Hardware Design for Cryptographers

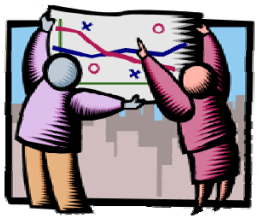
Ingrid Verbauwhede, Nele Mentens
ingrid.verbauwhede-at-esat.kuleuven.be

KU Leuven, ESAT- COSIC
Computer Security and Industrial Cryptography

Acknowledgements:
Current and former PhD students at UCLA and KU Leuven

Outline

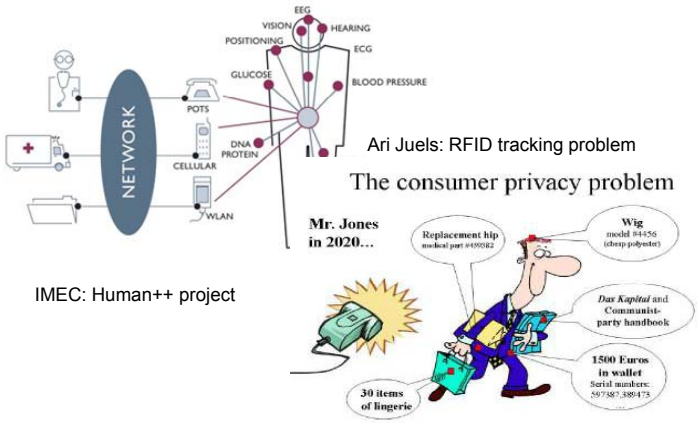
- Given: cryptographic algorithm
- Request: “efficient” hardware design
- This lecture about Efficiency
- Later this week: cost of side-channel resistance



Design Parameters

Embedded security:
Area, delay, power, energy,
physical security

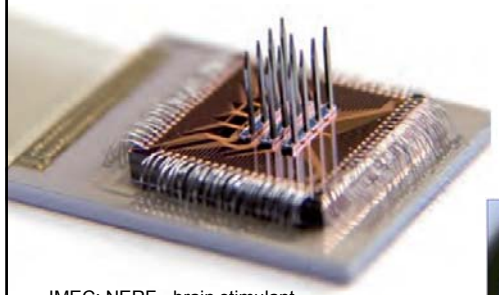
Embedded crypto everywhere



Ari Juels: RFID tracking problem

IMEC: Human++ project

Embedded crypto everywhere

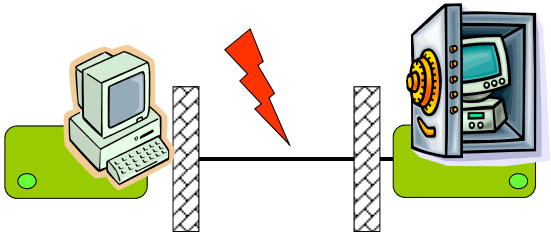


IMEC: NERF - brain stimulant

Deep Brain stimulation
[Sources: J. Rabaey, National Institutes of Health, Neurology Journal]

KUL - COSIC Summer School- 5 Šibenik Croatia, June 2014

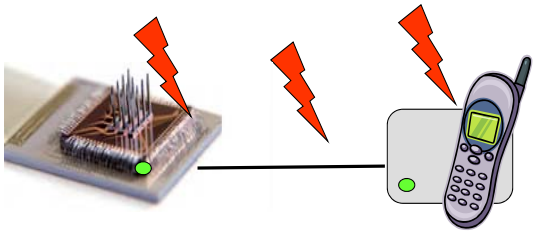
Embedded Security



Old Model (simplified view):
-Attack on channel *between* communicating parties
-Encryption and cryptographic operations in *black* boxes
-Protection by strong mathematic algorithms and protocols

KUL - COSIC Summer School- 6 Šibenik Croatia, June 2014

Embedded Security



New Model (also simplified view):
-Attack channel *and* endpoints
-Encryption and cryptographic operations in *gray* boxes
-Protection by strong mathematic algorithms and protocols
-Protection by secure implementation

