# A Cryptographic Processor for Low-Resource Devices: Canning ECDSA and AES Like Sardines

**Michael Hutter, Martin Feldhofer, and Johannes Wolkerstorfer**

WISTP 2011

01. - 03.06.2011, Heraklion, Greece

IAIK TU Graz

Institute for Applied Information Processing and Communications (IAIK)

Graz University of Technology

# Outline

- Project CRYPTA
- Motivation
- The Processor
  - The System Architecture
  - Memory Unit and Datapath
  - Microcontroller
- Attack Countermeasures
- Results
- Conclusion

# Project CRYPTA

- Joint work
  - IAIK (TUGraz)
  - Austriamicrosystems
  - RF-iT Solutions
- Objectives
  - Design of a passive RFID Tag
  - Complete system
  - Proof-of-concept demonstrator
  - Crypto support:
    - AES-128 (enc+dec)
    - SHA-1
    - ECDSA NIST P-192




# Motivation

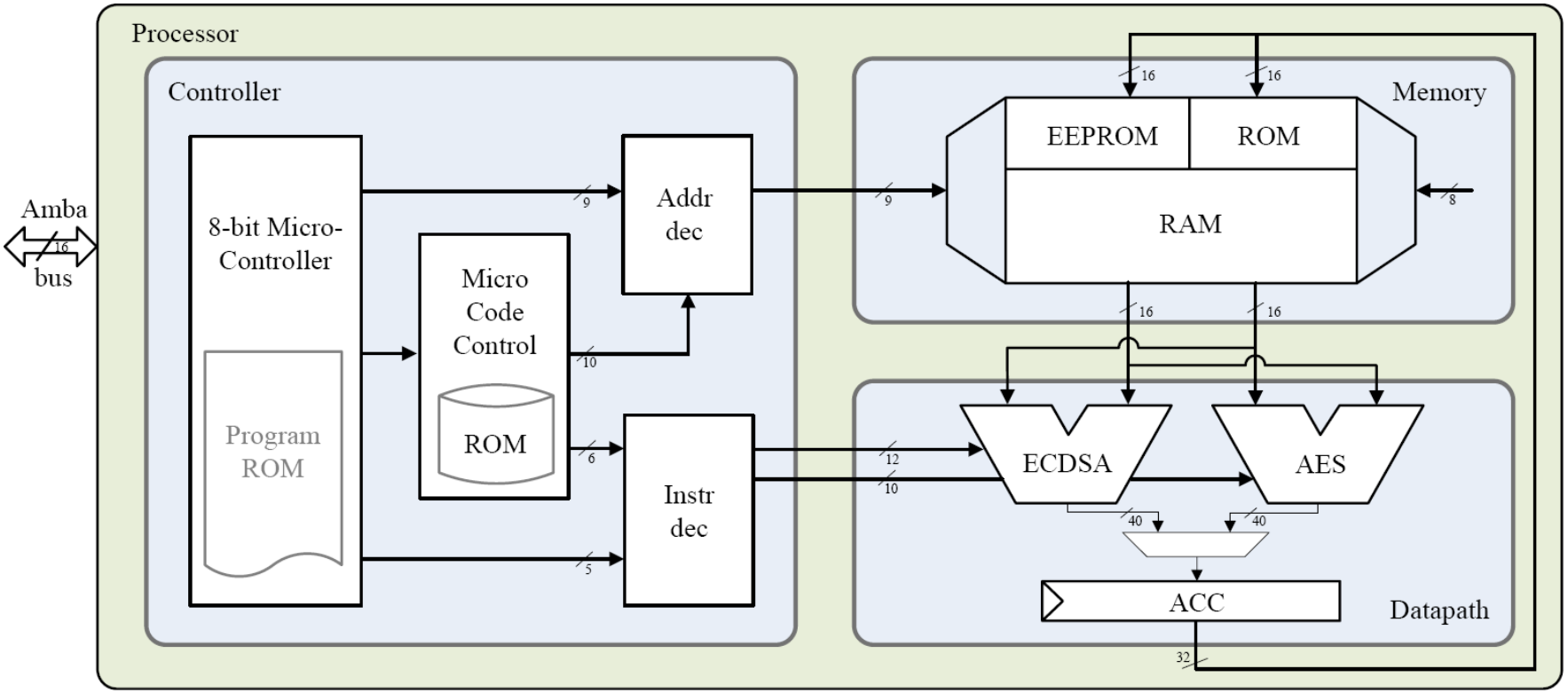
- Prevent counterfeit goods
  - Problem all over the world
  - Styrian pumpkin-seed oil or cheap copy?
  - When you choose a product,
    you want the quality you've paid for…

