# Cost of cryptography in hardware

**Ingrid Verbauwhede**
ingrid.verbauwhede-at-esat.kuleuven.be

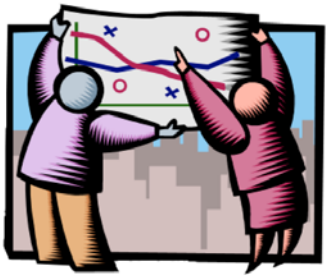
K.U.Leuven, ESAT- SCD - COSIC
Computer Security and Industrial Cryptography

Acknowledgements:
**Current and former Ph.D. students at UCLA and K.U.Leuven**



## Outline & Goal

- Workshop on “Elliptic Curve and Hyper-Elliptic Curve cryptography”
- All advanced topics
- Focus here: how much does it cost in hardware??
  - Hardware parameters
  - Compare public-key with secret key
  - Computation-communication cost of crypto based protocols



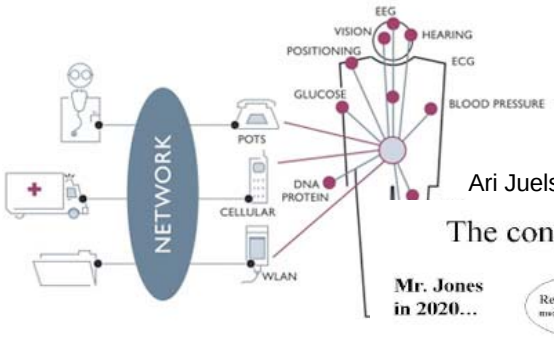
## Design Parameters

Embedded security:

Area, delay, power, energy,
physical security

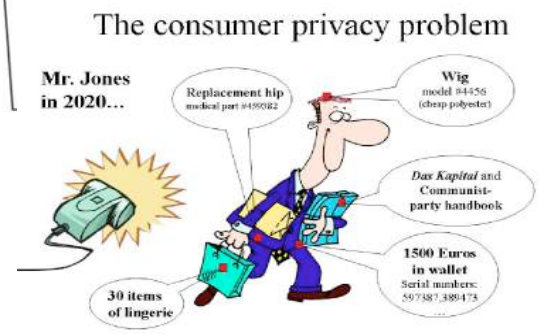


## Embedded crypto everywhere

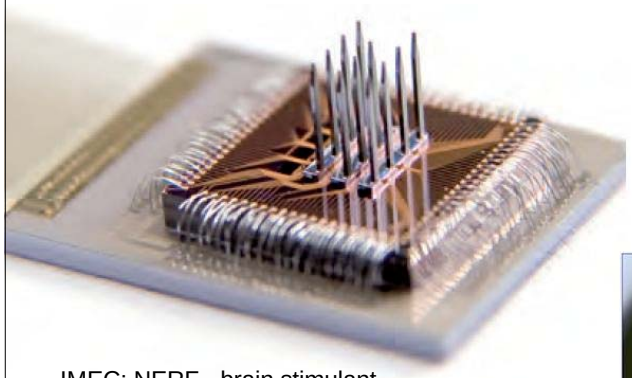



IMEC: Human++ project

Ari Juels: RFID tracking problem



## Embedded crypto everywhere



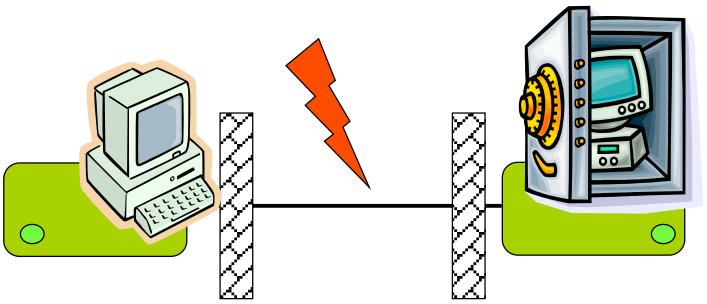
IMEC: NERF - brain stimulant

Deep Brain stimulation

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



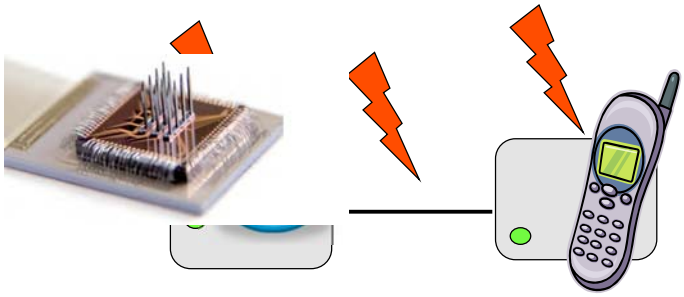
## Embedded Security



**Old Model (simplified view):**

- **Attack on channel *between* communicating parties**
- **Encryption and cryptographic operations in *black* boxes**
- **Protection by strong mathematic algorithms and protocols**

## 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**

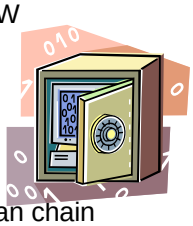
***Need secure implementations not only algorithms***



## Embedded Security

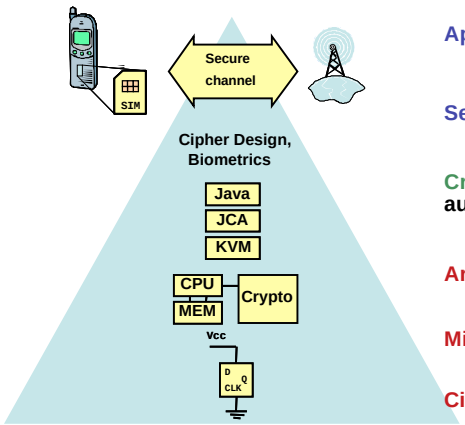
NEED BOTH

- 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

# 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 definition

- Area
- Time
- Power, Energy
- Physical Security
- NRE (Non Recurring Engineering) cost

## Design parameters

- Speed or throughput:
  - HW: Gbits/sec or Mbits/sec/slice
  - SW: Cycles/byte, independent of clock frequency
- Area:
  - HW: mm2 (gate or transistor count)
  - SW: memory footprint
- Power or energy consumption:
  - Power (Watts) for cooling or transmission (RFID)
  - Energy (Joule): battery operated devices
- Security: difficult to measure, but we want it
  - Entropy, leakage functions?
  - Measurements until disclosure?



## Throughput: Real-time

• Extremely high throughput (Radar or fiber optics)
  • One operator (= hardware unit, e.g. adder, shifter, register)
  • for each operation (= algorithmic, e.g. addition, multiplication, delay)

