Practical post-quantum key exchange

Douglas Stebila McMaster University

Funding acknowledgements:

NSERC CRSNG

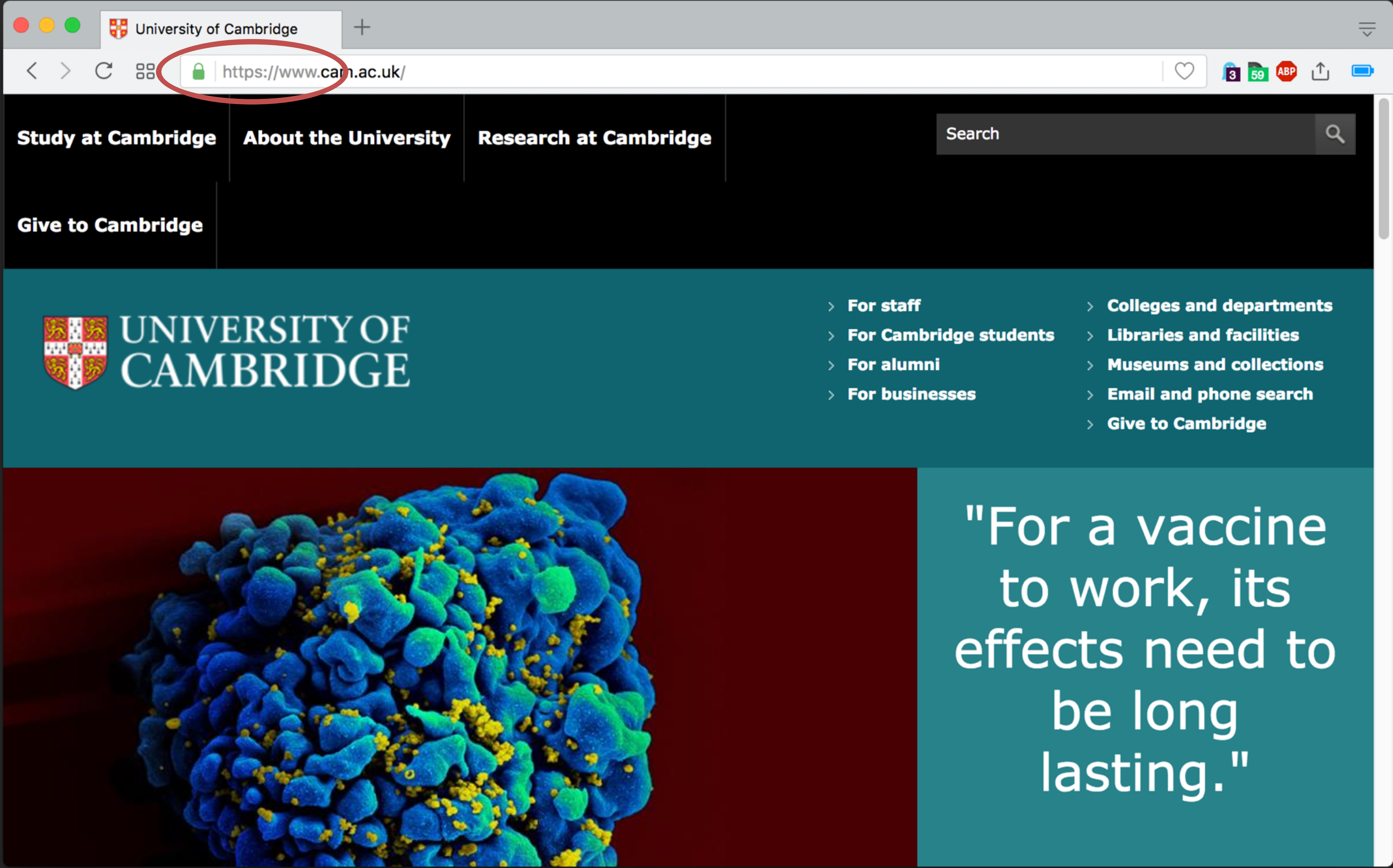
QCrypt 2017 • University of Cambridge • September 18, 2017

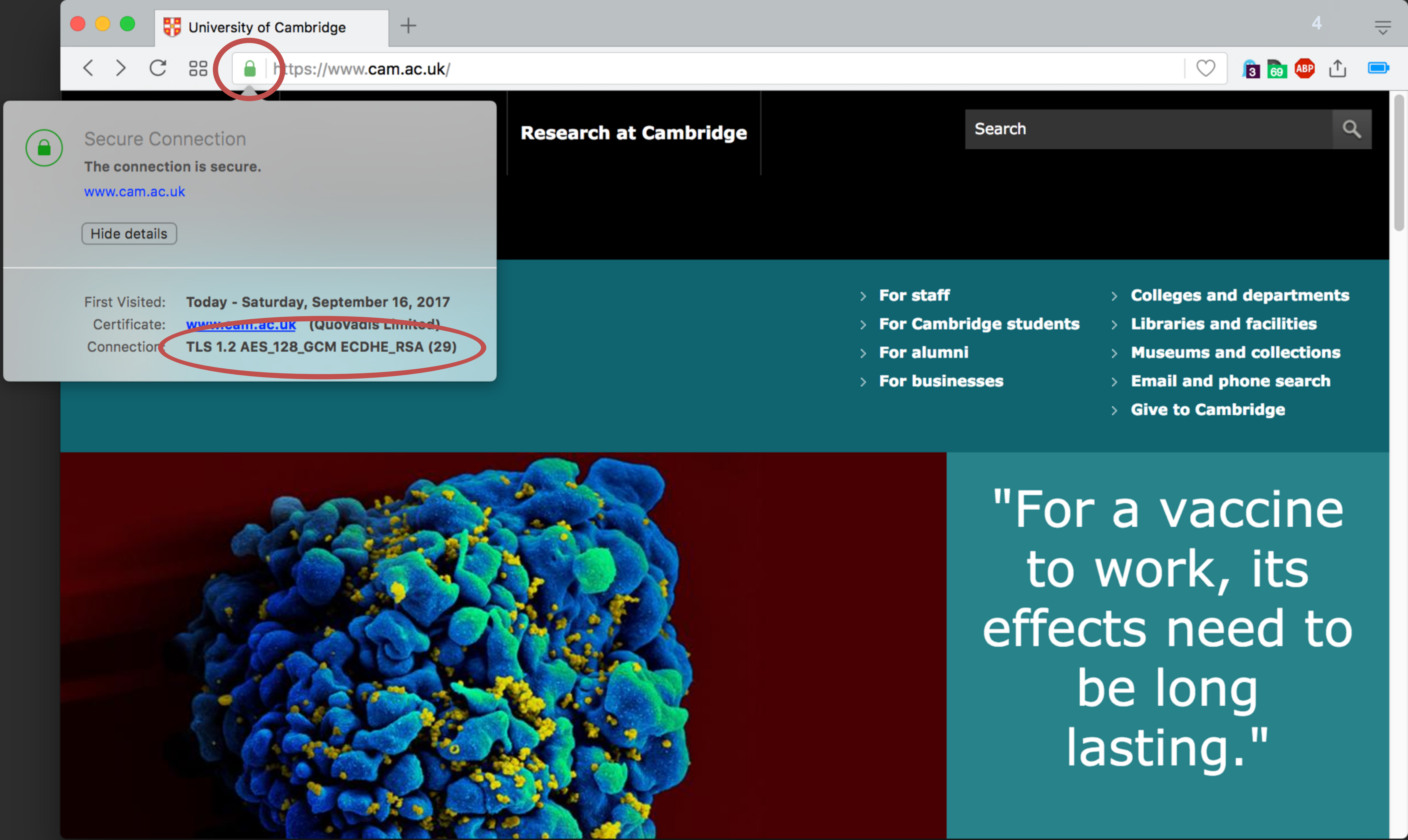
Key exchange on the Internet

SSL – Secure Sockets Layer protocol

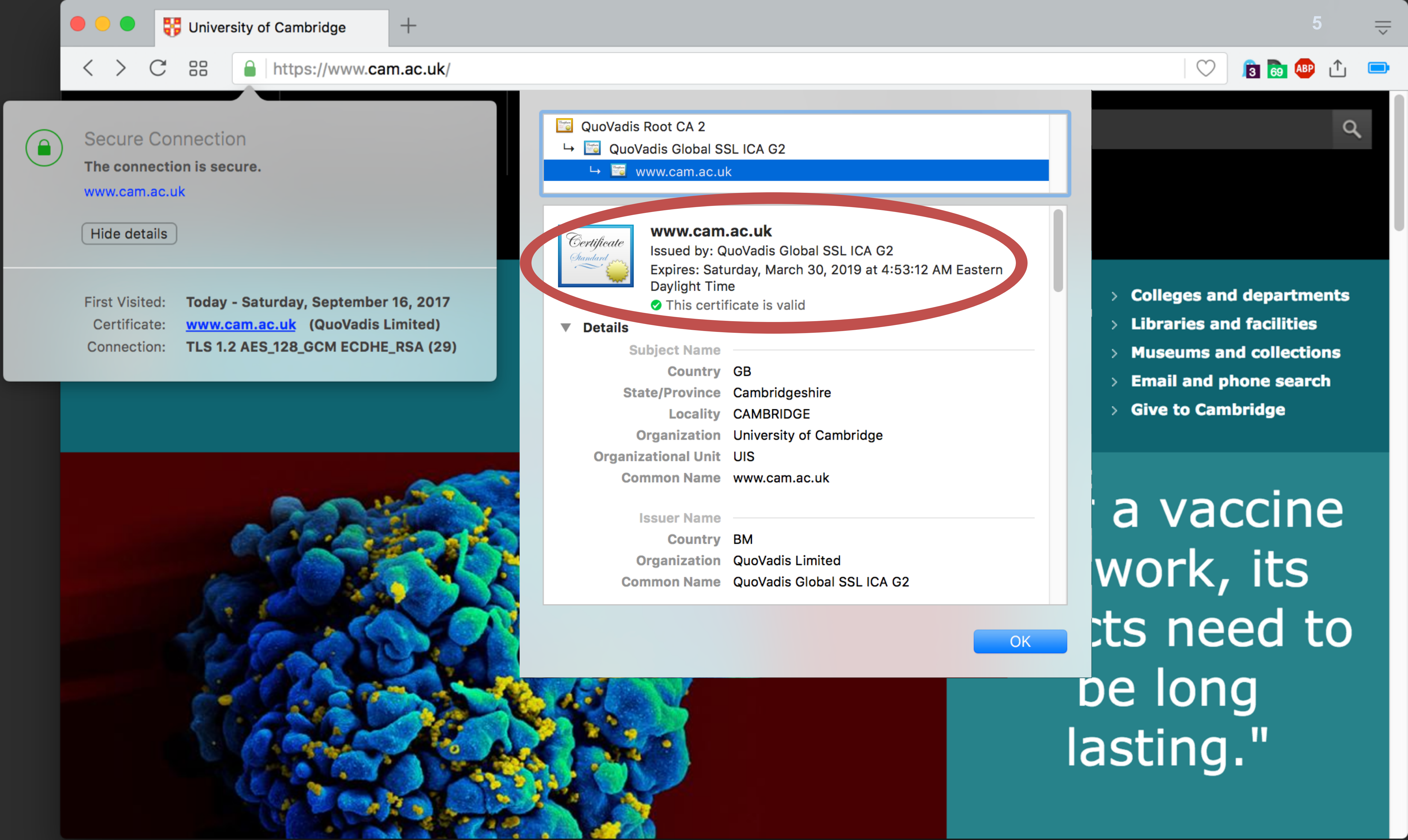
TLS – Transport Layer Security protocol

HTTPS – HTTP using SSL/TLS


University of Cambridge
https://www.cam.ac.uk/
Study at Cambridge
About the University
Research at Cambridge
Give to Cambridge
Search
UNIVERSITY OF CAMBRIDGE
For staff
For Cambridge students
For alumni
For businesses
Colleges and departments
Libraries and facilities
Museums and collections
Email and phone search
Give to Cambridge
"For a vaccine to work, its effects need to be long lasting."

