Post-quantum key exchange for the Internet

Douglas Stebila McMaster University

Selected Areas in Cryptography • August 12, 2016

Acknowledgements

Collaborators

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

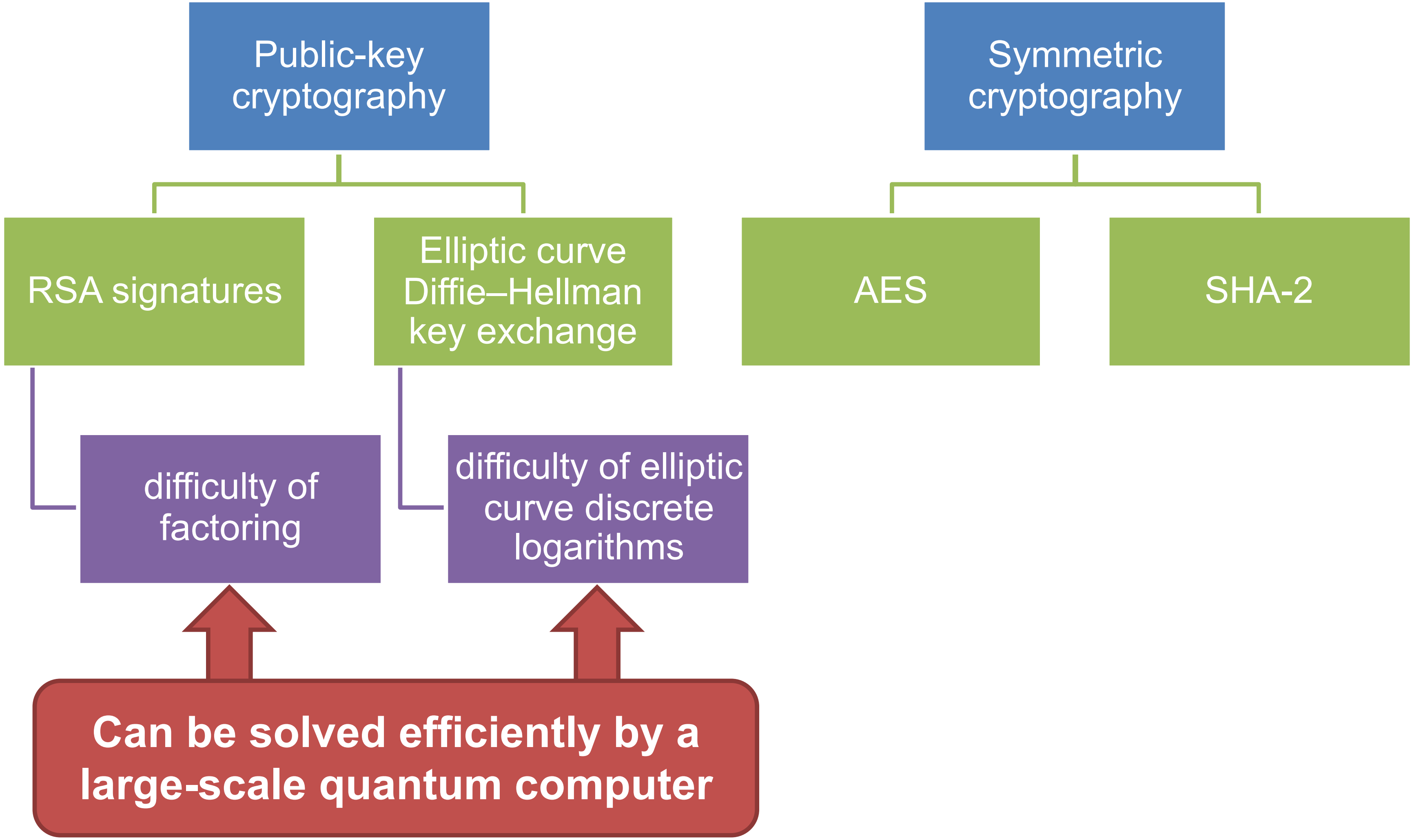
Support

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

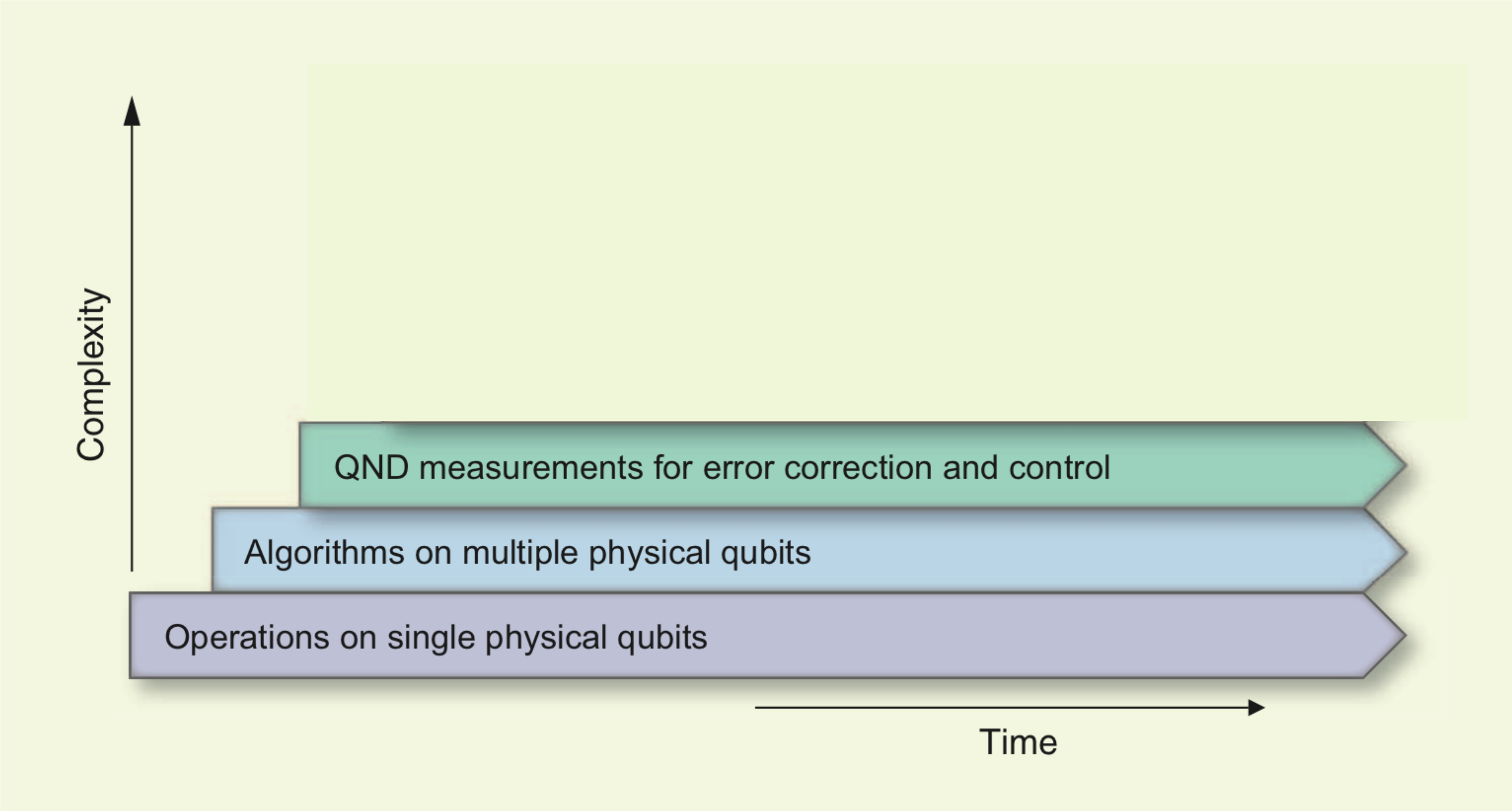
Motivation

Contemporary cryptography

TLS-ECDHE-RSA-AES128-GCM-SHA256

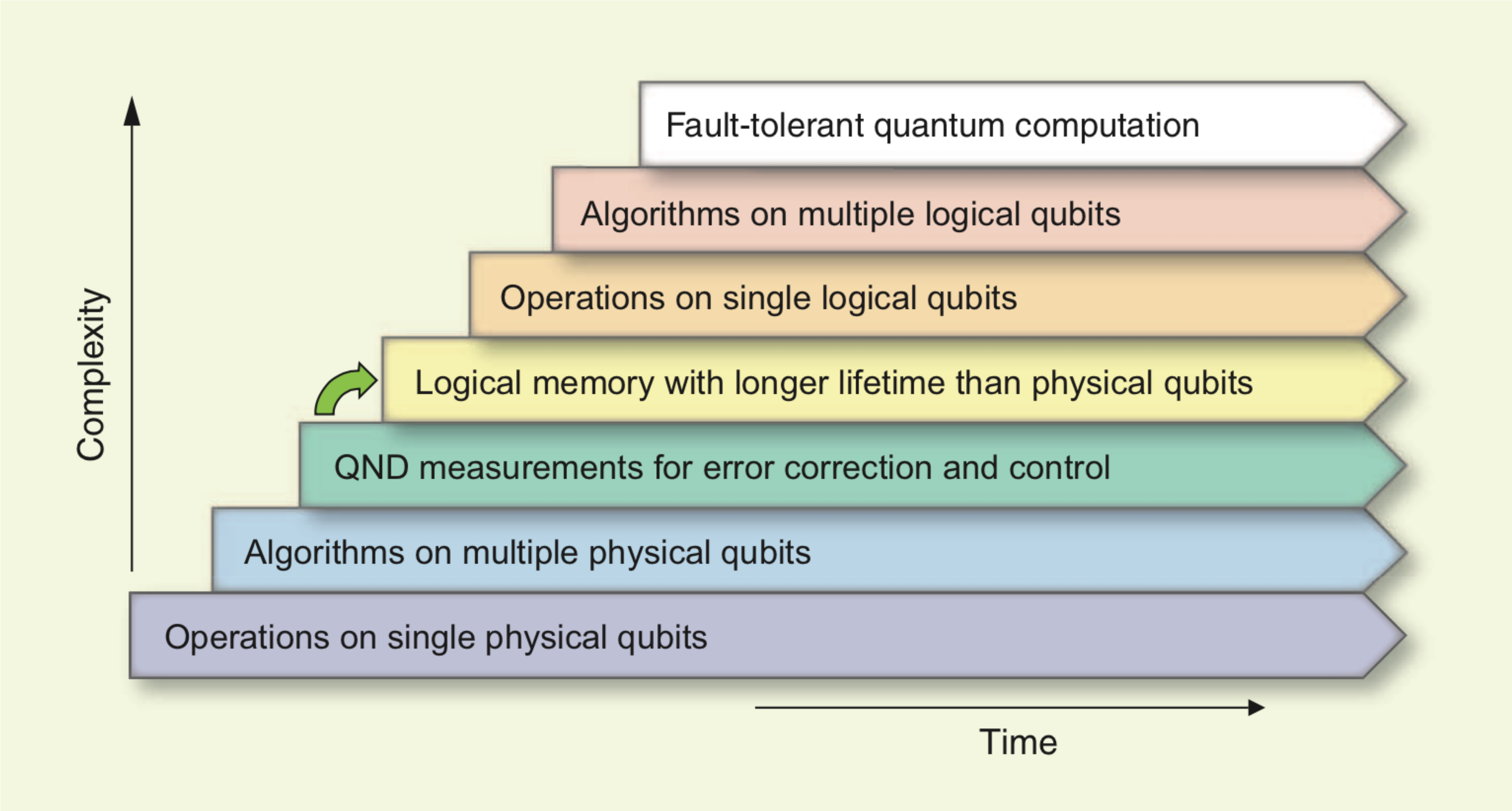


Building quantum computers



Devoret, Schoelkopf. *Science* 339:1169–1174, March 2013.

