

# A brief introduction to lattice-based cryptography

**Douglas Stebila**

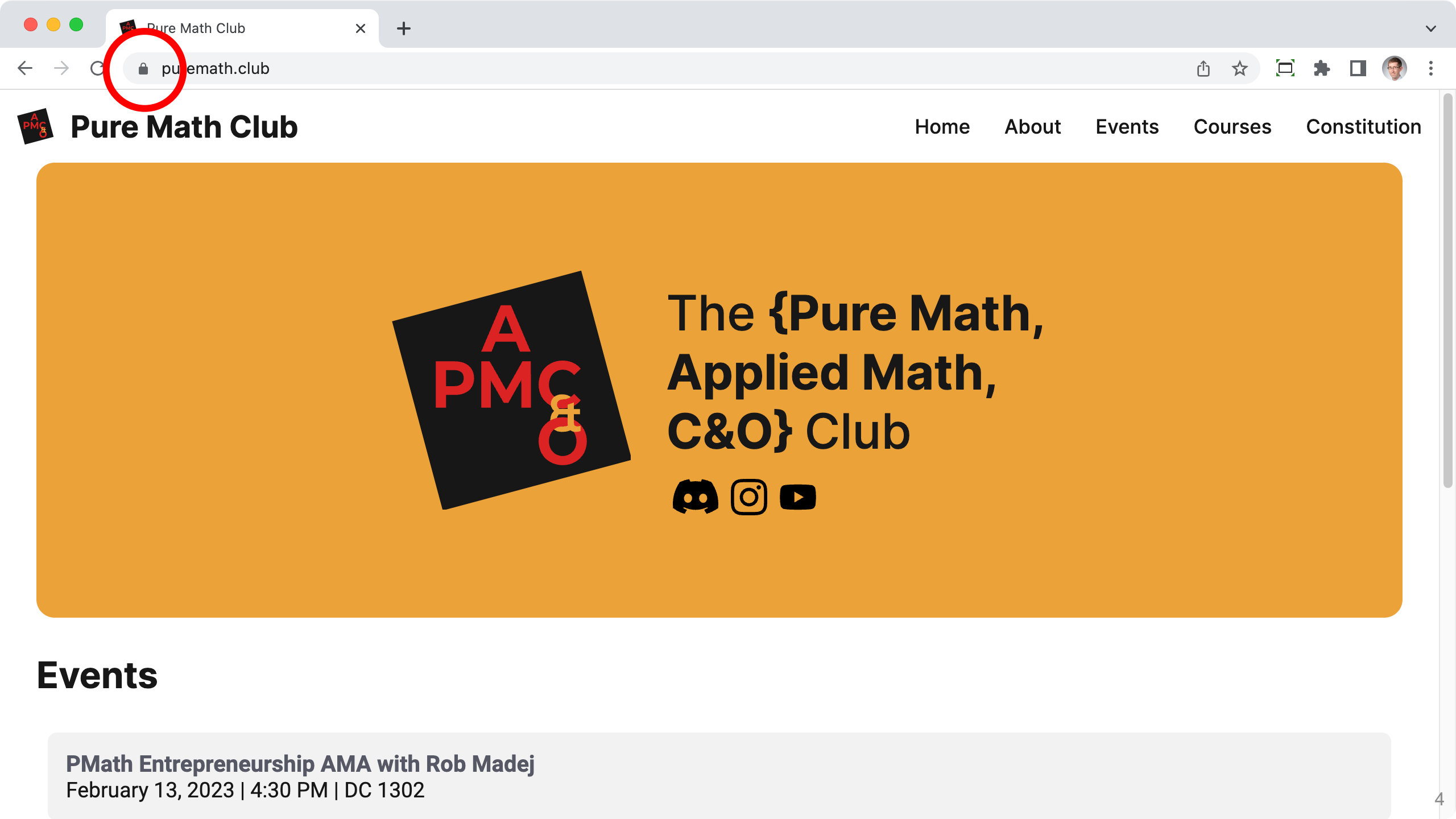


# Outline

- 1. Background: Why post-quantum?**
- 2. Learning with errors problems**
- 3. Public key encryption from LWE**
- 4. Difficulty of LWE and lattice problems**
- 5. Standardization of post-quantum cryptography**

# 1. Background

# Why post-quantum?



The {Pure Math,  
Applied Math,  
C&O} Club



## Events

**PMath Entrepreneurship AMA with Rob Madej**  
February 13, 2023 | 4:30 PM | DC 1302

# Pure Math Club



## The {Pure Math, Applied Math, C&O} Club



## Events

**PMath Entrepreneurship AMA with Rob Madej**  
February 13, 2023 | 4:30 PM | DC 1302

**Short Attention Span Math Seminars**  
February 10, 2023 | 3:00 PM | DC 1302

- Overview
- Main origin (secure)
  - https://puremath.club
- Secure origins
  - https://polyfill.io
- Unknown / canceled
  - https://fonts.googleapis.com
  - https://cdnjs.cloudflare.com
  - https://fonts.gstatic.com

### 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 1.3, X25519, and AES\_256\_GCM.
- Resources - all served securely**  
All resources on this page are served securely.

# Pure Math Club



## Events

**PMath Entrepreneurship AMA with Rob M**  
 February 13, 2023 | 4:30 PM | DC 1302

**Short Attention Span Math Seminars**  
 February 10, 2023 | 3:00 PM | DC 1302

<b>puremath.club</b>		R3	ISRG Root X1
<b>Subject Name</b>			
Common Name	puremath.club		
<b>Issuer Name</b>			
Country	US		
Organization	Let's Encrypt		
Common Name	R3		
<b>Validity</b>			
Not Before	Tue, 07 Feb 2023 23:46:41 GMT		
Not After	Mon, 08 May 2023 23:46:40 GMT		
<b>Subject Alt Names</b>			
DNS Name	puremath.club		
<b>Public Key Info</b>			
Algorithm	RSA		
Key Size	2048		
Exponent	65537		
Modulus	F6:83:C7:BF:B1:3F:E6:8D:21:95:C5:0E:2F:3C:24:38:CA:35:66:38:E9:CE:2C: 5F:E4:A1:79:4E:0F:81:FA:9D:AA:65:A5:D2:1D:2B:3E:7D:BA:A4:84:89:1B:8C:F 7:26:B8:D1:38:6B:3E:5B:F1:2A:DE:F5:A4:EF:EE:F6:59:50:6F:0E:F1:79:0B:44: 93:74:19:C2:AB:37:30:34:9F:F9:7C:FD:EB:4C:A3:D8:58:0B:3A:41:C1:55:6D: 5A:4D:7E:82:EE:67:39:C7:42:E8:60:4E:1E:60:73:10:16:B6:FA:EF:F8:2A:D3:0 6:20:EA:2B:70:1E:71:A5:3B:01:0C:43:8B:24:0C:83:BE:C1:33:46:DC:A3:29:C 1:88:98:5E:8E:FD:EE:DA:A2:CF:FB:A1:65:CC:AB:93:26:4B:36:A1:EC:E0:F9:E F:84:E3:FD:AF:33:FE:5F:90:95:51:D4:40:7A:29:EA:92:54:70:80:D0:DC:FD:5 9:02:4C:B6:79:BB:36:F5:B7:16:6E:92:52:BA:8D:4E:8B:AE:49:C7:32:8B:70:C 3:AF:E6:17:34:DA:C1:23:F3:6D:CB:3D:C8:FF:5A:90:F2:7D:5C:0C:1E:53:CF:1 A:42:93:F0:D8:6E:74:BF:E4:C9:B2:E4:00:9E:32:C3:C1:B4:15:E2:6C:D4:00:0 1:9B		

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 1.3, X25519, and AES\_256\_GCM.

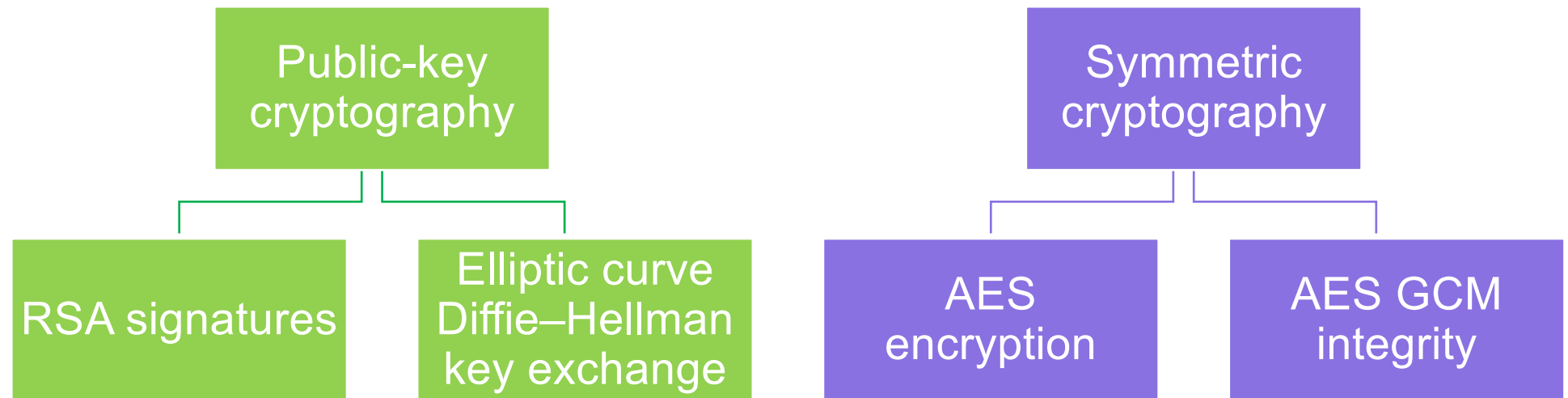
Resources - **all served securely**  
 All resources on this page are served securely.

## Certificate - valid and trusted

The connection to this site is using a valid, trusted server certificate issued by DigiCert TLS **RSA** SHA256 2020 CA1.

## Connection - secure connection settings

The connection to this site is encrypted and authenticated using TLS 1.3, **X25519** and **AES\_128\_GCM**.



