Low budget cryptography to enable wireless security

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

K.U.Leuven, COSIC
Computer Security and Industrial Cryptography
www.esat.kuleuven.be/cosic

with input from:
current and former Ph.D. students

Outline: embedded security

• Settings: applications
• Design goals: area - time - energy/power
• Cost of wireless link
• Cost of crypto primitives
• Example(s)
• Conclusions & Future work
Embedded crypto everywhere

IMEC: Human++ project

Ari Juels: RFID tracking problem
The consumer privacy problem

Mr. Jones
in 2020...

IMEC: NERF - brain stimulant

Deep Brain stimulation
[Sources: J. Rabaey, National Institutes of Health, Neurology journal]
Embedded crypto: challenge (1)

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 crypto: challenge (2)

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 crypto: challenge (3)**

**NEED BOTH**

- Efficient, lightweight implementations
  - Within power, area, timing budgets
  - Public key: 2048 bits RSA, 200 bit ECC 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

**Design Parameters**

Measures for security?
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, resistance to attacks: difficult to measure, but still we want it …
  - Entropy, leakage functions?
  - Measurements until disclosure?
  - Cost versus benefit
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)

\[ \text{clock frequency} = \text{sample frequency} \]

- Most designs: time multiplexing

\[ \text{clock frequency} / \text{sample frequency} \]

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

- Goal: low clock frequency for low power

Power density will increase

[Diagram showing increase in power density from 1970 to 2010 with labels such as Rocket Nozzle, Nuclear Reactor, Hot Plate, Pentium® proc. Power density too high to keep junctions at low temp.

[Author: S. Borkar, Intel]
Low Energy: battery capacity

- Rabaey slide battery capacity

One AAA battery: 1300 to 5000 Joule

Power and Energy are not the same!

- Power = $P = I \times V$ (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!

- Low clock for low power does not necessarily result in low energy …
Heat and parallelism

Reduce power = reduce WASTE !!

\[
\text{Power (Heat)}
\]

\[
P_{\text{mono}} = CV^2f \text{ (Watt)}
\]

\[
4 \left(\frac{C}{4}\right)^2 \left(\frac{f}{4}\right) = \frac{P_{\text{mono}}}{4}
\]

but since \( f \sim V \)
can be even \( \frac{P_{\text{mono}}}{4^3} \)

TREND: MULTI-CORE!!

Medical implants

• Power is limited
  – Cooling!!
  – Implanted devices only temperature \( \Delta < 1 \) °C

• Battery is limited
  – Pace maker battery is not rechargeble

• Budget is less than 0.5 microWatt
Cost of wireless links

[source: G. Dolmans IMEC NL]
Budget is 1 micro Joule

Back of the envelope calculation
Transmit budget
- 300 bits in BAN (Body Area Network)
- 11 bits Bluetooth
- 3 bits Zigbee

Cost of crypto primitives

Energy - flexibility trade-off
Illustrate with examples

- Example 1: Secret Key: AES
- Example 2: NIST SHA3 – how not to do it
- Example 3: Public key, ECC for RFID
- Example 4: light weight algorithms?
- Example 5: cost of physical security

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

Throughput – Energy numbers

<table>
<thead>
<tr>
<th></th>
<th>Throughput</th>
<th>Power</th>
<th>Figure of Merit (Gb/s/W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES 128bit key 128bit data</td>
<td>3.84 Gbits/sec</td>
<td>350 mW</td>
<td>11 (1/1)</td>
</tr>
<tr>
<td>0.18μm CMOS</td>
<td>1.32 Gbit/sec</td>
<td>490 mW</td>
<td>2.7 (1/4)</td>
</tr>
<tr>
<td>FPGA [1]</td>
<td>31 Mbit/sec</td>
<td>240 mW</td>
<td>0.13 (1/85)</td>
</tr>
<tr>
<td>ASM StrongARM [2]</td>
<td>648 Mbits/sec</td>
<td>41.4 W</td>
<td>0.015 (1/800)</td>
</tr>
<tr>
<td>Pentium III [3]</td>
<td>133 Kbits/sec</td>
<td>120 mW</td>
<td>0.0011 (1/10,000)</td>
</tr>
<tr>
<td>C Emb. Sparc [4]</td>
<td>450 bits/sec</td>
<td>120 mW</td>
<td>0.0000037 (1/3,000,000)</td>
</tr>
</tbody>
</table>

[1] Amphion CS5230 on Virtex2 + Xilinx Virtex2 Power Estimator
[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 machine

Energy - flexibility trade-off

1 microJoule

- 11000 bits AES (optimized version)
- 3000 to 10K gates area = small
SHA3 – competition:

One size fits all

“Flexibility” Requirements

- Wide range of platforms
- Wide range of message digests

[of course, also security requirements]
SHA-3: “cost” requirements

Computational efficiency essentially refers to the throughput of an implementation. NIST will use the

C.2.2 Memory requirements: The memory required for hardware and software implementations of the candidate algorithm will be considered during the evaluation process.

