Post-quantum key exchange for the Internet

Douglas Stebila  McMaster University
Acknowledgements

Collaborators
- Joppe Bos
- Craig Costello and Michael Naehrig
- Léo Ducas
- Ilya Mironov and Ananth Raghunathan
- Michele Mosca
- Valeria Nikolaenko

Support
- Australian Research Council (ARC)
- Natural Sciences and Engineering Research Council of Canada (NSERC)
- Queensland University of Technology
- Tutte Institute for Mathematics and Computing
Motivation
Contemporary cryptography

TLS-ECDHE-RSA-AES128-GCM-SHA256

Public-key cryptography
- RSA signatures
- Elliptic curve Diffie–Hellman key exchange
  - difficulty of factoring
  - difficulty of elliptic curve discrete logarithms

Symmetric cryptography
- AES
- SHA-2

Can be solved efficiently by a large-scale quantum computer
Building quantum computers

Building quantum computers

When will a large-scale quantum computer be built?

“I estimate a 1/7 chance of breaking RSA-2048 by 2026 and a 1/2 chance by 2031.”

— Michele Mosca, November 2015
https://eprint.iacr.org/2015/1075
Post-quantum cryptography in academia

Conference series
- PQCrypto 2006
- PQCrypto 2008
- PQCrypto 2010
- PQCrypto 2011
- PQCrypto 2013
- PQCrypto 2014
- PQCrypto 2016
Post-quantum cryptography in government

“IAAD will initiate a transition to quantum resistant algorithms in the not too distant future.”

– NSA Information Assurance Directorate, Aug. 2015
NIST Post-quantum Crypto Project timeline

<table>
<thead>
<tr>
<th>Date</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 16, 2016</td>
<td>Feedback on call for proposals</td>
</tr>
<tr>
<td>Fall 2016</td>
<td>Formal call for proposals</td>
</tr>
<tr>
<td>November 2017</td>
<td>Deadline for submissions</td>
</tr>
<tr>
<td>Early 2018</td>
<td>Workshop – submitters’ presentations</td>
</tr>
<tr>
<td>3-5 years</td>
<td>Analysis phase</td>
</tr>
<tr>
<td>2 years later</td>
<td>Draft standards ready</td>
</tr>
</tbody>
</table>

http://www.nist.gov/pqcrypto
Post-quantum / quantum-safe crypto

No known exponential quantum speedup

Hash-based
- Merkle signatures
- Sphincs

Code-based
- McEliece

Multivariate
- multivariate quadratic

Lattice-based
- NTRU
- learning with errors
- ring-LWE

Isogenies
- supersingular elliptic curve isogenies
Lots of questions

- Design better post-quantum key exchange and signature schemes
- Improve classical and quantum attacks
- Pick parameter sizes
- Develop fast, secure implementations
- Integrate them into the existing infrastructure
This talk

• Two key exchange protocols from lattice-based problems
  • BCNS15: key exchange from the ring learning with errors problem
  • Frodo: key exchange from the learning with errors problem

• Open Quantum Safe project
  • A library for comparing post-quantum primitives
  • Framework for easing integration into applications like OpenSSL
Why key exchange?

Premise: large-scale quantum computers don’t exist right now, but we want to protect today’s communications against tomorrow’s adversary.

- Signatures still done with traditional primitives (RSA/ECDSA)
  - we only need authentication to be secure now
  - benefit: use existing RSA-based PKI
- Key agreement done with ring-LWE, LWE, …
  - Also consider “hybrid” ciphersuites that use post-quantum and traditional elliptic curve
Learning with errors problems
Solving systems of linear equations

Linear system problem: given blue, find red
Solving systems of linear equations

Linear system problem: given blue, find red

\[
\begin{array}{cccc}
4 & 1 & 11 & 10 \\
5 & 5 & 9 & 5 \\
3 & 9 & 0 & 10 \\
1 & 3 & 3 & 2 \\
12 & 7 & 3 & 4 \\
6 & 5 & 11 & 4 \\
3 & 3 & 5 & 0 \\
\end{array}
\times
\begin{array}{c}
6 \\
9 \\
11 \\
11 \\
\end{array}
=
\begin{array}{c}
4 \\
8 \\
1 \\
10 \\
4 \\
12 \\
9 \\
\end{array}
\]