# RSA digital signatures

## Key generation

- Pick large random primes  $p, q \approx 2^{1024}$
- Compute
$$n = pq,$$
$$\phi(n) = (p - 1)(q - 1)$$
- Pick  $e \in \mathbb{Z}_n^*$
- Compute
$$d = e^{-1} \bmod \phi(n)$$
- Public key:  $n, e$
- Secret key:  $n, d$

## Signing

To sign message  $m$  using secret key  $n, d$ :

- Signature:
$$\sigma = H(m)^d \bmod n$$

**Hard to forge signatures  
if factoring is hard\***

## Verification

Get a trusted copy of the signer's public key  $n, e$

To verify message  $m$  against signature  $\sigma$  and public key  $n, e$

- Check if
$$\sigma^e = H(m) \bmod n$$



# Diffie–Hellman key exchange

Public parameters:  $g$  is a generator of an abelian group of prime order  $q$

**Alice**

$$x \in_R \mathbb{Z}_q$$

$$X \leftarrow g^x$$

send  $X \rightarrow$

$\leftarrow$  send  $Y$

$$k \leftarrow Y^x = g^{xy}$$

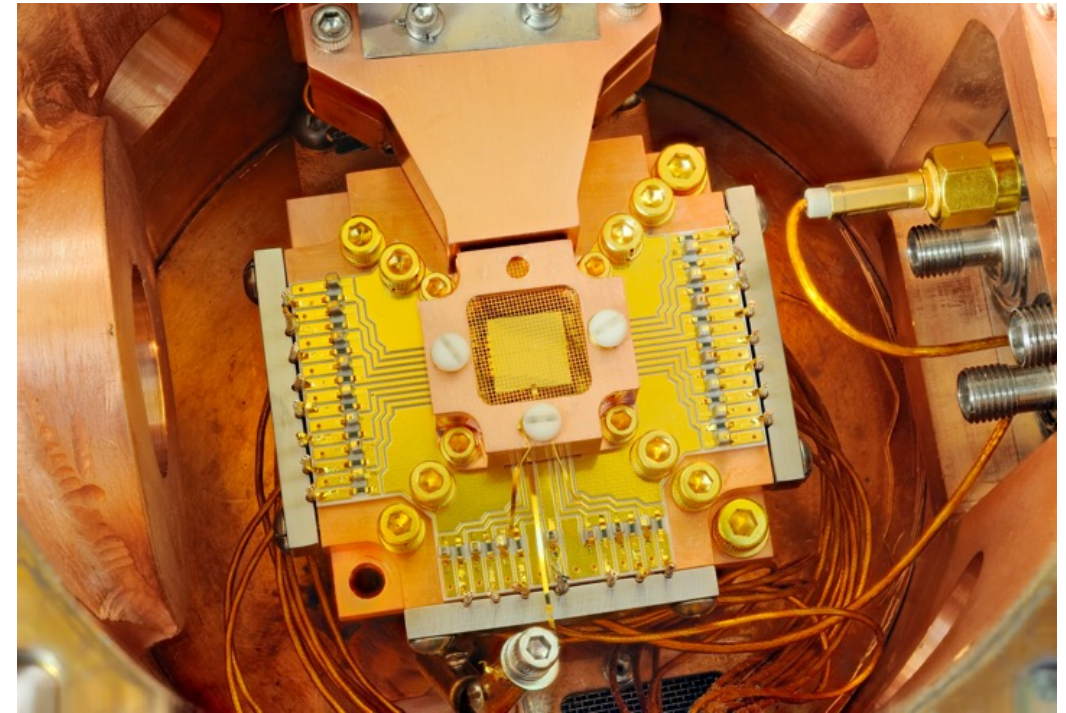
Hard to compute shared  
secret if discrete  
logarithms are hard\*

**Bob**

$$y \in_R \mathbb{Z}_q$$

$$Y \leftarrow g^y$$

$$k \leftarrow X^y = g^{xy}$$



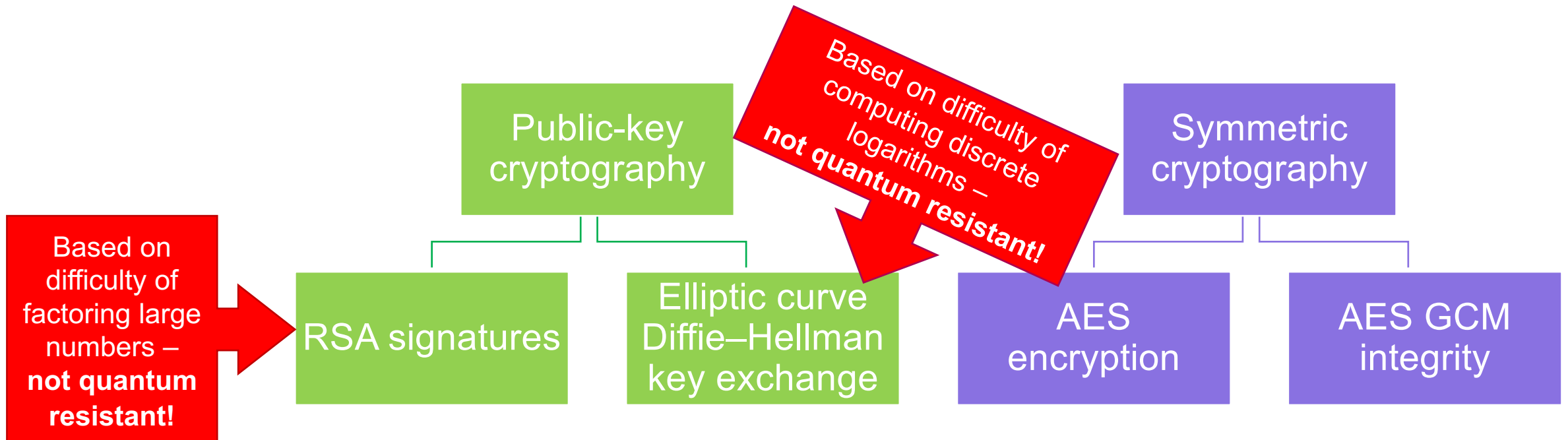
Theorem (Shor, 1984):  
There exists a polynomial-time quantum algorithm that can factor and compute discrete logarithms.

## Certificate - valid and trusted

The connection to this site is using a valid, trusted server certificate issued by DigiCert TLS **RSA** SHA256 2020 CA1.

## Connection - secure connection settings

The connection to this site is encrypted and authenticated using TLS 1.3, **X25519** and **AES\_128\_GCM**.



# Post-quantum cryptography

a.k.a. quantum-resistant algorithms

**Cryptography based on computational assumptions believed to be resistant to attacks by quantum computers**

Uses only classical (non-quantum) operations to implement

# 2. Learning with errors problems

# Solving systems of linear equations

$$\begin{matrix} \mathbb{Z}_{13}^{7 \times 4} \\ \begin{array}{|c|c|c|c|} \hline 4 & 1 & 11 & 10 \\ \hline 5 & 5 & 9 & 5 \\ \hline 3 & 9 & 0 & 10 \\ \hline 1 & 3 & 3 & 2 \\ \hline 12 & 7 & 3 & 4 \\ \hline 6 & 5 & 11 & 4 \\ \hline 3 & 3 & 5 & 0 \\ \hline \end{array} \end{matrix} \times \begin{matrix} \text{secret} \\ \mathbb{Z}_{13}^{4 \times 1} \\ \begin{array}{|c|} \hline \text{red} \\ \hline \text{red} \\ \hline \text{red} \\ \hline \text{red} \\ \hline \end{array} \end{matrix} = \begin{matrix} \mathbb{Z}_{13}^{7 \times 1} \\ \begin{array}{|c|} \hline 4 \\ \hline 8 \\ \hline 1 \\ \hline 10 \\ \hline 4 \\ \hline 12 \\ \hline 9 \\ \hline \end{array} \end{matrix}$$

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

# Solving systems of linear equations

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

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

$\times$

**secret**  
 $\mathbb{Z}_{13}^{4 \times 1}$

6
9
11
11

$=$

$\mathbb{Z}_{13}^{7 \times 1}$

4
8
1
10
4
12
9

Easily solved using  
Gaussian elimination  
(MATH 136)

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

# Learning with errors problem

[Regev 2005]

