Breaking Ciphers with COPACOBANA
A Cost-Optimized Parallel Code Breaker

or

How to Break DES for 8,980 €

CHES 2006, Yokohama, October 10-13, 2006

Sandeep Kumar, Jan Pelzl, Gerd Pfeiffer,
Manfred Schimmler, Christof Paar

http://www.copacobana.org
Acknowledgements

• Joint project with the University of Kiel (Gerd Pfeiffer, Manfred Schimmler)

• Special thanks to François-Xavier Standaert and Jean-Jacques Quisquater (Université Catholique de Louvain) for the core of the DES architecture
Introduction: A Naming Tale

What does COPACOBANA stand for?

Possible abbr. of „Cost-optimized Parallel Code-Breaker“:

- CPCB?
- COPCOB?
- COPCOBRA?
- COOPACOB?
- COPACOBRA?

► COPACOBANA
What’s in a name?

Copachelana

Copacabana
Outline

• Security vs. Cost

• COPACOBANA Design

• Application 1: Brute Force Attack on DES

• Application 2: ECC Attack

• Conclusion and Outlook
Outline

• Security vs. Cost

• COPACOBANA Design

• Application 1: Brute Force Attack on DES

• Application 2: ECC Attack

• Conclusion and Outlook
When is a Cipher Secure?

Symmetric ciphers

• (hopefully) only brute-force attack possible
• „secure“ key lengths: 112…256 bit (attack compl. $2^{112}…2^{256}$)
• but in practice wide variety of keys: AES, DES, RC4, A5, MD5, SHA-1, …
  (attack compl. $2^{56}…2^{256}$)

Asymmetric ciphers (RSA, ECC, DL)

• algorithmic attacks (e.g., factorization) dictate larger keys
• key lengths in practice:
  • RSA, DL: 1024 … 4096 bit
  • ECC: 160 … 256 bit
• attack complexities: $2^{80}$ (?) … $2^{128}$
Security and Computation

- Traditional: security of ciphers = **complexity** of attacks
- However: What really matters are the **costs** of an attack
- State-of-the-art
  - $< 2^{50}$ steps can be done with PC networks (more or less conveniently)
  - $> 2^{80}$ steps are very hard with today‘s technology (probably also for intelligence agencies)

**Major question:** Cost of attack for ciphers with 50…80 bit security (RSA1024, ECC160, SHA-1, DES, A5, …)
Introduction: Massive Computing

Supercomputers (Cray, SG, …)
- General (= complex & expensive) parallel computing architectures
- fast I/O, large memory, easy to program
  ► poor cost-performance ratio for (most) cryptanalysis

Distributed computing (conventional PCs)
- Dedicated clients in clusters, or
- Using PC’s idle time: E.g., SETI@home (BOINC framework)
  ► Problem of motivation, confidentiality issues

Special-purpose hardware
- ASIC - Application Specific Integrated Circuits (high NRE)
- FPGA - Field Programmable Gate Arrays (low NRE)
  ► best cost-performance ratio
Security of ciphers is related to complexity of attacks:

• **Symmetric ciphers:**
  – „Good“ ciphers: only exhaustive key search possible
  – an exhaustive key search should be infeasible
  – **Secure key lengths:** 80…256 bit
  – But many legacy systems with 56…64 bit (DES and such)

• **Asymmetric ciphers (e.g., RSA, ECC)**
  – longer keys due to analytical attacks
  – **Secure key lengths**
    • RSA: 1024…4096
    • ECC: 130-256 bit
Possible solutions to computational extensive problems:

- **Large supercomputers:**
  - Complex and expensive parallel computing architectures
  - Fast I/O, large memory, easy to program
  - E.g., Cray-XD1
  - Too complex for (most) cryptanalysis (bad cost-performance ratio)

- **Distributed computing (conventional PCs):**
  - Dedicated clients in clusters, or
  - Using PC’s idle time: E.g., SETI@home (BOINC framework)
  - Problem of motivating for cryptanalytic challenges, confidentiality issues

- **Special purpose hardware:**
  - Application Specific Integrated Circuits (ASICs, high NRE)
  - Field Programmable Gate Arrays (FPGAs, low NRE)
  - Optimized for one particular objective
  - Tradeoff between reprogrammability and price per piece, best cost-performance ratio
Introduction: Advantage of Hardware

Cost-performance ratio of DES\(^1\): PC vs. FPGA

- DES encryptions / decryptions per second

  - **Pentium4@3GHz:** \(\approx 2 \times 10^6\)
  - price per device (retail): € 80

  - **Xilinx XC3S1000@100MHz:** \(\approx 400 \times 10^6\)
  - price per device (retail): € 40

  ► Cost-performance ratio differs by 2-3 orders of magnitude!

