Post-quantum cryptography from the learning with errors problem

Douglas Stebila

UNIVERSITY OF WATERLOO

2021 IEEE North American School of Information Theory • 2021-06-22

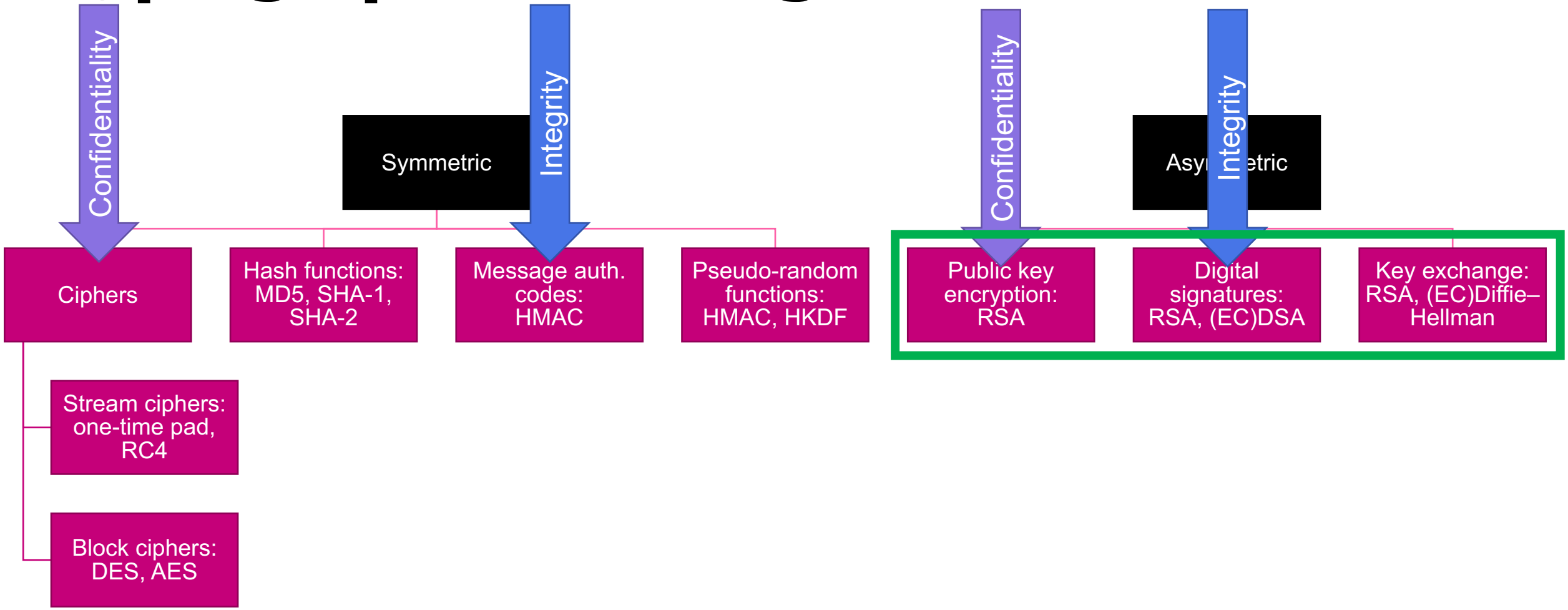
Outline

1. **Background**
   a) Public key / private key cryptography
   b) Post-quantum motivation
2. **Learning with errors problems**
3. **Cryptography from LWE**
   a) Public key encryption / key exchange
   b) Digital signatures
   c) Advanced constructions
4. **Difficulty of LWE**
   a) Lattice problems
   b) Cryptanalysis
5. **Standardization of PQ cryptography**

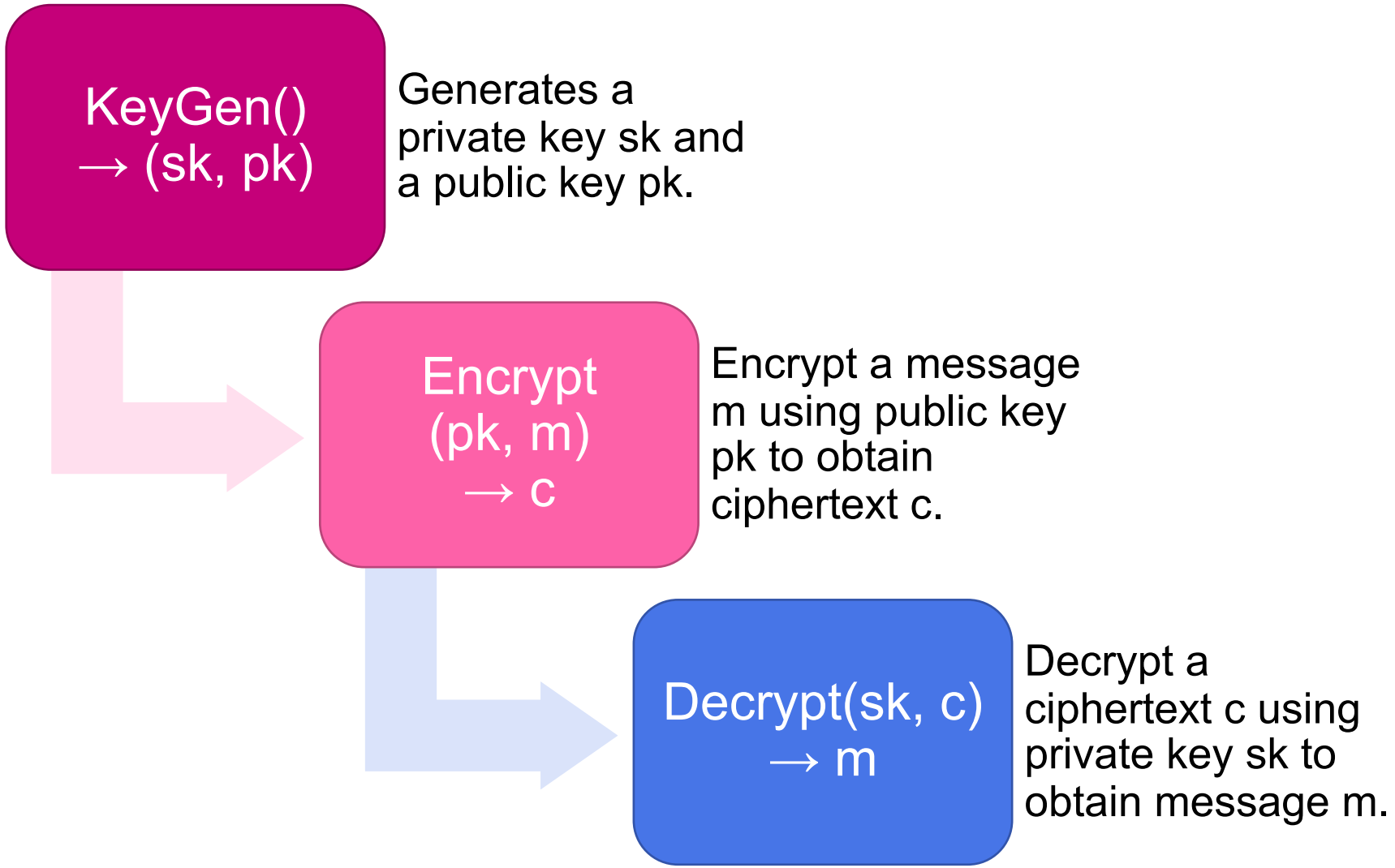
1. Background

Public key / private key cryptography

Cryptographic Building Blocks



Public Key Encryption



Public Key Encryption: Security

Security goal: indistinguishability under adaptive chosen ciphertext attack (IND-CCA2)

Adaptive chosen ciphertext attack

- adversary can adaptively obtain decryptions of any ciphertexts of his choosing

Indistinguishability

- the adversary cannot distinguish which of two messages m_0 or m_1 of its choosing was encrypted

RSA public key encryption

Key generation

1. Bob picks two large primes p and q and computes n = pq
2. Bob picks a value e and computes $d = e^{-1} \bmod \phi(n)$
3. Public key: (n, e)
4. Secret key: (n, d)

Encryption

To encrypt a message to send to Bob:

1. Alice encodes the message as a number m in 1…n
2. Alice computes $c = m^e \bmod n$
3. Alice sends the ciphertext c to Bob

[Rivest, Shamir, Adleman, 1978]

RSA public key encryption

Decryption

To decrypt a ciphertext c:

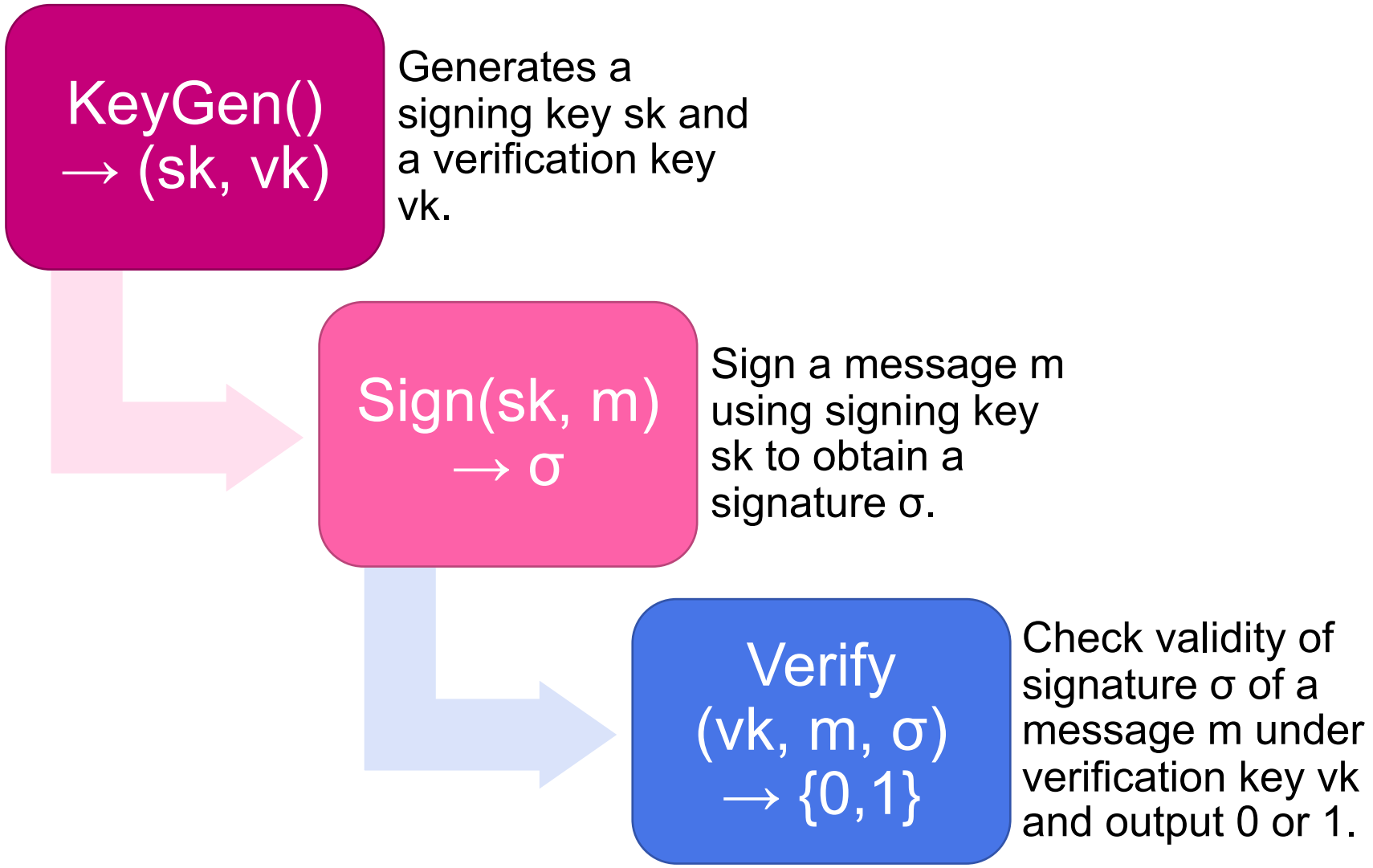
1. Bob computes $m = c^d \bmod n$
2. Bob decodes the number m into the message

Decryption works because

- $c^d = m^{ed} \equiv m^1 = m \bmod n$
- Since $ed = 1 \bmod \phi(n)$

[Rivest, Shamir, Adleman, 1978]

Digital Signatures



Digital Signatures: Security

Security goal: existential unforgeability under chosen message attack (EUCMA).

Chosen message attack

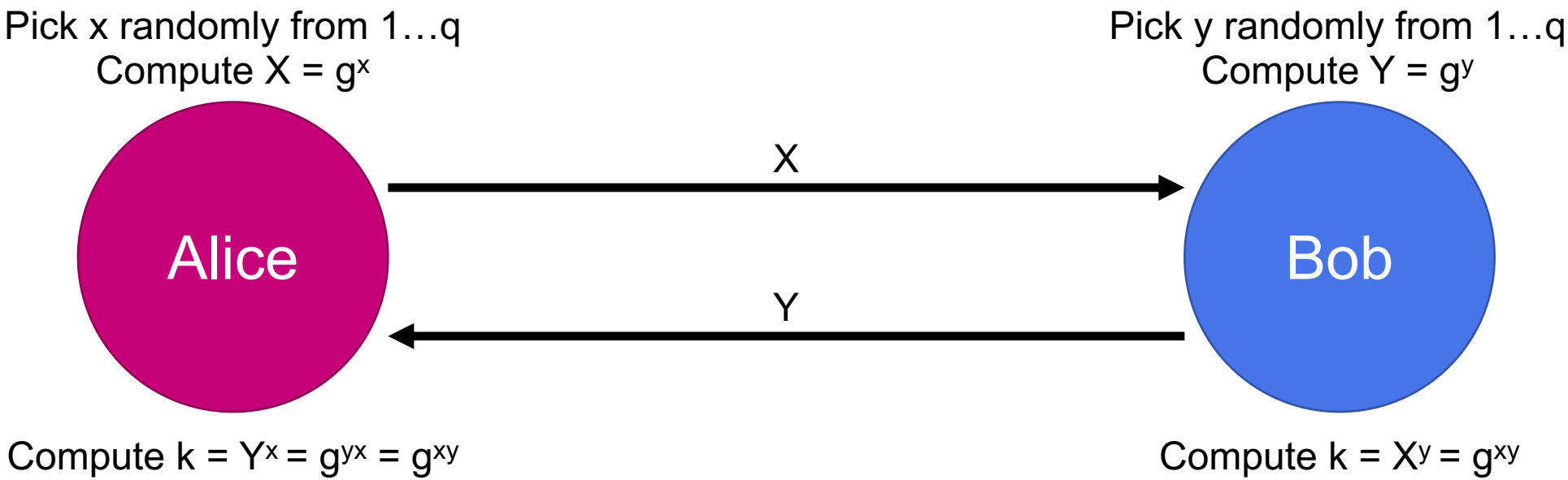
- adversary can adaptively obtain signatures for any messages of his choosing

Existential unforgeability

- hard to construct a new valid signature/message pair (note: message doesn't have to be "meaningful")

Diffie–Hellman key exchange

Fixed public parameters: generator g of a group of prime order q



[Diffie, Hellman, 1976]

Signed Diffie–Hellman key exchange

Generate signature key pair (vk_A, sk_A)

Obtain authentic copy of vk_A

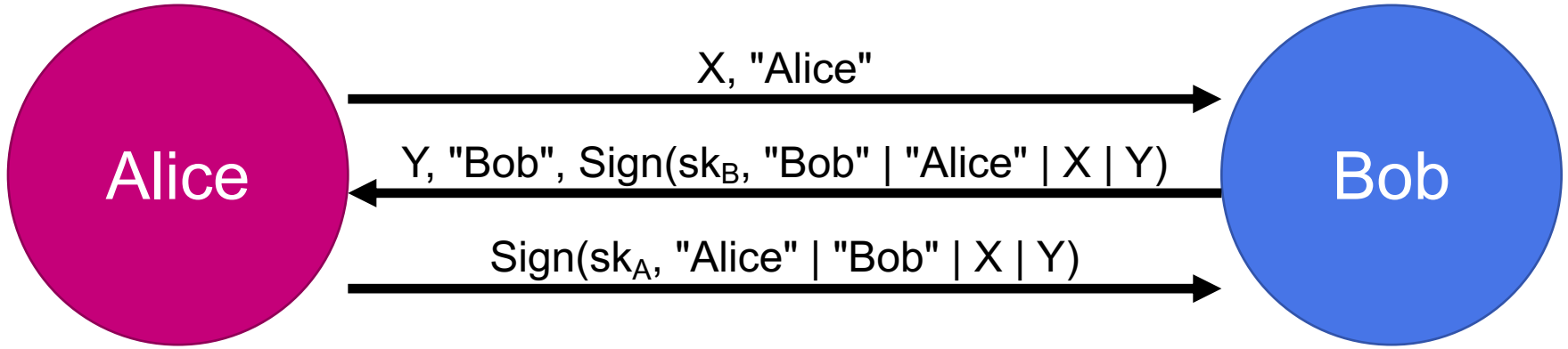
Obtain authentic copy of vk_B

Generate signature key pair (vk_B, sk_B)

Fixed public parameters: generator g of a group of prime order q

Pick x randomly from 1…q
Compute $X = g^x$

Pick y randomly from 1…q
Compute $Y = g^y$



Lookup vk_B
Verify signature using vk_B
Compute $k = Y^x$

Lookup vk_A
Verify signature using vk_A
Compute $k = X^y$

| means concatenation

Key Exchange: Security

Security goal: indistinguishability of session keys under various attack scenarios.

Attack scenarios

- adversary can control communications,
- learn session keys of other sessions,
- learn parties' long-term keys ("forward secrecy")
- learn parties' random coins

Indistinguishability of session key

- hard to distinguish the real session key from random string of the same length

1. Background

Why post-quantum?

2021 IEEE North American School of Information Theory

Welcome

Please check out this brief Welcome video with a few words about the school and logistics from the organizers.

Updates

Tutorials including Q&A will be synchronous events, without recording.

We would like to encourage interactions between tutorial speakers and (student) attendees, in the spirit of the School with and for students, and similar to the in-person NASIT events. **Therefore we will also not record sessions.**

Questions can be asked during the tutorials, and tutorial moderators will help with asking and addressing questions that have been posted in the chat.

About the School

The 2021 IEEE North American School of Information Theory will be held **Monday, June 21 through Friday, June 25 2021**.

This will be the 13th Annual North American School of Information Theory and follows a series of events designed to provide graduate students with opportunities to:

- Learn from senior lecturers in the field who will present long-format tutorials;
- Participate in a stimulating and inviting forum of scientists;
- Present their own work for feedback and potential collaboration;
- Deepen their connections with the community.

Program

The school will be a **virtual event** held over five days and will consists of

- Senior lecturers presenting long-format (2 1/2 hour) tutorials;
- Students presenting their own work in poster sessions.

Participation

The tutorials will be delivered in zoom meetings and the poster sessions will take place in virtual poster rooms using the web-conferencing space Gather.town.

Links to these events will be sent out to registered participants on June 20, 2021.

Lecturers

2021 IEEE North American Sch

conferences.ece.ubc.ca/nasit2021/

NASIT 2021

- General Info
- Schedule
- Tutorials
- Poster
- Sessions and Titles
- Abstracts
- Registration

2021 IEEE North American School of Information Theo

Welcome

Please check out this brief Welcome video with a few words about the school and logistics from t

Updates

Tutorials including Q&A will be synchronous events, without recording.

We would like to encourage interactions between tutorial speakers and (student) attendees, in t and similar to the in-person NASIT events. **Therefore we will also not record sessions.**

Questions can be asked during the tutorials, and tutorial moderators will help with asking and a the chat.

About the School

The 2021 IEEE North American School of Information Theory will be held **Monday, June 21 t**

This will be the 13th Annual North American School of Information Theory and follows a series students with opportunities to:

- Learn from senior lecturers in the field who will present long-format tutorials;
- Participate in a stimulating and inviting forum of scientists;
- Present their own work for feedback and potential collaboration;
- Deepen their connections with the community.

Program

The school will be a **virtual event** held over five days and will consists of

- Senior lecturers presenting long-format (2 1/2 hour) tutorials;
- Students presenting their own work in poster sessions.

Participation

The tutorials will be delivered in zoom meetings and the poster sessions will take place in virtua space Gather.town.

Links to these events will be sent out to registered participants on June 20, 2021

Elements Console Sources Security × 4 2

Overview

Main origin

Reload to view details

Security overview

This page is secure (valid HTTPS).

Certificate - valid and trusted

The connection to this site is using a valid, trusted server certificate issued by R3.

View certificate

Connection - secure connection settings

The connection to this site is encrypted and authenticated using TLS 1.2, ECDHE_RSA with P-256, and AES_256_GCM.

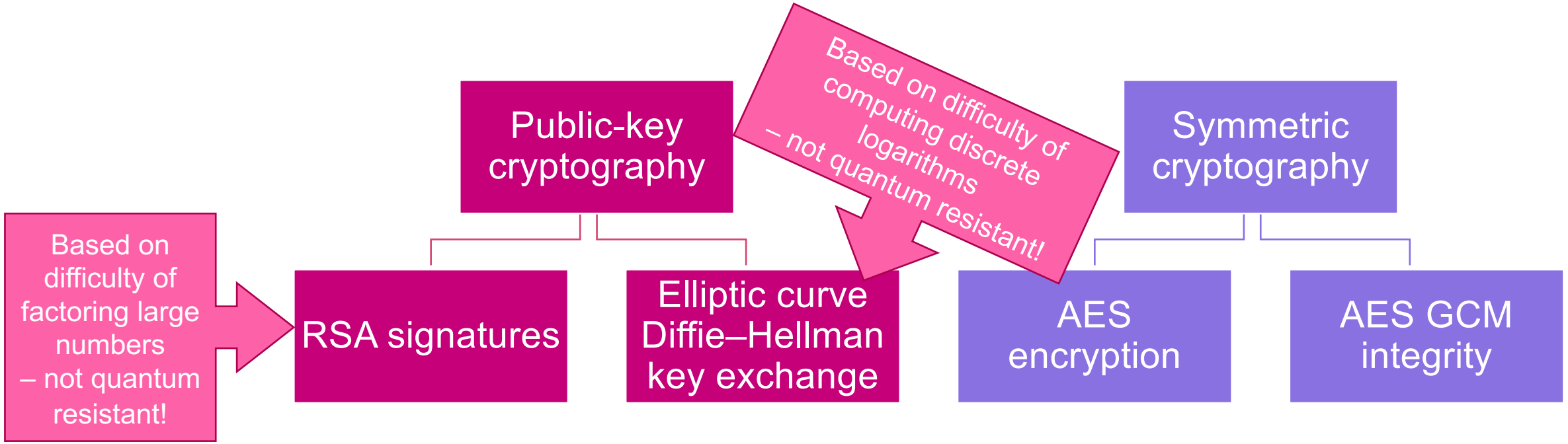
Resources - all served securely

All resources on this page are served securely.