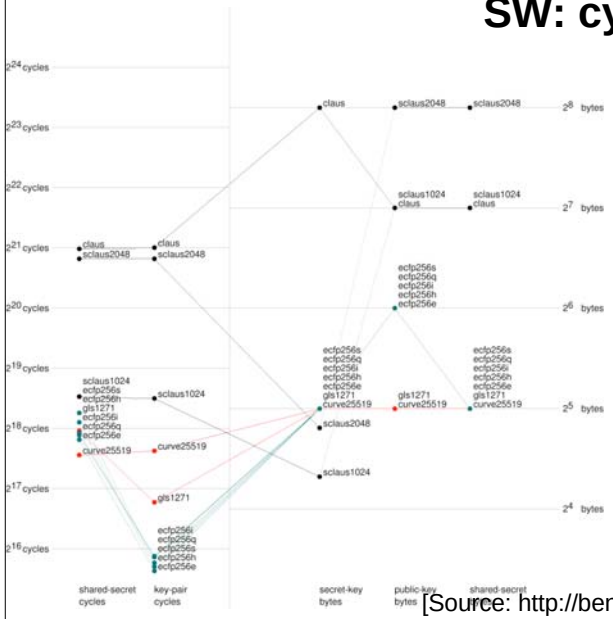
⇨ clock frequency = sample frequency

• Most designs: time multiplexing

clock frequency ≠ sample frequency

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

## SW: cycles & bytes

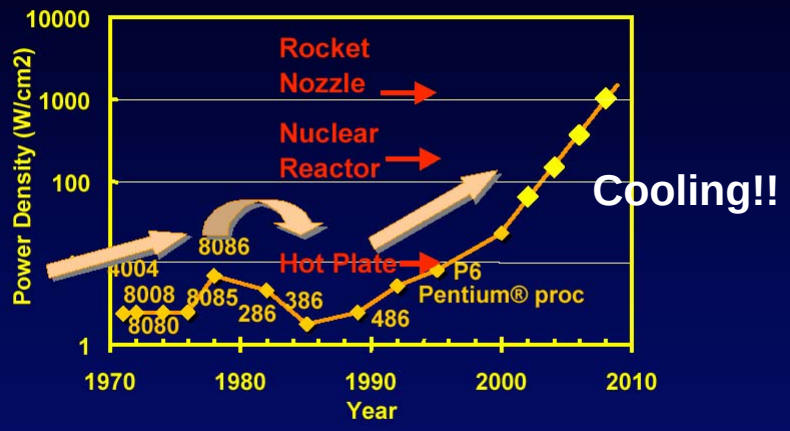



- "independent" of clock *frequency* or machine
- "match" of algorithm to architecture

[Source: http://bench.cr.yp.to/results-dh.html]

KUL - COSIC ECC Workshop - 13 Nancy, Sept 2011

## Power density will increase




Power density too high to keep junctions at low temp

intel

[Author: S. Borkar, Intel]

16

## What can one do with 1 cm³?
### *Energy Storage*

| | J/cm³ | µW/cm³/year |
|---|---|---|
| Micro Fuel cell | 3500 | 110 |
| Primary battery | 2880 | 90 |
| Secondary battery | 1080 | 34 |
| Ultra-capacitor | 100 | 3.2 |


© J. Rabaey - 06

Power-Intro 20

**One AAA battery: 1300 to 5000 Joule**

KUL - COSIC ECC Workshop - 15 Nancy, Sept 2011

## Power and Energy are not the same!

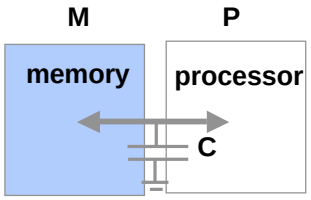
- Power = P = I x V (current x voltage) (= Watt)
  - instantaneous
  - Typically checked for cooling or for peak performance
- Energy = Power x execution time (= Joule)
  - Battery content is expressed in Joules
  - Gives idea of how much Joules to get the job done

Low power processor ≠ low energy solution !

