Introduction to Post-Quantum Cryptography
CERG: Cryptographic Engineering Research Group

3 faculty members, 9 Ph.D. students, 6 affiliated scholars
Previous Cryptographic Contests

- **AES**
  - IX.1997
  - X.2000
  - 15 block ciphers → 1 winner

- **NESSIE**
  - I.2000
  - XII.2002

- **CRYPTREC**
  - XI.2004

- **eSTREAM**
  - IV.2008
  - X.2007
  - 34 stream ciphers → + 4 SW winners

- **SHA-3**
  - XI.2004
  - IV.2008
  - 51 hash functions → 1 winner

- **CAESAR**
  - I.2013
  - II. 2019
  - 57 authenticated ciphers → multiple winners

- 57 authenticated ciphers

- 4 HW winners

- 34 stream ciphers

- 51 hash functions

- time

- 97 98 99 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17
Cryptographic Contests 2007-Present

51 hash functions → 1 winner
X.2007: SHA-3 → X.2012

57 authenticated ciphers → multiple winners
I.2013: CAESAR

69 Public-Key Post-Quantum Cryptography Schemes
II.2019: Post-Quantum

56 Lightweight authenticated ciphers & hash functions
XII.2016: Lightweight

Completed: VIII.2018
In progress: TBD, TBD
Why Cryptographic Contests?

- Avoid **back-door** theories
- Speed-up the **acceptance** of a new standard
- **Stimulate** non-classified research on methods of designing a specific cryptographic transformation
- **Focus** the effort of a relatively small cryptographic community
Evaluation Criteria

Security

Software Efficiency
- μProcessors
- μControllers

Hardware Efficiency
- FPGAs
- ASICs

Flexibility
Simplicity
Licensing
AES Contest, 1997-2000: Block Ciphers

GMU FPGA Results

Straw Poll @ NIST AES 3 conference

Rijndael second best in FPGAs, selected as a winner due to much better performance in software & ASICs
SHA-3 Round 2: 14 candidates

Throughput vs. Area: Normalized to Results for SHA-2 and Averaged over 7 FPGA Families
 SHA-3 Contest, 2007-2012: Hash Functions

**Keccak** – an early leader in hardware and winner of the competition
CAESAR, 2013-2018: Authenticated Ciphers

Relative (w.r.t. AES-GCM) Throughput in Virtex 6

14-16x better than the current standard

AEGIS and MORUS – finalists in the use case High-Performance Applications
Post-Quantum Cryptography (PQC)

• Public-key cryptographic algorithms for which there are no known attacks using quantum computers

• Capable of being implemented using any traditional methods, including software and hardware

• Running efficiently on any modern computing platforms: PCs, tablets, smartphones, servers with FPGA accelerators, etc.

• Term introduced by Dan Bernstein in 2003
• Equivalent terms: quantum-proof, quantum-safe or quantum-resistant
• Based entirely on traditional semiconductor VLSI technology!
Quantum Computers

- First perceived by physicists (Richard Feynman, David Deutsch) in 1980s
- First significant quantum algorithms (capable of running on quantum computers only) developed in 1990s
- First practical realization in 1998 (2 qubits)
- Significant technological advances during the last 20 years
- One of ten breakthrough technologies of 2017

Photo: https://www.technologyreview.com
Major advances during the last 20 years

Timeline of Quantum Computing:

Source: Vandersypen, PQCrypto 2017
Quantum Computers

- Substantial investments by: Google, IBM, Intel, Microsoft, Alcatel-Lucent, NTT
- Quantum computers based on superconducting circuits operating in the temperature close to absolute 0 (~0.01 K)

- November 2017: IBM’s 50-qubit chip
- January 2018: Intel’s 49-qubit chip, “Tangle-Lake”
- March 2018: Google’s 72-qubit chip “Bristlecone”

Photos: https://www.technologyreview.com
Progress in Quantum Computing

Photos: https://www.technologyreview.com
Remaining Challenges in Quantum Computing

1. High sensitivity to manufacturing variations
   Solution: Best industry cleanrooms, e.g., QuTech-Intel collaboration toward quantum-dot arrays made @ Intel 300mm wafers

2. Scalable control circuits (currently bulky & expensive)
   Solution: Tailored cryo-CMOS digital control

3. Multitude of interconnects and external pins
   Solution: Multiplexing electronics co-integrated with qubits

4. Non-standard architecture & limited programmability
   Solution: System layer approach
   Likely to be overcome in the next 10-15 years

Source: Vandersypen, PQCrypto 2017
System Layer Approach

Challenges in each layer

Layers are highly interrelated

Source: Vandersypen, PQCrypto 2017
What Quantum Computers Can Do?