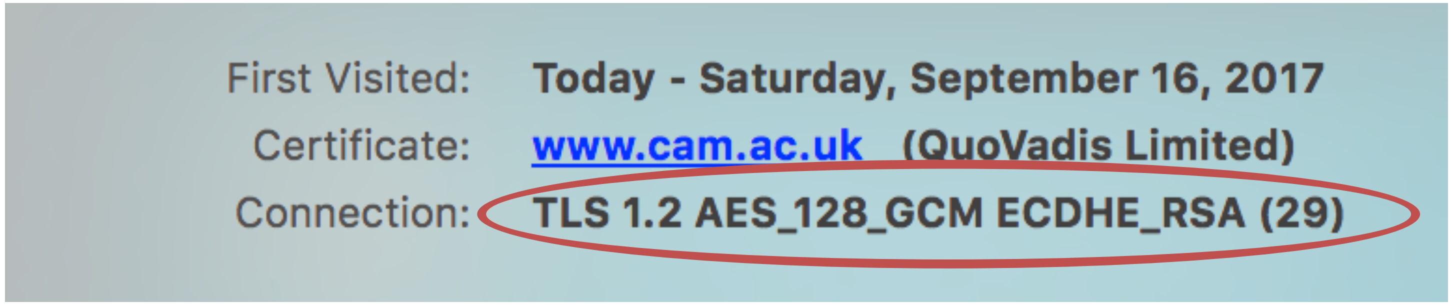


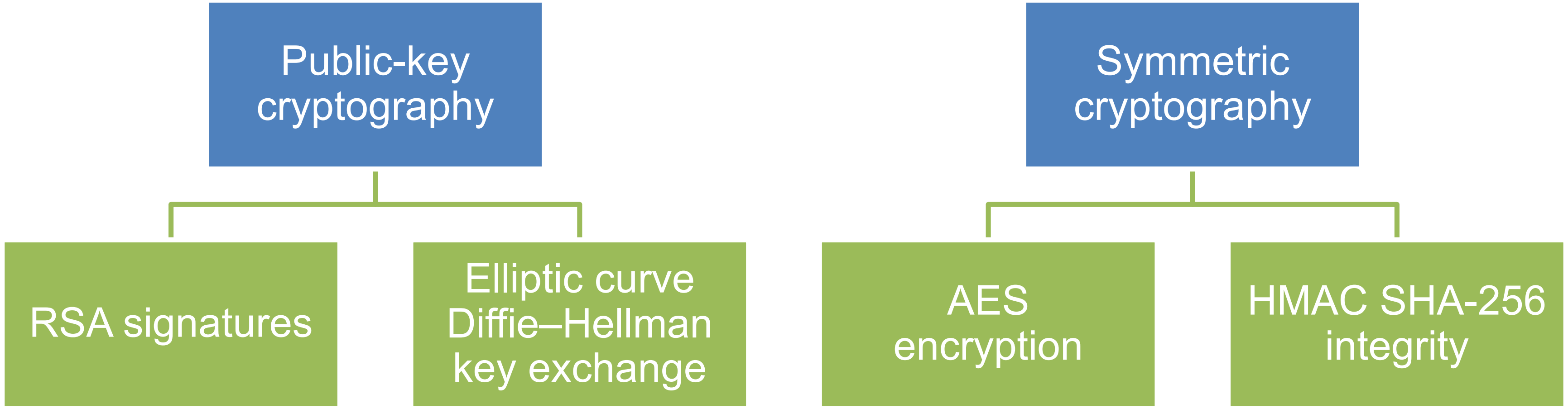
University of Cambridge
https://www.cam.ac.uk/
Secure Connection
The connection is secure.
www.cam.ac.uk
Hide details
First Visited: Today - Saturday, September 16, 2017
Certificate: www.cam.ac.uk (QuoVadis Limited)
Connection: TLS 1.2 AES_128_GCM ECDHE_RSA (29)
Research at Cambridge
Search
For staff
For Cambridge students
For alumni
For businesses
Colleges and departments
Libraries and facilities
Museums and collections
Email and phone search
Give to Cambridge
"For a vaccine to work, its effects need to be long lasting."



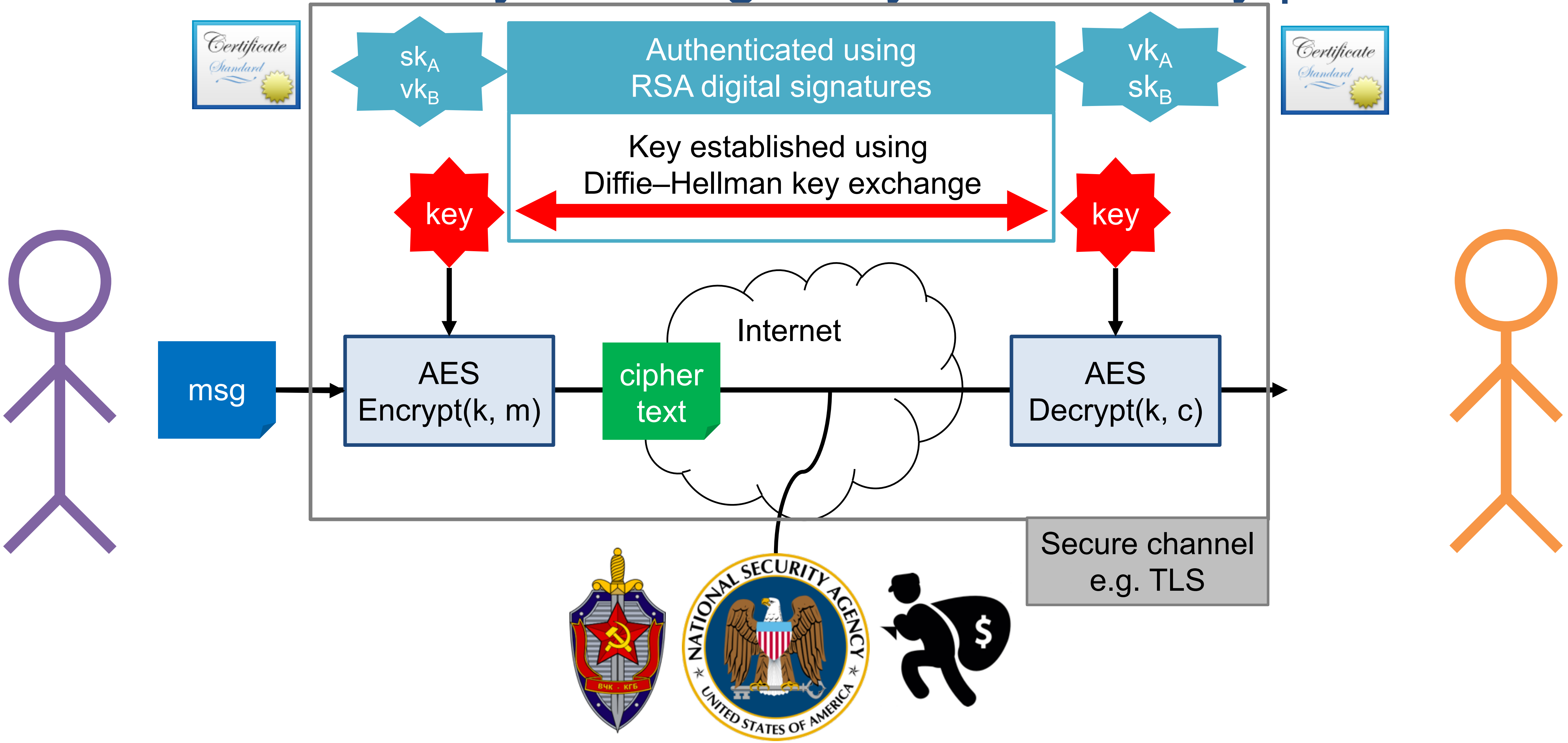
University of Cambridge
https://www.cam.ac.uk/
Secure Connection
The connection is secure.
www.cam.ac.uk
Hide details
First Visited: Today - Saturday, September 16, 2017
Certificate: www.cam.ac.uk (QuoVadis Limited)
Connection: TLS 1.2 AES_128_GCM ECDHE_RSA (29)
QuoVadis Root CA 2
QuoVadis Global SSL ICA G2
www.cam.ac.uk
www.cam.ac.uk
Issued by: QuoVadis Global SSL ICA G2
Expires: Saturday, March 30, 2019 at 4:53:12 AM Eastern Daylight Time
This certificate is valid
Details
Subject Name
Country GB
State/Province Cambridgeshire
Locality CAMBRIDGE
Organization University of Cambridge
Organizational Unit UIS
Common Name www.cam.ac.uk
Issuer Name
Country BM
Organization QuoVadis Limited
Common Name QuoVadis Global SSL ICA G2
OK
Colleges and departments
Libraries and facilities
Museums and collections
Email and phone search
Give to Cambridge
a vaccine
work, its
cts need to
be long
lasting."