Cryptographic building blocks

Connection - secure connection settings

The connection to this site is encrypted and authenticated using TLS 1.2, ECDHE_RSA with P-256, and AES_128_GCM.



What can go wrong

- Mathematical advances break cryptographic assumptions
- Good cryptography is used improperly in applications and protocols
- Bugs in how good cryptography is implemented in software & hardware

Quantum computing

Represent and process information using **quantum mechanics**

"Classical" computers handle information as **bits:**

- 0 and 1

Quantum computers handle information as **qubits:**

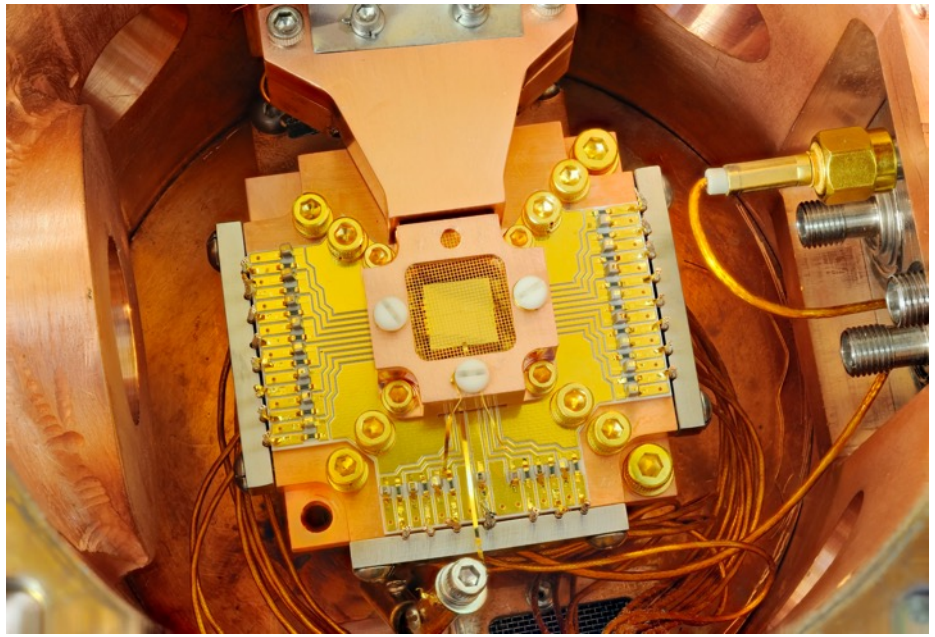
- Any "superposition" of 0 and 1

Processing information in superposition can dramatically speed some computations

- Chemical reaction simulations
- Optimization problems
- Arithmetic

But not magic

- Doesn't dramatically speed up all computations



Home

www.dwavesys.com

D-Wave
The Quantum Computing Company™

Welcome to the Future

Quantum Computing for the Real World Today

Scalable quantum computers

uwaterloo.ca/institute-for-quantum-computing/news/scalable-quantum-computers-within-reach

UNIVERSITY OF WATERLOO

ADMISSIONS ABOUT WATERLOO FACULTIES & ACADEMICS OFFICES & SERVICES SUPPORT WATERLOO SEARCH

INSTITUTE FOR QUANTUM COMPUTING

- Institute for Quantum Computing home
- About IQC
- Our people
- Available positions
- Research
- Programs
- Outreach
- News
- Events
- Blog

INFORMATION FOR

- Researchers
- Students
- Visitors
- Media
- Alumni and friends

Institute for Quantum Computing » News » 2017 » September »

Scalable quantum computers within reach

MONDAY, SEPTEMBER 18, 2017

Quantum machine learning and artificial intelligence, quantum-safe cryptography, and simulation of quantum systems all rely on the power of quantum computing.

A team of researchers at the Institute for Quantum Computing (IQC) have taken a step closer to realizing the powerful possibilities of a universal quantum computer. The Laboratory for Digital Quantum Matter, led by faculty member Matteo Mariantoni, is developing technologies for extensible quantum computing architectures based on superconducting quantum devices.

Superconducting quantum circuits have close to zero electrical resistance and offer enhanced efficiency and processing power compared to traditional electrical circuits. Mariantoni's research group uses nanofabrication tools and semiconductor technology to fabricate on-chip superconducting quantum circuits which operate at microwave frequencies.

The source of the quantum information in the superconducting quantum circuit is the qubit. The qubit is similar to an electronic circuit found in a classical computer that is characterized by two states, 0 or 1. However, the qubit can also be prepared in superposition states – both 0 and 1 at the same time – made possible by quantum mechanics.

Quantum mechanical states are fragile and interact easily with their environment. As a result, qubits cannot store information for very long times; the interaction with the environment in the circuit eventually causes the bit to decay, transitioning from one state to another in a random, unwanted fashion. These errors must be mitigated to implement a universal quantum computer.

Google's Quantum Dream Ma

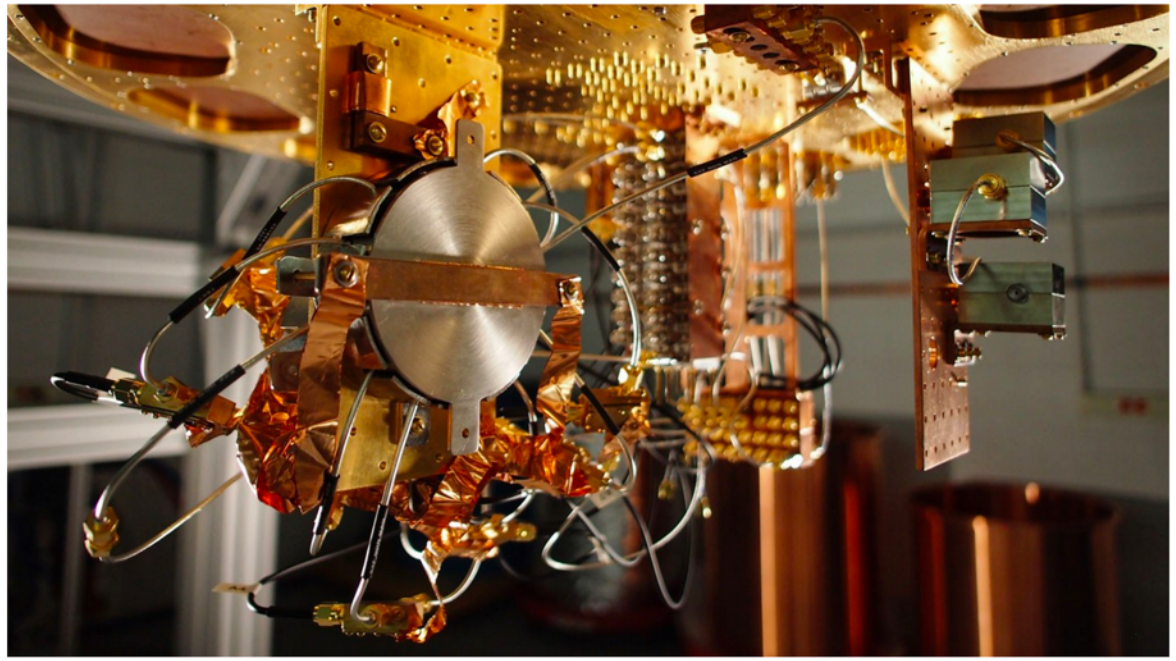
www.technologyreview.com/s/544421/googles-quantum-dream-machine/

MIT Technology Review

Log in / Register Search

Topics+ The Download Magazine Events More+

Subscribe



Intelligent Machines

Google's Quantum Dream Machine

Physicist John Martinis could deliver one of the holy grails of computing to Google—a machine that dramatically speeds up today's applications and makes new ones possible.

Quantum computing | Micros

www.microsoft.com/en-us/quantum/default.aspx

Microsoft Cloud Mobility Productivity Sign in

Quantum Team Technology Resources

Empowering the quantum revolution

Your path to powerful, scalable quantum computing starts here.

Learn more

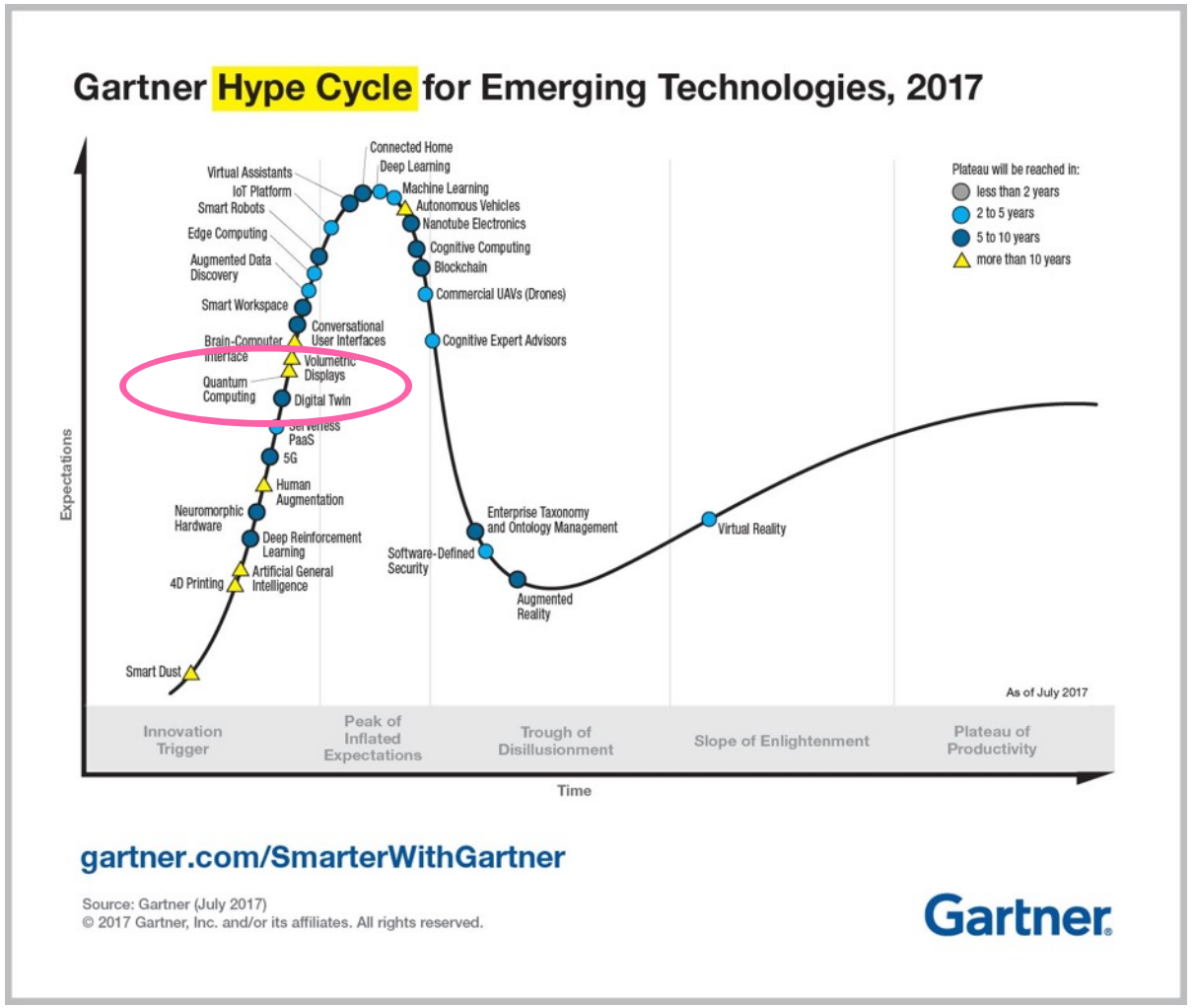
Join us at the leading edge of opportunity

Quantum computing takes a giant leap forward from today's technology—one that will forever alter our economic, industrial, academic, and societal landscape. In just hours or days, a quantum computer can solve complex problems that would otherwise take billions of years for classical computing to solve. This has massive implications for research in healthcare, energy, environmental systems, smart materials, and more. The quantum economy is coming. And Microsoft envisions a future where customers use Azure for both classical and quantum computing.

Stay updated


The Economist
Ryancare attacked from left and right
IS up against the wall in Mosul
Taiwan and the one-China fiction
Is there a bubble in the markets?
Quantum leaps
A mind-bending technology goes mainstream

March 2017


Gartner Hype Cycle for Emerging Technologies, 2017
Expectations
Time
Innovation Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity
Plateau will be reached in:
less than 2 years
2 to 5 years
5 to 10 years
more than 10 years
Connected Home
Deep Learning
Virtual Assistants
IoT Platform
Machine Learning
Smart Robots
Autonomous Vehicles
Edge Computing
Nanotube Electronics
Augmented Data Discovery
Cognitive Computing
Blockchain
Smart Workspace
Commercial UAVs (Drones)
Conversational User Interfaces
Brain-Computer Interface
Cognitive Expert Advisors
Volumetric Displays
Quantum Computing
Digital Twin
PaaS
5G
Human Augmentation
Neuromorphic Hardware
Deep Reinforcement Learning
Artificial General Intelligence
4D Printing
Smart Dust
Enterprise Taxonomy and Ontology Management
Software-Defined Security
Augmented Reality
Virtual Reality
As of July 2017
gartner.com/SmarterWithGartner
Source: Gartner (July 2017)
© 2017 Gartner, Inc. and/or its affiliates. All rights reserved.
Gartner

Quantum threat to information security

Large-scale general-purpose quantum computers could break some encryption schemes

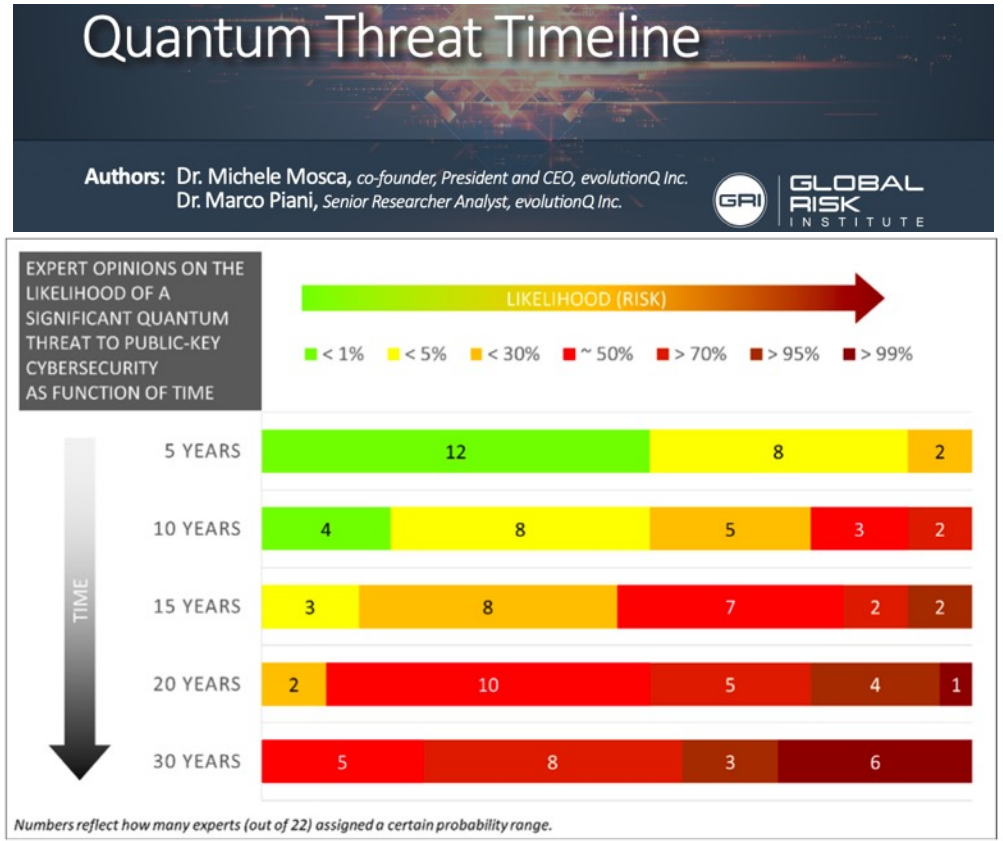
Need to migrate encryption to quantum-resistant algorithms

When should you start the process?

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, University of Waterloo



https://eprint.iacr.org/2015/1075
http://qurope.eu/system/files/u7/93056_Quantum%20Manifesto_WEB.pdf
https://globalriskinstitute.org/publications/quantum-threat-timeline/

Post-quantum cryptography

a.k.a. quantum-resistant algorithms

Cryptography believed to be resistant to attacks by quantum computers

Uses only classical (non-quantum) operations to implement

Not as well-studied as current encryption

- Less confident in its security
- More implementation tradeoffs

Hash-based & symmetric

Multivariate quadratic

Code-based

Lattice-based

Elliptic curve isogenies

Confidence in quantum-resistance

Fast computation

Small communication

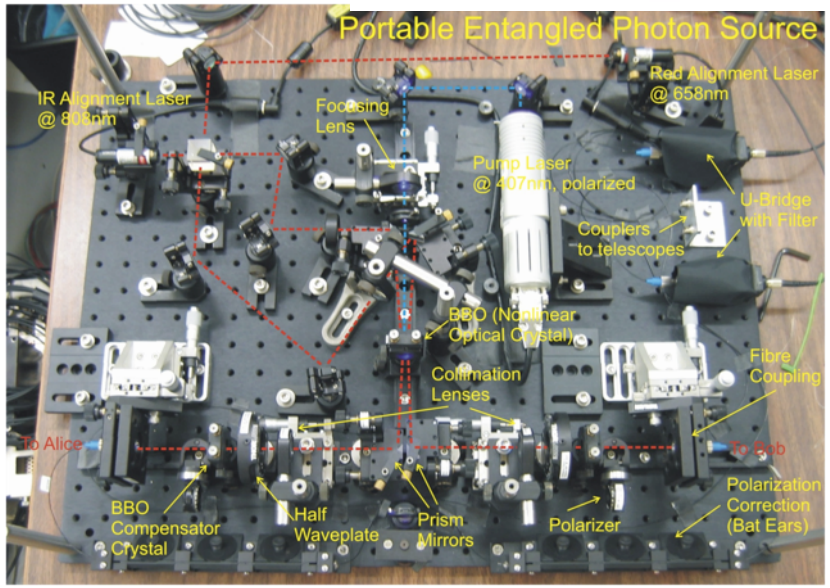
Quantum key distribution

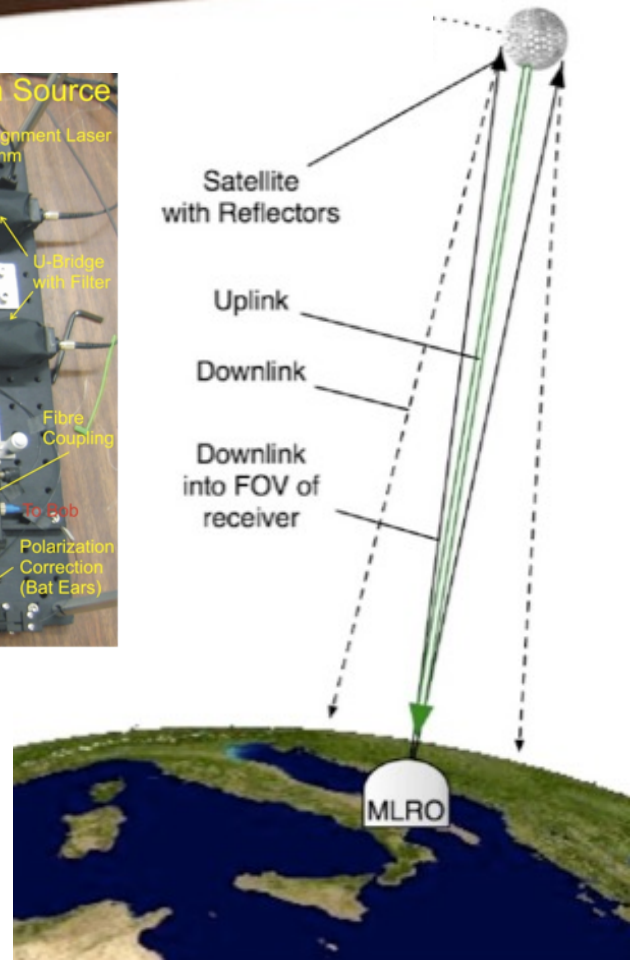
Uses quantum mechanics to protect information