Easily solved using Gaussian elimination (Linear Algebra 101)
Learning with errors problem

<table>
<thead>
<tr>
<th>random $\mathbb{Z}_7^{4 \times 13}$</th>
<th>secret $\mathbb{Z}_4^{1 \times 13}$</th>
<th>small noise $\mathbb{Z}_7^{1 \times 13}$</th>
<th>$\mathbb{Z}_1^{13}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 1 11 10</td>
<td>6</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>5 5 9 5</td>
<td>9</td>
<td>-1</td>
<td>7</td>
</tr>
<tr>
<td>3 9 0 10</td>
<td>11</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>1 3 3 2</td>
<td></td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>12 7 3 4</td>
<td></td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>6 5 11 4</td>
<td></td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>3 3 5 0</td>
<td></td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>

$\times$ $+$ $=$
Learning with errors problem

### Computational LWE problem: given blue, find red

<table>
<thead>
<tr>
<th>Random $\mathbb{Z}_{13}^{7 \times 4}$</th>
<th>Secret $\mathbb{Z}_{13}^{4 \times 1}$</th>
<th>Small Noise $\mathbb{Z}_{13}^{7 \times 1}$</th>
<th>Result $\mathbb{Z}_{13}^{7 \times 1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 1 11 10</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5 5 9 5</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>3 9 0 10</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>1 3 3 2</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>12 7 3 4</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6 5 11 4</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>3 3 5 0</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>
**Decision learning with errors problem**

<table>
<thead>
<tr>
<th>random $\mathbb{Z}_{13}^{7 \times 4}$</th>
<th>secret $\mathbb{Z}_{13}^{4 \times 1}$</th>
<th>small noise $\mathbb{Z}_{13}^{7 \times 1}$</th>
<th>looks random $\mathbb{Z}_{13}^{7 \times 1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 1 11 10</td>
<td>5 5 9 5</td>
<td>3 9 0 10</td>
<td>1 3 3 2</td>
</tr>
<tr>
<td>5 5 9 5</td>
<td>12 7 3 4</td>
<td>6 5 11 4</td>
<td>3 3 5 0</td>
</tr>
<tr>
<td>3 9 0 10</td>
<td>x</td>
<td>+</td>
<td>=</td>
</tr>
<tr>
<td>1 3 3 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 7 3 4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Decision LWE problem:** given blue, distinguish green from random
Toy example versus real-world example

\[ \mathbb{Z}^{7 \times 4}_{13} \]

\[
\begin{array}{cccc}
4 & 1 & 11 & 10 \\
5 & 5 & 9 & 5 \\
3 & 9 & 0 & 10 \\
1 & 3 & 3 & 2 \\
12 & 7 & 3 & 4 \\
6 & 5 & 11 & 4 \\
3 & 3 & 5 & 0 \\
\end{array}
\]

\[ \mathbb{Z}^{752 \times 8}_{2^{15}} \]

\[
\begin{array}{cccc}
2738 & 3842 & 3345 & 2979 \\
2896 & 595 & 3607 \\
377 & 1575 \\
2760 \\
\end{array}
\]

\[ 752 \times 28 \times 15 \text{ bits} = 11 \text{ KiB} \]
Ring learning with errors problem

**random**
\[ \mathbb{Z}_{13}^{7 \times 4} \]

<table>
<thead>
<tr>
<th>4</th>
<th>1</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

Each row is the cyclic shift of the row above
Ring learning with errors problem

Each row is the cyclic shift of the row above

... with a special wrapping rule: $x$ wraps to $-x \mod 13$. 

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
<td>11</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>1</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>12</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>12</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>9</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

$\mathbb{Z}^{7 \times 4}_{13}$
Ring learning with errors problem

Random $\mathbb{Z}_{13}^{7 \times 4}$

| 4 | 1 | 11 | 10 |

Each row is the cyclic shift of the row above

... with a special wrapping rule:

$x$ wraps to $-x \mod 13$.