random

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

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

secret

$\mathbb{Z}_{13}^{4 \times 1}$

6
9
11
11

×

+

small noise

$\mathbb{Z}_{13}^{7 \times 1}$

0
-1
1
1
1
0
-1

=

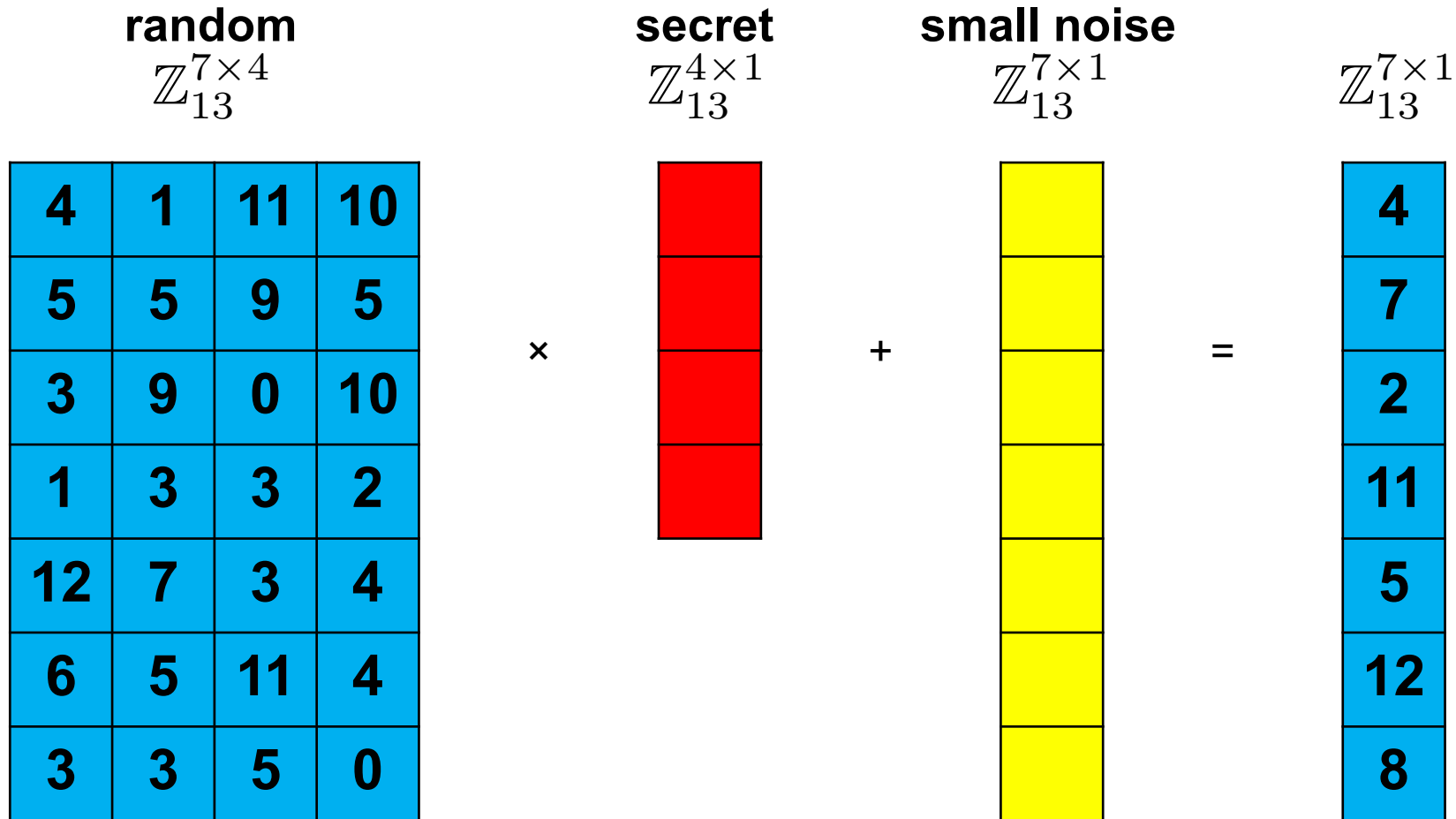
$\mathbb{Z}_{13}^{7 \times 1}$

4
7
2
11
5
12
8



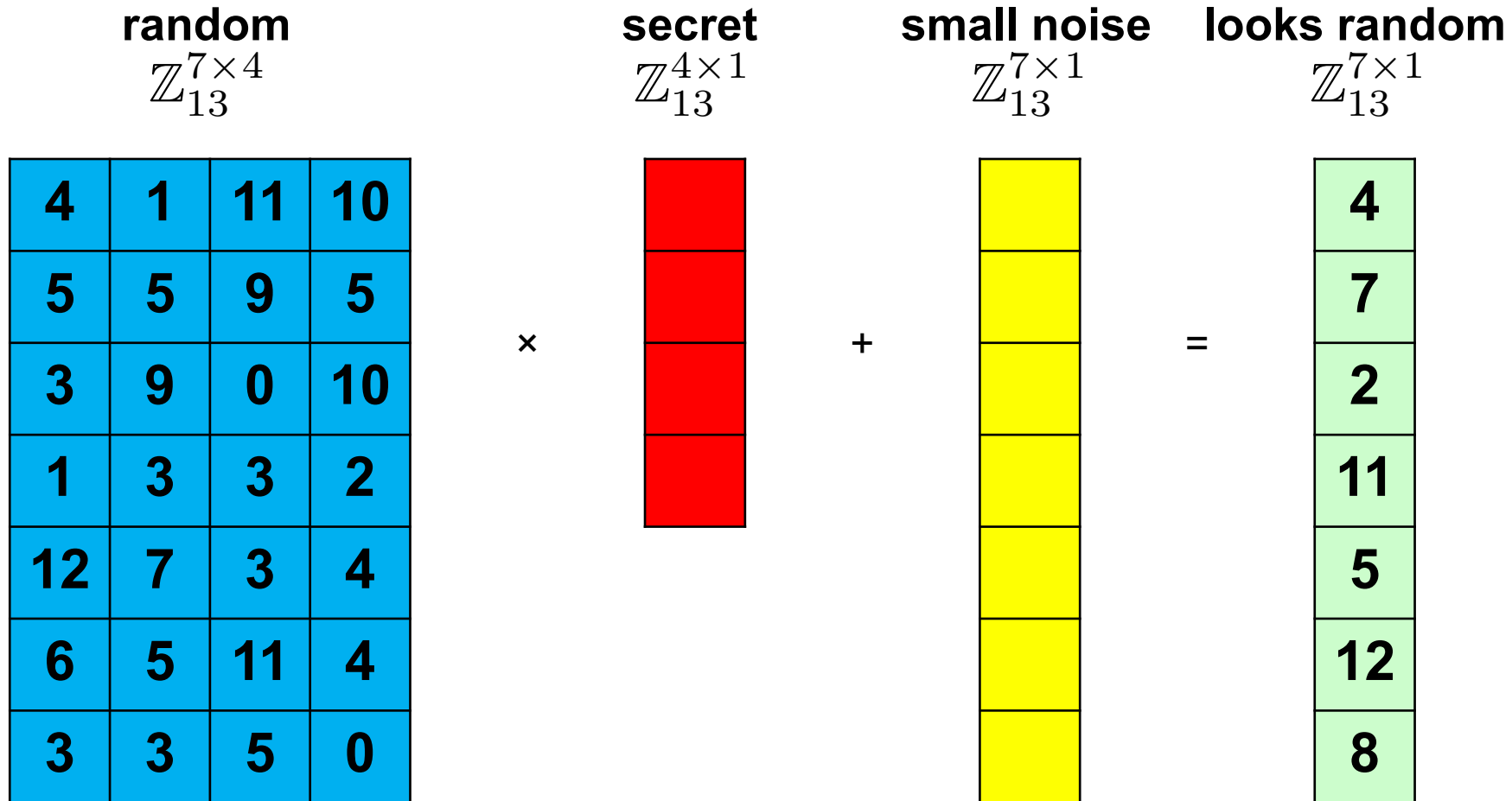
# Learning with errors problem

[Regev 2005]



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

# Decision learning with errors problem



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

# 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.

# 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

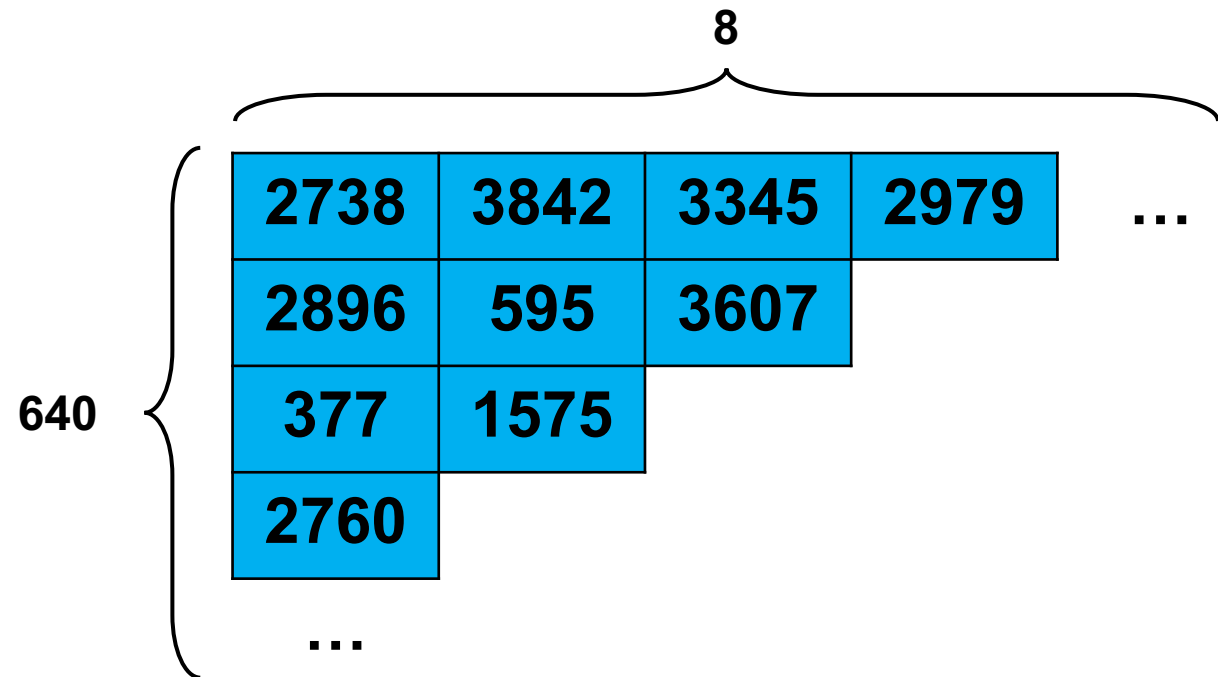
- The secret distribution  $\chi_s$  was originally taken to be the uniform distribution
- **Short secrets:** use  $\chi_s = \chi_e$
- There's a tight reduction showing that LWE with short secrets is hard if LWE with uniform secrets is hard.