Doesn't require a full quantum computer

But does require new communications infrastructure and hardware

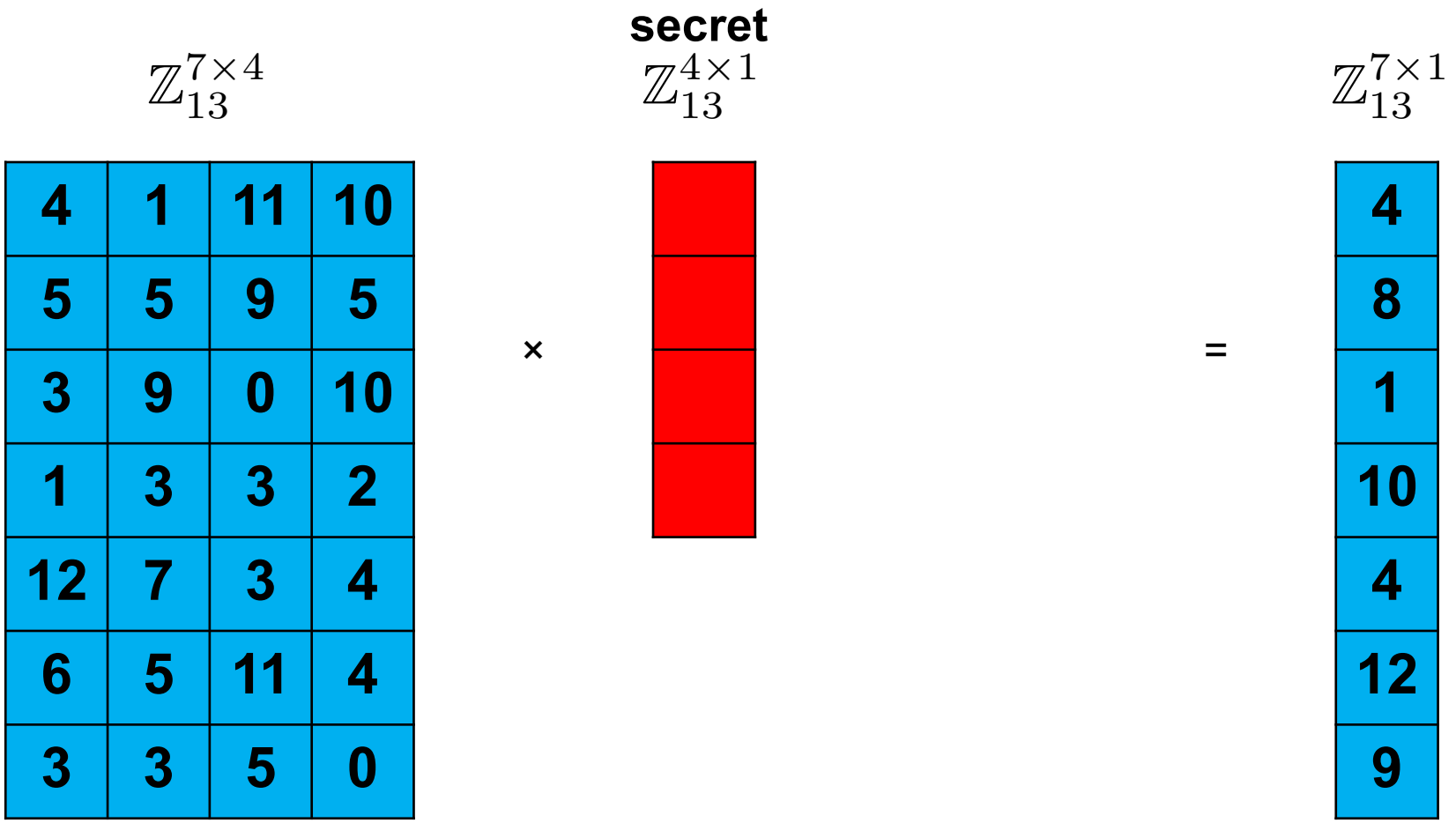
=> Not the subject of this talk





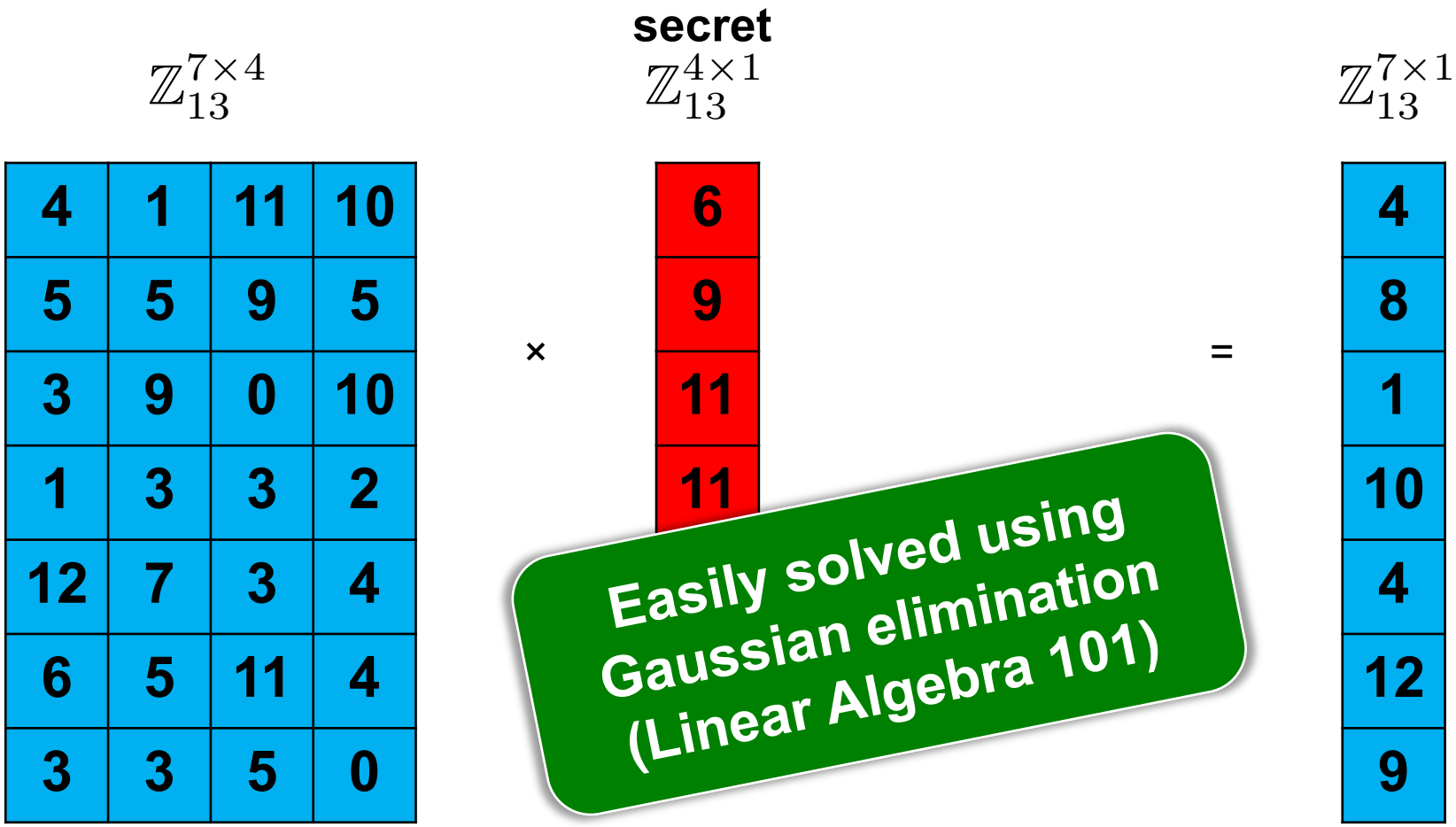
2. Learning with errors problems

Solving systems of linear equations



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

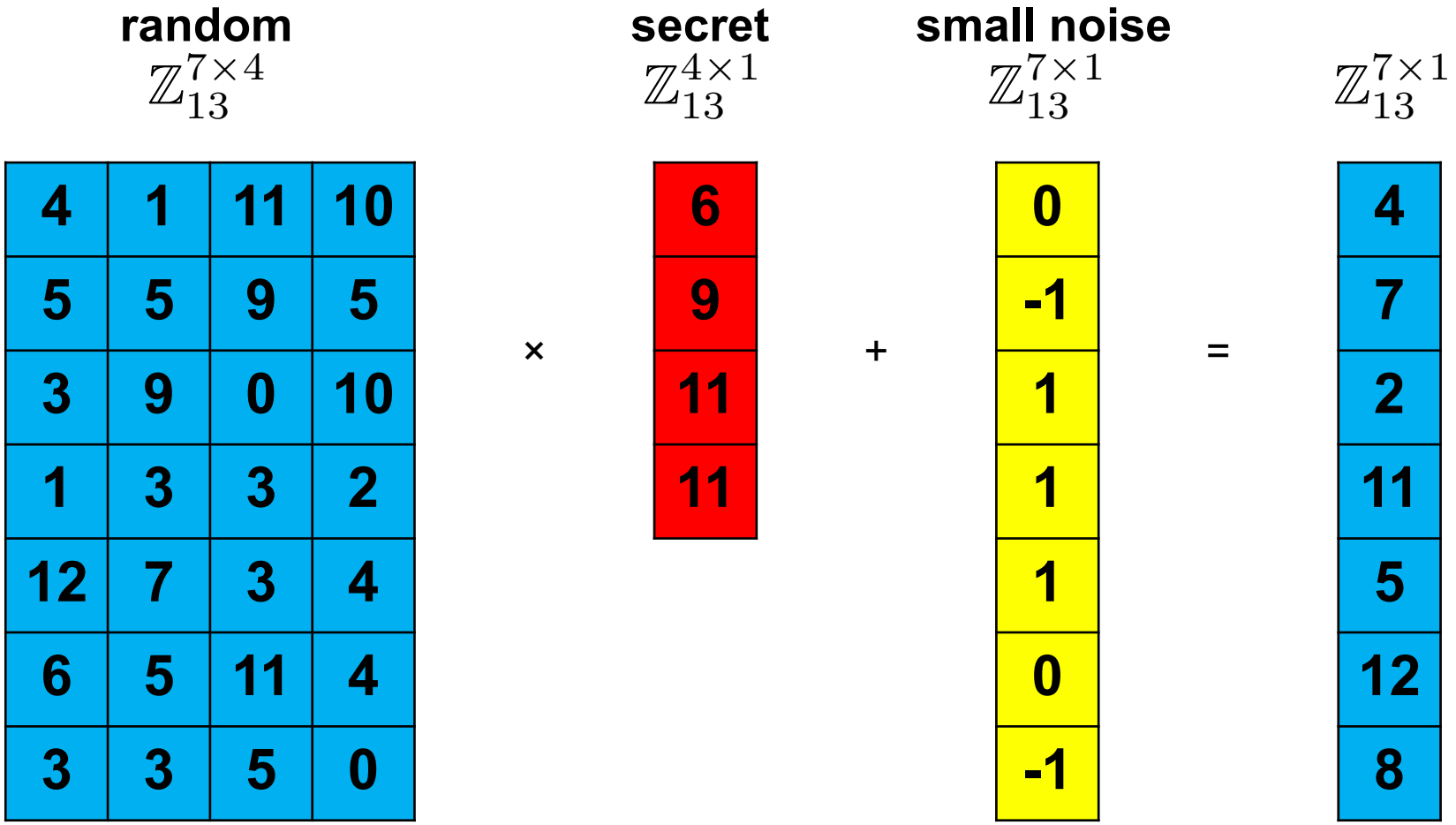
Solving systems of linear equations



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

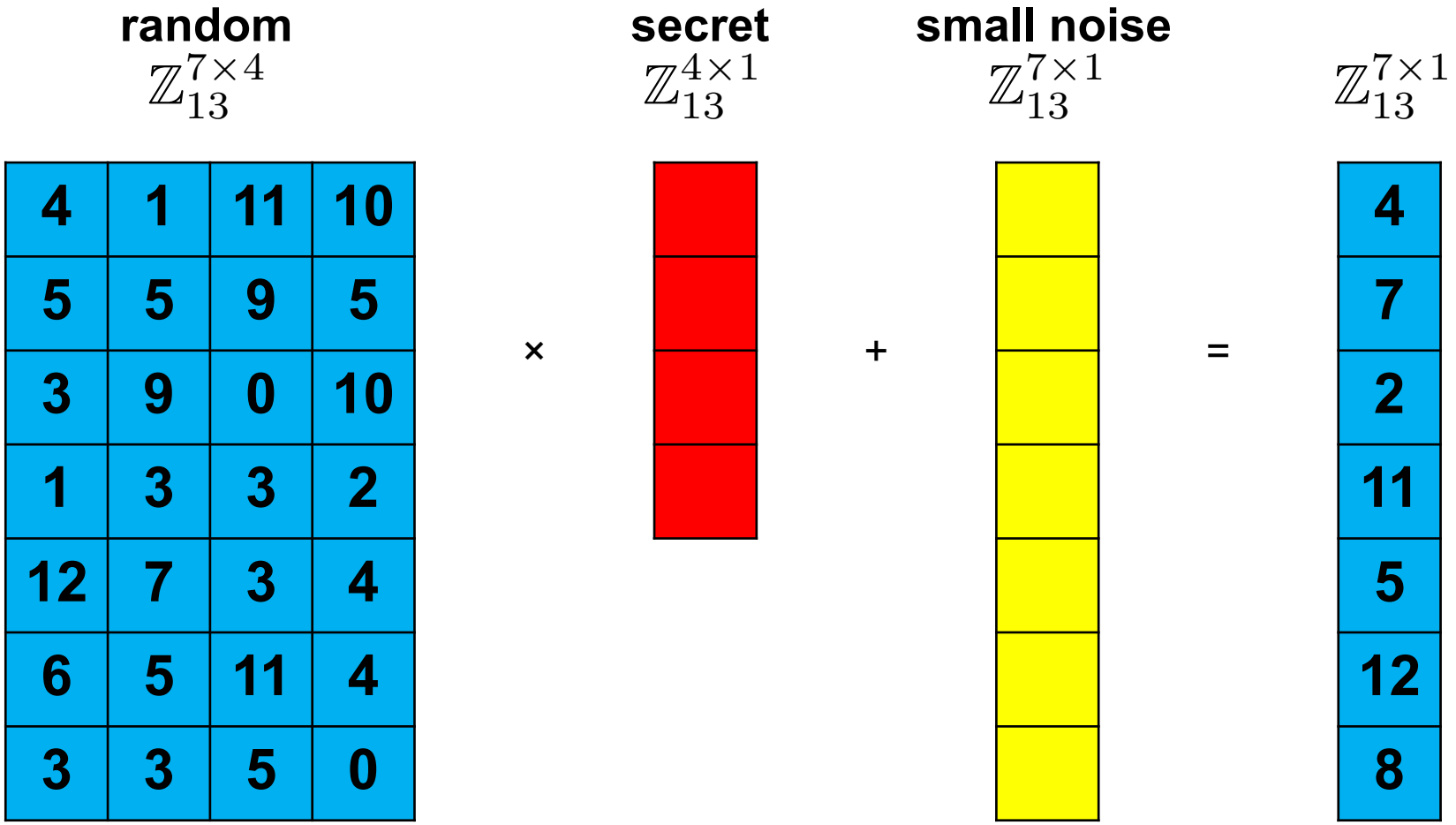
Learning with errors problem

[Regev 2005]



Learning with errors problem

[Regev 2005]



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

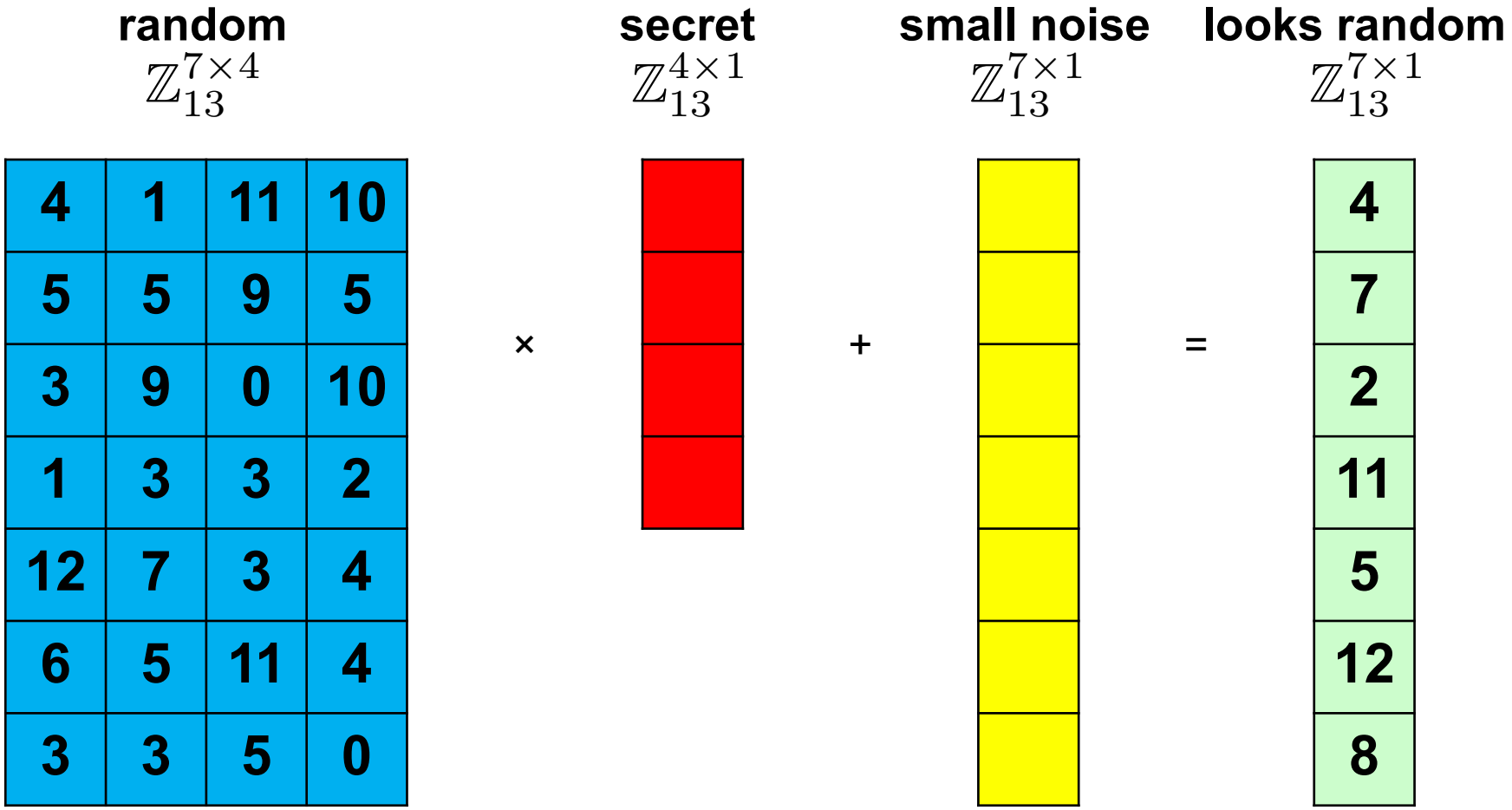
The *search LWE problem* for $(n, m, q, \chi_s, \chi_e)$ is to find $\mathbf{s}$ given $(\mathbf{a}_i, b_i)_{i=1}^m$.

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

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

[Regev STOC 2005]

Decision learning with errors problem



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

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

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 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 $\alpha < 1$ for error rate $s = \alpha q$

- 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.

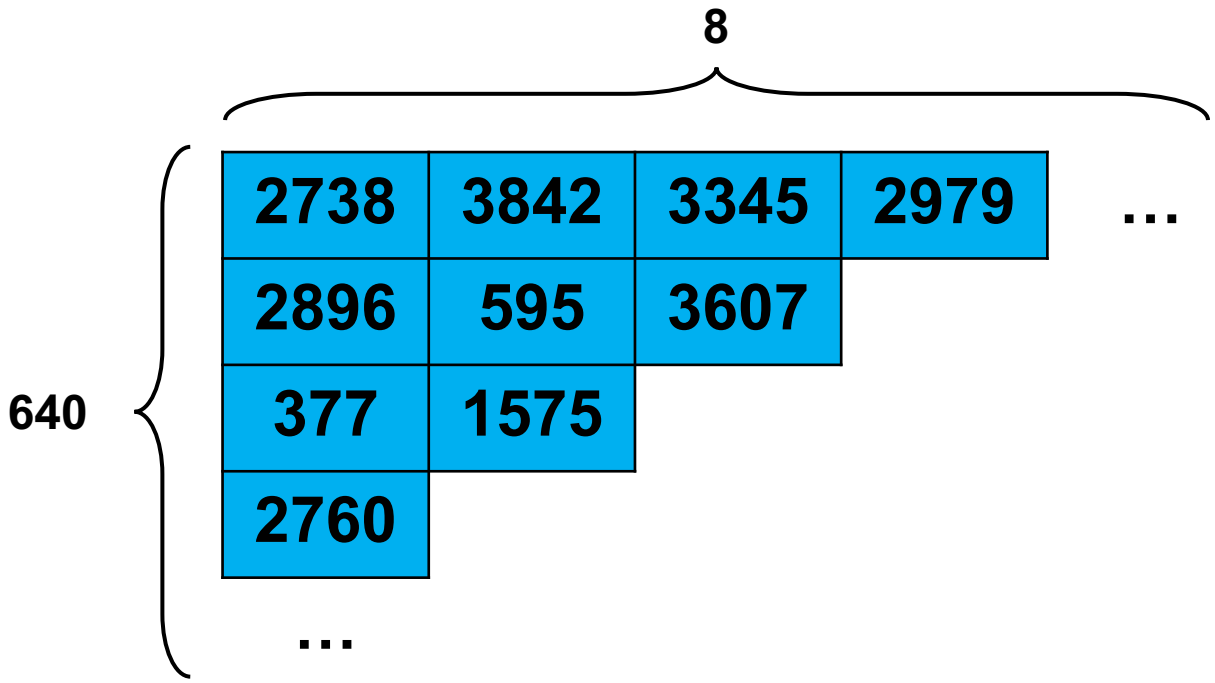
[Applebaum et al., CRYPTO 2009]

Toy example versus real-world example

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

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

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



640 × 8 × 15 bits = **9.4 KiB**

Ring learning with errors problem

[Lyubashevsky, Peikert, Regev 2010]

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

[Lyubashevsky, Peikert, Regev 2010]

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

[Lyubashevsky, Peikert, Regev 2010]

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

[Lyubashevsky, Peikert, Regev 2010]

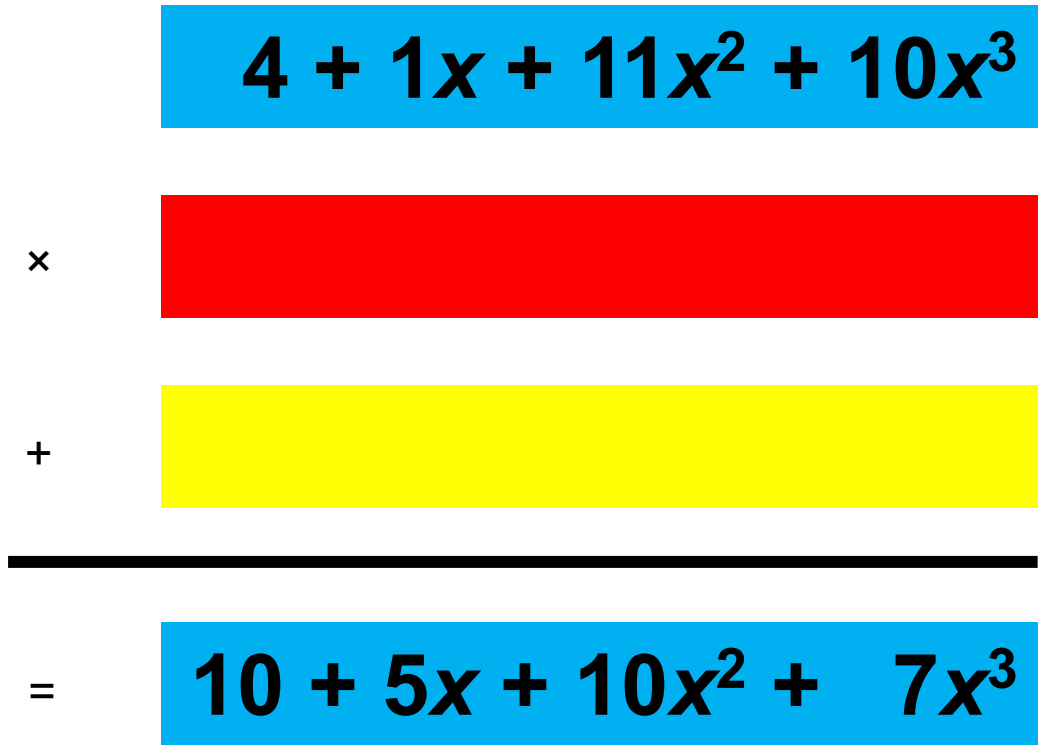
$\mathbb{Z}_{13}[x]/\langle x^4 + 1\rangle$

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

[Lyubashevsky, Peikert, Regev 2010]

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



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 $s \xleftarrow{\$} \chi_s$. Let $a \xleftarrow{\$} \mathcal{U}(R_q)$, $e \xleftarrow{\$} \chi_e$, 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

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

[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:

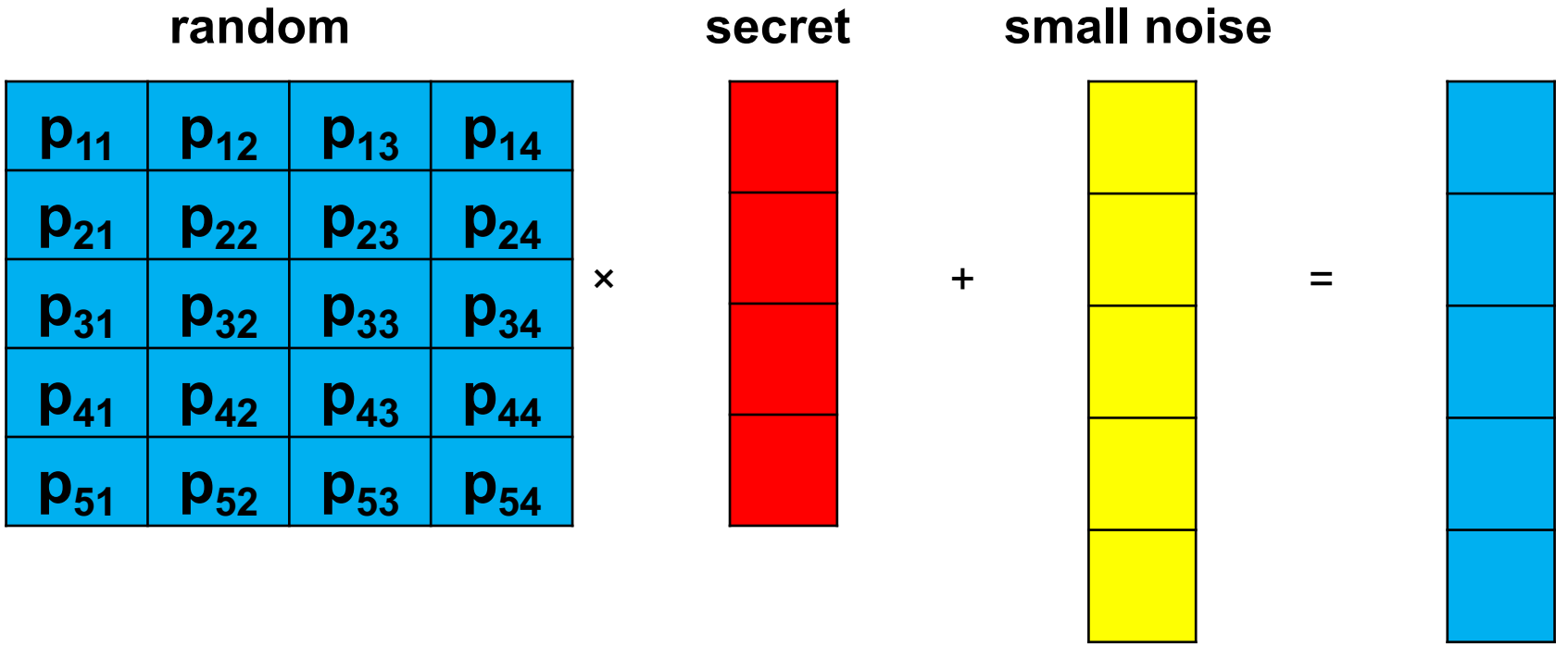
- $O_{\chi_e,s}$: $a \xleftarrow{\$} \mathcal{U}(R_q)$, $e \xleftarrow{\$} \chi_e$; return $(a, as + e)$.
- U: $a, u \xleftarrow{\$} \mathcal{U}(R_q)$; return (a, u).

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

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

Module learning with errors problem



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

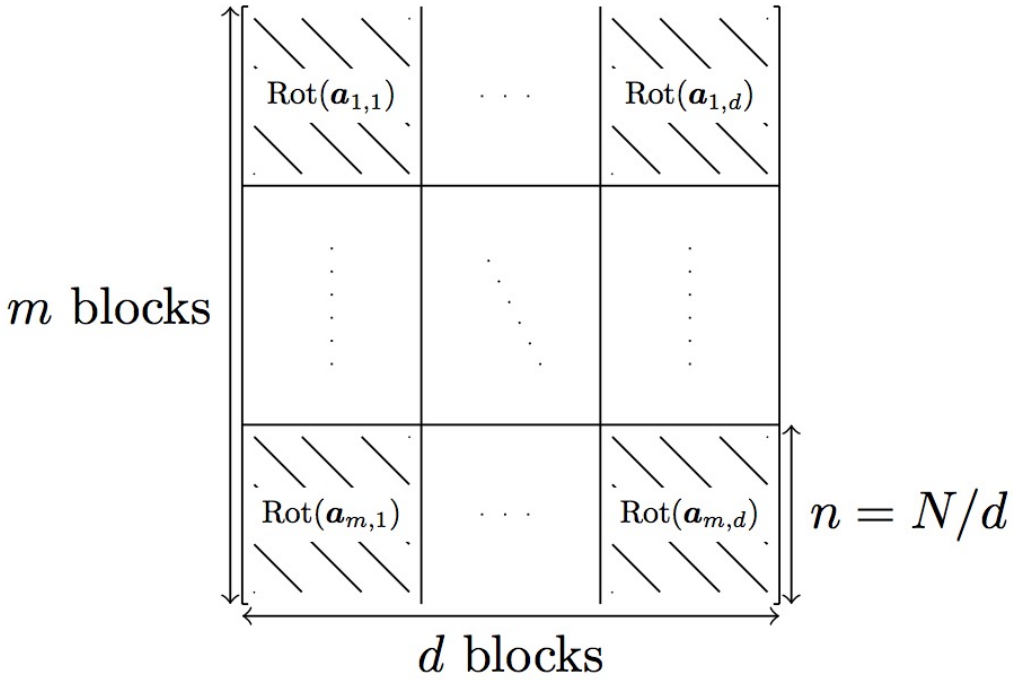
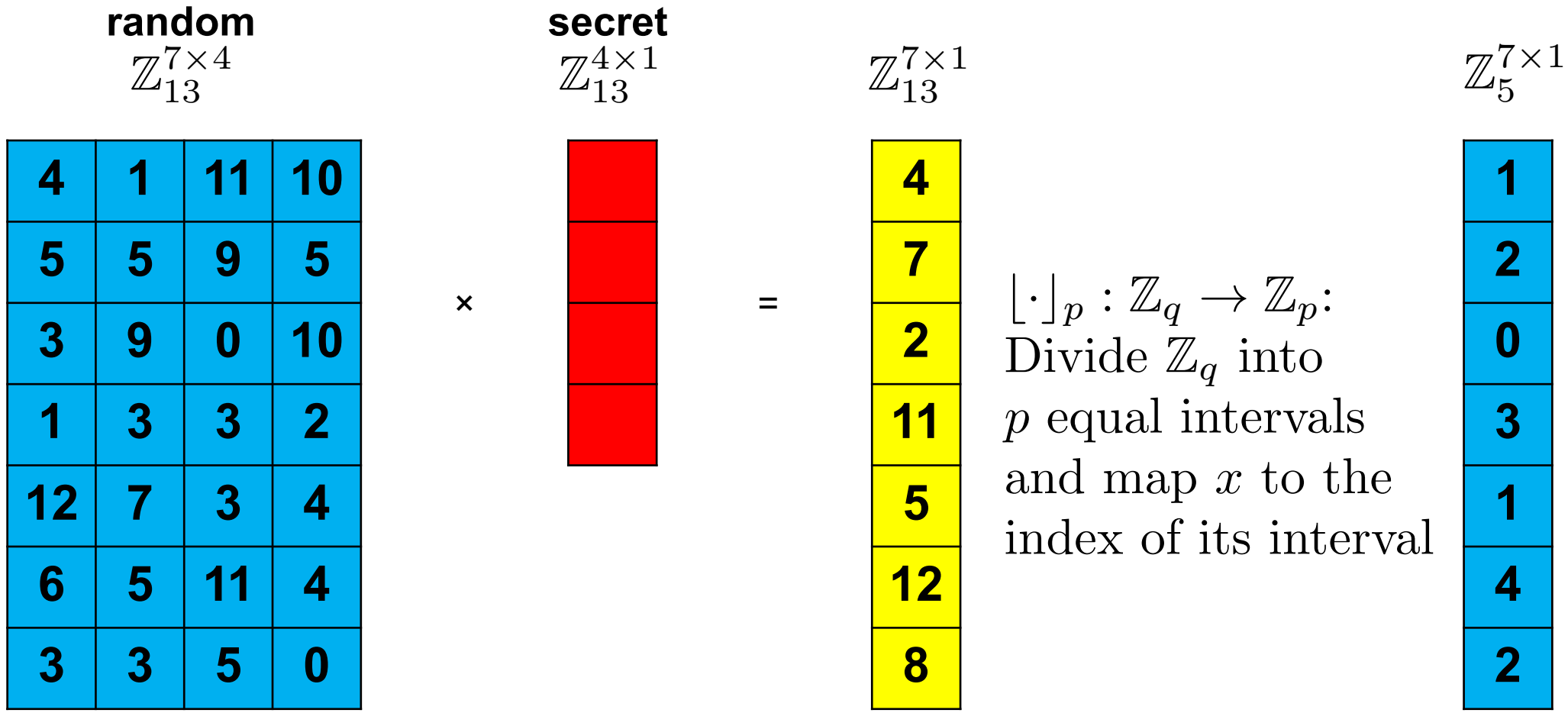


Figure from https://eprint.iacr.org/2012/090.pdf

Learning with rounding problem



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 the **NTRU distribution** $N_{s,\chi}$ to be the distribution that outputs $e/s \in R_q$ where $e \leftarrow \chi$.

Definition [NTRU decision problem]. 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 on R_q,

distinguish which is the case.

This is a "**noisy quotient**" problem.

[Hoffstein, Pipher, Silverman ANTS 1998]

NTRU versus LWE

NTRU:
noisy quotient
e/s

LWE:
noisy product
$as + e$

Problems

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

3. Cryptography from learning with errors

Public key encryption

ElGamal public key encryption

- A public key encryption scheme built from Diffie–Hellman key exchange

Key generation

- Parameters g and q same as for Diffie–Hellman

1. Bob picks a random integer y as his fixed private key and publishes his public key: Y = gy.

ElGamal public key encryption

Encryption

To encrypt a message to send to Bob:

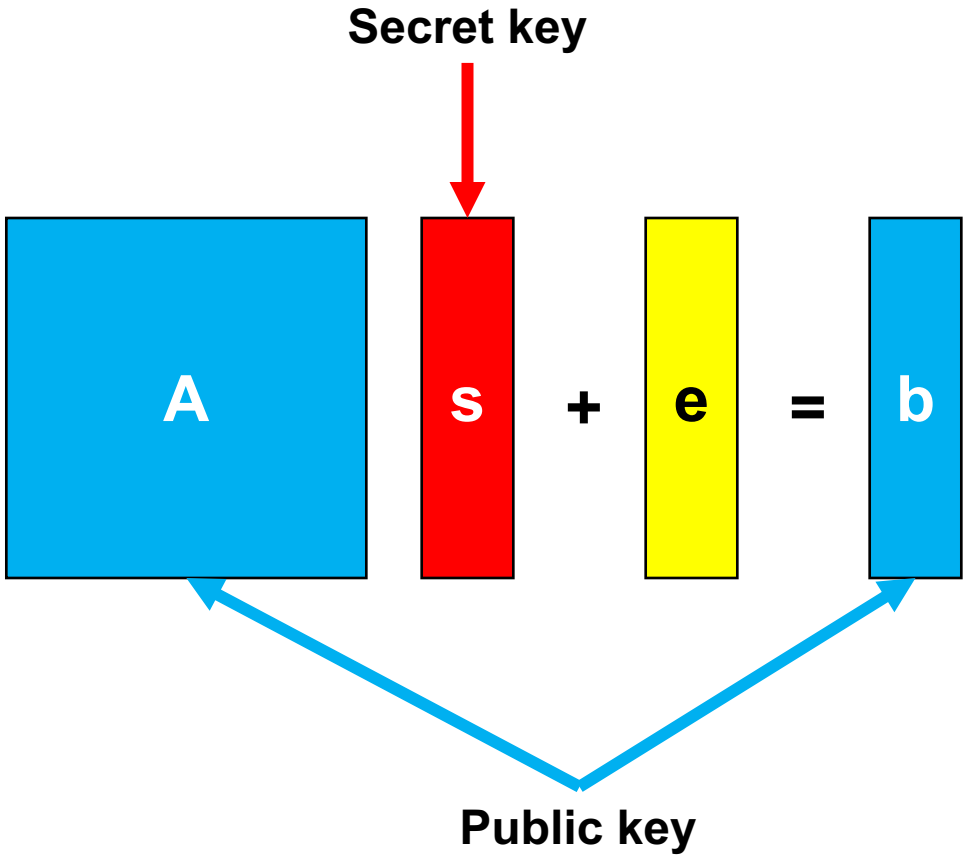
1. Alice encodes the message as a group element m
2. Alice picks a random integer r from 1…q and computes
   a) $C_1 = g^r$
   b) $(g^y)^r = g^{yr}$
   c) $C_2 = m \times g^{yr}$
3. Alice sends the ciphertext (C_1 and C_2) to Bob

Decryption

To recover the message from a ciphertext:

1. Bob computes the shared secret $(C_1)^y = (g^r)^y = g^{ry}$
2. Recovers $m = C_2 / g^{ry}$

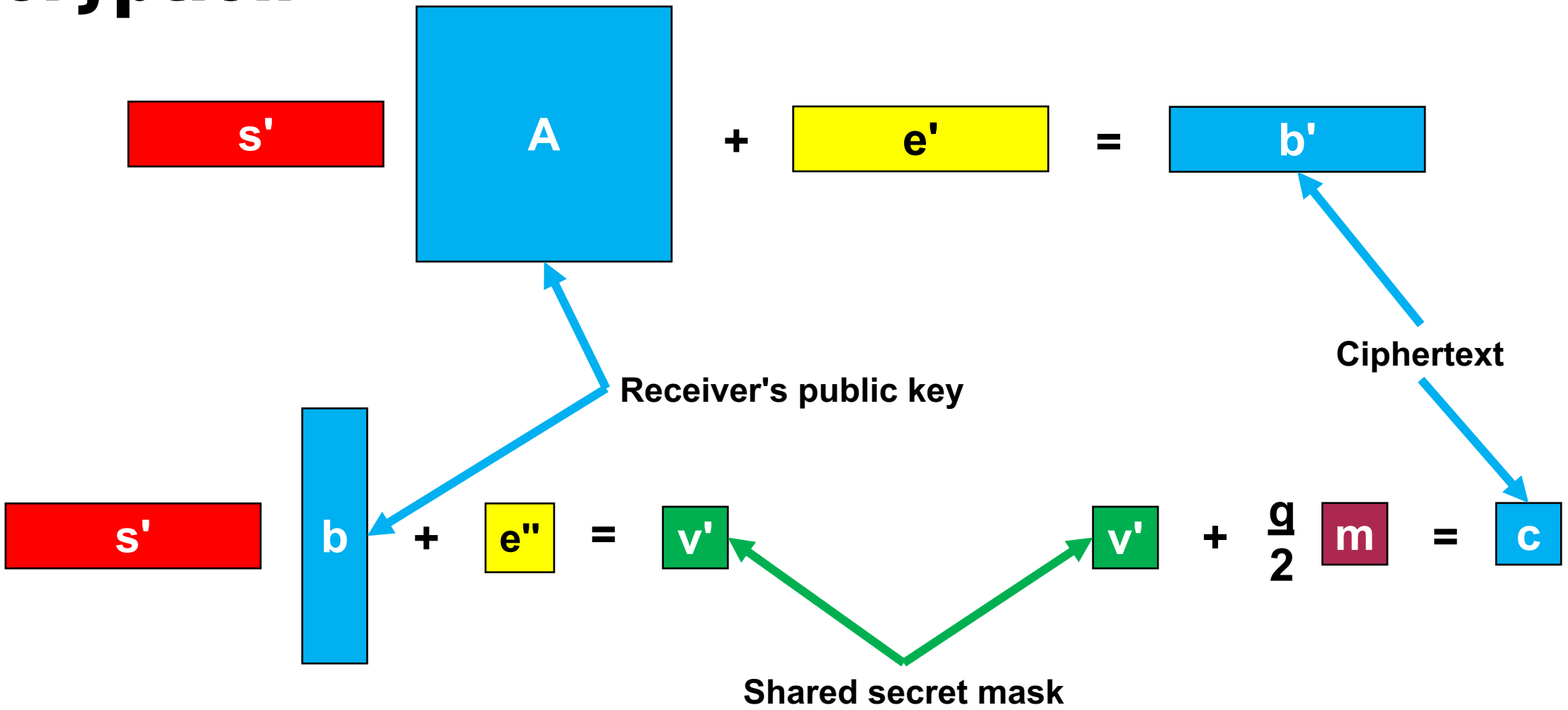
Public key encryption from LWE
Key generation



[Lindner, Peikert. CT-RSA 2011]

Public key encryption from LWE

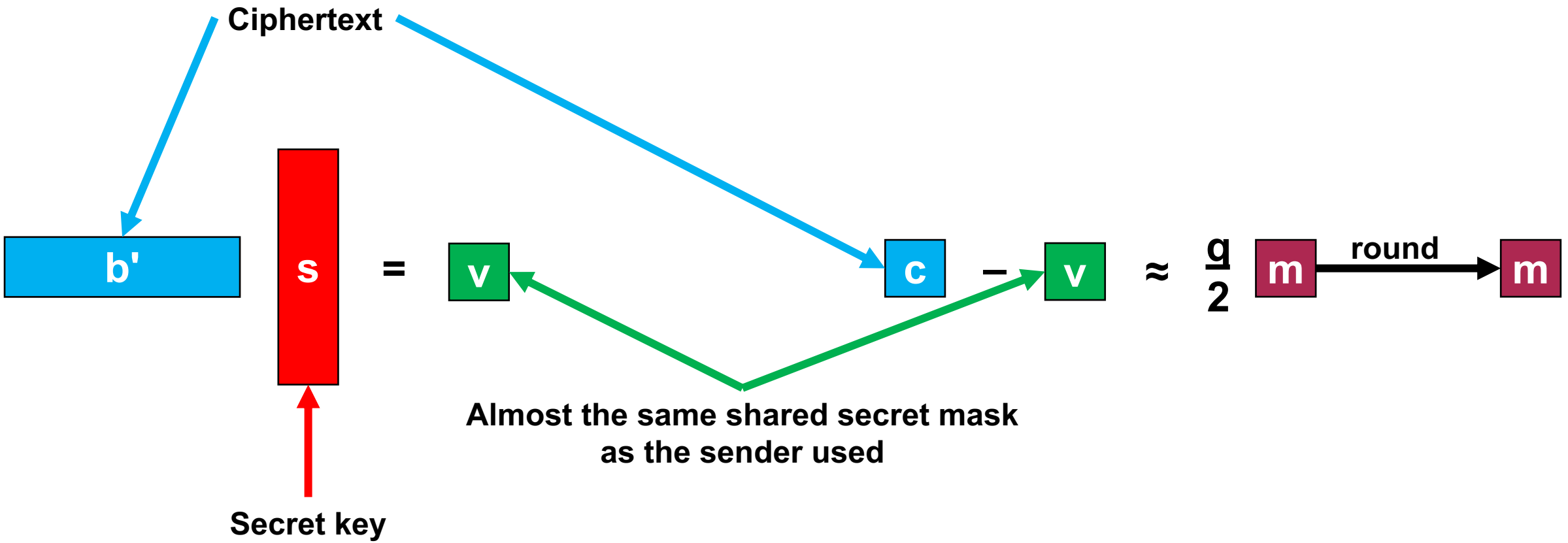
Encryption



[Lindner, Peikert. CT-RSA 2011]

Public key encryption from LWE
Decryption

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



[Lindner, Peikert. CT-RSA 2011]

Approximately equal shared secret

The sender uses

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

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

$\approx s' A s$

The receiver uses

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

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

$\approx s' A s$

Lindner–Peikert public key encryption

Let n, q, χ be LWE parameters.

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

Regev's public key encryption scheme

Let n, m, q, χ be LWE parameters.

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

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

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

[Regev; STOC 2005]

Difference between Regev and Lindner–Peikert

Regev:

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

Lindner–Peikert:

- Bob's public key is $\mathbf{s}'\mathbf{A} + \mathbf{e}'$ where $\mathbf{s}' \xleftarrow{\$} \chi_e$
- 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:

1: $\mathbf{A} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n\times n})$
2: $\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi(\mathbb{Z}_q^n)$
3: $\tilde{\mathbf{b}} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{e}$
4: $\mathbf{s}', \mathbf{e}' \xleftarrow{\$} \chi(\mathbb{Z}_q^n)$
5: $\tilde{\mathbf{b}}' \leftarrow \mathbf{s}'\mathbf{A} + \mathbf{e}'$
6: $e'' \xleftarrow{\$} \chi(\mathbb{Z}_q)$
7: $\tilde{v}' \leftarrow \mathbf{s}'\tilde{\mathbf{b}} + e''$
8: $c_0 \leftarrow \mathrm{encode}(0) + \tilde{v}'$
9: $c_1 \leftarrow \mathrm{encode}(1) + \tilde{v}'$
10: $b^* \xleftarrow{\$} \mathcal{U}(\{0,1\})$
11: **return** $(\mathbf{A}, \tilde{\mathbf{b}}, \tilde{\mathbf{b}}', c_{b^*})$

1: $\mathbf{A} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n\times n})$
2: $\tilde{\mathbf{b}} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^n)$
3: $\mathbf{s}', \mathbf{e}' \xleftarrow{\$} \chi(\mathbb{Z}_q^n)$
4: $\tilde{\mathbf{b}}' \leftarrow \mathbf{s}'\mathbf{A} + \mathbf{e}'$
5: $e'' \xleftarrow{\$} \chi(\mathbb{Z}_q)$
6: $\tilde{v}' \leftarrow \mathbf{s}'\tilde{\mathbf{b}} + e''$
7: $c_0 \leftarrow \mathrm{encode}(0) + \tilde{v}'$
8: $c_1 \leftarrow \mathrm{encode}(1) + \tilde{v}'$
9: $b^* \xleftarrow{\$} \mathcal{U}(\{0,1\})$
10: **return** $(\mathbf{A}, \tilde{\mathbf{b}}, \tilde{\mathbf{b}}', c_{b^*})$

1: $\mathbf{A} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n\times n})$
2: $\tilde{\mathbf{b}} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^n)$
3: $\mathbf{s}' \xleftarrow{\$} \chi(\mathbb{Z}_q^n)$
4: $[\mathbf{e}' \| e''] \xleftarrow{\$} \chi(\mathbb{Z}_q^{n+1})$
5: $[\tilde{\mathbf{b}}' \| \tilde{v}'] \leftarrow \mathbf{s}'[\mathbf{A} \| \tilde{\mathbf{b}}] + [\mathbf{e}' \| e'']$
6: $c_0 \leftarrow \mathrm{encode}(0) + \tilde{v}'$
7: $c_1 \leftarrow \mathrm{encode}(1) + \tilde{v}'$
8: $b^* \xleftarrow{\$} \mathcal{U}(\{0,1\})$
9: **return** $(\mathbf{A}, \tilde{\mathbf{b}}, \tilde{\mathbf{b}}', c_{b^*})$

IND-CPA security of Lindner–Peikert

Game 2: → Decision-LWE →

1: $\mathbf{A} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n\times n})$
2: $\tilde{\mathbf{b}} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^{n})$
3: $\mathbf{s}' \xleftarrow{\$} \chi(\mathbb{Z}_q^{n})$
4: $[\mathbf{e}'\|e''] \xleftarrow{\$} \chi(\mathbb{Z}_q^{n+1})$
5: $[\tilde{\mathbf{b}}'\|\tilde{v}'] \leftarrow \mathbf{s}'[\mathbf{A}\|\tilde{\mathbf{b}}] + [\mathbf{e}'\|e'']$
6: $c_0 \leftarrow \mathrm{encode}(0) + \tilde{v}'$
7: $c_1 \leftarrow \mathrm{encode}(1) + \tilde{v}'$
8: $b^* \xleftarrow{\$} \mathcal{U}(\{0,1\})$
9: **return** $(\mathbf{A}, \tilde{\mathbf{b}}, \tilde{\mathbf{b}}', c_{b^*})$

Game 3: → Rewrite →

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

Game 4:

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

Independent of hidden bit

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, 2017-2021]

FrodoKEM construction

IND-CPA secure FrodoPKE

FrodoPKE.KeyGen

FrodoPKE.Enc

FrodoPKE.Dec

Algorithm 9 FrodoPKE.KeyGen.