So I only need to tell you the first row.
Ring learning with errors problem

\[ \mathbb{Z}_{13}[x]/\langle x^4 + 1 \rangle \]

\[ 4 + 1x + 11x^2 + 10x^3 \]

\[ \times \]

\[ 6 + 9x + 11x^2 + 11x^3 \]

\[ + \]

\[ 0 - 1x + 1x^2 + 1x^3 \]

\[ = 10 + 5x + 10x^2 + 7x^3 \]

random

secret

small noise
Ring learning with errors problem

\[
\mathbb{Z}_{13}[x]/\langle x^4 + 1 \rangle
\]

random

\[
4 + 1x + 11x^2 + 10x^3
\]

secret

+ 

\[
10 + 5x + 10x^2 + 7x^3
\]

small noise

Computational ring-LWE problem: given blue, find red
Decision ring learning with errors problem

\[ \mathbb{Z}_{13}[x]/\langle x^4 + 1 \rangle \]

\[ 4 + 1x + 11x^2 + 10x^3 \]

\[ \times \]

\[ \text{random} \]

\[ \text{secret} \]

\[ + \]

\[ \text{small noise} \]

\[ = 10 + 5x + 10x^2 + 7x^3 \]

Decision ring-LWE problem: given blue, distinguish green from random
Decision ring learning with errors problem with small secrets

$$\mathbb{Z}_{13}[x]/\langle x^4 + 1 \rangle$$

random

$$4 + 1x + 11x^2 + 10x^3$$

small secret

$$1 + 0x - 1x^2 + 2x^3$$

small noise

$$= 10 + 5x + 10x^2 + 7x^3$$

looks random

Decision ring-LWE problem: given blue, distinguish green from random
Problems

Computational LWE problem

Decision LWE problem

with or without short secrets

Computational ring-LWE problem

Decision ring-LWE problem


Key agreement from ring-LWE

Bos, Costello, Naehrig, Stebila.
Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. 

https://www.douglas.stebila.ca/research/papers/SP-BCNS15/
Definition. Let $n$ be a power of 2, $q$ be a prime, and $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ be the ring of polynomials in $X$ with integer coefficients modulo $q$ and polynomial reduction modulo $X^n + 1$. Let $\chi$ be a distribution over $R_q$.

Let $s \leftarrow \chi$.

Define:

- $O_{\chi,s}$: Sample $a \leftarrow \mathcal{U}(R_q)$, $e \leftarrow \chi$; return $(a, as + e)$.

- $U$: Sample $(a, b') \leftarrow \mathcal{U}(R_q \times R_q)$; return $(a, b')$.

The decision R-LWE problem with short secrets for $n, q, \chi$ is to distinguish $O_{\chi,s}$ from $U$. 
Hardness of decision ring-LWE

Practice:
- Assume the best way to solve DRLWE is to solve LWE.
- Assume solving LWE involves a lattice reduction problem.
- Estimate parameters based on runtime of lattice reduction algorithms e.g. [APS15]
- (Ignore non-tightness.) [CKMS16]

worst-case approximate shortest (independent) vector problem (SVP/SIVP) on ideal lattices in $\mathbb{R}$

poly-time [LPR10]

search ring-LWE

decision ring-LWE

tight [ACPS09]

decision ring-LWE with short secrets

poly-time [LPR10]
Basic ring-LWE-DH key agreement (unauthenticated)

- Reformulation of Peikert’s ring-LWE KEM (PQCrypto 2014)

Alice

secret:
random “small” \( s, e \) in \( R_q \)

Bob

secret:
random “small” \( s’, e’ \) in \( R_q \)

- \( a = a \cdot s + e \)
- \( b’ = a \cdot s’ + e’ \)

shared secret:

\( s \cdot b’ = s \cdot (a \cdot s’ + e’) \approx s \cdot a \cdot s’ \)

These are only approximately equal \( \Rightarrow \) need rounding
Rounding

- Each coefficient of the polynomial is an integer modulo $q$
- Treat each coefficient independently
Basic rounding

- Round either to 0 or $q/2$
- Treat $q/2$ as 1

This works most of the time: prob. failure $2^{-10}$.