©everythingpumpkin.wordpress.com

- Authenticity of a product through Digital Signatures
  - Transferable proof of origin
  - Message authentication, non-repudiation, data integrity
- Need of asymmetric cryptography

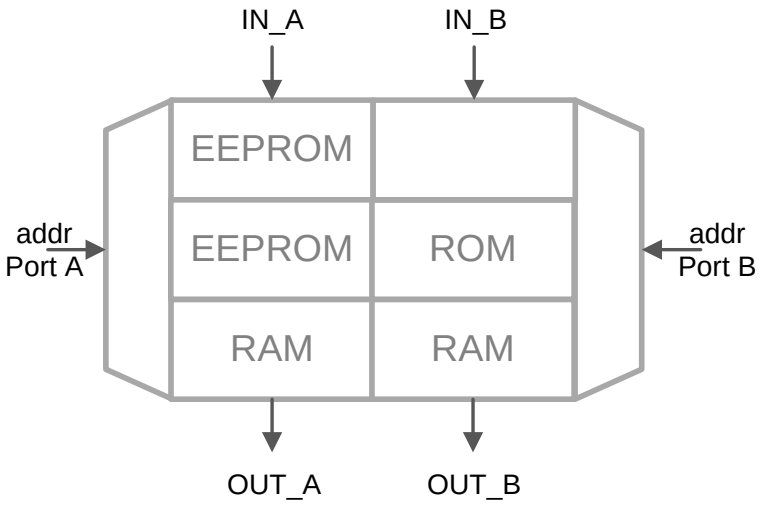
# What we did?