Model complex molecules

Health: Quantum chemistry for medicine

Model complex materials

Energy: Room-temperature superconductivity

Solve complex math problems

Security: factoring and code breaking

Nobel 2012 citation: “The quantum computer may change our everyday lives in this century in the same radical way as the classical computer did in the last century.”

Source: Vandersypen, PQCrypto 2017
Quantum Computers & Cryptography

1994: Shor’s Algorithm, breaks major public key cryptosystems based on

Factoring: RSA

Discrete logarithm problem (DLP): DSA, Diffie-Hellman

Elliptic Curve DLP: Elliptic Curve Cryptosystems

independently of the key size assuming

a sufficiently powerful and reliable quantum computer available
Underlying Mathematical Problem - RSA

\[ N = P \times Q \text{ (P, Q random primes)} \]

\[
\begin{align*}
12301866845301177551304949583849627207728535695 \\
953347921973224521517264005072636575187452021997 \\
86469389956474942774063845925192557326303453731 \\
548268507917026122142913461670429214311602221240 \\
479274737794080665351419597459856902143413 \\
= \\
334780716989568987860441698482126908177047949837 \\
137685689124313889828837938780022876147116525317 \\
43087737814467999489 \\
* \\
36746043666799590428244633799627952632279158164 \\
343087642676032283815739666511279233373417143396 \\
810270092798736308917
\end{align*}
\]

Record Using Classical Computers, 232 digits, 768 bits
“There is a 1 in 7 chance that some fundamental public-key crypto will be broken by quantum by 2026, and a 1 in 2 chance of the same by 2031.”

Dr. Michele Mosca
Deputy Director of the Institute for Quantum Computing, University of Waterloo
April 2015
If \( z < y + x \), then worry!

\( y \) – Time to Develop & Deploy PQC Standards

\( z \) – Time to Build Quantum Computers

\( x \) – Time Information Must Remain Protected

Encrypted Data Stored by Powerful Adversaries

No Announcement when Quantum Computer Available to NSA, Foreign Governments, or Organized Crime
NIST PQC Standardization Process

- **Feb. 2016:** NIST announcement of standardization plans at PQCrypto 2016, Fukuoka, Japan,
- **Dec. 2016:** NIST Call for Proposals and Request for Nominations for Public-Key Post-Quantum Cryptographic Algorithms:
  - **Nov. 30, 2017:** Deadline for submitting candidates
- **Dec. 2017:** Announcement of the First Round Candidates
Three Types of PQC Schemes

1. Public Key Encryption
2. Digital Signature
3. Key Encapsulation Mechanism (KEM)
## Five Security Categories

<table>
<thead>
<tr>
<th>Level</th>
<th>Security Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>At least as hard to break as AES-128 using exhaustive key search</td>
</tr>
<tr>
<td>II</td>
<td>At least as hard to break as SHA-256 using collision search</td>
</tr>
<tr>
<td>III</td>
<td>At least as hard to break as AES-192 using exhaustive key search</td>
</tr>
<tr>
<td>IV</td>
<td>At least as hard to break as SHA-384 using collision search</td>
</tr>
<tr>
<td>V</td>
<td>At least as hard to break as AES-256 using exhaustive key search</td>
</tr>
</tbody>
</table>
## Leading PQC Families

<table>
<thead>
<tr>
<th>Family</th>
<th>Encryption/KEM</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hash-based</td>
<td>XX</td>
<td>XX</td>
</tr>
<tr>
<td>Code-based</td>
<td>XX</td>
<td>X</td>
</tr>
<tr>
<td>Lattice-based</td>
<td>XX</td>
<td>X</td>
</tr>
<tr>
<td>Multivariate</td>
<td>X</td>
<td>XX</td>
</tr>
<tr>
<td>Isogeny-based</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

XX – high-confidence candidates,  X – medium-confidence candidates
Underlying Mathematical Problem – PQC

Closest Vector Problem

Lattice in dimension $n=2$:
Set of points given by
\[ p = i \cdot b_1 + j \cdot b_2 \]
where $i$ and $j$ are arbitrary integers

Problem:
Find the point of lattice given by the base vectors $b_1$ and $b_2$ closest to the arbitrary point of an $n$-dimensional space $t$

Imagine $n$ in the range of 825
Solving a system of $m$ quadratic equations with $n$ unknowns

\[ p^{(1)}(x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=i}^{n} p^{(1)}_{ij} \cdot x_i x_j + \sum_{i=1}^{n} p^{(1)}_i \cdot x_i \left( + p^{(1)}_0 \right) \]

\[ p^{(2)}(x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=i}^{n} p^{(2)}_{ij} \cdot x_i x_j + \sum_{i=1}^{n} p^{(2)}_i \cdot x_i \left( + p^{(2)}_0 \right) \]

\[ \vdots \]

\[ p^{(m)}(x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=i}^{n} p^{(m)}_{ij} \cdot x_i x_j + \sum_{i=1}^{n} p^{(m)}_i \cdot x_i \left( + p^{(m)}_0 \right) \]