Input: None.

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

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

FrodoKEM construction

IND-CPA secure FrodoPKE

FrodoPKE.KeyGen

FrodoPKE.Enc

FrodoPKE.Dec

Algorithm 9 FrodoPKE.KeyGen.

Input: None.

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

1: Choose a uniformly random seed $\mathsf{seed}_\mathbf{A} \leftarrow\!\!\$\ U(\{0,1\}^{\mathsf{len}_\mathbf{A}})$

2: Generate the matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$ via $\mathbf{A} \leftarrow \mathsf{Frodo.Gen}(\mathsf{seed}_\mathbf{A})$

3: Choose a uniformly random seed $\mathsf{seed}_\mathbf{E} \leftarrow\!\!\$\ U(\{0,1\}^{\mathsf{len}_\mathbf{E}})$

4: Sample error matrix $\mathbf{S} \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_\mathbf{E}, n, \overline{n}, T_\chi, 1)$

5: Sample error matrix $\mathbf{E} \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_\mathbf{E}, n, \overline{n}, T_\chi, 2)$

6: Compute $\mathbf{B} = \mathbf{AS} + \mathbf{E}$

7: **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 $c = (\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{\overline{m} \times n} \times \mathbb{Z}_q^{\overline{m} \times \overline{n}}$.

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

Basic LWE ciphertext

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 $c = (\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{\overline{m} \times n} \times \mathbb{Z}_q^{\overline{m} \times \overline{n}}$.

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

Basic LWE ciphertext

Shared secret

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 $c = (\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{\overline{m} \times n} \times \mathbb{Z}_q^{\overline{m} \times \overline{n}}$.

1: Generate $\mathbf{A} \leftarrow \mathsf{Frodo.Gen}(\mathsf{seed}_{\mathbf{A}})$
2: Choose a uniformly random seed $\mathsf{seed}_{\mathbf{E}} \leftarrow\!\!\$\ U(\{0,1\}^{\mathsf{len}_{\mathbf{E}}})$
3: Sample error matrix $\mathbf{S}' \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_{\mathbf{E}}, \overline{m}, n, T_\chi, 4)$
4: Sample error matrix $\mathbf{E}' \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_{\mathbf{E}}, \overline{m}, n, T_\chi, 5)$
5: Sample error matrix $\mathbf{E}'' \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_{\mathbf{E}}, \overline{m}, \overline{n}, T_\chi, 6)$
6: Compute $\mathbf{B}' = \mathbf{S}'\mathbf{A} + \mathbf{E}'$ and $\mathbf{V} = \mathbf{S}'\mathbf{B} + \mathbf{E}''$
7: **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 $\mu' \in \mathcal{M}$.

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

FrodoKEM construction

IND-CPA secure FrodoPKE

FrodoPKE.KeyGen

FrodoPKE.Enc

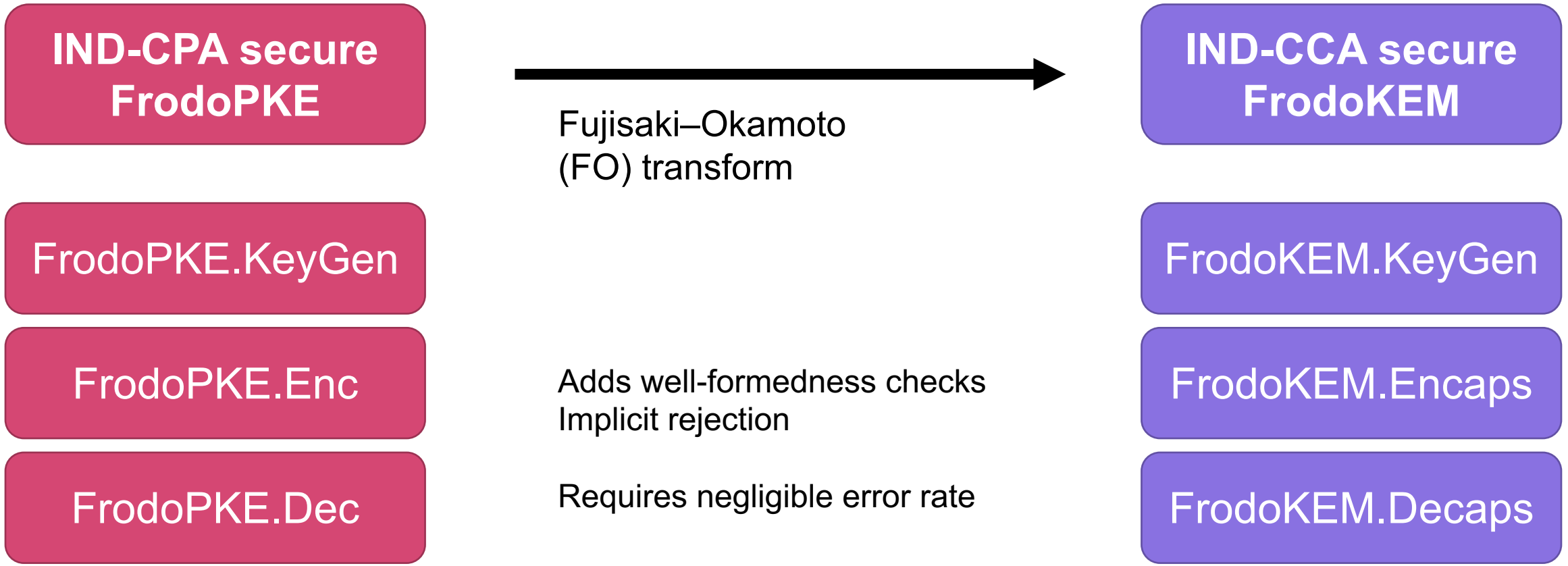
FrodoPKE.Dec

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 $\mu' \in \mathcal{M}$.

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

FrodoKEM construction



FrodoKEM parameters

	FrodoKEM-640	FrodoKEM-976
Dimension n	640	976
Modulus q	2^{15}	2^{16}
Error distribution	Approx. Gaussian [-12, ..., 12], σ = 2.8	Approx. Gaussian [-10, ..., 10], σ = 2.3
Failure probability	2^{-138}	2^{-199}
Ciphertext size	9,720 bytes	15,744 bytes
Estimated security (cryptanalytic)	2^{145} classical 2^{132} quantum	2^{210} classical 2^{191} quantum
Runtime (encaps; AES)	0.48 msec	0.89 msec

3. Cryptography from learning with errors

Digital signature schemes

Signature schemes from identification schemes

- Identification scheme: interactive protocol between a prover and a verifier in which the prover authenticates to the verifier.

- Setup:
  - Prover: generates long term key pair

$$(vk, sk) \xleftarrow{\$} \mathrm{KeyGen}()$$

  - Verifier: obtains a copy of the verification key

General flow of an identification scheme

$$
\begin{array}{lcl}
\underline{\mathrm{Prover}(sk)} & & \underline{\mathrm{Verifier}(vk)} \\
(Y, y) \xleftarrow{\$} \mathrm{Cmt}(sk) & \xrightarrow{Y} & \\
 & \xleftarrow{ch} & ch \xleftarrow{\$} \mathrm{Ch}(Y) \\
z \xleftarrow{\$} \mathrm{Resp}(sk, y, ch) & \xrightarrow{z} & \\
 & & \mathrm{Vfy}(vk, Y, ch, z)
\end{array}
$$

Identification Schemes: Security

Security goal: secure against impersonations under passive attack (IMP-PA).

Passive attack	Impersonations
• adversary gets transcripts of protocol executions	• hard to make an honest verifier accept

Schnorr identification scheme

$\underline{\text{KeyGen}():}$

1: $x \xleftarrow{\$} \mathbb{Z}_q$
2: $X \xleftarrow{\$} g^x$
3: **return** $(vk, sk) \leftarrow (X, x)$

$\underline{\text{Prover}(sk)}$		$\underline{\text{Verifier}(vk)}$
$y \xleftarrow{\$} \mathbb{Z}_q$		
$Y \leftarrow g^y$	$\xrightarrow{Y}$	
	$\xleftarrow{ch}$	$ch \xleftarrow{\$} \mathbb{Z}_q$
$z \leftarrow x \cdot ch + y \mod q$	$\xrightarrow{z}$	
		$Y \stackrel{?}{=} g^z X^{-ch}$

Fiat–Shamir transform

- Turn an interactive identification scheme into a non-interactive signature scheme
- The prover generates the challenge herself
  - Challenge is H(commitment, message)

- **Theorem**: If H is a random oracle and the identification scheme is IMP-PA secure, then the resulting signature scheme is existentially unforgeable under chosen message attack.

Schnorr signature scheme

- Apply Fiat–Shamir transform to Schnorr identification scheme
- The DSA and ECDSA signature schemes are Schnorr-like signature schemes

$\underline{\mathrm{Sign}(sk = x, m)}$

1: $y \xleftarrow{\$} \mathbb{Z}_q$
2: $Y \leftarrow g^y$
3: $ch \leftarrow H(Y \| m)$
4: $z \leftarrow x \cdot ch + y \mod q$
5: **return** $\sigma \leftarrow (Y, z)$

$\underline{\mathrm{Vfy}(vk = X, m, \sigma = (Y, z))}$

1: $ch \leftarrow H(Y \| m)$
2: **return** $(Y = g^z X^{-ch})$

Lyubashevsky's Identification Scheme

- KeyGen:
  1. Choose $\mathbf{s} \leftarrow_{\$} \{0,1\}^m$, $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n \times m}$.
  2. Secret key is $\mathbf{s}$
  3. Public key is $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} \bmod q)$
- $\mathrm{Cmt}(pk)$:
  1. Choose $\mathbf{v} \leftarrow_{\$} \{0,1,\ldots,5m-1\}^m$
  2. Commitment is $\mathbf{y} = \mathbf{A}\mathbf{v} \bmod q$
- $\mathrm{Ch}(pk)$:
  1. Choose $c \leftarrow_{\$} \{0,1\}$
  2. Challenge is c
- $\mathrm{Resp}(pk, sk = \mathbf{s}, c, \mathbf{y})$:
  1. Compute $\mathbf{z} = c \cdot \mathbf{s} + \mathbf{v} \bmod q$
  2. If $\mathbf{z}$ is safe ($\mathbf{z} \in \{1,\ldots,5m-1\}^m$), then return $\mathbf{z}$
  3. Else, abort
- $\mathrm{Vfy}(pk = (\mathbf{A}, \mathbf{b}), \mathbf{y}, c, \mathbf{z})$:
  1. Accept iff $\mathbf{A}\mathbf{z} = c \cdot \mathbf{b} + \mathbf{y} \bmod q$ and $\|\mathbf{z}\| \leq 5m^{1.5}$

Correctness of Lyubashevsky's ID scheme

- The check $\mathbf{z} \in \{1, \ldots, 5m-1\}^m$ avoids degenerate case that might leak information about the secret key
- For m > 10, this check passes with probability at least 81%; can repeat to amplify success probability

Security of Lyubashevsky's ID scheme

- Lyubashevsky's identification scheme is IMP-PA-secure if the SIS problem is hard with parameter beta=$15m^{1.5}$.

Short integer solution problem

- Parameters:
  - Dimensions n, m
  - Modulus q
  - Integer beta < q
- Pick $A \leftarrow_{\$} \mathbb{Z}_q^{n \times m}$
- Find non-zero $\mathbf{v} \in \mathbb{Z}^m$ such that:
  - $A\mathbf{v} \equiv 0 \bmod q$
  - $\|\mathbf{v}\| \leq \beta$

Short integer solution problem

- Without the length constraint, $\|\mathbf{v}\| \leq \beta$
  SIS can be easily solved using Gaussian elimination
- SIS can be viewed as a short vector problem in the dual lattice of A

- As with LWE, can make
  - Ring-SIS
  - Module-SIS

Constructing a lattice-based signature scheme

- Could construct a lattice-based signature scheme by applying the Fiat–Shamir transform to a lattice-based identification scheme, but the generic transform is rather inefficient

- Use a direct construction
- Similar, but somewhat different

Lyubashevsky's signature scheme

KeyGen:

1. Choose $\mathbf{S} \leftarrow_{\$} \{0, \pm 1, \ldots, \pm d\}^{m \times k}$
2. Choose $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n \times m}$
3. Secret key is $\mathbf{S}$
4. Public key is $(\mathbf{A}, \mathbf{T} = \mathbf{AS} \bmod q)$

Sign:

1. Sample $\mathbf{y} \leftarrow_{\$} D_\sigma^m$
2. Compute $\mathbf{c} \leftarrow H(\mathbf{Ay}, \mu)$ where $H : \{0,1\}^* \rightarrow \{\mathbf{v} : \mathbf{v} \in \{0, \pm 1\}^k, \|\mathbf{v}\|_1 \leq \kappa\}$
3. Compute $\mathbf{z} \leftarrow \mathbf{Sc} + \mathbf{y}$
4. Output $(\mathbf{z}, \mathbf{c})$ with probability $\min\left(\frac{D_\sigma^m(\mathbf{z})}{M D_{\mathbf{Sc},\sigma}^m(\mathbf{z})}, 1\right)$ otherwise restart signing

Vfy:

1. Accept iff $\|\mathbf{z}\|_2 \leq \eta\sigma\sqrt{m}$ and $\mathbf{c} = H(\mathbf{Az} - \mathbf{Tc}, \mu)$

Rejection sampling

- Line 4 of the Sign algorithm is a technique called "rejection sampling"

- The (surprising) effect of this rejection sampling step is to cause the distribution of outputs to be statistically independent of the secret key S
- But it still outputs something with fairly high probability (just a few repetitions required on average)

Correctness and security of Lyubashevsky's signature scheme

Correctness

- With probability at least $1 - 2^{-m}$, $\|\mathbf{z}\|_2 \leq \eta\sigma\sqrt{m}$

Security

- **Theorem**: If H is a random oracle and the SIS problem is hard, then Lyubashevsky's signature scheme is existentially unforgeable under chosen message attack.

Derivations

- Lyubashevsky's signature scheme is the basis of many lattice-based schemes:
  - BLISS (Ducas et al. CRYPTO 2013)
  - Bai and Galbraith (CT-RSA 2014)
  - TESLA, qTESLA (using ring-SIS)
  - NIST Round 3 finalist:
    - **CRYSTALS-Dilithium** (using module-SIS)

RSA full domain hash digital signatures

KeyGen

1. Alice picks two large primes p and q and computes n = pq
2. Alice picks a value e and computes $d = e^{-1} \bmod \phi(n)$
3. Public verification key: vk = (n, e)
4. Secret signing key: sk = (n, d)

Signing

For Alice to sign a message m:

1. Alice hashes m to get H(m) in {1, …, n}
2. Alice computes $sig = H(m)^d \bmod n$
3. Alice sends (m, sig) to Bob

Verification

For Bob to verify a signature (m', sig'):

1. Bob gets an authentic copy of Alice's public verification key vk = (n, e)
2. Bob computes $h = (sig')^e \bmod n$
3. Bob checks if H(m') = h

GPV framework

KeyGen():

1. Using a "trapdoor", generate $A \in \mathbb{Z}_q^{n \times m}$ and $B \in \mathbb{Z}_q^{m \times m}$ such that $B \times A^T = 0 \bmod q$

Vfy$(A, \mu, \mathbf{s})$:

1. Accept iff $\mathbf{s}A^T = H(\mu)$ and $\mathbf{s}$ is short

Sign(B, μ):

1. Compute $\mathbf{c}_0$ such that $\mathbf{c_0}A^T = H(\mu)$ (using standard linear algebra)
2. Use B to compute a vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ close to $\mathbf{c}$
3. Signature is $\mathbf{s} = \mathbf{c}_0 - \mathbf{v}$

Gentry, Peikert and Vaikuntanathan, ACM STOC 2008

GPV framework

- First instantiated by the GGH97 and NTRUSign (2003) schemes
- But those were completely insecure

- GPV is provably secure in the random oracle model under the SIS assumption
- GPV computes v in a subtly different but important way

- NIST Round 3 candidate:
  - Falcon (GPV framework using NTRU)

3. Cryptography from learning with errors

Advanced constructions

Fully homomorphic encryption from LWE

- KeyGen(): $\mathbf{s} \xleftarrow{\$} \chi(\mathbb{Z}_q^n)$
- $\mathrm{Enc}(sk, \mu \in \mathbb{Z}_2)$: 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 $e \equiv \mu \bmod 2$.
- $\mathrm{Dec}(sk, \mathbf{c})$: 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

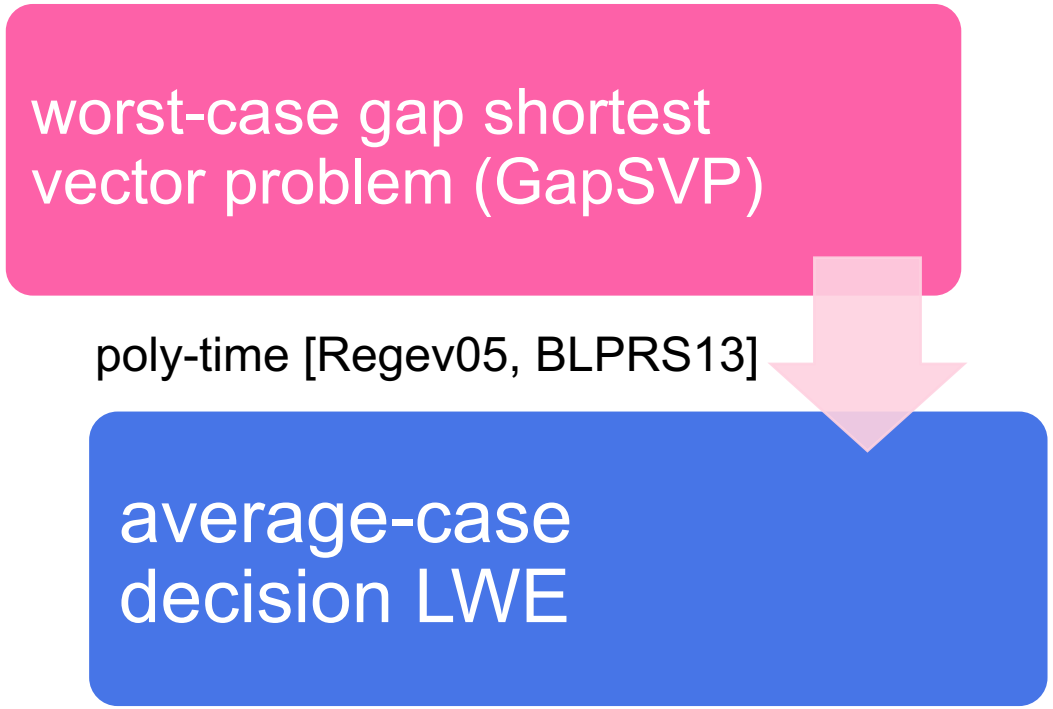
- Error conditions mean that the number of additions and multiplications is limited.
- Multiplication increases the dimension (exponentially), so the number of multiplications is again limited.

- 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.