# Toy example versus real-world example

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

4	1	11	10
5	5	9	5
3	9	0	10
1	3	3	2
12	7	3	4
6	5	11	4
3	3	5	0

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



$$640 \times 8 \times 15 \text{ bits} = 9.4 \text{ KiB}$$

# 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  $-x \bmod 13$ .



# 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  $-x \pmod{13}$ .

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

$$\times \quad 6 + 9x + 11x^2 + 11x^3$$

secret

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

small noise

---

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

# 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

×

$$\text{secret}$$

secret

+

$$\text{small noise}$$

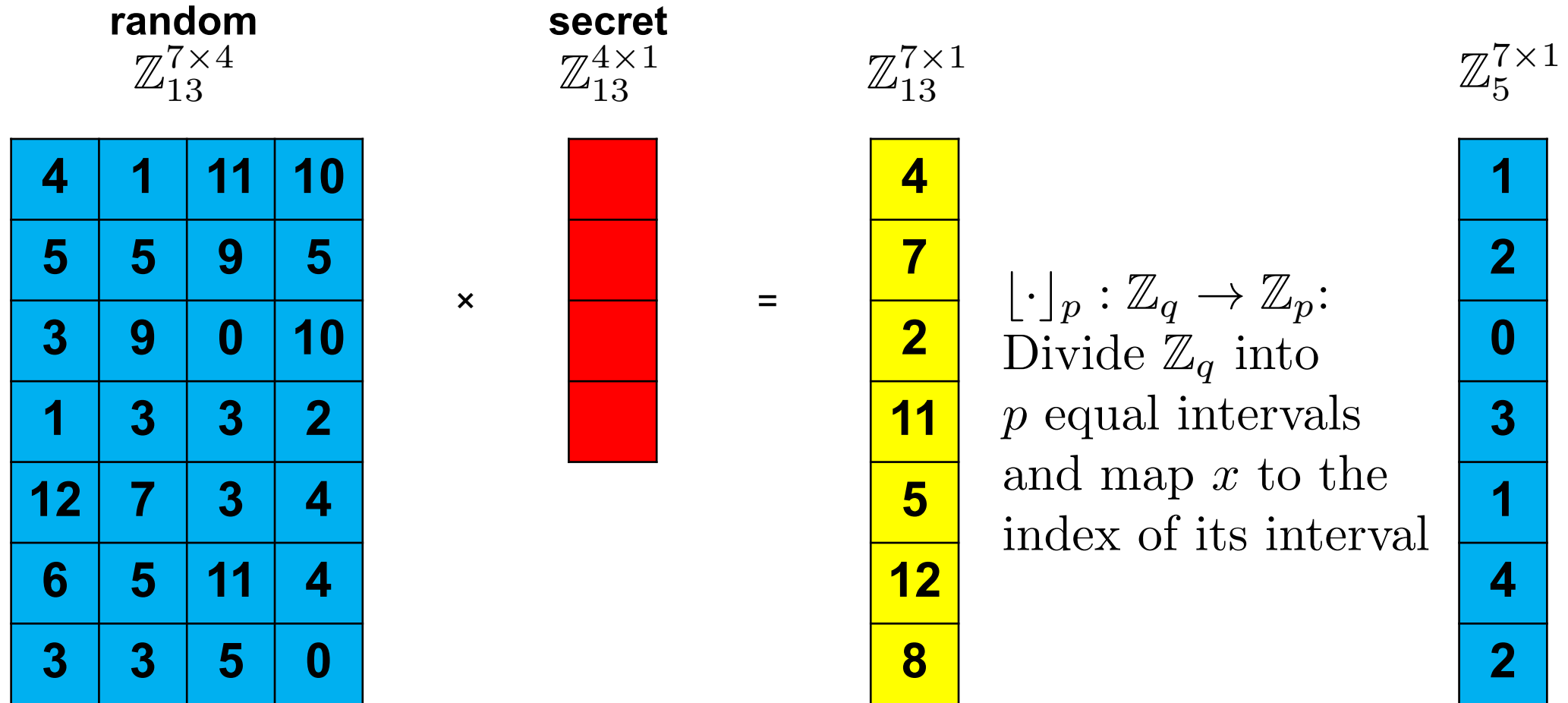
small noise

=

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

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

# Learning with rounding problem



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

# Problems

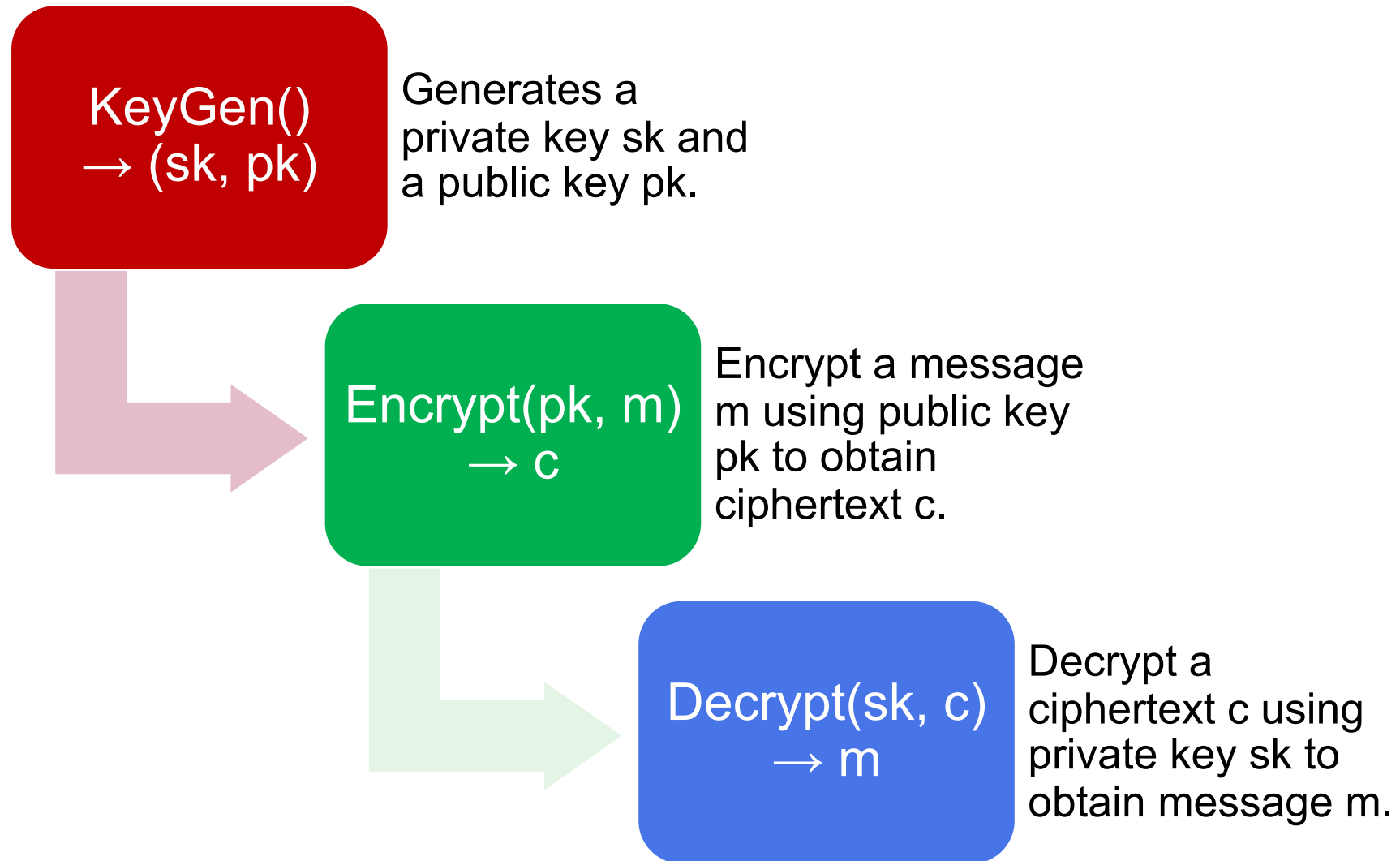
<b>Learning with errors</b>		
<b>Module-LWE</b>	<b>Search</b>	<b>With uniform secrets</b>
<b>Ring-LWE</b>		
<b>Learning with rounding</b>	<b>Decision</b>	<b>With short secrets</b>
<b>NTRU problem</b>		

# **3. Public key encryption from learning with errors**

# Public Key Encryption: Overview

- Alice creates a private key / public key pair
- Anyone can encrypt messages for Alice based on her public key, but only Alice can decrypt those messages
- Goal: Provide confidentiality

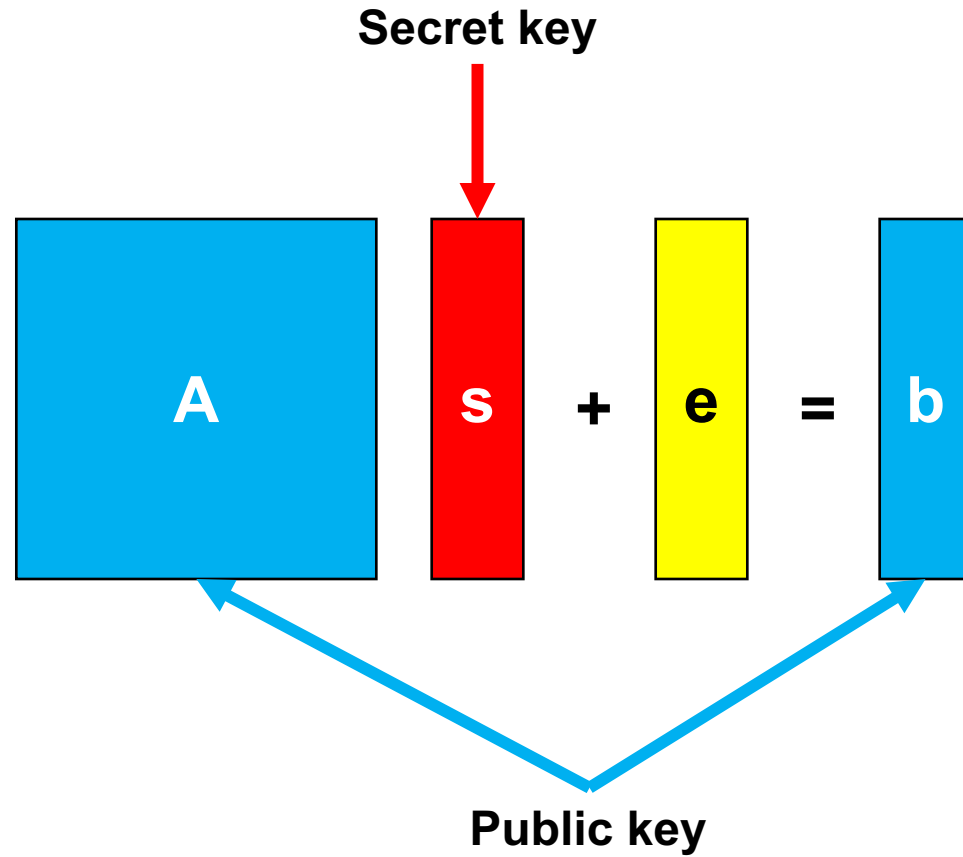
# Public Key Encryption: Algorithms





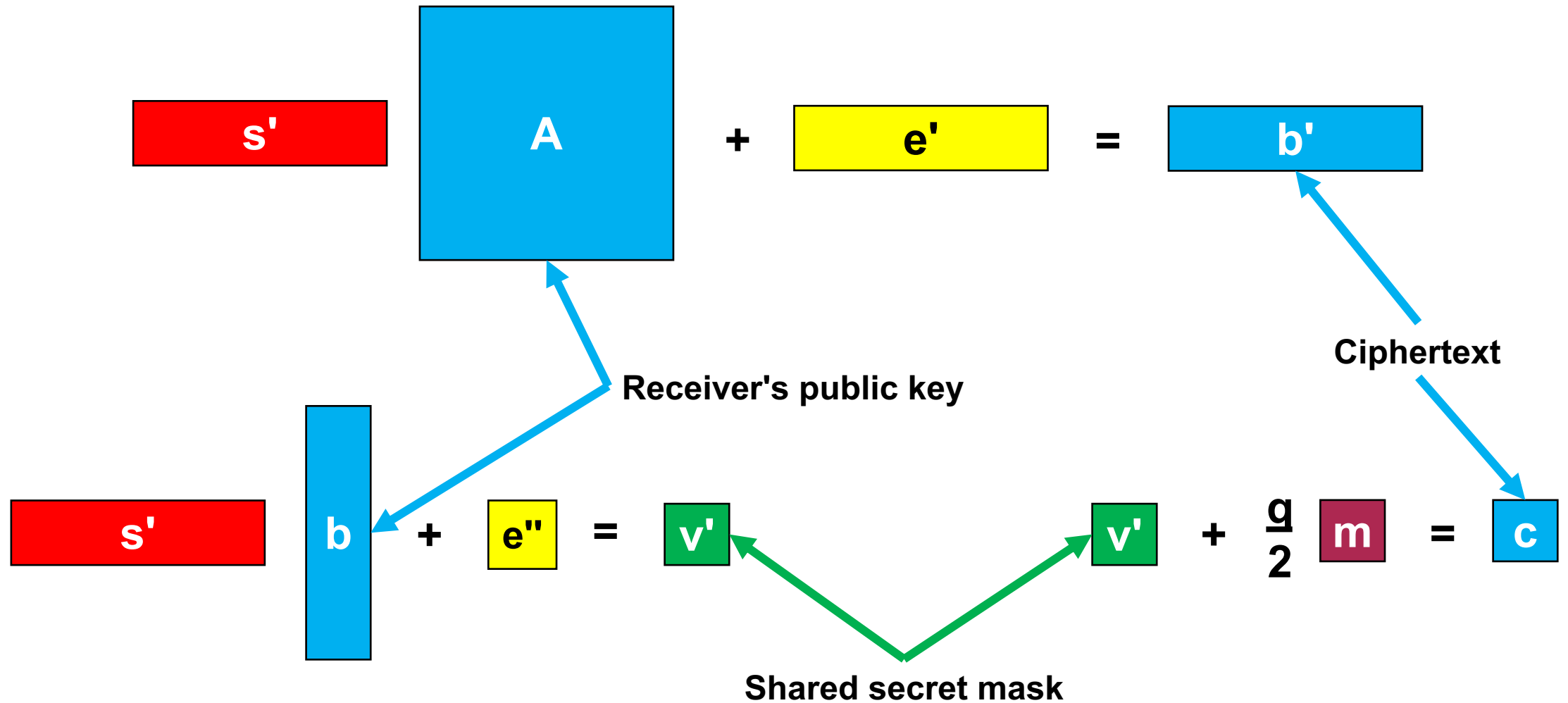
# Public key encryption from LWE

## Key generation



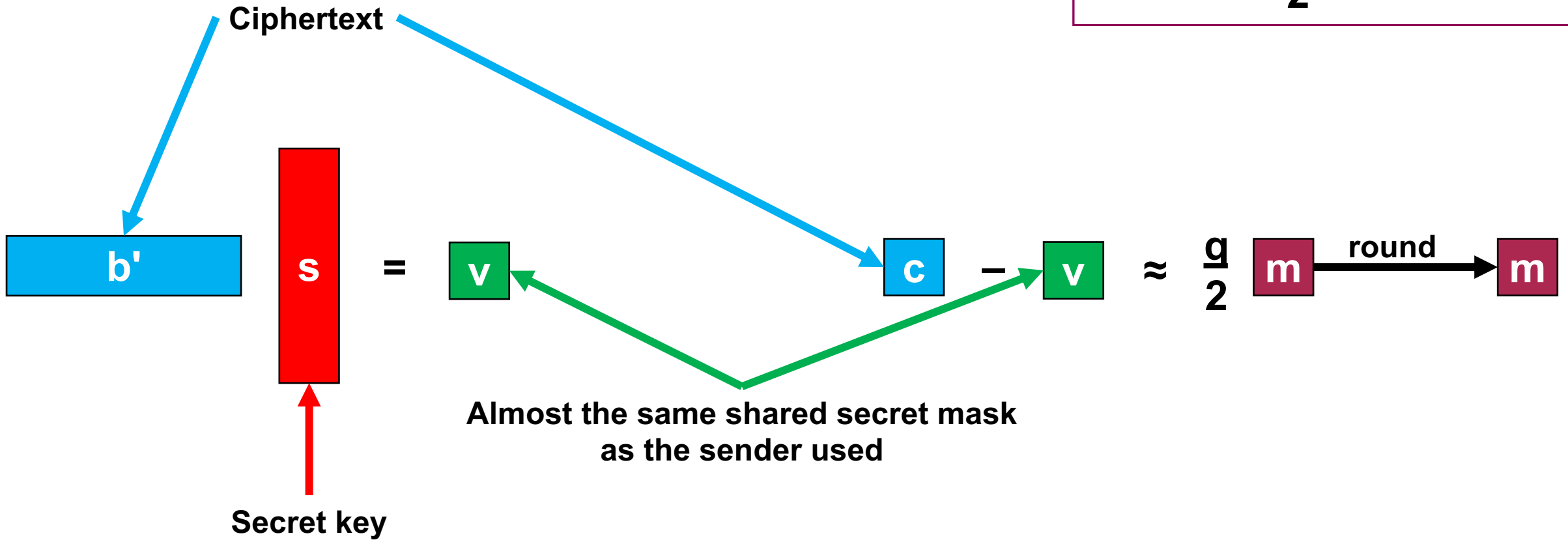
# Public key encryption from LWE

## Encryption



# Public key encryption from LWE Decryption

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



# Approximately equal shared secret

The sender uses

The receiver uses

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

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

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

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

$$\approx s' A s$$

$$\approx s' A s$$

=> Can decrypt as long as noise terms are small with high probability

# Security of public key encryption

## Theorem:

If the decision learning with errors problem is hard, then this public key encryption scheme is semantically secure against chosen plaintext attacks.