Not good enough: we need exact key agreement.
Better rounding (Peikert)

Bob says which of two regions the value is in: \( \frac{q}{4} \) or \( \frac{3q}{4} \)

- If \( \frac{q}{4} \leq x < \frac{q}{2} \), round to \( 1 \)
- If \( \frac{q}{2} \leq x < \frac{3q}{4} \), round to \( 0 \)
• If $|alice - bob| \leq q/8$, then this always works.

• For our parameters, probability $|alice - bob| > q/8$ is less than $2^{-128000}$.

• Security not affected: revealing $\bullet$ or $\bullet$ leaks no information
Exact ring-LWE-DH key agreement (unauthenticated)

- Reformulation of Peikert’s R-LWE KEM (PQCrypto 2014)

public: “big” a in $R_q = \mathbb{Z}_q[x]/(x^n+1)$

Alice

- secret: random “small” $s, e$ in $R_q$

Bob

- secret: random “small” $s', e'$ in $R_q$

$b = a \cdot s + e$

$b' = a \cdot s' + e'$, or $

shared secret: round(s \cdot b')$

shared secret: round($b' \cdot s'$)
# Ring-LWE-DH key agreement

<table>
<thead>
<tr>
<th>Public parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision R-LWE parameters $q, n, \chi$</td>
</tr>
<tr>
<td>$a \leftarrow \mathcal{U}(R_q)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s, e \leftarrow \chi$</td>
<td></td>
</tr>
<tr>
<td>$b \leftarrow as + e \in R_q$</td>
<td></td>
</tr>
<tr>
<td>$s', e' \leftarrow \chi$</td>
<td></td>
</tr>
<tr>
<td>$b' \leftarrow as' + e' \in R_q$</td>
<td></td>
</tr>
<tr>
<td>$e'' \leftarrow \chi$</td>
<td></td>
</tr>
<tr>
<td>$v \leftarrow bs' + e'' \in R_q$</td>
<td></td>
</tr>
<tr>
<td>$\overline{v} \leftarrow \text{dbl}(v) \in R_{2q}$</td>
<td></td>
</tr>
<tr>
<td>$b', c \leftarrow \text{rec}(2b', s, \overline{v}) \in {0, 1}^n$</td>
<td></td>
</tr>
<tr>
<td>$c \leftarrow \langle \overline{v} \rangle_{2q, 2} \in {0, 1}^n$</td>
<td></td>
</tr>
<tr>
<td>$k_B \leftarrow</td>
<td>\overline{v}</td>
</tr>
</tbody>
</table>

Secure if decision ring learning with errors problem is hard.
Parameters

160-bit classical security, 80-bit quantum security

- $n = 1024$
- $q = 2^{32} - 1$
- $\chi =$ discrete Gaussian with parameter $\sigma = 8/\sqrt{2\pi}$

- Failure: $2^{-12800}$

- Total communication: 8.1 KiB
Implementation aspect 1:  
**Polynomial arithmetic**

- Polynomial multiplication in $R_q = \mathbb{Z}_q[x]/(x^{1024}+1)$ done with Nussbaumer’s FFT:

  If $2^m = rk$, then

  $$\frac{R[X]}{\langle X^{2^m} + 1 \rangle} \cong \frac{\left( \frac{R[Z]}{\langle Z^r + 1 \rangle} \right)[X]}{\langle X^k - Z \rangle}$$

- Rather than working modulo degree-1024 polynomial with coefficients in $\mathbb{Z}_q$, work modulo:
  - degree-256 polynomial whose coefficients are themselves polynomials modulo a degree-4 polynomial,
  - or degree-32 polynomials whose coefficients are polynomials modulo degree-8 polynomials whose coefficients are polynomials
  - or …
Implementation aspect 2:

Sampling discrete Gaussians

- Security proofs require “small” elements sampled within statistical distance $2^{-128}$ of the true discrete Gaussian
- We use inversion sampling: precompute table of cumulative probabilities
  - For us: 52 elements, size = 10000 bits
- Sampling each coefficient requires six 192-bit integer comparisons and there are 1024 coefficients
  - $51 \cdot 1024$ for constant time

$$D_{\mathbb{Z}, \sigma}(x) = \frac{1}{S} e^{-\frac{x^2}{2\sigma^2}} \quad \text{for } x \in \mathbb{Z}, \sigma \approx 3.2, S = 8$$
Sampling is expensive