KUL - COSIC ECC Workshop - 16 Nancy, Sept 2011

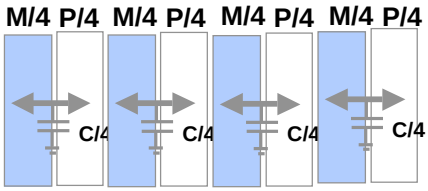
## Heat and parallelism

**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!!**

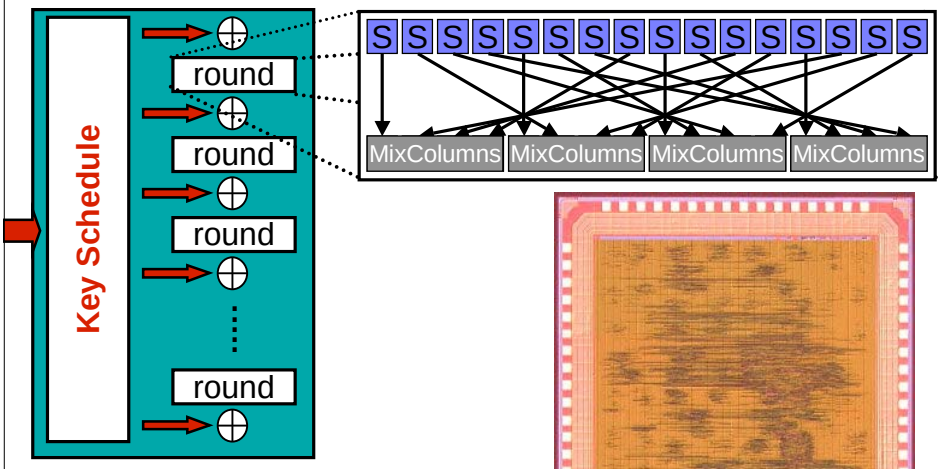


## Cost of crypto primitives

Energy - flexibility trade-off

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

## Example: Rijndael/AES




- key length: 16/24/32 bytes
- block length: 16/24/32 bytes

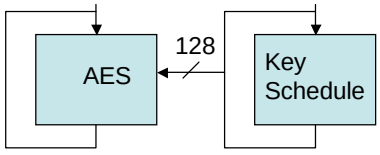


## Efficiency - adapt HW platform to application

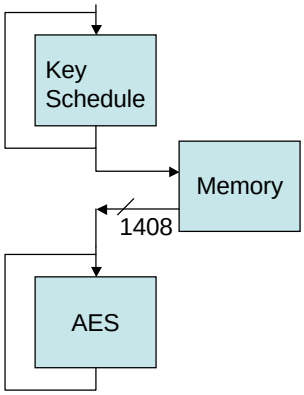
Simple example: Key Schedule for secret key

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**

## Throughput – Energy numbers

| AES 128bit key 128bit data | Throughput | Power | Figure of Merit (Gb/s/W) |
|---|---|---|---|
| 0.18μm CMOS | 3.84 Gbits/sec | 350 mW | 11 (1/1) |
| FPGA [1] | 1.32 Gbit/sec | 490 mW | 2.7 (1/4) |
| 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) |

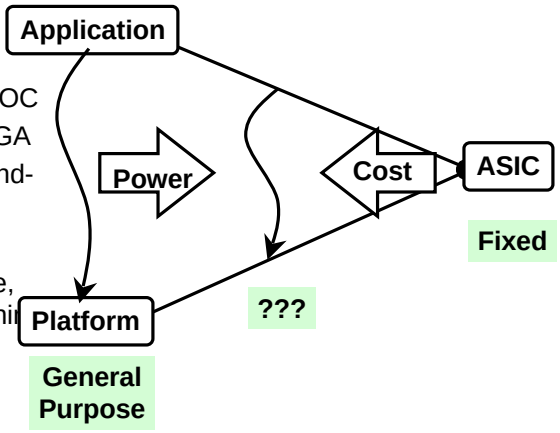
[1] Amphion CS5230 on Virtex2 + Xilinx Virtex2 Power Estimator
[2] Dag Arne Osvik: 544 cycles AES – ECB on StrongArm SA-1110
[3] Helger Lipmaa PIII assembly handcoded + Intel Pentium III (1.13 GHz) Datasheet
[4] gcc, 1 mW/MHz @ 120 Mhz Sparc – assumes 0.25 u CMOS
[5] Java on KVM (Sun J2ME, non-JIT) on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



## Match between algorithm & platform

Close the gap:

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




**Energy - flexibility trade-off**

## 1 microJoule

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

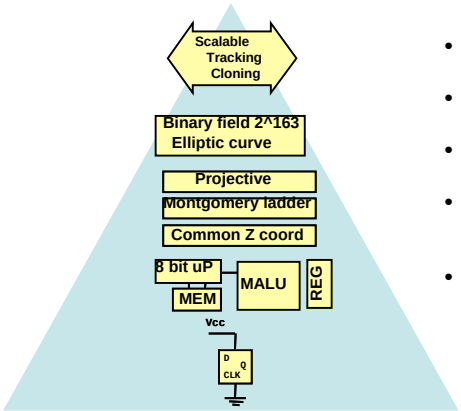


## Example 2:Public key - Elliptic Curve Cryptography

**Push** for lowest energy
to fit budget of RFID

# Challenge: low power public key …

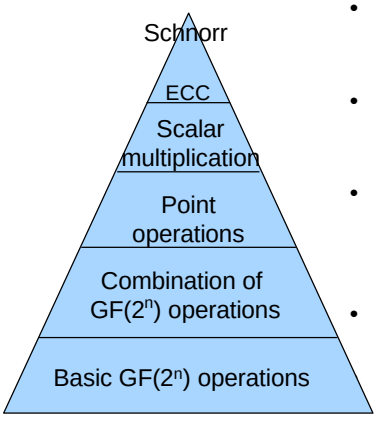
**Address at all design abstraction levels!**