4. Difficulty of LWE

Lattice problems

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**

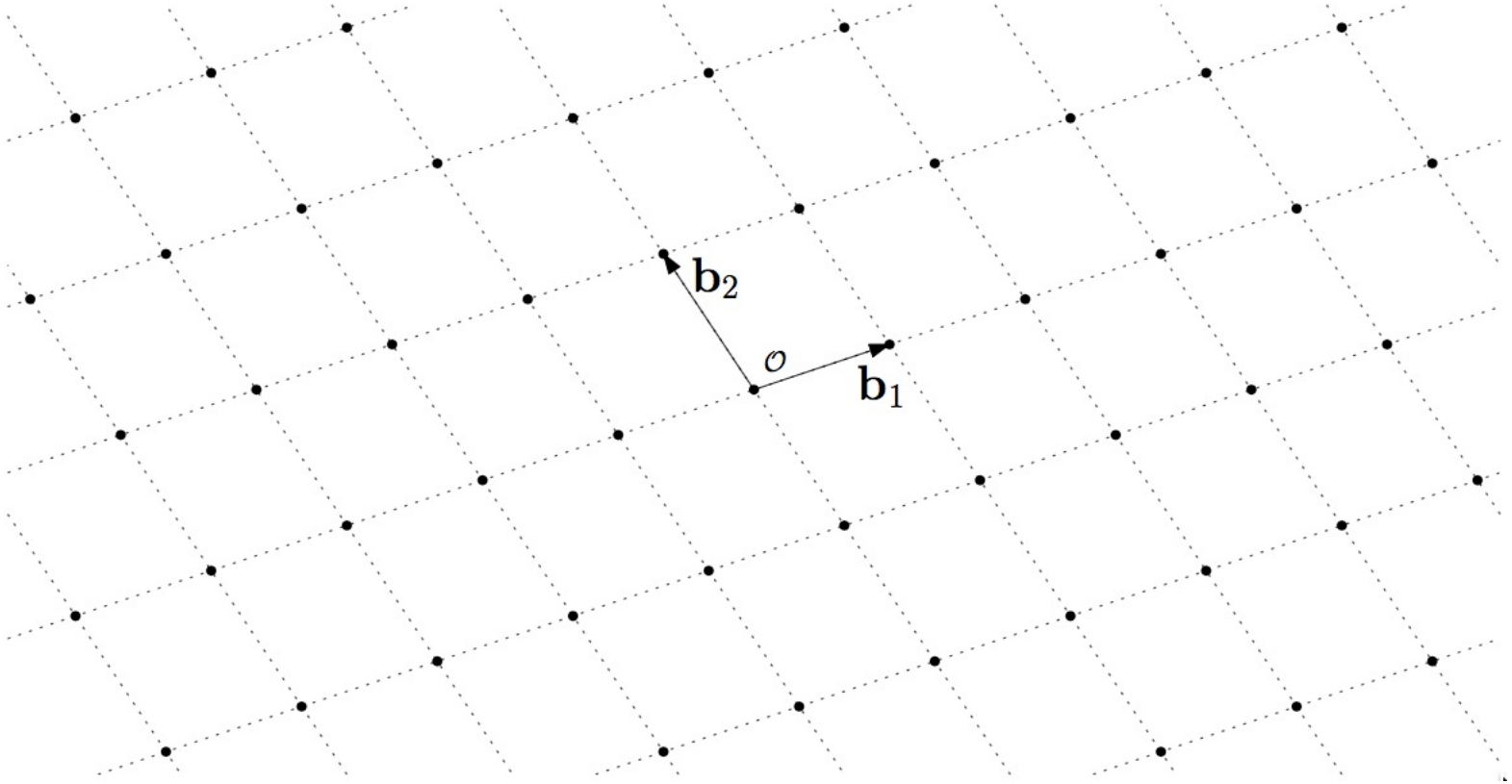
$$\mathcal{L} = \mathcal{L}(\mathbf{B}) = \left\{ \sum_{i=1}^{n} z_i \mathbf{b}_i : z_i \in \mathbb{Z} \right\} \quad .$$

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

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

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

Lattices

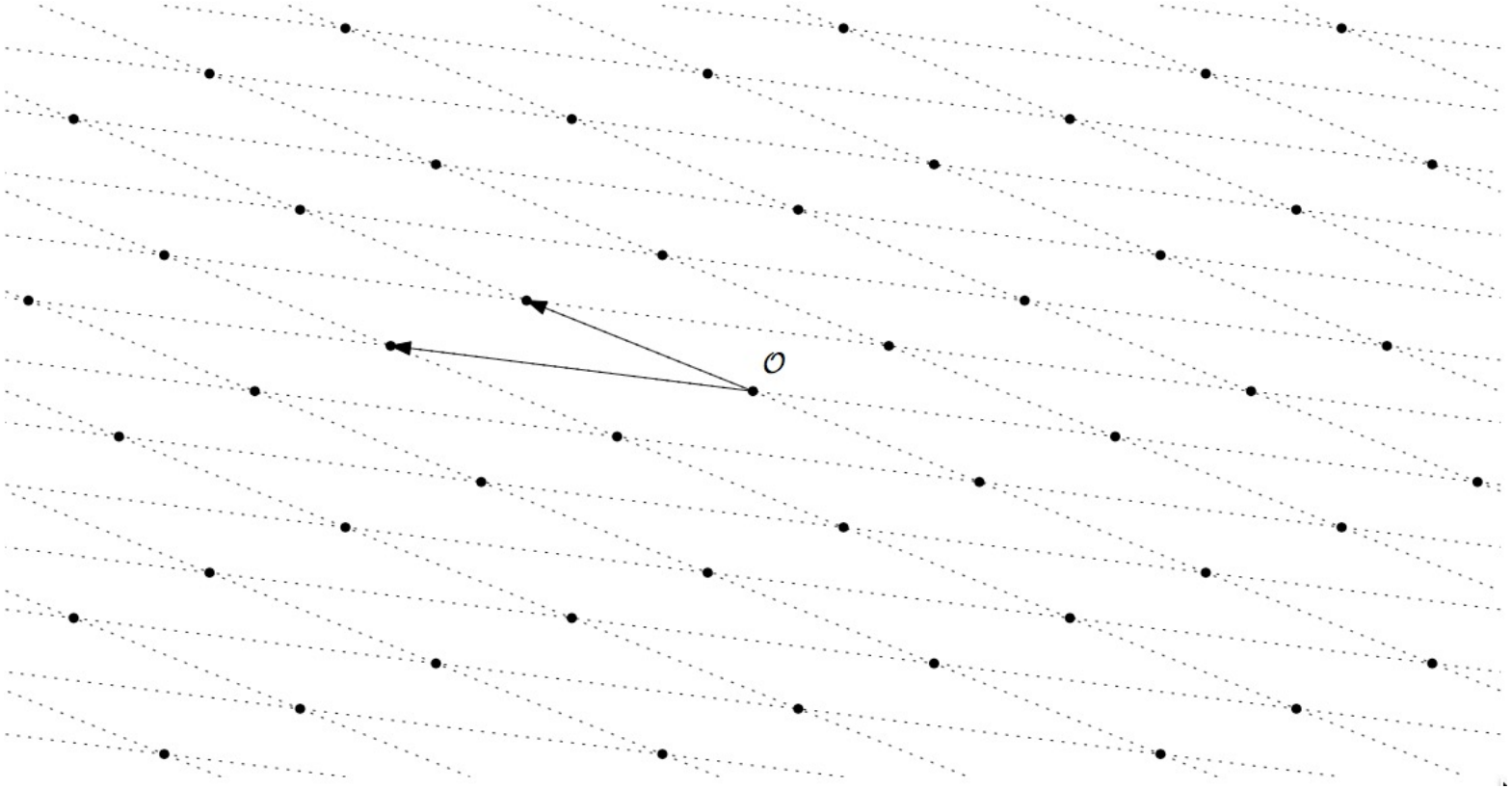


Discrete additive subgroup of $\mathbb{Z}^n$

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

Equivalence of bases

Two $n \times n$ matrices B and B' generate the same lattice $\mathcal{L}$ if and only if B and B' are related by a unimodular matrix, i.e. $B' = BU$ where U is a $n \times n$ matrix with integer entries and determinant ± 1.

Hermite normal form

Definition. An $m \times n$ matrix A is in **Hermite normal form** if (informally) it is lower triangular and its largest entry in each row is on the diagonal.

Fact. The HNF H of an integer matrix A is unique, and there is an $n \times n$ unimodular matrix U such that $H = AU$.

Fundamental parallelepiped

Definition. The **fundamental parallelepiped** of the set $\{\vec{b}_1, \ldots, \vec{b}_n\} \subseteq \mathbb{R}^n$ is the set

$$\left\{ \sum_{i=1}^{n} x_i \vec{b}_i : 0 \leq x_i < 1 \right\} .$$

Fact. The volume of the fundamental parallelepiped of $\{\vec{b}_1, \ldots, \vec{b}_n\}$ is $|\det(B)|$ where B is the column matrix.

Lemma [Galbraith, Lemma 16.1.9]. The volume of the fundamental parallelepiped of a lattice is independent of the choice of basis.

Successive minima

Definition. Let $\mathcal{L}$ be a full rank-lattice of dimension n. The n **successive minima** of $\mathcal{L}$ are $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$ such that each λ_i is minimal such that there exist i linearly independent vectors $\vec{v}_1, \ldots, \vec{v}_i \in \mathcal{L}$ with $\|\vec{v}_j\| \leq \lambda_j$ for $1 \leq j \leq i$. Sometimes we write $\lambda_i = \lambda_i(\mathcal{L})$.

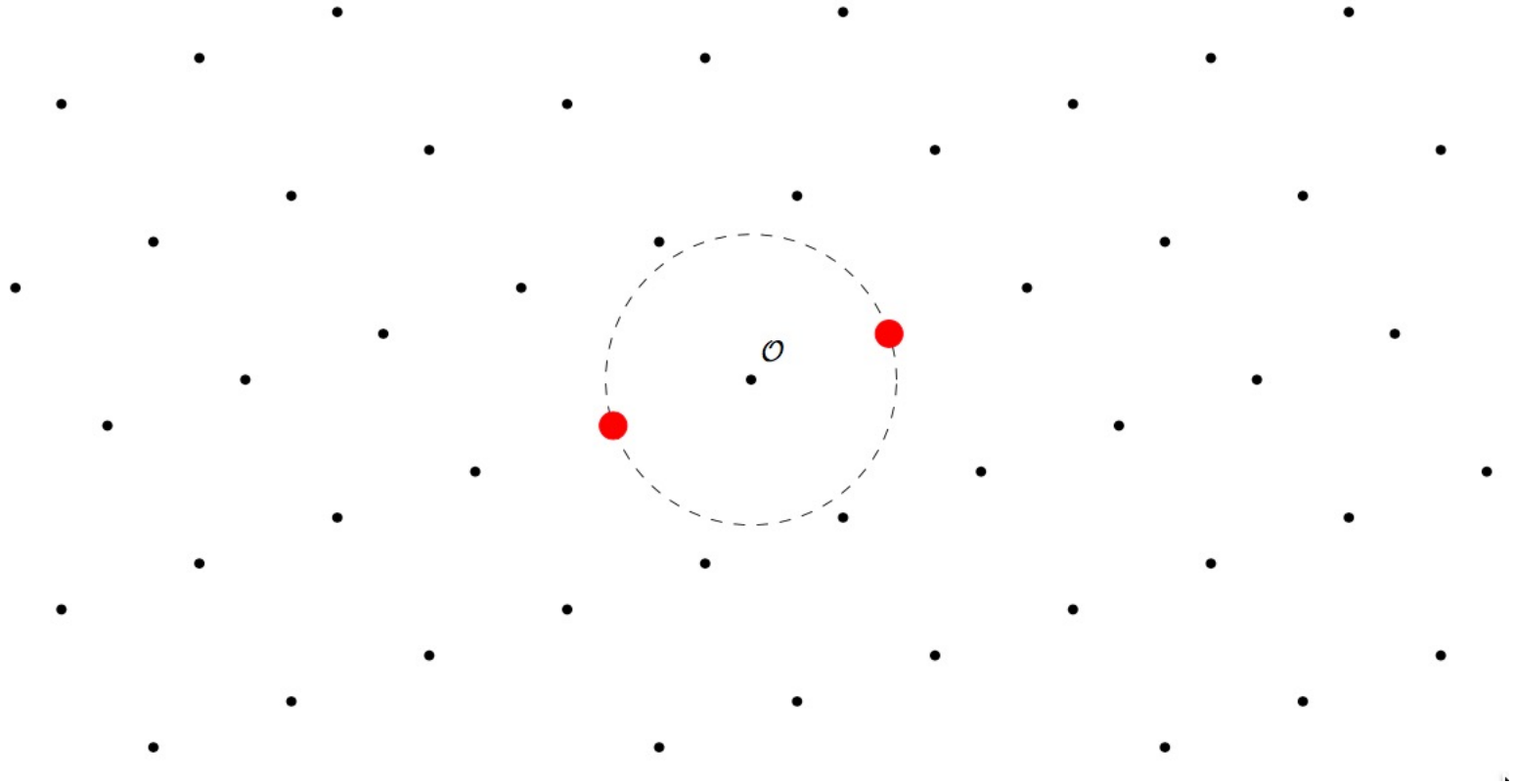
In other words, λ_1 is the length of the shortest non-zero vector in $\mathcal{L}$, and $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.

Easy lattice problems

The following problems can be solved using standard linear algebra techniques:

- **Lattice membership**: Given an $m \times n$ basis matrix B for a lattice $\mathcal{L} \subseteq \mathbb{Z}^m$ and a vector $\vec{v} \in \mathbb{Z}^m$, determine whether $v \in \mathcal{L}$.
- **Lattice basis**: Given a set of possibly linearly dependent vectors $\vec{b}_1, \ldots, \vec{b}_n \in \mathbb{Z}^m$, find a basis for the lattice generated by them.
- **Kernel lattice**: Given an $n \times m$ integer matrix A, compute a basis for the lattice $\ker(A) = \{\vec{x} : A\vec{x} = \vec{0}\}$.
- **Kernel lattice modulo** M: Given an $n \times m$ integer matrix A and an integer M, compute a basis for the lattice $\ker(A) = \{\vec{x} : A\vec{x} \equiv \vec{0} \mod M\}$.

Shortest vector problem



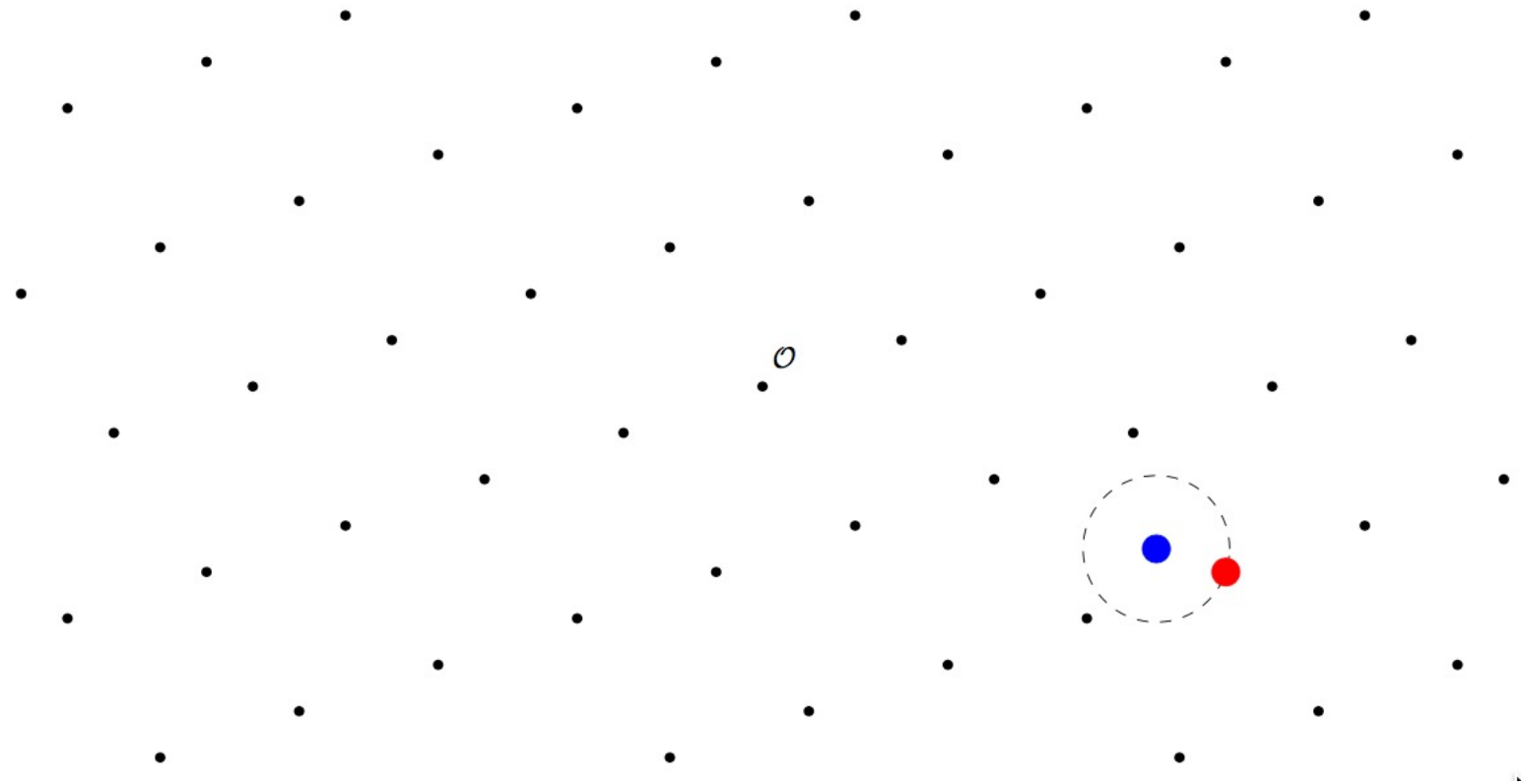
Given some basis for the lattice, find the shortest non-zero lattice point.

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

Shortest vector problems

- **Shortest vector problem (SVP):** Given a basis B for $\mathcal{L}$, find a vector $\vec{v} \in \mathcal{L}$ such that $\|\vec{v}\| = \lambda_1(\mathcal{L})$.
- **Approximate shortest vector problem (SVP_γ):** Fix $\gamma > 1$. Given a basis B for $\mathcal{L}$, find a non-zero vector $\vec{v} \in \mathcal{L}$ such that $\|\vec{v}\| \leq \gamma \cdot \lambda_1(\mathcal{L})$.
- **Decision approximate shortest vector problem (GapSVP_γ):** Fix $\gamma > 1$ and $r > 0$. Given a basis B for $\mathcal{L}$ where either $\lambda_1(\mathcal{L}) \leq r$ or $\lambda_1(\mathcal{L}) \geq \gamma \cdot r$, determine which is the case. Sometimes this is stated with $r = 1$.
- **Shortest independent vector problem (SIVP_γ):** Fix $\gamma > 1$. Given a basis B for a lattice $\mathcal{L}$, find a linearly independent set $\{\vec{v}_1, \ldots, \vec{v}_n\}$ such that $\max_i \|\vec{v}_i\| \leq \gamma \cdot \lambda_n(\mathcal{L})$.

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

Closest vector problems

- **Closest vector problem** (CVP)**:** Given a basis B for $\mathcal{L}$ and a vector $w \in \mathbb{Q}^m$, find a vector $\vec{v} \in \mathcal{L}$ such that $\|\vec{w} - \vec{v}\|$ is minimal.
- **Bounded distance decoding problem** (BDD_α)**:** Fix $0 < \alpha < 1/\sqrt{2}$. Given a basis B for a lattice $\mathcal{L}$ and a vector $w \in \mathbb{Q}^m$ such that there is a lattice point $\vec{v}$ with $\|\vec{w} - \vec{v}\| \leq \alpha\lambda_1(\mathcal{L})$, find $\vec{v}$.

  (This is a CVP instance that is especially close to a lattice point.)

Modular lattices a.k.a. q-ary lattices

Let $B \in \mathbb{Z}^{m \times n}$. Define

$$\mathcal{L}_q(B) = \{\vec{z} \in \mathbb{Z}^m : \vec{z} \equiv B\vec{y} \bmod q \text{ for some } \vec{y} \in \mathbb{Z}^n\} \ .$$

Let $A \in \mathbb{Z}^{n \times m}$. Define

$$\mathcal{L}_q^{\perp}(A) = \{\vec{y} \in \mathbb{Z}^m : A\vec{y} \equiv \vec{0} \bmod q\} \supseteq q\mathbb{Z}^m \ .$$

Modular lattices are related to error correcting codes.

- $\mathcal{L}_q(B)$ corresponds to the code generated by the columns of B.
- $\mathcal{L}_q^{\perp}(A)$ corresponds to the code whose parity check matrix is A.

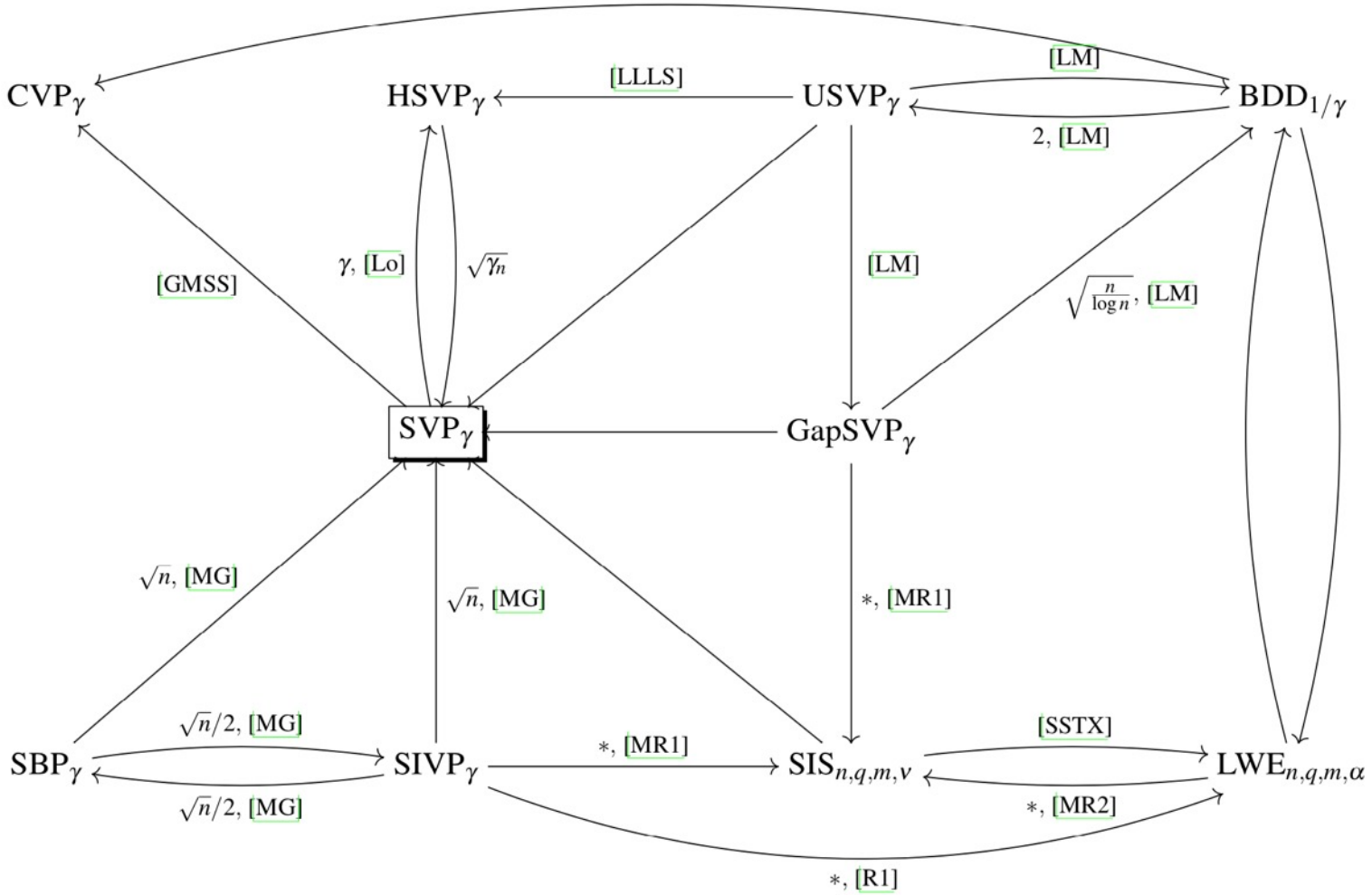
Short integer solutions problem

Definition [Short integer solutions problem ($\mathsf{SIS}_{n,q,m,\beta}$)]. Fix modulus q and $\beta < q$. Given $A \in \mathbb{Z}_q^{n \times m}$, find non-zero $\vec{v} \in \mathbb{Z}^m$ such that $A\vec{v} \equiv \vec{0} \bmod q$ and $\|\vec{v}\| \leq \beta$.

Although this is not formulated directly as a lattice problem, it is a lattice problem in the lattice $\mathcal{L}_q^{\perp}(A)$.