<table>
<thead>
<tr>
<th>Operation</th>
<th>constant-time</th>
<th>Cycles non-constant-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>sample $\leftarrow \chi$</td>
<td>1 042 700</td>
<td>668 000</td>
</tr>
<tr>
<td>FFT multiplication</td>
<td>342 800</td>
<td>—</td>
</tr>
<tr>
<td>FFT addition</td>
<td>1 660</td>
<td>—</td>
</tr>
<tr>
<td>\text{dbl}(\cdot) and crossrounding $\langle \cdot \rangle_{2q,2}$</td>
<td>23 500</td>
<td>21 300</td>
</tr>
<tr>
<td>rounding $\lfloor \cdot \rfloor_{2q,2}$</td>
<td>5 500</td>
<td>3 700</td>
</tr>
<tr>
<td>reconciliation $\text{rec}(\cdot, \cdot)$</td>
<td>14 400</td>
<td>6 800</td>
</tr>
</tbody>
</table>
“NewHope”
Alkim, Ducas, Pöppelman, Schwabe.
USENIX Security 2016

- New parameters
- Different error distribution
- Improved performance
- Pseudorandomly generated parameters

- Further performance improvements by others [GS16, LN16, …]

[LN16] Longa, Naehrig. ePrint 2016/504.
Key agreement from LWE

Bos, Costello, Ducas, Mironov, Naehrig, Nikolaenko, Raghunathan, Stebila.
Frodo: Take off the ring! Practical, quantum-safe key exchange from LWE.

https://eprint.iacr.org/2016/659
Decision learning with errors problem with short secrets

**Definition.** Let $n, q \in \mathbb{N}$. Let $\chi$ be a distribution over $\mathbb{Z}$. Let $s \overset{\$}{\leftarrow} \chi^n$.

Define:

- $O_{\chi,s}$: Sample $a \overset{\$}{\leftarrow} \mathcal{U}(\mathbb{Z}_q^n)$, $e \overset{\$}{\leftarrow} \chi$; return $(a, a \cdot s + e)$.

- $U$: Sample $(a, b') \overset{\$}{\leftarrow} \mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)$; return $(a, b')$.

The decision LWE problem with short secrets for $n, q, \chi$ is to distinguish $O_{\chi,s}$ from $U$. 
Hardness of decision LWE

Practice:
- Assume the best way to solve DLWE is to solve LWE.
- Assume solving LWE involves a lattice reduction problem.
- Estimate parameters based on runtime of lattice reduction algorithms.
- (Ignore non-tightness.)

worst-case gap shortest vector problem (GapSVP)

poly-time [BLPRS13]

decision LWE

tight [ACPS09]

decision LWE with short secrets

Generic vs. ideal lattices

- Ring-LWE matrices have additional structure
  - Relies on hardness of a problem in ideal lattices

- LWE matrices have no additional structure
  - Relies on hardness of a problem in generic lattices

- NTRU also relies on a problem in a type of ideal lattices

- Currently, best algorithms for ideal lattice problems are essentially the same as for generic lattices
  - Small constant factor improvement in some cases

- If we want to eliminate this additional structure, can we still get an efficient algorithm?
"Frodo": LWE-DH key agreement

Uses two matrix forms of LWE:
- Public key is $n \times n$ matrix
- Shared secret is $m \times n$ matrix

Secure if decision learning with errors problem is hard (and Gen is a secure PRF).
Rounding

• We extract 4 bits from each of the 64 matrix entries in the shared secret.
  • More granular form of Peikert’s rounding.

Parameter sizes, rounding, and error distribution all found via search scripts.