- **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

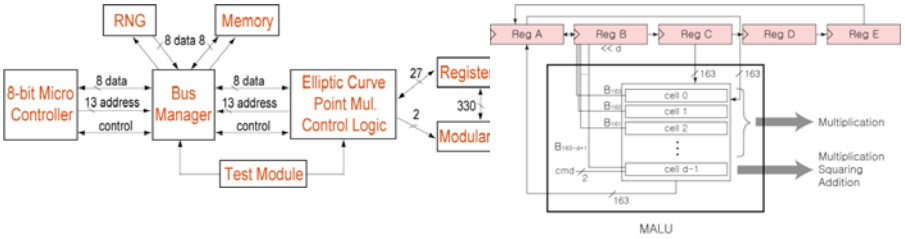



- One (simple) Schnorr protocol requires **one** elliptic curve point multiplication (compared to **two** at the reader)
- One point multiplication with Montgomery ladder requires **N** point additions & doublings (N = key length)
- With modified Lopez –Dahab common Z coordinate, one point addition and 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)

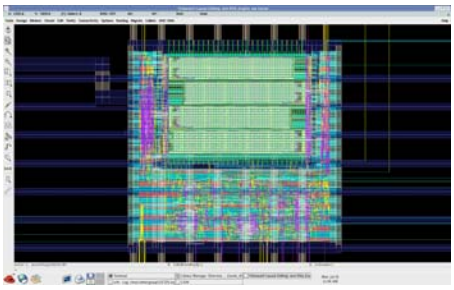
# Results

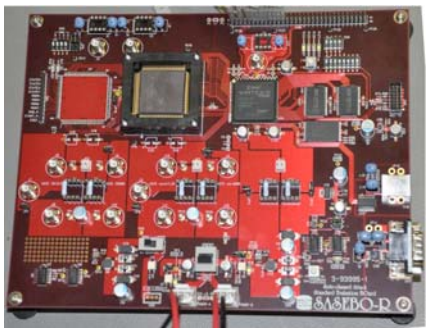
- Results: ECC co-processor that can compute:
  - ECC point multiplications (163 by 4)
  - Scalar modular operations (8 bit processor with redundancy)
- Schnorr (secure ID transfer, but no tracking protection): **one** PM
- More advanced protocols: up to **four** PM on tag
- 14K gates, 79K cycles
- At 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
- Side channel testing in progress

## 1 microJoule

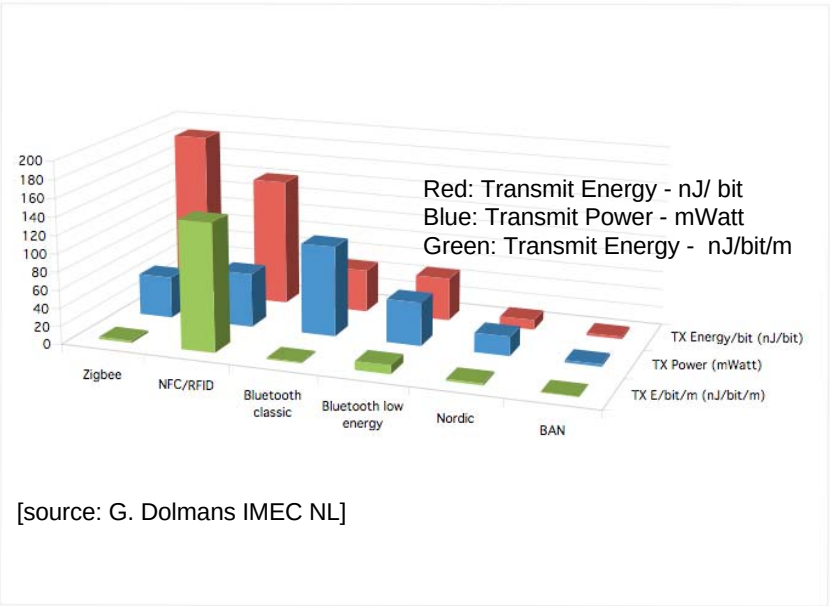
- 11000 bits AES encryption
- 500 bits SHA3 hash
- 1/5 of one point multiplication