1) Based on actual optimized implementations
COPACOBANA: Design Principles

• Ability to perform $\geq 2^{56}$ crypto operations
• Re-programmable: Applicable to many ciphers
• Strictly optimized cost-performance ratio:
  – off-the-shelf hardware (low-cost)
  – many logic resources (performance)
• < 9,000 € (including fabrication and material cost)
• Parallel architecture, based on 120 low-cost FPGAs
• Sacrifices
  – no global memory
  – no high-speed communication („only“ Mbit/s)
COPACOBANA: Realization

Scales easily:

- 20 FPGA modules/machine (120 FPGAs/machine)
- multiple machines via USB
COPACOBANA: Basic Design

- Modular design:
  1. Backplane
  2. FPGA modules (each with 6 low-cost FPGAs)
  3. Controller card with USB interface

- Easily extendable:
  - Up to 20 FPGA modules with 6 FPGAs each
  - Connect multiple COPACOBANAs via USB
COPACOBANA: FPGA Modules

Functionality:

• 6x Spartan-3 FPGAs (xc3s1000) per module
  – BGA packaging (FT256)
  – Internal clock rate up to 300 MHz
• Addressing:
  – HW decoded address of FPGA modules (GAL on backplane)
  – HW decoded address of single FPGA
  – Further addresses (5-bit) for FPGA-internal processing
• 64-bit data connection to backplane (bi-directional)
• 64-bit local bus (per module)
• Host cryptanalytical applications, e.g.,
  – Key search engines for DES
  – ECM engines
  – Pollard Rho engines
COPACOBANA: FPGA Modules
(Schematic)

6x Spartan 3 FPGA (xc3s1000, FT256 packaging)

Connection to backplane (64-bit data bus)
COPACOBANA: Alpha Prototype
COPACOBANA: Controller Module

Functionality:

• Programming of FPGAs:
  – Individual (download per FPGA)
  – Concurrent (download to all/ subset FPGAs)

• Communication with FPGAs:
  – Initialization of FPGA logic
  – Polling of FPGAs

• Communication with host-PC:
  – Redirecting results
  – Simple pre- and post processing
COPACOBANA: Applications

First flexible cryptanalytical machine outside government agencies

1. Exhaustive key search of DES
   - ciphers with $2^{56} \ldots 2^{64}$ attack steps possible

2. Real-world systems such as ePass, Norton Diskreet, ...

3. Elliptic Curve Discrete Logarithm Problem (ECDLP)
   - Parallelized Pollard‘s Rho

4. Factorization
   - Parallelized Elliptic Curve Method (ECM) as subroutine for GNFS (see GMU‘s talk later)

Attacks feasible

Robust security estimations

Improves other attacks
Outline

• Security vs. Cost

• COPACOBANA Design

• Application 1: Brute Force Attack on DES

• Application 2: ECC Attack

• Conclusion and Outlook
Data Encryption Standard (DES):
• Block cipher with 56-bit key
• Expired standard, but still used (legacy products, ePass, Norton Diskreet, …)

Exhaustive key search (conventional technology):
• Check $2^{55}$ keys on average
• PC (e.g., Pentium4@3GHz) ≈ 2 mio. keys/sec
• Average key search with one PC ≈ $2^{34}$ sec = 545 years!

► Can do much better with special-purpose hardware!
Attacks on DES

FPGA-based attacks on the Data Encryption Standard (DES):

- Exhaustive key search (FPGA based):
  - 4 completely pipelined DES engines per FPGA (courtesy of the crypto group of UCL)
  - one key per clock cycle per DES engine
  - One FPGA@100MHz: 400 mio. keys/ sec
Attacks on DES

- COPACOBANA: average key search of **8.7 days** @ 100 MHz
- Somewhat higher clock rates possible
- FPGA vs. PC (average key search in 8.7 days)
  - 22,865 Pentium 4 (€ 3.6 million incl. overhead)
    - or
  - COPACOBANA (total cost € 9000 incl. overhead)
- Alpha version of COPACABANA runs stable
- Life attack at http://www.copacobana.org/live
A Historical Perspective: The Power of Moore's Law

DeepCrack, 1998
$250,000

COPACOBANA, 2006
$10,000

Moore's Law: 50% cost reduction / 1.5 years
2006-1998 = 8 years ≈ 5 \times 1.5 \text{ years}
Prediction: $250,000 / 2^5 \approx $8,000 (close to actual $10,000)
Outline

• Security vs. Cost

• COPACOBANA Design

• Application 1: Brute Force Attack on DES

• Application 2: ECC Attack

• Conclusion and Outlook
ECDL Problem

• Many real-world applications rely on hardness of ECDLP
  • ECDSA,
  • ECDH,
  • ...