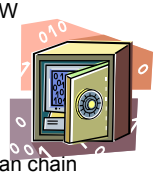
Need secure implementations not only algorithms

KUL - COSIC Summer School- 7 Šibenik Croatia, June 2014

Embedded Security

NEED BOTH

- Efficient, light-weight Implementation
  - Within power, area, timing budgets
  - Public key: 1024 bits RSA on 8 bit μC and 100 μW
  - Public key on a passive RFID tag



- Trustworthy implementation
  - Resistant to attacks
  - Active attacks: probing, power glitches, JTAG scan chain
  - Passive attacks: side channel attacks, including power, timing and electromagnetic leaks

KUL - COSIC Summer School- 8 Šibenik Croatia, June 2014

Cost definition

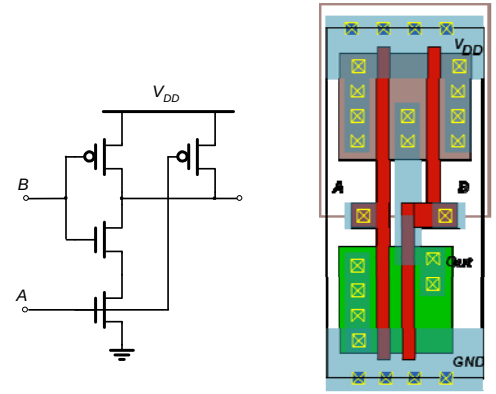
- Area
- Time: throughput versus latency
- Power, Energy
- Physical Security
- NRE (Non Recurring Engineering) cost

AREA

Area

- ASIC = Application Specific Integrated Circuit
  - Gate count
  - Unit = NAND gate = 4 transistors
- FPGA = Field Programmable Gate Area
  - Unit is LUT, flip-flops, see lecture Nele
- Embedded micro-controllers
  - Memory size = program size + data size

One Standard cell NAND gate



TIME

Clock frequency versus sample frequency
Throughput versus latency

Real-time, throughput, latency

- Throughput = associated with ***application***
  - Amount of data processed per time unit
  - Video: Gbits/sec, Internet: Gpackets/sec
  - ***Real-time sample rate***: HW has to work as fast as application dictates
- Latency = associated with ***application***
  - Delay from input to output
  - Measure of reaction speed or turn-around time
- High throughput and low latency don't go together

Clock Frequency

Clock frequency is a property of the hardware!
= 1 / max (longest combinatorial path)
= 1 / (critical path)

- Extremely high throughput (Radar or fiber optics)
  - One operat**or** (= hardware unit, e.g. adder, shifter, register)
  - for each operat**ion** (= algorithmic, e.g. addition, multiplication, delay)

⇨ clock frequency = sample frequency

- Most designs: time multiplexing

clock frequency ≠ sample frequency

$$\frac{\text{clock frequency}}{\text{sample frequency}} = \text{number of clock cycles available for the job}$$

Example: AES variations

- There is only one AES 128 algorithm
- There are multiple AES hardware implementation options.
- Basic operatIONS:
  - Byte-sub: non-linear operation on every byte
  - Shift-row: Circular shifting of bytes in each row
  - Mix-column: multiplying the round data with a fixed polynomial
  - Add-key: XORing the round data and round key

AES sequential

AES: one cycle per round

- Sample frequency < clock frequency
- Example: clock is 200 MHz, sample rate is 10MHz, 128 bits per sample
- Requested throughput: min 1.28 Gbits/sec
- Puts HW limit: one AES in 20 clock cycles or less

Solution:

Parallel datapath, sequential execution

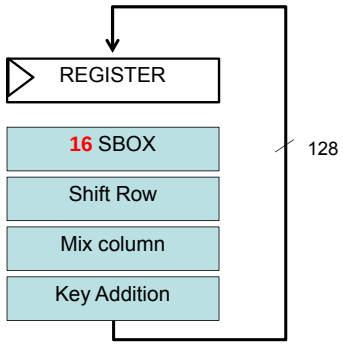
AES: one cycle per round

Solution:

- One 128 bit datapath
- 11 clock cycles to finish one AES
- (plus load and store clock cycle)
- 200 MHz clock

Throughput: ??

