Horst-Görtz Institut
für IT Sicherheit

On the Security of Elliptic Curve Cryptosystems against Attacks with Special-Purpose Hardware

SHARCS '06
April 3-4, 2006, Cologne

Tim Güneysu, Christof Paar, Jan Pelzl

/www.crypto.rub.de

Agenda

Horst-Görtz Institut für IT Sicherheit

Introduction and Motivation

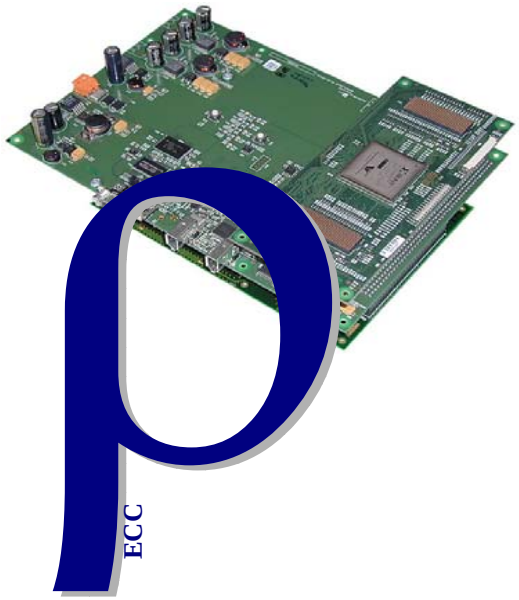
Mathematical Background

Implementation of Pollard‘s-Rho in Hardware

State of the Art in ECC

Results and Projections

Conclusions



Agenda

Horst-Görtz Institut für IT Sicherheit

Introduction and Motivation

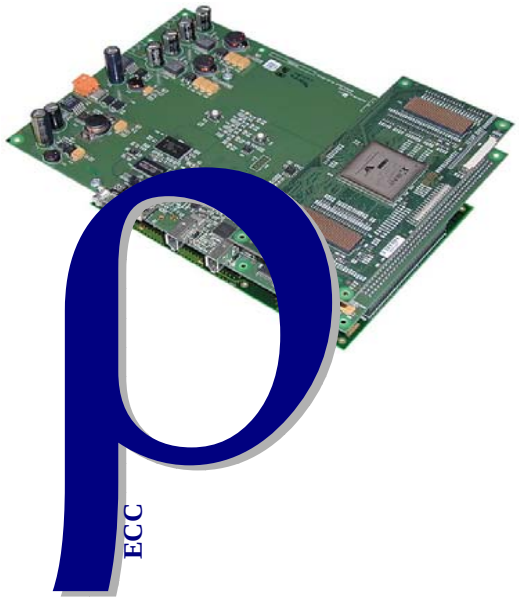
Mathematical Background

Implementation of Pollard‘s-Rho in Hardware

State of the Art in ECC

Results and Projections

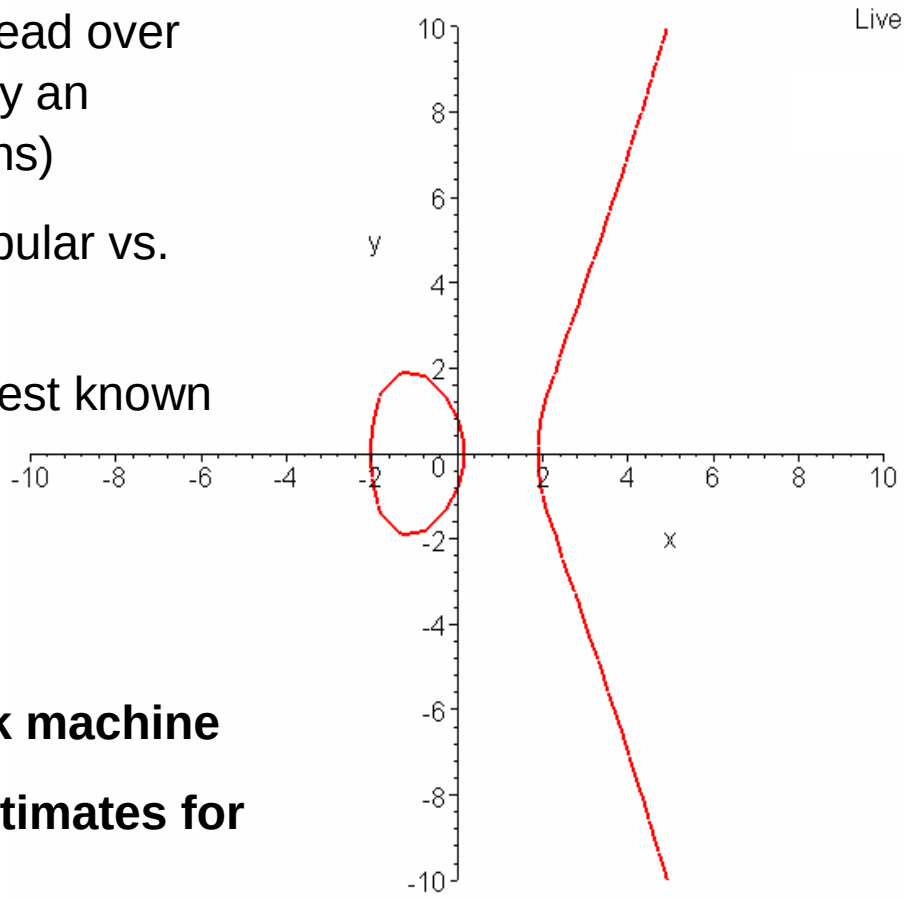
Conclusions



Motivation

Horst-Görtz Institut für IT Sicherheit

- ECC has become more wide-spread over the last 10 years (in part driven by an increase in embedded applications)
- Trend: ECC over GF(p) more popular vs. GF(2^m)
- In general: Generic attacks are best known attacks.



