



Light-Weight Cryptography for Ubiquitous Computing

Securing Cyberspace Workshop IV: Special purpose hardware
for cryptography – Attacks and Applications
University of California at Los Angeles, December 4, 2006

Christof Paar
Ruhr-University of Bochum
www.crypto.rub.de

Acknowledgements

Joint work with

- Sandeep Kumar
- Gregor Leander
- Axel Poschmann
- Kai Schramm

Contents

1. Security in Embedded Systems
2. Light-Weight Block Ciphers
3. Light-Weight Asymmetric Cryptography

Contents

1. **Security in Embedded Systems**
2. Light-Weight Block Ciphers
3. Light-Weight Asymmetric Cryptography

What are Embedded Systems?



- „Processor hidden in a product“, or
- „A computer that doesn't look like a computer“

Characteristics of Embedded Systems

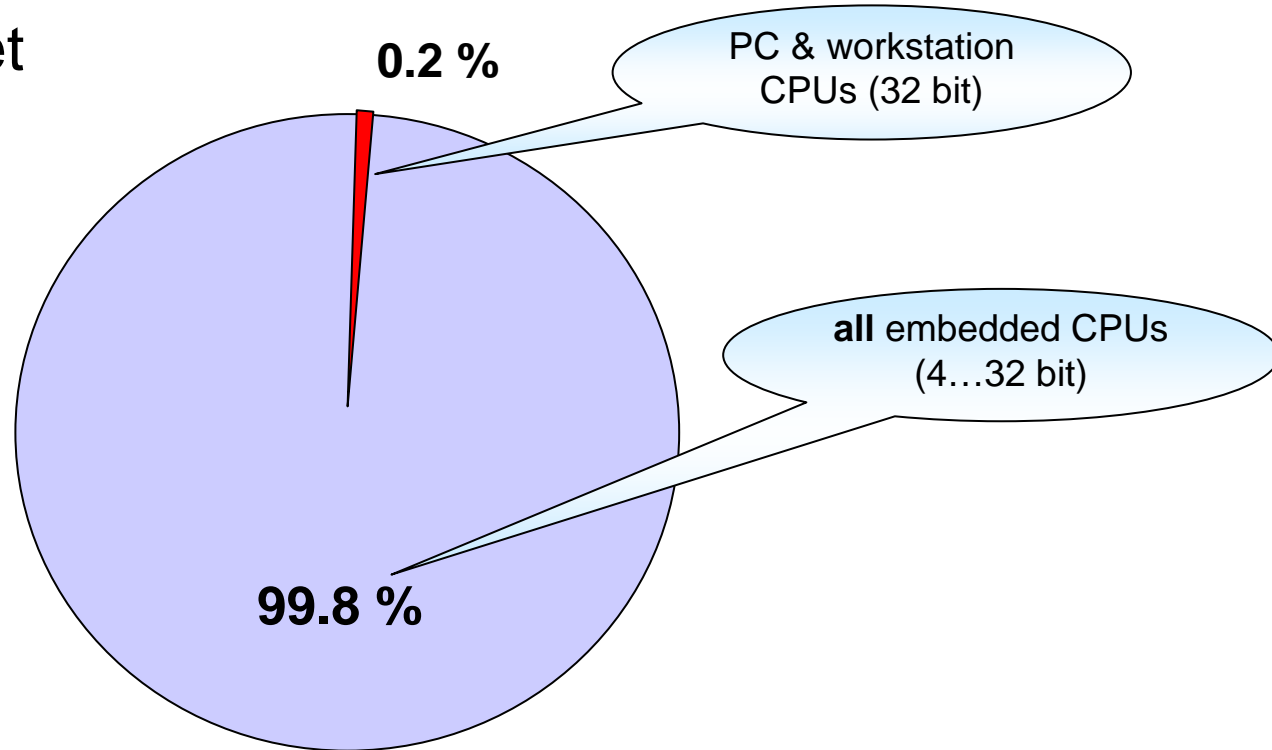
- Single purpose device



- Interacts with the world
- many,many applications

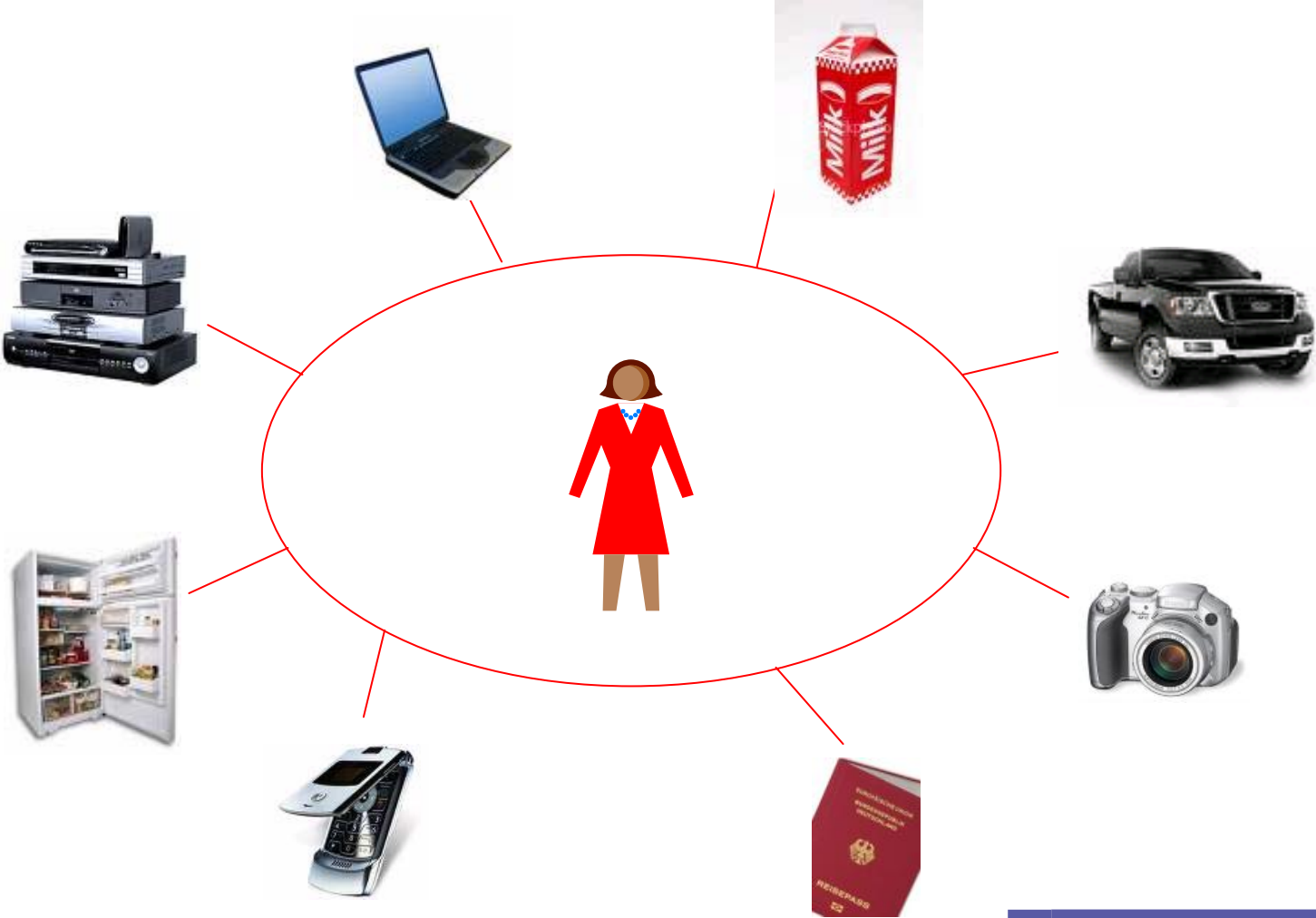
Is this really important ?