Latency: ??

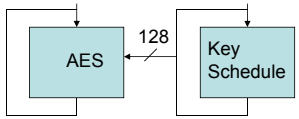


Efficiency - adapt HW platform to application

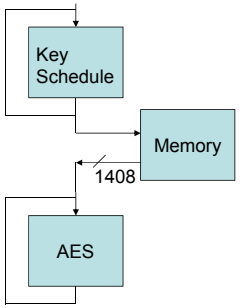
Simple example: Key Schedule for secret key

Two options:

- On the "fly" = just in time processing
- Pre-compute and store in memory

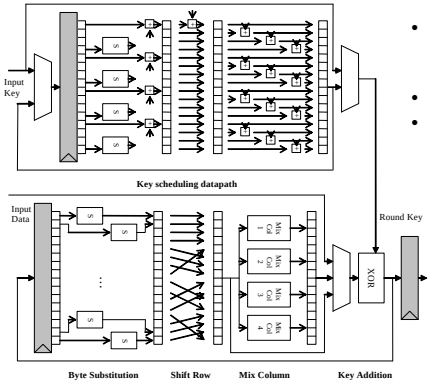


Typical for Hardware
1 cycle/round



Typical for Software
Minimum around **10 cycles/byte + bandwidth**

AES Core



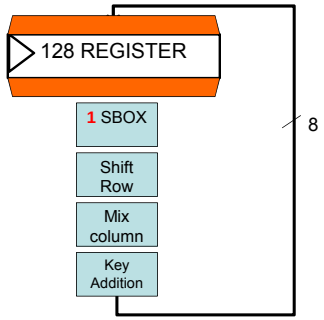
- Key schedule in parallel with data path
- 128-bit data and key
- One round implementation with minimum delay
  - One cycle per round
  - Direct implementation of Byte Substitution phase

11 cycles one encryption

Slide credit: Alireza Hodjat

AES compact: one SBOX

- Goal: low area
- Exercise:
  - Clock is 200MHz
  - Min number of clockcycles?
  - Throughput ?
  - Latency?
- Overhead:
  - muxes
  - Control logic

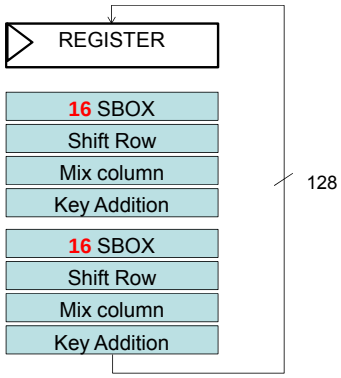


AES parallel and pipeline

AES parallel = loop unrolling

- Unroll twice
- Now 6 CC
- *IF* same clock
- THEN
- double throughput
- latency: ?
- (overhead ignored)

= expensive



AES unroll & pipeline

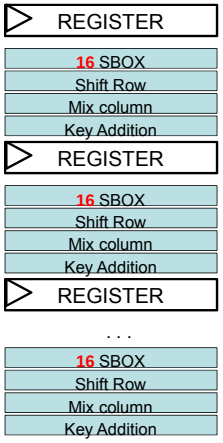
1 clock cycle per AES

11 data samples in pipeline

200 MHz clock

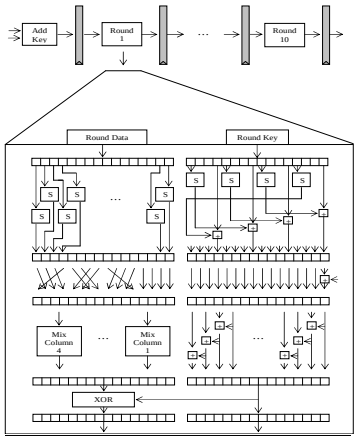
Throughput: ??

Latency: ??