Cryptographic building blocks

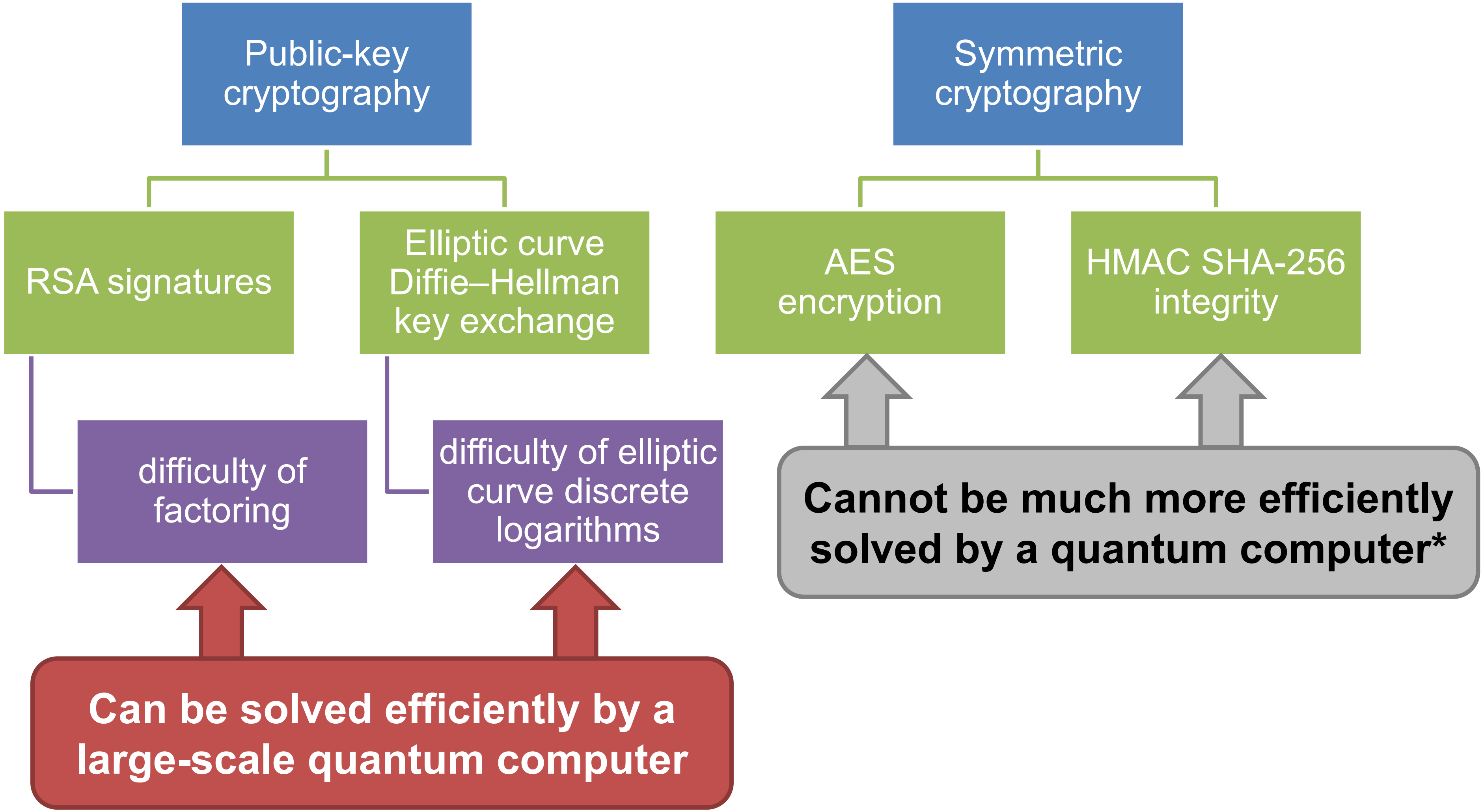




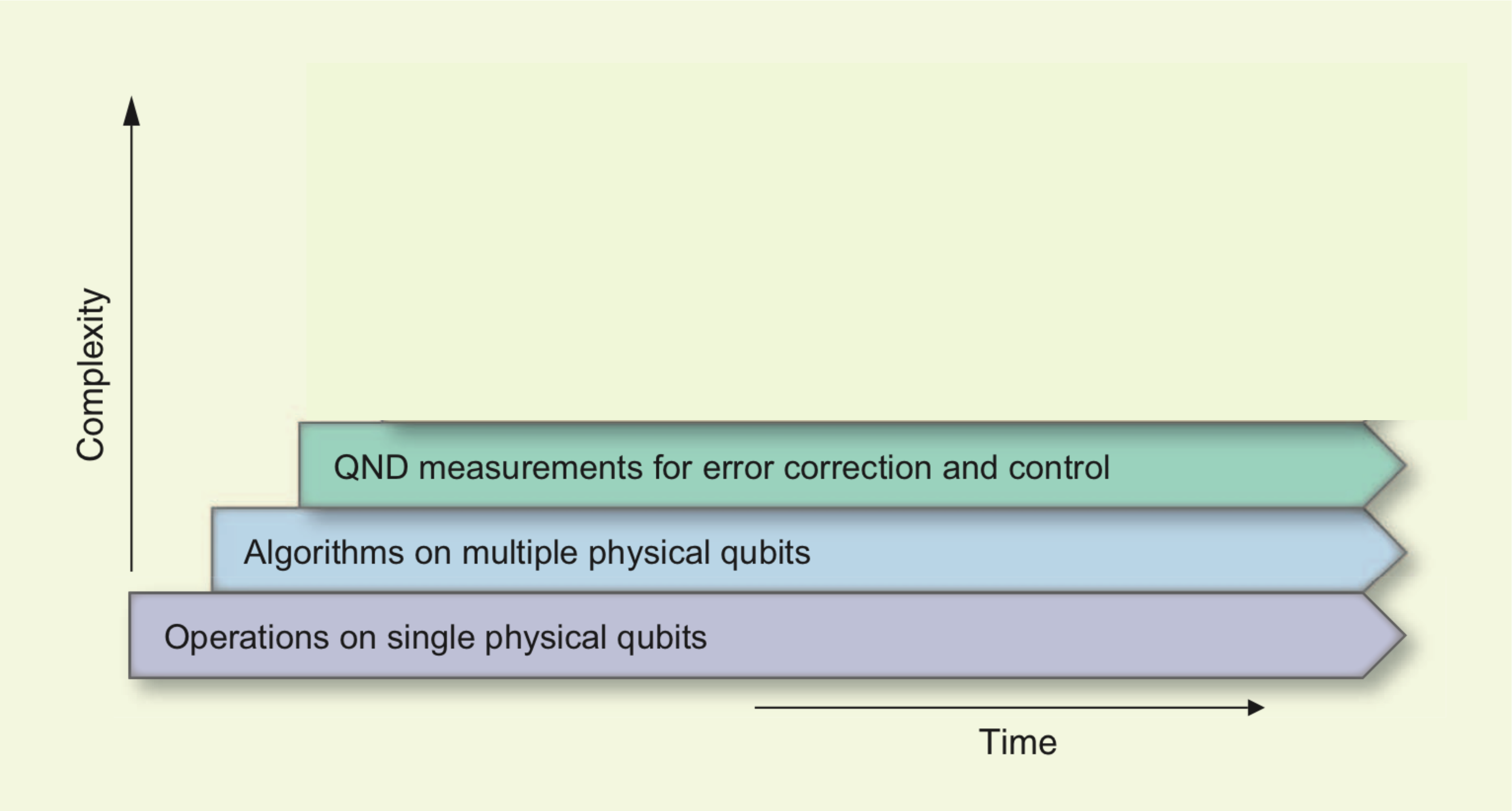
Authenticated key exchange + symmetric encryption



Cryptographic building blocks

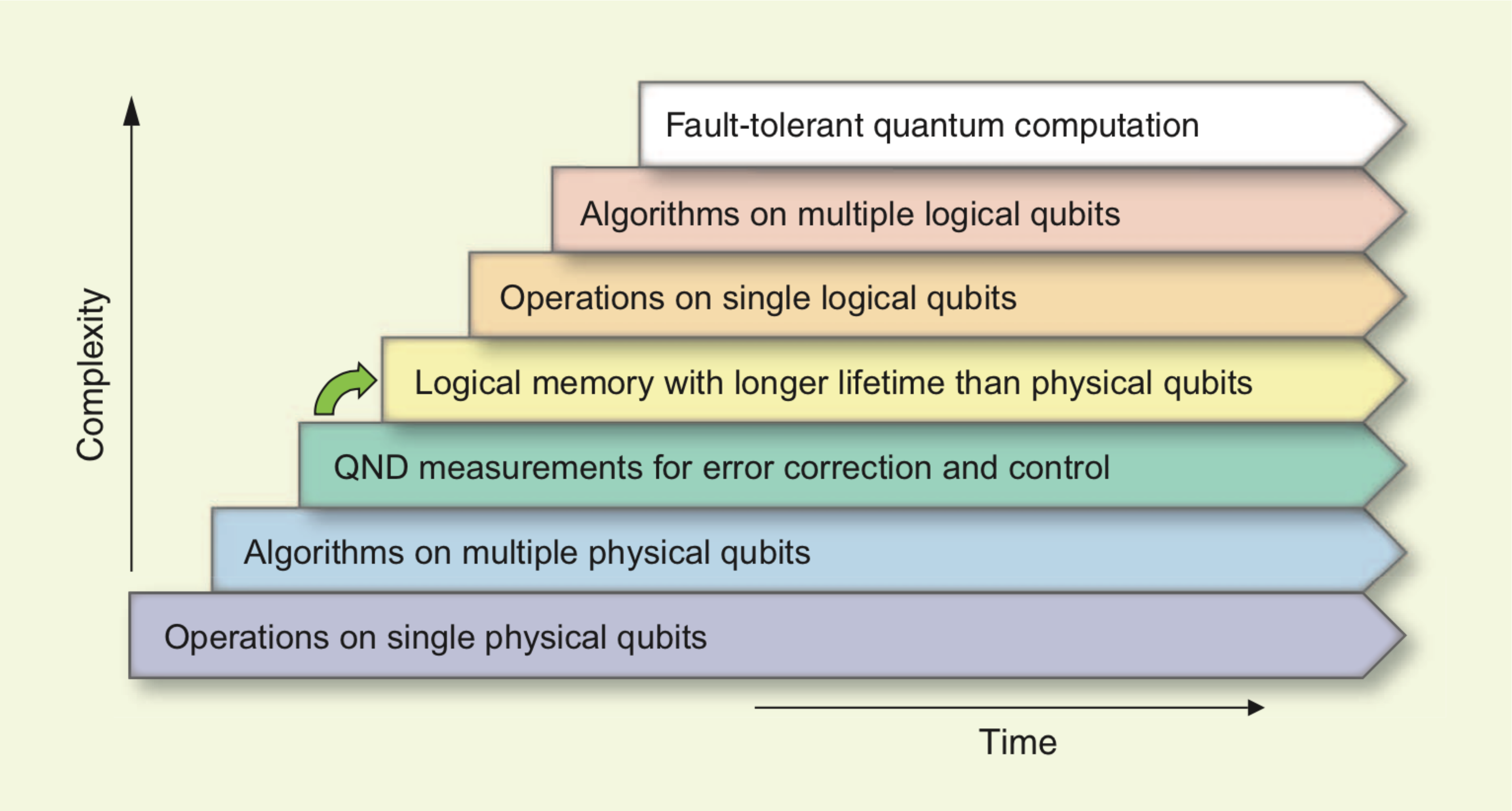


When will a large-scale quantum computer be built?



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

When will a large-scale quantum computer be built?



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

When will a large-scale quantum computer be built?

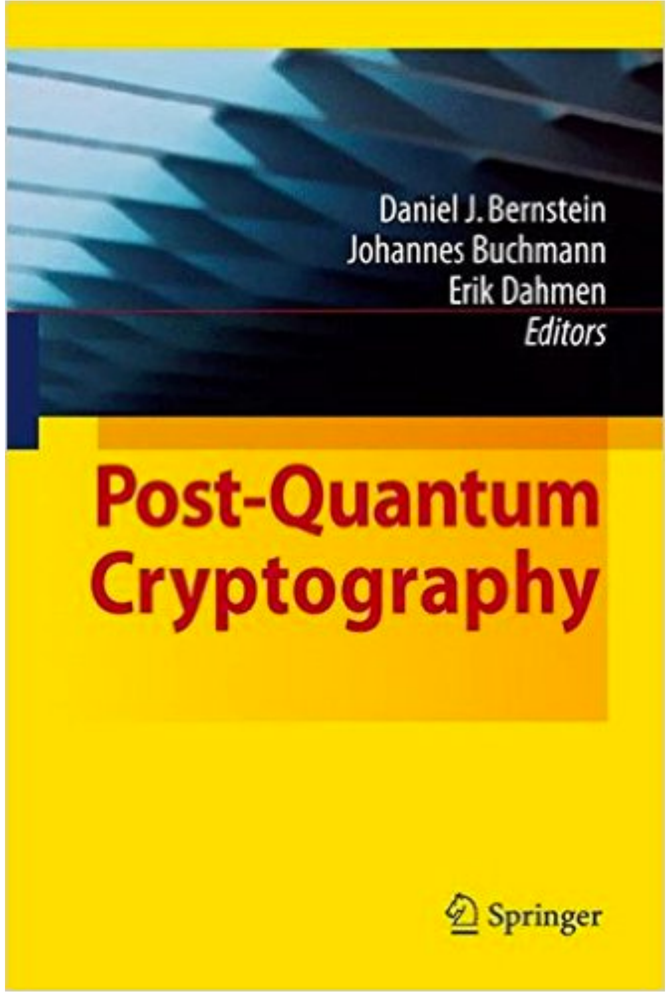
"I estimate a 1/7 chance of breaking RSA-2048 by 2026 and a 1/2 chance by 2031."

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

Post-quantum cryptography in academia

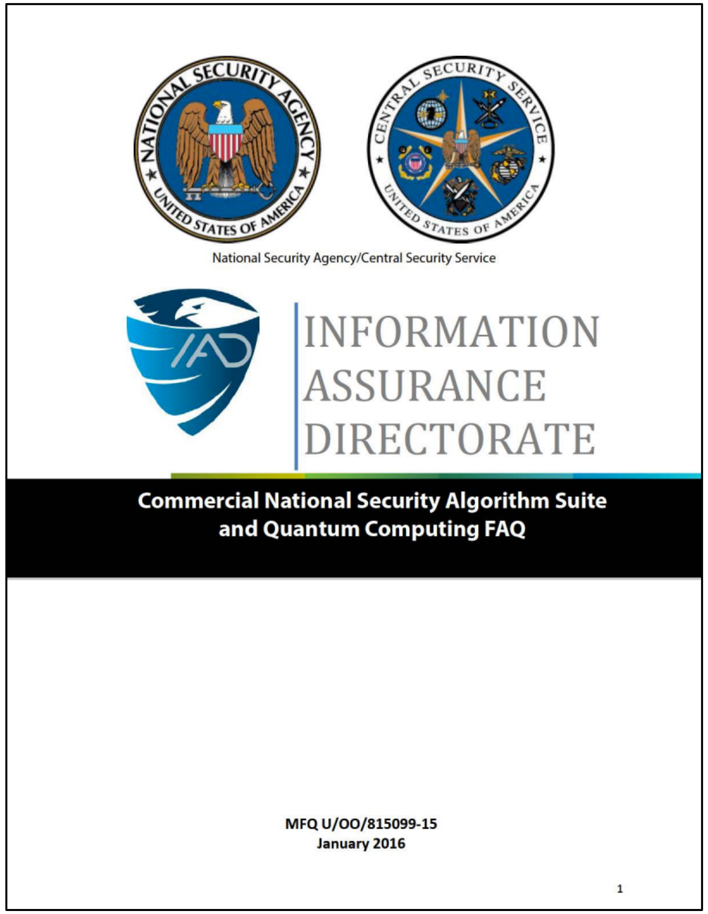
Conference series

- PQCrypto 2006
- PQCrypto 2008
- PQCrypto 2010
- PQCrypto 2011
- PQCrypto 2013
- PQCrypto 2014
- PQCrypto 2016
- PQCrypto 2017
- PQCrypto 2018