This work:

1. **First (?) hardware-based attack machine**
2. **More exact security (= cost) estimates for real-world ECC bit lengths.**

ECDL Problem

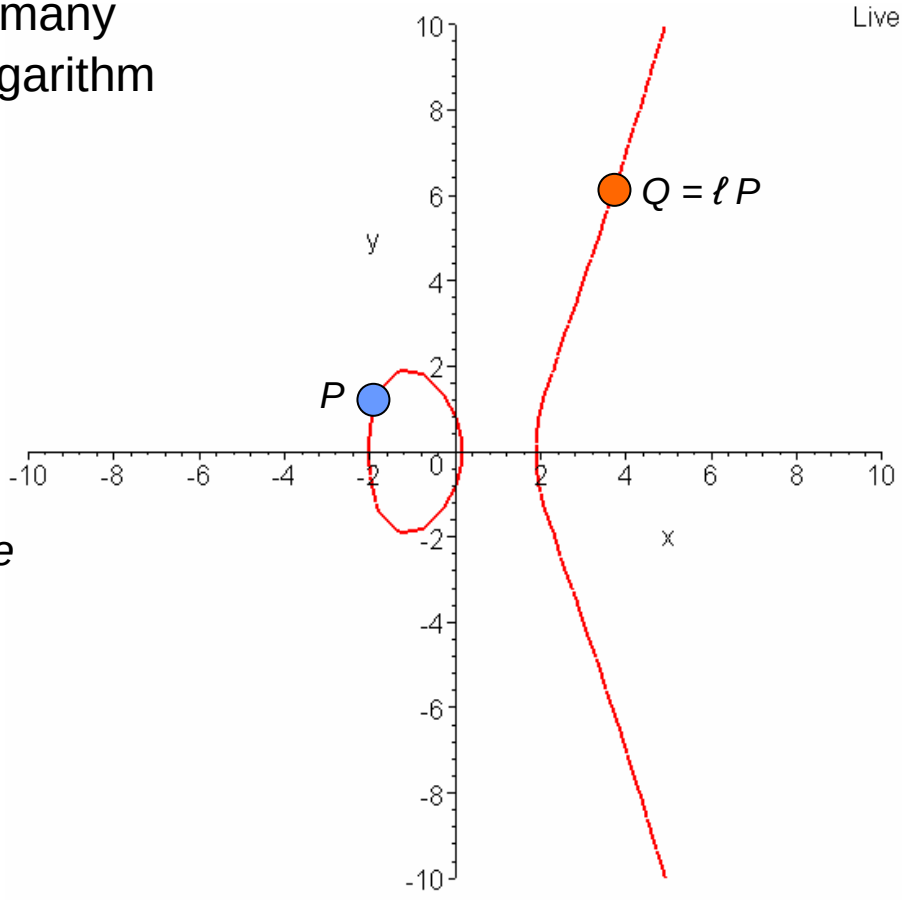
A cryptographic primitive of ECC used in many protocols is the Elliptic Curve Discrete Logarithm Problem (ECDLP)

- Given a Point *P* on an elliptic curve

  $E: y^2=x^3+ax+b$ over GF(p)

  with *p* prime and point order $n = ord(P)$
- Let P be a generator. Determine *discrete logarithm* ℓ of a point *Q such that*

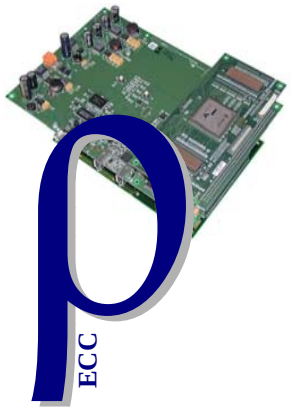
  $Q = \ell P.$



EC Attacks I: Generic Attacks

Horst-Görtz Institut für IT Sicherheit

Known generic methods to solve the ECDLP over GF(p)

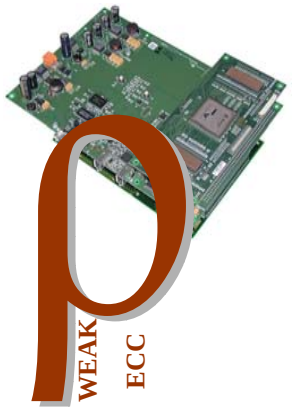


1. **Naïve Search**: Sequentially test P, *2P, 3P, 4P*,…
   - Brute force attack is infeasible EC groups larger ≈ 2^{80}
2. **Shank's Baby-Step-Giant-Step Method**
   - Complexity in time AND memory of about $\sqrt{n}$
3. **Pollard's Lambda method** (λ)
   - Efficient method for bounded search within an interval $1<b<n$
   - Complexity dependant on bound b with $3.28\sqrt{b}$
4. **Pollard's Rho method** (ρ)
   - Most efficient algorithm for solving general ECDLP known so far
   - Available proposal for parallel implementation
   - Complexity of $\sqrt{(\pi n / 2)}$