AES loop enroll AND pipelined

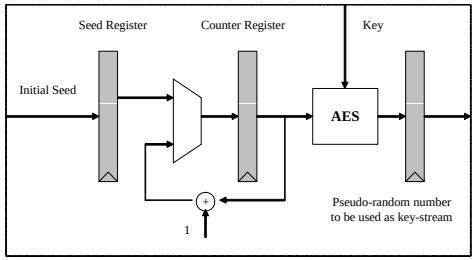
In combination with AES key schedule



Slide credit: Alireza Hodjat

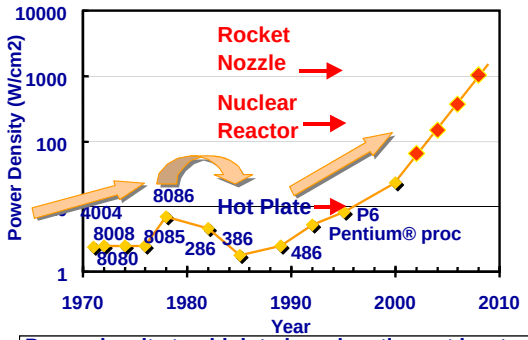
Non feedback modes of operation!!

- Example counter mode of operation
- Does not work with CBC or other feedback modes of operation



POWER

Power density



Power density too high to keep junctions at low temp
→ For desktops and servers it is a cooling problem

Courtesy, Intel

Cooling

- Cooling for the very small and the very large

Deep Brain stimulation
[Sources: J. Rabaey, National Institutes of Health, Neurology journal]

[San Francisco Chronicle online, source Google]

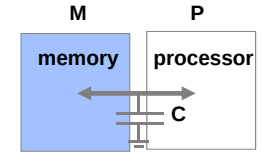
Power and Energy are not the same!

- Power = P = I x V (current x voltage) (= Watt)
  - instantaneous
  - Typically checked for cooling or for peak performance
- Energy = Power x execution time (= Joule)
  - Battery content is expressed in Joules
  - Gives idea of how much Joules to get the job done

Low power processor ≠ low energy solution !

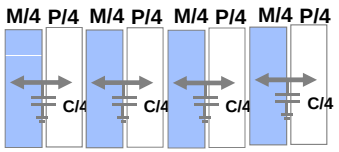
Heat and parallelism

Reduce power = reduce WASTE !!



Power (Heat)

$P_{mono} = CV^2f$ (Watt)



$4\,(C/4)V^2(f/4) = P_{mono}/4$

but since f ~ V

can be even $P_{mono}/4^3$

TREND: MULTI-CORE!!

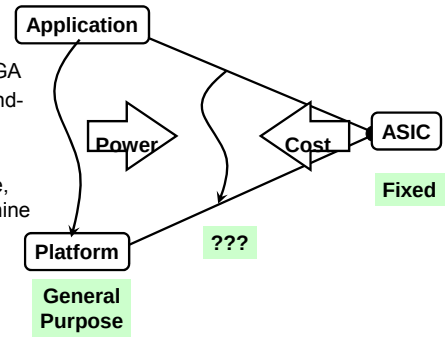
Throughput – Energy numbers

AES 128bit key 128bit data	Throughput	Power	Figure of Merit (Gb/s/W)
0.18mm CMOS	3.84 Gbits/sec	350 mW	11 (1/1)
FPGA [1]	1.32 Gbit/sec	490 mW	2.7 (1/4)
Intel ISA for AES	32 Gbit/sec	95 W	0.34 (1/33)
ASM StrongARM [2]	31 Mbit/sec	240 mW	0.13 (1/85)
Asm Pentium III [3]	648 Mbits/sec	41.4 W	0.015 (1/800)
C Emb. Sparc [4]	133 Kbits/sec	120 mW	0.0011 (1/10.000)
Java [5] Emb. Sparc	450 bits/sec	120 mW	0.0000037 (1/3.000.000)