- Without the length constraint $\|\vec{v}\| \leq \beta$, SIS can easily be solved by Gaussian elimination.
- Without the constraint $\beta < q$, $(q, 0, \ldots, 0)$ is trivially a solution.
- Any solution $\vec{x}$ for A can turned into a solution $[\vec{x}|\vec{0}]$ for $[A|A']$ with the same β. Thus SIS becomes easier as m increases.
- SIS becomes harder as n increases.

Relations among lattice problems



Almost all problems reduce to SVP_γ. For example, SIVP_γ reduces to SVP_γ: any method that solves all instances of SVP_γ can be used to solve instances of SIVP_γ, up to a loss of the factor of $\sqrt{n}$ in the subscript.

Laarhoven, van de Pol, and de Weger, Cryptology ePrint Archive 2012/533

Regev's reduction: LWE to shortest vector

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]

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 $\cap$ co-NP	$\geq \sqrt{n}$
	NP-hard	$n^{o(1)}$

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

4. Difficulty of LWE

Cryptanalysis

Strategies for solving LWE

SIS strategy

BDD strategy

Direct strategy

- See Albrecht, Player, Scott for a good survey

Albrecht, Player, Scott. Journal of Mathematical Cryptology 2015. Cryptology ePrint archive 2015/046.

Short integer solution strategy [APS S4.1]

Solve decision LWE by finding a short vector $\vec{v}$ such that $\langle \vec{v}, \vec{a} \rangle = 0$.

- Blum, Kalai, Wasserman algorithm [APS §5.2]: combinatorial method
- Lattice reduction [APS §5.3]: Use lattice reduction to find short vectors in the scaled dual lattice (LLL, BKZ)

If we want to solve search LWE, use the search-decision equivalence in combination with solving decision LWE.

Albrecht, Player, Scott. Journal of Mathematical Cryptology 2015. Cryptology ePrint archive 2015/046.

Bounded distance decoding strategy [APS S4.2]

Solve search LWE by finding a short e such that $\langle \vec{a}, \vec{x} \rangle = b - e$ for some unknown $\vec{x}$.

- Babai's nearest plane algorithm
- Lindner–Peikert nearest planes, BDD by enumeration [APS §5.4]
- Reducing BDD to unique SVP [APS §5.5]: use Kannan's embedding of the LWE lattice into a higher dimensional lattice with an appropriate structure, then solve uSVP e.g. using lattice reduction

Albrecht, Player, Scott. Journal of Mathematical Cryptology 2015. Cryptology ePrint archive 2015/046.

Direct strategy [APS S4.3]

Solve search LWE by finding an $\vec{s}'$ such that $\langle \vec{a}, \vec{s}' \rangle$ is close to b.

- Exhaustive search [APS §5.1]: Exhaustive search for each component of $\vec{s}$ based on the error distribution.
- Arora–Ge [APS §5.6]: solve a system of noiseless non-linear polynomials with $\vec{s}$ as the root

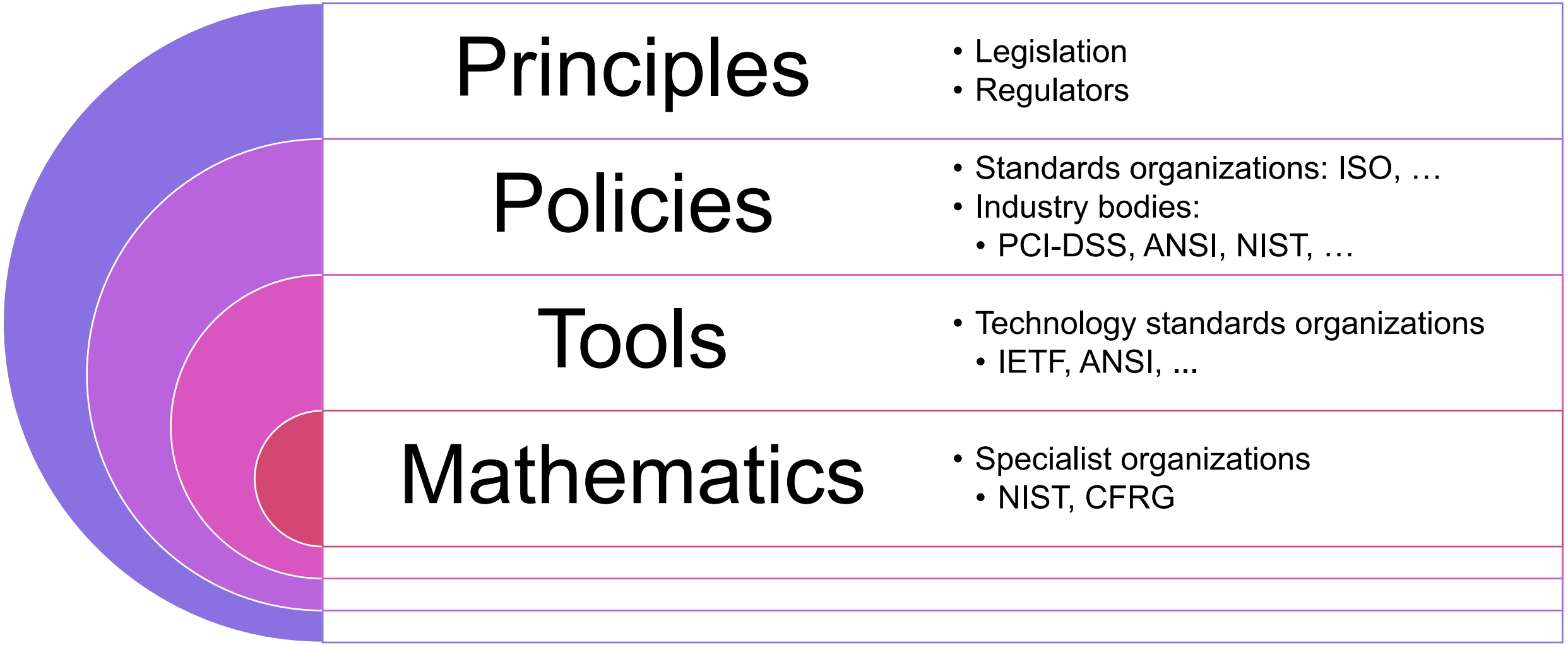
Picking concrete parameters

- Competing requirements:
  - Want small dimension (to reduce communication)
  - Want large dimension (to make problem harder)
  - Want small noise (to reduce probability of error)
  - Want large noise (to make problem harder)
  - Want small modulus (to make problem harder and save communication)
  - Want large modulus (to reduce probability of error)
- Picking concrete parameters is tricky
- Lots to consider and state of art is advancing
- Costing quantum attacks is subtle
- See NTRU and Kyber NIST submissions for worked examples

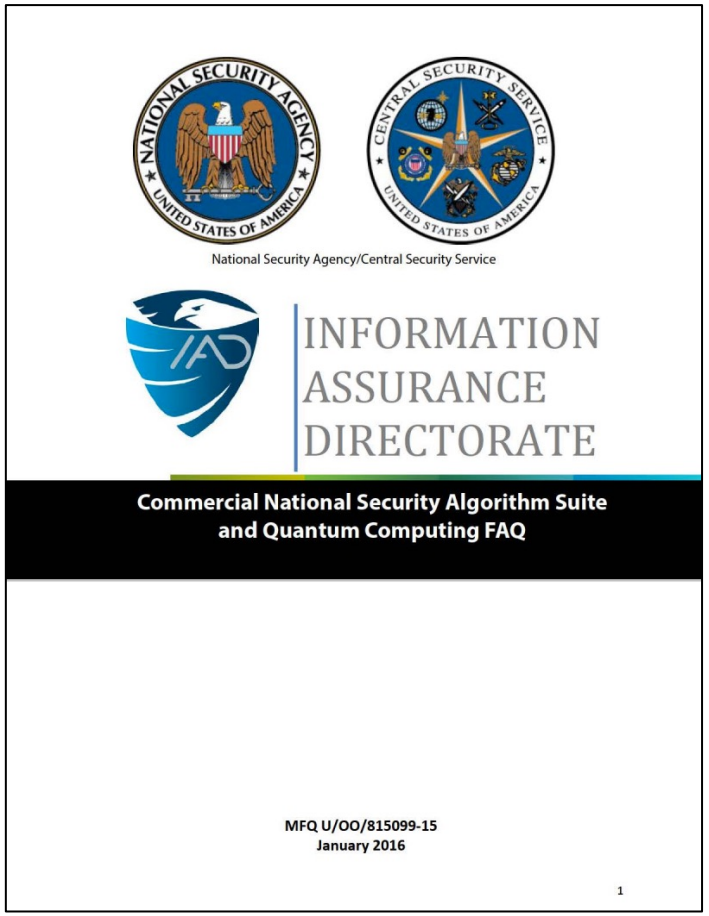
https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions

5. Standardization of PQ cryptography

The path to standardization



Standardizing post-quantum cryptography


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

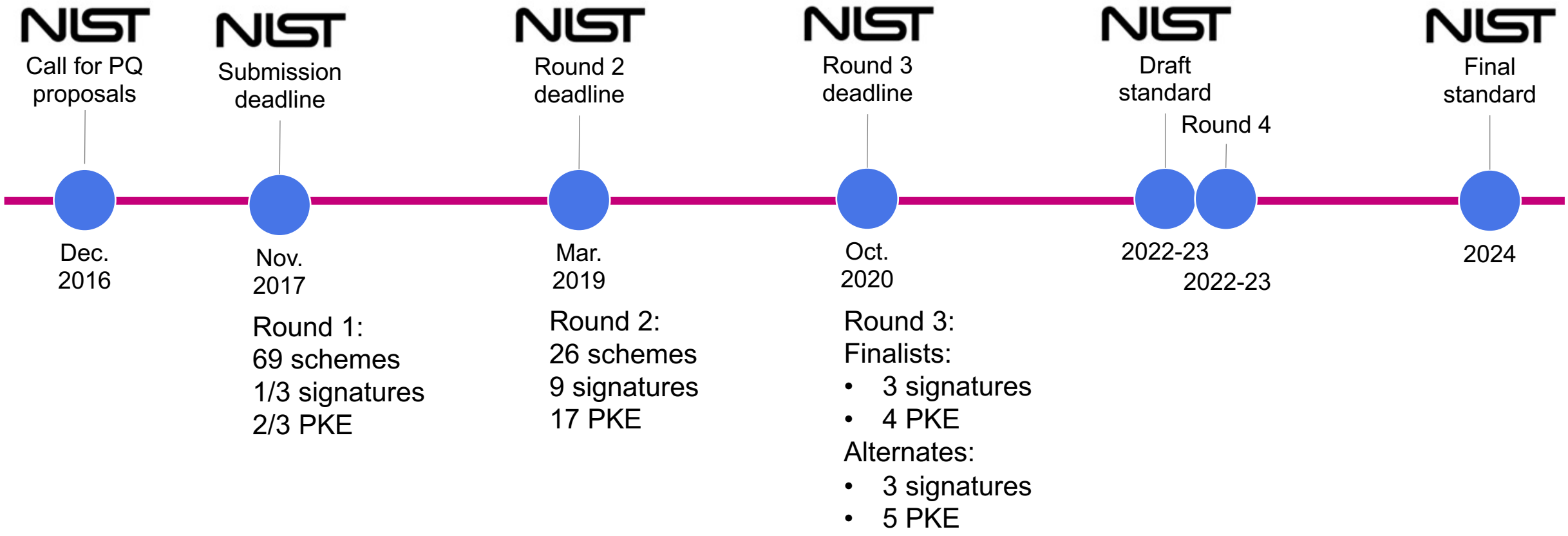
1

Aug. 2015 (Jan. 2016)

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

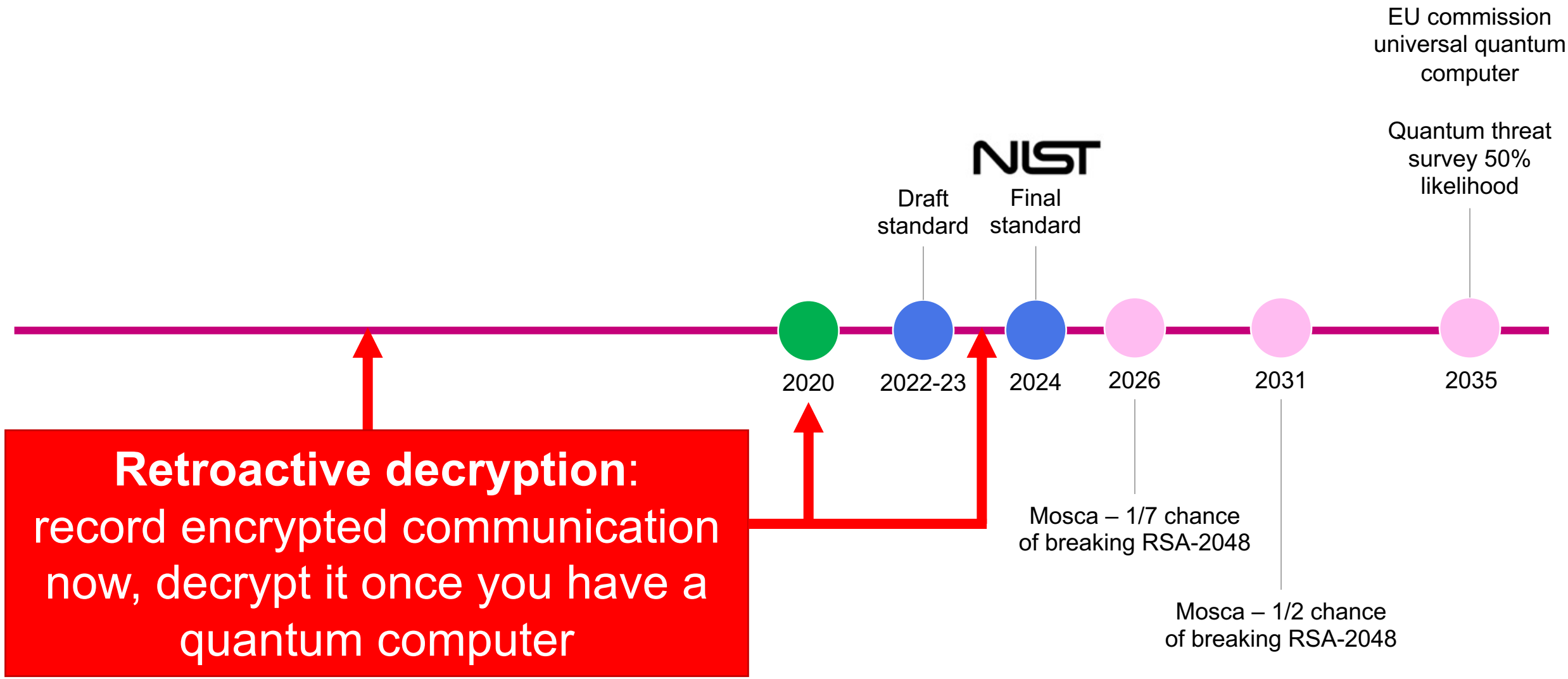
– NSA Information Assurance Directorate, Aug. 2015

NIST Post-quantum Crypto Project timeline

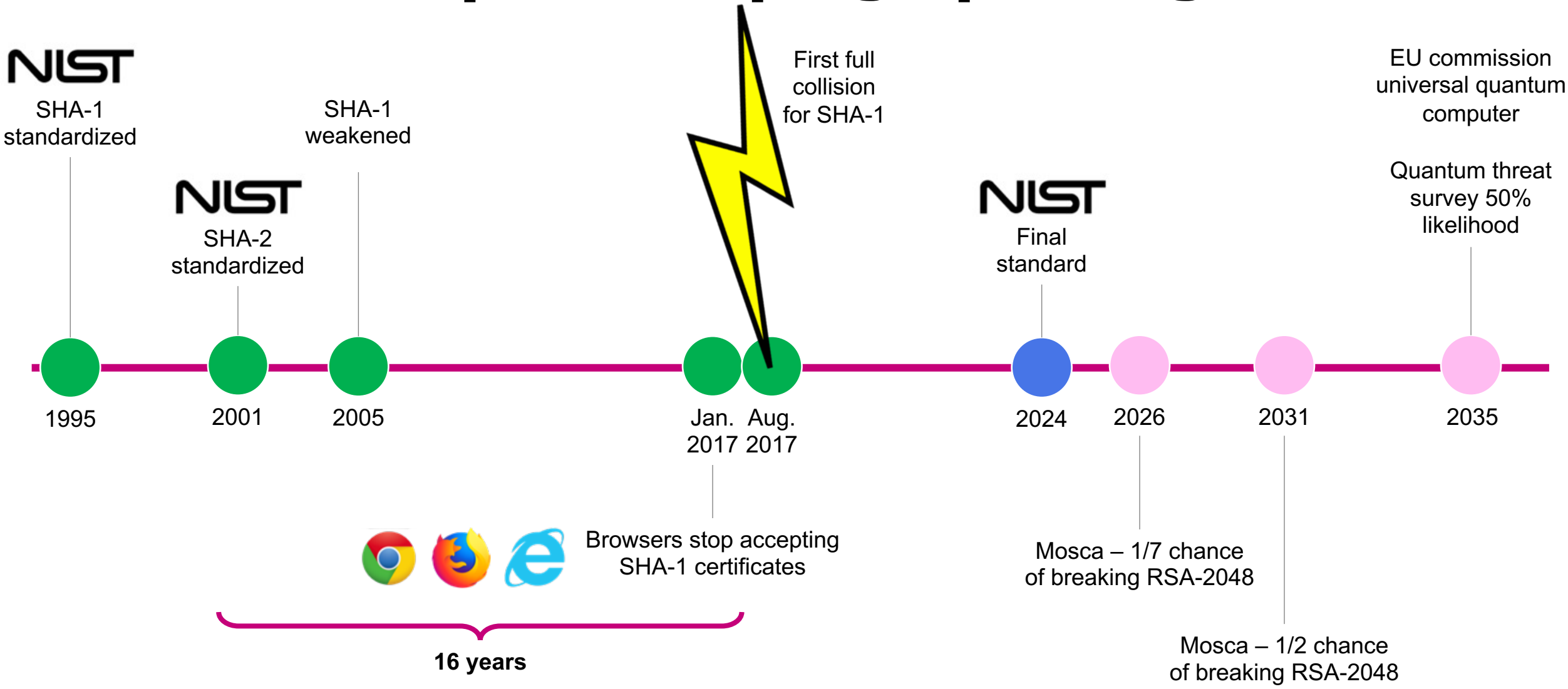


http://www.nist.gov/pqcrypto

Will we be ready in time?



Timeline to replace cryptographic algorithms



NIST Round 3

Finalists

Key encapsulation mechanisms

- Code-based: Classic McEliece
- Lattice-based: Kyber, NTRU, Saber
  - At most one of these 3 will be standardized

Signatures

- Lattice-based: Dilithium, Falcon
  - At most one of these 2 will be standardized
- Multivariate: Rainbow