2009

Post-quantum cryptography in government


National Security Agency/Central Security Service

INFORMATION ASSURANCE DIRECTORATE

Commercial National Security Algorithm Suite and Quantum Computing FAQ

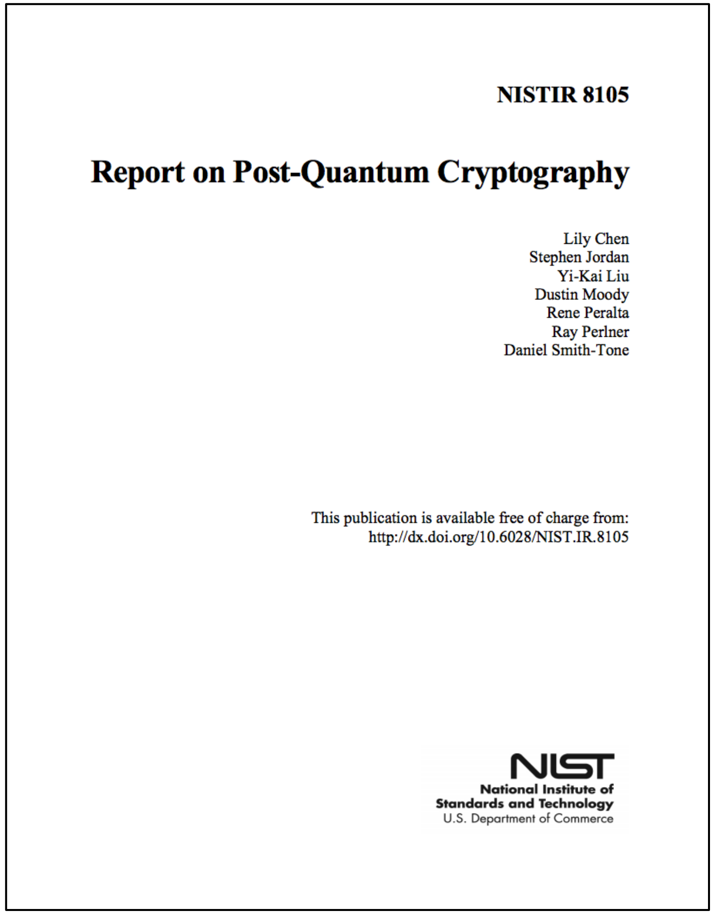
MFQ U/OO/815099-15
January 2016

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

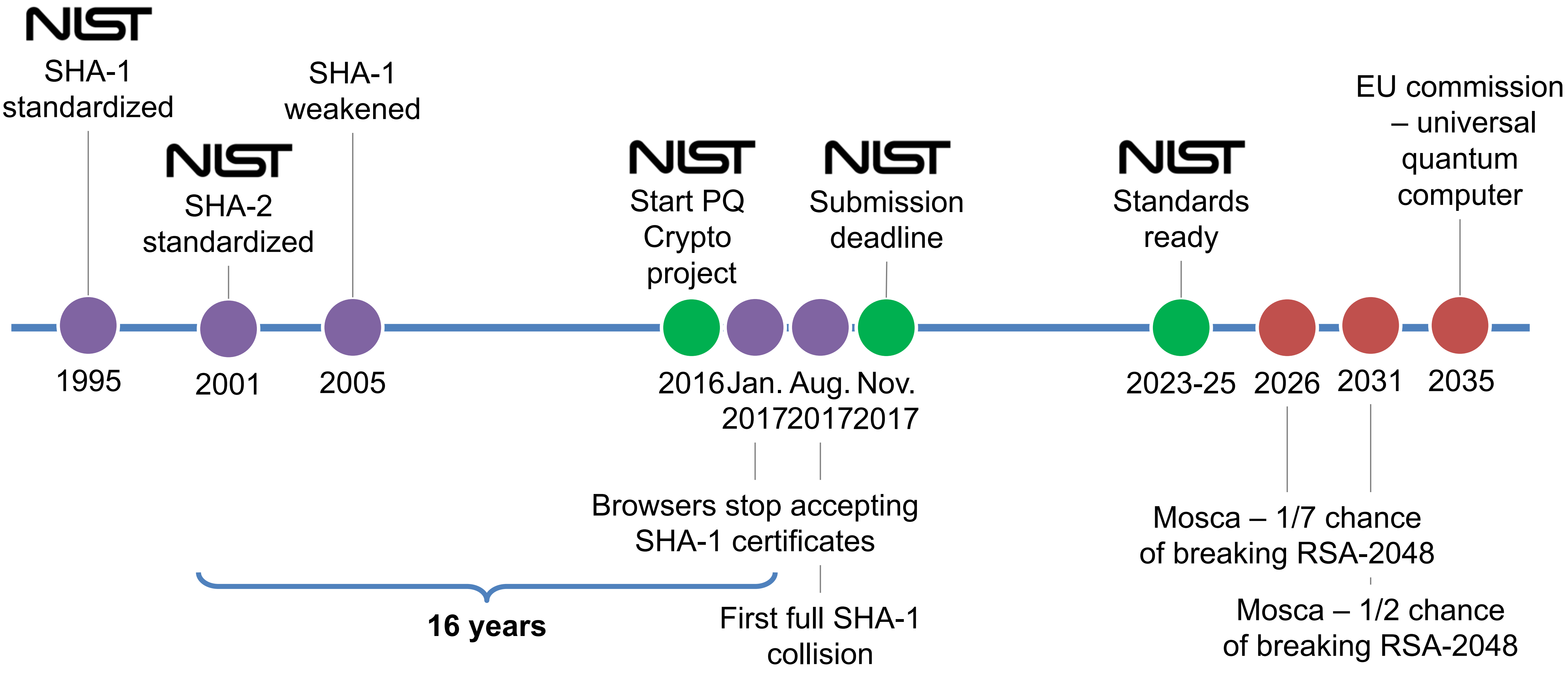
http://www.nist.gov/pqcrypto

December 2016	Formal call for proposals
November 2017	**Deadline for submissions**
3-5 years	Analysis phase
2 years later (2023-2025)	Draft standards ready

"**Our intention is to select a couple of options** for more immediate standardization, as well as to eliminate some submissions as unsuitable. … The goal of the process is **not primarily to pick a winner**, but to document the strengths and weaknesses of the different options, and to analyze the possible tradeoffs among them."

http://csrc.nist.gov/groups/ST/post-quantum-crypto/faq.html#Q7

Timeline


SHA-1 standardized
1995
SHA-2 standardized
2001
SHA-1 weakened
2005
Start PQ Crypto project
2016
Jan. 2017
Browsers stop accepting SHA-1 certificates
Aug. 2017
First full SHA-1 collision
Submission deadline
Nov. 2017
16 years
Standards ready
2023-25
2026
Mosca – 1/7 chance of breaking RSA-2048
2031
Mosca – 1/2 chance of breaking RSA-2048
EU commission – universal quantum computer
2035

Post-quantum crypto

Post-quantum crypto

a.k.a. quantum-resistant algorithms

Classical crypto with 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

Quantum-safe crypto

Classical post-quantum crypto

Hash-based
- Merkle signatures
- Sphincs

Code-based
- McEliece
- Niederreiter

Multivariate
- multivariate quadratic

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

Isogenies
- supersingular elliptic curve isogenies

Quantum crypto

Quantum key distribution

Quantum random number generators

Quantum channels

Quantum blind computation

Post-quantum crypto research agenda

- Design better post-quantum schemes
- Characterize classical and quantum attacks
- Pick parameter sizes
- Develop fast, secure implementations
- Integrate them into the existing infrastructure

This talk

- The **Learning with errors** problem
  - "Lattice-based"
  - Public key encryption
  - Key exchange
- Transitioning to post-quantum crypto
- Open Quantum Safe project
  - A library for comparing post-quantum primitives
  - Framework for easing integration into applications like OpenSSL

Learning with errors problems

Solving systems of linear equations

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

Linear system problem: given blue, find red

Solving systems of linear equations

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

Easily solved using Gaussian elimination (Linear Algebra 101)

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

Learning with errors problem

$$
\underset{\mathbb{Z}_{13}^{7\times 4}}{\textbf{random}}
\qquad
\underset{\mathbb{Z}_{13}^{4\times 1}}{\textbf{secret}}
\qquad
\underset{\mathbb{Z}_{13}^{7\times 1}}{\textbf{small noise}}
\qquad
\mathbb{Z}_{13}^{7\times 1}
$$

$$
\begin{bmatrix} 4 & 1 & 11 & 10 \\ 5 & 5 & 9 & 5 \\ 3 & 9 & 0 & 10 \\ 1 & 3 & 3 & 2 \\ 12 & 7 & 3 & 4 \\ 6 & 5 & 11 & 4 \\ 3 & 3 & 5 & 0 \end{bmatrix}
\times
\begin{bmatrix} 6 \\ 9 \\ 11 \\ 11 \end{bmatrix}
+
\begin{bmatrix} 0 \\ -1 \\ 1 \\ 1 \\ 1 \\ 0 \\ -1 \end{bmatrix}
=
\begin{bmatrix} 4 \\ 7 \\ 2 \\ 11 \\ 5 \\ 12 \\ 8 \end{bmatrix}
$$

Learning with errors problem

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

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

Decision learning with errors problem

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

Decision LWE problem: given blue, distinguish green from random

Search LWE problem

Let n, m, and q be positive integers.

Let χ_s and χ_e be distributions over $\mathbb{Z}$.

Sample $\mathbf{s} \xleftarrow{\$} \chi_s^n$.

For $i = 1, \ldots, m$:

- Sample $\mathbf{a}_i \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^n)$, $e_i \xleftarrow{\$} \chi_e$.
- Set $b_i \leftarrow \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \mod q$.

The **search LWE problem** is:

- given $(\mathbf{a}_i, b_i)_{i=1}^m$,
- find $\mathbf{s}$

Decision LWE problem

Let n and q be positive integers.

Let χ_s and χ_e be distributions over $\mathbb{Z}$.

Sample $\mathbf{s} \xleftarrow{\$} \chi_s^n$.

Define the following two oracles:

- $O_{\chi_e,\mathbf{s}}$: $\mathbf{a} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^n)$, $e \xleftarrow{\$} \chi_e$; return $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q)$.
- U: $\mathbf{a} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^n)$, $u \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q)$; return $(\mathbf{a}, u)$.

The **decision LWE problem** is: distinguish $O_{\chi,\mathbf{s}}$ from U.

Choice of error distribution

- Usually a discrete Gaussian distribution of width $s = \alpha q$ for error rate $\alpha < 1$

- Define the Gaussian function

$$\rho_s(\mathbf{x}) = \exp(-\pi \|\mathbf{x}\|^2 / s^2)$$

- The continuous Gaussian distribution has probability density function

$$f(\mathbf{x}) = \rho_s(\mathbf{x}) / \int_{\mathbb{R}^n} \rho_s(\mathbf{z}) d\mathbf{z} = \rho_s(\mathbf{x}) / s^n$$

Short secrets

- The secret distribution χ_s was originally taken to be the uniform distribution
- **Short secrets**: use $\chi_s = \chi_e$
- There's a tight reduction showing that LWE with short secrets is hard if LWE with uniform secrets is hard

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

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

$$\begin{array}{c c}
 & \overbrace{\begin{array}{ccccc} & & & & \end{array}}^{8} \\
752 \left\{ \right. & \begin{array}{ccccc}
2738 & 3842 & 3345 & 2979 & \dots \\
2896 & 595 & 3607 & & \\
377 & 1575 & & & \\
2760 & & & & \\
\dots & & & &
\end{array}
\end{array}$$

752 × 8 × 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**
=	$10 + 5x + 10x^2 + 7x^3$	

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

Problems

Search LWE problem	Decision LWE problem
Search ring-LWE problem	Decision ring-LWE problem

with or without short secrets

Search-decision equivalence

- **Easy fact**: If the search LWE problem is easy, then the decision LWE problem is easy.

- **Fact**: If the decision LWE problem is easy, then the search LWE problem is easy.
  - Requires nq calls to decision oracle
  - Intuition: test the each value for the first component of the secret, then move on to the next one, and so on.

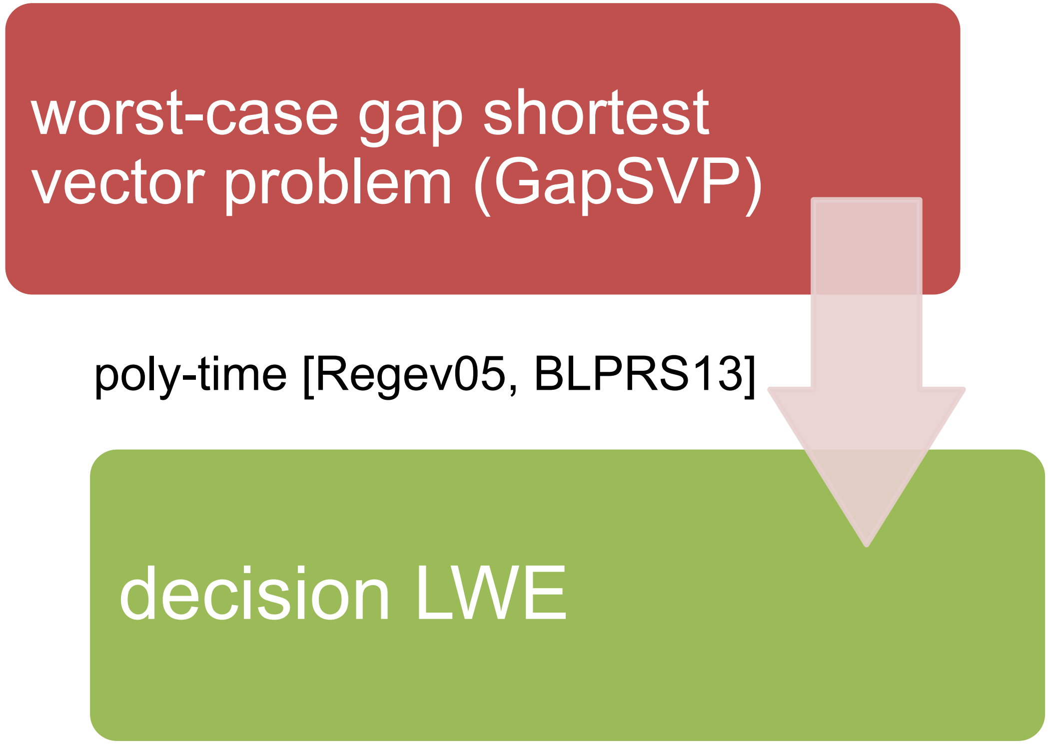
NTRU problem

For an invertible $s \in R_q^*$ and a distribution χ on R, define $N_{s,\chi}$ to be the distribution that outputs $e/s \in R_q$ where $e \stackrel{\$}{\leftarrow} \chi$.

The **NTRU learning problem** is: given independent samples $a_i \in R_q$ where every sample is distributed according to either: (1) $N_{s,\chi}$ for some randomly chosen $s \in R_q$ (fixed for all samples), or (2) the uniform distribution, distinguish which is the case.

"Lattice-based"

Hardness of decision LWE – "lattice-based"



Lattices

Let $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_n\} \subseteq \mathbb{Z}_q^{n \times n}$ be a set of linearly independent basis vectors for $\mathbb{Z}_q^n$. Define the corresponding **lattice**

$$\mathcal{L} = \mathcal{L}(\mathbf{B}) = \left\{ \sum_{i=1}^{n} z_i \mathbf{b}_i : z_i \in \mathbb{Z} \right\} \quad .$$

(In other words, a lattice is a set of *integer* linear combinations.)