current CPU market
by the numbers



So, how does embedded technology affect the future IT landscape?

Brave New Pervasive World



Contents

1. Security in Embedded Systems
- 2. Light-Weight Block Ciphers**
3. Light-Weight Asymmetric Cryptography

Light-Weight Cryptography

- “We need security with less than 2000 gates”
Sanjay Sarma, AUTO-ID Labs, CHES 2002



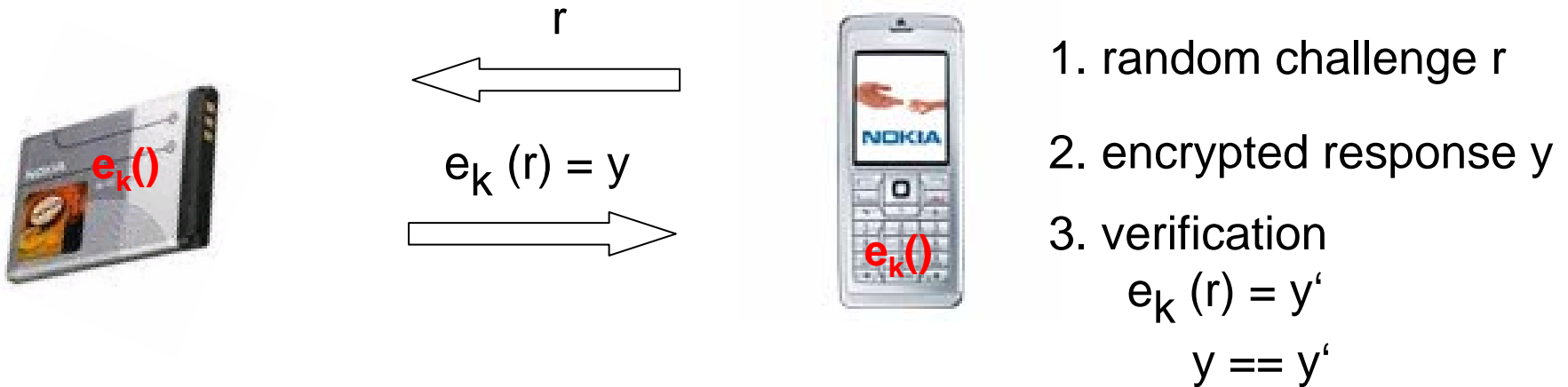
- \$3 trillions annually due to product piracy* (> US budget '07)



*Source: www.bascap.com

- ⇒ Authentication & identification problem: can both be fixed with cryptography
- ⇒ How cheap can we make crypto algorithms?