Alternate candidates

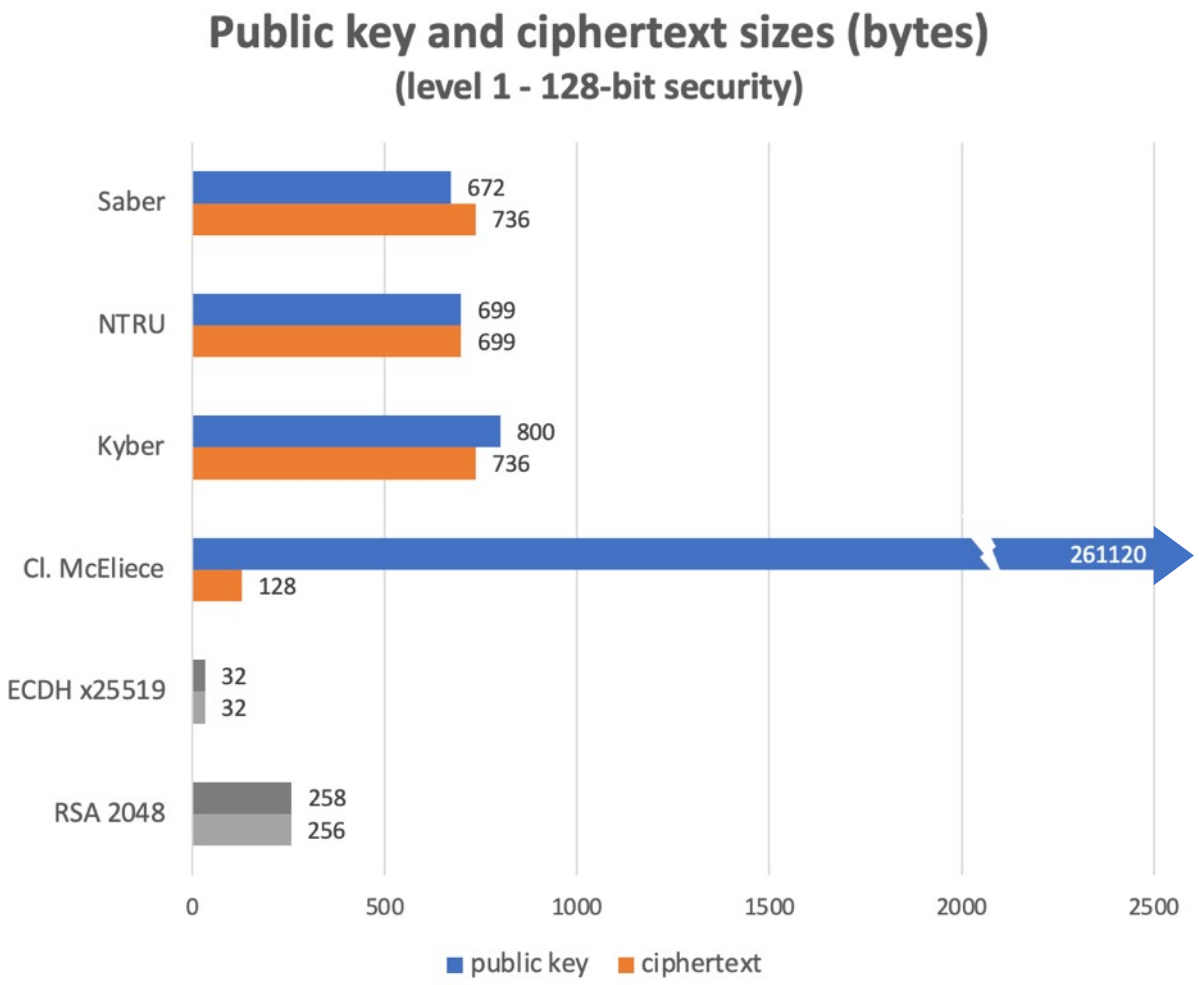
Key encapsulation mechanisms

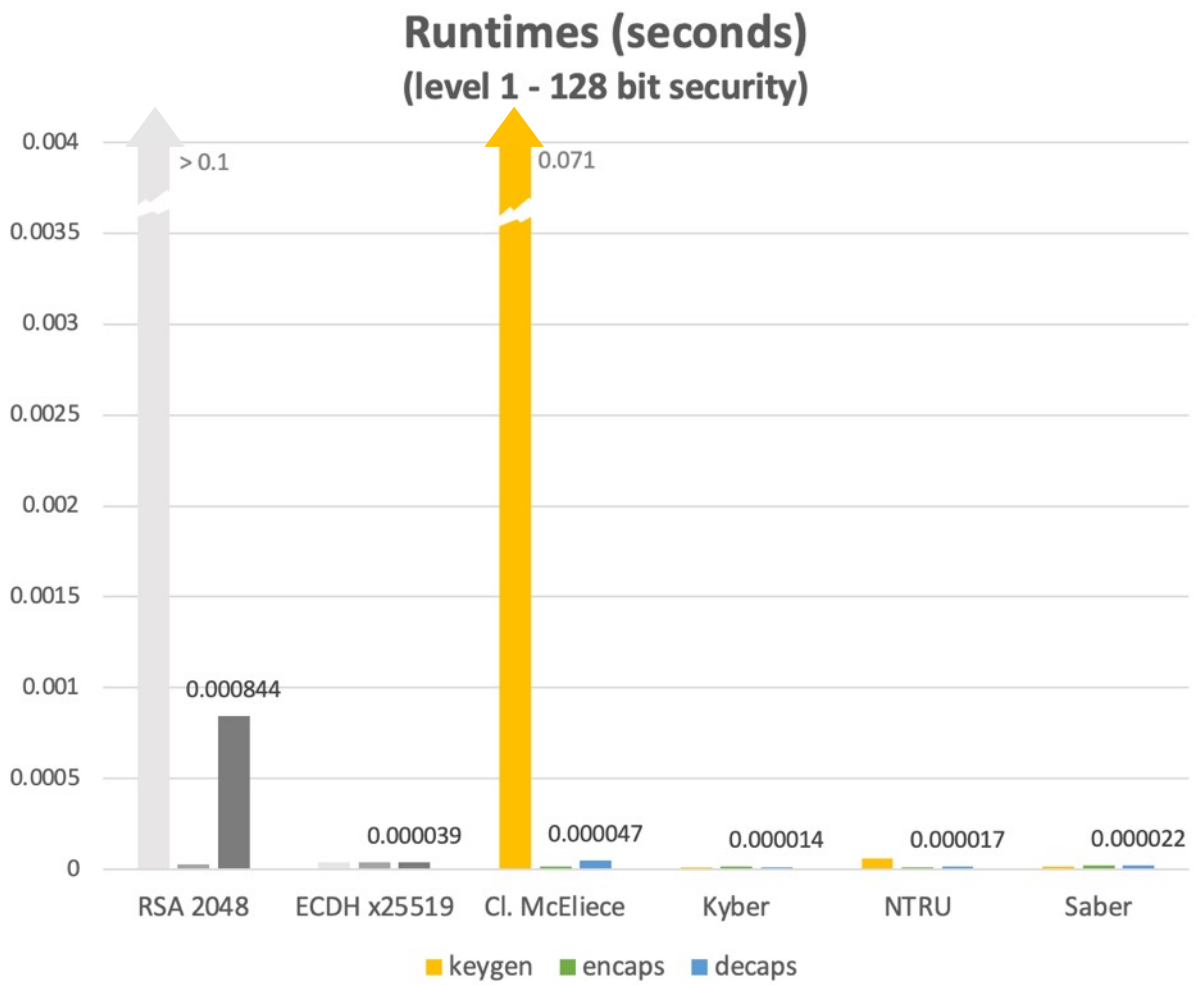
- Code-based: BIKE, HQC
- Lattice-based: FrodoKEM, NTRU Prime
- Isogeny-based: SIKE

Signatures

- Symmetric-based: Picnic, SPHINCS+
- Multivariate: GeMSS

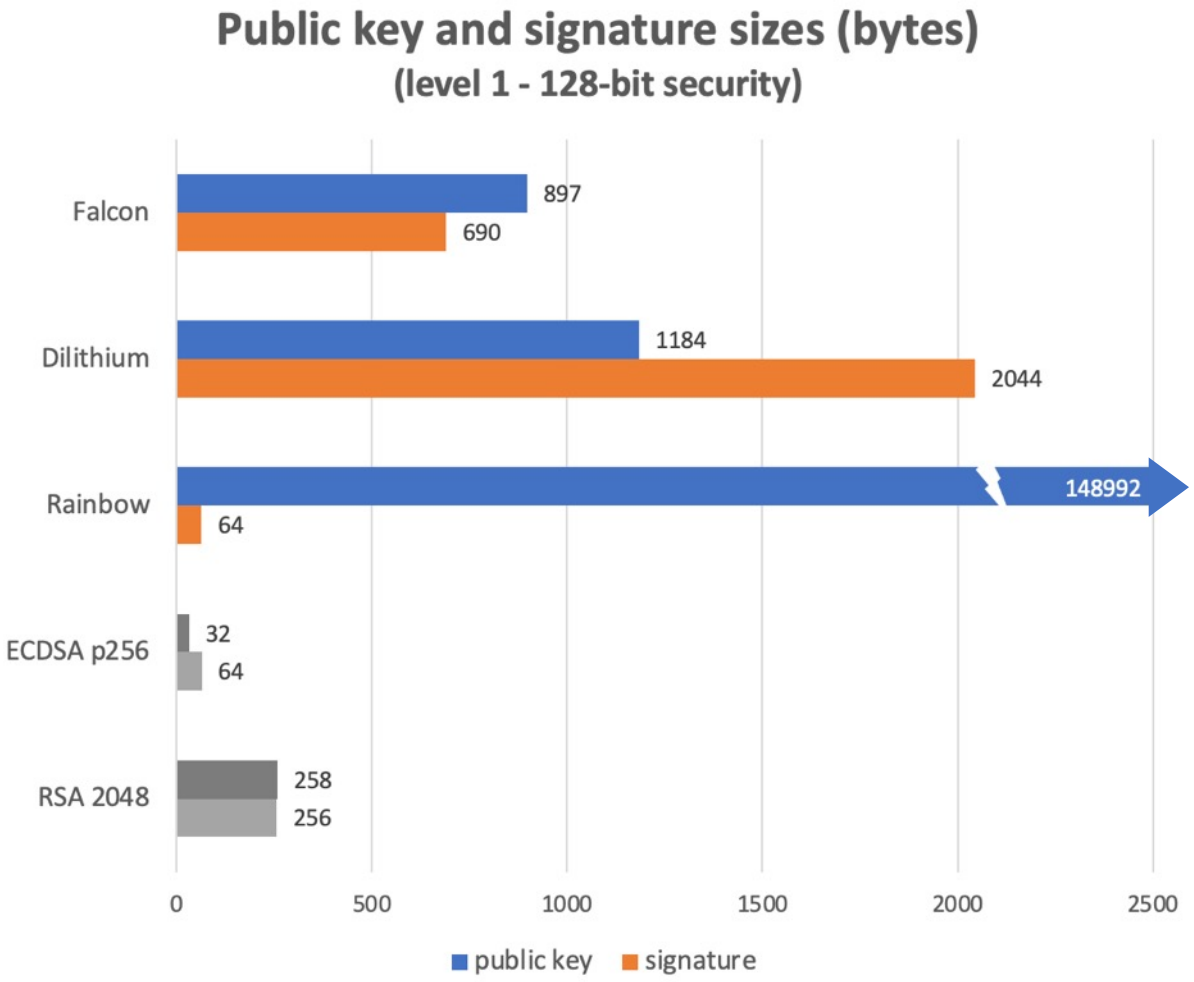
NIST Round 3 KEM Finalists

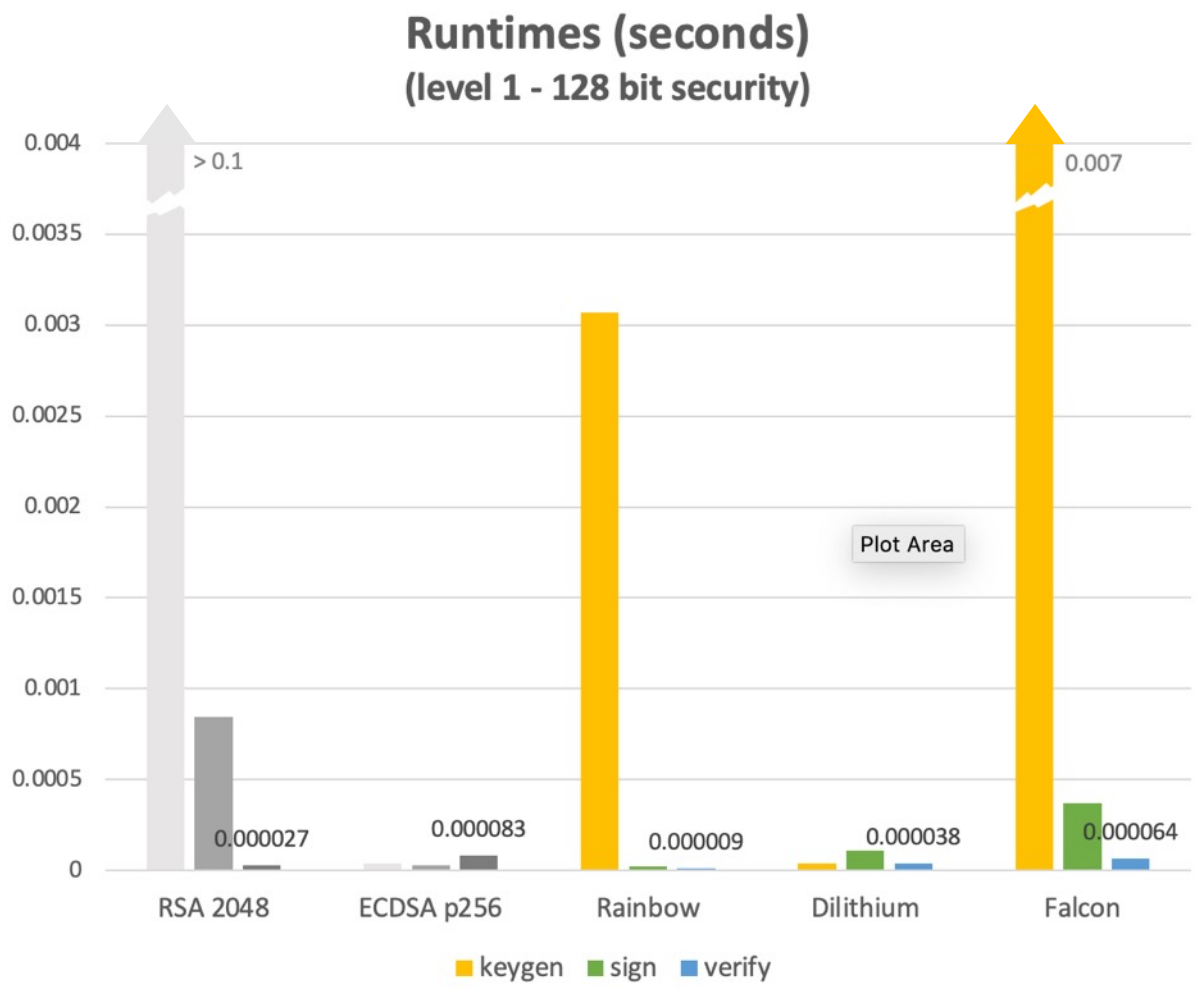




Based on Round 2 submission documents; AVX2 runtimes normalized

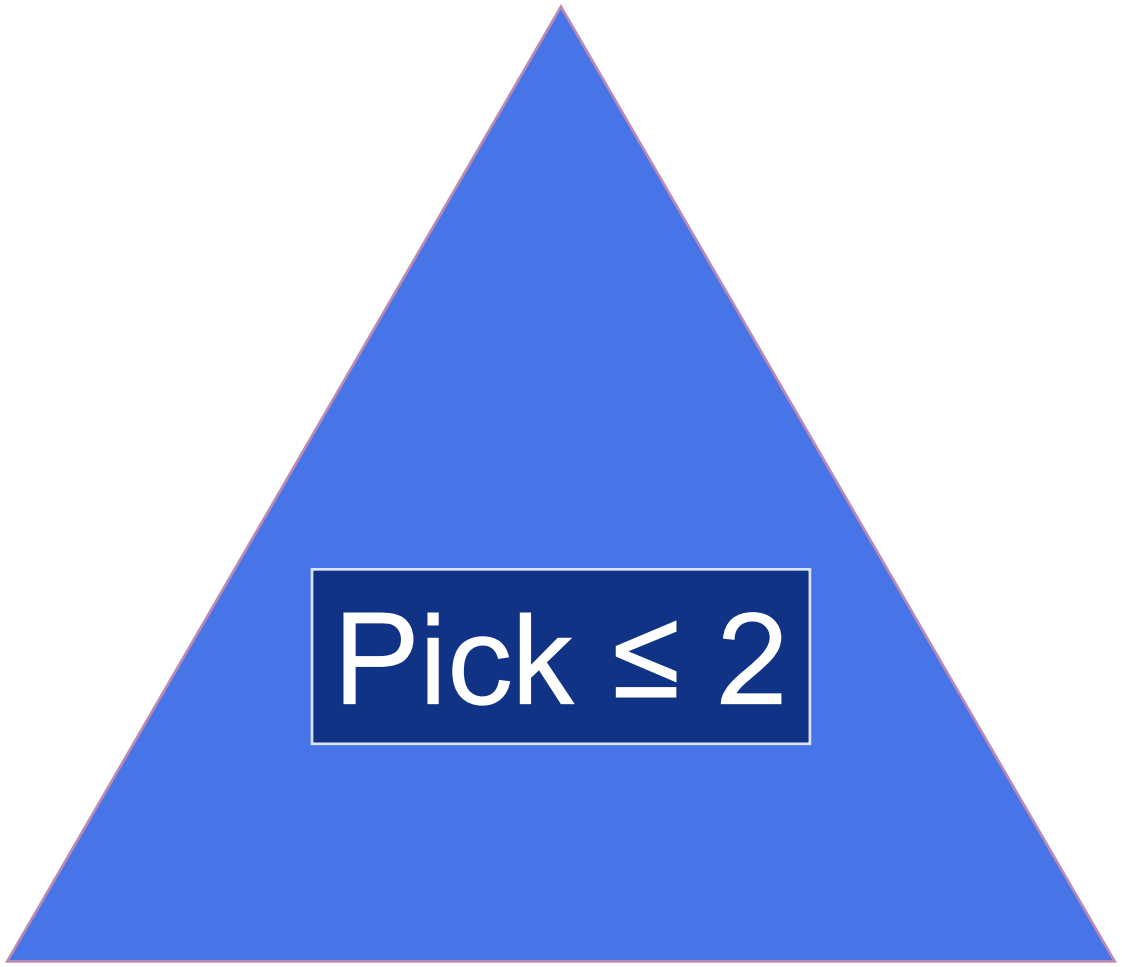
NIST Round 3 Signature Finalists





Based on Round 2 submission documents; AVX2 runtimes normalized

Confidence in quantum-resistant security



Fast computation

Small communication

NIST's priorities for Round 3 analysis

Cryptanalysis

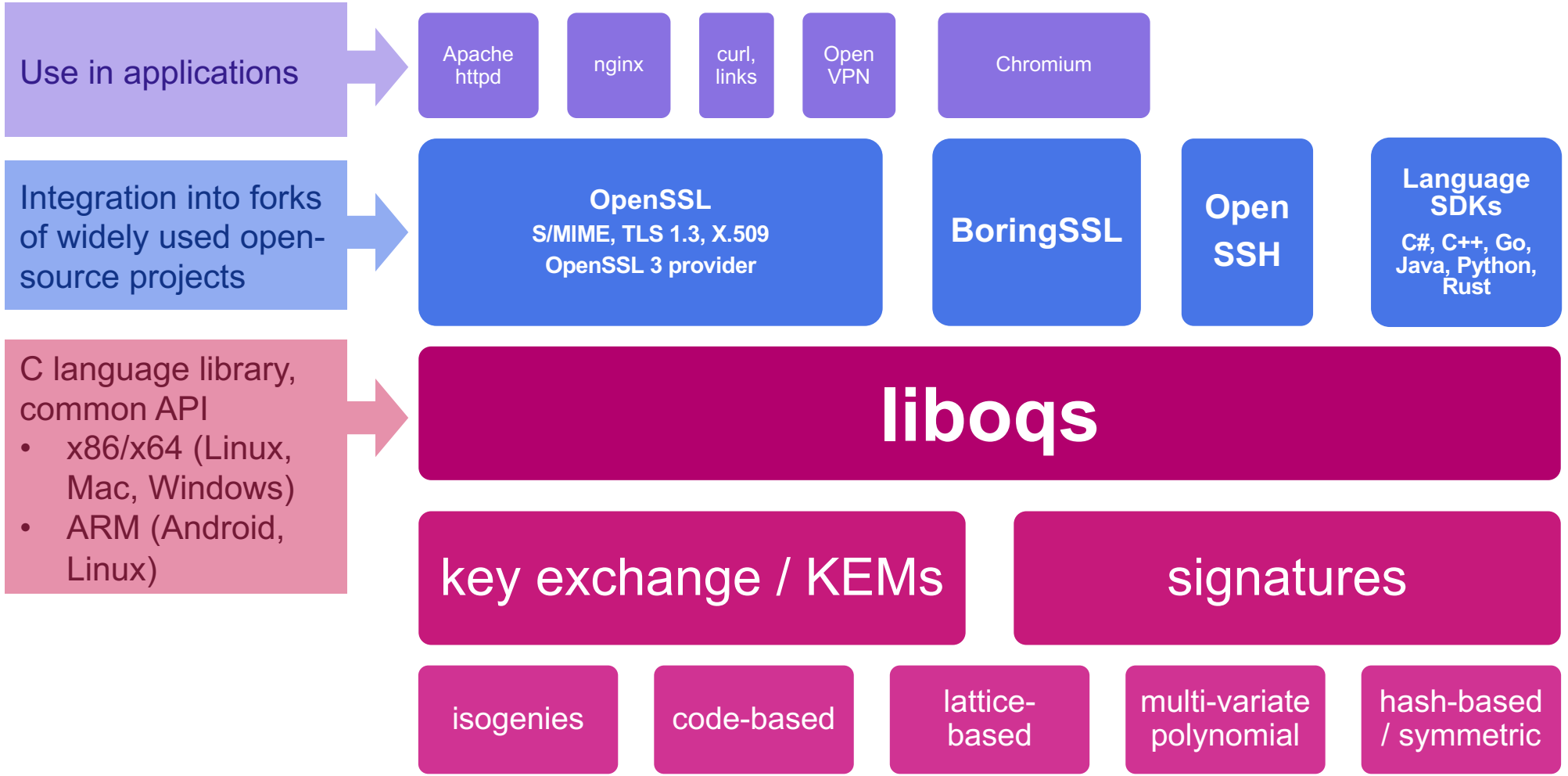
- Better understand CoreSVP hardness of lattice-based schemes
- Does choice of lattice structure matter?
- Decide between Kyber, NTRU, Saber
- Decide between Dilithium and Falcon

Implementations

- Side-channel resistant implementations
- Easy of implementation
- Performance data in Internet protocols
- Performance data for hardware implementations

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

Open Quantum Safe Project



Industry partners:
- Amazon Web Services
- evolutionQ
- IBM Research
- Microsoft Research

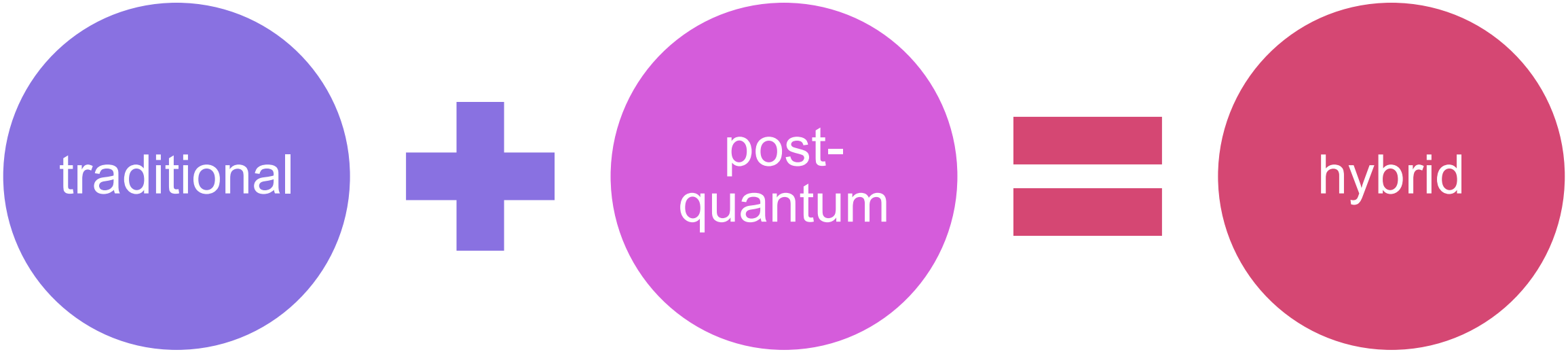
Additional contributors:
- Cisco
- Senetas
- PQClean project
- Individuals

Financial support:
- AWS
- Canadian Centre for Cyber Security
- NSERC
- Unitary Fund

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

Cautious "hybrid" approach

- Some proposed post-quantum solutions could be broken
- **Hybrid approach**: use traditional and post-quantum simultaneously to reduce risk during transition
- Focus on algorithms that advance through NIST process



Wrapping up

Post-quantum crypto at University of Waterloo

Main research areas:

- Design of post-quantum cryptosystems
- Cryptanalysis of post-quantum problems on classical or quantum computers
- Efficient implementations of post-quantum cryptography
- Adapting network protocols to post-quantum algorithms

Main mathematical problems:

- Isogeny-based
- Lattice-based (learning with errors, NTRU)

Involved in 4 (out of 16) NIST Round 3 candidates:

Finalists:

- **CRYSTALS-Kyber** (module learning with errors)
- **NTRU** (also lattice based)

Alternate candidates:

- **FrodoKEM** (learning with errors)
- **SIKE** (isogenies on elliptic curves)

More reading

NIST Round 3
https://nist.gov/pqcrypto

Quantum threat timeline
https://globalriskinstitute.org/publications/quantum-threat-timeline/

Background on post-quantum crypto

- Post-Quantum Cryptography, by Bernstein, Buchmann, Dahmen (2009)
  https://link.springer.com/book/10.1007/978-3-540-88702-7
- EU Overview Report (Feb 2021)
  https://www.enisa.europa.eu/publications/post-quantum-cryptography-current-state-and-quantum-mitigation

Lattice-based crypto

- Mathematics of Public Key Cryptography, by Steven Galbraith (2012)
  https://www.math.auckland.ac.nz/~sgal018/crypto-book/crypto-book.html
- A Decade of Lattice Cryptography, by Chris Peikert (2017)
  https://web.eecs.umich.edu/~cpeikert/pubs/lattice-survey.pdf
- On the concrete hardness of learning with errors, by Albrecht, Player, Scott (2015)
  https://eprint.iacr.org/2015/046