Introduction to post-quantum cryptography and learning with errors

Douglas Stebila McMaster University UNIVERSITY OF WATERLOO

Funding acknowledgements:

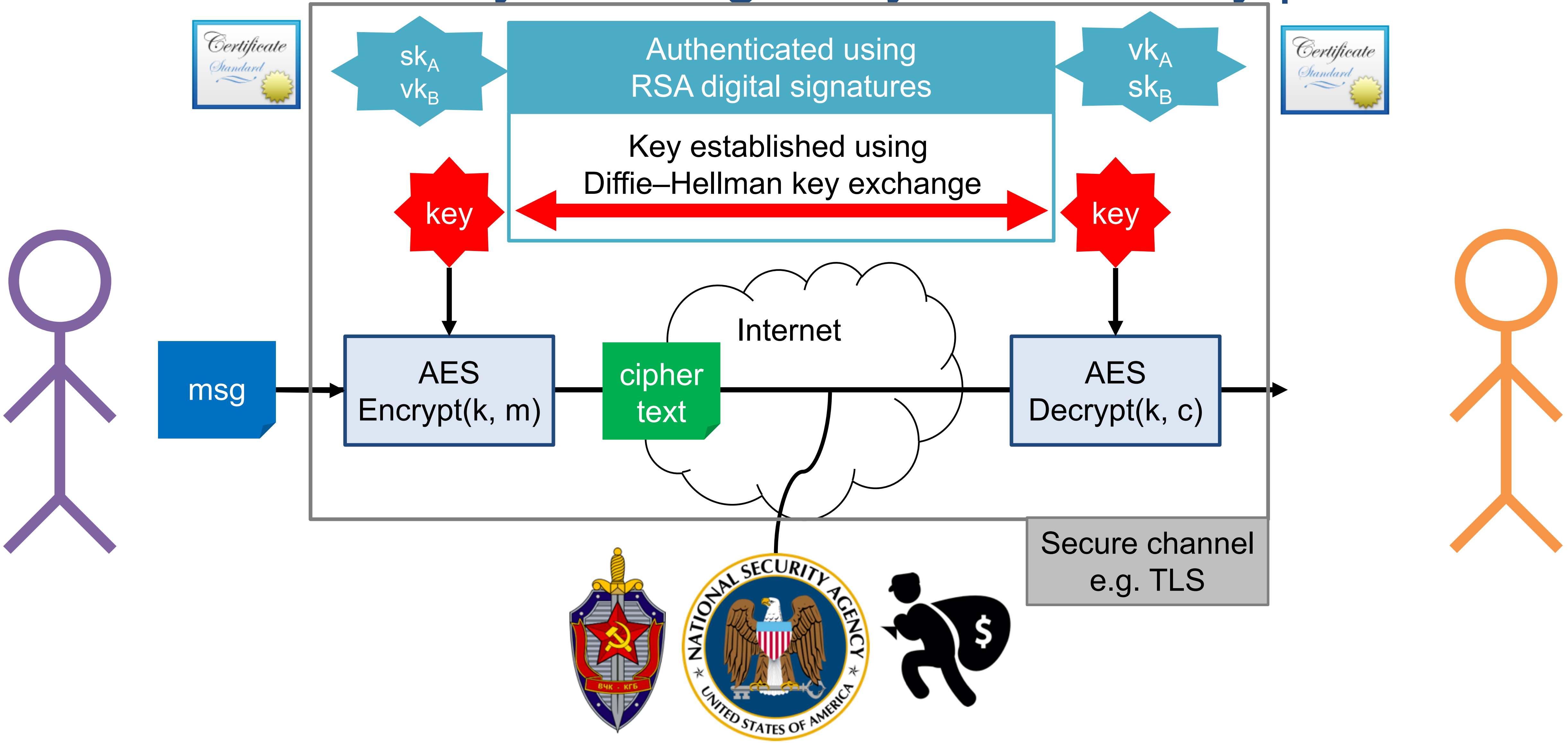
NSERC CRSNG

Summer School on real-world crypto and privacy • Šibenik, Croatia • June 11, 2018
https://www.douglas.stebila.ca/research/presentations

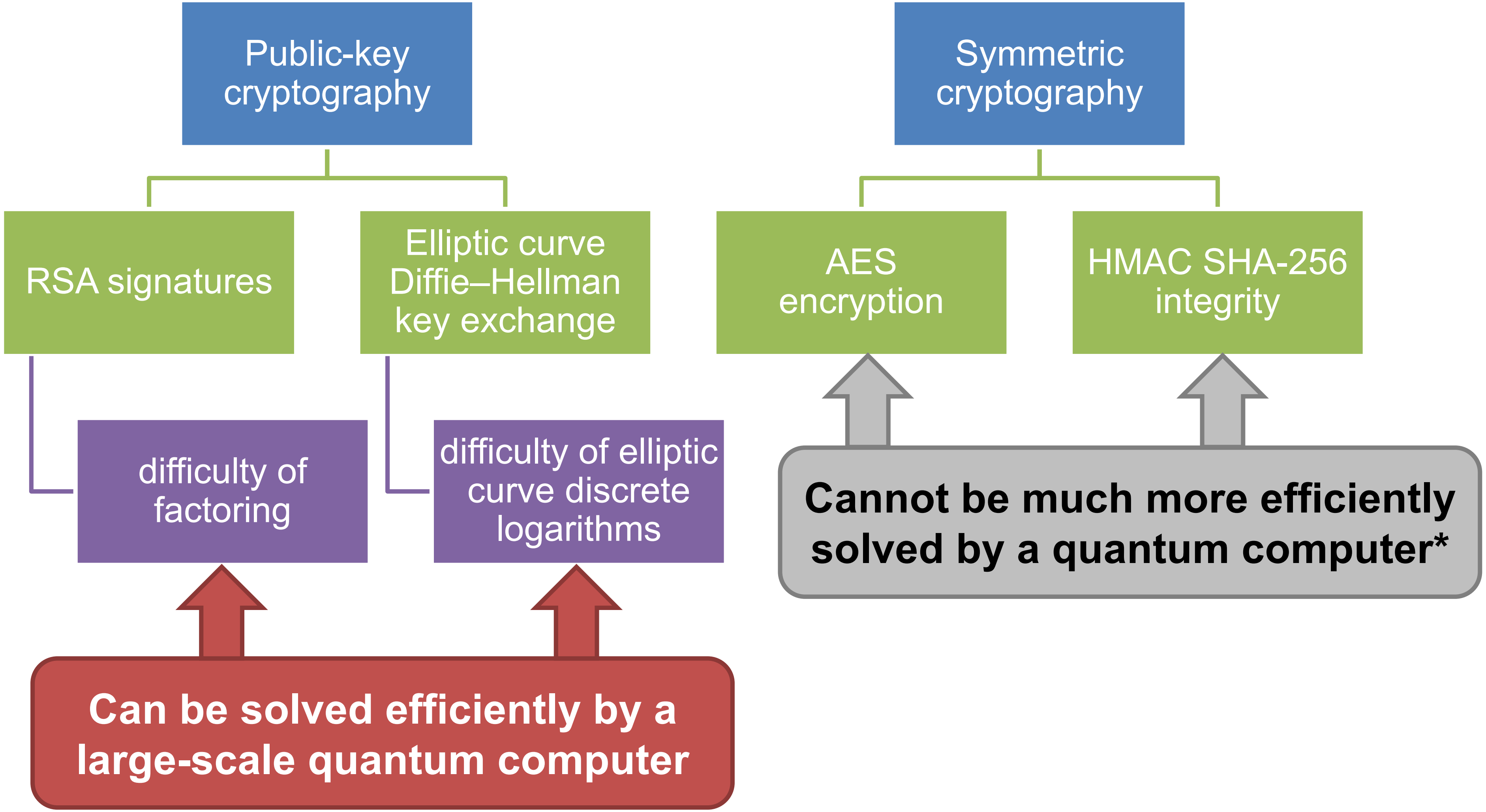
Summary

- Intro to post-quantum cryptography
- Learning with errors problems
  - LWE, Ring-LWE, Module-LWE, Learning with Rounding, NTRU
  - Search, decision
  - With uniform secrets, with short secrets
- Public key encryption from LWE
  - Regev
  - Lindner–Peikert
- Security of LWE
  - Lattice problems – GapSVP
- KEMs and key agreement from LWE
- Other applications of LWE
- PQ security models
- Transitioning to PQ crypto

Authenticated key exchange + symmetric encryption



Cryptographic building blocks



Public-key cryptography

RSA signatures

Elliptic curve Diffie–Hellman key exchange

difficulty of factoring

difficulty of elliptic curve discrete logarithms

Can be solved efficiently by a large-scale quantum computer

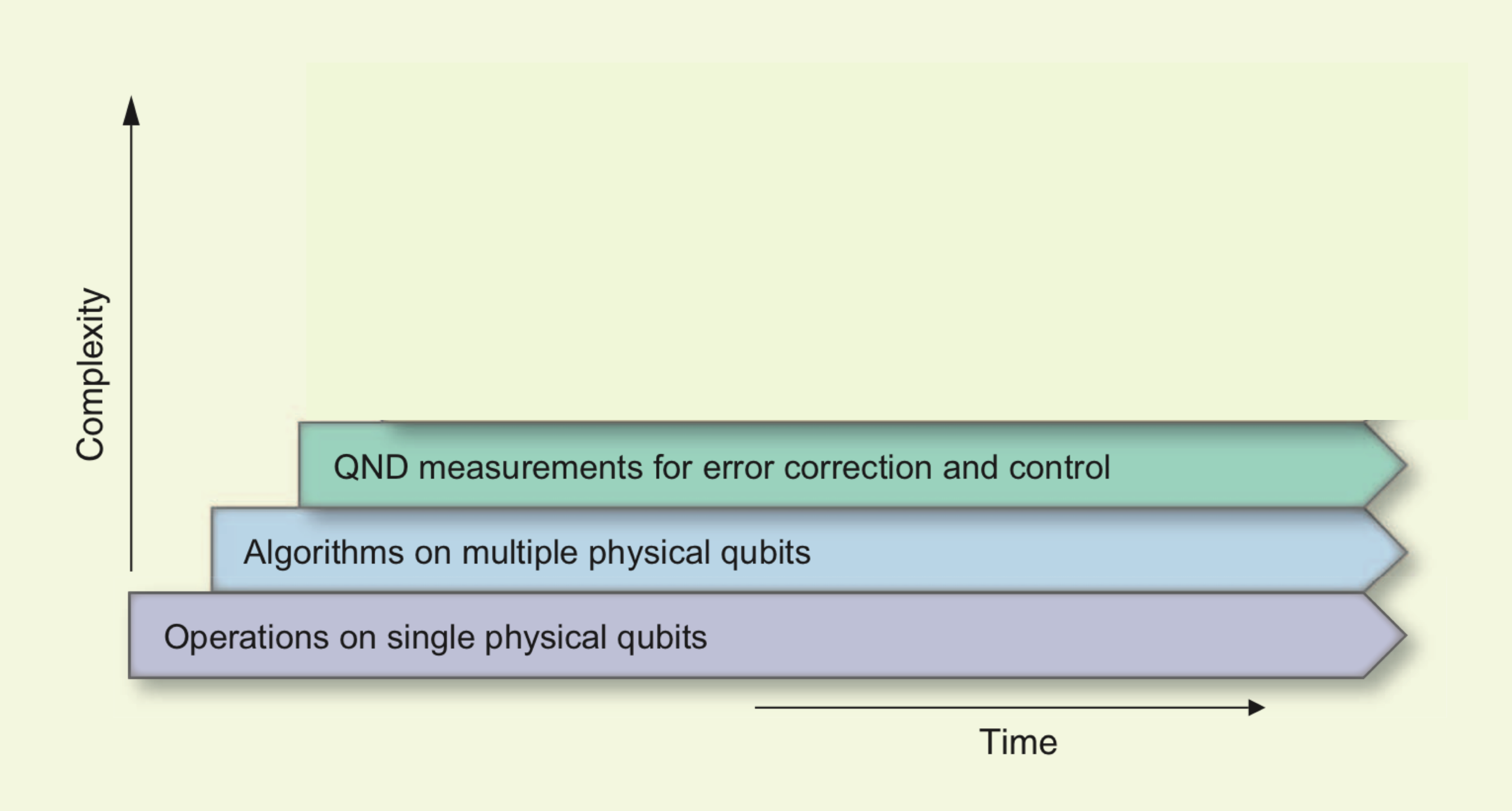
Symmetric cryptography

AES encryption

HMAC SHA-256 integrity

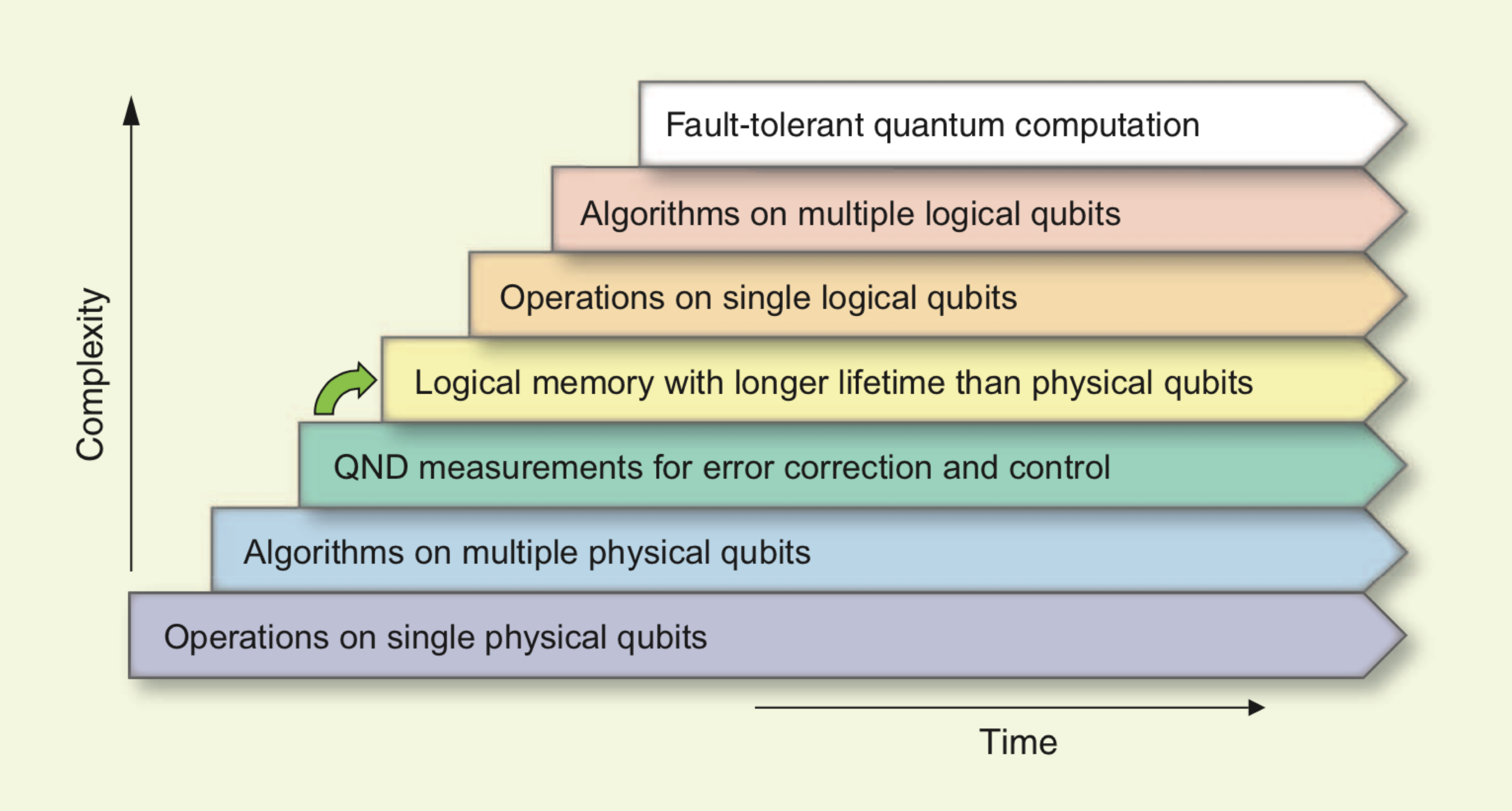
Cannot be much more efficiently solved by a quantum computer*

When will a large-scale quantum computer be built?


Complexity
QND measurements for error correction and control
Algorithms on multiple physical qubits
Operations on single physical qubits
Time

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

When will a large-scale quantum computer be built?


Fault-tolerant quantum computation
Algorithms on multiple logical qubits
Operations on single logical qubits
Logical memory with longer lifetime than physical qubits
QND measurements for error correction and control
Algorithms on multiple physical qubits
Operations on single physical qubits
Complexity
Time

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.”