- Is the decision learning with errors problem hard?

## 4. Difficulty of LWE

# Lattice problems

# 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**

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

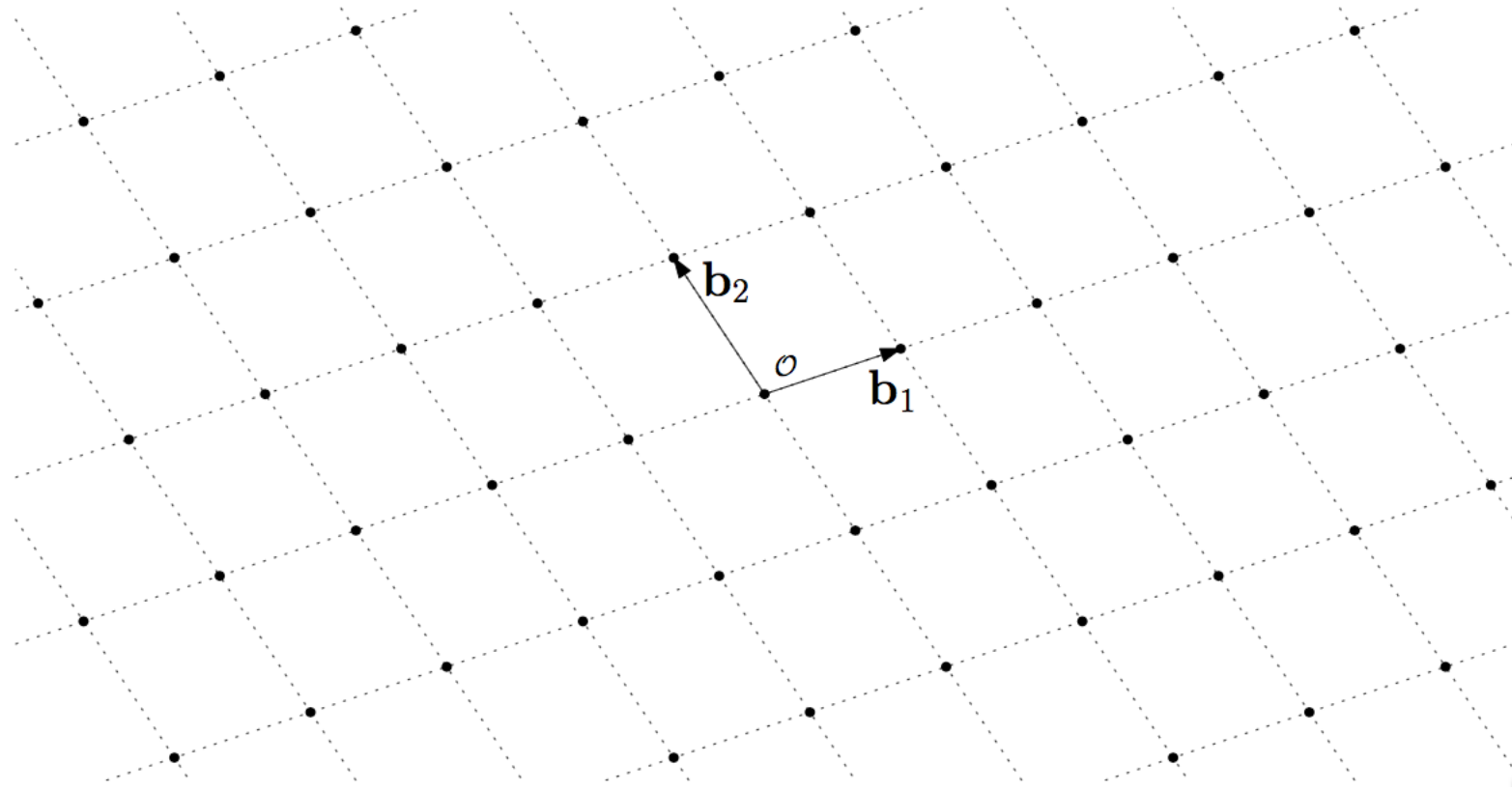
(In other words, a lattice is a set of *integer* linear combinations.)