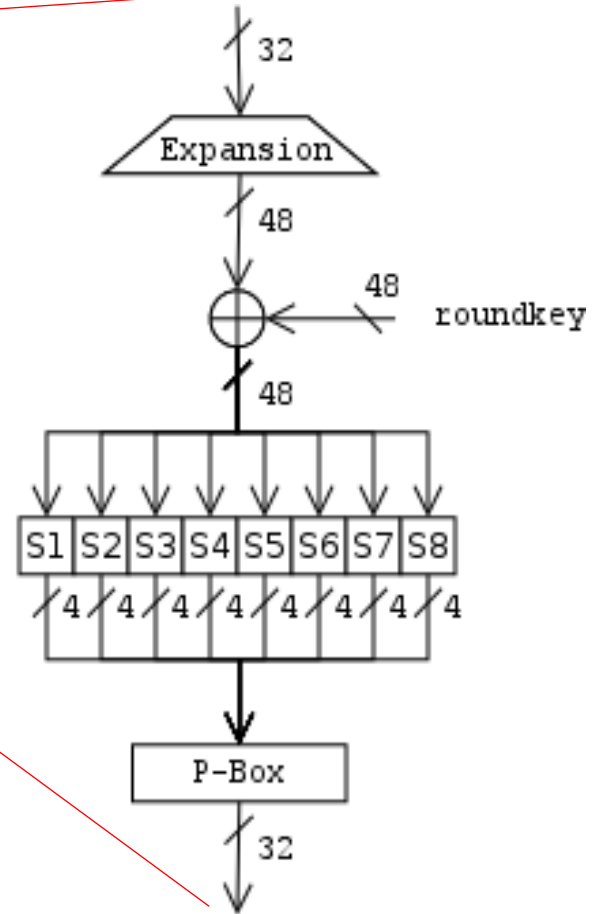
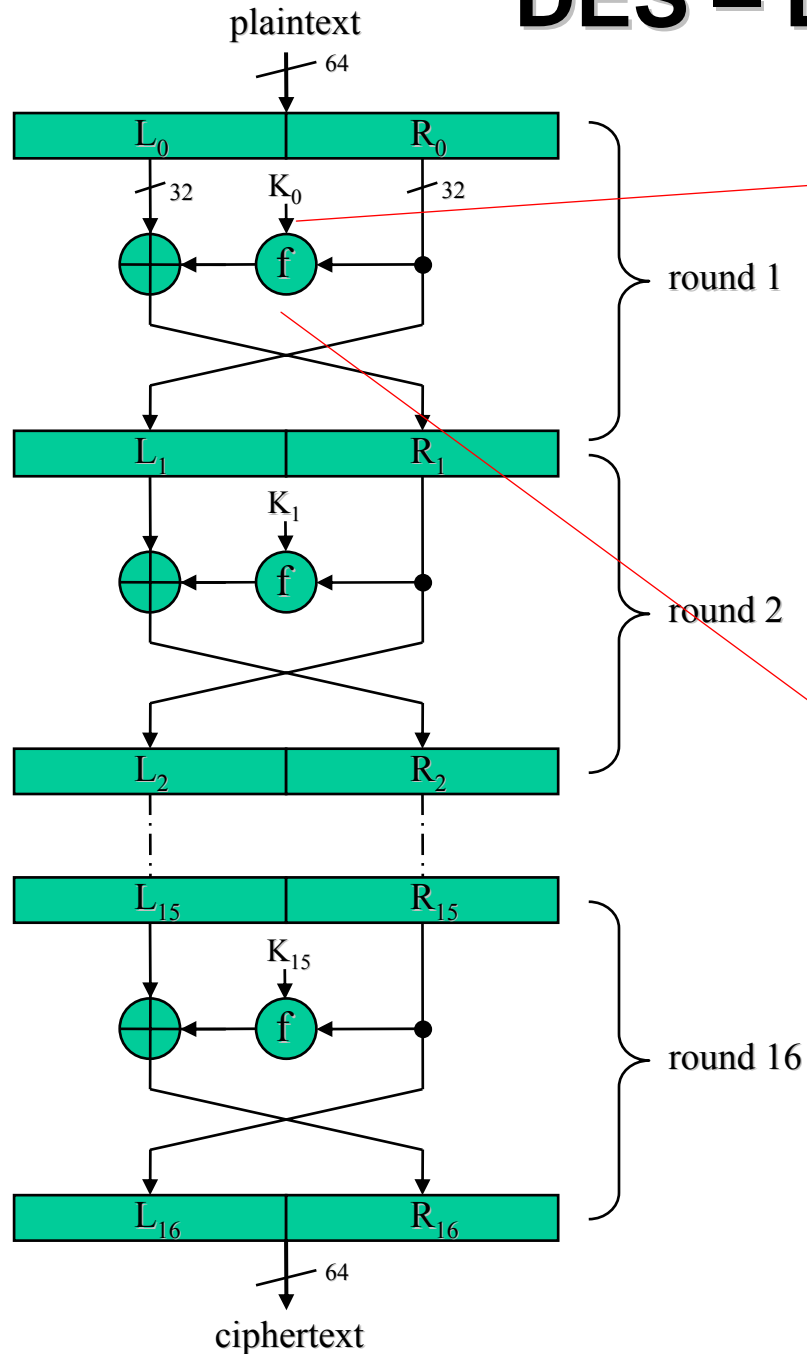
Strong Identification (w/ symmetric crypto)



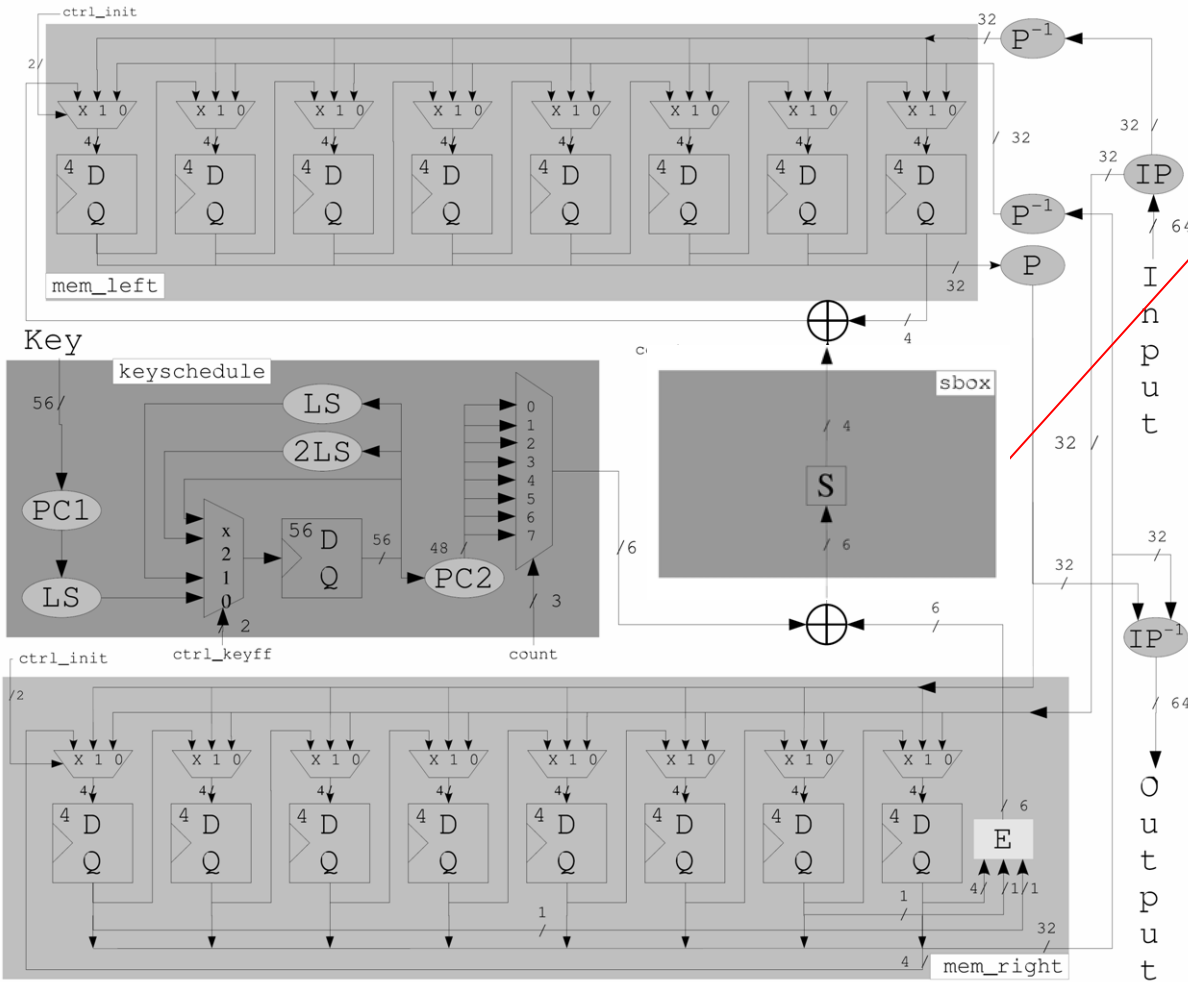
Challenge: Encryption function $e()$ at extremely low cost

- almost all symmetric ciphers optimized with SW in mind
- exception: DES

DES – Data Encryption Standard

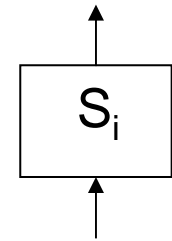


Serialized DES Architecture



S-Boxes

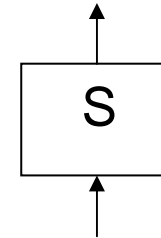
- 4-to-6 substitution tables



- crucial for security
- highly non-linear
→ high Boolean compl.
- 32% of area!