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



## Communication & computation

Back of the envelope




Red: Transmit Energy - nJ/ bit
Blue: Transmit Power - mWatt
Green: Transmit Energy - nJ/bit/m

[source: G. Dolmans IMEC NL]

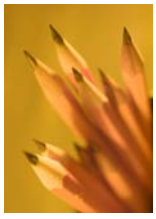
## 1 micro Joule

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: $K$ 

| | | |
|---|---|---|
| | $r_B$ → | $r_A$ |
| $r_B$ | | |
| | ← $T_1$ | $T_1 \leftarrow E_K(r_A // r_B)$ |
| *check* $r_B$ *compute* $r_A$ | | |
| | $T_2$ → | *Check response = decryption* |
| $T_2 \leftarrow E_K(r_B // r_A)$ | | |

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



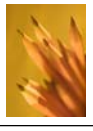
## ECC based randomized Schnorr

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



## Physical security??

Countermeasures against physical attacks, i.e. side-channel and fault attacks



## Attacks vs. countermeasures

| | | |
|---|---|---|
| Passive SCA | Timing analysis | Balanced PA/PD |
| | Simple power analysis | Double-and-add-always |
| | Differential power analysis | Montgomery Powering Ladder |
| | Template attack | |
| Active SCA | M safe-error | Base point blinding |
| | C safe-error | Random projective coordinates |
| | Invalid points | Randomized EC isomorphism |
| | Invalid curves | Randomized field isomorphism |
| | Twist curves | Point validity check |
| | Sign-change attacks | Curve integrity check |
| | Differential faults | Coherence check |

**Attackers need only a single successful attack to win.**

[source: Junfeng Fan]

## Attacks vs. countermeasures

√ : Effective
× : Attacked
? : Unclear

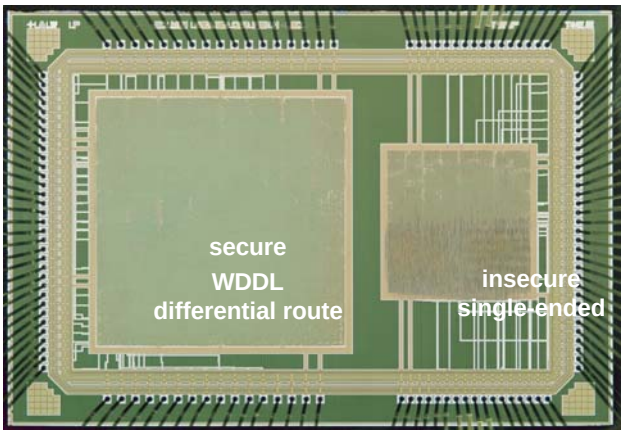
-- : Irrelevant
H : helps the attack

| Countermeasures | Passive Attacks | | | | | | | Active Attacks | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | Safe-error | | Weak curve | | | Differential | |
| [source: Junfeng Fan] | TA | SPA | Template | DPA | Comparative SCA | RPA/ZPA | Carry-based attack | M safe-error | C safe-error | Invalid point | Invalid curve | Twist curve | Sign change | Differential |
| Balanced PA/PD | √ | √ | -- | -- | ? | -- | -- | -- | -- | -- | -- | -- | -- | -- |
| Double-and-add-always | √ | √ | -- | -- | × | -- | -- | -- | ×H | -- | -- | -- | -- | -- |
| Montgomery Powering Ladder⊥ | √ | √ | -- | -- | × | × | -- | √ | √ | -- | -- | H | √ | -- |
| Montgomery Powering Ladder⊤ | √ | √ | -- | -- | × | × | -- | √ | √ | -- | -- | √ | -- | -- |
| Random scalar split | -- | -- | ? | √ | ? | √ | × | -- | ? | -- | -- | √ | ? | ? |
| Scalar randomization | -- | -- | × | × | × | √ | × | -- | ? | -- | -- | -- | ? | ? |
| Base point blinding | -- | -- | × | × | × | √ | -- | -- | -- | ? | -- | -- | -- | ? |
| Random projective coordinates | -- | -- | √ | √ | ? | × | -- | -- | -- | -- | -- | -- | -- | ? |
| Randomized EC isomorphism | -- | -- | ? | √ | ? | × | -- | -- | -- | -- | -- | -- | -- | ? |
| Randomized field isomorphism | -- | -- | ? | √ | ? | × | -- | -- | -- | -- | -- | -- | -- | ? |
| Point validity check | -- | -- | -- | -- | -- | -- | -- | -- | H | √ | ? | √ | H | √ |
| Curve integrity check | -- | -- | -- | -- | -- | -- | -- | -- | -- | ? | √ | -- | -- | -- |
| Coherence check | -- | -- | -- | -- | -- | -- | -- | -- | H | -- | ? | -- | √ | √ |