Imagine $m$ and $n$ in the range of 70 and above
### Round 1 Candidates

69 accepted as complete, 5 withdrawn
26 Countries, 260 co-authors

<table>
<thead>
<tr>
<th>Family</th>
<th>Signature</th>
<th>Encryption/KEM</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lattice-based</td>
<td>5</td>
<td>22</td>
<td>27</td>
</tr>
<tr>
<td>Code-based</td>
<td>2</td>
<td>16</td>
<td>18</td>
</tr>
<tr>
<td>Multivariate</td>
<td>7</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>Hash-based</td>
<td>2</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Isogeny-based</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Other</td>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>19</strong></td>
<td><strong>45</strong></td>
<td><strong>64</strong></td>
</tr>
</tbody>
</table>
Round 1 Submissions

69 Submissions, 26 Countries, 260 co-authors


Some attack scripts already posted causing total break or serious tweaks. Many more receiving detailed analysis.
NIST PQC Standardization Process

Jan. 30, 2019: Announcement of candidates qualified to Round 2

April 10, 2019: Publication of Round 2 submission packages

Aug. 22-24, 2019: Second NIST PQC Conference

2020: Beginning of Round 3 and/or selection of first future standards

2022-2024: Draft standards published
26 Candidates announced on January 30, 2019

<table>
<thead>
<tr>
<th>Family</th>
<th>Signature</th>
<th>Encryption/KEM</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lattice-based</td>
<td>3</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>Code-based</td>
<td></td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Multivariate</td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Symmetric-based</td>
<td>2</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Isogeny-based</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>9</td>
<td>17</td>
<td>26</td>
</tr>
</tbody>
</table>
Round 2 Submissions

- **Encryption/KEMs (17)**
  - CRYSTALS-KYBER
  - FrodoKEM
  - LAC
  - NewHope
  - NTRU (merger of NTRUEncrypt/NTRU-HRSS-KEM)
  - NTRU Prime
  - Round5 (merger of Hila5/Round2)
  - SABER
  - Three Bears

- **Digital Signatures (9)**
  - CRYSTALS-DILITHIUM
  - FALCON
  - qTESLA
  - Picnic
  - SPHINCS+

  - BIKE
  - Classic McEliece
  - HQC
  - LEDAcrypt (merger of LEDAkem/pkc)
  - NTS-KEM
  - ROLLO (merger of LAKE/LOCKER/Ouroboros-R)
  - RQC

  - SIKE

  - GeMSS
  - LUOV
  - MQDSS
  - Rainbow

  - Lattice-based
  - Code-based
  - Isogenies
  - Lattice-based
  - Symmetric-based
  - Multivariate

NIST Report on the 1st Round: [https://doi.org/10.6028/NIST.IR.8240](https://doi.org/10.6028/NIST.IR.8240)

Sources: Moody, PQCrypto May 2019
Similarities with Previous Contests

- Evaluation to be performed in **rounds** (12-18 months each)
- A **pool of candidates narrowed down** after each round
- Small **tweaks** allowed at the beginning of each round
- **Optimized software** implementation developed for various platforms
- No immediate plans for obligatory hardware implementations
Differences from Previous Contests

- Candidates not qualified to the next round (and not withdrawn by the authors) may be considered at a later date.
- Taking into account quantum attacks, possible only on the platforms that do not exist at the time of the standard development.
- Security analysis much more challenging and often controversial.
Adi Shamir’s Proposal

After the initial evaluation period (e.g., 3 years) the division of all schemes into the following categories:

2 Productions Schemes: recommended for actual wide-scale deployment, Highly Trusted

4 Development Schemes: Time-Tested, Trusted; at least 15 years of analysis behind them; Intended for initial R&D by industry.