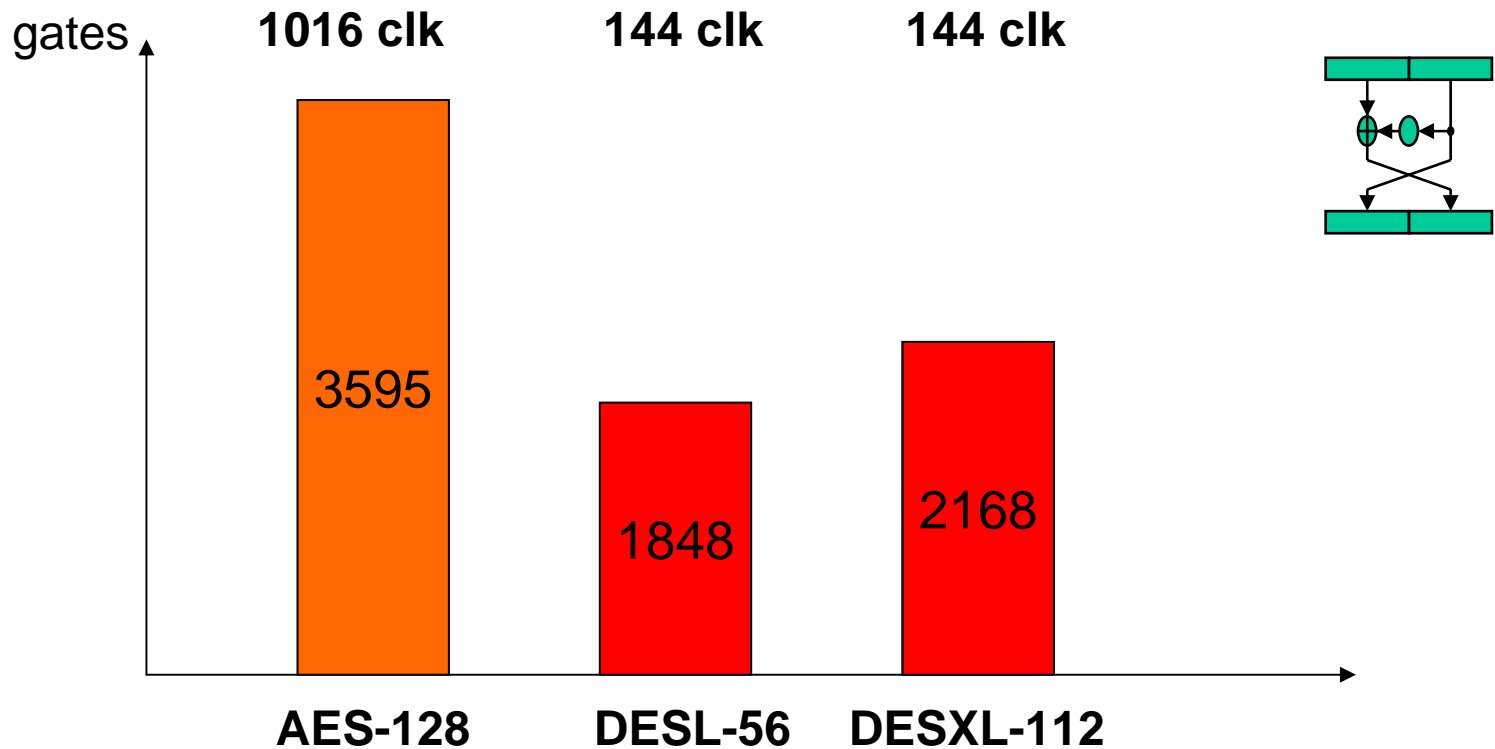
DESL: A Single S-Box DES Variant

- DESL: replacing $S_1 \dots S_8$ by S
- non-trivial problem
- no previous work (!)
- S must be robust against differential, linear, and David-Murphy attack
- New S more robust against known attacks than $S_1 \dots S_8$



S															
14	5	7	2	11	8	1	15	0	10	9	4	6	13	12	3
5	0	8	15	14	3	2	12	11	7	6	9	13	4	1	10
4	9	2	14	8	7	13	0	10	12	15	1	5	11	3	6
9	6	15	5	3	8	4	11	7	1	12	2	0	14	10	13