Note: All attacks are *exponentia*l in the bit length of the group order

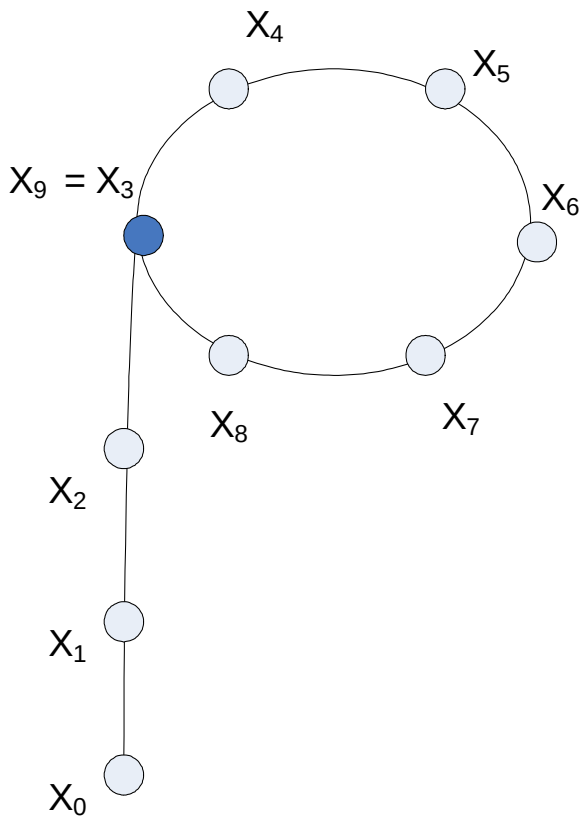
EC Attacks II: Special Curves

Known methods to solve the ECDLP on special (weak) EC over GF(p) with *subexponential* complexity



- Supersingular Curves
- Anomalous Curves (Curves over GF(p) with exactly p points)
  - Attack by Araki-Satoh-Semaev-Smart
- Curves vulnerable to Weil and Tate Pairing attacks
  - Attack in polynomial time when $n \mid q^k - 1$ for small k

Single Processor Pollard Rho (SPPR)



Collision path of pseudo-random walk

SPPR originally proposed by J. Pollard in 1978

Idea: Find a collision of two arbitrary points X_k, X_l while monitoring their relative distance to P and Q via

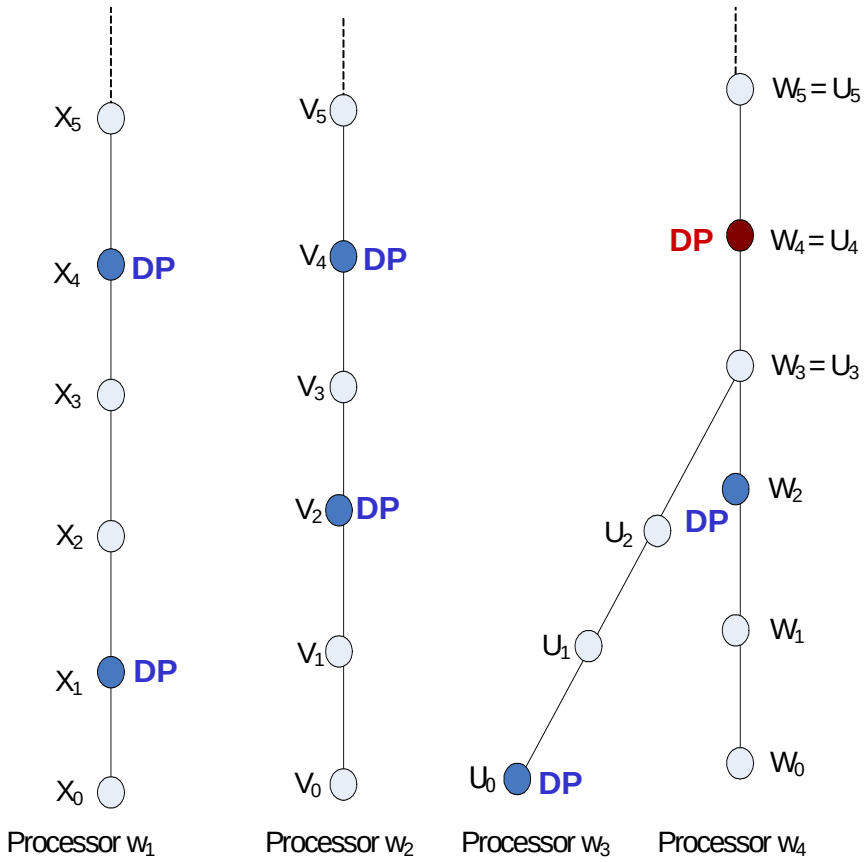
$$c_k P + d_k Q = X_k = X_l = c_l P + d_l Q.$$

Then, the ECDLP is given by

$$\ell = (c_k - c_l)\,(d_l - d_k)^{-1} \bmod n$$

Collisions are detected with Floyd's cycle-finding algorithm using a pseudo random walk

Multi Processor Pollard Rho (MPPR)



Colliding DP trails of multiple processors w_i

MPPR proposed by van Oorschot and Wiener in 1999

Multiple processors have individual search paths for “Distinguished Points” (DP) which are sent to a central server

