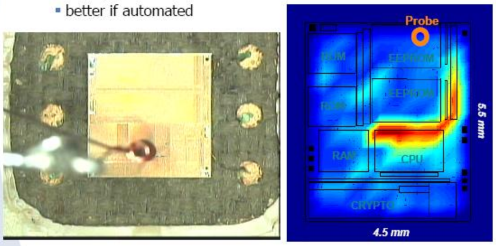
EMA signal

- Raw signals (TIA : transfer into accumulator instruction)
  - Power is less noisy
  - But EM signatures are sharper !

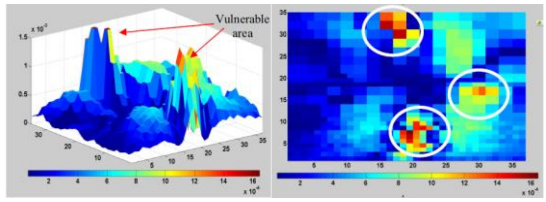


Spatial Positioning

- Horizontal cartography (XY plane)
  - to pinpoint instruction related areas
  - better if automated



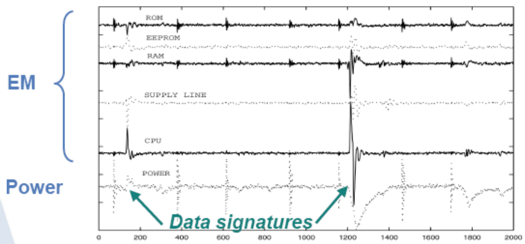
Spectral density of the chip surface



Spatial Positioning

- EM signals versus XY probe position

Differential traces between (00h ⊕ 00h) and (FFh ⊕ 00h) picked up at different locations



EMA (cont'd)

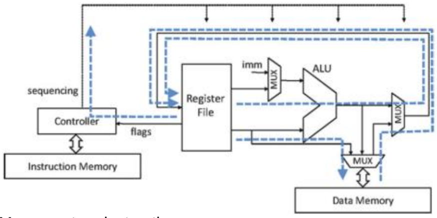
- Advantage of EMA versus PA
  - Local information more "data correlated"
  - EMA bypasses current smoothers
  - EMA goes through HW countermeasures: shields, randomized logic
- Drawbacks
  - Experimentally more complicated
  - Geometrical scanning can be tedious
  - Low level and noisy signals (decapsulation required)

Countermeasures

- Software (crypto routines)
  - Coding techniques
  - Same as anti DPA/SPA (data whitening…)
- Hardware (chip designers)
  - Confine the radiation (metal layer)
  - Blur the radiation (e.g. by an active emitting grid)
  - Reduce the radiation (technology trends to shrinking)
  - Cancel the radiation (dual logic)

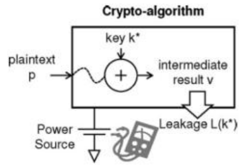
Side-Channel Attacks and Countermeasures for Embedded Microcontrollers

Source of side-channel leakage in a microcontroller



- Memory-store instructions
- Memory-load instructions
- Arithmetic instructions
- Control-flow instructions

Side-Channel Attacks on Microcontrollers



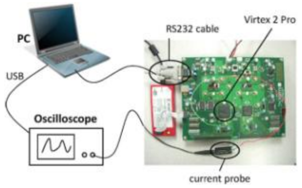
Objective: retrieve the internal secret key k^* of a crypto-algorithm

- The leakage caused by v is a function of the key value k^*, and it can be expressed as follows:

  $$L(k^*) = f_{k^*}(p) + \varepsilon$$

  The function f_{k^*} is dependent on the crypto-algorithm as well as on the nature of the implementation in hardware and software. The error ε is an independent noise variable.

Side-Channel Attacks on Microcontrollers



- The PC sends a sample plaintext to the PowerPC on the FPGA for encryption. During the encryption, the digital oscilloscope captures the power consumption from the board. After the encryption is completed, the PC downloads the resulting power trace from the oscilloscope, and proceeds with the next sample plaintext.