Results – Light-Weight DES

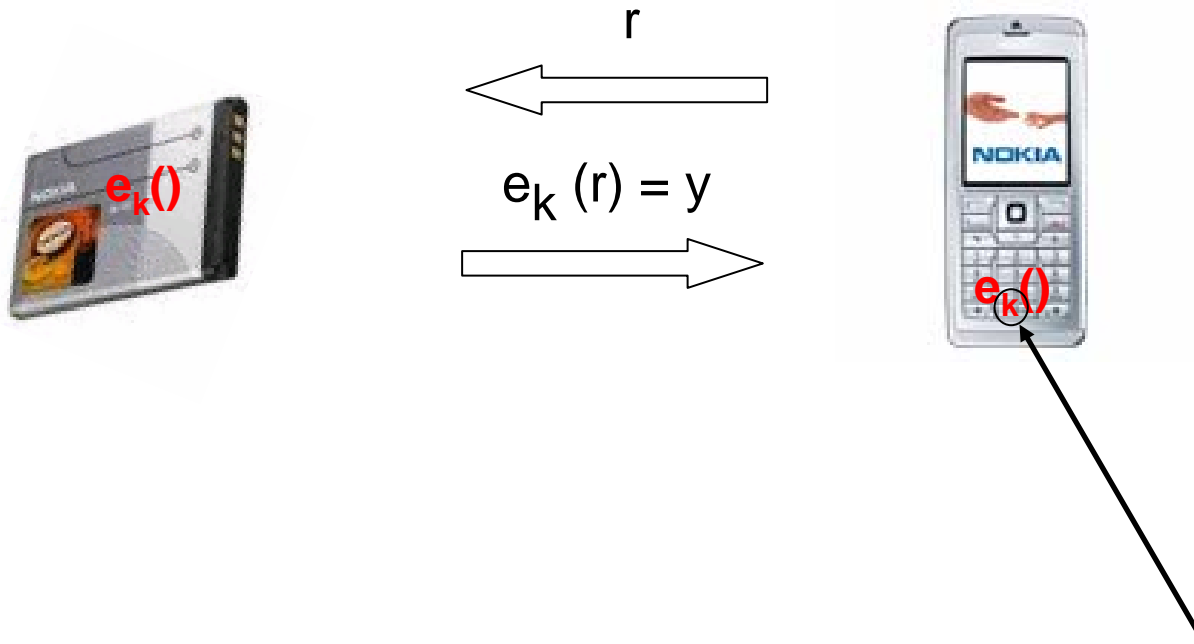


- smallest known secure block cipher
- TA product 12-14 times better than smallest AES architecture
- only block cipher based on HW-optimum design

Contents

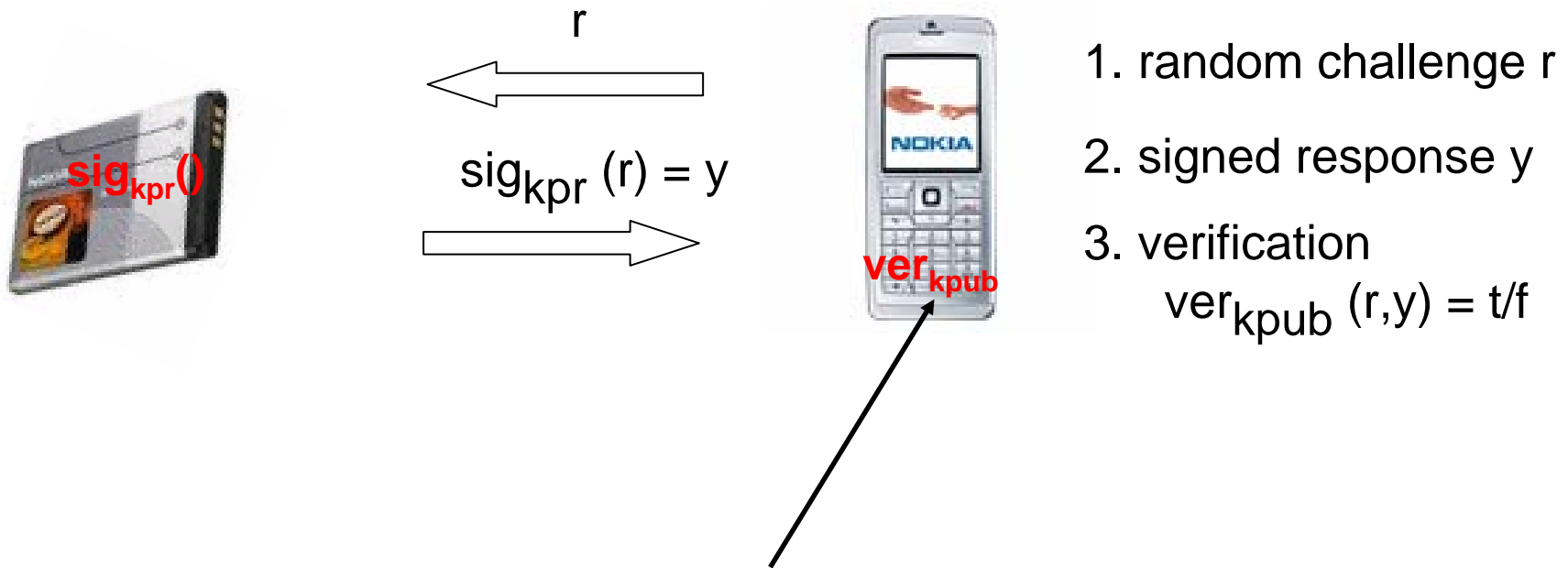
1. Security in Embedded Systems
2. Light-Weight Block Ciphers
3. **Light-Weight Asymmetric Cryptography**

Strong Identification (w/ symmetric crypto)



Potential weakness: attacker gets access to key on host device (e.g. firmware exploits) and starts cloning batteries

Strong Identification (w/ asymmetric crypto)



Attacker can only access public key from host device

- But how cheap can we build public-key algorithms?
- Idea: use OTS 8bit μP (< \$1)

Elliptic Curve Primitive

- Given a Point P on an elliptic curve E over $GF(p)$:

$$E: y^2 = x^3 + ax + b \pmod{p}$$

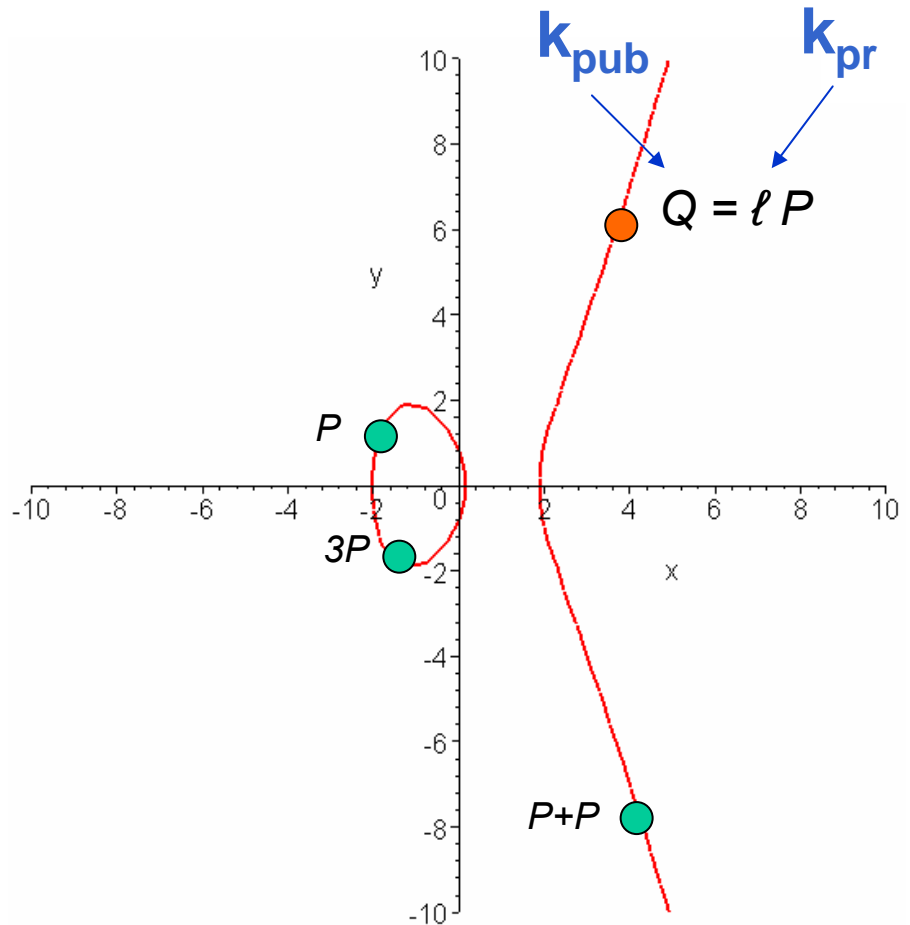
- Public key Q is multiple of base point P