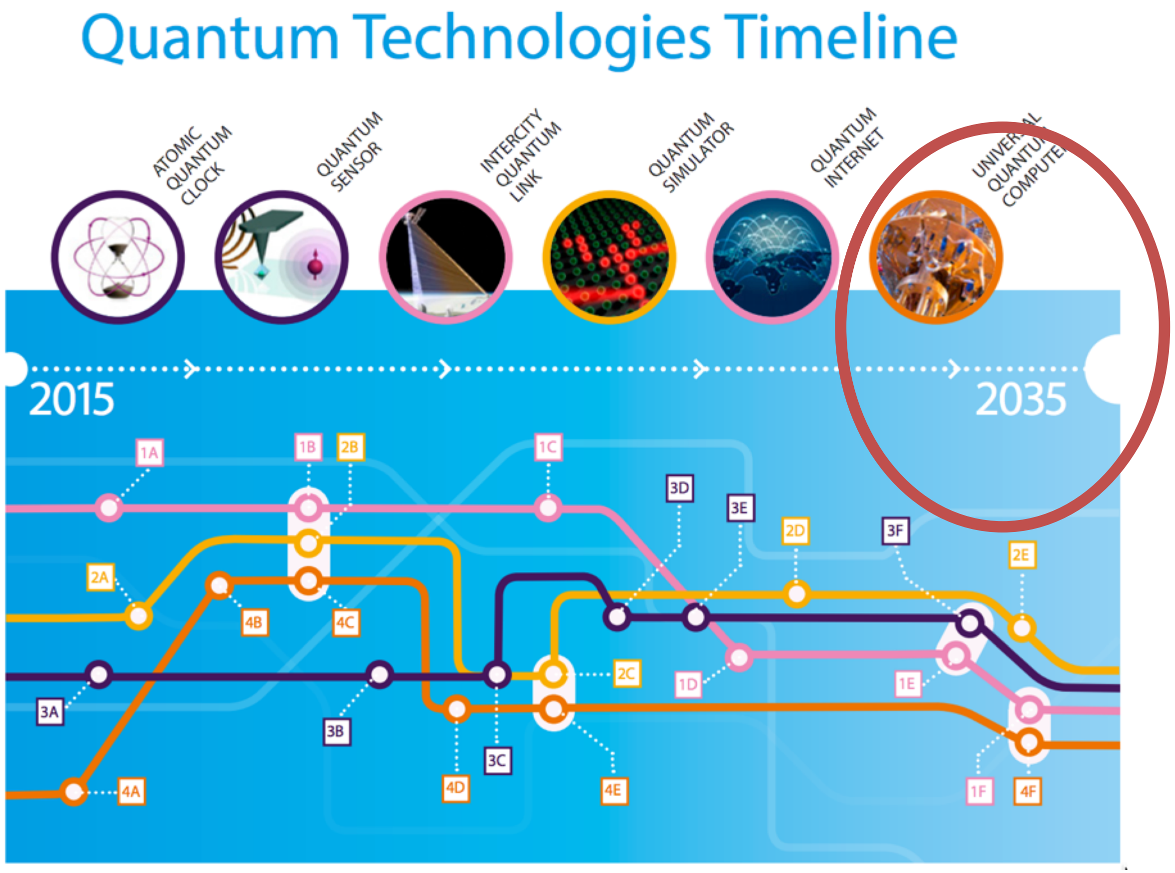
— Michele Mosca, November 2015
https://eprint.iacr.org/2015/1075

When will a large-scale quantum computer be built?

Quantum Manifesto

A New Era of Technology

May 2016

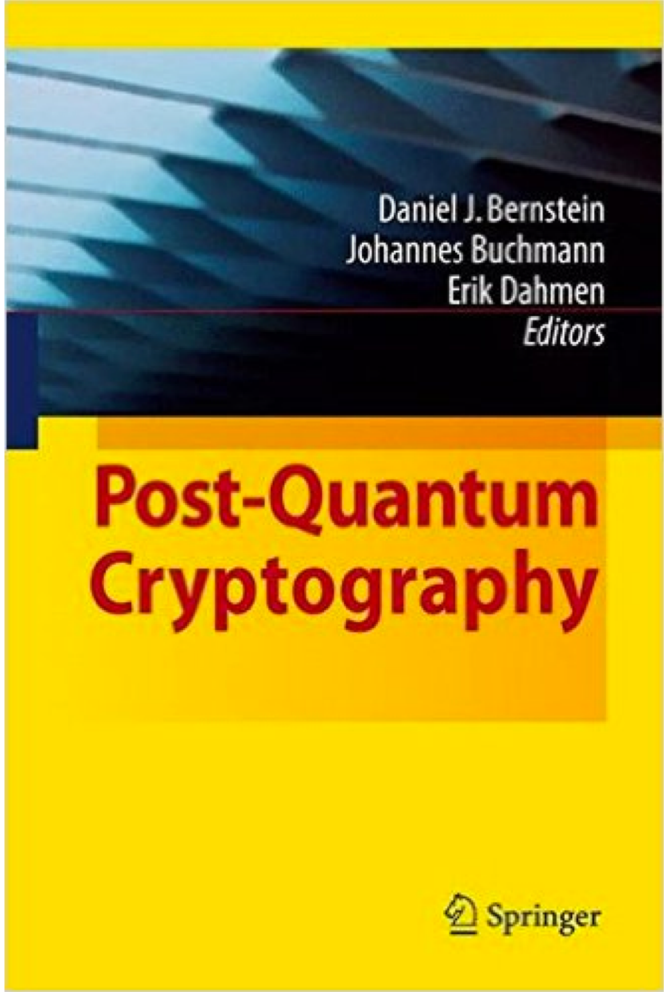


http://qurope.eu/system/files/u7/93056_Quantum%20Manifesto_WEB.pdf

Post-quantum cryptography in academia

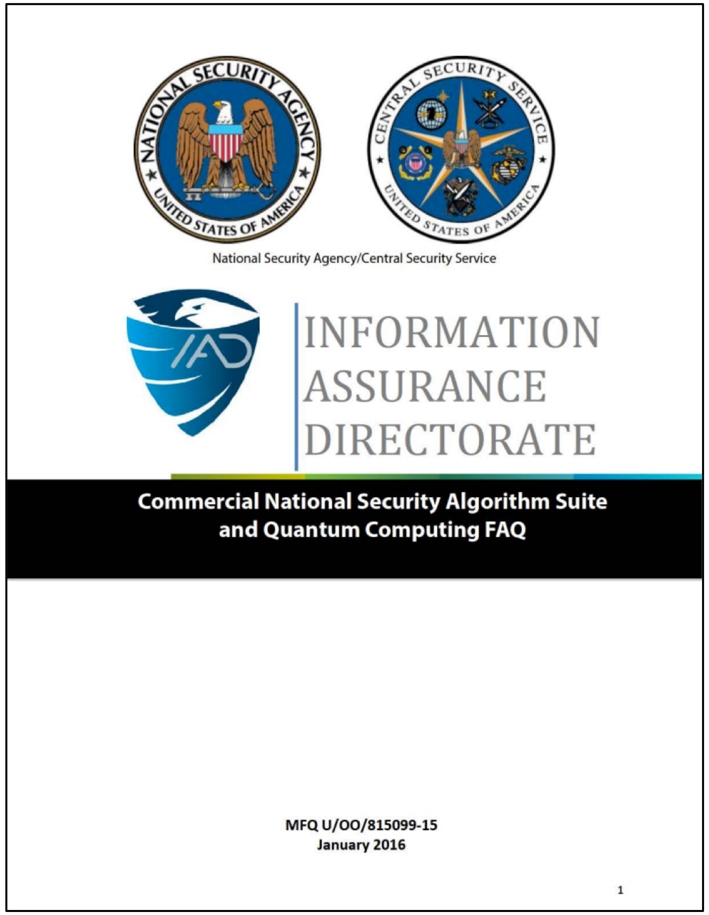
<u>Conference series</u>

- 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

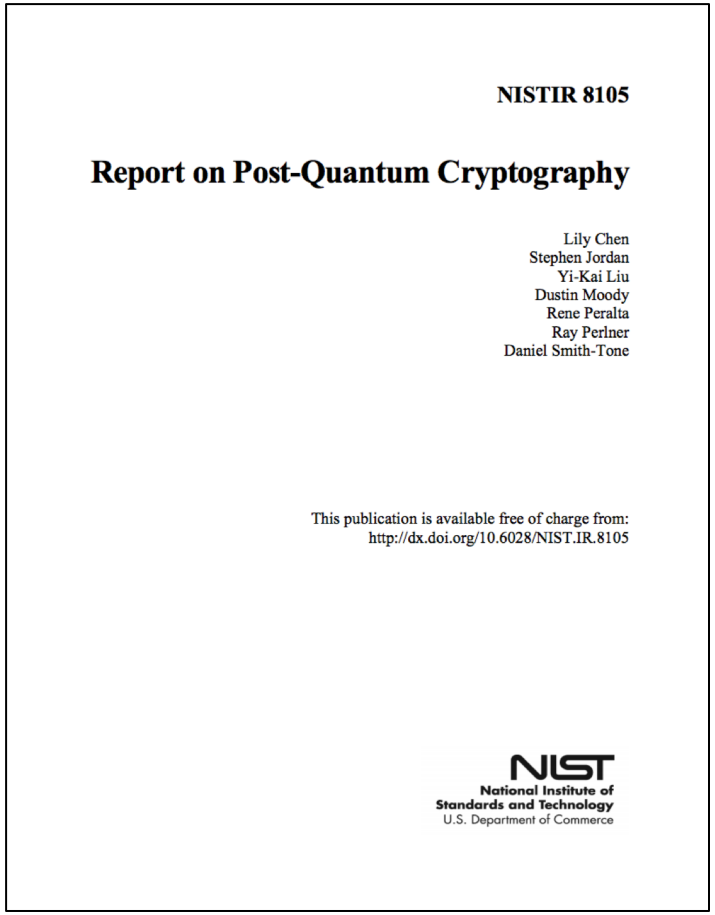
MFQ U/OO/815099-15
January 2016

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 69 submissions 1/3 signatures, 2/3 KEM/PKE
3–5 years	**Analysis phase**
2 years later (2023–2025)	Draft standards ready

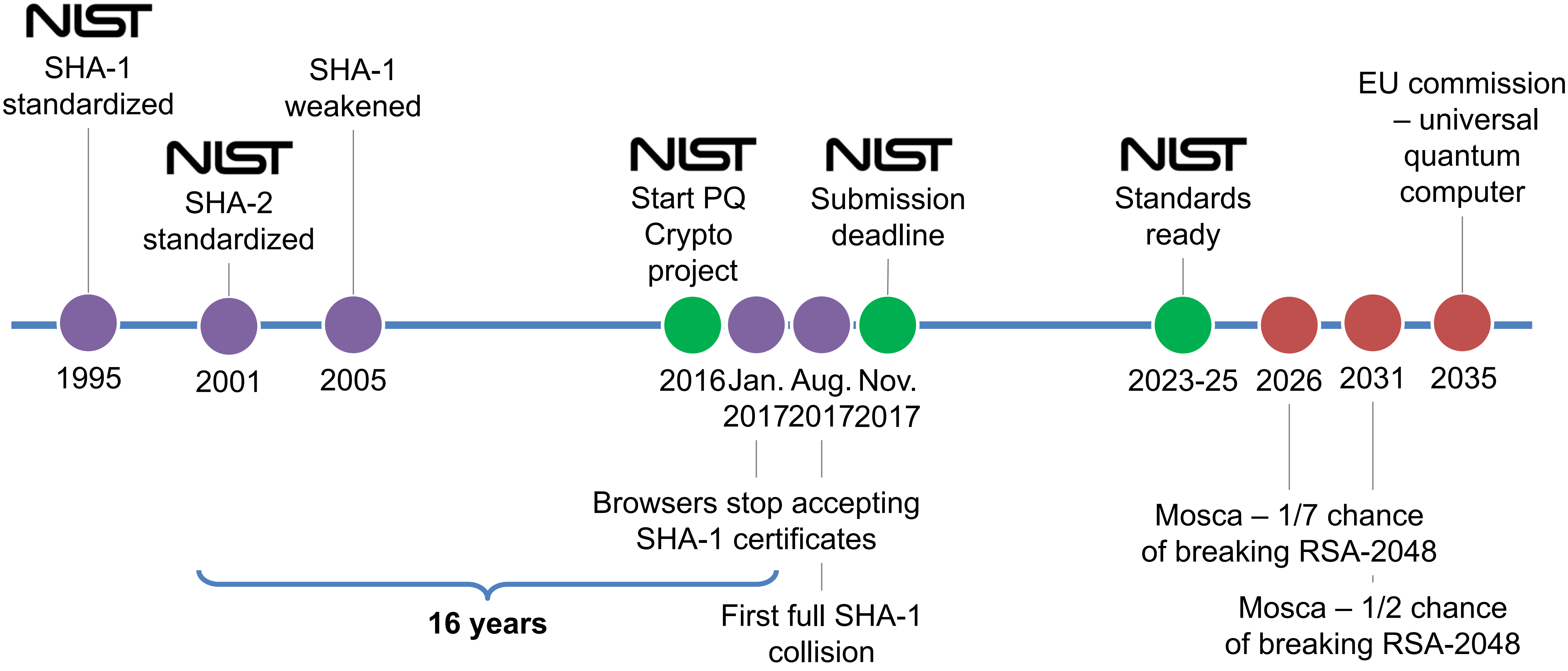
NIST Post-quantum Crypto Project

http://www.nist.gov/pqcrypto

"**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

Classical crypto with no known exponential quantum speedup

Hash- & symmetric-based

- Merkle signatures
- Sphincs
- Picnic

Code-based

- McEliece
- Niederreiter

Multivariate

- multivariate quadratic

Lattice-based

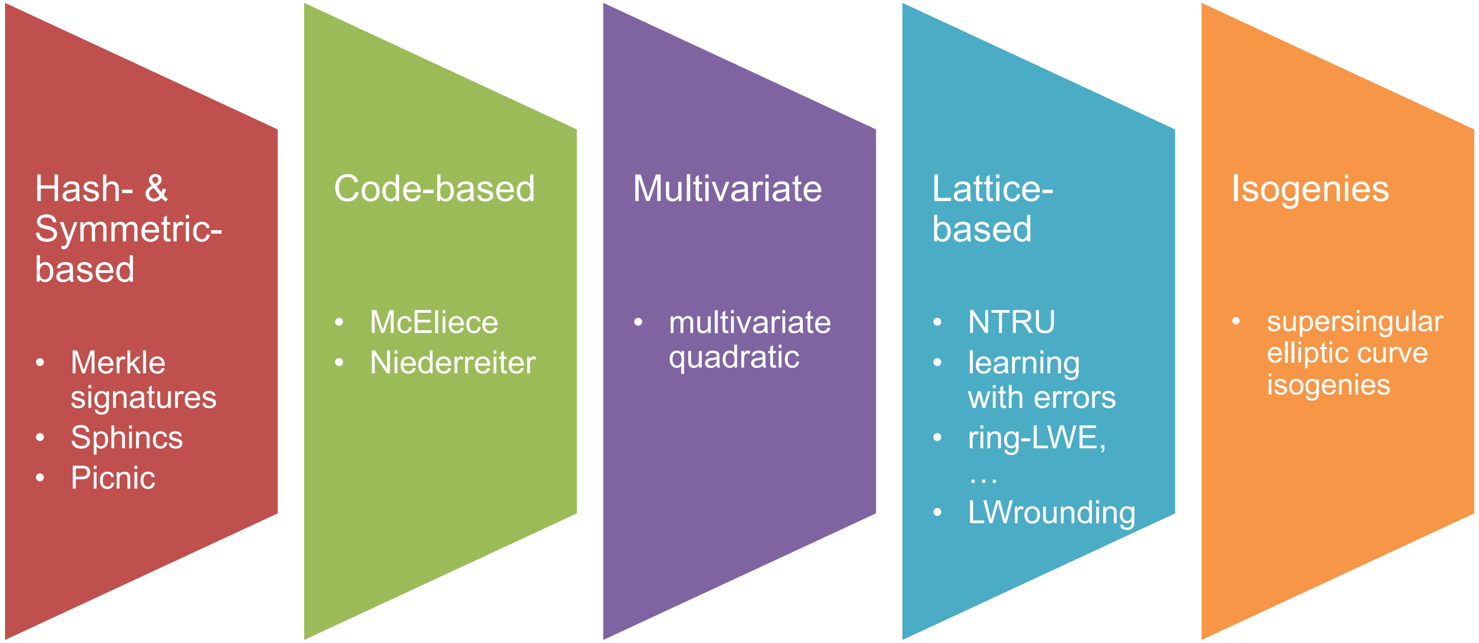
- NTRU
- learning with errors
- ring-LWE, …
- LWrounding

Isogenies

- supersingular elliptic curve isogenies

Quantum-resistant crypto
Quantum-safe crypto

Classical post-quantum crypto



Hash- & Symmetric-based

- Merkle signatures
- Sphincs
- Picnic

Code-based

- McEliece
- Niederreiter

Multivariate

- multivariate quadratic

Lattice-based

- NTRU
- learning with errors
- ring-LWE, …
- LWrounding

Isogenies

- supersingular elliptic curve isogenies

Quantum crypto

Quantum key distribution

Quantum random number generators

Quantum channels

Quantum blind computation

Families of post-quantum cryptography

Hash- & symmetric-based

- Can only be used to make signatures, not public key encryption
- Very high confidence in hash-based signatures, but large signatures required for many signature-systems

Code-based

- Long-studied cryptosystems with moderately high confidence for some code families
- Challenges in communication sizes

Multivariate quadratic

- Variety of systems with various levels of confidence and trade-offs

Lattice-based

- High level of academic interest in this field, flexible constructions
- Can achieve reasonable communication sizes
- Developing confidence