Building quantum computers



Devoret, Schoelkopf. *Science* 339:1169–1174, March 2013.

When will a large-scale quantum computer be built?

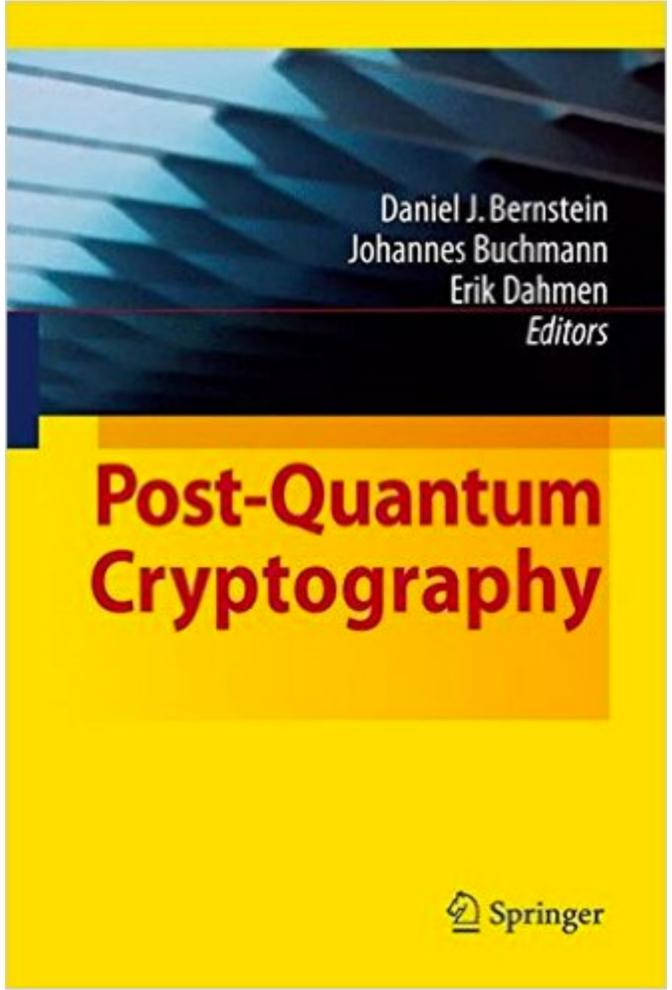
"I estimate a 1/7 chance of breaking RSA-2048 by 2026 and a 1/2 chance by 2031."

— Michele Mosca, November 2015
https://eprint.iacr.org/2015/1075

Post-quantum cryptography in academia

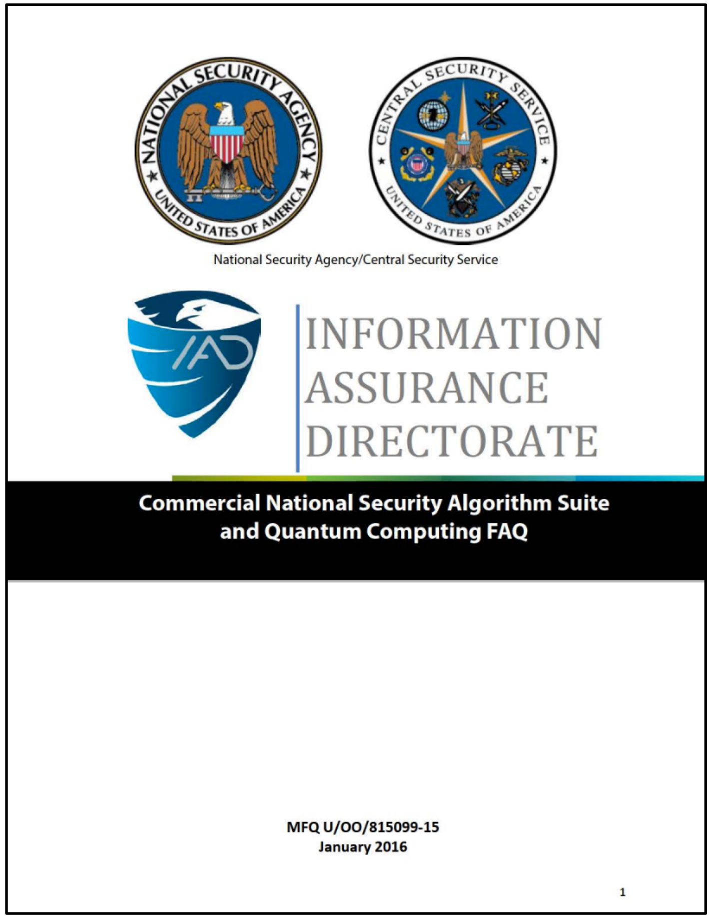
<u>Conference series</u>

- PQCrypto 2006
- PQCrypto 2008
- PQCrypto 2010
- PQCrypto 2011
- PQCrypto 2013
- PQCrypto 2014
- PQCrypto 2016



