Towards Reliable and Secure Post-Quantum Co-Processors based on RISC-V

Tim Fritzmann, Uzair Sharif, Daniel Müller-Gritschneder, Cezar Reinbrecht, Johanna Sepúlveda
Agenda

- Motivation
- Post-Quantum Cryptography and Performance Bottlenecks
- RISC-V Platform with Accelerators
- HW Trojan and Fault Attack Vulnerability
- Performance Evaluation
- Conclusion
Long-term Security

IoT
Today traditional cryptography is considered secure

Side-Channel

Attacker

Cryptoanalysis

Traditional Cryptography
(RSA, ECC)
Long-term Security

Attacker

Traditional Cryptography
(RSA, ECC)

Side-Channel

Cryptoanalysis

Traditional cryptography can be broken

T. Fritzmann TU Munich

IBM's five qubits Quantum Processor (www.ibm.com)
Long-term Security

Post-Quantum Cryptography (P-Q)

Traditional cryptography can be broken

Side-Channel

Attacker

Cryptoanalysis

IBM’s five qubits Quantum Processor (www.ibm.com)

T. Fritzmann TU Munich
RISC-V and Cryptography

- RISC-V is a simple, free and open ISA
- Allows development of open-source hardware
- Ideal environment for research
- Modern Cryptography is based on Kerckhoffs's principle:
  “A cryptosystem should be secure even if everything about the system, except the key, is public.”

T. Fritzmann TU Munich
Post-Quantum Cryptography

- Code
- Hash
- Multivariate
- Lattice
  - NTRU Problem
  - Learning with Errors Problem

T. Fritzmann TU Munich
Performance Bottlenecks:
- Generation of random polynomials

\[ a = a_0 + a_1 x + \ldots + a_{n-1} x^{n-1} \in \mathbb{R}_N, q = \mathbb{Z}_q[x] / (x^{N+1}) \]
Performance Bottlenecks:

- Polynomial Multiplication

\[ c = a \cdot b \mod (x^n + 1) \]

\[ c_i = \sum_{j=0}^{i} a_j b_{i-j} - \sum_{j=i+1}^{n-1} a_j b_{n+i-j} \]

\[ \mathcal{O}(n^2) \]

How can we improve this?
Accelerating Polynomial Multiplication:

– Use Number Theoretic Transform (NTT)

\[ \text{mod } x^n + 1 = NTT^{-1}(NTT(a) \circ NTT(b)) \]

where \( \circ \) denotes coefficient-wise multiplication.

\[ \mathcal{O}(n \log(n)) \]
NTT Transformation

- **Forward Transformation:**
  \[
  \hat{a}_i = \sum_{j=0}^{n-1} \gamma_n^j \cdot \omega_n^{ij} \cdot a_j
  \]

- **Backward Transformation:**
  \[
  a_i = n^{-1} \cdot \gamma_n^{-i} \sum_{j=0}^{n-1} \omega_n^{-ij} \cdot \hat{a}_j
  \]

**Algorithm 1: NTT transform**

- **Input:** Coefficients \(a_i\), pre-calculated values of \(\omega_m\)
- **Result:** Coefficients \(\hat{a}_i\)

1. \(a \leftarrow \text{BitReversal}(a)\)
2. for \(m = 2\) to \(n\) by \(m = 2m\) do
   3. \(\omega_m \leftarrow \omega_n^{n/m}, \quad \omega \leftarrow 1\)
   4. for \(j = 0\) to \(m/2 - 1\) by \(1\) do
      5. for \(k = 0\) to \(n - 1\) by \(m\) do
         6. \(t \leftarrow \omega \cdot a_{k+j+m/2}\)
         7. \(u \leftarrow a_{k+j}\)
         8. \(a_{k+j} \leftarrow u + t\)
         9. \(a_{k+j+m/2} \leftarrow u - t\)
      end
   end
   11. \(\omega \leftarrow \omega \cdot \omega_m\)
end

- Butterfly Operation
- Update Twiddle Factor

T. Fritzmann TU Munich
RISC-V Co-Processor Platform

RISC-V Core (RI5CY)

AHB Instruction Memory

AHB Inst.

AHB Data Memory

AHB Data Interconnect

UART

GPIO

Hash

NTT

T. Fritzmann TU Munich
## RISC-V Co-Processor Platform

<table>
<thead>
<tr>
<th>Address Range</th>
<th>Peripheral</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01000000 - 0x0100FFFF</td>
<td>Data Memory</td>
<td>Data memory</td>
</tr>
<tr>
<td>0x1A100000 - 0x1A10FFFF</td>
<td>GPIO</td>
<td>GPIO pins (connected to LEDs)</td>
</tr>
<tr>
<td>0x1B100000 - 0x1B10FFFF</td>
<td>TIMER</td>
<td>Cycle accurate timer</td>
</tr>
<tr>
<td>0x1C100000 - 0x1C10FFFF</td>
<td>UART</td>
<td>Interface to the HOST PC</td>
</tr>
<tr>
<td>0x1D100000 - 0x1D10FFFF</td>
<td>NTT</td>
<td>Hardware NTT accelerator</td>
</tr>
<tr>
<td>0x1E100000 - 0x1E10FFFF</td>
<td>Hash</td>
<td>Hardware hash accelerator</td>
</tr>
</tbody>
</table>
Hash Accelerator

T. Fritzmann TU Munich
NTT Accelerator

T. Fritzmann TU Munich
NTT Accelerator

T. Fritzmann TU Munich
Fault Attack and HW Trojan Vulnerability

- **Fault Attacks**
  - Intentionally manipulate a cryptographic system at the physical level to leak information about secret parameters
  - Optical attacks have a high localization/timing precision
  - Can even set or reset a single bit

- **HW Trojan**
  - Malicious modification of the HW
  - Outsourcing production increases risk
Fault Attack and HW Trojan Vulnerability

**SW Attack Model**

\[
a_i = n^{-1} \cdot \gamma_n^{-i} \sum_{j=0}^{n-1} \omega_{n}^{-ij} \cdot \hat{a}_j
\]

\[
\gamma = \gamma_n^{-1}
\]

- Forcing one of these constants temporarily to zero with using a laser
- E.g. attacking registers R19 and R29
- Failure propagates and sets all coefficients to zero

**HW Attack Model**

- [Diagram of HW Trojan SEL]

T. Fritzmann TU Munich
Why is this critical?

- Bob has a public polynomial $a$, private secret $s$, error $e$ and message $m$
- Bob calculates ciphertext: $c = a \cdot s + e + m$
- $a \cdot s = NTT^{-1}(NTT(a) \cdot NTT(s))$
- $NTT^{-1}$ is attacked $\rightarrow a \cdot s = 0$
- Bob transmits $c = a \times s + e + m \approx m$ to Alice over insecure channel
Performance Evaluation


T. Fritzmann TU Munich
Efficient and secure post-quantum cryptography to ensure long-term security is required

Hardware accelerators can be used to increase performance

Research on open platform helps to identify vulnerabilities

Be aware of side-channel attacks, fault attacks and HW Trojans
Thank you for your attention!