Duplicate distinguished points detected on the server reveal ECDLP

Advantage: Linear speed-up by number of employed processors

Agenda

Horst-Görtz Institut für IT Sicherheit

Introduction and Motivation

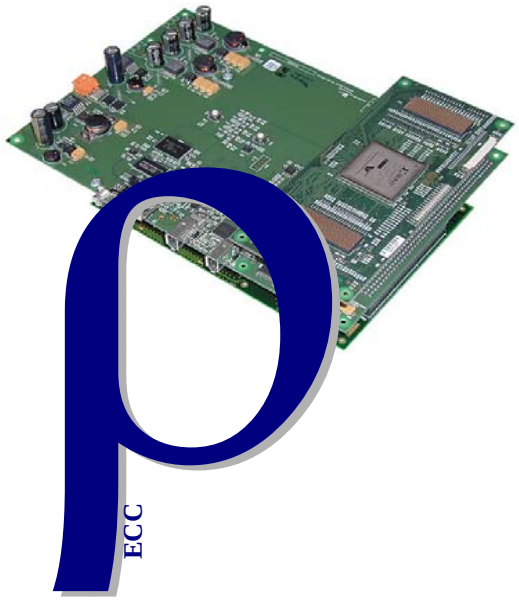
Mathematical Background

Implementation of Pollard‘s-Rho in Hardware

State of the Art in ECC

Results and Projections

Conclusions



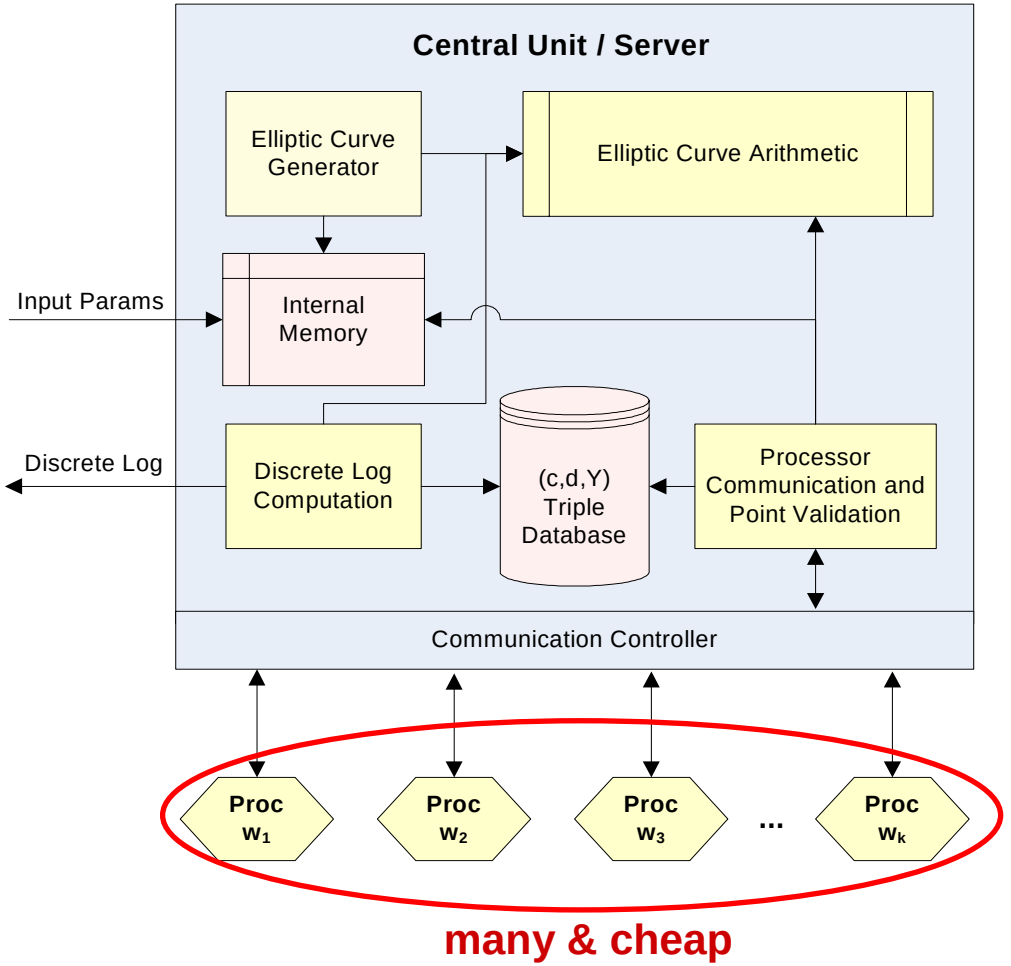
General MPPR Attack Model

Central server (Software based)