2009

Post-quantum cryptography in government


National Security Agency/Central Security Service

INFORMATION ASSURANCE DIRECTORATE

Commercial National Security Algorithm Suite and Quantum Computing FAQ

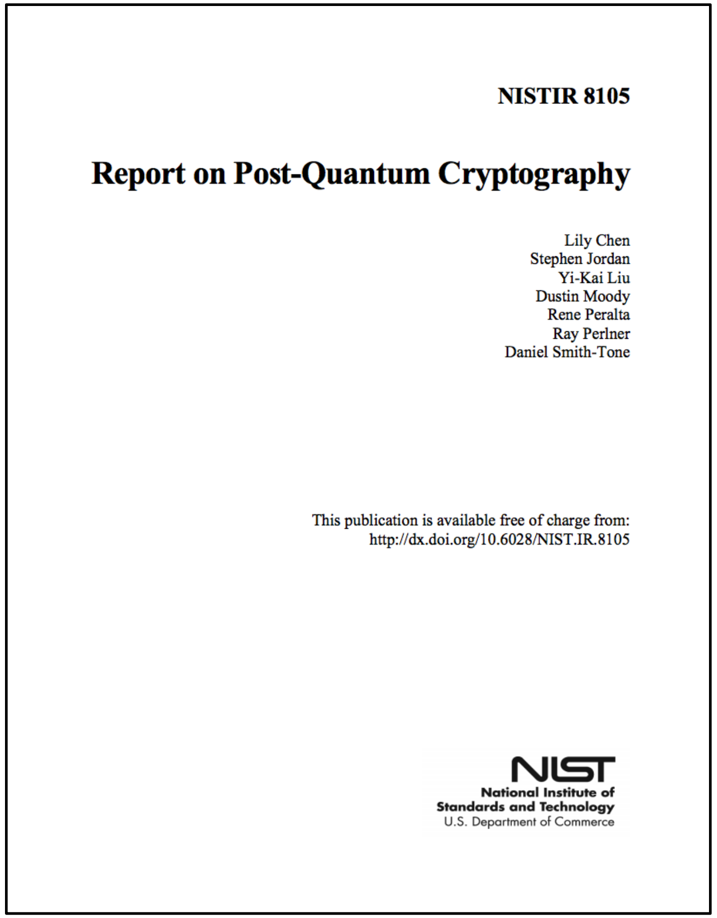
MFQ U/OO/815099-15
January 2016

1

Aug. 2015 (Jan. 2016)

"IAD will initiate a transition to quantum resistant algorithms in the not too distant future."

– NSA Information Assurance Directorate, Aug. 2015


NISTIR 8105

Report on Post-Quantum Cryptography

Lily Chen
Stephen Jordan
Yi-Kai Liu
Dustin Moody
Rene Peralta
Ray Perlner
Daniel Smith-Tone

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.IR.8105

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Apr. 2016

NIST Post-quantum Crypto Project timeline

September 16, 2016	**Feedback on call for proposals**
Fall 2016	Formal call for proposals
November 2017	Deadline for submissions
Early 2018	Workshop – submitters' presentations
3-5 years	Analysis phase
2 years later	Draft standards ready

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

$$\underset{\begin{matrix}\\\mathbb{Z}_{13}^{7\times 4}\end{matrix}}{\begin{bmatrix} 4 & 1 & 11 & 10 \\ 5 & 5 & 9 & 5 \\ 3 & 9 & 0 & 10 \\ 1 & 3 & 3 & 2 \\ 12 & 7 & 3 & 4 \\ 6 & 5 & 11 & 4 \\ 3 & 3 & 5 & 0 \end{bmatrix}} \times \underset{\begin{matrix}\textbf{secret}\\\mathbb{Z}_{13}^{4\times 1}\end{matrix}}{\begin{bmatrix} \\ \\ \\ \end{bmatrix}} = \underset{\begin{matrix}\\\mathbb{Z}_{13}^{7\times 1}\end{matrix}}{\begin{bmatrix} 4 \\ 8 \\ 1 \\ 10 \\ 4 \\ 12 \\ 9 \end{bmatrix}}$$

Linear system problem: given **blue**, find **red**

Solving systems of linear equations

$$\begin{array}{ccccc} \mathbb{Z}_{13}^{7\times 4} & & \begin{array}{c}\textbf{secret}\\ \mathbb{Z}_{13}^{4\times 1}\end{array} & & \mathbb{Z}_{13}^{7\times 1} \\ \begin{bmatrix} 4 & 1 & 11 & 10 \\ 5 & 5 & 9 & 5 \\ 3 & 9 & 0 & 10 \\ 1 & 3 & 3 & 2 \\ 12 & 7 & 3 & 4 \\ 6 & 5 & 11 & 4 \\ 3 & 3 & 5 & 0 \end{bmatrix} & \times & \begin{bmatrix} 6 \\ 9 \\ 11 \\ 11 \end{bmatrix} & = & \begin{bmatrix} 4 \\ 8 \\ 1 \\ 10 \\ 4 \\ 12 \\ 9 \end{bmatrix} \end{array}$$

Easily solved using Gaussian elimination (Linear Algebra 101)

Linear system problem: given **blue**, find **red**

Learning with errors problem

$$\overset{\substack{\textbf{random} \\ \mathbb{Z}_{13}^{7\times 4}}}{\begin{bmatrix} 4 & 1 & 11 & 10 \\ 5 & 5 & 9 & 5 \\ 3 & 9 & 0 & 10 \\ 1 & 3 & 3 & 2 \\ 12 & 7 & 3 & 4 \\ 6 & 5 & 11 & 4 \\ 3 & 3 & 5 & 0 \end{bmatrix}} \times \overset{\substack{\textbf{secret} \\ \mathbb{Z}_{13}^{4\times 1}}}{\begin{bmatrix} 6 \\ 9 \\ 11 \\ 11 \end{bmatrix}} + \overset{\substack{\textbf{small noise} \\ \mathbb{Z}_{13}^{7\times 1}}}{\begin{bmatrix} 0 \\ -1 \\ 1 \\ 1 \\ 1 \\ 0 \\ -1 \end{bmatrix}} = \overset{\mathbb{Z}_{13}^{7\times 1}}{\begin{bmatrix} 4 \\ 7 \\ 2 \\ 11 \\ 5 \\ 12 \\ 8 \end{bmatrix}}$$

Learning with errors problem

$$\begin{array}{ccccccc} \textbf{random} & & \textbf{secret} & & \textbf{small noise} & & \\ \mathbb{Z}_{13}^{7\times 4} & & \mathbb{Z}_{13}^{4\times 1} & & \mathbb{Z}_{13}^{7\times 1} & & \mathbb{Z}_{13}^{7\times 1} \\ \begin{bmatrix} 4 & 1 & 11 & 10 \\ 5 & 5 & 9 & 5 \\ 3 & 9 & 0 & 10 \\ 1 & 3 & 3 & 2 \\ 12 & 7 & 3 & 4 \\ 6 & 5 & 11 & 4 \\ 3 & 3 & 5 & 0 \end{bmatrix} & \times & \begin{bmatrix} \\ \\ \\ \end{bmatrix} & + & \begin{bmatrix} \\ \\ \\ \\ \\ \\ \end{bmatrix} & = & \begin{bmatrix} 4 \\ 7 \\ 2 \\ 11 \\ 5 \\ 12 \\ 8 \end{bmatrix} \end{array}$$

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

Decision learning with errors problem

$$\underset{\mathbb{Z}_{13}^{7\times 4}}{\overset{\textbf{random}}{\begin{bmatrix} 4 & 1 & 11 & 10 \\ 5 & 5 & 9 & 5 \\ 3 & 9 & 0 & 10 \\ 1 & 3 & 3 & 2 \\ 12 & 7 & 3 & 4 \\ 6 & 5 & 11 & 4 \\ 3 & 3 & 5 & 0 \end{bmatrix}}} \times \underset{\mathbb{Z}_{13}^{4\times 1}}{\overset{\textbf{secret}}{\begin{bmatrix} \\ \\ \\ \end{bmatrix}}} + \underset{\mathbb{Z}_{13}^{7\times 1}}{\overset{\textbf{small noise}}{\begin{bmatrix} \\ \\ \\ \\ \\ \\ \end{bmatrix}}} = \underset{\mathbb{Z}_{13}^{7\times 1}}{\overset{\textbf{looks random}}{\begin{bmatrix} 4 \\ 7 \\ 2 \\ 11 \\ 5 \\ 12 \\ 8 \end{bmatrix}}}$$

Decision LWE problem: given **blue**, distinguish **green** from random

Toy example versus real-world example

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

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

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

8

752

2738	3842	3345	2979	…
2896	595	3607		
377	1575			
2760				
…				

752 × 28 × 15 bits = **11 KiB**

Ring learning with errors problem

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

4	1	11	10
10	4	1	11
11	10	4	1
1	11	10	4
4	1	11	10
10	4	1	11
11	10	4	1

Each row is the cyclic shift of the row above

Ring learning with errors problem

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

4	1	11	10
3	4	1	11
2	3	4	1
12	2	3	4
9	12	2	3
10	9	12	2
11	10	9	12

Each row is the cyclic shift of the row above

…

with a special wrapping rule: *x* wraps to –*x* mod 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

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

	$4 + 1x + 11x^2 + 10x^3$	**random**
×	$6 + 9x + 11x^2 + 11x^3$	**secret**
+	$0 - 1x + 1x^2 + 1x^3$	**small noise**
=	$10 + 5x + 10x^2 + 7x^3$	

Ring learning with errors problem

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

	$4 + 1x + 11x^2 + 10x^3$	**random**
×		**secret**
+		**small noise**
=	$10 + 5x + 10x^2 + 7x^3$	

Computational ring-LWE problem: given **blue**, find **red**

Decision ring learning with errors problem

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

	$4 + 1x + 11x^2 + 10x^3$	**random**
$\times$		**secret**
$+$		**small noise**
$=$	$10 + 5x + 10x^2 + 7x^3$	**looks random**

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

Decision ring learning with errors problem **with small secrets**

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

	$4 + 1x + 11x^2 + 10x^3$	random
×	$1 + 0x - 1x^2 + 2x^3$	small secret
+		small noise
=	$10 + 5x + 10x^2 + 7x^3$	looks random

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

Problems

[Reg05] Regev, *STOC 2005*; *J. ACM 2009*.