8 Research Schemes: Promising Properties, Good Performance. May contain some high-risk candidates. Main Goal: Concentrate the effort of the research community.
PQCRYPTO Consortium

11 universities and companies
Funded by European Commission under the H2020 program

Initial Recommendations published in 2015
Co-authors of 22 Submissions
The Most Trusted Schemes – Encryption/KEM

Classical McEliece

- Proposed **40 years ago** as an alternative to RSA
- **Code-based** family
- Based on **binary Goppa codes**
- **No patents**
- **Conservative parameters (Category 5, 256-bit security):**
  - a) length n=6960, dimension k= 5413, errors=119
  - b) length n=8192, dimension k= 6528, errors=128
- **Complexity of the best attack identical after 40 years of analysis,** and more than 30 papers devoted to thorough cryptanalysis
- **Sizes:**
  - **Public key:**  a) 1,047,319 bytes, b) 1,357,824 bytes
  - **Private key:** a) 13,908 bytes, b) 14,080 bytes
  - **Ciphertext:** a) 226 bytes, b) 240 bytes
- **Efficient Software** (Haswell, larger parameter set)
  - ☆ 295,930 for encryption, 355,152 for decryption
- ☆ **Constant time**
- **Efficient Hardware (Yale)**
The Most Trusted Schemes – Signatures

Hash-based Schemes:

Security based on the security of a single underlying primitive hash function

Representatives:

SPHINCS-256 => SPHINCS+

Features:

Efficient signature generation and verification
Relatively large keys (~ tens of kilobytes)
Likely Development Schemes – Lattice-based

• Efficient encryption & decryption
• Relatively small key sizes (kilobytes)
• Suitable for constrained environments

• No proof of security
Likely Development Schemes – Lattice-based

• New lattice-based schemes with extended security proof, smaller key sizes, and better efficiency

Ring-LWE (Learning with Errors) Binary RLWE

NewHope CRYSTALS-KYBER CRYSTALS-DILITHIUM

Pilot Hardware Implementations:

• Ruhr University of Bochum, Germany
• Technical University Munich, Germany
• ESAT/COSIC KU Leuven, Belgium
• The University of Texas at Austin, USA
• George Mason University, USA, etc.
PQC Major Challenges

Fairness

Number of Candidates
Challenges of Post-Quantum Cryptography

- **Mathematical complexity**
- Large amount of **man-power**
- Large **keys and internal states**
- **Hardware resources** required
- New types of **basic operations**
- Need for **random sampling** not only from uniform but also from discrete Gaussian and/or other distributions
- **Constant-time** implementations
- Need for **new SCA (Side-Channel Attack) countermeasures** against power and electromagnetic analysis
- **Plug-and-play replacement** for current public-key cryptography units
- Intermediate use of **hybrid systems**
Hardware Benchmarking
Major Optimization Targets

- Parallel processing
- Constant-time
- Parametric code

Lightweight

- Small area, power, energy per bit
- Resistance to power & electromagnetic analysis
## Round 2 Candidates in Hardware

<table>
<thead>
<tr>
<th></th>
<th>#Round 2 candidates</th>
<th>Implemented in hardware</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>5</td>
<td>5</td>
<td>100%</td>
</tr>
<tr>
<td>SHA-3</td>
<td>14</td>
<td>14</td>
<td>100%</td>
</tr>
<tr>
<td>CAESAR</td>
<td>29</td>
<td>28</td>
<td>97%</td>
</tr>
<tr>
<td>PQC</td>
<td>26</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Software/Hardware Codesign
Software/Hardware Codesign

Software

Hardware

Most time-critical operation
SW/HW Codesign: Motivational Example 1

Software

- Major: 91%
- Other: 9%

Software/Hardware

- Major: ~1%
- Other: 9%

Time saved: 90%

91% major operation(s)
9% other operations

~1% major operation(s) in HW
9% other operations in SW

speed-up ≥ 100

Total Speed-Up ≥ 10
SW/HW Codesign: Motivational Example 2

- **Software**: 99% major operation(s), 1% other operations

- **Software/Hardware**: 1% major operation(s) in HW, 1% other operations in SW

**Speed-up**: $\geq 100$

**Time saved**: 98%

**Total Speed-Up**: $\geq 50$
SW/HW Codesign: Advantages

- Focus on a few major operations, known to be easily parallelizable
  - much shorter development time (at least by a factor of 10)
  - guaranteed substantial speed-up
  - high-flexibility to changes in other operations (such as candidate tweaks)
- Insight regarding performance of future instruction set extensions of modern microprocessors
- Possibility of implementing multiple candidates by the same research group, eliminating the influence of different
  - design skills
  - operation subset (e.g., including or excluding key generation)
  - interface & protocol
  - optimization target
  - platform
SW/HW Codesign: Potential Pitfalls

- Performance & ranking may strongly depend on
  
  A. features of a particular platform
     - Software/hardware interface
     - Support for cache coherency
     - Differences in max. clock frequency
  
  B. selected hardware/software partitioning
  
  C. optimization of an underlying software implementation

- Limited insight on ranking of purely hardware implementations

First step, not the ultimate solution!
Two Major Types of Platforms

FPGA Fabric & Hard-core Processors

Examples:
- Xilinx Zynq 7000 System on Chip (SoC)
- Xilinx Zynq UltraScale+ MPSoC
- Intel Arria 10 SoC FPGAs
- Intel Stratix 10 SoC FPGAs