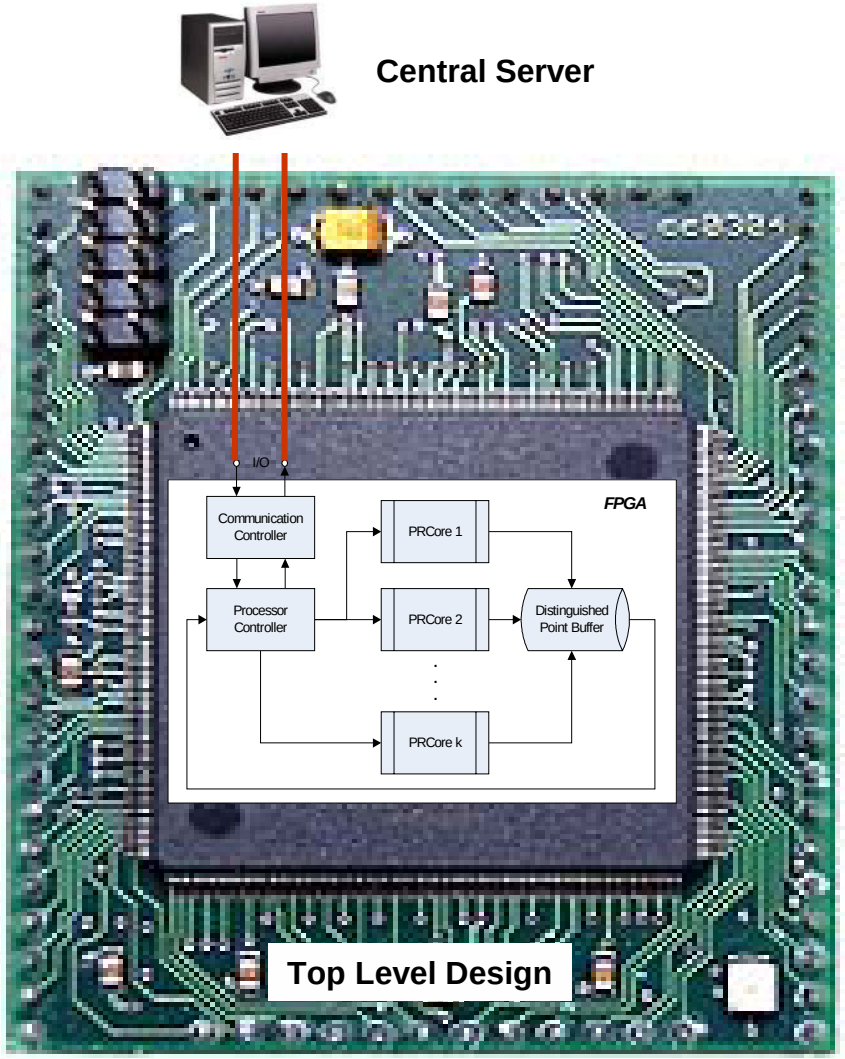
- Administrative tasks
- Centralized DP database
- Manages attached point processors w_i

Point processors w_i (HW based)

- Compute distinguished points and transfer them to server
- Implemented as an large array of FPGAs or ASICs
- FPGAs offer more design flexibility and will be used for a first implementation



Hardware Implementation (Top Layer)



Neither

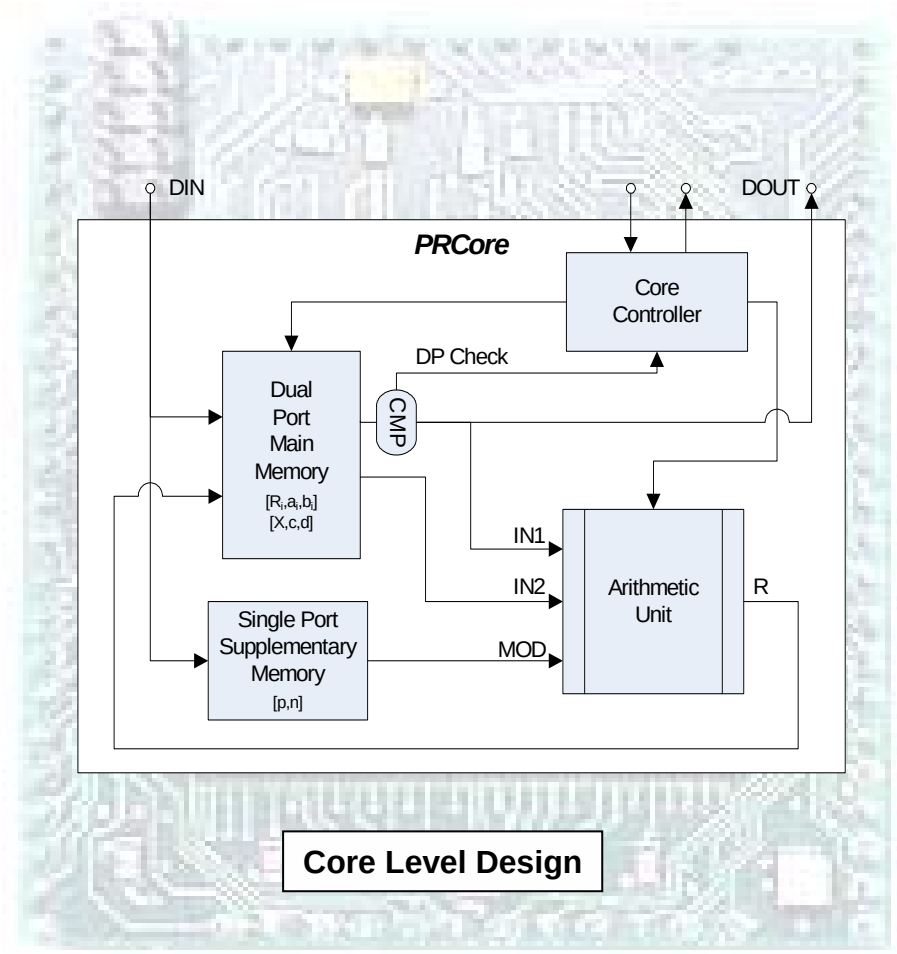
- fastest, nor
- smallest

implementations is needed, but

- **Time-Area Optimum**.