Computational LWE problem	Decision LWE problem
Computational ring-LWE problem	Decision ring-LWE problem

with or without short secrets

[LPR10] Lyubashevsky, Piekert, Regev. *EUROCRYPT 2010.*

Key agreement from ring-LWE

Bos, Costello, Naehrig, Stebila.
Post-quantum key exchange for the TLS protocol from the ring learning with errors problem.
IEEE Symposium on Security & Privacy (S&P) 2015.

https://www.douglas.stebila.ca/research/papers/SP-BCNS15/

Decision ring learning with errors problem with short secrets

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 χ be a distribution over R_q.

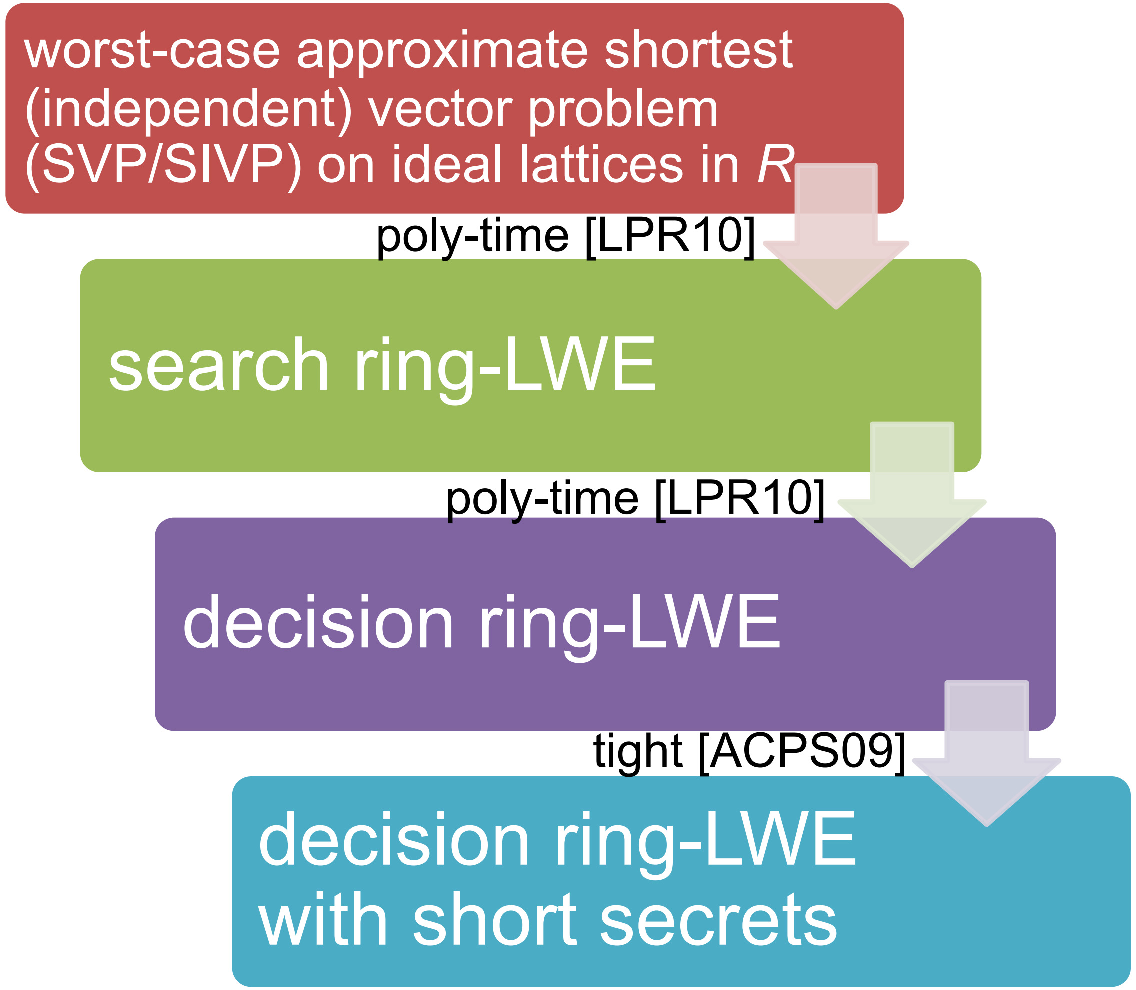
Let $s \xleftarrow{\$} \chi$.

Define:

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

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

Hardness of decision ring-LWE



<u>Practice:</u>

- 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]

[LPR10] Lyubashevsky, Piekert, Regev. *EUROCRYPT 2010.*
[ACPS15] Applebaum, Cash, Peikert, Sahai. *CRYPTO 2009.*
[CKMS16] Chatterjee, Koblitz, Menezes, Sarkar. ePrint 2016/360.

Basic ring-LWE-DH key agreement (unauthenticated)

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

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

Alice

secret:
random "small" s, e in R_q

$b = a \bullet s + e$ →

← $b' = a \bullet s' + e'$

shared secret:
$s \bullet b' = s \bullet (a \bullet s' \bullet e') \approx s \bullet a \bullet s'$

Bob

secret:
random "small" s', e' in R_q

shared secret:
$b \bullet s' \approx s \bullet a \bullet s'$

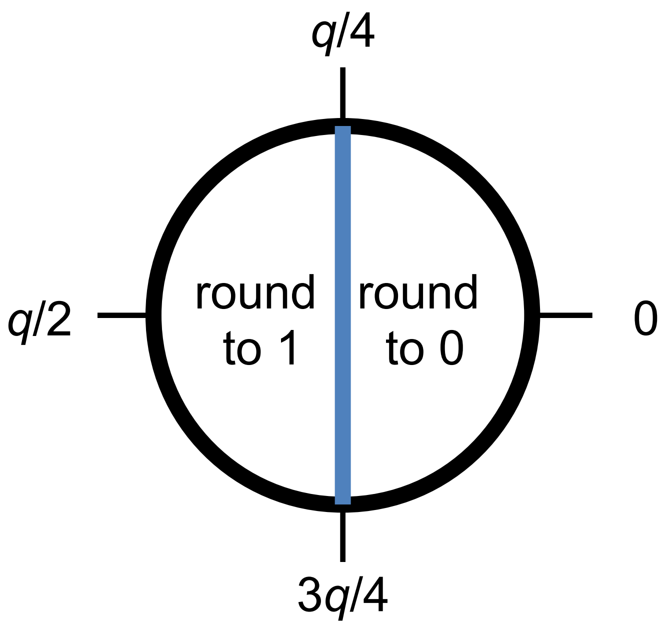
These are only approximately equal ⇒ need rounding

Rounding

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

Basic rounding

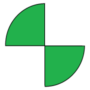
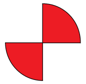
- Round either to 0 or *q*/2
- Treat *q*/2 as 1

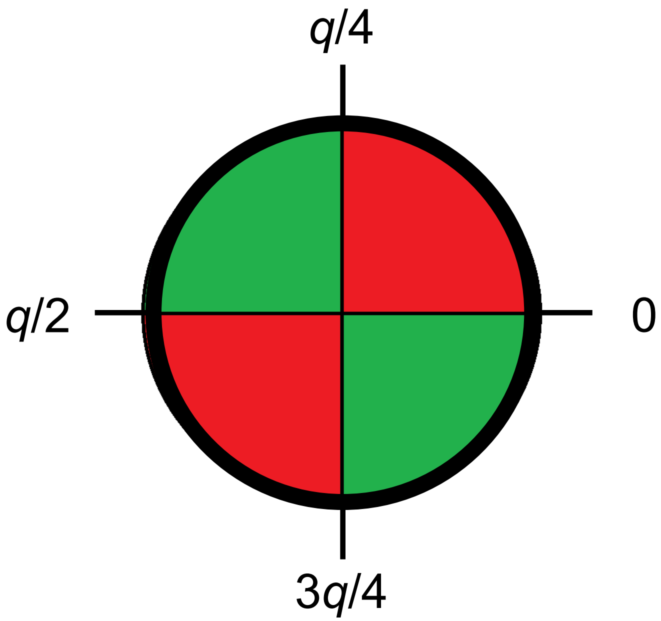


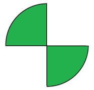
This works most of the time: prob. failure 2^{-10}.