Error distribution

• Close to discrete Gaussian in terms of Rényi divergence (1.000301)
• Only requires 12 bits of randomness to sample
Parameters

"Recommended"
- 156-bit classical security, 142-bit quantum security, 112-bit plausible lower bound
- $n = 752$, $m = 8$, $q = 2^{15}$
- $\chi$ = approximation to rounded Gaussian with 11 elements
- Failure: $2^{-36.5}$
- Total communication: 22.6 KiB

"Paranoid"
- 191-bit classical security, 174-bit quantum security, 138-bit plausible lower bound
- $n = 864$, $m = 8$, $q = 2^{15}$
- $\chi$ = approximation to rounded Gaussian with 13 elements
- Failure: $2^{-35.8}$
- Total communication: 25.9 KiB

All known variants of the sieving algorithm require a list of vectors to be created of this size.
Standalone performance
## Implementations

### Our implementations
- BCNS15
- Frodo

### Pure C implementations
- Constant time

### Compare with others
- RSA 3072-bit (OpenSSL 1.0.1f)
- ECDH `nistp256` (OpenSSL)
  - Use assembly code
- NewHope
- NTRU `EES743EP1`
- SIDH (Isogenies) (MSR)

Pure C implementations
### Standalone performance

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Alice0 (ms)</th>
<th>Bob (ms)</th>
<th>Alice1 (ms)</th>
<th>Communication (bytes) A→B</th>
<th>B→A</th>
<th>Claimed security</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA 3072-bit</td>
<td>—</td>
<td>0.09</td>
<td>4.49</td>
<td>387 / 0*</td>
<td>384</td>
<td>128</td>
</tr>
<tr>
<td>ECDH nistp256</td>
<td>0.366</td>
<td>0.698</td>
<td>0.331</td>
<td>32</td>
<td>32</td>
<td>128</td>
</tr>
<tr>
<td>BCNS</td>
<td>1.01</td>
<td>1.59</td>
<td>0.174</td>
<td>4,096</td>
<td>4,224</td>
<td>163</td>
</tr>
<tr>
<td>NewHope</td>
<td>0.112</td>
<td>0.164</td>
<td>0.034</td>
<td>1,824</td>
<td>2,048</td>
<td>229</td>
</tr>
<tr>
<td>NTRU EES743EP1</td>
<td>2.00</td>
<td>0.281</td>
<td>0.148</td>
<td>1,027</td>
<td>1,022</td>
<td>256</td>
</tr>
<tr>
<td>SIDH</td>
<td>135</td>
<td>464</td>
<td>301</td>
<td>564</td>
<td>564</td>
<td>192</td>
</tr>
<tr>
<td>Frodo Recomm.</td>
<td>1.13</td>
<td>1.34</td>
<td>0.13</td>
<td>11,377</td>
<td>11,296</td>
<td>156</td>
</tr>
<tr>
<td>Frodo Paranoid</td>
<td>1.25</td>
<td>1.64</td>
<td>0.15</td>
<td>13,057</td>
<td>12,976</td>
<td>191</td>
</tr>
</tbody>
</table>

Note somewhat incomparable security levels

---

x86_64, 2.6 GHz Intel Xeon E5 (Sandy Bridge) – Google n1-standard-4
## Standalone performance

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Performance</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA 3072-bit</td>
<td>Fast (4 ms)</td>
<td>Small (0.3 KiB)</td>
</tr>
<tr>
<td>ECDH nistp256</td>
<td>Very fast (0.7 ms)</td>
<td>Very small (0.03 KiB)</td>
</tr>
<tr>
<td>BCNS</td>
<td>Fast (1.5 ms)</td>
<td>Medium (4 KiB)</td>
</tr>
<tr>
<td>NewHope</td>
<td>Very fast (0.2 ms)</td>
<td>Medium (2 KiB)</td>
</tr>
<tr>
<td>NTRU EES743EP1</td>
<td>Fast (0.3–1.2 ms)</td>
<td>Medium (1 KiB)</td>
</tr>
<tr>
<td>SIDH</td>
<td>Very slow (400 ms)</td>
<td>Small (0.5 KiB)</td>
</tr>
<tr>
<td>Frodo Recommended</td>
<td>Fast (1.4 ms)</td>
<td>Large (11 KiB)</td>
</tr>
<tr>
<td>McBits*</td>
<td>Very fast (0.5 ms)</td>
<td>Very large (360 KiB)</td>
</tr>
</tbody>
</table>


Note somewhat incomparable security levels.
TLS integration and performance
Integration into TLS 1.2

**New ciphersuite:**
TLS-KEX-SIG-AES256-GCM-SHA384

- SIG = RSA or ECDSA signatures for authentication
- KEX = Post-quantum key exchange
- AES-256 in GCM for authenticated encryption
- SHA-384 for HMAC-KDF
Security within TLS 1.2