- Each FPGA: multiple point engines (PRCore) each computing a separate trail.
- All cores store distinguished points in a shared point buffer.
- Buffer locking & host communication are needed to transfer DPs to the server.
- FPGA to Host communication via serial (for debugging) or proprietary bus interface.

Hardware Implementation (Core Layer)



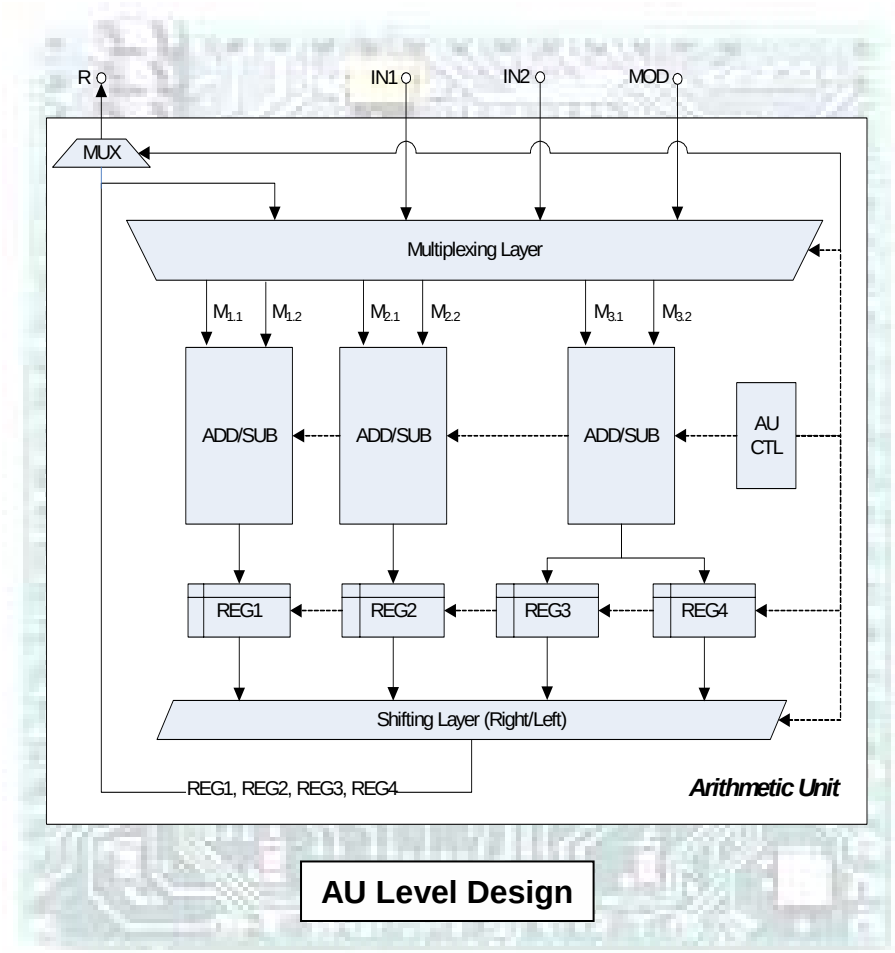
Core Level Design

- Each core has an **Arithmetic Unit** (AU) for modular computations.
- Storage for current point X_i and coefficents c_i, d_i with $X_i = c_iP + d_iQ$
- 16 random points $R_1 \ldots R_{16}$
- Pseudo-random walk

$$X_{i+1} = X_i + R_\theta$$

- Distinguished point detection unit (comparison if m LSBs are zero)

Hardware Implementation (AU Layer)



AU Level Design

ECC Computations use **affine coordinates** to preserve a simple DP property.

Modular operations: Addition, Subtraction, Multiplication and Inversion

AU uses **Montgomery representation** (MR) for efficient modular reduction:

- Montgomery Multiplication
- Modified Kaliski Inversion Algorithm

Search for DPs is performed completely in Montgomery domain.

Agenda

Horst-Görtz Institut für IT Sicherheit

Introduction and Motivation

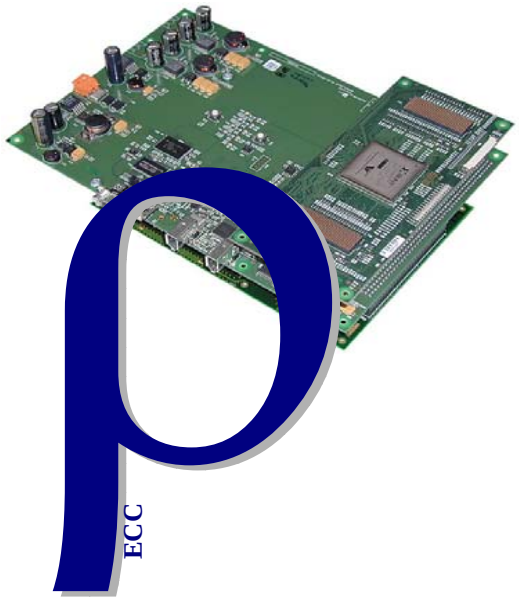
Mathematical Background

Implementation of Pollard‘s-Rho in Hardware

State of the Art in ECC

Results and Projections

Conclusions



ECC Attacks: Status Quo

Horst-Görtz Institut für IT Sicherheit

Certicom challenges for ECC over GF(P) and GF(2^m) .