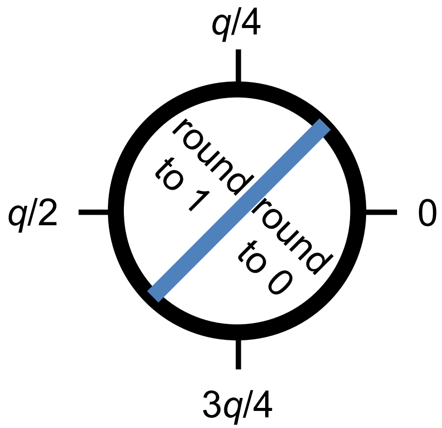
Not good enough: we need exact key agreement.

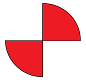
Better rounding (Peikert)

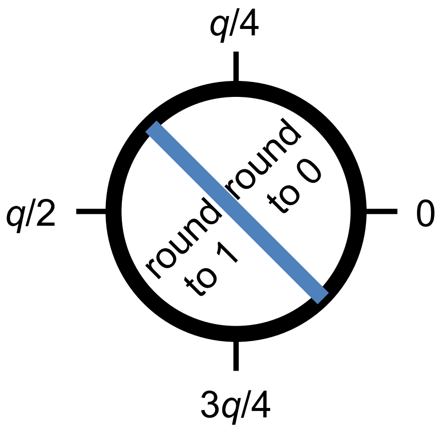
Bob says which of two regions the value is in:  or 



If 

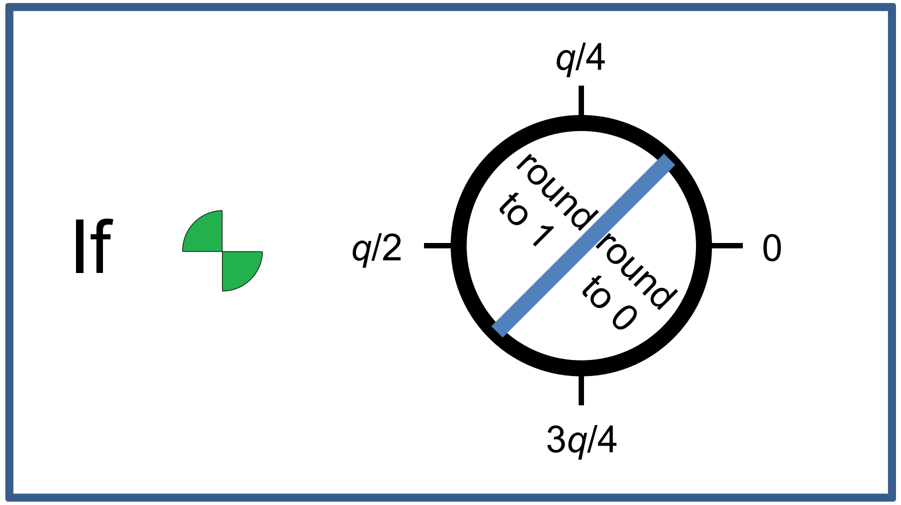


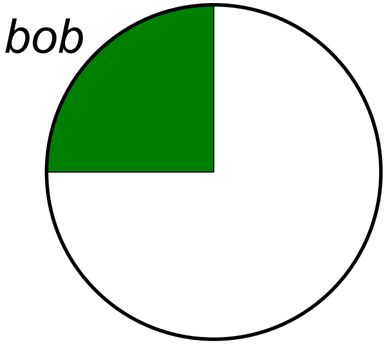
If 

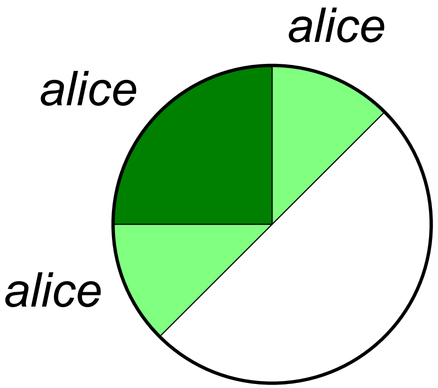


Better rounding (Peikert)

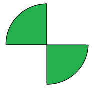
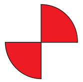
- If | *alice* – *bob* | ≤ $q/8$, then this always works.







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

- Security not affected: revealing  or  leaks no information

Exact ring-LWE-DH key agreement (unauthenticated)

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

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

Alice

secret:
random "small" s, e in R_q

Bob

secret:
random "small" s', e' in R_q

$b = a \bullet s + e$ →

← $b' = a \bullet s' + e'$, or

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

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

Ring-LWE-DH key agreement

Public parameters		
Decision R-LWE parameters q, n, χ		
$a \xleftarrow{\$} \mathcal{U}(R_q)$		
Alice		**Bob**
$s, e \xleftarrow{\$} \chi$		$s', e' \xleftarrow{\$} \chi$
$b \leftarrow as + e \in R_q$	$\xrightarrow{b}$	$b' \leftarrow as' + e' \in R_q$
		$e'' \xleftarrow{\$} \chi$
		$v \leftarrow bs' + e'' \in R_q$
		$\overline{v} \xleftarrow{\$} \mathrm{dbl}(v) \in R_{2q}$
	$\xleftarrow{b', c}$	$c \leftarrow \langle \overline{v} \rangle_{2q,2} \in \{0,1\}^n$
$k_A \leftarrow \mathrm{rec}(2b's, c) \in \{0,1\}^n$		$k_B \leftarrow \lfloor \overline{v} \rceil_{2q,2} \in \{0,1\}^n$

Secure if decision ring learning with errors problem is hard.

Parameters

160-bit classical security,
80-bit quantum security

- $n = 1024$
- $q = 2^{32} - 1$
- χ = discrete Gaussian with parameter sigma = 8/sqrt(2π)

- Failure: 2^{-12800}

- Total communication: 8.1 KiB

Implementation aspect 1:

Polynomial arithmetic

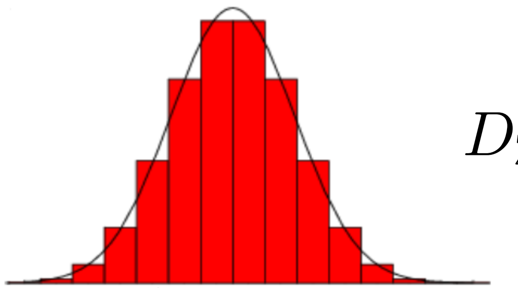
- Polynomial multiplication in R_q = $\mathbf{Z}_q[x]/(x^{1024}+1)$ done with Nussbaumer's FFT:

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 $\mathbf{Z}_q$, 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



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

- 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 • 1024 for constant time

Sampling is expensive

Operation	**Cycles**	
	constant-time	**non-constant-time**
sample $\xleftarrow{\$} \chi$	1 042 700	668 000
FFT multiplication	342 800	—
FFT addition	1 660	—
$\mathrm{dbl}(\cdot)$ and crossrounding $\langle\cdot\rangle_{2q,2}$	23 500	21 300
rounding $\lfloor\cdot\rceil_{2q,2}$	5 500	3,700
reconciliation $\mathrm{rec}(\cdot,\cdot)$	14 400	6 800

“NewHope”

Alkim, Ducas, Pöppelman, Scwabe.
USENIX Security 2016

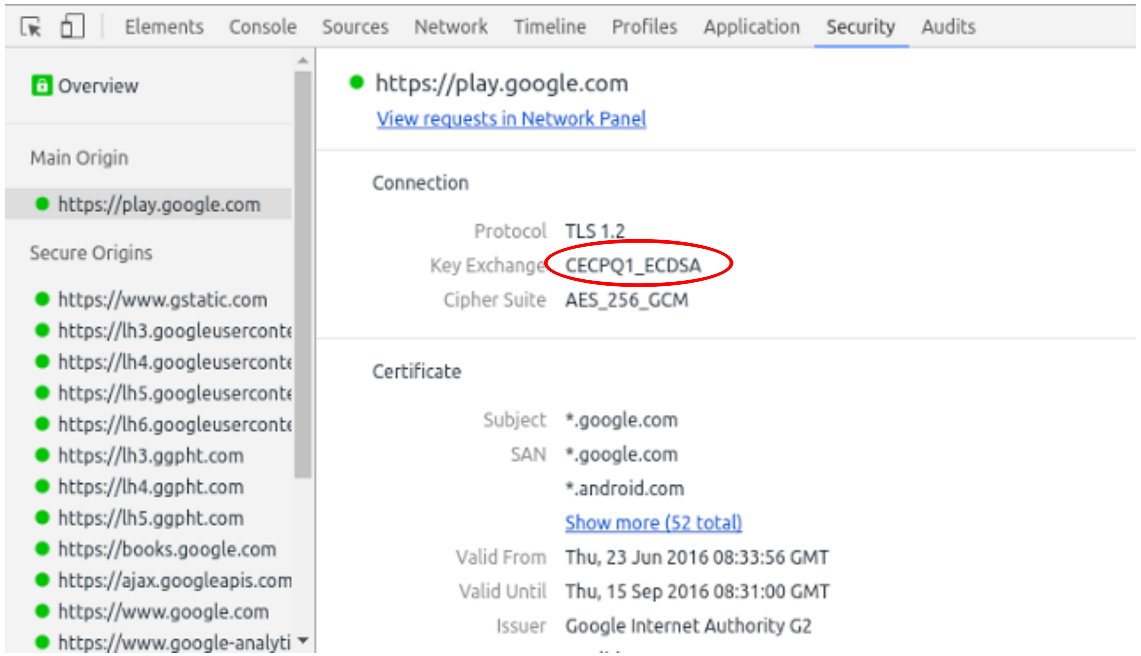
- New parameters
- Different error distribution
- Improved performance
- Pseudorandomly generated parameters