Elliptic curve isogenies

- Specialized but promising technique
- Small communication, slower computation

Learning with errors problems

Solving systems of linear equations

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

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

Solving systems of linear equations

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

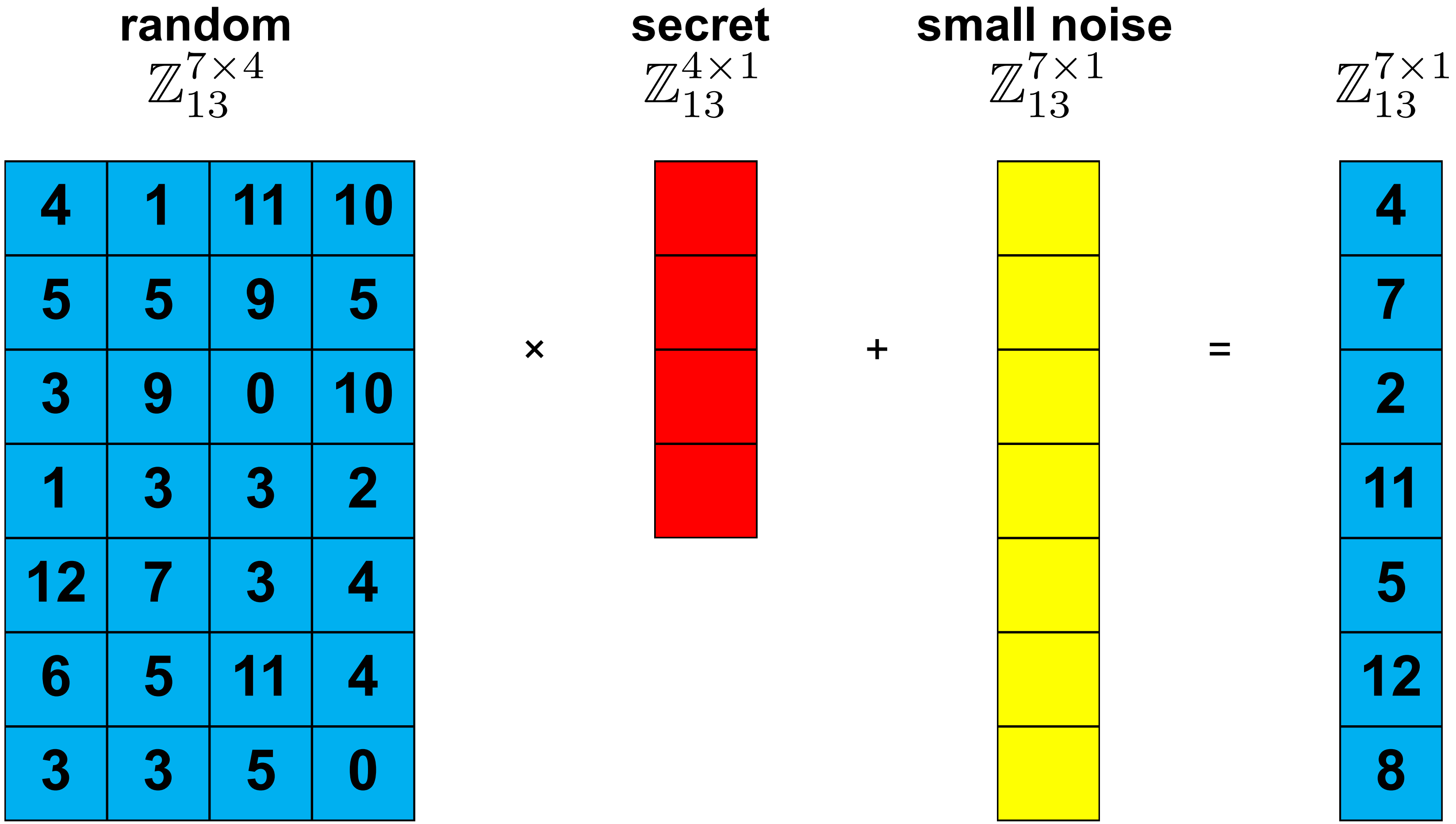
Easily solved using Gaussian elimination (Linear Algebra 101)

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

Learning with errors problem

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

Learning with errors problem



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

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

Search LWE problem

Let n, m, and q be positive integers. Let χ_s and χ_e be distributions over $\mathbb{Z}$. Let $\mathbf{s} \overset{\$}{\leftarrow} \chi_s^n$. Let $\mathbf{a}_i \overset{\$}{\leftarrow} \mathcal{U}(\mathbb{Z}_q^n)$, $e_i \overset{\$}{\leftarrow} \chi_e$, and set $b_i \leftarrow \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \mod q$, for $i = 1, \ldots, m$.

The *search LWE problem* for $(n, m, q, \chi_s, \chi_e)$ is to find s given $(\mathbf{a}_i, b_i)_{i=1}^m$.

In particular, for algorithm $\mathcal{A}$, define the advantage

$$\begin{aligned}\mathsf{Adv}^{\mathsf{lwe}}_{n,m,q,\chi_s,\chi_e}(\mathcal{A}) = \Pr\Big[&\mathbf{s} \overset{\$}{\leftarrow} \chi_s^n; \mathbf{a}_i \overset{\$}{\leftarrow} \mathcal{U}(\mathbb{Z}_q^n); e_i \overset{\$}{\leftarrow} \chi_e; \\ &b_i \leftarrow \langle \mathbf{a}_i, \mathbf{s}_i \rangle + e \bmod q : \mathcal{A}((\mathbf{a}_i, b_i)_{i=1}^m) = \mathbf{s})\Big] \ .\end{aligned}$$

[Regev STOC 2005]

Decision learning with errors problem

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

Decision LWE problem: given blue, distinguish green from random

Decision LWE problem

Let n and q be positive integers. Let χ_s and χ_e be distributions over $\mathbb{Z}$. Let $\mathbf{s} \overset{\$}{\leftarrow} \chi_s^n$. Define the following two oracles:

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

The *decision LWE problem* for (n, q, χ_s, χ_e) is to distinguish $O_{\chi,\mathbf{s}}$ from U.

In particular, for algorithm $\mathcal{A}$, define the advantage

$$\mathsf{Adv}^{\mathsf{dlwe}}_{n,q,\chi_s,\chi_e}(\mathcal{A}) = \left| \Pr(\mathbf{s} \overset{\$}{\leftarrow} \mathbb{Z}_q^n : \mathcal{A}^{O_{\chi_e,\mathbf{s}}}() = 1) - \Pr(\mathcal{A}^U() = 1) \right| .$$

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.

[Regev STOC 2005]

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.

[Applebaum et al., CRYPTO 2009]

Toy example versus real-world example

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

$$\begin{array}{|c|c|c|c|}
\hline
4 & 1 & 11 & 10 \\ \hline
5 & 5 & 9 & 5 \\ \hline
3 & 9 & 0 & 10 \\ \hline
1 & 3 & 3 & 2 \\ \hline
12 & 7 & 3 & 4 \\ \hline
6 & 5 & 11 & 4 \\ \hline
3 & 3 & 5 & 0 \\ \hline
\end{array}$$

$$\mathbb{Z}_{2^{15}}^{640\times 8}$$

$$640\left\{\overbrace{\begin{array}{ccccc}
2738 & 3842 & 3345 & 2979 & \dots \\
2896 & 595 & 3607 & & \\
377 & 1575 & & & \\
2760 & & & & \\
\dots & & & &
\end{array}}^{8}\right.$$

640 × 8 × 15 bits = **9.4 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

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

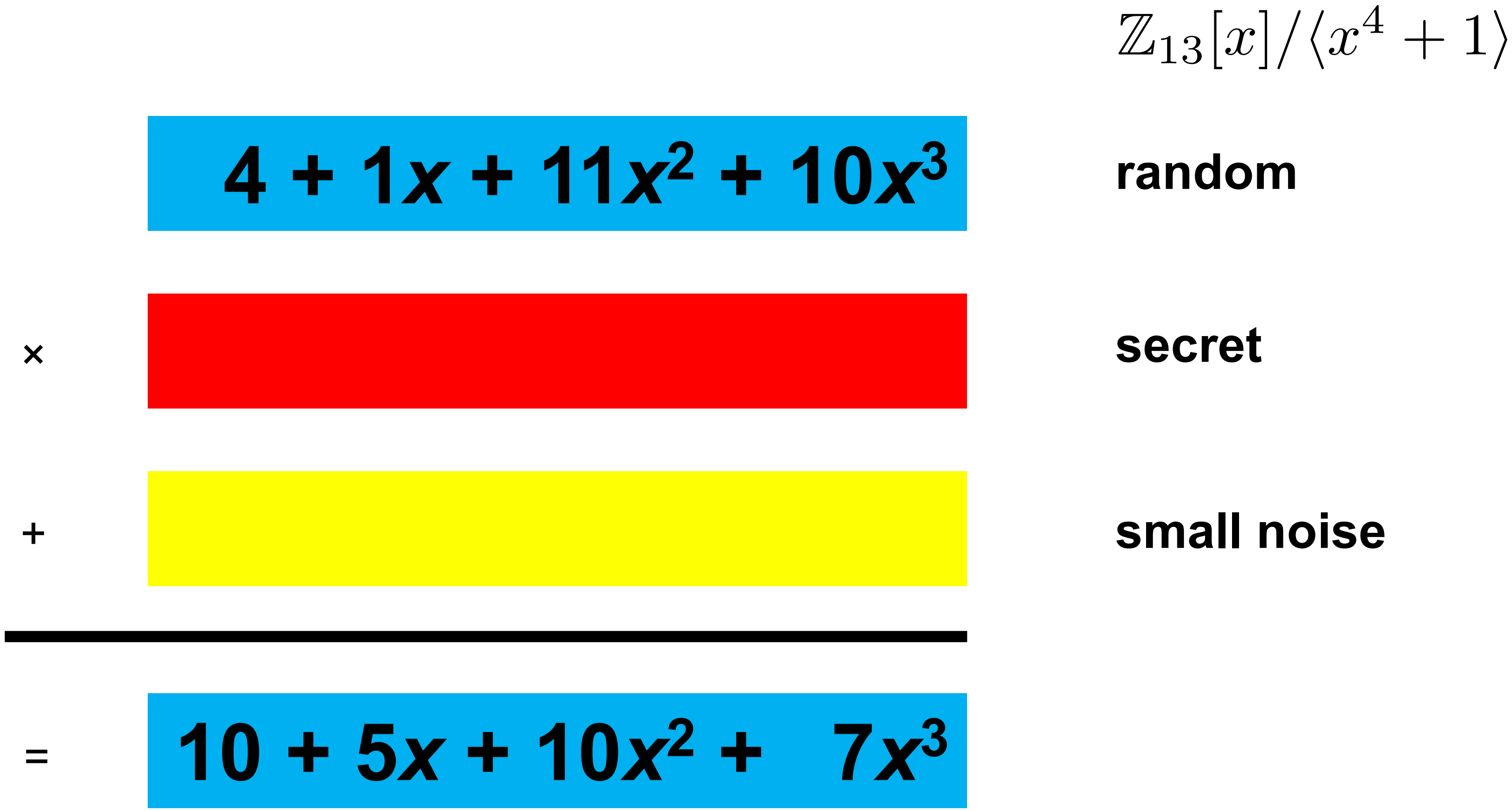
$4 + 1x + 11x^2 + 10x^3$ **random**

× $6 + 9x + 11x^2 + 11x^3$ **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**

Search ring-LWE problem

Let $R = \mathbb{Z}[X]/\langle X^n + 1 \rangle$, where n is a power of 2.

Let q be an integer, and define $R_q = R/qR$, i.e., $R_q = \mathbb{Z}_q[X]/\langle X^n + 1 \rangle$.

Let χ_s and χ_e be distributions over R_q. Let $s \overset{\$}{\leftarrow} \chi_s$. Let $a \overset{\$}{\leftarrow} \mathcal{U}(R_q)$, $e \overset{\$}{\leftarrow} \chi_e$, and set $b \leftarrow as + e$.

The *search ring-LWE problem* for (n, q, χ_s, χ_e) is to find s given (a, b).

In particular, for algorithm $\mathcal{A}$ define the advantage

$$\mathsf{Adv}^{\mathsf{rlwe}}_{n,q,\chi_s,\chi_e}(\mathcal{A}) = \Pr\left[s \overset{\$}{\leftarrow} \chi_s; a \overset{\$}{\leftarrow} \mathcal{U}(R_q); e \overset{\$}{\leftarrow} \chi_e; b \leftarrow as + e : \mathcal{A}(a, b) = s\right] .$$

[Lyubashesky, Peikert, Regev; EUROCRYPT 2010, JACM 2013]

Decision ring-LWE problem

Let n and q be positive integers. Let χ_s and χ_e be distributions over R_q. Let $s \xleftarrow{\$} \chi_s$. Define the following two oracles:

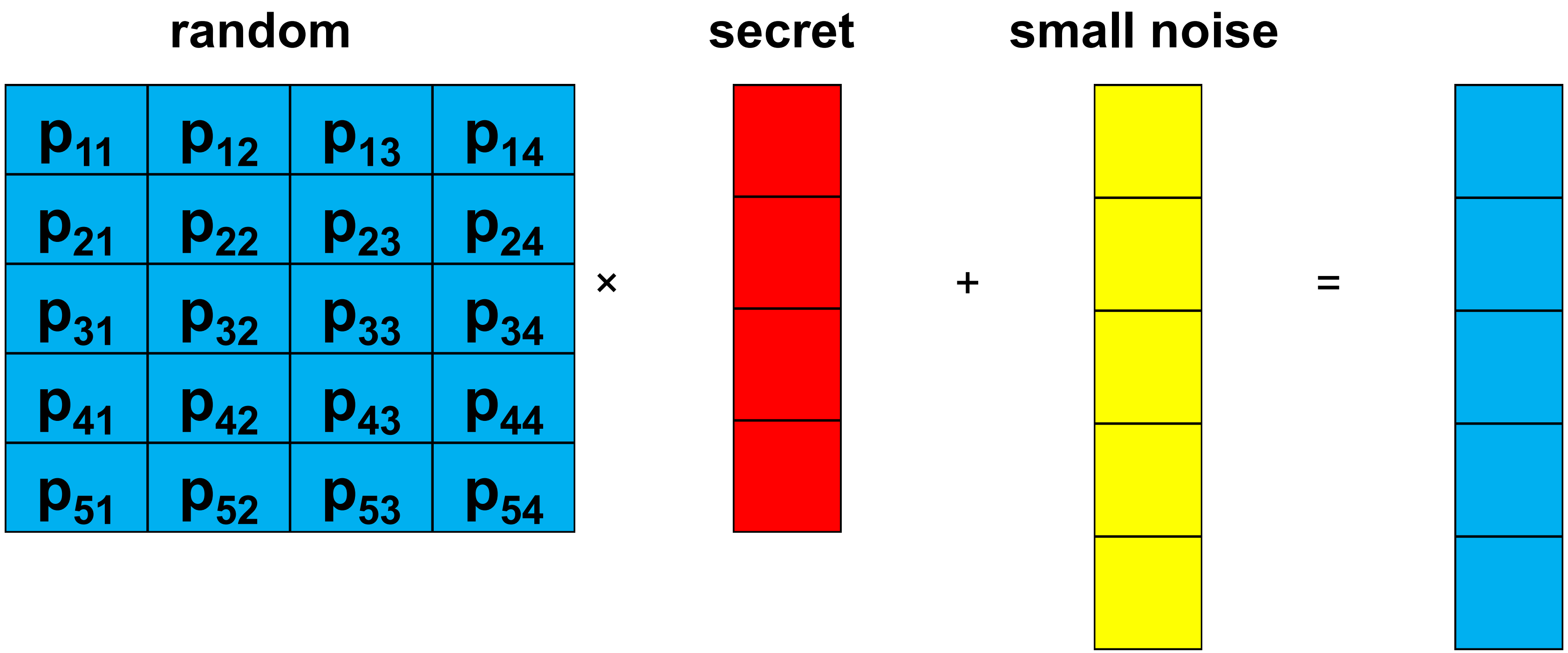
- $O_{\chi_e,s}$: $a \xleftarrow{\$} \mathcal{U}(R_q)$, $e \xleftarrow{\$} \chi_e$; return $(a, as + e)$.
- U: $a, u \xleftarrow{\$} \mathcal{U}(R_q)$; return (a, u).

The *decision ring-LWE problem* for (n, q, χ_s, χ_e) is to distinguish $O_{\chi_e,s}$ from U.

In particular, for algorithm $\mathcal{A}$, define the advantage

$$\mathsf{Adv}^{\mathsf{drlwe}}_{n,q,\chi_s,\chi_e}(\mathcal{A}) = \left| \Pr(s \xleftarrow{\$} R_q : \mathcal{A}^{O_{\chi_e,s}}() = 1) - \Pr(\mathcal{A}^{U}() = 1) \right| .$$

Module learning with errors problem



random

secret

small noise

$$\begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \\ p_{51} & p_{52} & p_{53} & p_{54} \end{pmatrix} \times \quad + \quad =$$

every matrix entry is a polynomial in $\mathbb{Z}_q[x]/(x^n+1)$

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

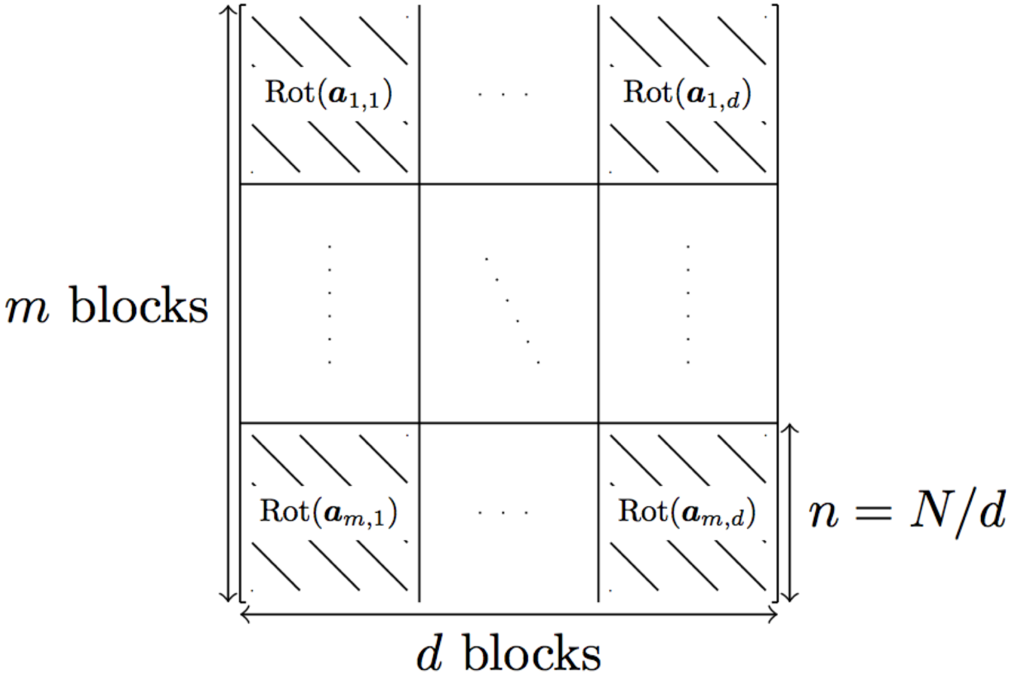
[Langlois & Stehlé, https://eprint.iacr.org/2012/090, DCC 2015]

Ring-LWE versus Module-LWE

Ring-LWE

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

Module-LWE



$\mathrm{Rot}(\boldsymbol{a}_{1,1})$
$\cdots$
$\mathrm{Rot}(\boldsymbol{a}_{1,d})$
m blocks
$\mathrm{Rot}(\boldsymbol{a}_{m,1})$
$\cdots$
$\mathrm{Rot}(\boldsymbol{a}_{m,d})$
$n = N/d$
d blocks

Figure from https://eprint.iacr.org/2012/090.pdf

Learning with rounding problem

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

$\lfloor \cdot \rceil_p : \mathbb{Z}_q \to \mathbb{Z}_p$:
Divide $\mathbb{Z}_q$ into
p equal intervals
and map x to the
index of its interval

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

[Banerjee, Peikert, Rosen EUROCRYPT 2012]

LWE versus LWR

LWE

- Noise comes from adding an explicit (Gaussian) error term

$$\langle \mathbf{a}, \mathbf{s} \rangle + e$$

LWR

- Noise comes from rounding to a smaller interval

$$\lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_p$$

- Shown to be as hard as LWE when modulus/error ratio satisfies certain bounds

https://eprint.iacr.org/2013/098, https://eprint.iacr.org/2015/769.pdf

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 $e \xleftarrow{\$} \chi$.