• Let P be a generator. Determine discrete logarithm \( \ell \) of a point Q such that

\[
Q = \ell P.
\]
Generic ECDLP Attacks

If parameters are chosen with care, only generic attacks are possible

   - Brute force attack is infeasible if \( \#E \geq 2^{80} \)

2. Shank’s Baby-Step-Giant-Step Method
   - Complexity in time AND memory of about \( \sqrt{\#E} \)

3. Pollard’s Rho method (\( \rho \))
   - Most efficient algorithm for general ECDLP
   - Complexity of \( \sqrt{\#E} \)

Note: All attacks are exponential in the bit length of the group order
Multi Processor Pollard Rho (MPPR)

Best known attack against general ECC

Proposed by van Oorschot/Wiener in 1999

Processors have individual search paths for “Distinguished Points” (DP)

DP are stored at central server

Duplicate DP = ECDLP solution

Ideal parallelization: speed up linear in number of employed processors

Colliding DP trails of multiple processors $w_i$
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.
ECDLP Attack Comparison: SW vs. HW for $10.000

Points per Second

SW Performance of 25 Pentium M@1.7GHz

HW Performance of 1 COPACOBANA (120 FPGA XC3S1000)
## ECDLP Attacks for US$ 1 million

<table>
<thead>
<tr>
<th>Bit size k</th>
<th>SW Reference Pentium M@1.7</th>
<th>COPACOBANA</th>
<th>est. ASIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>40.6 h</td>
<td>2.58 h</td>
<td>-</td>
</tr>
<tr>
<td>96</td>
<td>8.04 d</td>
<td>14.8 h</td>
<td>-</td>
</tr>
<tr>
<td>112 (SEC-1)*</td>
<td>6.48 y</td>
<td>262 d</td>
<td>1.29 d</td>
</tr>
<tr>
<td>128</td>
<td>1.94 x 10^3 y</td>
<td>213 y</td>
<td>1.03 y</td>
</tr>
<tr>
<td>160</td>
<td>1.51 x 10^8 y</td>
<td>2.58 x 10^7 y</td>
<td>1.24 x 10^5 y</td>
</tr>
</tbody>
</table>

* SECG (STANDARDS FOR EFFICIENT CRYPTOGRAPHY)
Conclusion

Pros and cons of COPACOBANA:

+ **efficient** hardware architecture
+ **reprogrammable** hardware (FPGAs)
+ very **cheap** to produce
+ **extendable** (per architecture, multiple architectures, …)
+ design option: **local memory**
+ design option: upgrade to future FPGA technologies
+ not restricted to code-breaking

- no global memory (only controller/ host-PC)
- relatively **slow communication**
- suited only for **particular problems** (e.g., cryptanalysis)
- requires programming in **VHDL**
Conclusion – COPACOBANA

• Results
  - DES in 8.6 days
  - ECCp163 attack currently $\approx$ $1$ trillion ($10^{12}$)
    - Moore’s Law: ECC 160 will stay secure for $\approx 20$ years
  - ECC112 (SEC-1 standard): insecure!
  - possibly real-time attack against ePass

• Many marginally weak ciphers are breakable

• „Strong“ ciphers (AES, RSA-1024, ECC-163, …) not breakable, but robust estimates by extrapolation of COPACOBANA results

• Several future applications are currently investigated

• Pictures, papers, and much more at www.copacobana.org

• We are looking for partners for other applications
Future work includes

• Completion of the COPACOBANA platform:
  – harden communication framework
  – run complete DES key search with 120 FPGAs
  – run (previous) ECC challenges on COPACOBANA, analyze SECG 80, 112, 128
  – implement parallel ECM for COPACOBANA
• Optimization of VHDL implementations
• Optimization of hardware platform (beyond prototype)
• Hardware based attacks demand for re-evaluation of security of, e.g., ECC
• Further applications: Smith-Waterman algorithm for scanning DNA sequences against databases