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

Google Security Blog

Experimenting with Post-Quantum Cryptography

July 7, 2016



[GS16] Gueron, Schlieker. ePrint 2016/467.
[LN16] Longa, Naehrig. ePrint 2016/504.

https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html

Key agreement from LWE

Bos, Costello, Ducas, Mironov, Naehrig, Nikolaenko, Raghunathan, Stebila.
Frodo: Take off the ring! Practical, quantum-safe key exchange from LWE.
ACM Conference on Computer and Communications Security (CCS) 2016.

https://eprint.iacr.org/2016/659

Decision learning with errors problem with short secrets

Definition. Let $n, q \in \mathbb{N}$. Let χ be a distribution over $\mathbb{Z}$.

Let $\mathbf{s} \xleftarrow{\$} \chi^n$.

Define:

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

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

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

Hardness of decision LWE

worst-case gap shortest vector problem (GapSVP)

poly-time [BLPRS13]

decision LWE

tight [ACPS09]

decision LWE with short secrets

<u>Practice:</u>

- 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.)

[BLPRS13] Brakerski, Langlois, Peikert, Regev, Stehlé. *STOC 2013.*
[ACPS15] Applebaum, Cash, Peikert, Sahai. *CRYPTO 2009*.

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

Alice

$\mathsf{seed}_{\mathbf{A}} \xleftarrow{\$} U(\{0,1\}^s)$

$\mathbf{A} \leftarrow \mathsf{Gen}(\mathsf{seed}_{\mathbf{A}})$

A generated pseudorandomly

$\mathbf{S}, \mathbf{E} \xleftarrow{\$} \chi(\mathbb{Z}_q^{n \times \bar{n}})$

$\boxed{\mathbf{B} \leftarrow \mathbf{AS} + \mathbf{E}}$

$\xrightarrow{\mathsf{seed}_{\mathbf{A}}, \mathbf{B}} \in \{0,1\}^s \times \mathbb{Z}_q^{n \times \bar{n}}$

Bob

$\mathbf{A} \leftarrow \mathsf{Gen}(\mathsf{seed}_{\mathbf{A}})$

$\mathbf{S}', \mathbf{E}' \xleftarrow{\$} \chi(\mathbb{Z}_q^{\bar{m} \times n})$

$\boxed{\mathbf{B}' \leftarrow \mathbf{S}'\mathbf{A} + \mathbf{E}'}$

$\mathbf{E}'' \xleftarrow{\$} \chi(\mathbb{Z}_q^{\bar{m} \times \bar{n}})$

$\boxed{\mathbf{V} \leftarrow \mathbf{S}'\mathbf{B} + \mathbf{E}''}$

$\mathbf{C} \leftarrow \langle \mathbf{V} \rangle_{2^B}$

$\xleftarrow{\mathbf{B}', \mathbf{C}} \in \mathbb{Z}_q^{\bar{m} \times n} \times \mathbb{Z}_2^{\bar{m} \times \bar{n}}$

Alice: $K \leftarrow \mathsf{rec}(\boxed{\mathbf{B}'\mathbf{S}}, \mathbf{C})$

Bob: $K \leftarrow \lfloor \mathbf{V} \rceil_{2^B}$

Uses two matrix forms of LWE:

- Public key is *n* x <u>*n*</u> matrix
- Shared secret is <u>*m*</u> x <u>*n*</u> matrix

Secure if decision learning with errors problem is hard

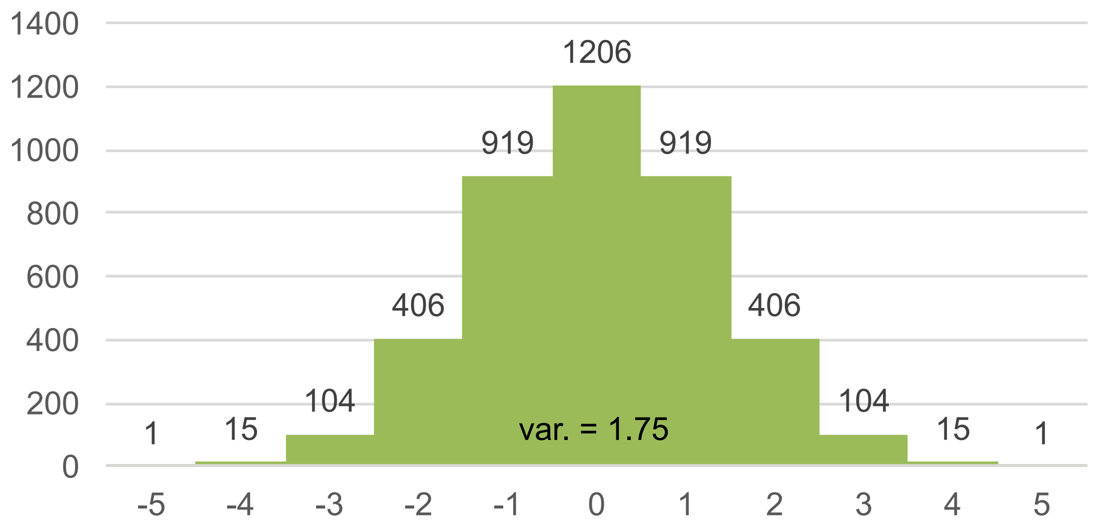
(and Gen is a secure PRF).

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

All known variants of the sieving algorithm require a list of vectors to be created of this size

"Recommended"

- 156-bit classical security,
  142-bit quantum security,
  112-bit plausible lower bound
- n = 752, m = 8, q = 2^{15}
- χ = 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}
- χ = approximation to rounded Gaussian with 13 elements
- Failure: $2^{-35.8}$
- Total communication: 25.9 KiB

Standalone performance

Implementations

<u>Our implementations</u>

- BCNS15
- Frodo

Pure C implementations
Constant time

<u>Compare with others</u>

- RSA 3072-bit (OpenSSL 1.0.1f)
- ECDH `nistp256` (OpenSSL)

Use assembly code

- NewHope
- NTRU `EES743EP1`
- SIDH (Isogenies) (MSR)

Pure C implementations

Standalone performance

Scheme	Alice0 (ms)	Bob (ms)	Alice1 (ms)	Communication (bytes) A→B	B→A	Claimed security classical	quantum
RSA 3072-bit	—	0.09	4.49	387 / 0*	384	128	—
ECDH `nistp256`	0.366	0.698	0.331	32	32	128	—
BCNS	1.01	1.59	0.174	4,096	4,224	163	76
NewHope	0.112	0.164	0.034	1,824	2,048	229	206
NTRU `EES743EP1`	2.00	0.281	0.148	1,027	1,022	256	128
SIDH	135	464	301	564	564	192	128
Frodo Recomm.	**1.13**	**1.34**	**0.13**	**11,377**	**11,296**	**156**	**142**
Frodo Paranoid	1.25	1.64	0.15	13,057	12,976	191	174

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

Note somewhat incomparable security levels

Standalone performance

RSA 3072-bit	Fast (4 ms)	Small (0.3 KiB)
ECDH `nistp256`	Very fast (0.7 ms)	Very small (0.03 KiB)
BCNS	Fast (1.5 ms)	Medium (4 KiB)
NewHope	Very fast (0.2 ms)	Medium (2 KiB)
NTRU `EES743EP1`	Fast (0.3–1.2 ms)	Medium (1 KiB)
SIDH	Very slow (400 ms)	Small (0.5 KiB)
Frodo Recommended	Fast (1.4 ms)	Large (11 KiB)
McBits*	Very fast (0.5 ms)	Very large (360 KiB)

* McBits results from source paper [BCS13] Bernstein, Chou, Schwabe. *CHES 2013.*