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

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



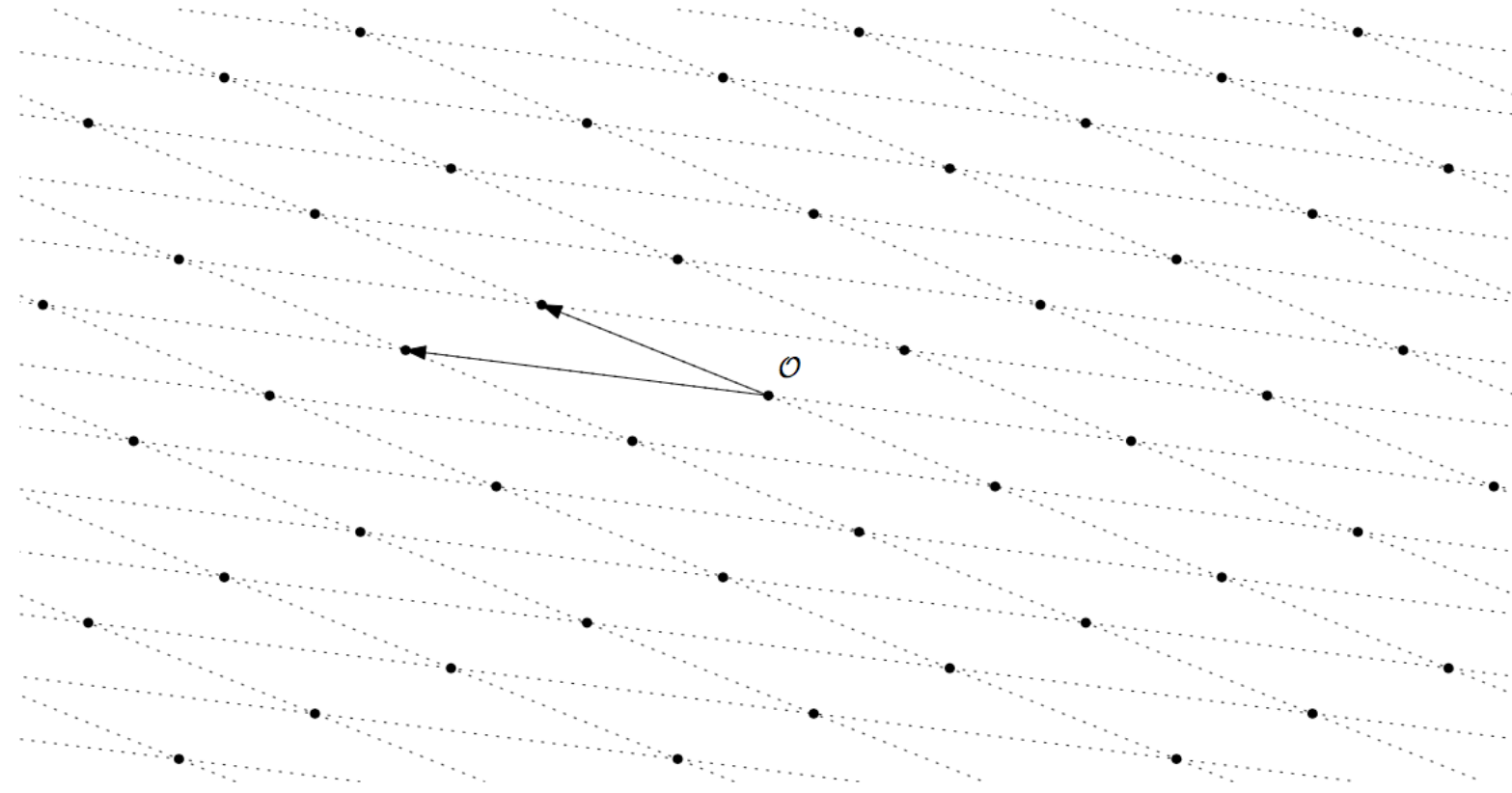
# Lattices



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

Equivalently, integer linear combinations of a basis

# Lattices

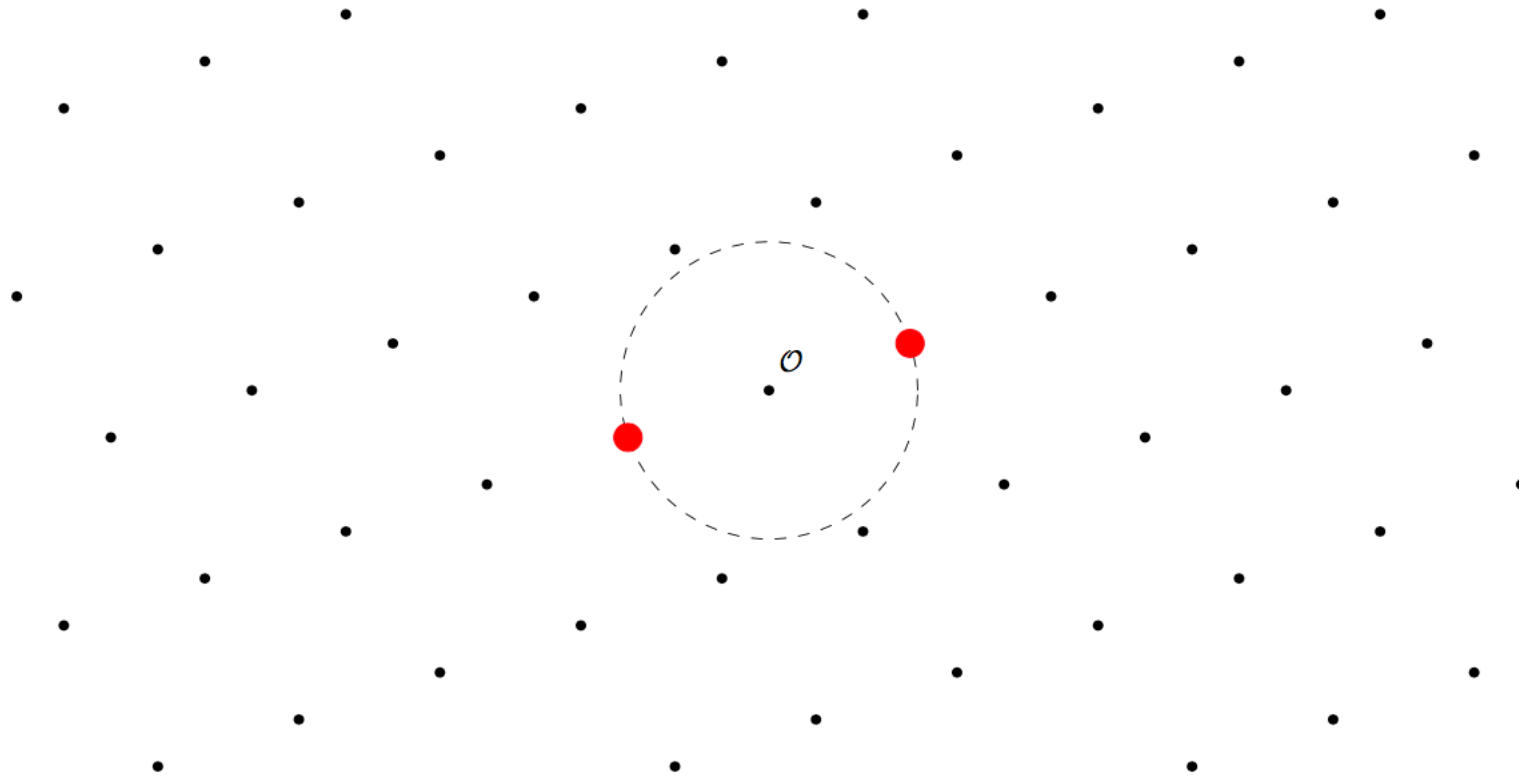


There are many bases for the same lattice – some short and orthogonalish, some long and acute.

# 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  $\pm 1$ .

# Shortest vector problem

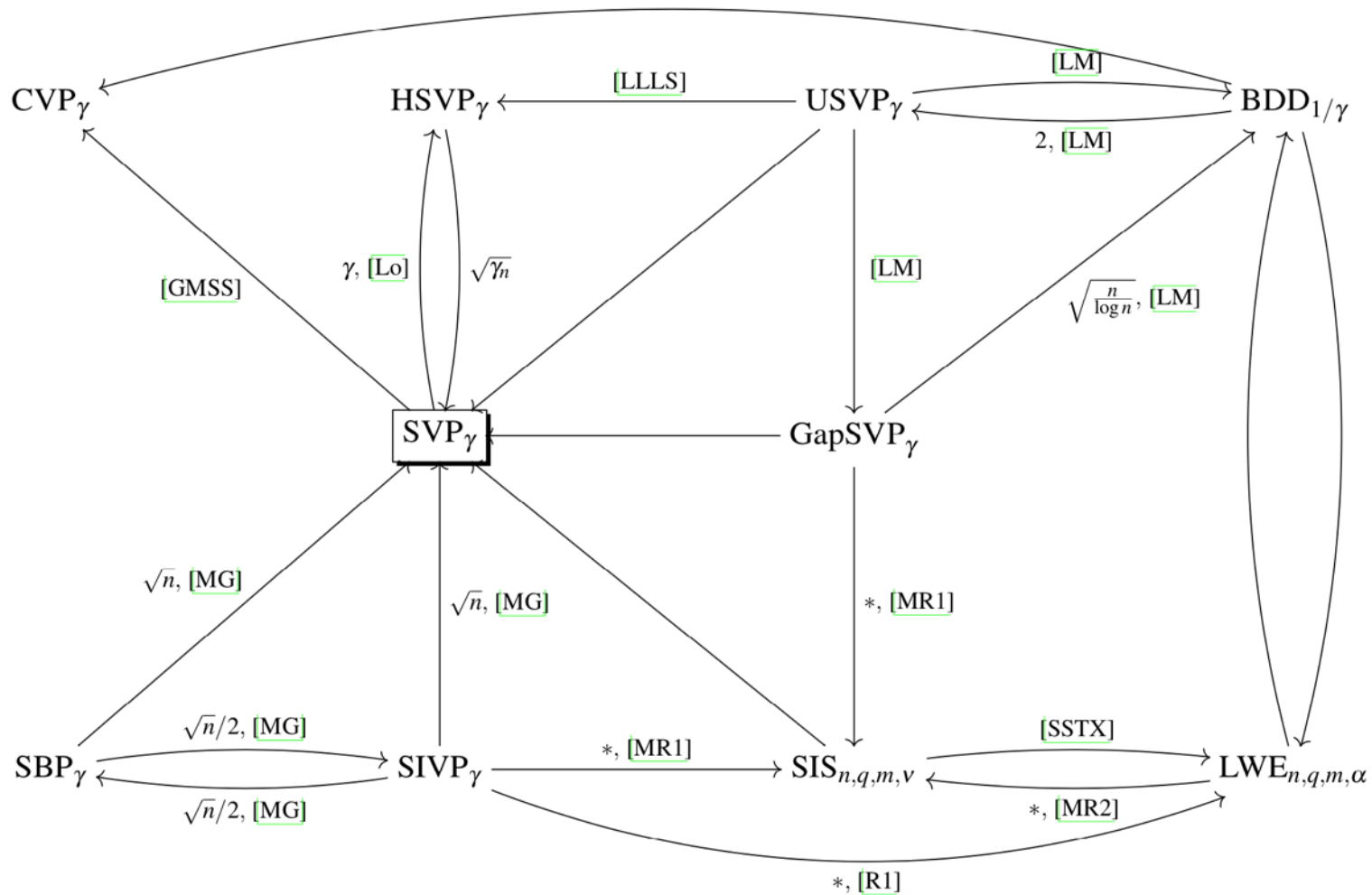


Given some basis for the lattice, find the shortest non-zero lattice point.

# 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 ( $\text{SVP}_\gamma$ ):** 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 ( $\text{GapSVP}_\gamma$ ):** 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 ( $\text{SIVP}_\gamma$ ):** Fix  $\gamma > 1$ . Given a basis  $B$  for a lattice  $\mathcal{L}$ , find a linearly independent set  $\{\vec{v}_1, \dots, \vec{v}_n\}$  such that  $\max_i \|\vec{v}_i\| \leq \gamma \cdot \lambda_n(\mathcal{L})$ .

# Relations among lattice problems



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

# Regev's reduction: LWE to shortest vector

**Theorem.** [Reg05] For any modulus  $q \leq 2^{\text{poly}(n)}$  and any discretized Gaussian error distribution  $\chi$  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  $\text{GapSVP}_\gamma$  and  $\text{SIVP}_\gamma$  on arbitrary  $n$ -dimensional lattices for some  $\gamma = \tilde{O}(n/\alpha)$ .

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

# 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

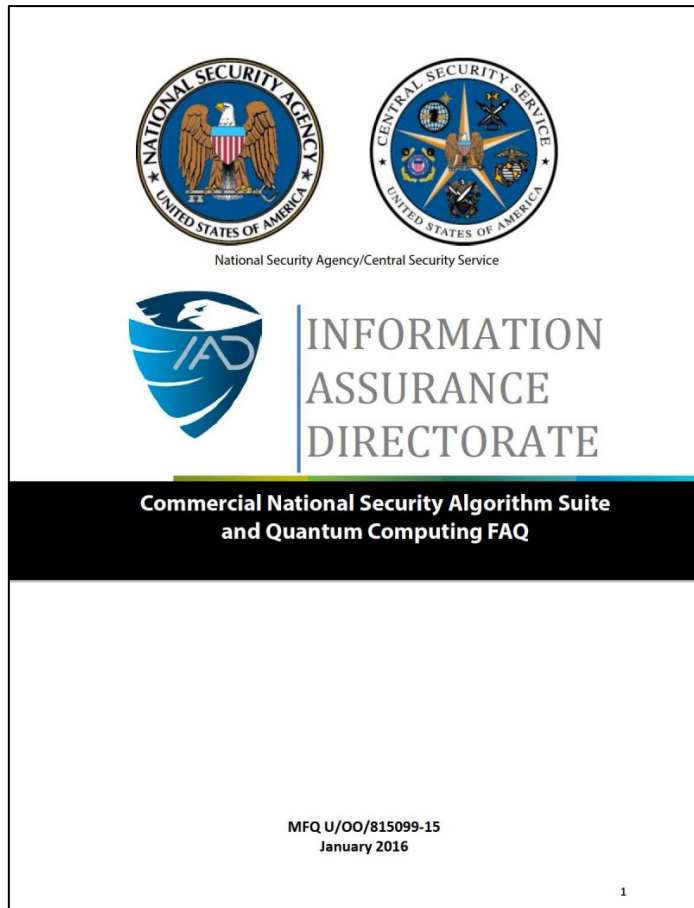
The complexity of  $\text{GapSVP}_\gamma$  depends heavily on how  $\gamma$  and  $n$  relate, and get harder for smaller  $\gamma$ .

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

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

# 5. Standardization of PQ cryptography

# Standardizing post-quantum cryptography



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

– NSA Information Assurance Directorate,  
Aug. 2015



Aug. 2015 (Jan. 2016)

# Primary goals for post-quantum crypto

**Confidentiality** in the public key setting

- **Public key encryption schemes**


- Alternatively: key encapsulation mechanisms
  - KEMs are a generalization of two-party Diffie–Hellman-style key exchange
  - Easy to convert KEM into PKE and vice versa

**Authentication & integrity** in the public key setting

- **Digital signature schemes**

# 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
- Substantial break of Rainbow algorithm in Round 3

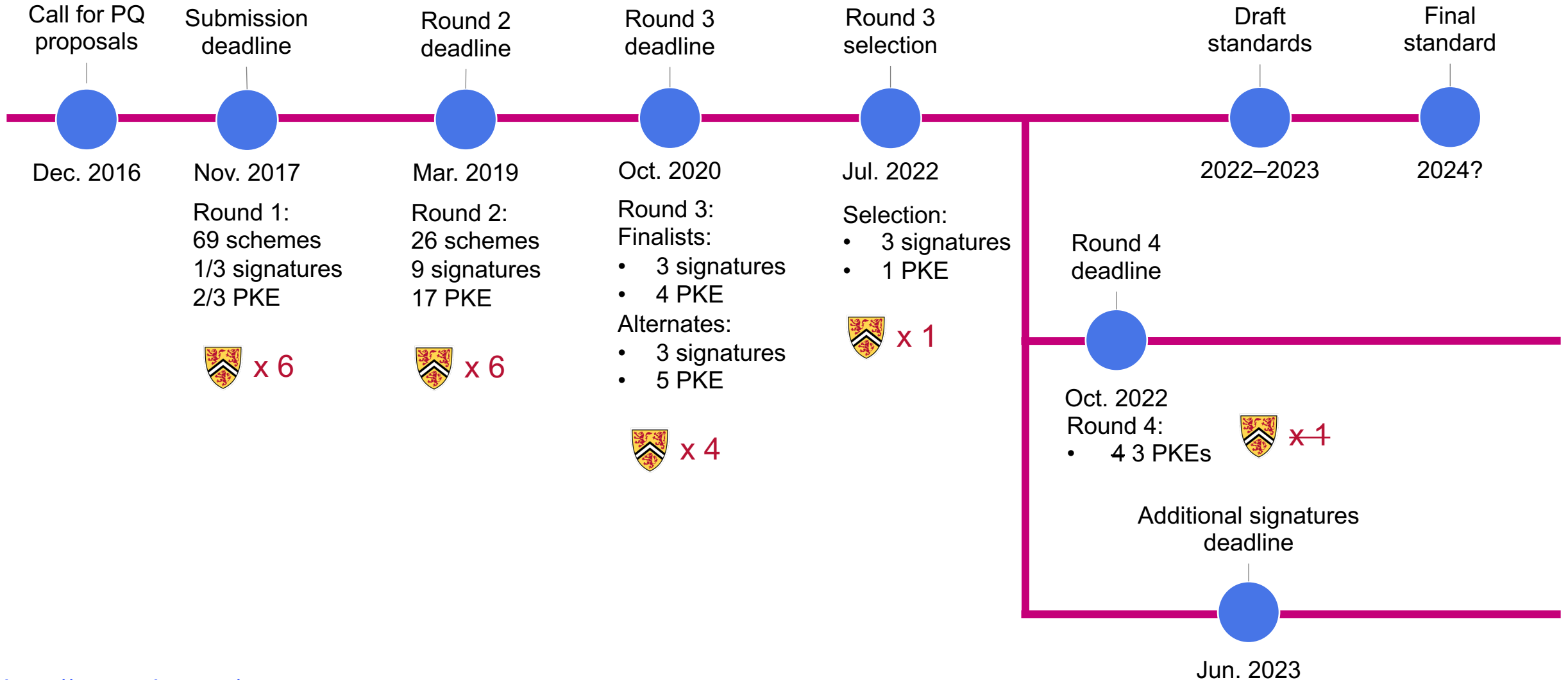
## Lattice-based

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

## Elliptic curve isogenies

- Newest mathematical construction
- Small communication, slower computation
- Substantial break of SIKE in Round 4