Model:
- authenticated and confidential channel establishment (ACCE) [JKSS12]

Theorem:
- signed LWE/ring-LWE ciphersuite is ACCE-secure if underlying primitives (signatures, LWE/ring-LWE, authenticated encryption) are secure
  - Interesting technical detail for ACCE provable security people: need to move server’s signature to end of TLS handshake because oracle-DH assumptions don’t hold for ring-LWE or use an IND-CCA KEM for key exchange via e.g. [FO99]

**TLS performance**

<table>
<thead>
<tr>
<th>Handshake latency</th>
<th>Connection throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Time from when client sends first TCP packet till client receives first application data</td>
<td>- Number of connections per second at server before server latency spikes</td>
</tr>
<tr>
<td>- No load on server</td>
<td></td>
</tr>
</tbody>
</table>
TLS handshake latency compared to NewHope-ECDSA

- ECDH nistp256: 1.33x
- BCNS: 1.69x
- NewHope: 1.64x
- NTRU: 1.65x
- Frodo Recom.: 1.71x

Note somewhat incomparable security levels.
TLS connection throughput
ECDSA signatures

Payload size

x86_64, 2.6 GHz Intel Xeon E5 (Sandy Bridge) – server Google n1-standard-4, client –32

Note somewhat incomparable security levels

bigger (top) is better
Hybrid ciphersuites

- Use both post-quantum key exchange and traditional key exchange

- Example:
  - ECDHE + NewHope
    - Used in Google Chrome experiment
  - ECDHE + Frodo

- Session key secure if either problem is hard

- Why use post-quantum?
  - (Potential) security against future quantum computer

- Why use ECDHE?
  - Security not lost against existing adversaries if post-quantum cryptanalysis advances
TLS connection throughput – hybrid w/ECDHE
ECDSA signatures

Payload size

x86_64, 2.6 GHz Intel Xeon E5 (Sandy Bridge) – server Google n1-standard-4, client -32

Note somewhat incomparable security levels
Open Quantum Safe

Collaboration with Mosca et al., University of Waterloo

https://github.com/open-quantum-safe/
Open Quantum Safe

- Open source C library
- Common interface for key exchange and digital signatures

1. Collect post-quantum implementations together
   - Our own software
   - Thin wrappers around existing open source implementations
   - Contributions from others

2. Enable direct comparison of implementations

3. Support prototype integration into application level protocols
   - Don’t need to re-do integration for each new primitive – how we did Frodo experiments
Open Quantum Safe architecture

OQS benchmark
Apache httpd
OpenSSL
OTR
...

Open Quantum Safe Library

OQS-KEX
Ring-LWE
LWE
McEliece
NTRU
SIDH
BCNS15
New Hope

OQS-SIG
Hash
LWE/ring-LWE

Application integrations
API
Primitive implementations

API
Primitive implementations
Getting involved and using OQS

https://github.com/open-quantum-safe/

If you’re writing post-quantum implementations:
  • We’d love to coordinate on API
  • And include your software if you agree

If you want to prototype or evaluate post-quantum algorithms in applications:
  • Maybe OQS will be helpful to you

We’d love help with:
  • Code review and static analysis
  • Signature scheme implementations
  • Additional application-level integrations
Summary
Summary

- Exciting research area – lots of opportunities!
- Ring-LWE is fast and fairly small
- LWE can achieve reasonable key sizes
- Hybrid ciphersuites will probably play a role in the transition
- Performance differences are muted in application-level protocols
- Parameter sizes and efficiency likely to evolve

- Post-quantum key exchange soon to be in demand
Now hiring!

- Post-doc in any area of post-quantum cryptography
  - Applied or theoretical


For more info:
https://www.douglas.stebila.ca/research/postdoc/
Links

Ring-LWE key exchange
- https://eprint.iacr.org/2014/599
- https://github.com/dstebila/rlwekex

LWE key exchange (Frodo)
- https://eprint.iacr.org/2016/659
- https://github.com/lwe-frodo/ (coming soon)

Open Quantum Safe
- https://github.com/open-quantum-safe/

Post-doc
- https://www.douglas.stebila.ca/research/postdoc/