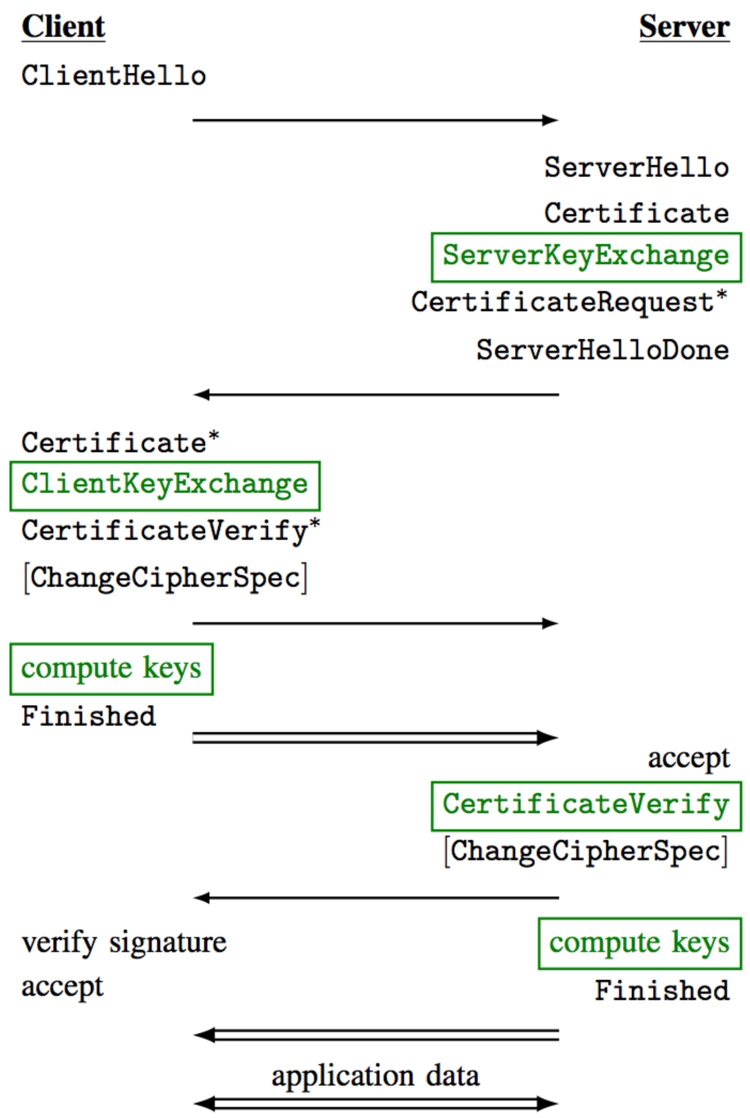
Note somewhat incomparable security levels

TLS integration and performance

Integration into TLS 1.2

<u>New ciphersuite:</u>
`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]

[JKSS12] Jager, Kohlar, Schäge, Schwenk. *CRYPTO 2012*.
[FO99] Fujisaki, Okamoto. *PKC 1999*.

TLS performance

Handshake latency

- Time from when client sends first TCP packet till client receives first application data
- No load on server

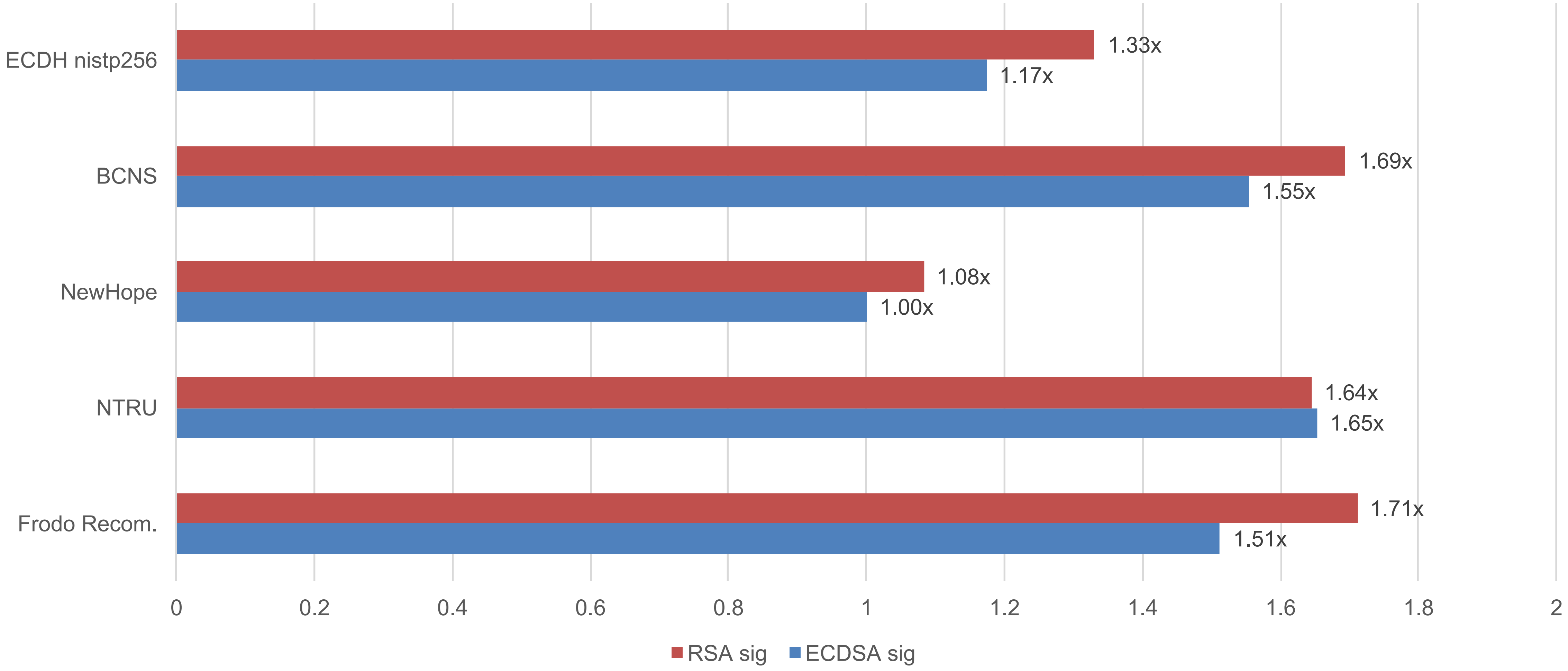
Connection throughput

- Number of connections per second at server before server latency spikes

TLS handshake latency

compared to NewHope-ECDSA

smaller (left) is better



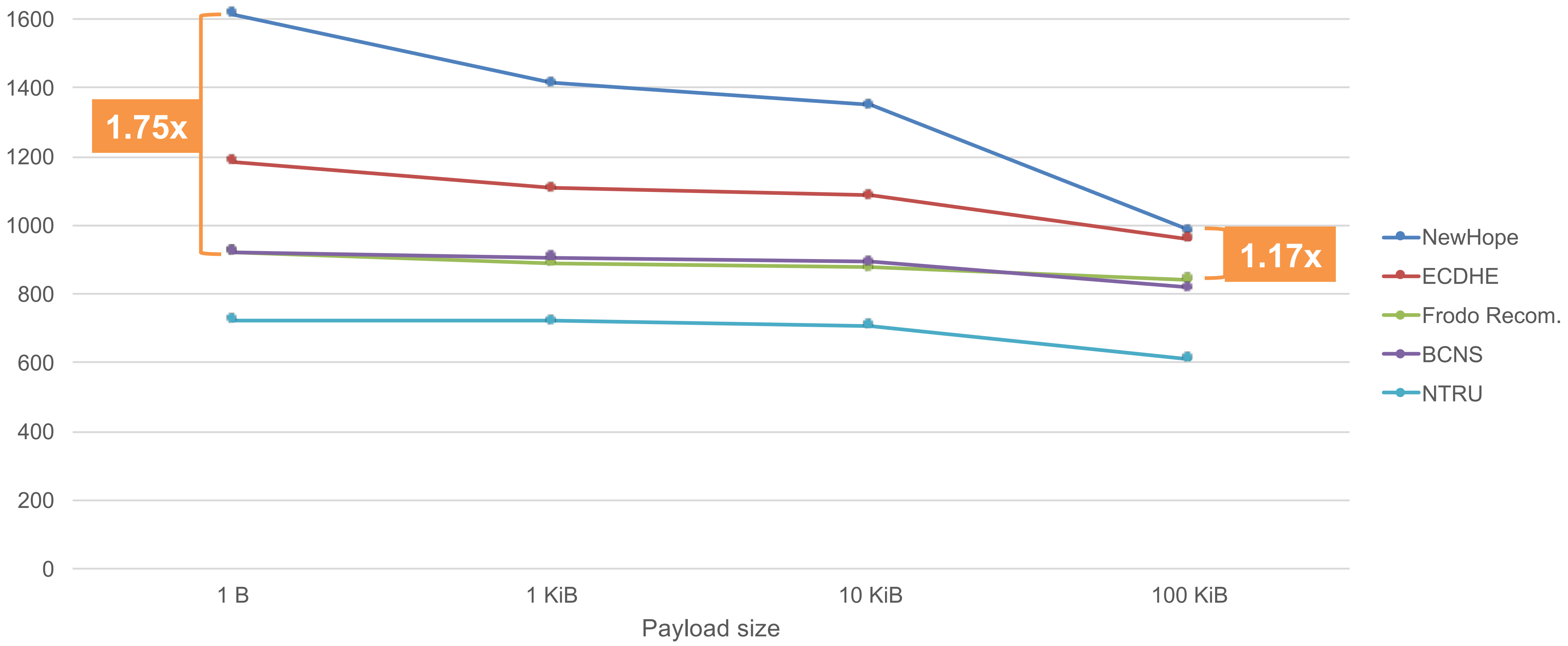
x86_64, 2.6 GHz Intel Xeon E5 (Sandy Bridge) – server Google `n1-standard-4`, client `-32`