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.

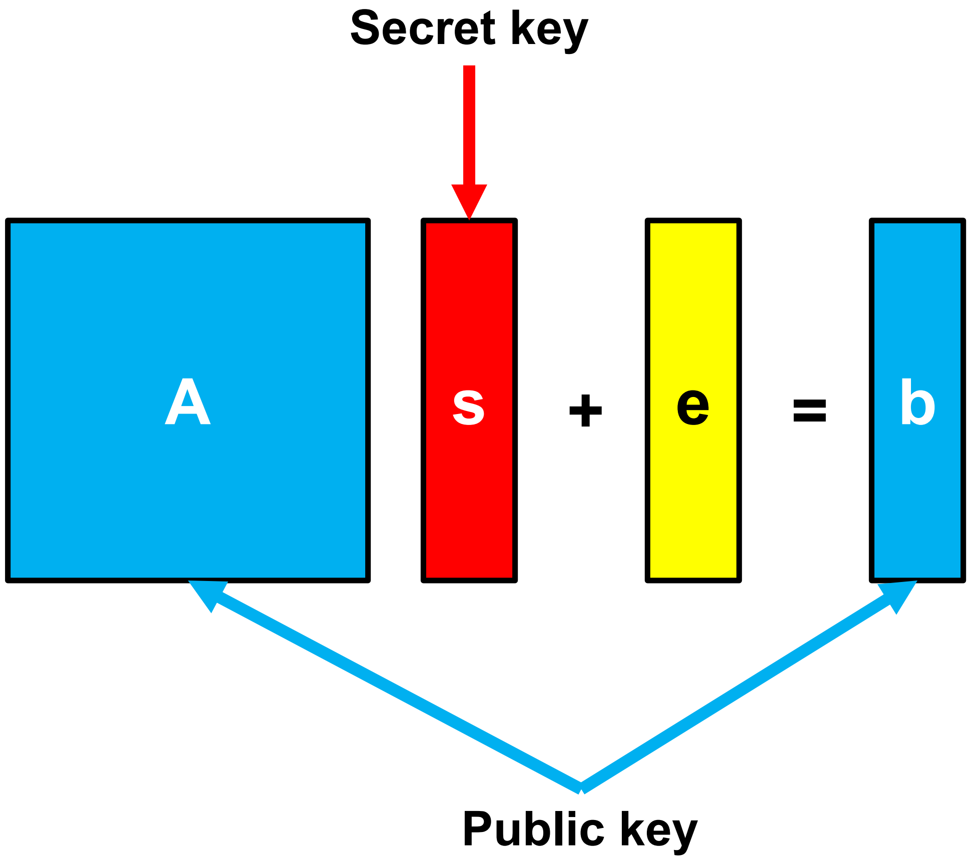
[Hoffstein, Pipher, Silverman ANTS 1998]

Problems

Learning with errors		
Module-LWE	Search	With uniform secrets
Ring-LWE		
Learning with rounding	Decision	With short secrets
NTRU problem		

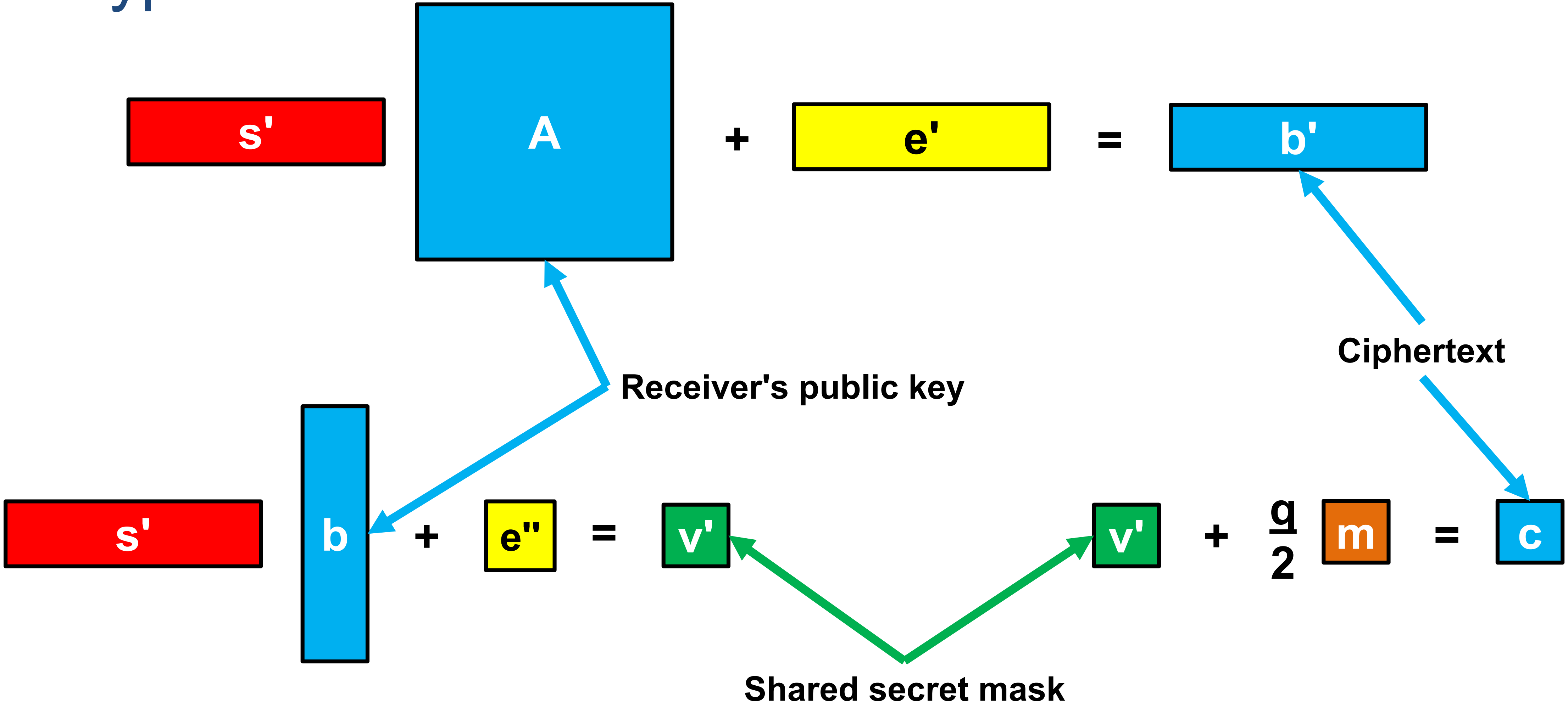
Public key encryption from LWE

Public key encryption from LWE
Key generation



[Lindner, Peikert. CT-RSA 2011]

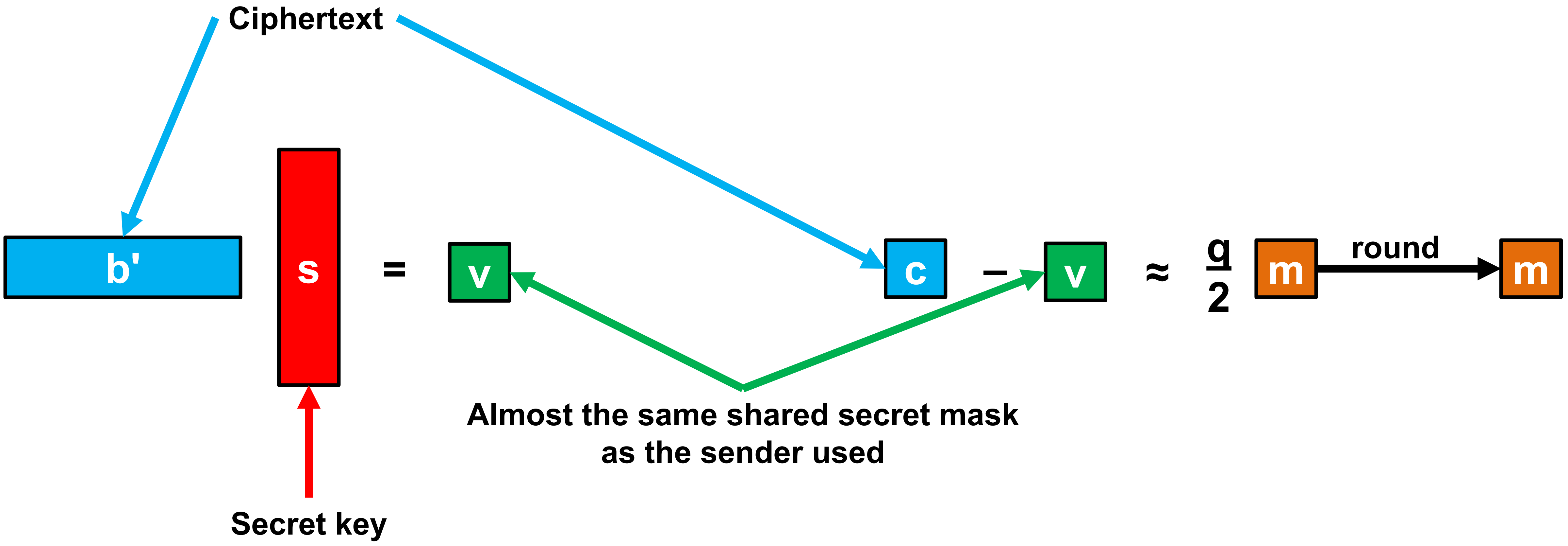
Public key encryption from LWE
Encryption



[Lindner, Peikert. CT-RSA 2011]

Public key encryption from LWE
Decryption

$$v' + \frac{q}{2} m = c$$



Ciphertext

Secret key

Almost the same shared secret mask as the sender used

[Lindner, Peikert. CT-RSA 2011]

Approximately equal shared secret

The sender uses

$$\boxed{v'} = s' (A s + e) + e''$$

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

$$\approx s' A s$$

The receiver uses

$$\boxed{v} = (s' A + e') s$$

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

$$\approx s' A s$$

Regev's public key encryption scheme

Let n, m, q, χ be LWE parameters.

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

[Regev; STOC 2005]

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]

Lindner–Peikert public key encryption

Let n, q, χ be LWE parameters.

- KeyGen(): $\mathbf{s} \overset{\$}{\leftarrow} \chi(\mathbb{Z}^n)$. $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n \times n}$. $\mathbf{e} \overset{\$}{\leftarrow} \chi(\mathbb{Z}^n)$. $\tilde{\mathbf{b}} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{e}$. Return $pk \leftarrow (\mathbf{A}, \tilde{\mathbf{b}})$ and $sk \leftarrow \mathbf{s}$.

- Enc($pk, x \in \{0,1\}$): $\mathbf{s}' \overset{\$}{\leftarrow} \chi(\mathbb{Z}^n)$. $\mathbf{e}' \overset{\$}{\leftarrow} \chi(\mathbb{Z}^n)$. $\tilde{\mathbf{b}}' \leftarrow \mathbf{s}'\mathbf{A} + \mathbf{e}'$. $e'' \overset{\$}{\leftarrow} \chi(\mathbb{Z})$. $\tilde{v}' \leftarrow \langle \mathbf{s}', \tilde{\mathbf{b}} \rangle + e''$. $c \leftarrow \text{encode}(x) + \tilde{v}'$. Return $ctxt \leftarrow (\tilde{\mathbf{b}}', c)$.

- Dec($sk, (\tilde{\mathbf{b}}', c)$): $v \leftarrow \langle \tilde{\mathbf{b}}', \mathbf{s} \rangle$. Return decode$(c - v)$.

[Lindner, Peikert; CT-RSA 2011]

Correctness

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

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

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

Difference between Regev and Lindner–Peikert

Regev:

- Bob's public key is $\mathbf{s}'\mathbf{A}$ where $\mathbf{s}' \overset{\$}{\leftarrow} \{0,1\}^m$
- Encryption mask is $\langle \mathbf{s}', \mathbf{b} \rangle$

Lindner–Peikert:

- Bob's public key is $\mathbf{s}'\mathbf{A} + \mathbf{e}'$ where $\mathbf{s}' \overset{\$}{\leftarrow} \chi_e$
- Encryption mask is $\langle \mathbf{s}', \mathbf{b} \rangle + e''$

In Regev, Bob's public key is a subset sum instance. In Lindner–Peikert, Bob's public key and encryption mask is just another LWE instance.

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]

IND-CPA security of Lindner–Peikert

Game 0: → Decision-LWE → Game 1: → Rewrite → Game 2:

Game 0:

```
$\mathbf{A} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n \times n})$
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi(\mathbb{Z}_q^n)$
$\tilde{\mathbf{b}} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{e}$
$\mathbf{s}', \mathbf{e}' \xleftarrow{\$} \chi(\mathbb{Z}_q^n)$
$\tilde{\mathbf{b}}' \leftarrow \mathbf{s}'\mathbf{A} + \mathbf{e}'$
$e'' \xleftarrow{\$} \chi(\mathbb{Z}_q)$
$\tilde{v}' \leftarrow \mathbf{s}'\tilde{\mathbf{b}} + e''$
$c_0 \leftarrow \mathrm{encode}(0) + \tilde{v}'$
$c_1 \leftarrow \mathrm{encode}(1) + \tilde{v}'$
$b^* \xleftarrow{\$} \mathcal{U}(\{0, 1\})$
**return** $(\mathbf{A}, \tilde{\mathbf{b}}, \tilde{\mathbf{b}}', c_{b^*})$
```

Game 1:

```
$\mathbf{A} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n \times n})$
$\boxed{\tilde{\mathbf{b}} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^n)}$
$\mathbf{s}', \mathbf{e}' \xleftarrow{\$} \chi(\mathbb{Z}_q^n)$
$\tilde{\mathbf{b}}' \leftarrow \mathbf{s}'\mathbf{A} + \mathbf{e}'$
$e'' \xleftarrow{\$} \chi(\mathbb{Z}_q)$
$\tilde{v}' \leftarrow \mathbf{s}'\tilde{\mathbf{b}} + e''$
$c_0 \leftarrow \mathrm{encode}(0) + \tilde{v}'$
$c_1 \leftarrow \mathrm{encode}(1) + \tilde{v}'$
$b^* \xleftarrow{\$} \mathcal{U}(\{0, 1\})$
**return** $(\mathbf{A}, \tilde{\mathbf{b}}, \tilde{\mathbf{b}}', c_{b^*})$
```

Game 2:

```
$\mathbf{A} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n \times n})$
$\tilde{\mathbf{b}} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^n)$
$\mathbf{s}' \xleftarrow{\$} \chi(\mathbb{Z}_q^n)$
$\boxed{[\mathbf{e}' \| e''] \xleftarrow{\$} \chi(\mathbb{Z}_q^{n+1})}$
$\boxed{[\tilde{\mathbf{b}}' \| \tilde{v}'] \leftarrow \mathbf{s}'[\mathbf{A} \| \tilde{\mathbf{b}}] + [\mathbf{e}' \| e'']}$
$c_0 \leftarrow \mathrm{encode}(0) + \tilde{v}'$
$c_1 \leftarrow \mathrm{encode}(1) + \tilde{v}'$
$b^* \xleftarrow{\$} \mathcal{U}(\{0, 1\})$
**return** $(\mathbf{A}, \tilde{\mathbf{b}}, \tilde{\mathbf{b}}', c_{b^*})$
```

[Lindner, Peikert; CT-RSA 2011]

IND-CPA security of Lindner–Peikert

Game 2:

```
$\mathbf{A} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n\times n})$
$\tilde{\mathbf{b}} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n})$
$\mathbf{s}' \xleftarrow{\$} \chi(\mathbb{Z}_q^{n})$
$\boxed{[\mathbf{e}' \| e''] \xleftarrow{\$} \chi(\mathbb{Z}_q^{n+1})}$
$\boxed{[\tilde{\mathbf{b}}' \| \tilde{v}'] \leftarrow \mathbf{s}'[\mathbf{A} \| \tilde{\mathbf{b}}] + [\mathbf{e}' \| e'']}$
$c_0 \leftarrow \mathrm{encode}(0) + \tilde{v}'$
$c_1 \leftarrow \mathrm{encode}(1) + \tilde{v}'$
$b^* \xleftarrow{\$} \mathcal{U}(\{0,1\})$
$\mathbf{return}$ $(\mathbf{A}, \tilde{\mathbf{b}}, \tilde{\mathbf{b}}', c_{b^*})$
```