- A cryptographic-enabled RFID processor



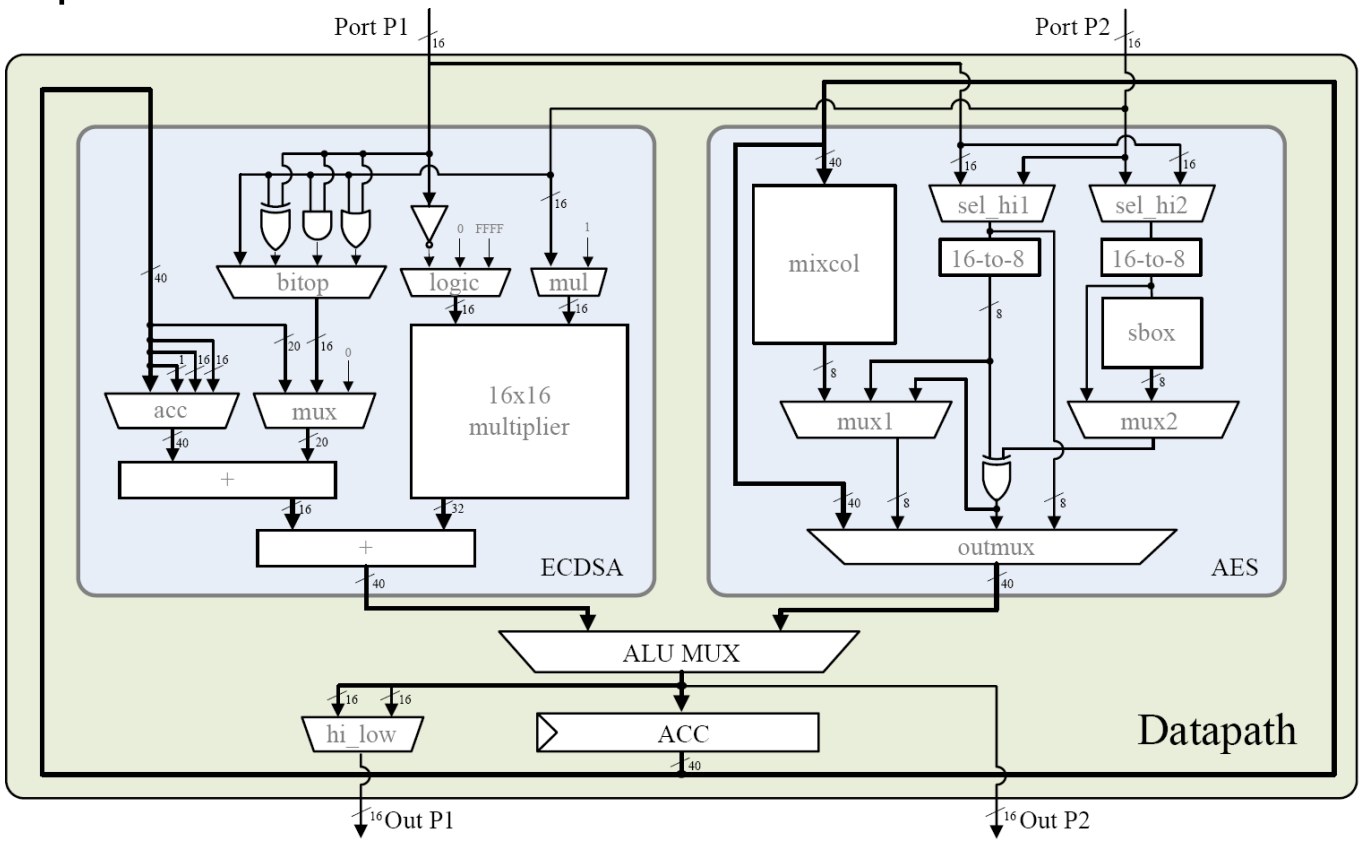
# Memory Unit

- 16-bit dual ported interface
  - Concurrently read/write from/to two ports
- RAM macro (128x16 bit)
- ROM
  - ECC constants (e.g. base point P)
- EEPROM
  - Stores the private key
  - Stores the certificate