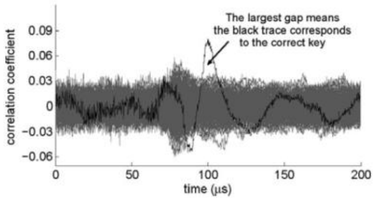
Correlation Power Analysis

- Two important aspects of a practical CPA:
  - The selection of the power model

    The power model is chosen so that it has a dependency on a part of the secret key. A good candidate is the output of the substitution step.
  - The definition of the attack success metric

    Measurements to Disclosure (MTD): the more measurements that are required to successfully attack a cryptographic design with side-channel analysis, the more secure that design is.

Practical Hypothesis Tests



- An example of 256 correlation coefficient traces. Around time 100 us, the black trace which corresponds to the correct key byte emerges from all the other 255 traces.

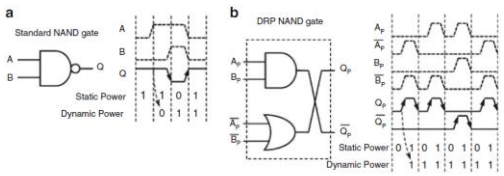
Side Channel Countermeasures for Microcontrollers

- Two different kinds of countermeasures:
  - Algorithm-Level Countermeasures

    Transform the C program so that the generation of dangerous side-channel leakage is avoided.
  - Architecture-Level Countermeasures

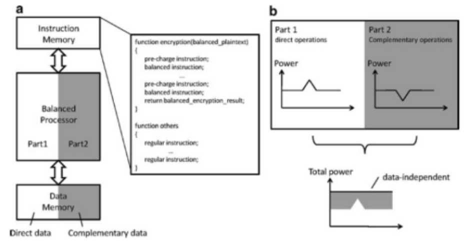
    Create a better microcontroller, for example using special circuit techniques, so that no side-channel leakage is generated.

Dual Rail Precharge



- (a) A CMOS standard NAND has data-dependent power dissipation; (b) A DRP NAND gate has a data-independent power dissipation
- DRP requires the execution of the direct and complementary data paths in parallel.

VSC: Porting DRP into software



- (a) Concept of balanced processor and VSC programming;
  (b) The balanced processor does not show side-channel leakage
- The power dissipation from the direct operation always has a complementary counterpart from the complementary operation. The sum of these two is a constant.

References

- [1] Mohammad Tehranipoor and Cliff Wang. Introduction to Hardware Security and Trust. Springer, pp.175-191, 263-281, 2012
- [2] Weaver J, Horowitz M (2007) Measurement of supply pin current distributions in integrated circuit packages. IEEE Electrical Performance of Electronic Packaging, October 2007
- [3] Kocher P, Jaffe J, Jun B (1999) Differential power analysis. In: 19th Annual International
- Cryptology Conference (CRYPTO), vol 2139. Springer-Verlag, Berlin, Heidelberg, New York, August 1999
- [4] Daemen J, Rijmen V (2002) The Design of Rijndael. Secaucus, NJ, USA: Springer, New York, Inc.
- [5] Tiri K, Verbauwhede I (2003) Securing encryption algorithms against DPA at the logic level: next generation smart card. In: CHES 2003, vol LNCS 2779, pp. 125–136
- [6] Biham E (1997) A fast new DES implementation in software. In: FSE'97: Proceedings of the 4th International Workshop on Fast Software Encryption. Springer, London, UK, pp. 260–272.
- [7] Chen Z, Sinha A, Schaumont P (2010) Implementing virtual secure circuit using a custom-instruction approach. In: Proceedings of the 2010 international conference on Compilers, architectures and synthesis for embedded systems, CASES'10. ACM, New York, NY, USA, pp. 57–66