→ Decision-LWE →

Game 3:

```
$\mathbf{A} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n\times n})$
$\tilde{\mathbf{b}} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n})$
$\boxed{[\tilde{\mathbf{b}}' \| \tilde{v}'] \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n+1})}$
$c_0 \leftarrow \mathrm{encode}(0) + \tilde{v}'$
$c_1 \leftarrow \mathrm{encode}(1) + \tilde{v}'$
$b^* \xleftarrow{\$} \mathcal{U}(\{0,1\})$
$\mathbf{return}$ $(\mathbf{A}, \tilde{\mathbf{b}}, \tilde{\mathbf{b}}', c_{b^*})$
```

→ Rewrite →

Game 4:

```
$\mathbf{A} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n\times n})$
$\tilde{\mathbf{b}} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n})$
$[\tilde{\mathbf{b}}' \| \tilde{v}'] \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n+1})$
$b^* \xleftarrow{\$} \mathcal{U}(\{0,1\})$
$\mathbf{return}$ $(\mathbf{A}, \tilde{\mathbf{b}}, \tilde{\mathbf{b}}', \tilde{v}')$
```

Independent of hidden bit

[Lindner, Peikert; CT-RSA 2011]

Lattice-based KEM/PKEs submitted to NIST

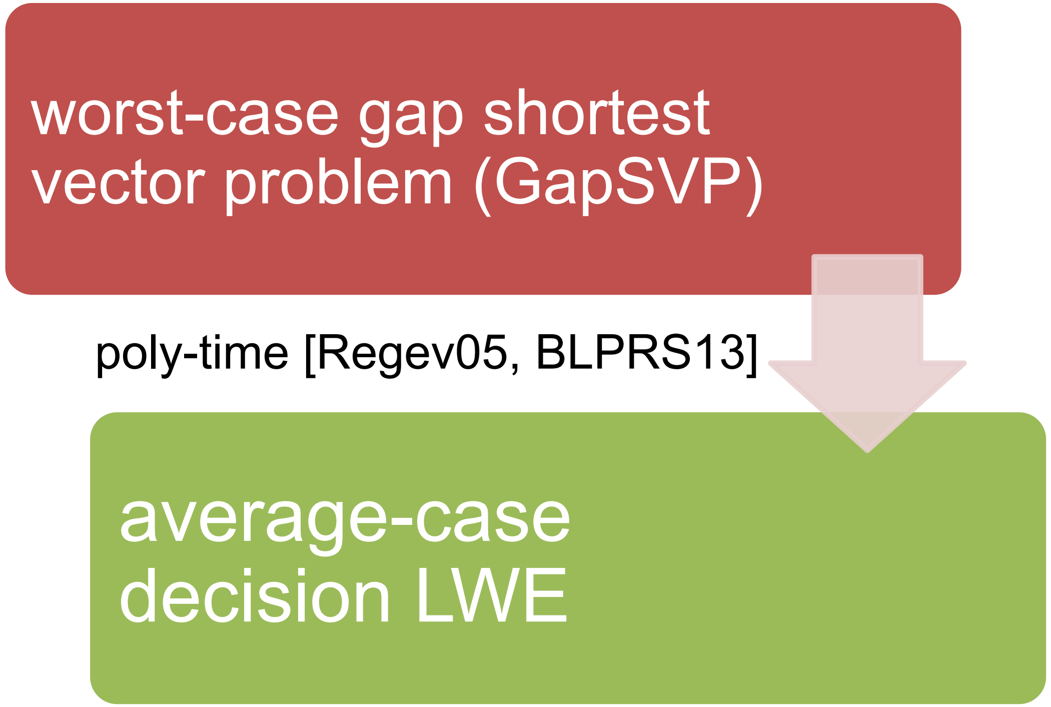
- BabyBear, MamaBear, PapaBear (ILWE)
- CRYSTALS-Kyber (MLWE)
- Ding Key Exchange (RLWE)
- Emblem (LWE, RLWE)
- FrodoKEM (LWE)
- HILA5 (RLWE)
- KCL (MLWE, RLWE)
- KINDI (MLWE)
- LAC (PLWE)
- LIMA (RLWE)
- Lizard (LWE, LWR, RLWE, RLWR)
- Lotus (LWE)
- NewHope (RLWE)
- NTRU Prime (RLWR)
- NTRU HRSS (NTRU)
- NTRUEncrypt (NTRU)
- Round2 (RLWR, LWR)
- Saber (MLWR)
- Titanium (PLWE)

https://estimate-all-the-lwe-ntru-schemes.github.io/docs/

Security of LWE-based cryptography

"Lattice-based"

Hardness of decision LWE – "lattice-based"



worst-case gap shortest vector problem (GapSVP)

poly-time [Regev05, BLPRS13]

average-case decision LWE

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**

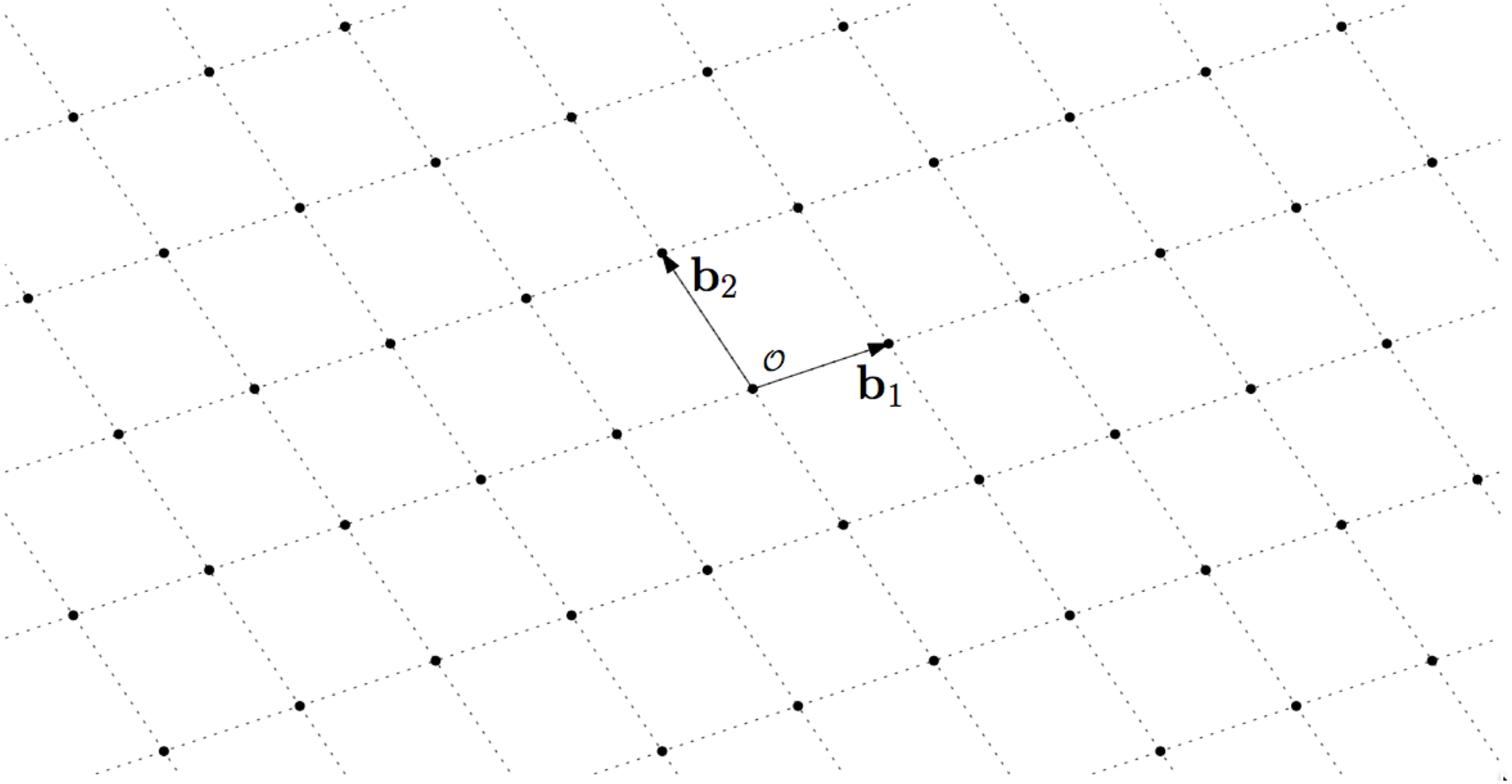
$$\mathcal{L} = \mathcal{L}(\mathbf{B}) = \left\{ \sum_{i=1}^{n} z_i \mathbf{b}_i : z_i \in \mathbb{Z} \right\} .$$

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

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

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

Lattices

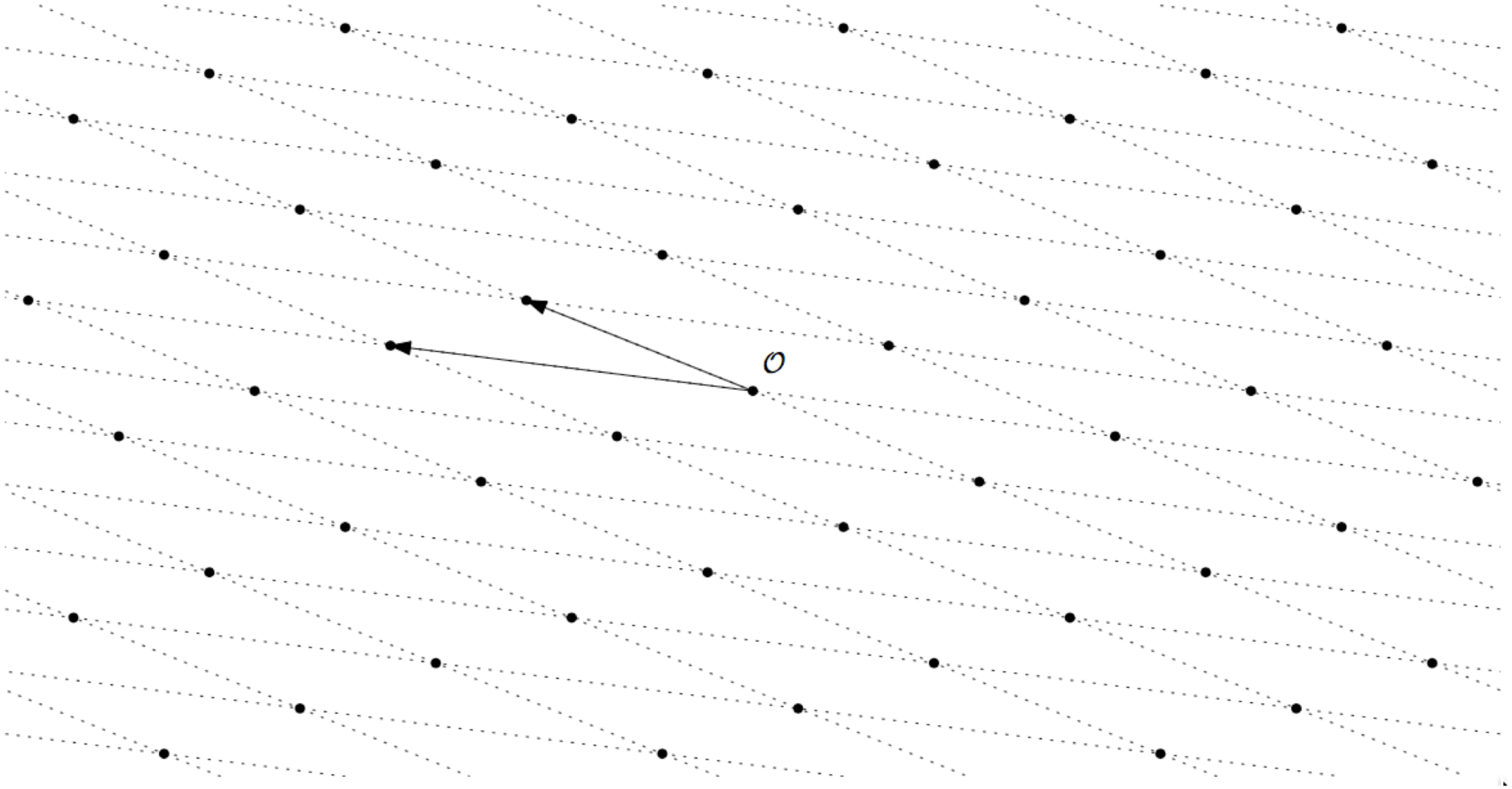


Discrete additive subgroup of $\mathbb{Z}^n$

Equivalently, integer linear combinations of a basis

Diagram from http://www.cs.bris.ac.uk/pgrad/csjhvdp/files/bkz.pdf

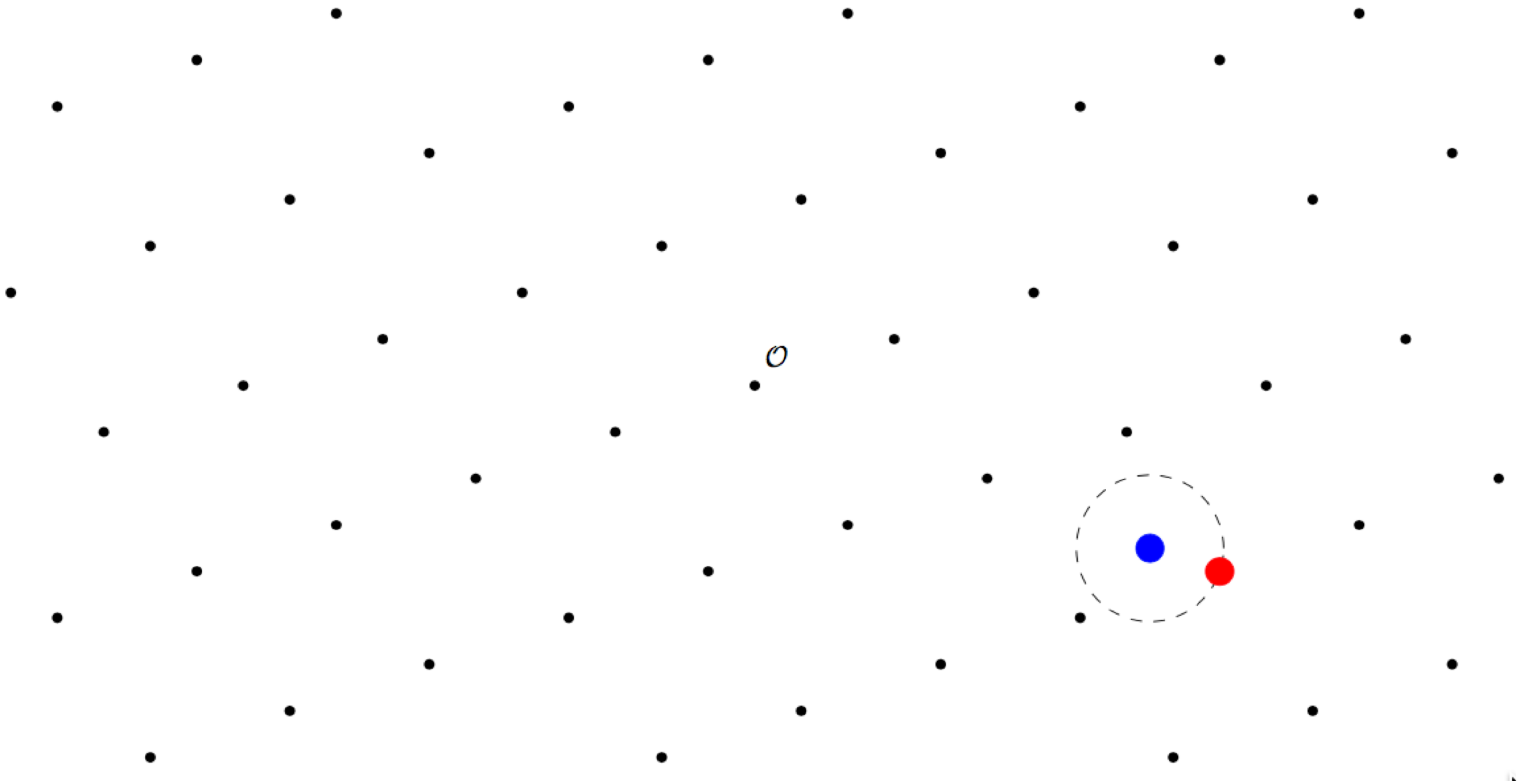
Lattices



There are many bases for the same lattice – some short and orthogonalish, some long and acute.