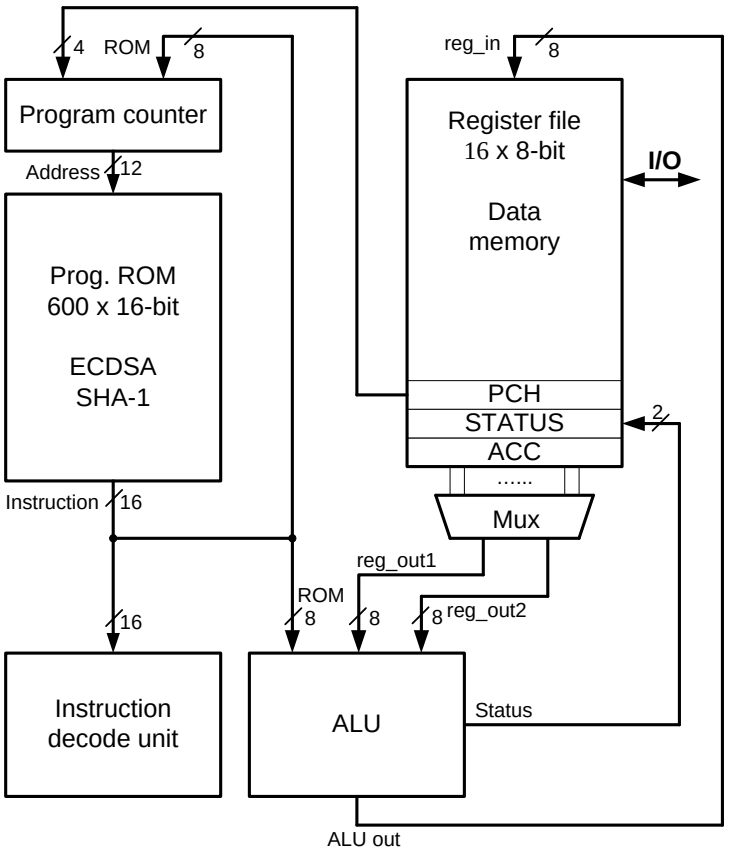


# 16-bit Datapath

- 16x16-bit multiply accumulate (MAC) unit
- 1 cycle 16-bit operations
- 40-bit ACCU



# 8-bit Microcontroller




- 32 instructions supported
  - Arithmetic operations (ADD, SUB,…)
  - Logical operations (OR, AND,…)
  - Control operations (GOTO, CALL,…)
- Register file and program ROM
- Instruction decoder, ALU, Counter,…
- Two-stage pipeline (fetch and execute)
- Call-stack support (3 recursive subroutines possible)
- Self-written Java compiler

# Micro-Code Control

- Micro-code patterns for AES, ECDSA, and SHA1
  - Can be executed by the microcontroller by a MICRO instruction
- Implemented in 8 ROM tables
  - Area reduction through different table sizes
- Modular arithmetic
  - Addition: 32 cycles
  - Subtraction: 38 cycles
  - Multiplication: 204 cycles
  - NIST reduction applied ($p_{192} \equiv 2^{192} - 2^{64} - 1$)
- Montgomery arithmetic
  - Inversion: 20,823 cycles
  - Multiplication: 785 cycles

```
...
MovLF(ADDR1_REG, 0x4);
MICRO(INST_ADD, addr_par9, 19);
CALLR("NIST_RED");
...

LABEL("NIST_RED");
  MICRO(INST_RED1, addr_null, 4);
  MICRO(INST_RED2, addr_null, 8);
  BWS(STATUS, CU_NEXT_INSTR);
  BTC(STATUS, CU_CARRY);
  MICRO(INST_SUB, addr_par14, 19);
RET();
```

# The ECC Implementation

- Use of Montgomery Ladder

**Input:** $P \in E(\mathbb{F}_q)$ and $k = (k_{n-1}, \ldots, k_0)_2 \in \mathbb{N}$, with $k_{n-1} \neq 0$
**Output:** $Q = kP$

1: $R_0 \leftarrow P$; $R_1 \leftarrow 2P$
2: **for** $i = n-2$ downto 0 **do**
3: $\quad b \leftarrow k_i$; $R_{1-b} \leftarrow R_{1-b} + R_b$
4: $\quad R_b \leftarrow 2R_b$
5: **end for**
6: **return** $R_0$

- Use of x-coordinate only formulae (Brier and Joye)
- Combined double-and-add (Izu, Möller, and Takagi)
- Common-Z coordinate representation (Meloni, Lee)
  - Idea: share a common coordinate (*e.g.* X or Z)
  - Equivalent representation:

$$R_0 \cong (X_1', Z') \text{ and } R_1 \cong (X_2', Z')$$

where

$$X_1' = X_3 Z_4\,, \quad X_2' = X_4 Z_3\,, \text{ and } \quad Z' = Z_3 Z_4$$

# New Co-Z Formulae for Montgomery Ladder

- Final formulae

$$\begin{cases} X_1' = V[(X_1 + X_2)({X_1}^2 + {X_2}^2 - U + 2aZ^2) + 4bZ^3 - x_D ZU] \\ X_2' = U[({X_2}^2 - aZ^2)^2 - 8bZ^3 X_2] \\ Z' = UVZ \end{cases}$$