# NIST Post-quantum Crypto Project timeline



# NIST Round 3 selections and Round 4

## Selections

### Key encapsulation mechanisms

- Lattice-based: **Kyber** 

### Signatures

- Lattice-based: **Dilithium**, **Falcon**
- Hash-based: **SPHINCS+**

## Round 4

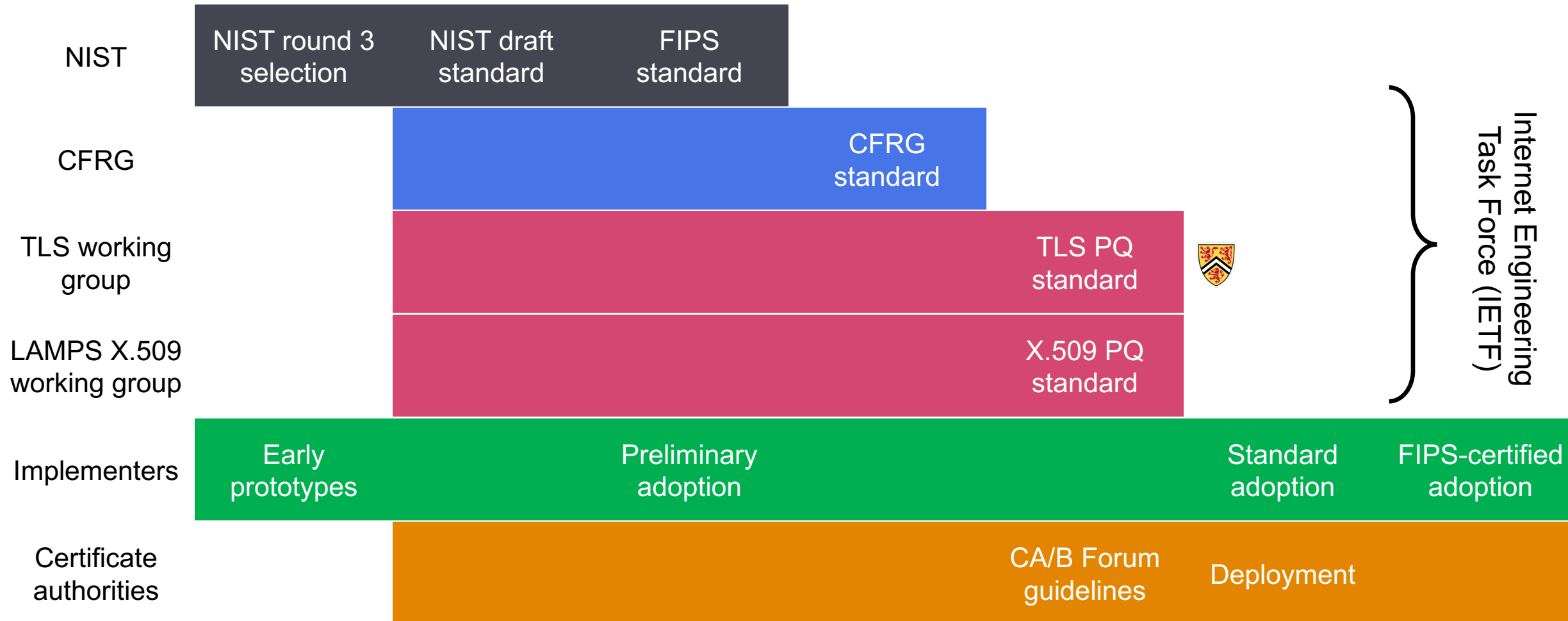
### Key encapsulation mechanisms

- Code-based: BIKE, Classic McEliece, HQC
- ~~Isogeny-based: SIKE~~ 

### Signatures

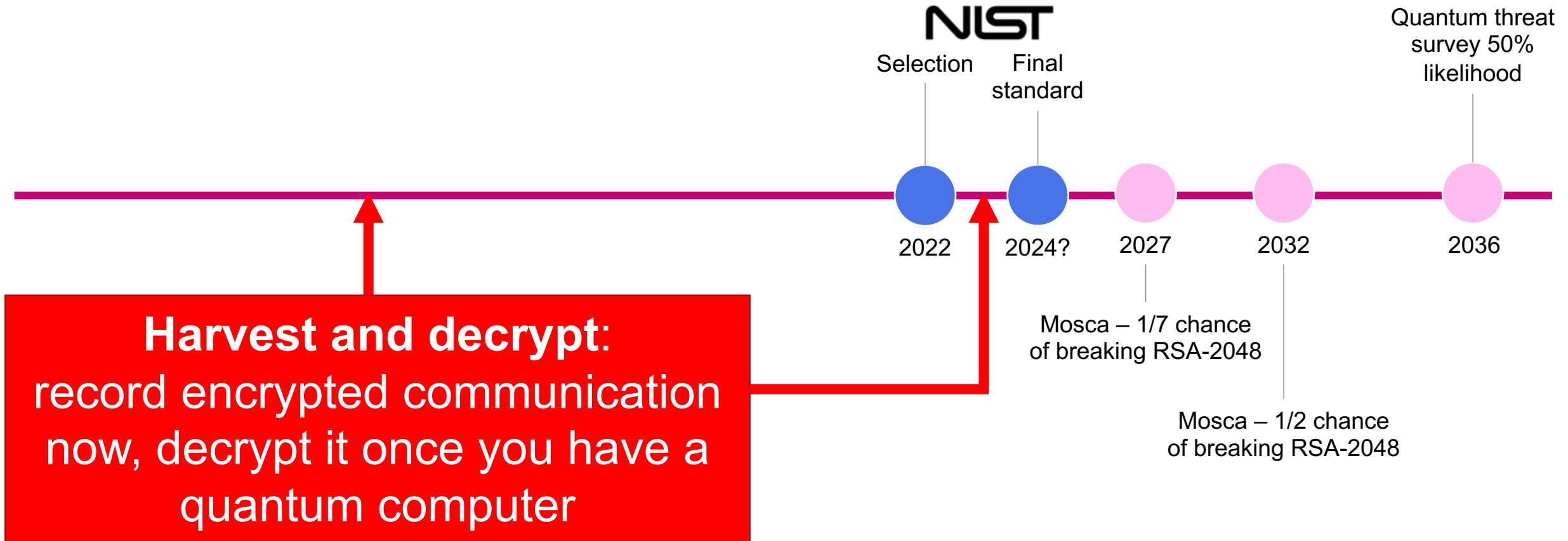
- Call for additional signature schemes

# Paths to standardization and adoption





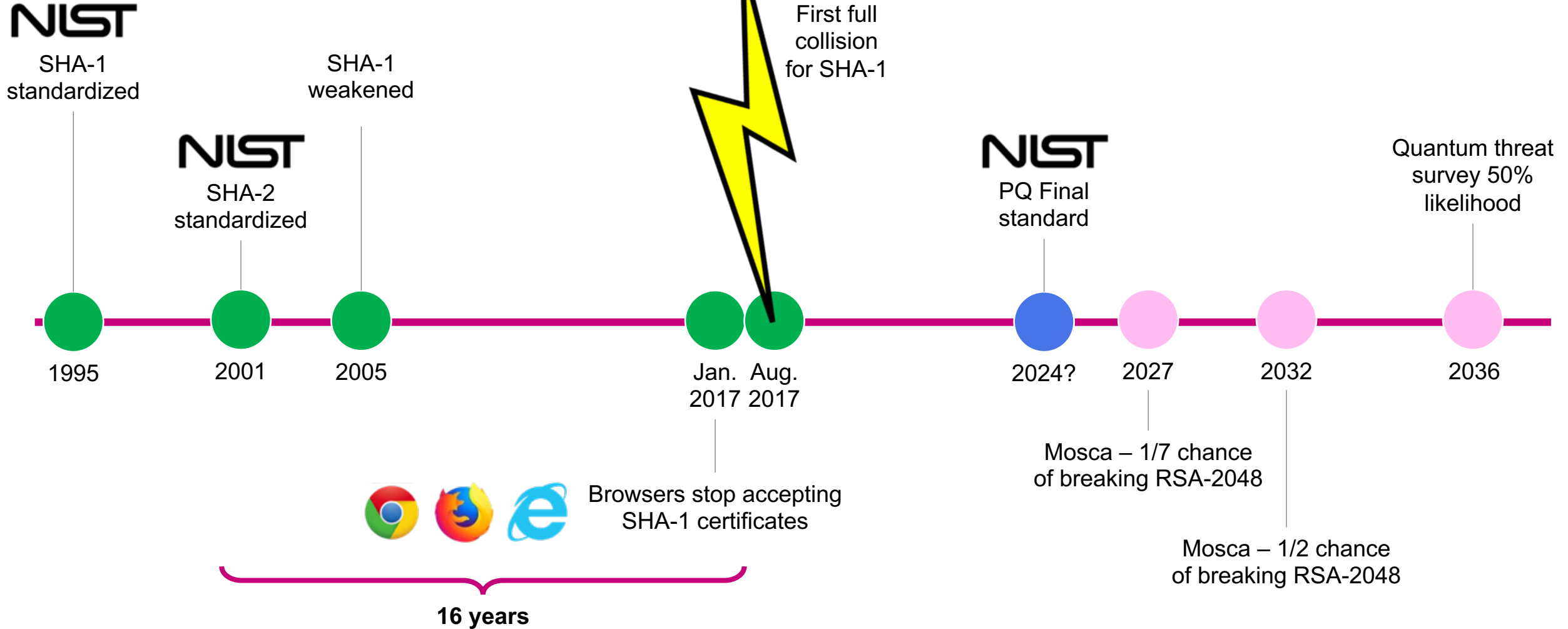
# Will we be ready in time?