group operation

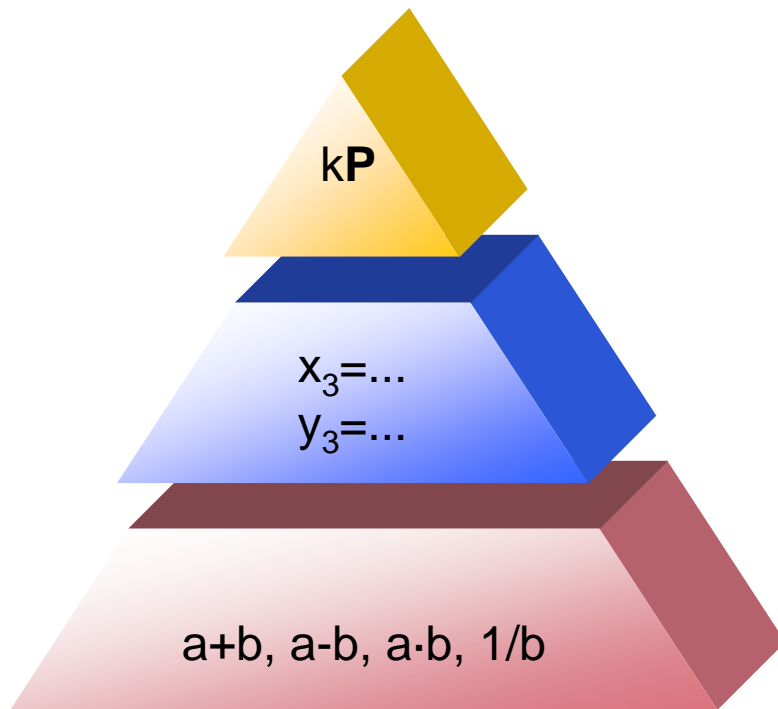
$$Q = P + P + \dots + P = \ell P$$

- EC discrete logarithm problem:

$$\ell = d\log_P(Q)$$



ECC System Design



- Protocol
 - Point Mult ($k \cdot P$)
- Group Operation
 - Point Add/Double
- Field Operations
 - Addition/Subtraction
 - Multiplication
 - Reduction
 - Inverse

Design Principles for *Tiny ECC Processor*

- Reduce memory requirements : memory amounts to more than 50% of design
- Reduce arithmetic unit area : avoid units like inverter + designed for specific size
- Keep it simple but efficient : reduce control logic area - multiplexers

Tiny ECC Processor Units

- Arithmetic Units

- Multiplier

- Most-Significant Bit Mult.

- Squarer

- inverter

- Point Multiplier

- Control Unit

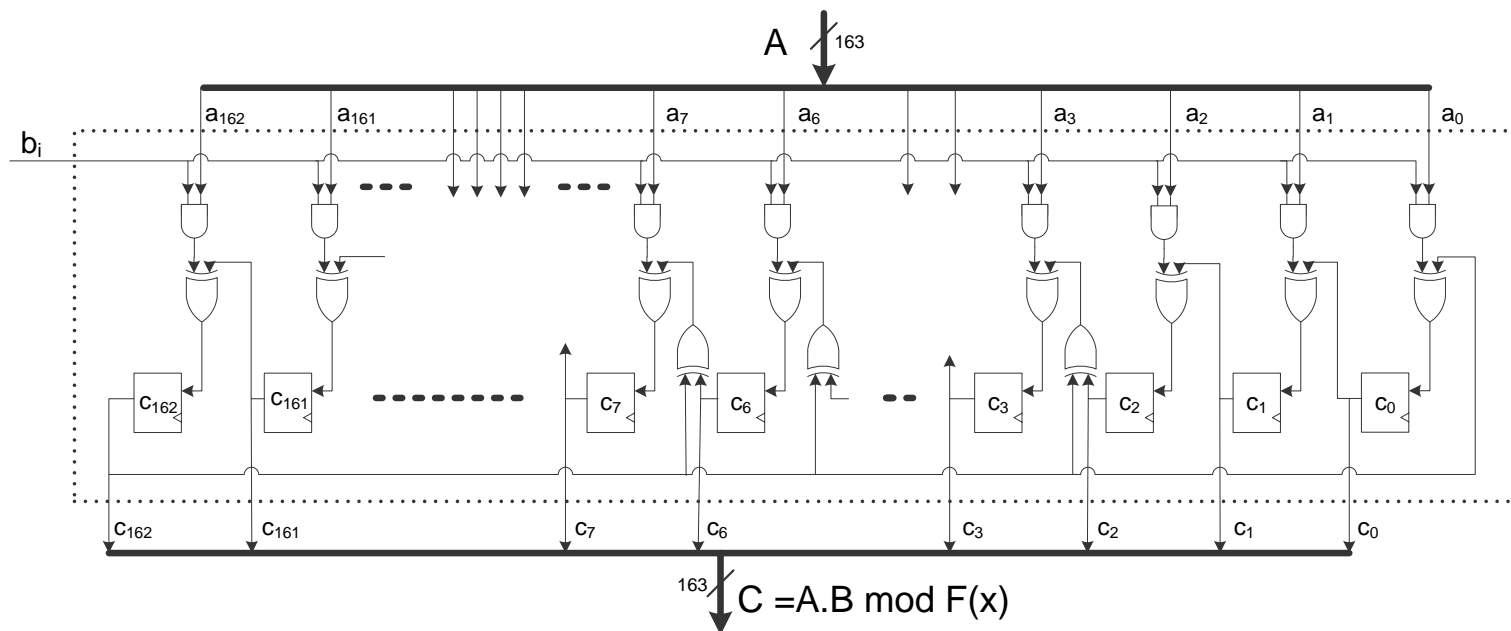
- Memory Unit

Most Significant Multiplier

- $A, B \in GF(2^n)$
- $A(x) = a_{m-1}x^{m-1} + \dots + a_1x + a_0$
- $C(x) = A(x) \times B(x)$
= $A \times \sum b_i x^i \text{ mod } F(x)$
= $(\dots(A \times b_{m-1}x + A \times b_{m-2})x \dots)x + A \times b_0 \text{ mod } F(x)$