Define the **minimum distance** of a lattice as

$$\lambda_1(\mathcal{L}) = \min_{\mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|\mathbf{v}\| \quad .$$

Shortest vector problem

The **shortest vector problem** (SVP) is: given a basis $\mathbf{B}$ for some lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$, find a shortest non-zero vector, i.e., find $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\| = \lambda_1(\mathcal{L})$.

The **decision approximate shortest vector problem** (GapSVP_γ) is: given a basis $\mathbf{B}$ for some lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$ where either $\lambda_1(\mathcal{L}) \leq 1$ or $\lambda_1(\mathcal{L}) > \gamma$, determine which is the case.

Regev's iterative reduction

Theorem. **[Reg05]** For any modulus $q \leq 2^{\text{poly}(n)}$ and any discretized Gaussian error distribution χ of parameter $\alpha q \geq 2\sqrt{n}$ where $0 < \alpha < 1$, solving the decision LWE problem for $(n, q, \mathcal{U}, \chi)$ with at most $m = \text{poly}(n)$ samples is at least as hard as quantumly solving GapSVP_γ and SIVP_γ on arbitrary n-dimensional lattices for some $\gamma = \tilde{O}(n/\alpha)$.

The polynomial-time reduction is extremely non-tight: approximately $O(n^{13})$.

[Regev; STOC 2005]

Solving the (approximate) shortest vector problem

The complexity of GapSVP_γ depends heavily on how γ and n relate, and get harder for smaller γ.

Algorithm	Time	Approx. factor γ
LLL algorithm	$\mathrm{poly}(n)$	$2^{\Omega(n \log \log n / \log n)}$
various	$2^{\Omega(n \log n)}$	$\mathrm{poly}(n)$
various	$2^{\Omega(n)}$ time and space	$\mathrm{poly}(n)$
Sch87	$2^{\tilde{\Omega}(n/k)}$	2^k
	NP $\cap$ co-NP	$\geq \sqrt{n}$
	NP-hard	$n^{o(1)}$

In cryptography, we tend to use $\gamma \approx n$.

Picking parameters

- Estimate parameters based on runtime of lattice reduction algorithms.

- Based on reductions:
  - Calculate required runtime for GapSVP or SVP based on tightness gaps and constraints in each reduction
  - Pick parameters based on best known GapSVP or SVP solvers or known lower bounds
- Based on cryptanalysis:
  - Ignore tightness in reductions.
  - Pick parameters based on best known LWE solvers relying on lattice solvers.

Ring-LWE

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

4	1	11	10

Cyclic structure

$\Rightarrow$ Save communication,
more efficient computation

4 KiB representation

LWE

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

8

752

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

752 × 8 × 15 bits = **11 KiB**

Why consider (slower, bigger) LWE?

Generic vs. ideal lattices

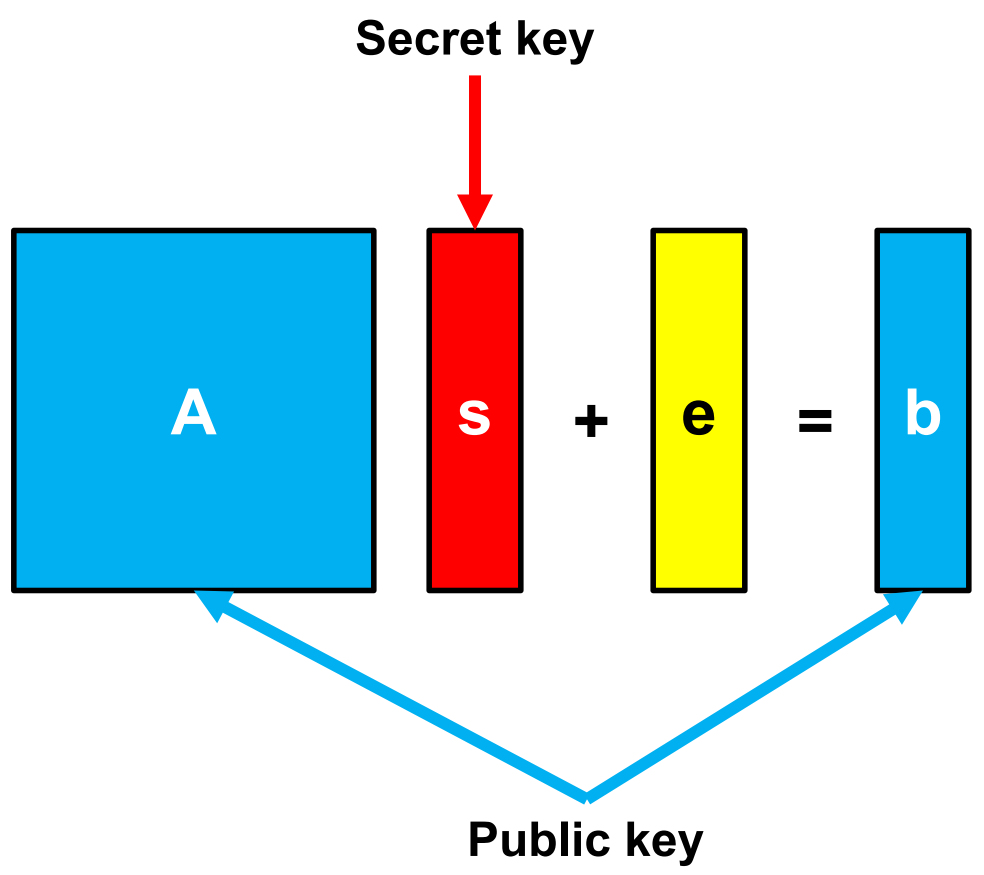
- Ring-LWE matrices have additional structure
  - Relies on hardness of a problem in **ideal** lattices
- LWE matrices have no additional structure
  - Relies on hardness of a problem in **generic** lattices
- NTRU also relies on a problem in a type of ideal lattices

- Currently, best algorithms for ideal lattice problems are essentially the same as for generic lattices
  - Small constant factor improvement in some cases
  - Recent quantum polynomial time algorithm for Ideal-SVP (http://eprint.iacr.org/2016/885) but not immediately applicable to ring-LWE

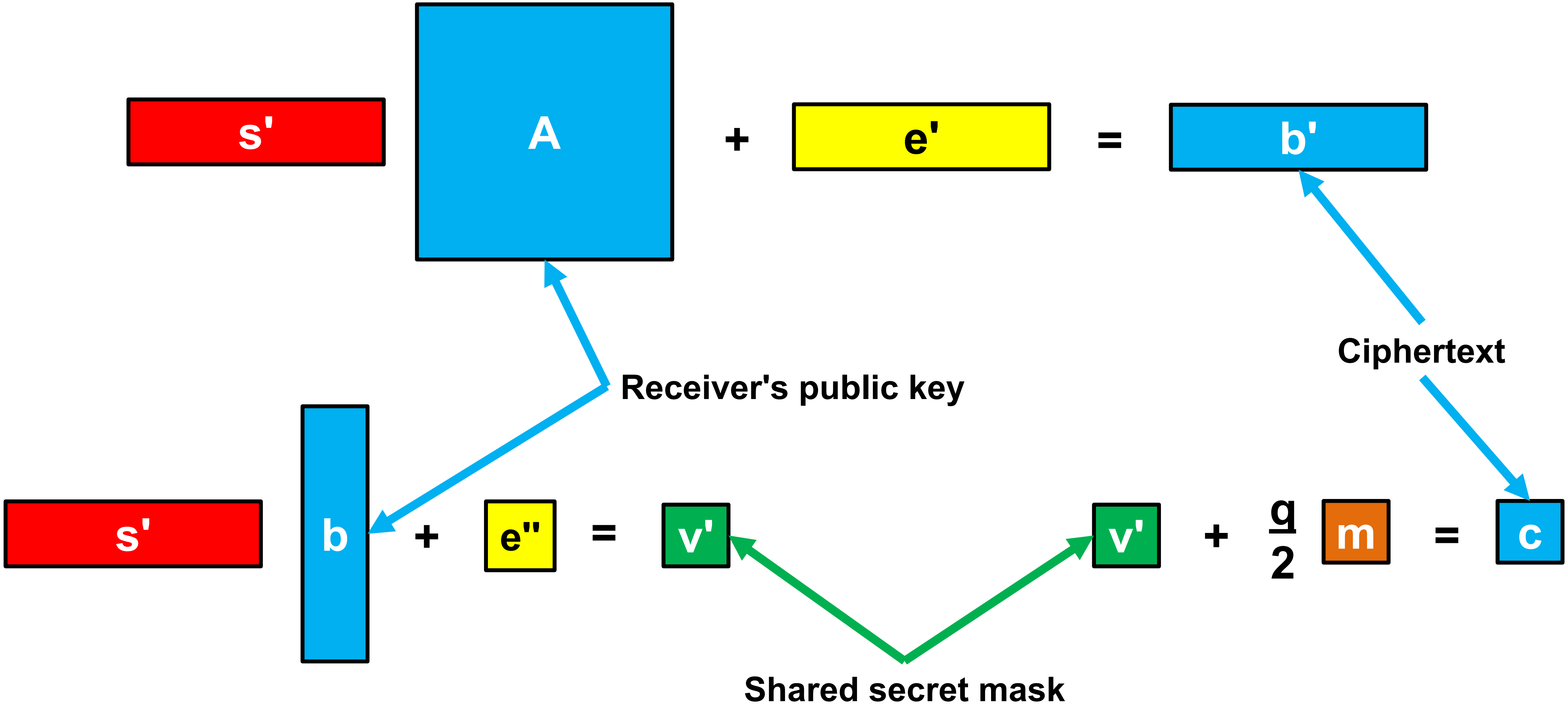
Public key encryption from LWE

Lindner–Peikert, CT-RSA 2011

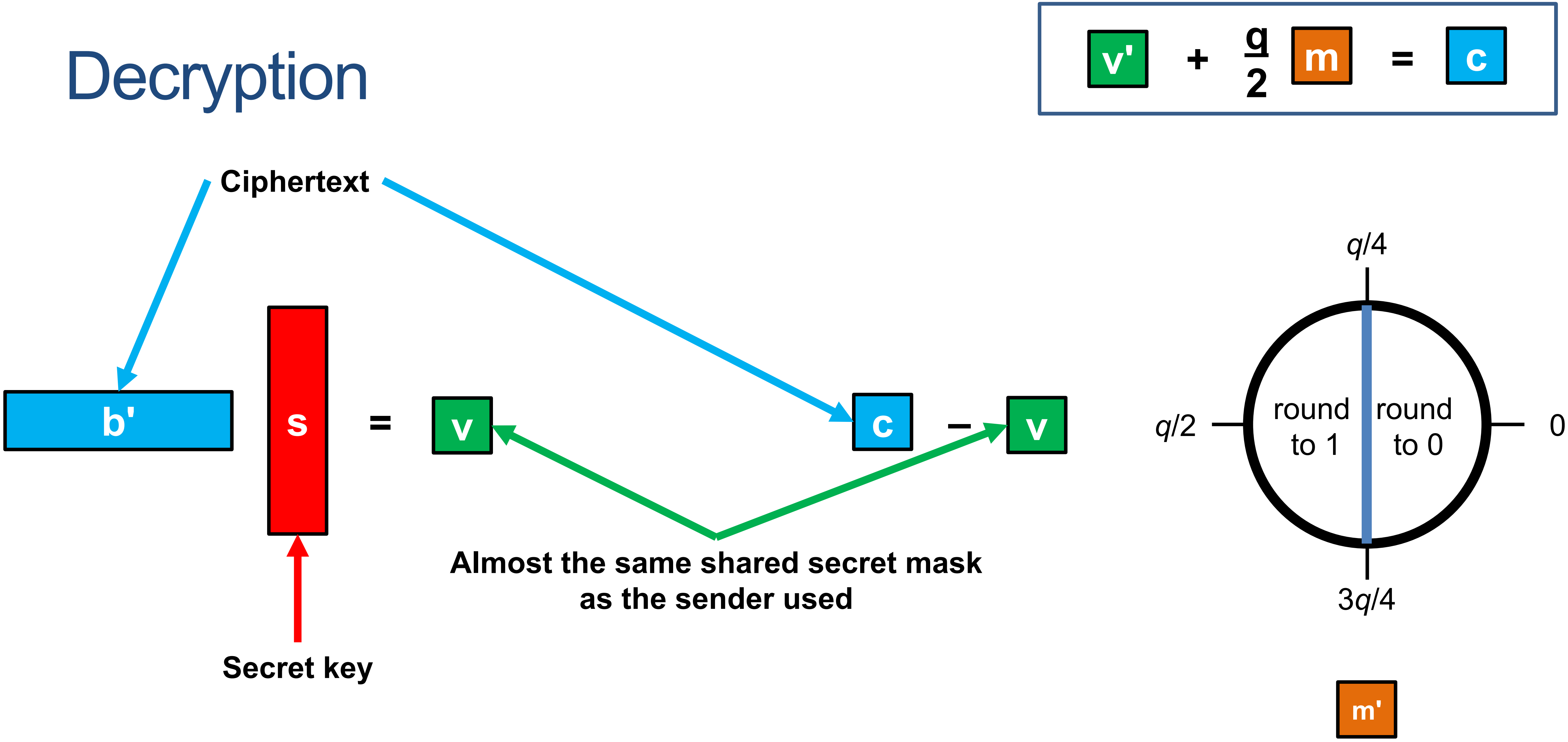
Key generation



Encryption


s'
A
+
e'
=
b'
Receiver's public key
Ciphertext
s'
b
+
e''
=
v'
v'
+
q
2
m
=
c
Shared secret mask