Diagram from http://www.cs.bris.ac.uk/pgrad/csjhvdp/files/bkz.pdf

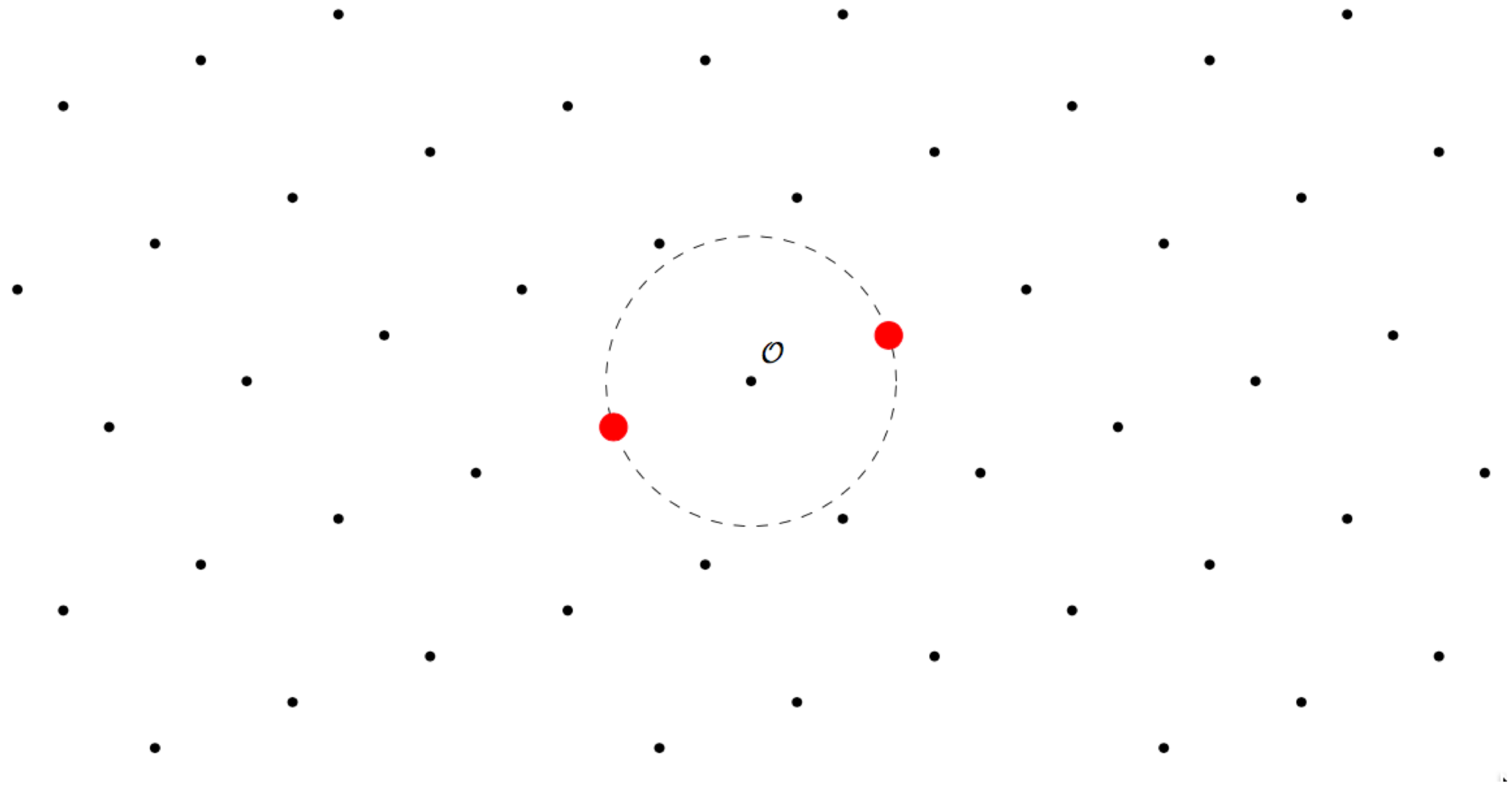
Closest vector problem



Given some basis for the lattice and a target point in the space, find the closest lattice point.

Diagram from http://www.cs.bris.ac.uk/pgrad/csjhvdp/files/bkz.pdf

Shortest vector problem



Given some basis for the lattice, find the shortest non-zero lattice point.

Diagram from http://www.cs.bris.ac.uk/pgrad/csjhvdp/files/bkz.pdf

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 $q \leq 2^{\mathrm{poly}(n)}$ 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 $m = \mathrm{poly}(n)$ 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]

Finding short vectors in lattices

LLL basis reduction algorithm

- Finds a basis close to Gram–Schmidt
- Polynomial runtime (in dimension), but basis quality (shortness/orthogonality) is poor

Block Korkine Zolotarev (BKZ) algorithm

- Trade-off between runtime and basis quality
- In practice the best algorithm for cryptographically relevant scenarios

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 ∩ 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
    - Reductions are typically non-tight (e.g., n^{13}); would lead to very large parameters
- Based on cryptanalysis:
  - Ignore tightness in reductions.
  - Pick parameters based on best known LWE solvers relying on lattice solvers.

KEMs and key agreement from LWE

Key encapsulation mechanisms (KEMs)

A ***key encapsulation mechanism (KEM)*** consists of three algorithms:

- $\mathrm{KeyGen}() \overset{\$}{\rightarrow} (pk, sk)$: A key generation algorithm that outputs a public key pk and secret key sk
- $\mathrm{Encaps}(pk) \overset{\$}{\rightarrow} (c, k)$ An encapsulation algorithm that outputs a ciphertext c and session key k
- $\mathrm{Decaps}(sk, c) \rightarrow k$: A decapsulation algorithm that outputs a session key k (or an error symbol)

Security properties for KEMs: IND-CPA, IND-CCA

Key exchange protocols

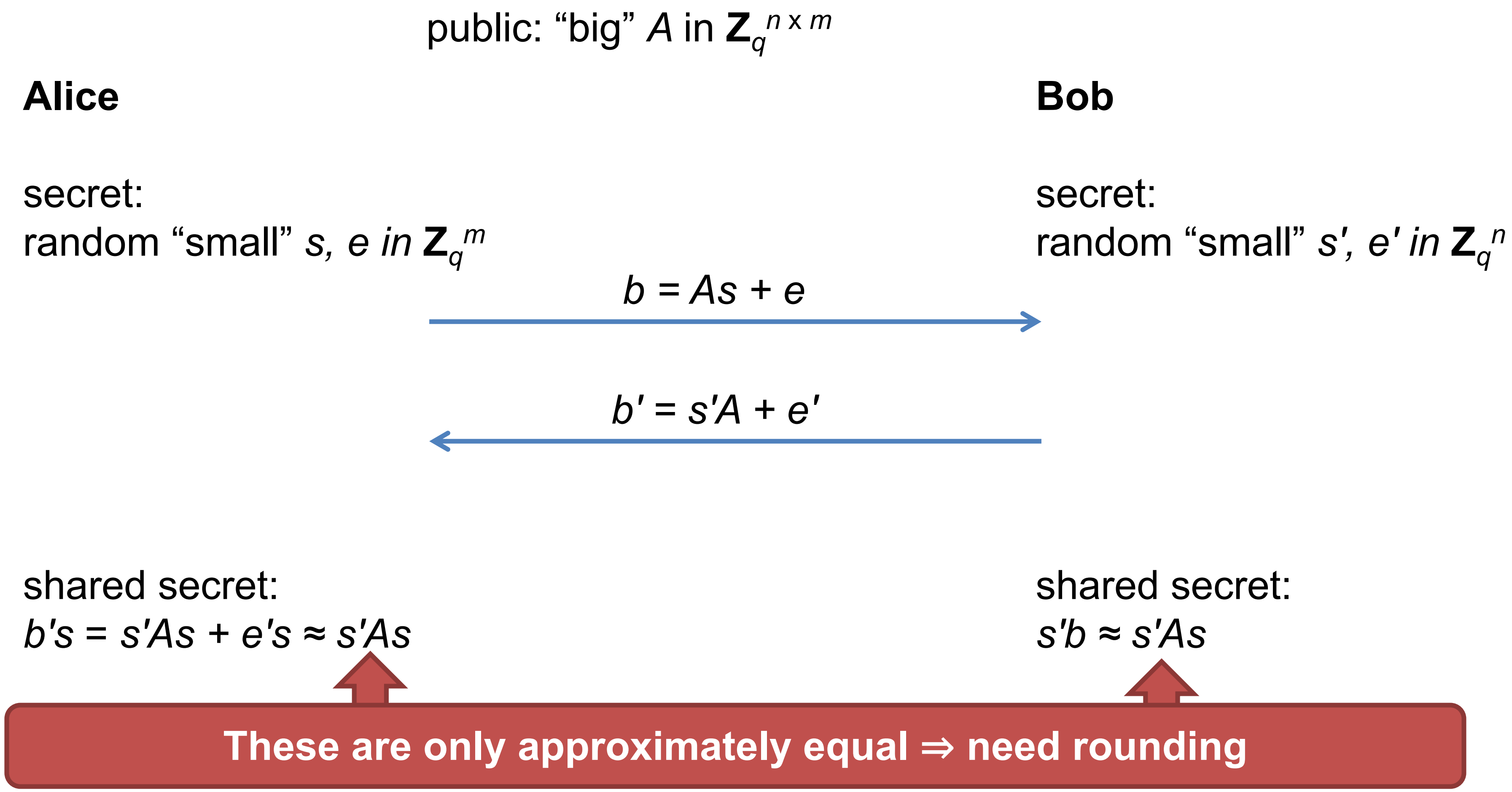
- A **key exchange protocol** is an interactive protocol carried out between two parties.
- The goal of the protocol is to output a session key that is indistinguishable from random.

- In **authenticated** key exchange protocols, the adversary can be active and controls all communications between parties; the parties are assumed to have authentically distributed trusted long-term keys out of band prior to the protocol.
- In **unauthenticated** key exchange protocols, the adversary can be passive and only obtains transcripts of communications between honest parties.

- IND-CPA KEMs can be viewed as a two flow unauthenticated key exchange protocol.

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

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

Bob

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

$$\xrightarrow{b = As + e}$$

$$\xleftarrow{b' = s'A + e'}$$

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 & reconciliation

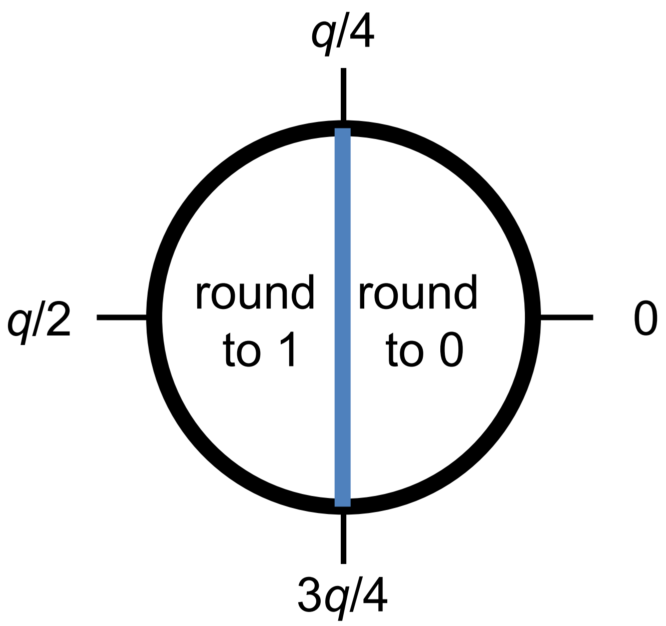
- Each coefficient of the polynomial is an integer modulo *q*
- Treat each coefficient independently
- Send a "reconciliation signal" to help with rounding

- 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

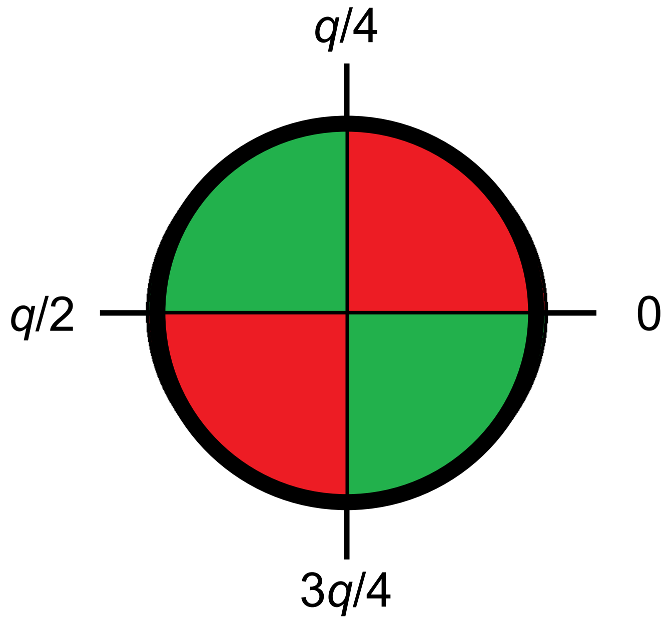


This works most of the time: prob. failure 2^{-10}.

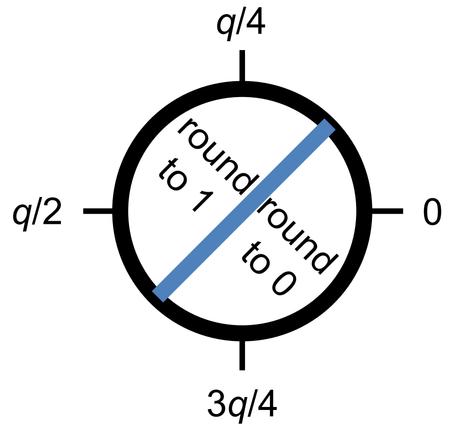
Not good enough: we need exact key agreement.

Rounding and reconciliation (Peikert)

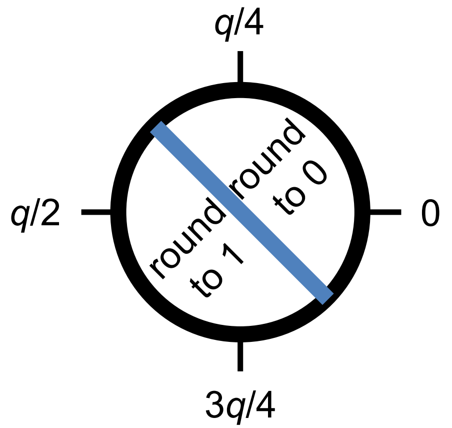
Bob says which of two regions the value is in: 🟩 or 🟥



If 🟩



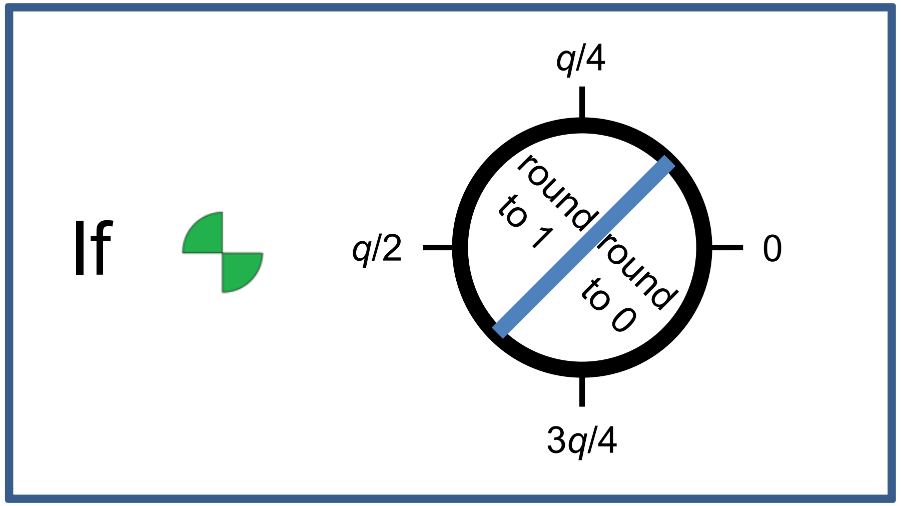
If 🟥

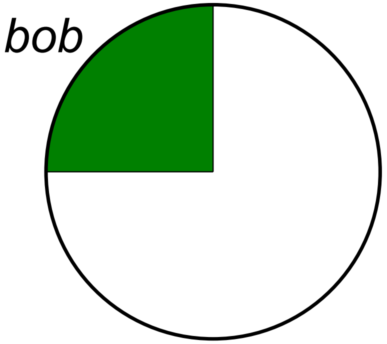


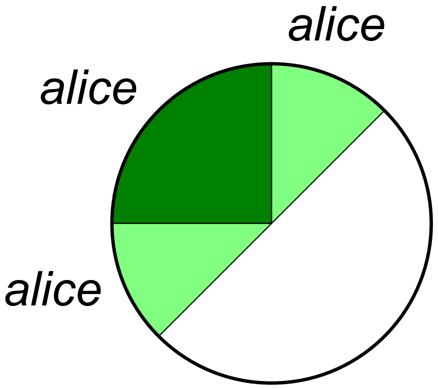
[Peikert; PQCrypto 2014]

Rounding and reconciliation (Peikert)