[1] Amphion CS5230 on Virtex2 + Xilinx Virtex2 Power Estimator
[2] Dag Arne Osvik: 544 cycles AES – ECB on StrongArm SA-1110
[3] Helger Lipmaa PIII assembly handcoded + Intel Pentium III (1.13 GHz) Datasheet
[4] gcc, 1 mW/MHz @ 120 Mhz Sparc – assumes 0.25 u CMOS
[5] Java on KVM (Sun J2ME, non-JIT) on 1 mW/MHz @ 120 MHz Sparc – assumes 0.25 u CMOS

Match between algorithm & architecture

Close the gap:
- Dedicated HW: ASIC
- Programmable HW: FPGA
- Custom instructions, hand-coded assembly
- Compiled code
- JAVA on virtual machine, compiled on a real machine



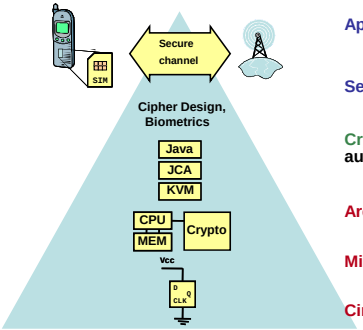
Energy power conclusions

- Low power processor is NOT low energy solution
- Power is limited
  - Cooling!!
  - Implanted devices only temperature $\Delta < 1$ °C
- Energy Battery is limited
  - Pace maker battery is not rechargeble
  - One AAA battery is 1300 … 5000 Joules
- **How much crypto in one micro Joule ?**

DESIGN METHODS

For low power/low energy

Design methodology: consider all design abstraction levels



Application: e-commerce, smart energy

Security analysis: TPM, light weight?

Crypto Algorithm/Protocol: crypto, entity authentication

Architecture: Co-design, HW/SW, SOC

Micro-Architecture: co-processor design

Circuit: Circuit techniques to combat side channel analysis attacks

WHY:
1. To get low power/ low energy
2. To be secure

Cost of crypto primitives

Energy - flexibility trade-off

1. Secret Key: AES
2. Public key: ECC

1 microJoule

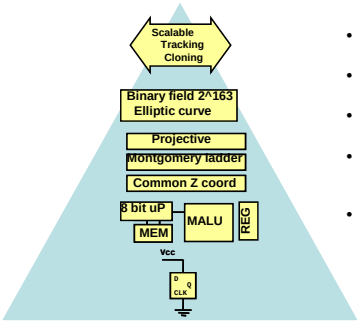
- 11000 bits AES (ASIC)
- 3000 to 10K gates area = small

Example 2:Public key - Elliptic Curve Cryptography

Push for lowest energy
to fit budget of RFID

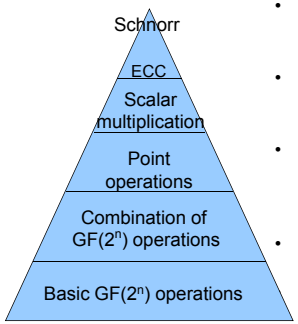
Challenge: low power public key …

Address at all design abstraction levels!



- **Protocol** : asymmetric (most work for the reader)
- **Algorithm**: Elliptic curve (163 bits) instead of RSA (min 1024 bits)
- **Field Operation**: Binary and not Prime fields: easier field operations
- **Projective** coordinate system: (X, Y, Z) instead of (x,y): no field inversions
- **Special coordinate system**: no need to store Y coordinates (Lopez-Dahab) and common Z (only one Z coordinate)
- **Minimize storage**: Only 5 registers (with mult/add/square unit) or 6 registers (with mult/add-only unit) compared to 9+ registers before.

Computation needs

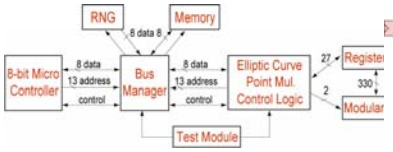


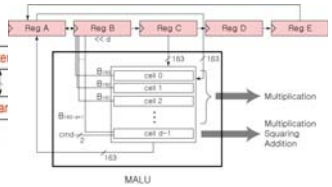
- One (simple) Schnorr protocol requires **one** elliptic curve point multiplication (compared to **two** at the reader)
- One point multiplication with Montgomery ladder requires **N** point additions & doublings (N = key length)
- With modified Lopez –Dahab common Z coordinate, one point addition and point doubling requires **7** field multiplications, **4** squarings and **3** additions
- One field multiplication requires 163/d clock cycles (d= digit size).