The 109-bit challenges have been solved by Pollard-Rho clusters:

- *ECC(p)-109 solved in Nov. 2002*
- *ECC(2^m)-109 solved in April 2004*

For ECC(p)-109, it took 10,000 computers (mostly PCs) running 24 hours a day for 549 days!

ECC(p) Challenge	Est. time to solve* (days)	Status
ECC(p)-79	146	**SOLVED**
ECC(p)-89	4360	**SOLVED**
ECC(p)-97	71982	**SOLVED**
ECC(p)-109	9.0×10^6	**SOLVED**
ECC(p)-131	2.3×10^{11}	UNSOLVED
ECC(p)-163	2.3×10^{15}	UNSOLVED
ECC(p)-191	4.8×10^{19}	UNSOLVED
ECC(p)-239	1.4×10^{27}	UNSOLVED

* based on Pentium 100

Agenda

Horst-Görtz Institut für IT Sicherheit

Introduction and Motivation

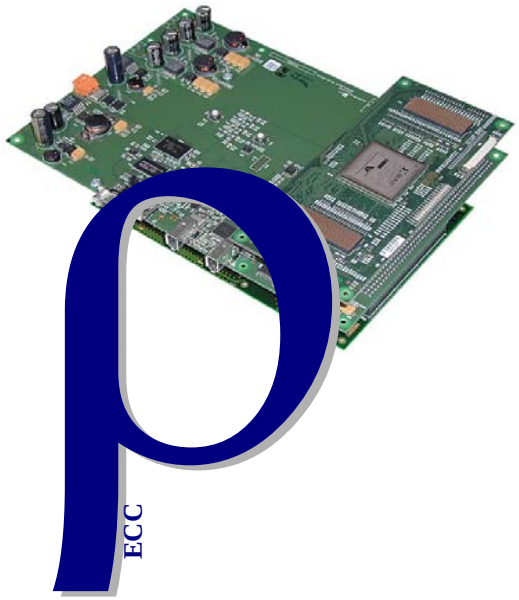
Mathematical Background

Implementation of Pollard‘s-Rho in Hardware

State of the Art in ECC

Results and Projections

Conclusions



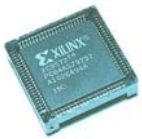
Attack Architecture Assumptions

Horst-Görtz Institut für IT Sicherheit

We assume the following three basic platform types

Software based architecture (Pentium M@1.7GHz)

– Costs: including housing ≈ US$ 400



FPGA based architecture (Xilinx XC3S1000; 10^6 Gates)

– Costs: based on COPACOBANA ≈ US$10,000 per 120 FPGAs

Estimated ASIC based architecture (10×10^6 transistors @ 500MHz)

– Costs: including overhead ≈ US$50 (excluding NRE)

Example: for US$100,000 we get

- **250 Pentiums, or**
- **1200 FPGAs, or**
- **2000 ASICs**

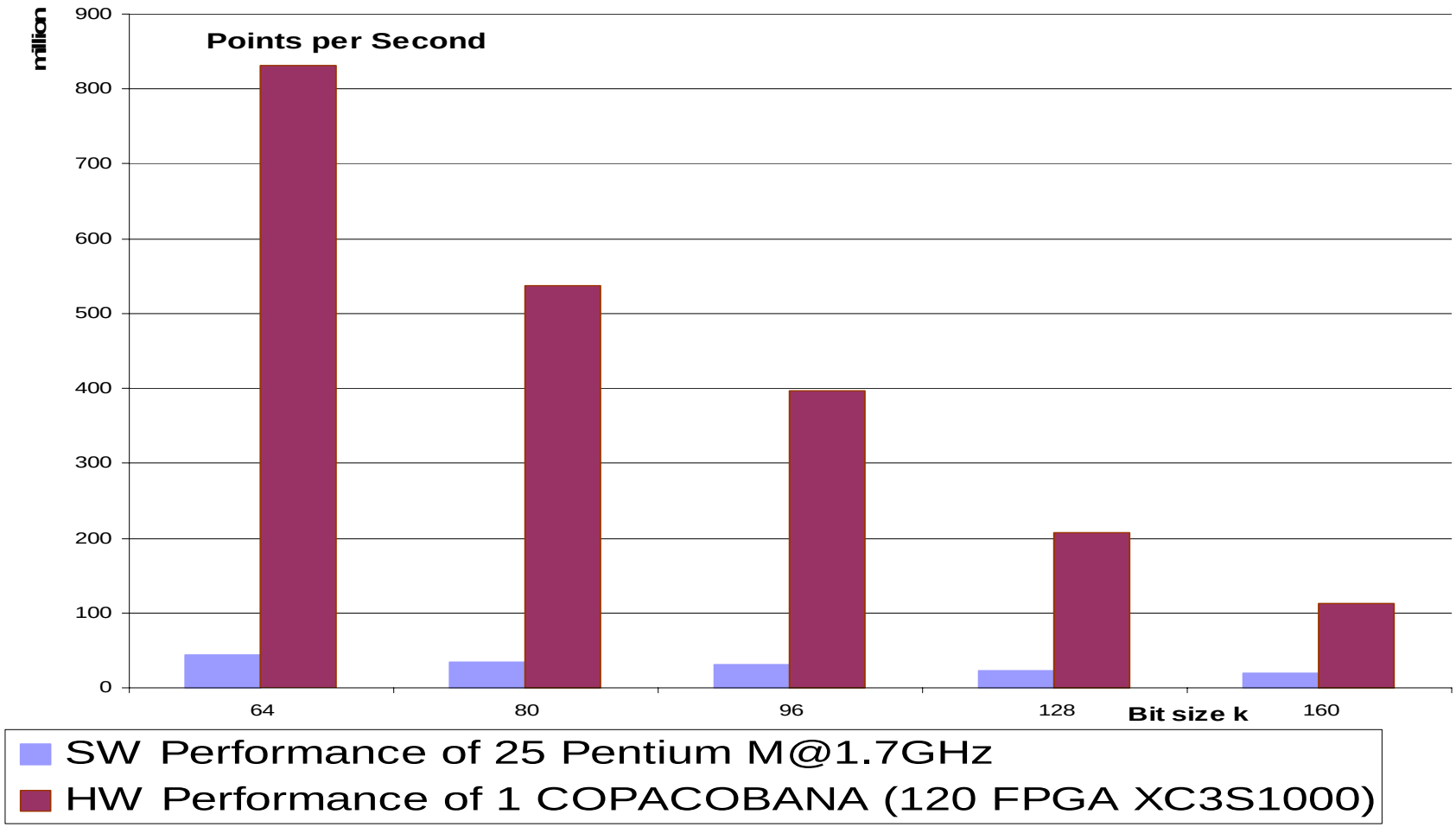
Results: Pollard-Rho Architecture (FPGA)