Memory requirements will include such factors as gate counts for hardware implementations, and code size and RAM requirements for software implementations.

• Power consumption?
• Energy to hash one message?

SHA3- results

• NIST asks for a Swiss knife

Bread knife

Surgeon’s knife

• But often you need a specialized knife
• Certainly for embedded applications
### SHA 3 ASIC (90nm) synthesis

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Throughput (@ 250MHz)</th>
<th>Gate (GE)</th>
<th>Energy (pJ/bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA256</td>
<td>2000</td>
<td>12K</td>
<td>2</td>
</tr>
<tr>
<td>Blake</td>
<td>6000</td>
<td>30K</td>
<td>2.5</td>
</tr>
<tr>
<td>Grøstl</td>
<td>13000</td>
<td>86K</td>
<td>2.5</td>
</tr>
<tr>
<td>JH</td>
<td>4600</td>
<td>30K</td>
<td>2</td>
</tr>
<tr>
<td>Keccak</td>
<td>15000</td>
<td>30K</td>
<td>1</td>
</tr>
<tr>
<td>Skein</td>
<td>6700</td>
<td>43K</td>
<td>6</td>
</tr>
</tbody>
</table>

[slide input: Miroslav Knežević]

---

### 1 microJoule

- 11000 bits AES encryption
- 500 bits SHA3 hash, 30K gates
Example 3: 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)

---

Step 3: EC Point Multiplication

- **k·P** \(\rightarrow\) **Montgomery Algorithm**
  \(-600\text{ GE (Control)}\)
- \(P+Q, 2·P\) \(\rightarrow\) **Lopez-Dahab Algorithm**
  \(-1.2k\text{ GE (Control)} + 6\times163\text{ registers}\)
- **a+b, a·b, a^2** \(\rightarrow\) **Sakiyama Modular ALU**
  \(-900\text{ GE (Control)} + 3\times163\text{ registers}\)

Total Area = **2.7 GE (Control) + 9\times163\text{ registers} \approx 80\%** !!
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
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 budget (= 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 ← E_K(r_A/r_B)

check r_B
compute r_A

T_2 ← 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: \( y, X = xP \)  
\( T_1, T_2 \) 
\( c \)

Tag: \( x, Y = yP \)  
\( T_1 = r_1P, T_2 = r_2Y \) 
\( v = r_1 + r_2 + cx \)

\( c^{-1}[vP - T_1 - y^{-1}T_2] = X \)

Tag: two point multiplications, two transmissions over BAN
\( Crypto \ dominates \sim 4 \ \text{microJoule} + 1 \ \text{microJoule} \)

Physical security??

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

Passive Attacks
- Timing analysis
- Simple power analysis
- Differential power analysis
- Template attack

Balanced PA/PD
- Double-and-add-always
- Montgomery Powering Ladder

Active SCA
- M safe-error
- C safe-error
- Invalid points
- Invalid curves
- Twist curves
- Sign-change attacks
- Differential faults

Base point blinding
- Random projective coordinates
- Randomized EC isomorphism
- Randomized field isomorphism
- Point validity check
- Curve integrity check
- Coherence check

Attacker need only a single successful attack to win.

[Source: Junfeng Fan]
Prototype IC – ThumbPodII

- AES, controller, fingerprint processor.

Area: factor 2.5  Power: factor 3 to 4!

Design Method: Security Partitioning

- Protocol/Algorithm-level validation
- Architecture-level validation
- Noncritical software
- Matching & Crypto SW
- Software driver
- Matching & Crypto HW
- Circuit-level attacks
- DPA-resistant HW
- Architecture-level attacks
- Microarchitecture-level validation
- Matching & Crypto HW
- Root-of-trust
Security partitioning - SOC

Thumbpod-II
- Processor & co-processor
- Security partitioning
  - Secure ASIC
  - Regular processor

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

Easily 100% overhead for physical security
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

Solutions at HW/SW level?

- Solutions in SW
  - Randomizations
  - Hiding
  - ...
- Solutions in HW
  - Masking
  - Hiding
  - ...
Intro to Static CMOS

- Consumes power when output makes a 0 to 1 transition
- Most popular circuit style!

<table>
<thead>
<tr>
<th>IN</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0→0</td>
<td>0</td>
</tr>
<tr>
<td>0→1</td>
<td>discharge</td>
</tr>
<tr>
<td>1→0</td>
<td>charge</td>
</tr>
<tr>
<td>1→1</td>
<td>0</td>
</tr>
</tbody>
</table>

WDDL library

- All functions of and2, or2 operator
- In addition: inverted input, output signals
- XOR2X4: OAI221X2:
- WDDL library: 128 cells
Differential Power Attack on AES key bytes

Future work:
Future applications:
- RFID, Bio implants
- Smart metering

Common themes:
- Do we have all the required properties covered?
- Is privacy covered?
- Light weight crypto?
- How to store keys securely?
- Use physical properties: PUF based security?
- What about side-channel security?