Decryption



v'
+
q
2
m
=
c
Ciphertext
b'
s
=
v
c
–
v
Almost the same shared secret mask
as the sender used
Secret key
q/4
q/2
0
3q/4
round
to 1
round
to 0
m'

Approximately equal shared secret

The sender uses

$$s' (A s + e) + e''$$

$$= s' A s + (s' e + e'')$$

$$\approx s' A s$$

The receiver uses

$$(s' A + e') s$$

$$= s' A s + (e' s)$$

$$\approx s' A s$$

IND-CPA security of Lindner–Peikert

Indistinguishable against chosen plaintext attacks

Theorem. If the decision LWE problem is hard, then Lindner–Peikert is IND-CPA-secure. Let n, q, χ be LWE parameters. Let $\mathcal{A}$ be an algorithm. Then there exist algorithms $\mathcal{B}_1, \mathcal{B}_2$ such that

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathbf{LP}[n,q,\chi]}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{dlwe}}_{n,q,\chi}(\mathcal{A} \circ \mathcal{B}_1) + \mathsf{Adv}^{\mathsf{dlwe}}_{n,q,\chi}(\mathcal{A} \circ \mathcal{B}_2)$$

[Lindner, Peikert; CT-RSA 2011]

Public key validation

- **No public key validation possible** in IND-CPA KEMs/PKEs from LWE/ring-LWE

- **Key reuse in LWE/ring-LWE** leads to real attacks following from search-decision equivalence
    - Comment in [Peikert, PQCrypto 2014]
    - Attack described in [Fluhrer, Eprint 2016]

- Need to ensure usage is okay with just IND-CPA
- Or construct IND-CCA KEM/PKE using Fujisaki–Okamoto transform or quantum-resistant variant [Targhi–Unruh, TCC 2016] [Hofheinz et al., Eprint 2017]

Direct key agreement

LWE and ring-LWE public key encryption and key exchange

Regev
STOC 2005
- Public key encryption from LWE

Lyubashevsky, Peikert, Regev
Eurocrypt 2010
- Public key encryption from ring-LWE

Lindner, Peikert
ePrint 2010, CT-RSA 2011
- Public key encryption from LWE and ring-LWE
- Approximate key exchange from LWE

Ding, Xie, Lin
ePrint 2012
- Key exchange from LWE and ring-LWE with single-bit reconciliation

Peikert
PQCrypto 2014
- Key encapsulation mechanism based on ring-LWE and variant single-bit reconciliation

Bos, Costello, Naehrig, Stebila
IEEE S&P 2015
- Implementation of ring-LWE key exchange, testing in TLS 1.2

Basic LWE key agreement (unauthenticated)

Based on Lindner–Peikert LWE public key encryption scheme

public: "big" A in $\mathbf{Z}_q^{n \times m}$

Alice		**Bob**
secret: random "small" s, e in $\mathbf{Z}_q^m$		secret: random "small" s', e' in $\mathbf{Z}_q^n$
	$b = As + e$ →	
	← $b' = s'A + e'$	
shared secret: $b's = s'As + e's \approx s'As$		shared secret: $s'b \approx s'As$

These are only approximately equal ⇒ need rounding

Rounding

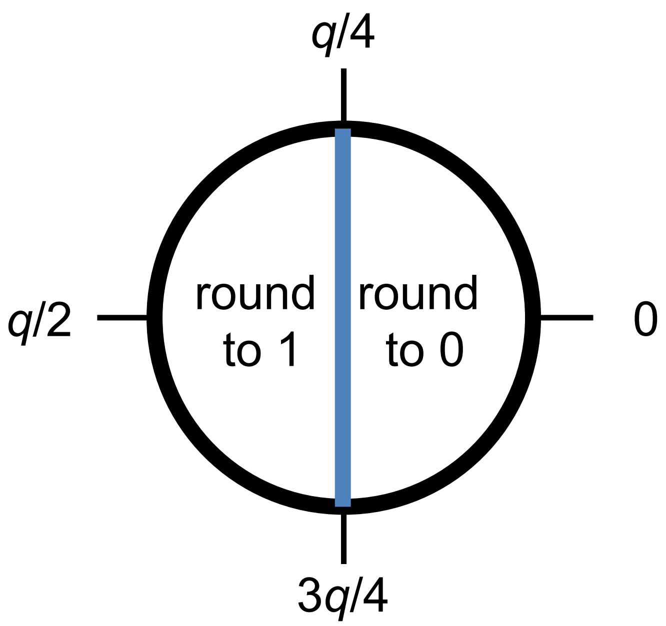
- Each coefficient of the polynomial is an integer modulo q
- Treat each coefficient independently

- Techniques by Ding [Din12] and Peikert [Pei14]

[Ding; eprint 2012] [Peikert; PQCrypto 2014]

Basic rounding

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

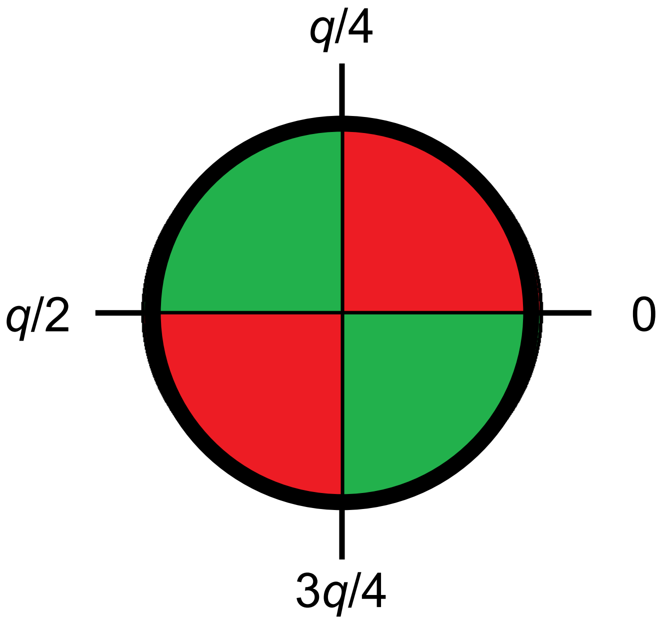


This works most of the time: prob. failure 2^{-10}.

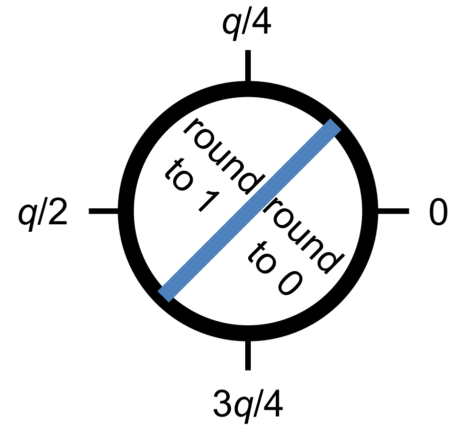
Not good enough: we need exact key agreement.

Rounding (Peikert)

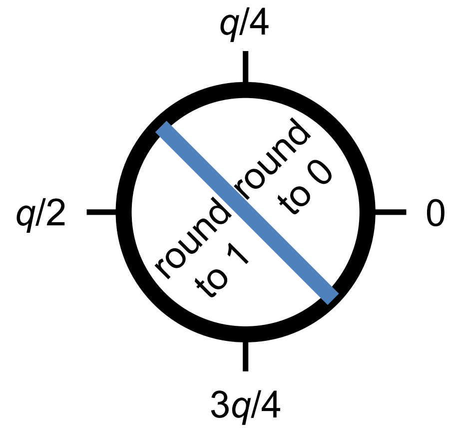
Bob says which of two regions the value is in:  or 



If 



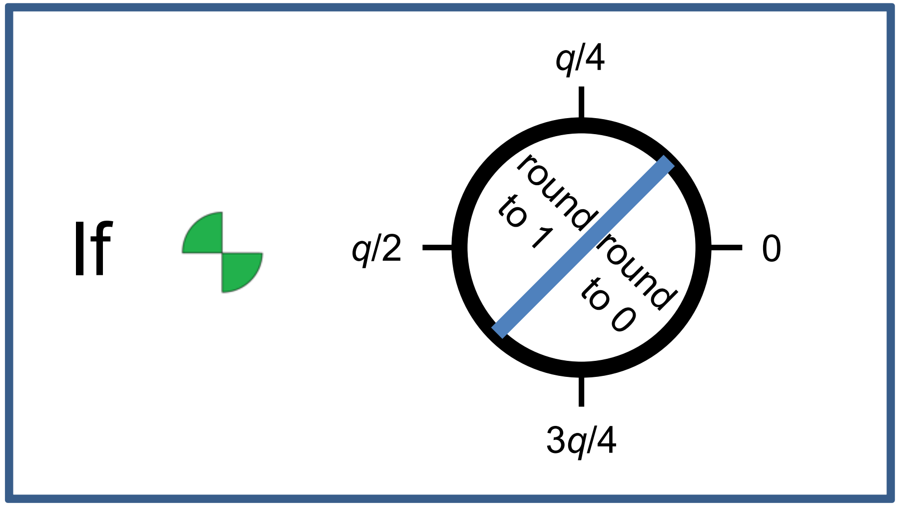
If 

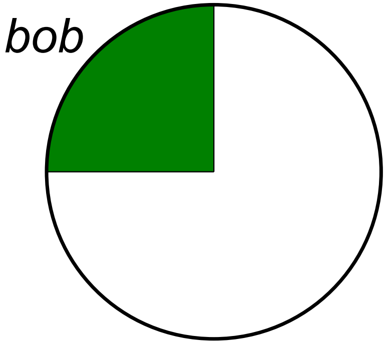


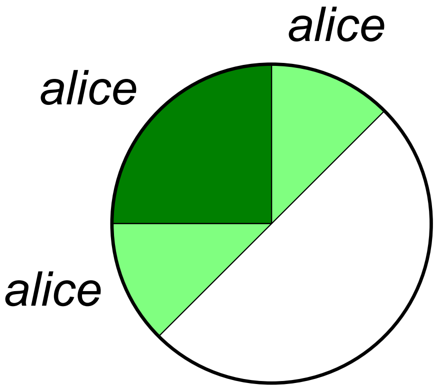
[Peikert; PQCrypto 2014]

Rounding (Peikert)

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







- Security not affected: revealing  or  leaks no information

[Peikert; PQCrypto 2014]

Exact LWE key agreement (unauthenticated)

public: "big" A in $\mathbf{Z}_q^{n \times m}$

Alice

secret:
random "small" s, e in $\mathbf{Z}_q^m$

Bob

secret:
random "small" s', e' in $\mathbf{Z}_q^n$

$b = As + e$ →

← $b' = s'A + e'$,  or 

Alice

shared secret:
round($b's$)

Bob

shared secret:
round($s'b$)

Exact ring-LWE key agreement (unauthenticated)

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($b \bullet s'$)

Exact LWE key agreement – "Frodo"

Alice		Bob
$\mathsf{seed}_{\mathbf{A}} \xleftarrow{\$} U(\{0,1\}^s)$		
$\mathbf{A} \leftarrow \mathsf{Gen}(\mathsf{seed}_{\mathbf{A}})$		
$\mathbf{S}, \mathbf{E} \xleftarrow{\$} \chi(\mathbb{Z}_q^{n\times\bar{n}})$		
$\boxed{\mathbf{B} \leftarrow \mathbf{A}\mathbf{S} + \mathbf{E}}$	$\xrightarrow{\mathsf{seed}_{\mathbf{A}}, \mathbf{B}}$ $\in \{0,1\}^s \times \mathbb{Z}_q^{n\times\bar{n}}$	$\mathbf{A} \leftarrow \mathsf{Gen}(\mathsf{seed}_{\mathbf{A}})$
		$\mathbf{S}', \mathbf{E}' \xleftarrow{\$} \chi(\mathbb{Z}_q^{\bar{m}\times n})$
		$\boxed{\mathbf{B}' \leftarrow \mathbf{S}'\mathbf{A} + \mathbf{E}'}$
		$\mathbf{E}'' \xleftarrow{\$} \chi(\mathbb{Z}_q^{\bar{m}\times\bar{n}})$
		$\boxed{\mathbf{V} \leftarrow \mathbf{S}'\mathbf{B} + \mathbf{E}''}$
	$\xleftarrow{\mathbf{B}', \mathbf{C}}$ $\in \mathbb{Z}_q^{\bar{m}\times n} \times \mathbb{Z}_2^{\bar{m}\times\bar{n}}$	$\mathbf{C} \leftarrow \langle \mathbf{V} \rangle_{2^B}$
$K \leftarrow \mathsf{rec}(\boxed{\mathbf{B}'\mathbf{S}}, \mathbf{C})$		$K \leftarrow \lfloor \mathbf{V} \rceil_{2^B}$

A generated pseudorandomly

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 random oracle).