  For digit size 4, **79000** cycles (should stay below 100K)

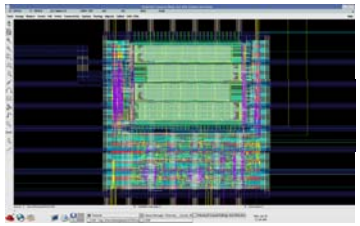
Results

- Results: ECC co-processor that can compute:
  - ECC point multiplications (163 by 4)
  - Scalar modular operations (8 bit processor with redundancy)
- Schnorr (secure ID transfer, but no tracking protection): **one** PM
- More advanced protocols: up to **four** PM on tag
- 14K gates, 79K cycles
- At 500 KHz, corresponds to 30 microWatt and 158 msec
- One point multiplication = **4.8 microJoule**





RFID co-processor prototype



- Combination full-custom – standard cells
- HW and SW co-design
- DPA Side channel resistance

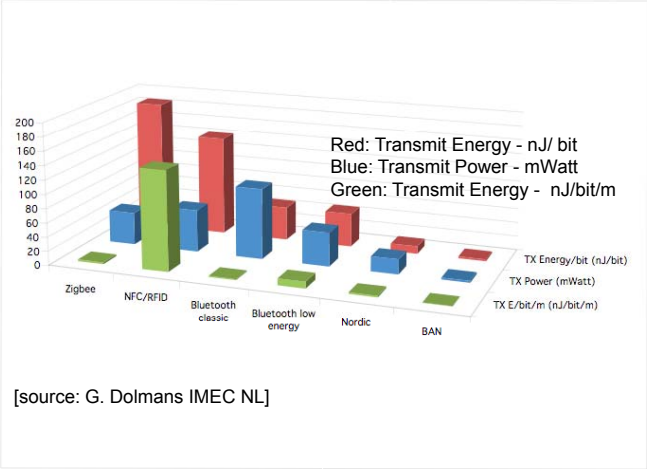
1 microJoule

- 11000 bits AES encryption
- 500 bits SHA3 hash
- 1/5 of one point multiplication

Still to add physical security …
(i.e. side-channel and fault attack resistant)

Communication & computation

Back of the envelope

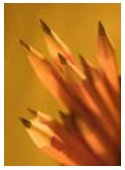


[source: G. Dolmans IMEC NL]

1 micro Joule

Transmission:
- 300 bits in BAN
- 11 bits Bluetooth
- 3 bits Zigbee

Encryption:
- 11000 bits AES
- 500 bits SHA3 hash
- 1/5 of one point multiplication



Ignores receive buget (= listening)
Ignores "overhead" of adding authentication bits, etc.

Example1 : Mutual Authentication Symmetric shared key

Reader: K 

Tag: K 

r_B

r_B →

r_A

← T_1

$T_1 \leftarrow E_K(r_A//r_B)$

check r_B
compute r_A

$T_2 \leftarrow E_K(r_B//r_A)$

→ T_2

Check response = decryption

Tag: two AES encryptions, one transmission over Bluetooth
128 bit Bluetooth + 2 x AES ≈ 10 microJoule

ECC based randomized Schnorr

Reader: y, $X = xP$ 

Tag: x, $Y = yP$ 

$r_1, r_2,$

← T_1, T_2

$T_1 = r_1P$, $T_2 = r_2Y$

c

c →

$v = r_1 + r_2 + cx$

← v

$c^{-1}[vP - T_1 - y^{-1}T_2] =? X$

Tag: two point multiplications, two transmissions over BAN
Crypto dominates ≈ 4 microJoule + 1 microJoule

Conclusions

- Time has many faces: real-time, throughput, latency, clock frequency, critical path, …
- Power is not same as energy !
- Energy - flexibility trade-off = orders of magnitude !
- Communication- computation trade-off !

With thanks to Nele Mentens for presenting this!!