The Implementation: MSB Multiplier

$$C(x) = A(x) \times B(x) = (\dots(A \times b_{m-1}x + A \times b_{m-2})x \dots)x + A \times b_0 \text{ mod } F$$



Most-Significant Bit (MSB) Multiplier: N cycles for n-bit multiplier

Tiny ECC Processor: Design decisions

- Arithmetic Units
 - Multiplier
 - Squarer
 - inverter
 - Point Multiplier
 - Control Unit
 - Memory Unit
- Most-Significant Bit Mult.
 - Parallel Squaring

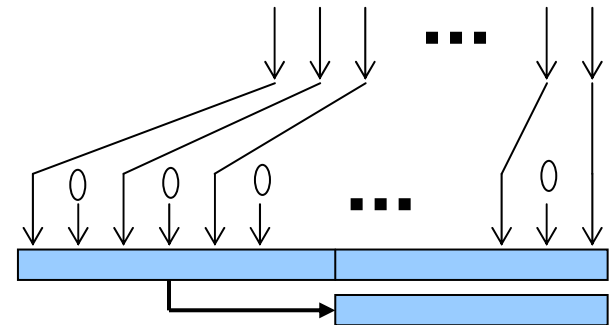
Squaring

- $A \in GF(2^n)$
- $A(x) = a_{m-1}x^{m-1} + \dots + a_1x + a_0$
- $A^2(x) =$

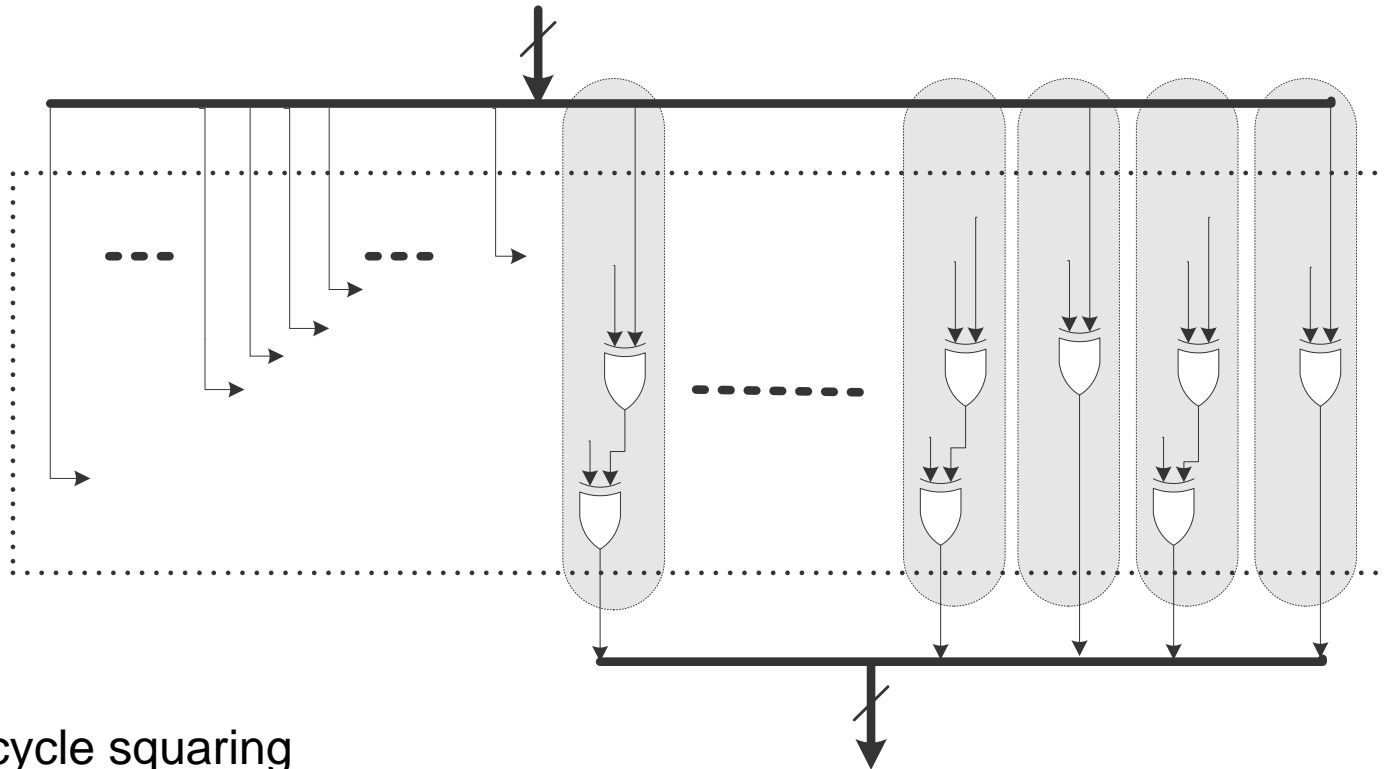
Step1: $a_{m-1}x^{2(m-1)} + \dots + a_1x^2 + a_0$

Step2: $(a_{m-1}x^{2(m-1)} + \dots + a_1x^2 + a_0) \bmod F(x)$

$$= (a_{m-1}x^{2(m-1)} + \dots + a_{m/2}x^m) \bmod F(x) + (a_{m/2-1}x^{(m-2)} + \dots + a_1x^2 + a_0)$$



The Implementation: Squarer



- single cycle squaring
- low gate count
- low critical path

Tiny ECC Processor Units

- Arithmetic Units
 - Multiplier
 - Squarer
 - inverter
 - Most-Significant Bit Mult.
 - Parallel Squaring
 - Fermat's Little Theorem
- Point Multiplier
 - Control Unit
- Memory Unit

The Implementation: inverter

Fermat's Little Theorem

$$A^{-1} \times A^{2^m-2} \bmod F(x) \text{ if } A \in GF(2^m)$$

For $m=163$: $A^{2^{163}-2}$

Straightforward exponentiation: 161 **Mult.** + 162 **Sqr.**

Exploit exponent structure:

Inversion using Itoh-Tsujii

$$A^{2^{163}-2} = A^{\underbrace{[111 \cdots 1]_2}_{162} 0}_2$$

$$\underbrace{[111 \cdots 1]_2}_{162} = \underbrace{[111 \cdots 1]_2}_{81} \cdot 2^{81} + \underbrace{[111 \cdots 1]_2}_{81}$$

$$\underbrace{[111 \cdots 1]_2}_{80} \cdot 2 + 1$$

$$\underbrace{[111 \cdots 1]_2}_{40} \cdot 2^{40} + \underbrace{[111 \cdots 1]_2}_{40}$$

The Implementation: Inverter

Fermat's Little Theorem

$$A^{-1} \times A^{2^m-2} \text{ mod } F(x) \text{ if } A \in \text{GF}(2^m)$$

For $m=163$: $A^{2^{163}-2}$

Straightforward exponentiation: 161 **MUL** + 162 **SQ**

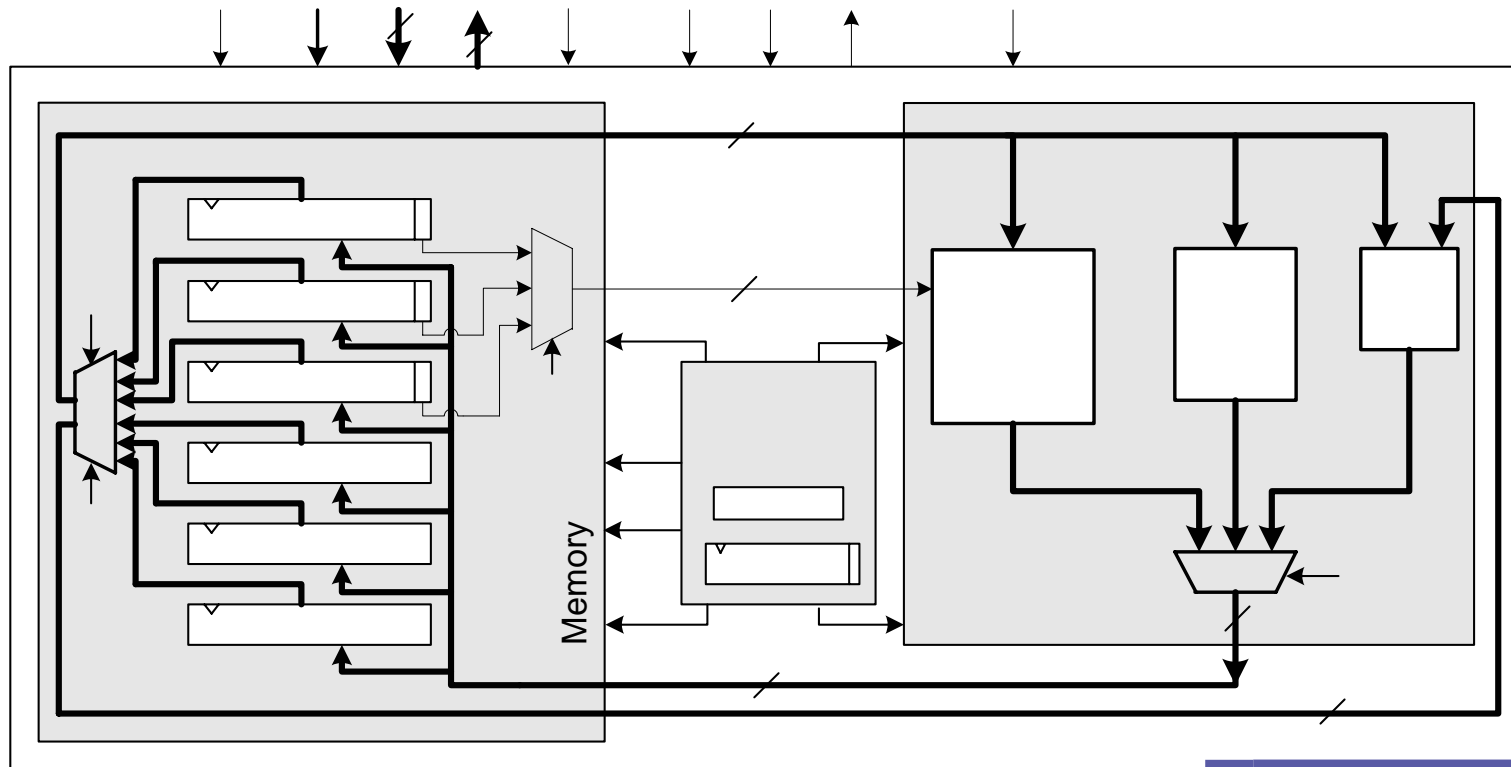
Exploit exponent structure:

$(\log_2(m-1) + \text{HW}(m-1) - 1)$ **MUL** + $(m-1)$ **SQ**

For $m=163$: 9 **MUL** + 162 **SQ**

The Tiny ECC Processor Design

- ECC processor implementation for 2^{113} , 2^{131} , 2^{163} , 2^{193}



Performance and Results

Performance @ 4 MHz for standardized curves

Field Size	Arithmetic Unit(gates)	Memory (gates)	Total (gates)	Time (ms)
113	1,625	6,686	10,112	47
131	2,071	7,747	11,969	61
163	2,572	9,632	15,094	108
193	2,776	11,400	17,723	139

131, 163 bit: very practical bit sizes

Security levels?

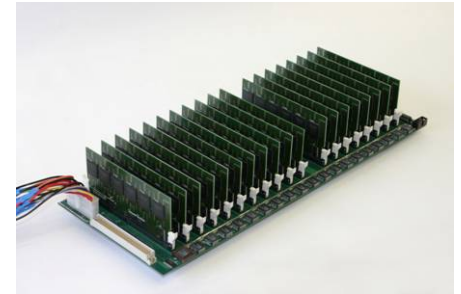
Security of mid-size ECC

Costs for breaking ECC in *one year*
w/ optimized attack ASICs:

ECC131p \approx \$2 million

ECC163p: \approx \$1 trillion (> 20 years security)

cf [CHES06 & Jan Pelzl's talk at this workshop]



Related Workshops



escar – Embedded Security in Cars
November 2006, Berlin, Germany

SASC – Stream Ciphers Revisited
January 2007, Bochum, Germany



RFIDSec 2007
January 2007, Malaga, Spain

CHES 2007
Vienna, Austria



CHES – Cryptographic Hardware and Embedded Systems
September 2007, Vienna, Austria