Horst-Görtz Institut für IT Sicherheit

Bit size k	# Cores	Device Usage	Max Freq.	Time per Operation	Pts/sec per Core	Pts/sec per FPGA
160	2	83 %	40.0 MHz	21.4 µs	46,800	93,600
128	3	98 %	40.1 MHz	17.3 µs	57,800	173,000
96	4	98 %	44.3 MHz	12.1 µs	82,700	331,000
80	4	88 %	50.9 MHz	8.94 µs	111,900	447,000
64	5	88 %	52.0 MHz	7.21 µs	138,600	693,000

Synthesis Results on SPARTAN-3 XC3S1000 FPGA

Results: SW and HW for US$10.000



Results: Comparing Architectures for US$ 1 million

Horst-Görtz Institut für IT Sicherheit

Bit size k	SW Reference Pentium M@1.7	Implementation XC3S1000 FPGA	Estimated ASIC Performance
80	40.6 h	2.58 h	-
96	8.04 d	14.8 h	-
SEC-1* (112 bit)	6.48 y	262 d	1.29 d
128	1.94×10^3 y	213 y	1.03 y
160	1.51×10^8 y	2.58×10^7 y	1.24×10^5 y

*by SECG (STANDARDS FOR EFFICIENT CRYPTOGRAPHY)

Results: Attack Estimates for Your Budget

Horst-Görtz Institut für IT Sicherheit

Expected runtimes in years for attacks based on ASICs with *k* bits funded by:

k	**US$ 10^5**	**US$ 10^6**	**US$ 10^7**	**US$ 10^8**
128	1.03×10^1 y	1.03 y	0.103 y	0.0103 y
160	1.24×10^6 y	1.24×10^5 y	1.24×10^4 y	1.24×10^3 y
192	9.64×10^{10} y	9.64×10^9 y	9.64×10^8 y	9.64×10^7 y
256	1.09×10^{21} y	1.09×10^{20} y	1.09×10^{19} y	1.09×10^{18} y

Agenda

Horst-Görtz Institut für IT Sicherheit

Introduction and Motivation

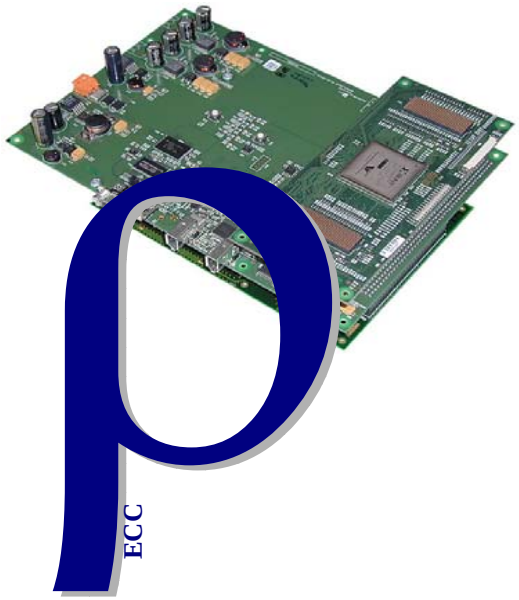
Mathematical Background

Implementation of Pollard‘s-Rho in Hardware

State of the Art in ECC

Results and Projections

Conclusions



Conclusions

Horst-Görtz Institut für IT Sicherheit

- First implementation of complete Pollard Rho attack for ECC over GF(p)
- ECC seems very secure with current attacks and technology,
  e.g., ASIC attack @ US$ 5million for ECC-131 within one year.
- ECC-163 attack within one year: US$ 5.8 $x10^{11}$.
  - According to Moore‘s Law it will take about 20 years to perform the same attack for US$ 1million
- SEC-1 standard by SECG with 112 bits is insecure!
- **If** estimates for breaking RSA1024 in one year with special machines are correct (depending on machine: $20m…$200m)

  $\Rightarrow$ ECC-163 would be 2.900 times more expensive!