- where

$$U = (X_1 - X_2)^2 \text{ and } V = 4X_2({X_2}^2 + aZ^2) + 4bZ^3$$

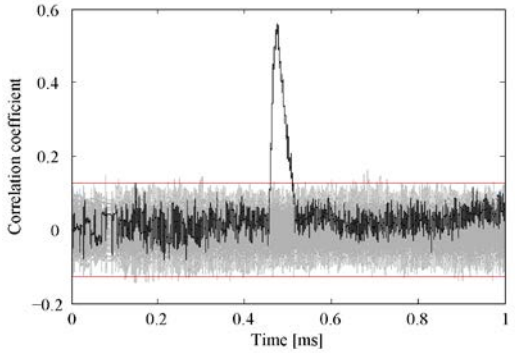
- Complexity is only 9M+5S+1M$_a$+1M$_{4b}$
- Only 7 registers necessary
- New type of formulae:
  - Out-of-place operations

→ See Africacrypt 2011

**Require:** $X_1, X_2, Z, x_D, a, 4b$
**Ensure:** $X_1, X_2, Z$

1: $R_2 \leftarrow Z^2$
2: $R_3 \leftarrow a \times R_2$
3: $R_1 \leftarrow Z \times R_2$
4: $R_2 \leftarrow 4b \times R_1$
5: $R_1 \leftarrow {X_2}^2$
6: $R_5 \leftarrow R_1 - R_3$
7: $R_4 \leftarrow {R_5}^2$
8: $R_1 \leftarrow R_1 + R_3$
9: $R_5 \leftarrow X_2 \times R_1$
10: $R_5 \leftarrow R_5 + R_5$
11: $R_5 \leftarrow R_5 + R_5$
12: $R_5 \leftarrow R_5 + R_2$
13: $R_1 \leftarrow R_1 + R_3$
14: $R_3 \leftarrow {X_1}^2$
15: $R_1 \leftarrow R_1 + R_3$
16: $X_1 \leftarrow X_1 - X_2$
17: $X_2 \leftarrow X_2 + X_2$
18: $R_3 \leftarrow X_2 \times R_2$
19: $R_4 \leftarrow R_4 - R_3$
20: $R_3 \leftarrow {X_1}^2$
21: $R_1 \leftarrow R_1 - R_3$
22: $X_1 \leftarrow X_1 + X_2$
23: $X_2 \leftarrow X_1 \times R_1$
24: $X_2 \leftarrow X_2 + R_2$
25: $R_2 \leftarrow Z \times R_3$
26: $Z \leftarrow x_D \times R_2$
27: $X_2 \leftarrow X_2 - Z$
28: $X_1 \leftarrow R_5 \times X_2$
29: $X_2 \leftarrow R_3 \times R_4$
30: $Z \leftarrow R_2 \times R_5$
2: **return** $(X_1, X_2, Z)$

# Countermeasures for ECDSA



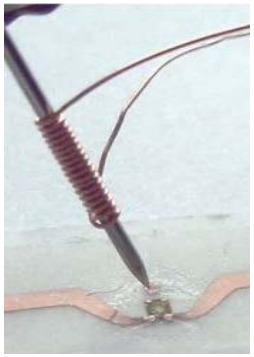

- SPA
  - Montgomery Ladder
- DPA
  - Randomized Projective Coordinates (S. Coron)
  - First-order blinding of the private-key multiplication

$$s = k^{-1}e + (k^{-1}r)d \quad \text{instead of} \quad s = k^{-1}(e + dr)$$

- Fault Injections
  - Point Validity Check (Ebeid and Lambert)