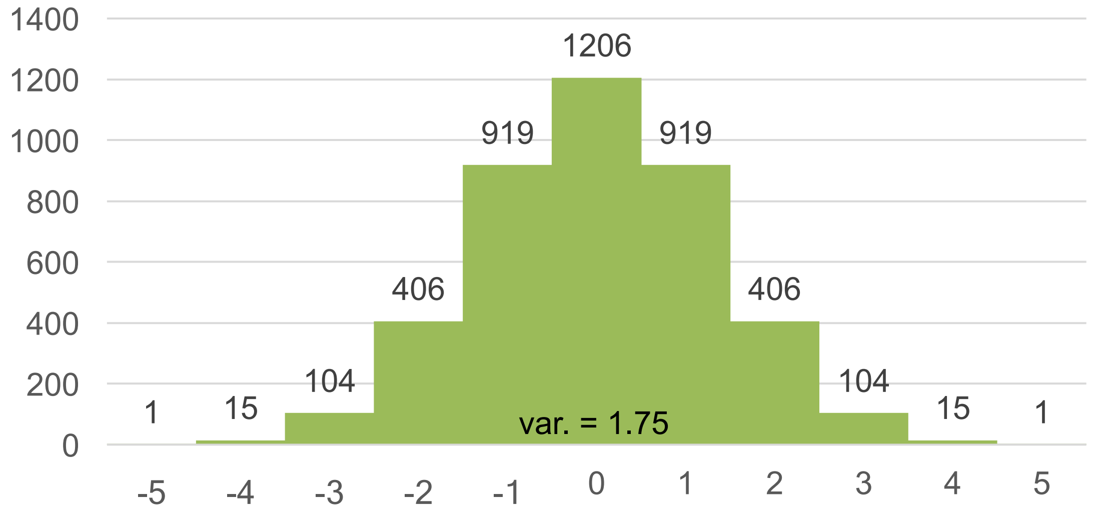
[Bos et al.; ACM CCS 2016]

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”

- 144-bit classical security,
  130-bit quantum security,
  103-bit plausible lower bound

- $n = 752$, $m = 8$, $q = 2^{15}$
- χ = approximation to rounded Gaussian with 11 elements

- Failure: $2^{-38.9}$

- Total communication: 22.6 KiB

“Paranoid”

- 177-bit classical security,
  161-bit quantum security,
  128-bit plausible lower bound

- $n = 864$, $m = 8$, $q = 2^{15}$
- χ = approximation to rounded Gaussian with 13 elements

- Failure: $2^{-33.8}$

- Total communication: 25.9 KiB

Exact ring-LWE key agreement – "BCNS15"

BCNS15		
Public parameters: $n, q, \chi,\ a \leftarrow_{\$} \mathcal{U}(R_q)$		
Alice		**Bob**
$s, e \leftarrow_{\$} \chi(R_q)$		
$\boxed{\tilde{b} \leftarrow as + e} \in R_q$	$\xrightarrow{b}$	$s', e' \leftarrow_{\$} \chi(R_q)$
		$\boxed{\tilde{b}' \leftarrow as' + e'} \in R_q$
		$e'' \leftarrow_{\$} \chi(R_q)$
		$\boxed{\tilde{v} \leftarrow bs' + e''} \in R_q$
		$\overline{v} \leftarrow_{\$} \mathrm{dbl}(\tilde{v}) \in R_{2q}$
	$\xleftarrow{\tilde{b}', c}$	$c \leftarrow \langle \overline{v}/2 \rangle_2 \in \{0,1\}^n$
$k_A \leftarrow \mathrm{rec}_2(\boxed{2\tilde{b}'s}, c) \in \{0,1\}^n$		$k_B \leftarrow \lfloor \overline{v}/2 \rceil_2 \in \{0,1\}^n$

[Bos, Costello, Naehrig, Stebila; IEEE S&P 2015]

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

"NewHope"

Alkim, Ducas, Pöppelman, Schwabe.
USENIX Security 2016

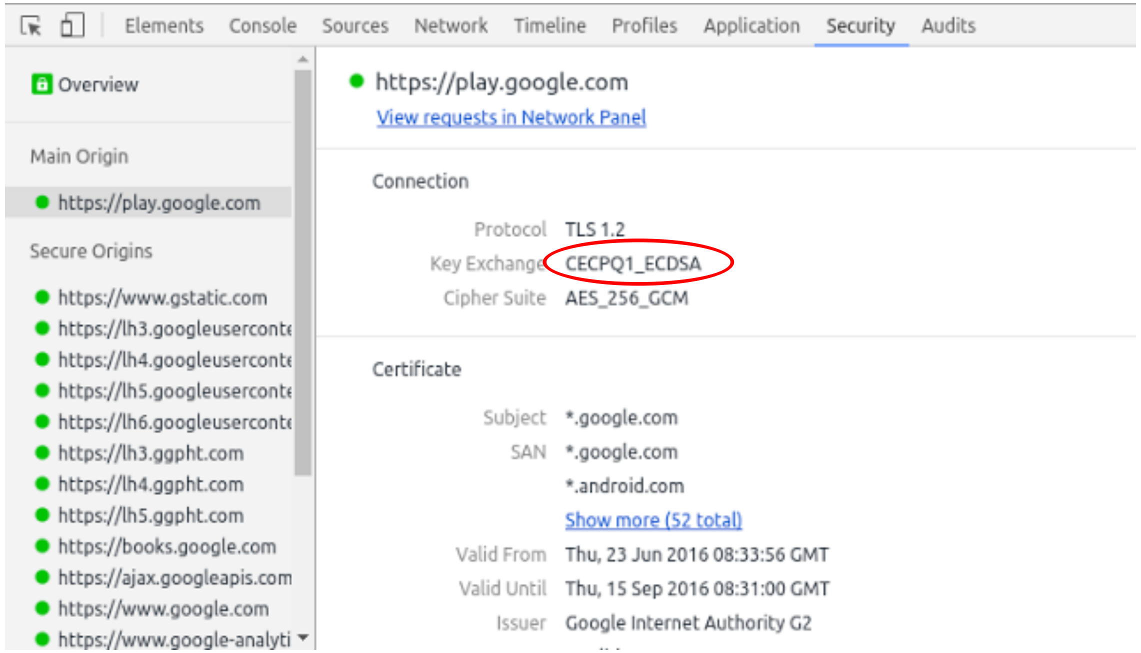
- New parameters
- Different error distribution
- Improved performance
- Pseudorandomly generated parameters

- Further performance improvements by others [GS16,LN16,AOPPS17,…]

Google Security Blog

Experimenting with Post-Quantum Cryptography

July 7, 2016



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

Evaluation

Our implementations

- Ring-LWE BCNS15
- LWE Frodo

Pure C implementations

Constant time

Compare with others

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

Use assembly code

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

Pure C implementations

Post-quantum key exchange performance

	Speed		Communication	
RSA 3072-bit	Fast	4 ms	Small	0.3 KiB
ECDH `nistp256`	Very fast	0.7 ms	Very small	0.03 KiB
Code-based	Very fast	0.5 ms	Very large	360 KiB
NTRU	Very fast	0.3–1.2 ms	Medium	1 KiB
Ring-LWE	Very fast	0.2–1.5 ms	Medium	2–4 KiB
LWE	Fast	1.4 ms	Large	11 KiB
Isogenies (SIDH)	Med.–slow	15–400 ms	Small	0.5 KiB

See [Bos, Costello, Ducas, Mironov, Naehrig, Nikolaenko, Raghunathan, Stebila, ACM CCS 2016] for details/methodology

Post-quantum signature sizes

	Public key		Signature	
RSA 3072-bit	Small	0.3 KiB	Small	0.3 KiB
ECDSA `nistp256`	Very small	0.03 KiB	Very small	0.03 KiB
Hash-based (stateful)	Small	0.9 KiB	Medium	3.6 KiB
Hash-based (stateless)	Small	1 KiB	Large	32 KiB
Lattice-based (ignoring tightness)	Medium	1.5–8 KiB	Medium	3–9 KiB
Lattice-based (respecting tightness)	Very large	1330 KiB	Small	1.2 KiB
Isogenies (SIDH)	Small	1.5 KiB	Very large	141 KiB

See [Bindel, Herath, McKague, Stebila PQCrypto 2017] for details

Transitioning to PQ crypto

Retroactive decryption

- A passive adversary that records today's communication can decrypt once they get a quantum computer
  - Not a problem for some people
  - Is a problem for other people

- How to provide potential post-quantum security to early adopters?

Hybrid ciphersuites

- Use pre-quantum and post-quantum algorithms together
- Secure if either one remains unbroken

Need to consider backward compatibility for non-hybrid-aware systems

Why hybrid?

- Potential post-quantum security for early adopters
- Maintain compliance with older standards (e.g. FIPS)
- Reduce risk from uncertainty on PQ assumptions/parameters

Hybrid ciphersuites

	Key exchange	Authentication
1	Hybrid traditional + PQ	Single traditional
2	Hybrid traditional + PQ	Hybrid traditional + PQ
3	Single PQ	Single traditional
4	Single PQ	Single PQ

Likely focus
for next 10 years

Hybrid key exchange in TLS 1.2

Create a new DH-style ciphersuite with a new key exchange method

- Within the ClientKeyExchange and ServerKeyExchange, convey an ECDH public key and a PQ public key using some internal concatenation format
- Compute two shared secrets, use their concatenation as the premaster secret

Experiments for hybrid key exchange in TLS 1.2

Several papers and prototypes:

- Bos, Costello, Naehrig, Stebila, S&P 2015
- Bos, Costello, Ducas, Mironov, Naehrig, Nikolaenko, Raghunathan, Stebila, ACM CCS 2016
- Google Chrome experiment
- liboqs OpenSSL fork
  - https://openquantumsafe.org/

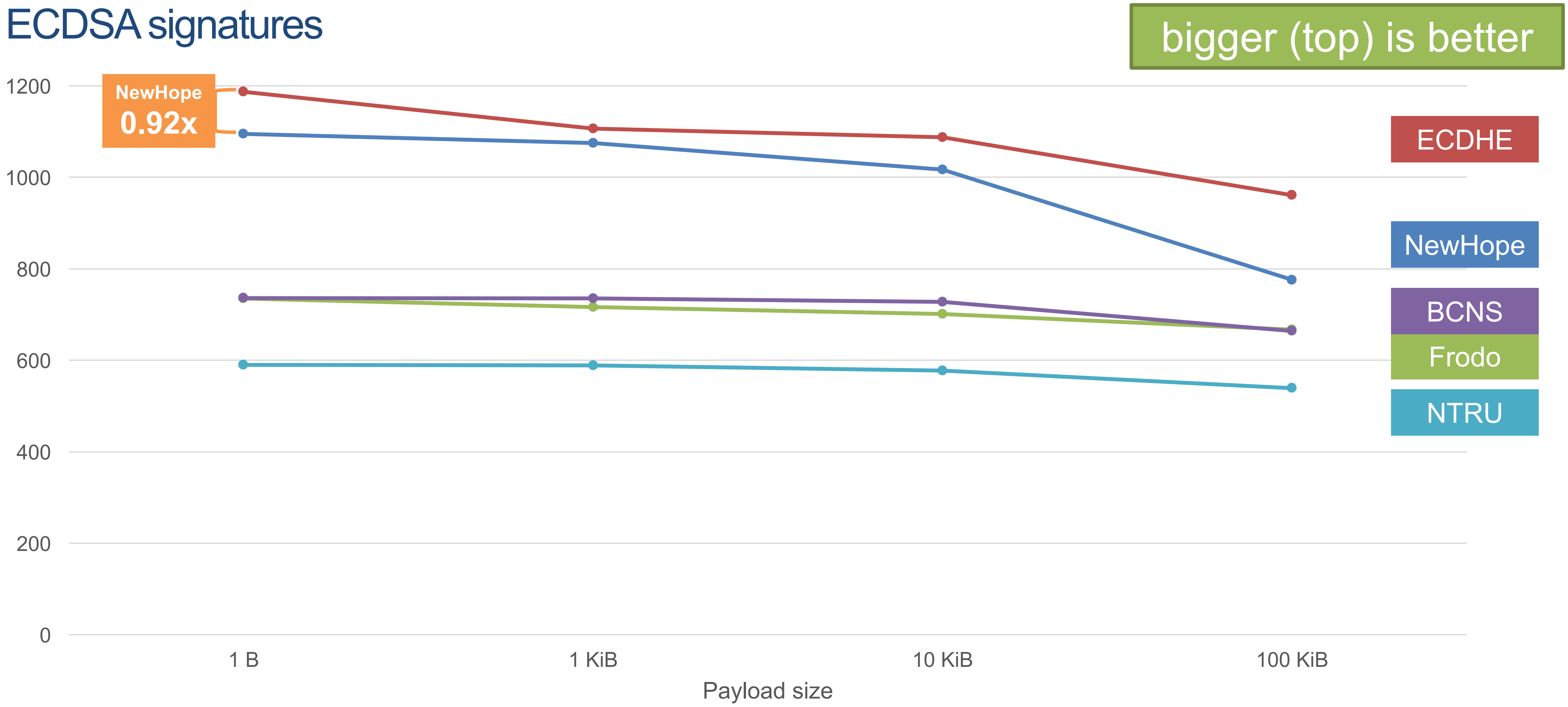
No backwards compatibility issues

- https://www.imperialviolet.org/2016/11/28/cecpq1.html

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

TLS connection throughput – hybrid w/ECDHE

ECDSA signatures



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

https://openquantumsafe.org/

Open Quantum Safe

- MIT-licensed open-source project on Github
  - https://openquantumsafe.org/
  - https://github.com/open-quantum-safe/
- liboqs: C language library, common API
- Builds on x86 (Linux, Mac, Windows), ARM (Android, Linux)