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



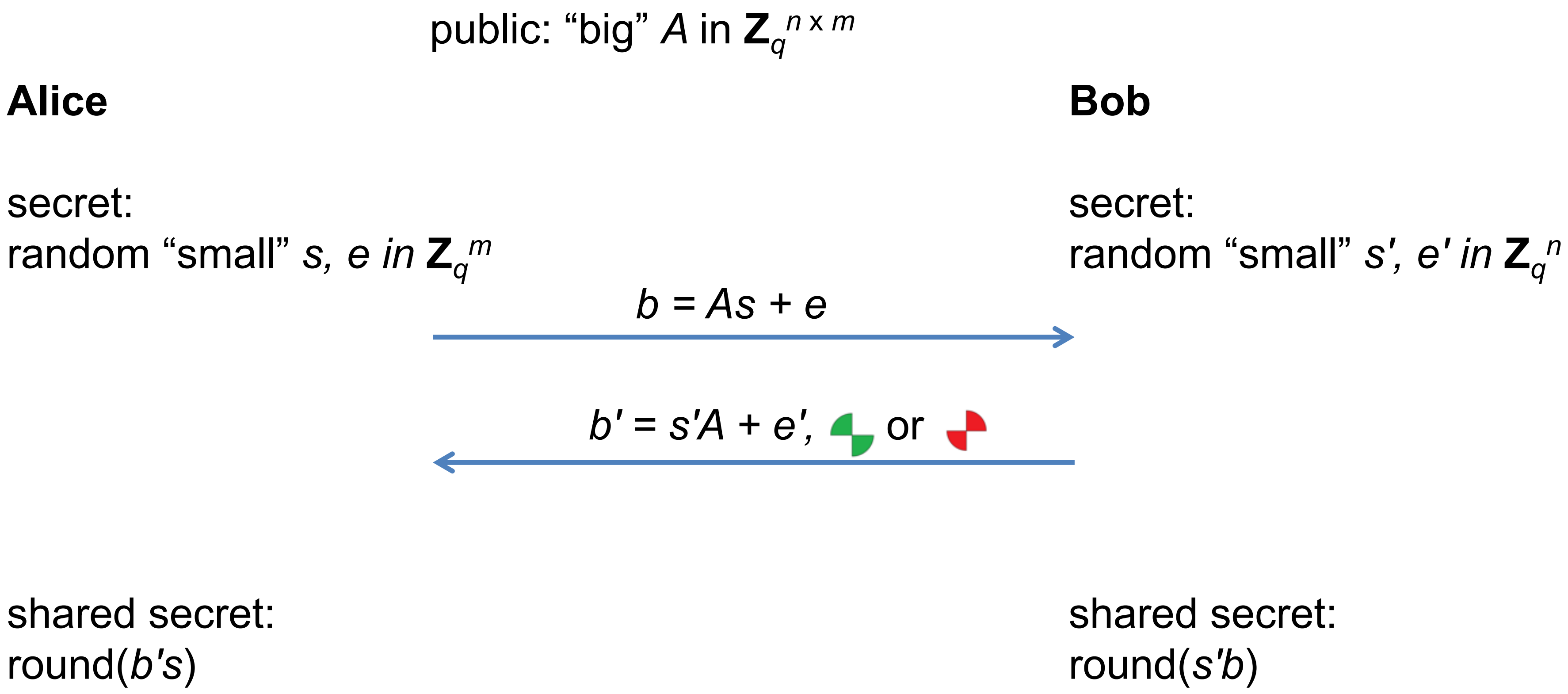




- Security not affected: revealing 🟩 or 🟥 leaks no information

[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 $\mathbf{Z}_q^n$

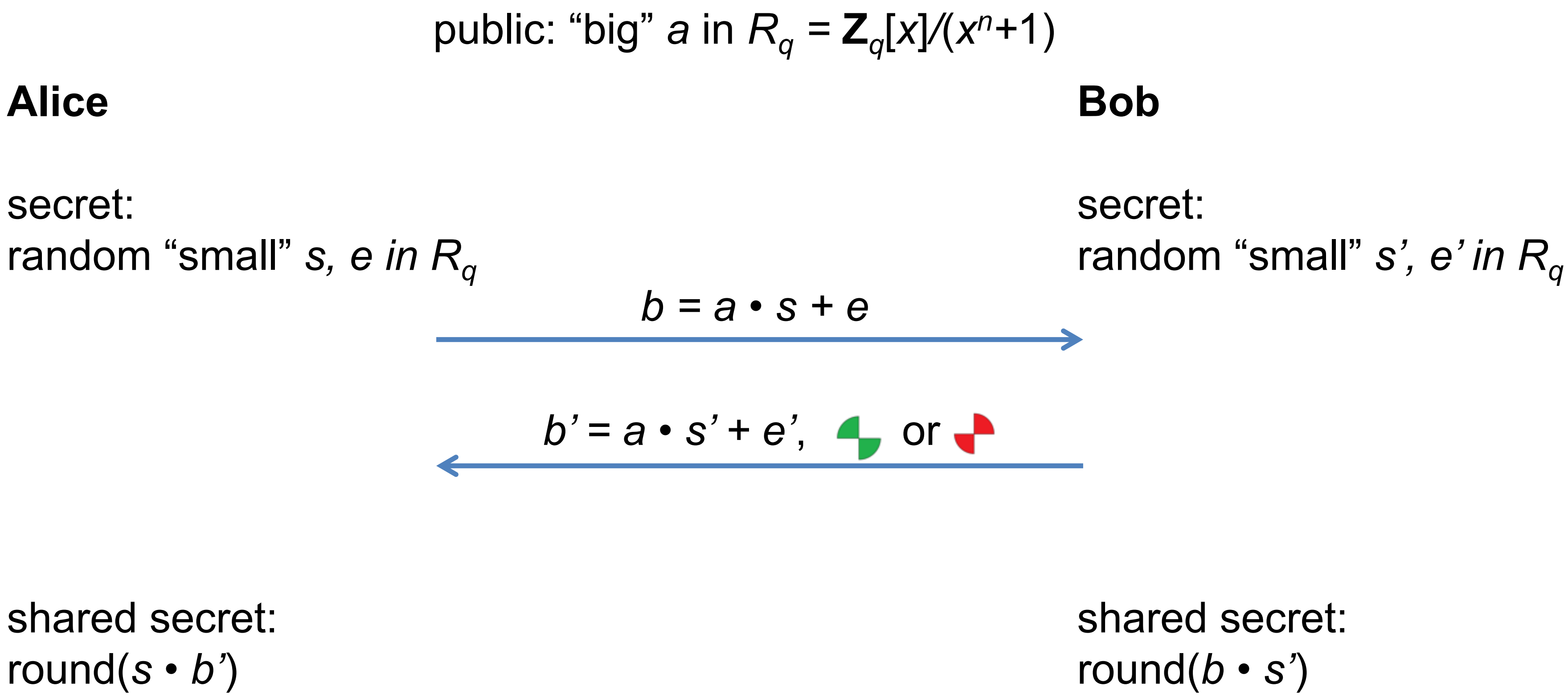
$b = As + e$

$b' = s'A + e'$, 🟩 or 🟥

shared secret:
round($b's$)

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

$b = a \cdot s + e$ →

← $b' = a \cdot s' + e'$, 🟢 or 🔴

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

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

Public key validation

- **No public key validation possible** for basic LWE/ring-LWE public keys

- **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 passive security (IND-CPA)
- Or construct actively secure (IND-CCA) KEM/PKE/AKE using Fujisaki–Okamoto transform or quantum-resistant variant [Targhi–Unruh, TCC 2016] [Hofheinz et al., Eprint 2017]

An example: FrodoKEM

- KEM: Key encapsulation mechanism (simplified key exchange protocol)
- Builds on basic (IND-CPA) LWE public key encryption
- Achieves IND-CCA security against adaptive adversaries
  - By applying a quantum-resistant variant of the Fujisaki–Okamoto transform
- Negligible error rate

- Simple design:
  - Free modular arithmetic ($q = 2^{16}$)
  - Simple Gaussian sampling
  - Parallelizable matrix-vector operations
  - No reconciliation
  - Simple to code

[Bos, Costello, Ducas, Mironov, Naehrig, Nikolaenko, Raghunathan, Stebila. ACM CCS 2016]
[Alkim, Bos, Ducas, Easterbrook, LaMacchia, Longa, Mironov, Naehrig, Nikolaenko, Peikert, Raghunathan, Stebila. FrodoKEM NIST Submission, 2017]

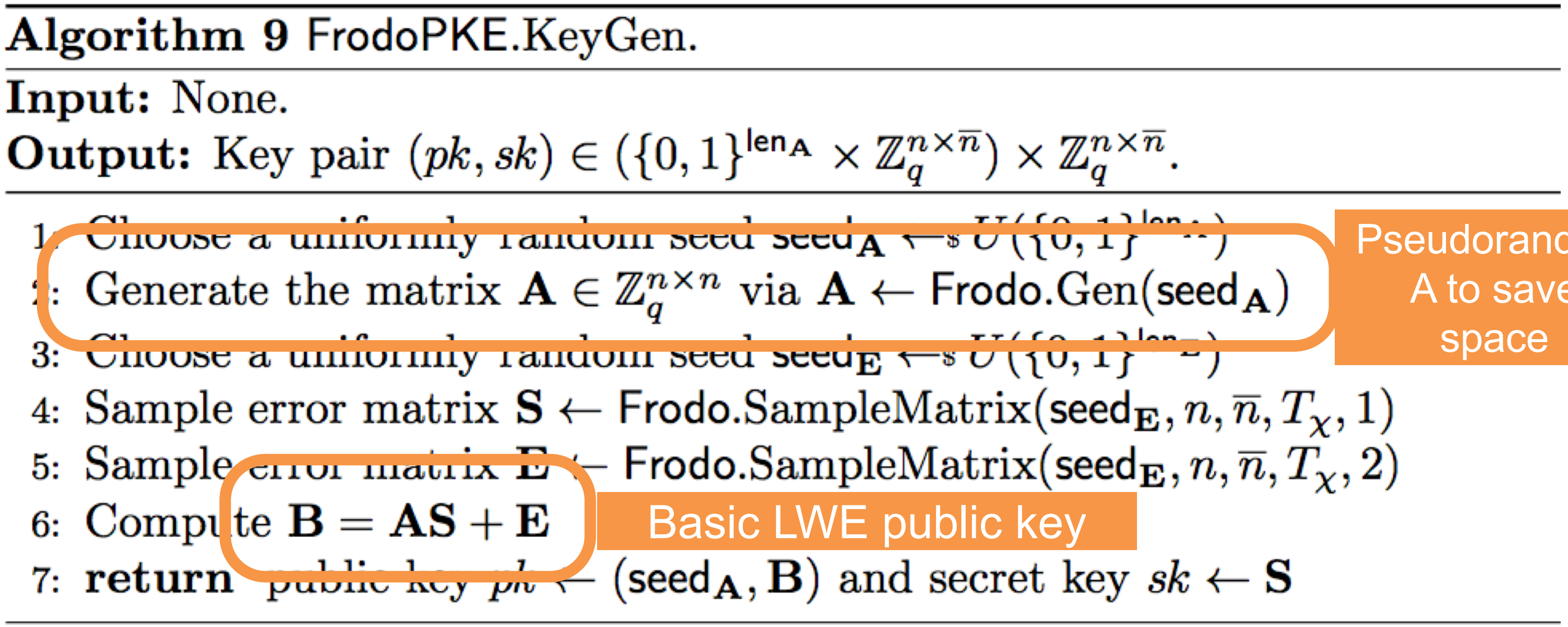
FrodoKEM construction

IND-CPA secure FrodoPKE

FrodoPKE.KeyGen

FrodoPKE.Enc

FrodoPKE.Dec



Algorithm 9 FrodoPKE.KeyGen.

Input: None.

Output: Key pair $(pk, sk) \in (\{0,1\}^{\mathsf{len}_\mathbf{A}} \times \mathbb{Z}_q^{n \times \overline{n}}) \times \mathbb{Z}_q^{n \times \overline{n}}$.

```
Choose a uniformly random seed $\mathsf{seed}_\mathbf{A} \leftarrow_\$ U(\{0,1\}^{\mathsf{len}_\mathbf{A}})$
Generate the matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$ via $\mathbf{A} \leftarrow \mathsf{Frodo.Gen}(\mathsf{seed}_\mathbf{A})$
Choose a uniformly random seed $\mathsf{seed}_\mathbf{E} \leftarrow_\$ U(\{0,1\}^{\mathsf{len}_\mathbf{E}})$
Sample error matrix $\mathbf{S} \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_\mathbf{E}, n, \overline{n}, T_\chi, 1)$
Sample error matrix $\mathbf{E} \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_\mathbf{E}, n, \overline{n}, T_\chi, 2)$
Compute $\mathbf{B} = \mathbf{AS} + \mathbf{E}$
return public key $pk \leftarrow (\mathsf{seed}_\mathbf{A}, \mathbf{B})$ and secret key $sk \leftarrow \mathbf{S}$
```

Pseudorandom A to save space

Basic LWE public key

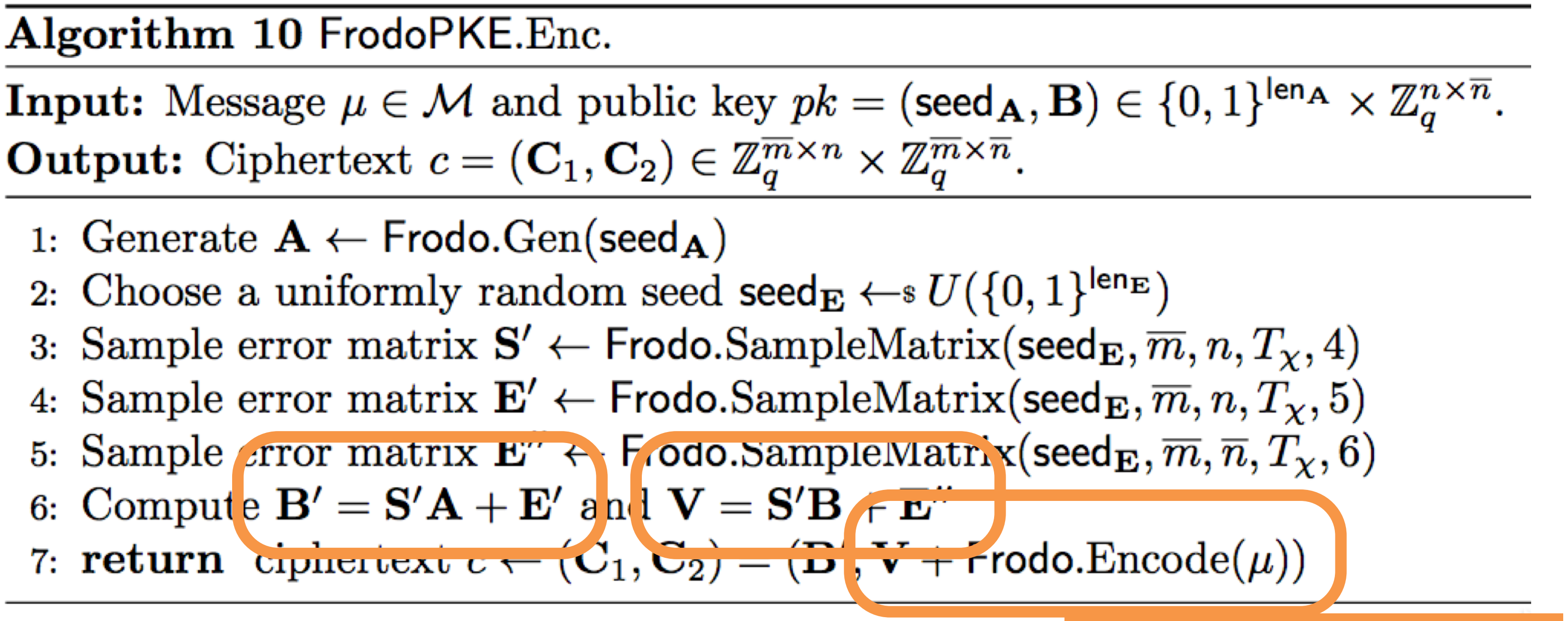
FrodoKEM construction

IND-CPA secure FrodoPKE

FrodoPKE.KeyGen

FrodoPKE.Enc

FrodoPKE.Dec



Algorithm 10 FrodoPKE.Enc.

Input: Message $\mu \in \mathcal{M}$ and public key $pk = (\mathsf{seed}_{\mathbf{A}}, \mathbf{B}) \in \{0,1\}^{\mathsf{len}_{\mathbf{A}}} \times \mathbb{Z}_q^{n \times \overline{n}}$.
Output: Ciphertext $c = (\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{\overline{m} \times n} \times \mathbb{Z}_q^{\overline{m} \times \overline{n}}$.

```
Generate $\mathbf{A} \leftarrow \mathsf{Frodo.Gen}(\mathsf{seed}_{\mathbf{A}})$
Choose a uniformly random seed $\mathsf{seed}_{\mathbf{E}} \leftarrow\!\!\$\; U(\{0,1\}^{\mathsf{len}_{\mathbf{E}}})$
Sample error matrix $\mathbf{S}' \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_{\mathbf{E}}, \overline{m}, n, T_\chi, 4)$
Sample error matrix $\mathbf{E}' \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_{\mathbf{E}}, \overline{m}, n, T_\chi, 5)$
Sample error matrix $\mathbf{E}'' \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_{\mathbf{E}}, \overline{m}, \overline{n}, T_\chi, 6)$
Compute $\mathbf{B}' = \mathbf{S}'\mathbf{A} + \mathbf{E}'$ and $\mathbf{V} = \mathbf{S}'\mathbf{B} + \mathbf{E}''$
return ciphertext $c \leftarrow (\mathbf{C}_1, \mathbf{C}_2) = (\mathbf{B}', \mathbf{V} + \mathsf{Frodo.Encode}(\mu))$
```