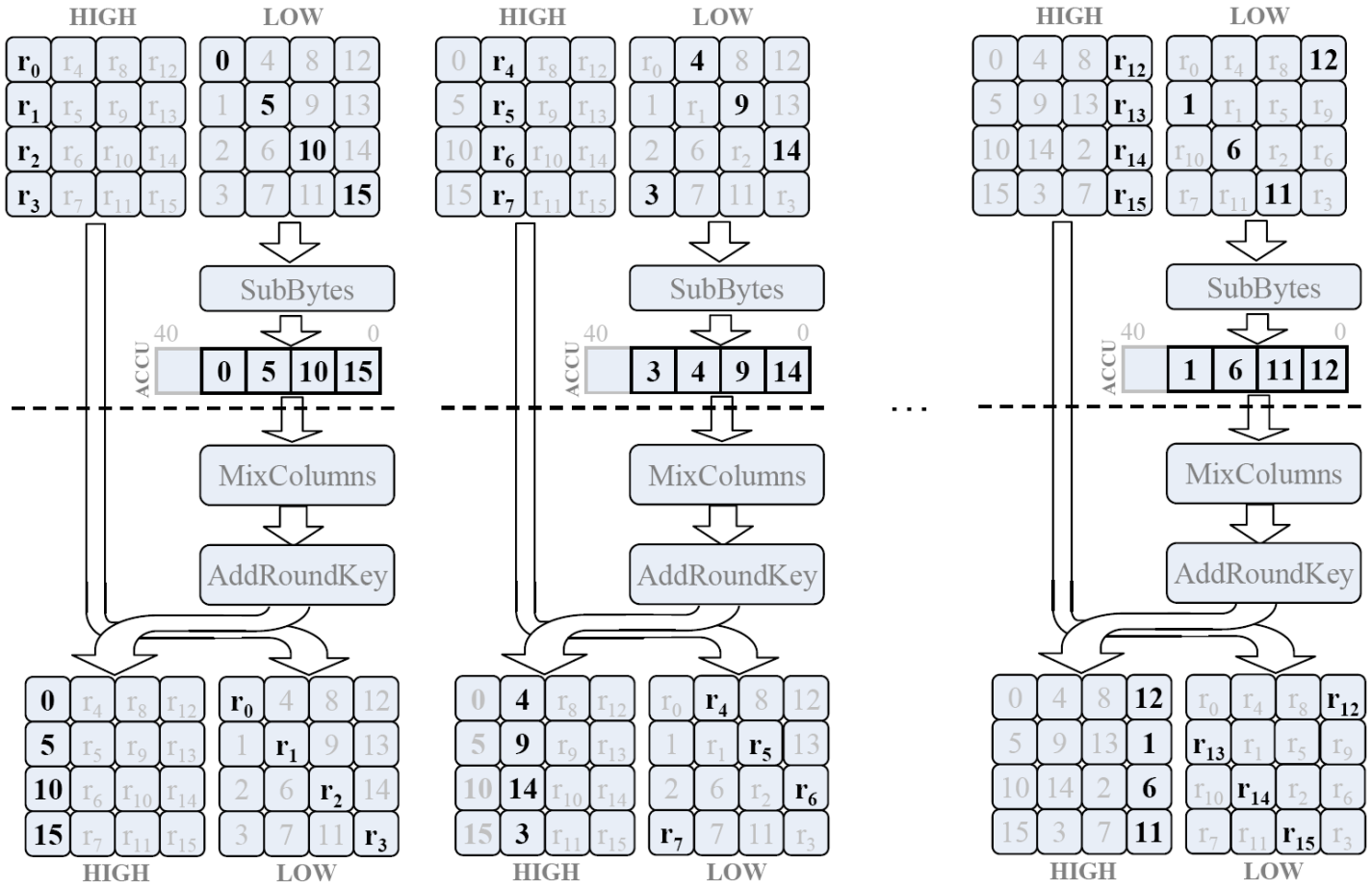
$$Z(Y^2 - bZ^2) = X(X^2 + aZ^2)$$

  - Y recovery necessary



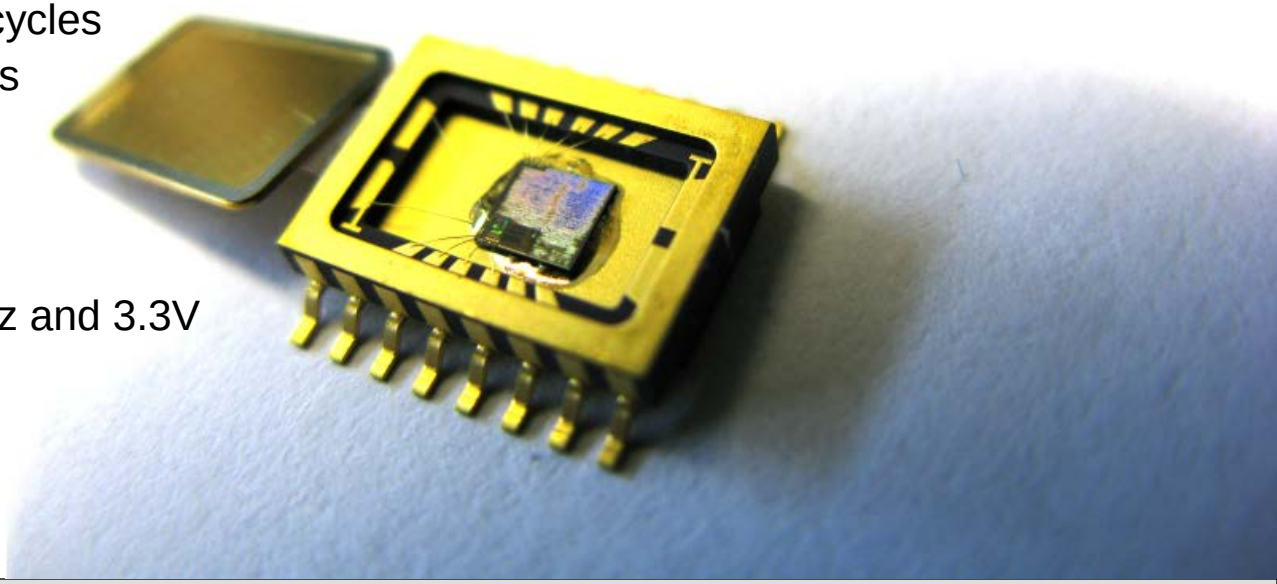
# Countermeasures for AES

- Shuffling and Dummy Operations



# Results

- Area
  - Total: 21,502 GEs
  - Overhead of AES: 2,387 GEs
  - Overhead of SHA1: 889 GEs
  - RAM macro: 8,727 GEs
- Speed
  - ECDSA: 863,109 cycles
  - SHA1: 3,639 cycles
  - AES: 4,529 cycles
- Power
  - ~485 µA @ 847kHz and 3.3V (0.35µm CMOS)
- Features
  - ISO 14443A 1-4 compliant
  - NFC Forum Type 4 compatible
  - Stores PK certificate in EEPROM



# Forgery-Proof Prototype

- Product authenticity
  - Through digital signatures
  - “Proof of Origin”
- Anti-Counterfeiting
  - Step toward preventing illicit copying intellectual property and goods
- Touch & Verify
  - Using NFC-enabled mobile phones, e.g. Nokia 6212
  - Up to 3cm reading range

# Conclusions

- Low-resource processor for RFID/NFC
  - Asymmetric crypto on Tag (ECDSA P-192)
  - Symmetric crypto (AES-128)
- Fully capable digital signature generating device
  - Allows proof of origin to prevent product counterfeiting
- State of the art countermeasures included
- Sample implementation
  - Processor has been integrated in an NFC-compliant HF tag
  - Fabricated in February 2011

# Thanks for your attention!

http://www.iaik.tugraz.at/content/research/rfid/crypta/


**Michael Hutter**

IAIK – Graz University of Technology
michael.hutter@iaik.tugraz.at
www.iaik.tugraz.at