FPGA Fabric, including Soft-core Processors

Examples:
- Xilinx Virtex UltraScale+ FPGAs
- Intel Stratix 10 FPGAs, including
  - Xilinx MicroBlaze
  - Intel Nios II
  - RISC-V, originally UC Berkeley
## Two Major Types of Platform

<table>
<thead>
<tr>
<th>Feature</th>
<th>FPGA Fabric and Hard-core Processor</th>
<th>FPGA Fabric with Soft-core Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>ARM</td>
<td>MicroBlaze, NIOS II, RISC-V, etc.</td>
</tr>
<tr>
<td>Clock frequency</td>
<td>&gt;1 GHz</td>
<td>max. 200-450 MHz</td>
</tr>
<tr>
<td>Portability</td>
<td>similar FPGA SoCs</td>
<td>various FPGAs, FPGA SoCs, and ASICs</td>
</tr>
<tr>
<td>Hardware accelerators</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Instruction set extensions</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Ease of design</td>
<td>Easy</td>
<td>Dependent on a particular soft-core processor and tool chain</td>
</tr>
<tr>
<td>(methodology, tools, OS support)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Xilinx Zynq UltraScale+ MPSoC**

1.2 GHz ARM Cortex-A53 + UltraScale+ FPGA logic
Experimental Setup

All elements located on a single chip
Code Release

• Full Code & Configuration of the Experimental Setup

• Software/Hardware Codesign of Round 1 NTRUEncrypt
to be made available at

https://cryptography.gmu.edu/athena

under PQC

by August 31, 2019
GMU CERG
Case Study
## SW/HW Codesign: Case Study

### 7 IND-CCA*-secure Lattice-Based Key Encapsulation Mechanisms (KEMs)
representing
5 NIST PQC Round 2 Submissions

<table>
<thead>
<tr>
<th>LWE (Learning with Error)-based:</th>
<th>NTRU-based:</th>
</tr>
</thead>
<tbody>
<tr>
<td>FrodoKEM</td>
<td>NTRU</td>
</tr>
<tr>
<td><strong>RLWR (Ring Learning with Rounding)-based:</strong></td>
<td>• NTRU-HPS</td>
</tr>
<tr>
<td>Round5</td>
<td>• NTRU-HRSS</td>
</tr>
<tr>
<td><strong>Module-LWR-based:</strong></td>
<td>NTRU Prime</td>
</tr>
<tr>
<td>Saber</td>
<td>• Streamlined NTRU Prime</td>
</tr>
<tr>
<td></td>
<td>• NTRU LPRime</td>
</tr>
</tbody>
</table>

* IND-CCA = with Indistinguishability under Chosen Ciphertext Attack
SW/HW Partitioning

Top candidates for offloading to hardware

**From profiling:**
- Large percentage of the execution time
- Small number of function calls

**From manual analysis of the code:**
- Small size of inputs and outputs
- Potential for combining with neighboring functions

**From knowledge of operations and concurrent computing:**
- High potential for parallelization
Example: LightSaber Decapsulation

- **MatrixVectorMul**: 43.44%
- **InnerProduct**: 43.52%
- **GenMatrix**: 5.03%
- **Hash**: 3.30%
- **GenSecret**: 2.30%
- **Other**: 2.40%
LightSaber Decapsulation

Execution time of functions to be moved to hardware 97.60%
Execution time of functions remaining in software 2.40%

Accelerator Speed-Up = 97.60/8.77=11.1
Total Speed-Up = 100/11.17=9.0
Tentative Results
Total Execution Time in Software [μs]

Encapsulation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
<th>Level 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round5</td>
<td>16,192</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saber</td>
<td>34,609</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Str NTRU Prime</td>
<td>62,076</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTRU LPrime</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTRU-HRSS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTRU-HPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FrodoKEM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63,505</td>
</tr>
</tbody>
</table>
Total Execution Time in Software/Hardware [\(\mu s\)]

Encapsulation

<table>
<thead>
<tr>
<th></th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
<th>Level 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round5</td>
<td>1</td>
<td>2</td>
<td>5→3</td>
<td>3→4</td>
<td>6→5</td>
</tr>
<tr>
<td>Saber</td>
<td>1,223</td>
<td>1,642</td>
<td>2,186</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>NTRU-HRSS</td>
<td>4→6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTRU Prime</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Str NTRU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTRU-HPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTRU-LPrime</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FrodoKEM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
</tr>
</tbody>
</table>