Basic LWE ciphertext

Shared secret

Key transport using public key encryption

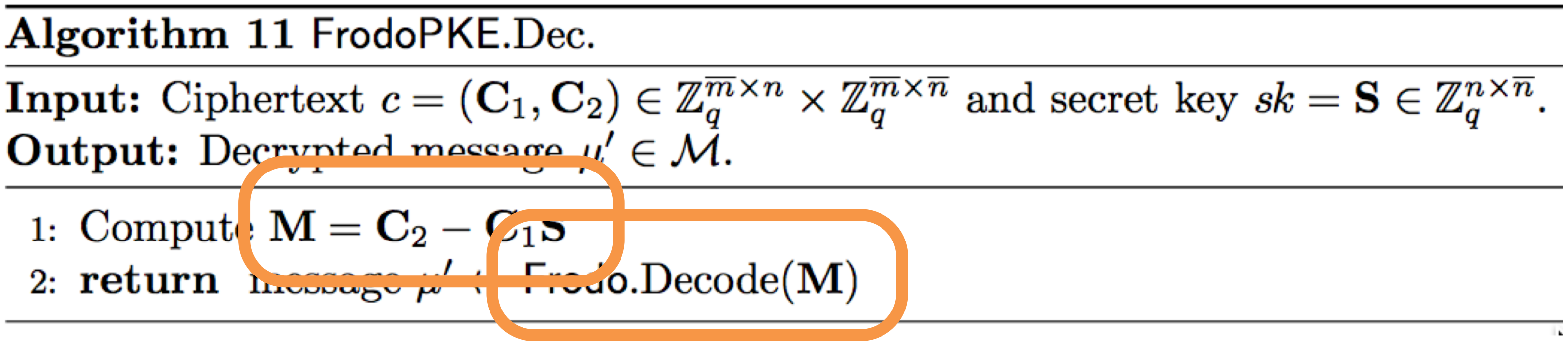
FrodoKEM construction

IND-CPA secure FrodoPKE

FrodoPKE.KeyGen

FrodoPKE.Enc

FrodoPKE.Dec

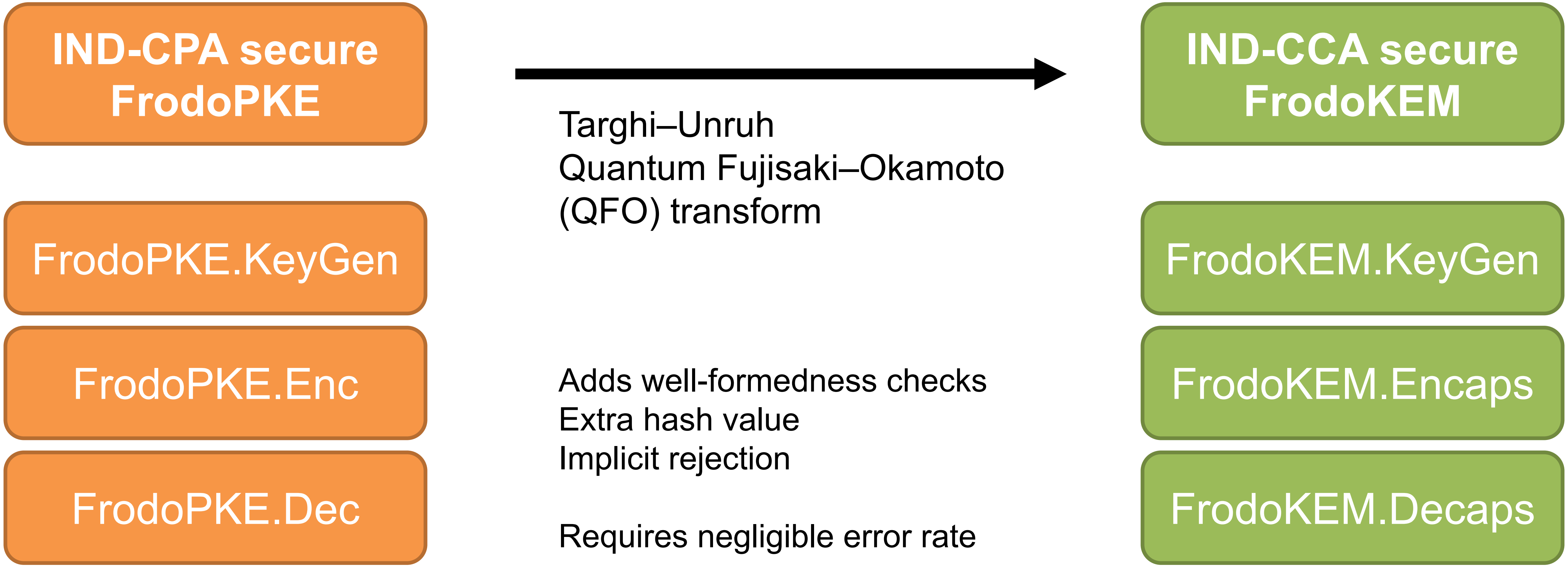


Algorithm 11 FrodoPKE.Dec.

Input: Ciphertext $c = (\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{\overline{m}\times n} \times \mathbb{Z}_q^{\overline{m}\times\overline{n}}$ and secret key $sk = \mathbf{S} \in \mathbb{Z}_q^{n\times\overline{n}}$.
Output: Decrypted message $\mu' \in \mathcal{M}$.

```
Compute $\mathbf{M} = \mathbf{C}_2 - \mathbf{C}_1\mathbf{S}$
return message $\mu' \leftarrow$ Frodo.Decode($\mathbf{M}$)
```

FrodoKEM construction



IND-CPA secure FrodoPKE

FrodoPKE.KeyGen

FrodoPKE.Enc

FrodoPKE.Dec

Targhi–Unruh
Quantum Fujisaki–Okamoto
(QFO) transform

Adds well-formedness checks
Extra hash value
Implicit rejection

Requires negligible error rate

IND-CCA secure FrodoKEM

FrodoKEM.KeyGen

FrodoKEM.Encaps

FrodoKEM.Decaps

FrodoKEM parameters

	FrodoKEM-640	FrodoKEM-976
Dimension n	640	976
Modulus q	2^{15}	2^{16}
Error distribution	Approx. Gaussian [-11, ..., 11], σ = 2.75	Approx. Gaussian [-10, ..., 10], σ = 2.3
Failure probability	2^{-148}	2^{-199}
Ciphertext size	9,736 bytes	15,768 bytes
Estimated security (cryptanalytic)	2^{143} classical 2^{103} quantum	2^{209} classical 2^{150} quantum
Runtime	1.1 msec	2.1 msec

Other applications of LWE

Fully homomorphic encryption from LWE

- KeyGen(): $\mathbf{s} \xleftarrow{\$} \chi(\mathbb{Z}_q^n)$
- Enc($sk, \mu \in \mathbb{Z}_2$): Pick $\mathbf{c} \in \mathbb{Z}_q^n$ such that $\langle \mathbf{s}, \mathbf{c} \rangle = e \bmod q$ where $e \in \mathbb{Z}$ satisfies $e \equiv \mu \bmod 2$.
- Dec($sk, \mathbf{c}$): Compute $\langle \mathbf{s}, \mathbf{c} \rangle \in \mathbb{Z}_q$, represent this as $e \in \mathbb{Z} \cap [-\frac{q}{2}, \frac{q}{2})$. Return $\mu' \leftarrow e \bmod 2$.

[Brakerski, Vaikuntanathan; FOCS 2011]

Fully homomorphic encryption from LWE

$\mathbf{c}_1 + \mathbf{c}_2$ encrypts $\mu_1 + \mu_2$:

$$\langle \mathbf{s}, \mathbf{c}_1 + \mathbf{c}_2 \rangle = \langle \mathbf{s}, \mathbf{c}_1 \rangle + \langle \mathbf{s}, \mathbf{c}_2 \rangle = e_1 + e_2 \bmod q$$

Decryption will work as long as the error $e_1 + e_2$ remains below $q/2$.

[Brakerski, Vaikuntanathan; FOCS 2011]

Fully homomorphic encryption from LWE

Let $\mathbf{c}_1 \otimes \mathbf{c}_2 = (c_{1,i} \cdot c_{2,j})_{i,j} \in \mathbb{Z}_q^{n^2}$ be the tensor product (or Kronecker product).

$\mathbf{c}_1 \otimes \mathbf{c}_2$ is the encryption of $\mu_1 \mu_2$ under secret key $\mathbf{s} \otimes \mathbf{s}$:

$$\langle \mathbf{s} \otimes \mathbf{s}, \mathbf{c}_1 \otimes \mathbf{c}_2 \rangle = \langle \mathbf{s}, \mathbf{c}_1 \rangle \cdot \langle \mathbf{s}, \mathbf{c}_2 \rangle = e_1 \cdot e_2 \bmod q$$

Decryption will work as long as the error $e_1 \cdot e_2$ remains below $q/2$.

[Brakerski, Vaikuntanathan; FOCS 2011]

Fully homomorphic encryption from LWE

- Error conditions mean that the number of additions and multiplications is limited.
- Multiplication increases the dimension (exponentially), so the number of multiplications is again limited.

- There are techniques to resolve both of these issues.
  - **Key switching** allows converting the dimension of a ciphertext.
  - **Modulus switching** and **bootstrapping** are used to deal with the error rate.

Digital signatures [Lyubashevsky 2011]

- KeyGen(): $\mathbf{S} \xleftarrow{\$} \{-d, \ldots, 0, \ldots, d\}^{m \times k}$, $A \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{T} \leftarrow \mathbf{AS}$.
  Secret key: $\mathbf{S}$; public key: $(\mathbf{A}, \mathbf{T})$.

- Sign($\mathbf{S}, \mu$): $\mathbf{y} \xleftarrow{\$} \chi^m$; $\mathbf{c} \leftarrow H(\mathbf{Ay}, \mu)$; $\mathbf{z} \leftarrow \mathbf{Sc} + \mathbf{y}$.
  With prob. $p(\mathbf{z})$ output $(\mathbf{z}, \mathbf{c})$, else restart Sign. "Rejection sampling"

- Vfy($(\mathbf{A}, \mathbf{T}), \mu, (\mathbf{z}, \mathbf{c})$): Accept iff $\|\mathbf{z}\| \leq \eta\sigma\sqrt{m}$ and $\mathbf{c} = H(\mathbf{Az} - \mathbf{Tc}, \mu)$

[Lyubashevsky; Eurocrypt 2012]

Lattice-based signature schemes submitted to NIST

- CRYSTALS-Dilithium (MLWE)
- Falcon (NTRU)
- pqNTRUsign (NTRU)
- qTESLA (RLWE)

https://estimate-all-the-lwe-ntru-schemes.github.io/docs/

Post-quantum security models

Post-quantum security models

- Is the adversary quantum?
- If so, at what stage(s) in the security experiment?
- If so, can the adversary interact with honest parties (make queries) quantumly?
- If so, and if the proof is in the random oracle model, can the adversary access the random oracle quantumly?

Public key encryption security models

IND-CCA

- A is classical

$\underline{\mathrm{Exp}_{\Pi}^{\mathrm{ind-cca}}(\mathcal{A})}$

1. $(pk, sk) \leftarrow_{\$} \mathrm{KeyGen}()$
2. $(m_0, m_1, st) \leftarrow_{\$} \mathcal{A}^{\mathrm{Enc}(pk,\cdot),\mathrm{Dec}(sk,\cdot)}(pk)$
3. $b \leftarrow_{\$} \{0,1\}$
4. $c^* \leftarrow_{\$} \mathrm{Enc}(pk, m_b)$
5. $b' \leftarrow_{\$} \mathcal{A}^{\mathrm{Enc}(pk,\cdot),\mathrm{Dec}(sk,\cdot\neq c^*)}(st, c^*)$

Quantum security models

- "Future quantum"
  - A is quantum in line 5 but always has only classical access to Enc and Dec
- "Post-quantum"
  - A is quantum in lines 2 and 5 but always has only classical access to Enc & Dec
- "Fully quantum"
  - A is quantum in lines 2 and 5 and has quantum (superposition) access to Enc and Dec

Symmetric crypto generally quantum-resistant, unless in fully quantum security models.
[Kaplan et al., CRYPTO 2016]

Quantum random oracle model

- If the adversary is locally quantum (e.g., future quantum, post-quantum), should the adversary be able to query its random oracle quantumly?
  - No: We imagine the adversary only interacting classically with the honest system.
  - Yes: The random oracle model artificially makes the adversary interact with something (a hash function) that can implement itself in practice, so the adversary could implement it quantumly.
    - QROM seems to be prevalent these days

- Proofs in QROM often introduce tightness gap
  - QROM proofs of Fujisaki–Okamoto transform from IND-CPA PKE to IND-CCA PKE very hot topic right now

[Boneh et al, ASIACRYPT 2011 https://eprint.iacr.org/2010/428]

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 scenarios
  - Is a problem for other scenarios

- 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 post-quantum key exchange

TLS 1.2

- Prototypes and software experiments:
  - Bos, Costello, Naehrig, Stebila, S&P 2015
  - Bos, Costello, Ducas, Mironov, Naehrig, Nikolaenko, Raghunathan, Stebila, ACM CCS 2016
  - Google Chrome experiment
    - https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html
    - https://www.imperialviolet.org/2016/11/28/cecpq1.html
  - liboqs OpenSSL fork
    - https://openquantumsafe.org/
  - Microsoft OpenVPN fork
    - https://www.bleepingcomputer.com/news/microsoft/microsoft-adds-post-quantum-cryptography-to-an-openvpn-fork/

TLS 1.3

- Prototypes:
  - liboqs OpenSSL fork
    - https://github.com/open-quantum-safe/openssl/tree/OQS-master
- Internet drafts:
  - Whyte et al.
    - https://tools.ietf.org/html/draft-whyte-qsh-tls13-06
  - Shank and Stebila
    - https://tools.ietf.org/html/draft-schanck-tls-additional-keyshare-00

Hybrid signatures

X.509 certificates

- How to convey multiple public keys & signatures in a single certificate?
- Proposal: second certificate in X.509 extension
- Experimental study of backward compatibility

Theory

- Properties of different combiners for multiple signature schemes
- Hierarchy of security notions based on quantumness of adversary

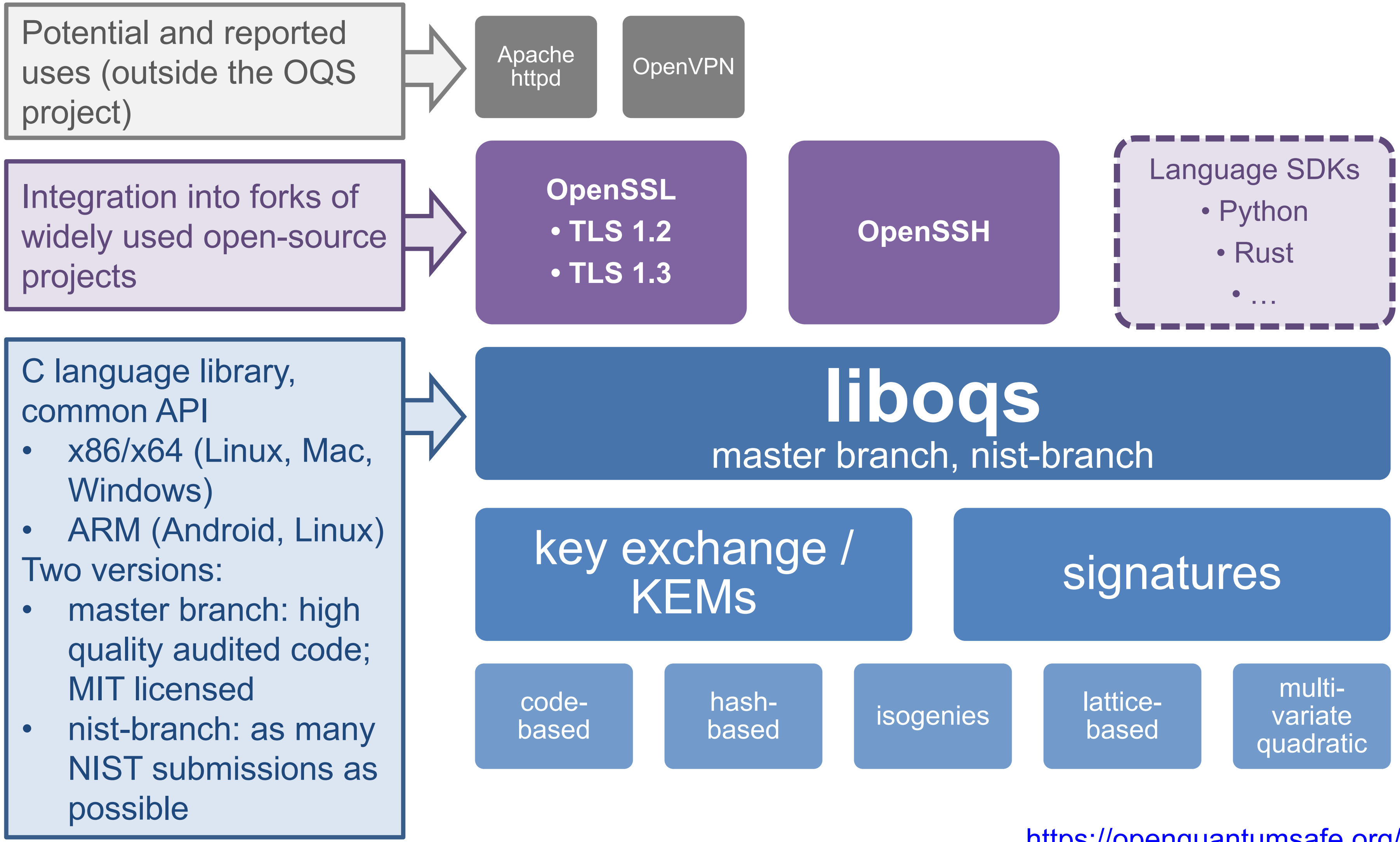
[Bindel, Herath, McKague, Stebila. PQCrypto 2017]

OPEN QUANTUM SAFE

software for prototyping quantum-resistant cryptography

https://openquantumsafe.org/

Open Quantum Safe Project



Potential and reported uses (outside the OQS project)

Apache httpd

OpenVPN

Integration into forks of widely used open-source projects

OpenSSL
- TLS 1.2
- TLS 1.3

OpenSSH

Language SDKs
- Python
- Rust
- …

C language library, common API
- x86/x64 (Linux, Mac, Windows)
- ARM (Android, Linux)

Two versions:
- master branch: high quality audited code; MIT licensed
- nist-branch: as many NIST submissions as possible

liboqs

master branch, nist-branch

key exchange / KEMs

signatures

code-based

hash-based

isogenies

lattice-based

multi-variate quadratic

https://openquantumsafe.org/, https://github.com/open-quantum-safe/

Summary

Summary

- Intro to post-quantum cryptography
- Learning with errors problems
  - LWE, Ring-LWE, Module-LWE, Learning with Rounding, NTRU
  - Search, decision
  - With uniform secrets, with short secrets
- Public key encryption from LWE
  - Regev
  - Lindner–Peikert
- Security of LWE
  - Lattice problems – GapSVP
- KEMs and key agreement from LWE
- Other applications of LWE
- PQ security models
- Transitioning to PQ crypto

More reading

- Post-Quantum Cryptography
  by Bernstein, Buchmann, Dahmen

- A Decade of Lattice Cryptography
  by Chris Peikert
  https://web.eecs.umich.edu/~cpeikert/pubs/lattice-survey.pdf

- NIST Post-quantum Cryptography Project
  http://nist.gov/pqcrypto