AN EFFICIENT MANY-CORE ARCHITECTURE FOR ELLIPTIC CURVE CRYPTOGRAPHY SECURITY ASSESSMENT

Marco Indaco, Fabio Lauri, Andrea Miele, Pascal Trotta
INTRODUCTION

• **Public-key cryptography (PKC)**: tools for key-exchange and digital signature, use of (public key, private key) pairs

• Elliptic curves (standard) public-key tool: use group of points and security based on discrete logarithm problem hardness in group

• Security: Pollard rho $O(\sqrt{|G|})$, using negation map $\approx \sqrt{\frac{\pi |G|}{4}}$

• But how long would it take to break it? Seconds, years, decades, aeons?
MOTIVATION

• Analyze performance of Pollard rho on new computing devices to assess what security level we can break in practice

• Only a few works on solving ECDLP on prime fields on FPGAs (no negation map)

• Main goal: cost of solving Certicom challenge ECCP-131 on cluster of FPGAs using Pollard rho with negation map
ELLiptic Curves

\[ y^2 = x^3 + a_1 x + a_0 \quad \#E(F_p) \approx p \]

Weierstrass coordinates: \((x, y)\)
\[ P = (x_1, y_1), \quad -P = (x_1, -y_1) \]

Affine addition: \(2m + l \cdot s + 6a + l\)
\[ P = (x_1, y_1), \quad Q = (x_2, y_2), \quad P + Q = (x_3, y_3) : \]
\[ x_3 = t^2 - x_1 - x_2, \quad y_3 = t(x_1 - x_3) - y_1 \]
\[ t = (y_1 - y_2)/(x_1 - x_2) \]
PARALLEL POLLARD RHO

- ECDLP in large prime order \(<P>\): given \(Q\) in \(<P>\) find integer \(k\) s. t. \(Q=kP\)

- Run \(m\) (speed-up: \(m\)) walks producing pseudo-random points till 2 collide
POLLAR RHO ITERATION

\[ F_0 = a_0 P + b_0 Q \]
\[ F_1 = a_1 P + b_1 Q \]
\[ \vdots \]
\[ F_j = a_j P + b_j Q \]
\[ \vdots \]
\[ F_{r-1} = a_{r-1} P + b_{r-1} Q \]

\[ l(P_i) = x_{P_i} \mod r = j \]
\[ (r=2^{r'}) \]

- Run \( m \) independent walks using the same table. Define distinguished points (easy to check property). Check for distinguished point collision.
USING THE NEGATION MAP

- Search for collision of equivalence classes of size 2 induced by the negation map: \( p \sim -p \), search for collision of \( \pm p \)

- Theoretical \( \sqrt{2} \) speed-up. Practice: little overhead + fruitless cycles
POLLARD RHO ON FPGAS

• Prime field of size $k$ (e.g., $k=131$)

• Addition/subtraction modulo prime: 1 clock cycle

• Montgomery multiplication: $k$ clock cycles

• Kaliski inversion: $2k$ clock cycles

• Negation map, cycle handling and analysis with large $r$ (e.g., $2^{14}$)
SINGLE-PIPE-MULTIPLE-WALK (SPMW) CORE

Walk <=> “HW” thread: pipeline elliptic curve addition

\[ \text{Walk}_{1,0} \rightarrow \text{ADD / SUB} \rightarrow \text{INV} \rightarrow \text{MUL} \rightarrow \text{ADD / SUB} \rightarrow \text{MUL} \rightarrow \text{ADD / SUB} \rightarrow \text{MUL} \rightarrow \text{ADD / SUB} \rightarrow \text{ADD / SUB} \rightarrow \text{MUL} \rightarrow \text{ADD / SUB} \rightarrow \text{ADD / SUB} \rightarrow \text{MUL} \rightarrow \text{ADD / SUB} \rightarrow \text{ADD / SUB} \rightarrow \text{MUL} \rightarrow \text{ADD / SUB} \rightarrow \text{ADD / SUB} \rightarrow \]
OPTIMIZATION: UNROLLING

- k-bit prime, MUL (k cycles), INV (2k cycles): $t_{\text{max}} = 2k$, throughput $\approx 1/t_{\text{max}}$
- Idea: split slowest stage into 2 stages (replicas) running for $t_{\text{max}}/2$ cycles
UNROLLING VS MORE CORES

- We want to maximize throughput under area constraints
- We can find the optimal values for \#Stages and \#Cores

For $k=131$ on a Xilinx Virtex 7-xc7v2000t

<table>
<thead>
<tr>
<th>#Cores</th>
<th>#Stages (#walks)</th>
<th>$t_{\text{max}}$</th>
<th>Freq</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>78</td>
<td>9</td>
<td>192 Mhz</td>
<td>26.9 W</td>
</tr>
</tbody>
</table>
SYSTEM OVERVIEW
RESULTS

Speed-up: 3.9x vs [GPP08] and 4.8x vs [JMS12]

Certicom ECCp-131 challenge

<table>
<thead>
<tr>
<th>FPGA</th>
<th>Unit price</th>
<th>Points/sec</th>
<th>Overall cost to solve in 1 year</th>
<th>#Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtex-5 vlx330t</td>
<td>8.4 K$</td>
<td>20.5 M</td>
<td>453 M$</td>
<td>44K</td>
</tr>
<tr>
<td>Virtex-6 vlx760</td>
<td>12.6 K$</td>
<td>67.3 M</td>
<td>207 M$</td>
<td>15K</td>
</tr>
<tr>
<td>Virtex-7 v2000t</td>
<td>17.4 K$</td>
<td>211.2 M</td>
<td>91 M$</td>
<td>4.6K</td>
</tr>
<tr>
<td>RIVYERA V7</td>
<td>500 K$</td>
<td>8448 M</td>
<td>65 M$</td>
<td>114</td>
</tr>
<tr>
<td>Virtex UltraScale 440</td>
<td>25 K$</td>
<td>738 M</td>
<td>34 M$</td>
<td>1.3K</td>
</tr>
</tbody>
</table>
CONCLUSION

- Solving Certicom ECCp-131 on FPGA cluster infeasible (near future?)

- Future work:
  1. Estimate performance of ASIC implementation
  2. Optimize arithmetic for k=131 and specific FPGA (e.g., low cost)
THANKS FOR YOUR ATTENTION!