Note somewhat incomparable security levels

TLS connection throughput

ECDSA signatures

bigger (top) is better



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

Note somewhat incomparable security levels

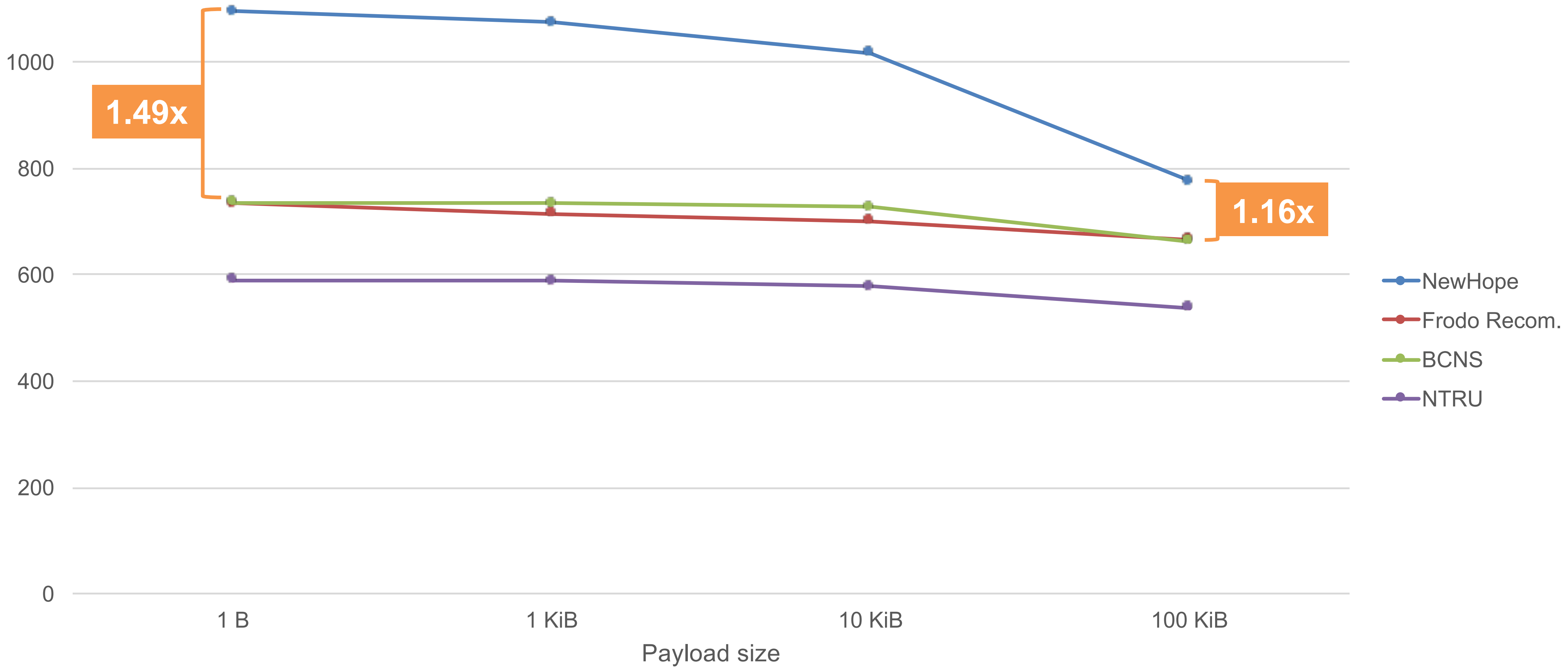
Hybrid ciphersuites

- 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

bigger (top) is better



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

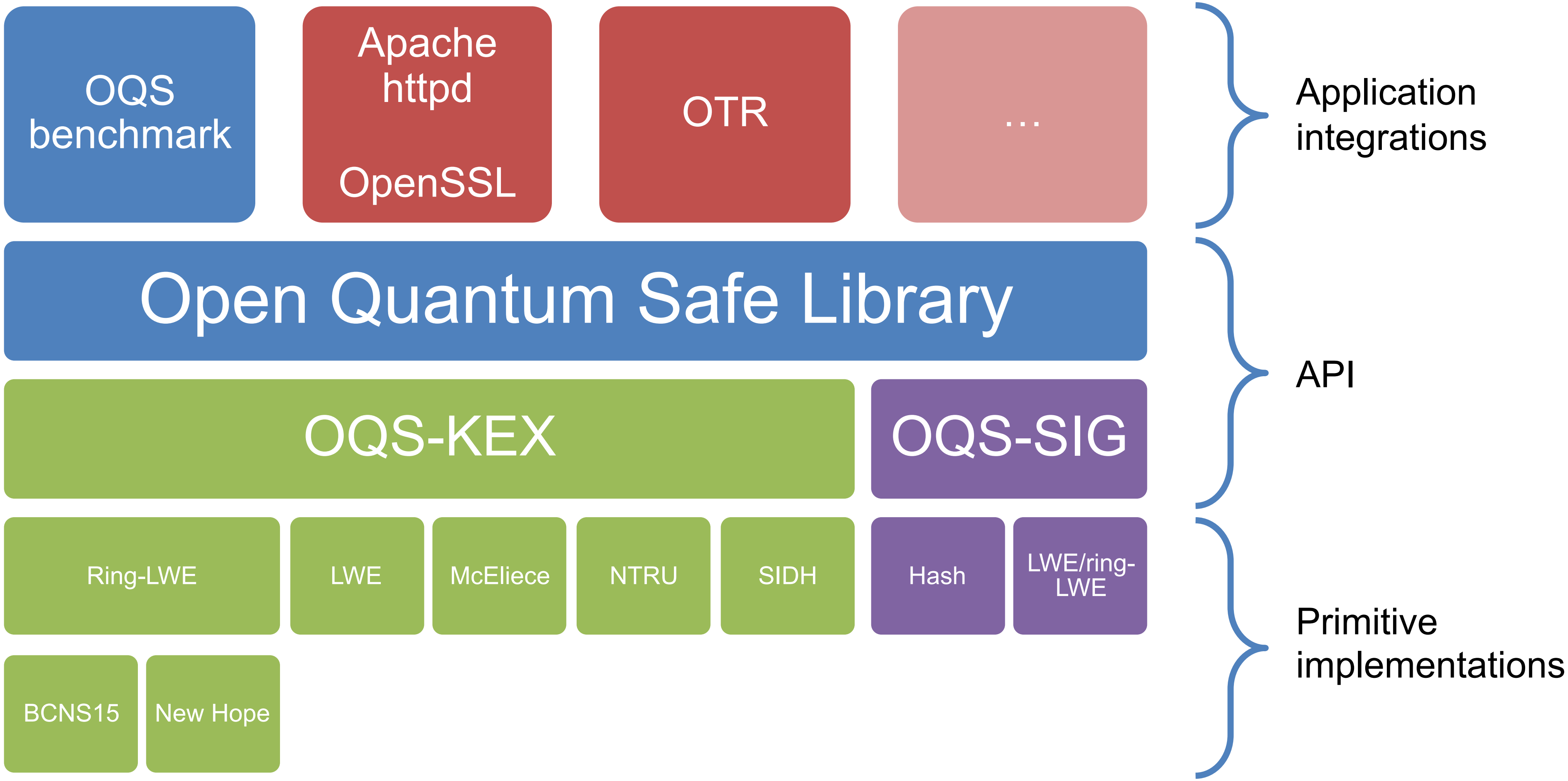
https://github.com/open-quantum-safe/

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



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!

McMaster University

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

- Deadline: August 25, 2016

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/