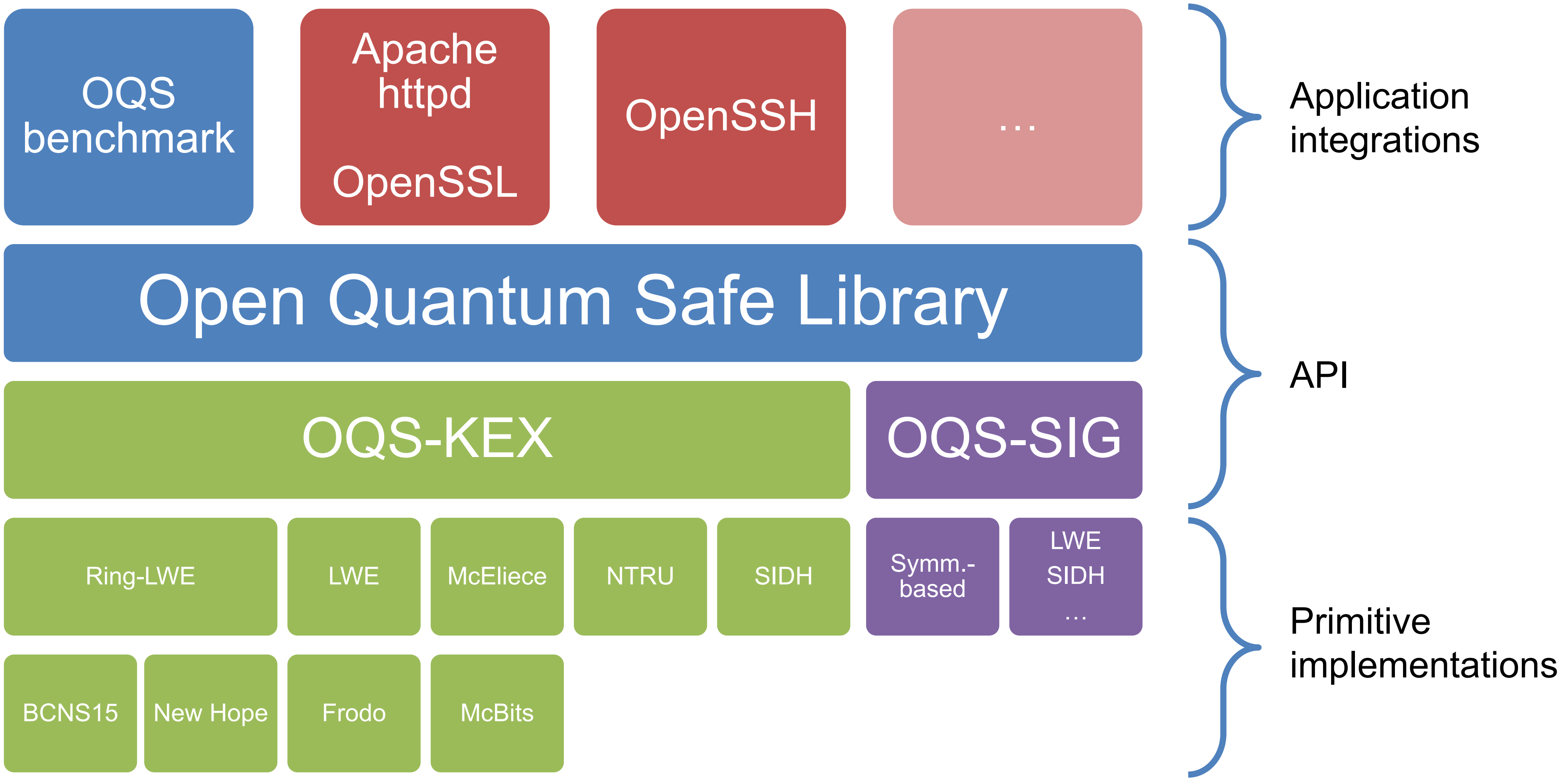
Open Quantum Safe

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
   - See also eBACS/SUPERCOP

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



liboqs: Current algorithms

Key exchange

- **Ring-LWE**:
  - BCNS15
  - NewHope
  - MSR NewHope improvements
- **LWE**: Frodo
- **M-LWE**: Kyber
- **NTRU**
- **SIDH (Supersingular isogeny Diffie–Hellman)**:
  - MSR
  - IQC
- **Code**: McBits

Digital signatures

- **Symmetric-based**:
  - Picnic

liboqs: Benchmarking

- Built-in key exchange benchmarking suite
  - `./test_kex --bench`
- Gives cycle counts and ms runtimes
- Also have memory usage benchmarks

liboqs: Application integrations

OpenSSL v1.0.2:

- Ciphersuites using key exchange algorithms from liboqs
- Integrated into `openssl speed` benchmarking command and `s_client` and `s_server` command-line programs

- Track OpenSSL 1.0.2 stable with regular updates
  - https://github.com/open-quantum-safe/openssl

- Successfully used in Apache httpd and OpenVPN (with no modifications!)

OpenSSH:

- Using key exchange algorithms from liboqs
- Patch contributed by Microsoft Research
  - https://github.com/Microsoft/PQCrypto-PatchforOpenSSH

Summary

Quantum-safe crypto

Classical post-quantum crypto

Hash-based
- Merkle signatures
- Sphincs

Code-based
- McEliece
- Niederreiter

Multivariate
- multivariate quadratic

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

Isogenies
- supersingular elliptic curve isogenies

Quantum crypto

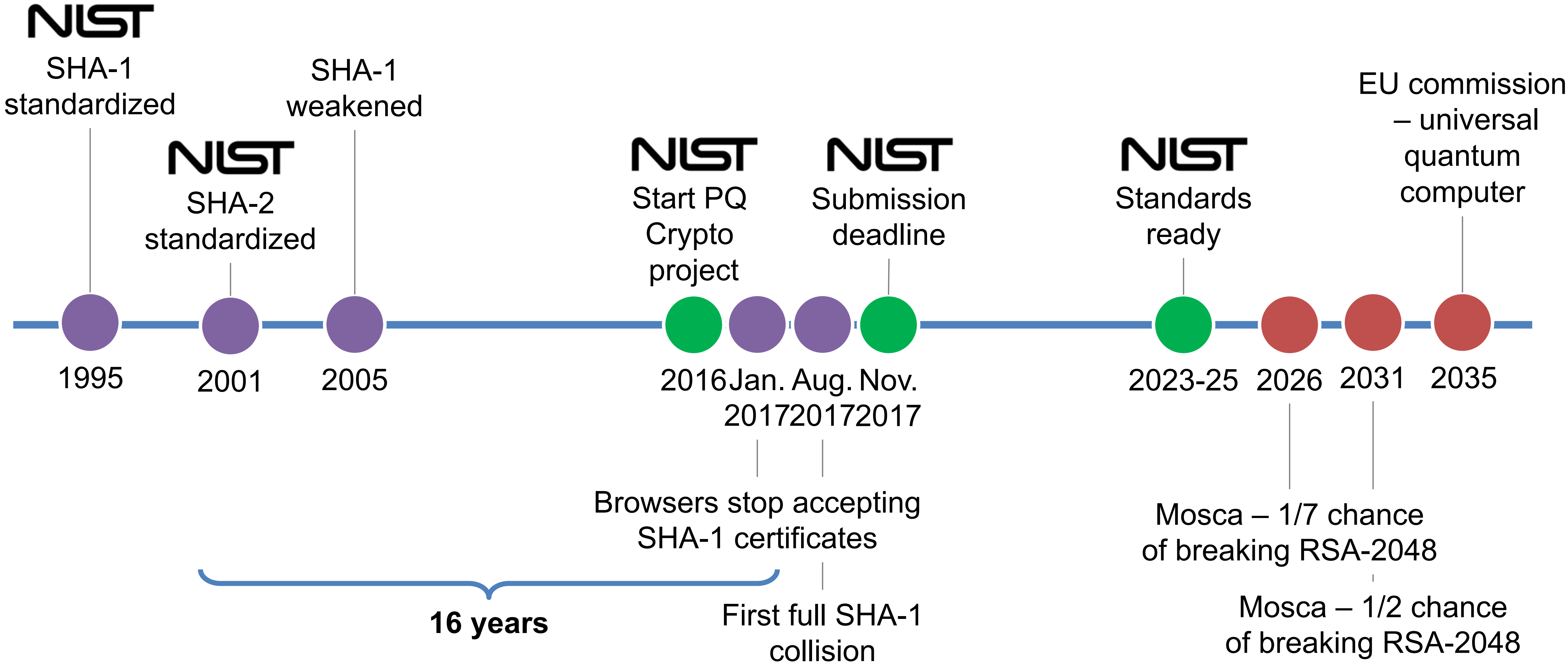
Quantum key distribution

Quantum random number generators

Quantum channels

Quantum blind computation

Timeline


SHA-1 standardized
1995
SHA-2 standardized
2001
SHA-1 weakened
2005
Start PQ Crypto project
2016
Jan. 2017
Browsers stop accepting SHA-1 certificates
Aug. 2017
First full SHA-1 collision
Submission deadline
Nov. 2017
16 years
Standards ready
2023-25
2026
Mosca – 1/7 chance of breaking RSA-2048
2031
Mosca – 1/2 chance of breaking RSA-2048
EU commission – universal quantum computer
2035

Practical post-quantum key exchange

Douglas Stebila
McMaster University

Survey paper

- https://eprint.iacr.org/2016/1017

Open Quantum Safe project

- https://openquantumsafe.org/

LWE key exchange (Frodo)

- https://github.com/lwe-frodo
- https://eprint.iacr.org/2016/659

Hybrid PKI

- https://eprint.iacr.org/2017/460

https://www.douglas.stebila.ca/research/presentations/

Appendix

Lindner–Peikert public key encryption

Let n, q, χ be LWE parameters.

- $\mathrm{KeyGen}()$: $\mathbf{s} \stackrel{\$}{\leftarrow} \chi(\mathbb{Z}^n)$. $\mathbf{A} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^{n \times n}$. $\mathbf{e} \stackrel{\$}{\leftarrow} \chi(\mathbb{Z}^n)$. $\tilde{\mathbf{b}} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{e}$.
  Return $pk \leftarrow (\mathbf{A}, \tilde{\mathbf{b}})$ and $sk \leftarrow \mathbf{s}$.
- $\mathrm{Enc}(pk, x \in \{0,1\})$: $\mathbf{s}' \stackrel{\$}{\leftarrow} \chi(\mathbb{Z}^n)$. $\mathbf{e}' \stackrel{\$}{\leftarrow} \chi(\mathbb{Z}^n)$. $\tilde{\mathbf{b}}' \leftarrow \mathbf{s}'\mathbf{A} + \mathbf{e}'$. $e'' \stackrel{\$}{\leftarrow} \chi(\mathbb{Z})$.
  $\tilde{v}' \leftarrow \langle \mathbf{s}', \tilde{\mathbf{b}} \rangle + e''$. $c \leftarrow \mathrm{encode}(x) + \tilde{v}'$. Return $ctxt \leftarrow (\tilde{\mathbf{b}}', c)$.
- $\mathrm{Dec}(sk, (\tilde{\mathbf{b}}', c))$: $v \leftarrow \langle \tilde{\mathbf{b}}', \mathbf{s} \rangle$. Return $\mathrm{decode}(c - v)$.

[Lindner, Peikert; CT-RSA 2011]

Encode/decode

$$\mathrm{encode}(x \in \{0,1\}) \leftarrow x \cdot \left\lfloor \frac{q}{2} \right\rfloor$$

$$\mathrm{decode}(\overline{x} \in \mathbb{Z}_q) \leftarrow \begin{cases} 0, & \text{if } \overline{x} \in [-\left\lfloor \frac{q}{4} \right\rfloor, \left\lfloor \frac{q}{4} \right\rfloor) \\ 1, & \text{otherwise} \end{cases}$$

[Regev; STOC 2005]

Correctness

Sender and receiver approximately compute the same shared secret $\mathbf{s}'\mathbf{A}\mathbf{s}$

$$\tilde{v}' = \langle \mathbf{s}', \tilde{\mathbf{b}} \rangle + e'' = \mathbf{s}'(\mathbf{A}\mathbf{s} + \mathbf{e}) + e'' = \mathbf{s}'\mathbf{A}\mathbf{s} + \langle \mathbf{s}', \mathbf{e} \rangle + e'' \approx \mathbf{s}'\mathbf{A}\mathbf{s}$$

$$v = \langle \tilde{\mathbf{b}}', \mathbf{s} \rangle = (\mathbf{s}'\mathbf{A} + \mathbf{e}')\mathbf{s} = \mathbf{s}'\mathbf{A}\mathbf{s} + \langle \mathbf{e}', \mathbf{s} \rangle \approx \mathbf{s}'\mathbf{A}\mathbf{s}$$