64
Total Speed-ups: Encapsulation

- NTRU-HRSS: Level 1 17.8
- FrodoKEM: Level 2 13.2
- Round5: Level 3 8.3
- NTRU-HPS: Level 4 10.2
- Saber: Level 5 10.6
- NTRU LPrime: Level 1 9.3
- Str NTRU Prime: Level 2 7.1
- 65
Accelerator Speed-ups: Encapsulation

- NTRU-HRSS
- NTRU-HPS
- FrodoKEM
- NTRU LPRime
- Str NTRU Prime
- Round5
- Saber

Levels:
- Level 1
- Level 2
- Level 3
- Level 4
- Level 5
SW Part Sped up by HW[%]: Encapsulation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
<th>Level 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round5</td>
<td>99.36</td>
<td>99.55</td>
<td>98.49</td>
<td>95.03</td>
<td>94.62</td>
</tr>
<tr>
<td>Saber</td>
<td>99.59</td>
<td>98.93</td>
<td>99.14</td>
<td>97.45</td>
<td>98.62</td>
</tr>
<tr>
<td>NTRU-HRSS</td>
<td>99.55</td>
<td>98.93</td>
<td>99.14</td>
<td>94.62</td>
<td>88.03</td>
</tr>
<tr>
<td>FrodoKEM</td>
<td>98.49</td>
<td>95.03</td>
<td>94.62</td>
<td>97.45</td>
<td>89.71</td>
</tr>
<tr>
<td>NTRU-HPS</td>
<td>98.93</td>
<td>95.03</td>
<td>94.62</td>
<td>97.45</td>
<td>70.35</td>
</tr>
<tr>
<td>NTRU LPrime</td>
<td>99.14</td>
<td>95.03</td>
<td>94.62</td>
<td>97.45</td>
<td>71.44</td>
</tr>
<tr>
<td>Str NTRU Prime</td>
<td>99.59</td>
<td>95.03</td>
<td>94.62</td>
<td>97.45</td>
<td>73.26</td>
</tr>
</tbody>
</table>
Total Execution Time in Software [$\mu$s]
Decapsulation

Order reversed compared to encapsulation

Level 1 Level 2 Level 3 Level 4 Level 5
Total Execution Time in Software/Hardware [μs]: Decapsulation

- **Round5**
- **Saber**
- **NTRU-HPS**
- **Str NTRU Prime**
- **NTRU-HRSS**
- **NTRU LPRime**
- **FrodoKEM**

<table>
<thead>
<tr>
<th>Level</th>
<th>Round5</th>
<th>Saber</th>
<th>NTRU-HPS</th>
<th>Str NTRU Prime</th>
<th>NTRU-HRSS</th>
<th>NTRU LPRime</th>
<th>FrodoKEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>5→3</td>
<td>4</td>
<td>6→5</td>
<td>3→6</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>1,319</td>
<td>1,866</td>
<td>3,120</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Level 1
- Level 2
- Level 3
- Level 4
- Level 5

69
Total Speed-ups: Decapsulation

- NTRU-HPS: 119.3
- NTRU-HRSS: 77.4
- Str NTRU Prime: 74.1
- FrodoKEM: 45.5
- Saber: 54.8
- Round5: 38.3
- NTRU LPRime: 12.3

Levels:
- Level 1
- Level 2
- Level 3
- Level 4
- Level 5
### Accelerator Speed-ups: Decapsulation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
<th>Level 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTRU-HRSS</td>
<td>188.1</td>
<td>182.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTRU-HPS</td>
<td>235.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Str NTRU Prime</td>
<td>112.0</td>
<td>86.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FrodoKEM</td>
<td>44.4</td>
<td>46.0</td>
<td>46.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTRU LPrime</td>
<td>27.5</td>
<td>32.8</td>
<td>37.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saber</td>
<td>11.1</td>
<td>17.7</td>
<td>24.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Round5</td>
<td>8.1</td>
<td>9.4</td>
<td>9.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Notes:*
- NTRU-HRSS and NTRU-HPS show significant speed-up gains.
- Str NTRU Prime has a notable improvement over other algorithms.
- FrodoKEM and NTRU LPrime exhibit consistent performance across different levels.
- Saber and Round5 show moderate speed-ups compared to the others.