# Comparison with Related Work

| | Area [GE] | Cycles | Field | Features |
|---|---|---|---|---|
| **This Work** | **21 674** | **863 109** | $\mathbb{F}_{p192}$ | **ECDSA, SHA-1, AES** |
| Wenger11 [WFF11] | 11 686 | 1 377 000 | $\mathbb{F}_{p192}$ | ECDSA, SHA-1 |
| Kern10 [KF10] | 18 247 | 511 864 | $\mathbb{F}_{p160}$ | SECG curve, ECDSA, SHA-1 |
| Auer08 [Aue08] | 24 745 | 1 031 000 | $\mathbb{F}_{p192}$ | ECDSA, SHA-1 |
| Fürbass07 [FW07] | 23 656 | 502 000 | $\mathbb{F}_{p192}$ | ECDSA (no SHA-1, no RNG) |
| Wolkerstorfer05 [Wol05] | 23 800 | 677 000 | $\mathbb{F}_{p192}$ | ECC |
| Öztürk04 [ÖSS04] | 30 333 | 545 440 | $\mathbb{F}_{(2^{167}+1)/3}$ | ECC |
| Satoh03 [ST03] | 29 655 | 4 165 000 | $\mathbb{F}_{p192}$ | ECC |
| Hein08 [HWF08b] | 11 904 | 296 000 | $\mathbb{F}_{2^{163}}$ | ECC |
| Bock08 [BBD+08] | 12 876 | 80 000 | $\mathbb{F}_{2^{163}}$ | ECC, DH, RNG |
| Lee08 [LSBV08] | 12 506 | 302 457 | $\mathbb{F}_{2^{163}}$ | ECC, Schnorr |
| Kumar06 [KP06] | 19 048 | 527 284 | $\mathbb{F}_{2^{193}}$ | ECC |
| Batina06 [BMS+06] | 8 104 | 353 000 | $\mathbb{F}_{2^{131}}$ | ECC, without memory |
| Schroeppel02 [SBG+03] | 191 000 | 93 000 | $\mathbb{F}_{2^{178}}$ | ECC, ElGamal, PRNG |
| Koschuch06 [KLW+06] | 29 491 | 1 416 000 | $\mathbb{F}_{2^{191}}$ | ECC, 8051 $\mu$C |
| Aigner04 [ABHW04] | 25 000 | 469 385 | $\mathbb{F}_{2^{191}}$ | ECDSA (no SHA-1, no RNG) |

# Montgomery Ladder

**Algorithm** Point-multiplication method using the improved Montgomery ladder in common Z projective coordinate system.

**Require:** Base point $P = (P_x, P_y) \in E(\mathbb{F}_{p192}), k \in_R [1, 2^{192} - 1]$, random $\lambda$
**Ensure:** $Q = kP$, where $Q = (x, y) \in E(\mathbb{F}_{p192})$
1: $X_0 \leftarrow \lambda P_x$, $Z_0 \leftarrow \lambda$
2: $(X_1, Z_1) \leftarrow Dbl(P)$
3: $X_0 \leftarrow X_0 \cdot Z_1$, $X_1 \leftarrow X_1 \cdot Z_0$, $Z \leftarrow Z_0 \cdot Z_1$
4: **for** $i = 190$ downto 0 **do**
5: $\quad (X_{k_i \otimes 1}, X_{k_i}, Z) \leftarrow CombinedDoubleAndAdd(X_{k_i}, X_{k_i \otimes 1}, Z)$
6: **end for**
7: $Y_0 \leftarrow Yrecovery(X_0, X_1, Z, P)$
8: **if** $Z'(Y_0'^2 - bZ'^2) \neq X_0'(X_0'^2 + aZ'^2)$ **abort.**
9: $x \leftarrow X_0 \cdot Z^{-1}$
10: Return $(x)$.