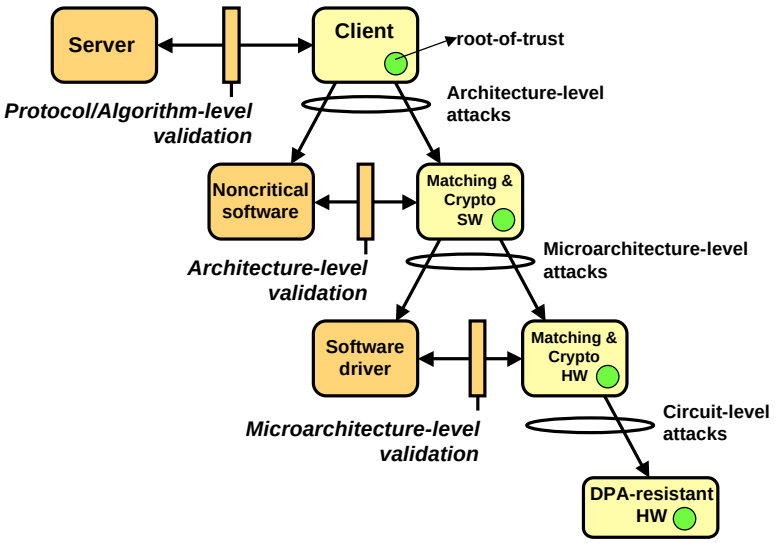
## Prototype IC – ThumbPodII

- AES, controller, fingerprint processor.




**Area: factor 2.5** **Power: factor 3 to 4 !**

# Design Method: Security Partitioning

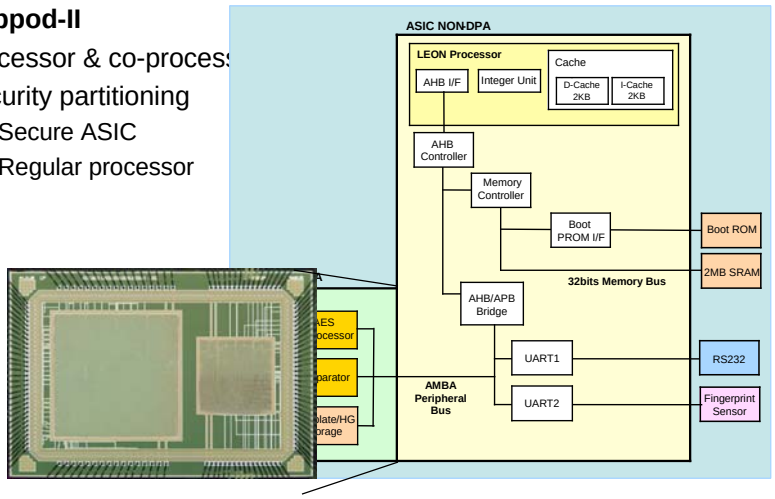





# Security partitioning - SOC

**Thumbpod-II**

- Processor & co-process
- Security partitioning
  - Secure ASIC
  - Regular processor



## Conclusions

- Power is not same as energy !
- Energy - flexibility trade-off = orders of magnitude !
- Communication- computation trade-off !

- Low budget is needed, but not there yet.
- Research topics:
  - Light weight crypto
  - Physically entangled crypto, link to PUFs and other devices
  - Design methods for security partitioning
- because:

**weakest link decides strength of chain**