*Level指示:*
- **Level 1:** Lowest performance level.
- **Level 2:** Intermediate performance level.
- **Level 3:** Highest performance level.
- **Level 4:** Specialized performance level.
- **Level 5:** Advanced performance level.
### SW Part Sped up by HW[%]: Decapsulation

<table>
<thead>
<tr>
<th></th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
<th>Level 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Round5</strong></td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td><strong>NTRU-HPS</strong></td>
<td>99.25</td>
<td>99.58</td>
<td>99.18</td>
<td>98.10</td>
<td>98.41</td>
</tr>
<tr>
<td><strong>NTRU-HRSS</strong></td>
<td>98.69</td>
<td>98.92</td>
<td>98.41</td>
<td>98.10</td>
<td>96.78</td>
</tr>
<tr>
<td><strong>Str NTRU Prime</strong></td>
<td>98.53</td>
<td>97.60</td>
<td>98.69</td>
<td>98.41</td>
<td>96.78</td>
</tr>
<tr>
<td><strong>Saber</strong></td>
<td>97.60</td>
<td>98.92</td>
<td>98.41</td>
<td>98.10</td>
<td>96.78</td>
</tr>
<tr>
<td><strong>FrodoKEM</strong></td>
<td>98.10</td>
<td>98.41</td>
<td>98.10</td>
<td>97.11</td>
<td>97.11</td>
</tr>
<tr>
<td><strong>NTRU LPRime</strong></td>
<td>93.96</td>
<td>96.78</td>
<td>97.11</td>
<td>77.46</td>
<td>76.47</td>
</tr>
</tbody>
</table>

- **Level 1**: 100.00% of the SW part sped up by HW[%]
- **Level 2**: 99.25% to 98.92% speedup
- **Level 3**: 98.53% to 97.60% speedup
- **Level 4**: 98.69% to 98.10% speedup
- **Level 5**: 97.60% to 96.78% speedup
Conclusions

- **Total speed-ups**
  - for encapsulation from 2.4 (Str NTRU Prime) to 28.4 (FrodoKEM)
  - for decapsulation from 3.9 (NTRU LPRime) to 119.3 (NTRU-HPS)

- Total speed-up **dependent on the percentage of the software execution time taken by functions offloaded to hardware and the amount of acceleration itself**

- **Hardware accelerators thoroughly optimized** using Register-Transfer Level design methodology

- Determining **optimal software/hardware partitioning requires more work**

- **Ranking of the investigated candidates affected, but not dramatically changed**, by hardware acceleration

- It is **possible to complete similar designs for all Round 2 candidates within the evaluation period (12-18 months)**

- **Additional benefit:** Comprehensive library of major operations in hardware
Future Work

Current work

- 4 Remaining Lattice-based KEMs
- 3 Lattice-based Digital Signatures
- 7 Code-based KEMs
- 7 Other Candidates

More operations moved to hardware / C code optimized for ARM Cortex-A53

Algorithmic optimizations of software and hardware

- Hardware library of basic operations of lattice-based candidates
- Hardware library of basic operations of code-based candidates
- Hardware library for PQC

Full hardware implementations

*collaboration with submission teams and other groups very welcome
High-Level Synthesis
High-Level Synthesis (HLS)

- High Level Language (C, C++, Java, Python, etc.)
- High-Level Synthesis (HLS)
- Hardware Description Language (VHDL or Verilog)
Popular HLS Tools

Commercial (FPGA-oriented):

• **Vivado HLS**: Xilinx – selected for this study
• **FPGA SDK for OpenCL**: Intel

Academic:

• **Bambu**: Politecnico di Milano, Italy
• **DWARV**: Delft University of Technology, The Netherlands
• **GAUT**: Universite de Bretagne-Sud, France
• **LegUp**: University of Toronto, Canada
Software/Hardware Codesign with HLS

Software

HLS-Generated Hardware

Most time-critical operation
1. Interface mapping
2. Addition of HLS Tool directives (pragmas)
3. Hardware-driven code refactoring
Using Existing Code

**Function:** Polynomial Multiplication in NTRUEncrypt

**Source of C code:** OnBoard Security

**Goal:** The same or comparable number of clock cycles as in the Register-Transfer Level (manual) implementation in VHDL.

**Attempt 1:** Reference implementation based on the grade school algorithm for multiplication (a.k.a. schoolbook, paper-and-pencil, etc.).

**Attempt 2:** Optimized implementation based on rotation. Multiple attempts at optimization using Vivado HLS directives (pragmas) and minor code changes.
Outcome & Revised Approach

**Outcome 1:** Tens of thousands of clock cycles, compared to the expected $n=743$ clock cycles

**Solution:** Rewriting the code in C in such a way to match the block diagram used to generate VHDL code

**Outcome 2:**
- Expected functionality
- *Around n clock cycles* of the execution time

Similar approach applied and successfully repeated for 4 NTRU-based Round 1 KEMs!
Sources of Productivity Gains

- Higher-level of abstraction
- Focus on datapath rather than control logic
- Debugging in software (C/C++)
  - Faster run time
  - No timing waveforms
## Achievements: Number of Clock Cycles

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RTL</th>
<th>HLS</th>
<th>HLS/RTL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Encapsulation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTRUEncrypt</td>
<td>744</td>
<td>750</td>
<td>1.008</td>
</tr>
<tr>
<td>NTRU-HRSS</td>
<td>702</td>
<td>708</td>
<td>1.009</td>
</tr>
<tr>
<td>Str NTRUE Prime</td>
<td>762</td>
<td>769</td>
<td>1.009</td>
</tr>
<tr>
<td>NTRU LPRime</td>
<td>1524</td>
<td>1538</td>
<td>1.009</td>
</tr>
<tr>
<td><strong>Decapsulation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTRUEncrypt</td>
<td>1491</td>
<td>1502</td>
<td>1.007</td>
</tr>
<tr>
<td>NTRU-HRSS</td>
<td>2111</td>
<td>2162</td>
<td>1.024</td>
</tr>
<tr>
<td>Str NTRUE Prime</td>
<td>2291</td>
<td>2346</td>
<td>1.024</td>
</tr>
<tr>
<td>NTRU LPRime</td>
<td>2287</td>
<td>2307</td>
<td>1.009</td>
</tr>
</tbody>
</table>

#Clock Cycles increased by less than 2.5%
## Overhead: Clock Frequency & Resource Utilization

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RTL</th>
<th>HLS</th>
<th>HLS/RTL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Clock Frequency [MHz]</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTRUEncrypt</td>
<td>330</td>
<td>270</td>
<td>0.82</td>
</tr>
<tr>
<td>NTRU-HRSS</td>
<td>300</td>
<td>261</td>
<td>0.87</td>
</tr>
<tr>
<td>Str NTRU Prime</td>
<td>255</td>
<td>191</td>
<td>0.75</td>
</tr>
<tr>
<td>NTRU LPRime</td>
<td>255</td>
<td>191</td>
<td>0.75</td>
</tr>
<tr>
<td><strong>Slices</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTRUEncrypt</td>
<td>4,431</td>
<td>7,361</td>
<td>1.66</td>
</tr>
<tr>
<td>NTRU-HRSS</td>
<td>5,476</td>
<td>7,358</td>
<td>1.34</td>
</tr>
<tr>
<td>Str NTRU Prime</td>
<td>9,699</td>
<td>18,428</td>
<td>1.90</td>
</tr>
<tr>
<td>NTRU LPRime</td>
<td>8,483</td>
<td>15,585</td>
<td>1.84</td>
</tr>
</tbody>
</table>

Clock Frequency reduced by 25% or less

#Slices increased by 90% or less
# Overhead: Resource Utilization

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RTL</th>
<th>HLS</th>
<th>HLS/RTL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LUTs</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTRUEncrypt</td>
<td>27,912</td>
<td>50,061</td>
<td>1.79</td>
</tr>
<tr>
<td>NTRU-HRSS</td>
<td>33,230</td>
<td>48,452</td>
<td>1.46</td>
</tr>
<tr>
<td>Str NTRU Prime</td>
<td>65,207</td>
<td>117,424</td>
<td>1.80</td>
</tr>
<tr>
<td>NTRU LPRime</td>
<td>52,297</td>
<td>94,062</td>
<td>1.80</td>
</tr>
<tr>
<td><strong>Flip-flops</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTRUEncrypt</td>
<td>24,697</td>
<td>24,728</td>
<td>1.00</td>
</tr>
<tr>
<td>NTRU-HRSS</td>
<td>30,327</td>
<td>33,235</td>
<td>1.10</td>
</tr>
<tr>
<td>Str NTRU Prime</td>
<td>32,929</td>
<td>51,253</td>
<td>1.56</td>
</tr>
<tr>
<td>NTRU LPRime</td>
<td>39,730</td>
<td>40,089</td>
<td>1.00</td>
</tr>
</tbody>
</table>

#LUTs increased by 80% or less
#Flip-flops increased by 56% or less
Possible Future Work

HLS-Generated Hardware
Minimal Hand Optimization

HLS-Generated Hardware
Thorough Hand Optimization

Most time-critical operation
PQC Opportunities & Challenges

- The biggest revolution in cryptography, since the invention of public-key cryptography in 1970s
- Efficient hardware implementations in FPGAs and ASICs desperately needed to prove the candidates suitability for high-performance applications and constrained environments. Collaboration sought by submission teams!
- Likely extensions to Instruction Set Architectures of multiple major microprocessors
- Start-up & new-product opportunities
- Once in the lifetime opportunity! Get involved!
Q&A

Thank You!

Questions?   Comments?

Suggestions?

CERG: http://cryptography.gmu.edu
ATHENa: http://cryptography.gmu.edu/athena