[Mosca] IEEE Security & Privacy 16(5):38–41, Sep/Oct 2018. <https://doi.org/10.1109/MSP.2018.3761723>

[Quantum threat] <https://evolutionq.com/quantum-threat-timeline-2021.html>

# Timeline to replace cryptographic algorithms



# Trade-offs with post-quantum crypto

Confidence in quantum-resistance



Pick ~2

Fast computation

Small communication

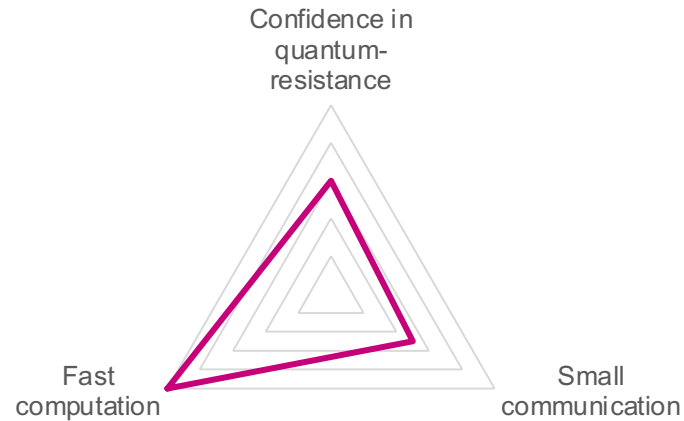
# Trade-offs with post-quantum crypto

## RSA and elliptic curves



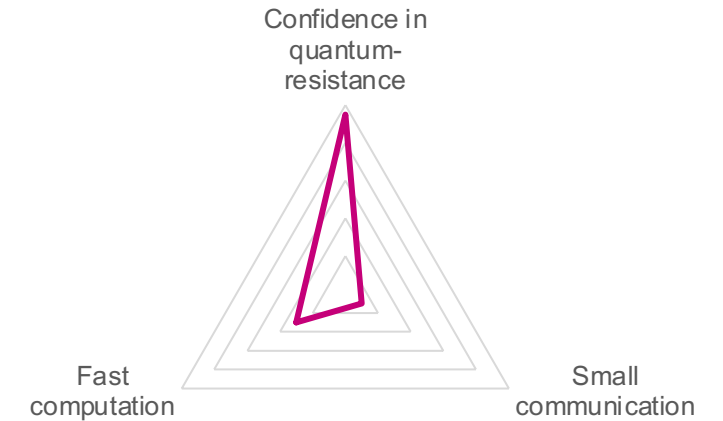
TLS handshake:  
1.3 KB

## Lattice-based cryptography



TLS handshake:  
11.2 KB

## Hash-based signatures



TLS handshake:  
24.6 KB

# Addressing the challenges of using PQ crypto

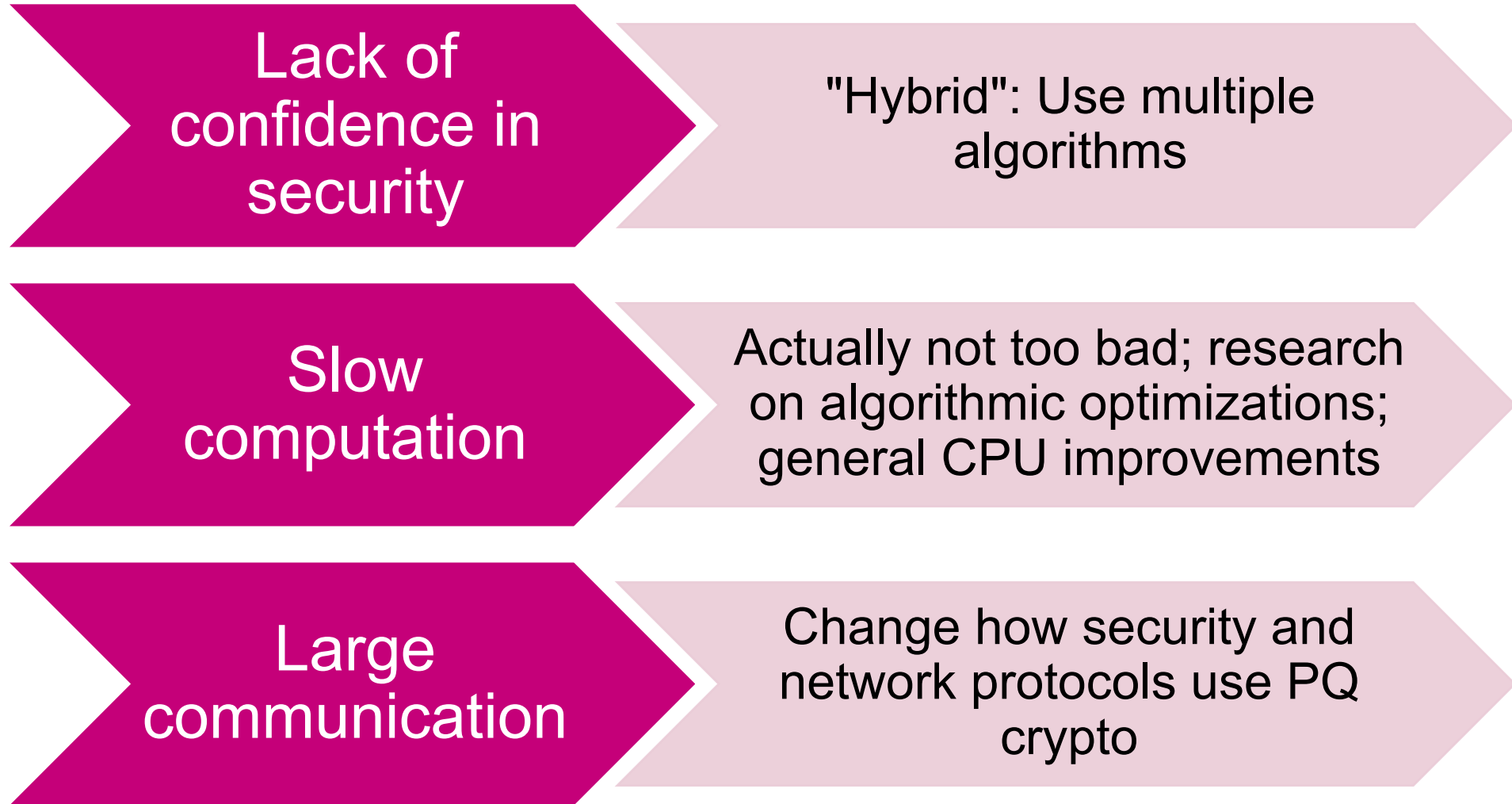
Lack of  
confidence in  
security

Slow  
computation

Large  
communication

Make better PQ crypto

# Addressing the challenges of using PQ crypto



**Hybrid approach:** use traditional and post-quantum simultaneously such that successful attack needs to break both



# 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 several NIST candidates:

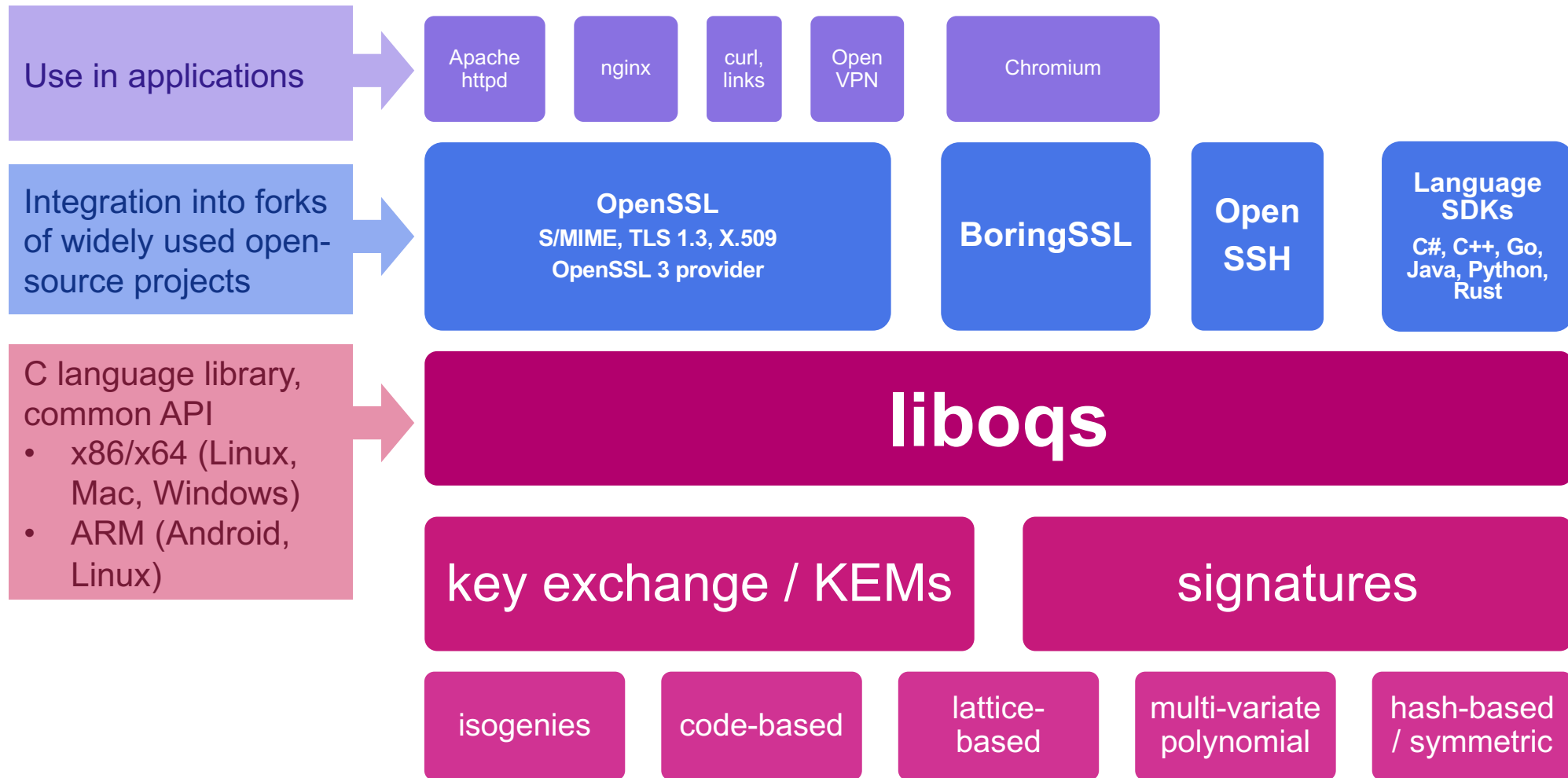
- Winner:
  - **CRYSTALS-Kyber** (module learning with errors)
- Round 3 alternates:
  - **FrodoKEM** (learning with errors)
  - **NTRU** (also lattice based)
  - **SIKE** (isogenies on elliptic curves)

Lead the Open Quantum Safe open-source software project

# OPEN QUANTUM SAFE

*software for prototyping  
quantum-resistant cryptography*

# Open Quantum Safe Project



Led by University of Waterloo

Industry partners:

- Amazon Web Services
- Cisco
- evolutionQ
- IBM Research
- Microsoft Research

Additional contributors:

- Senetas
- PQCclean project
- Individuals

Financial support:

- AWS
- Canadian Centre for Cyber Security
- Cisco
- NLNet
- NSERC
- Unitary Fund
- Verisign

# Where to learn more

## NIST Post-Quantum Crypto Standardiation

<https://nist.gov/pqcrypto>

## Quantum threat timeline

<https://globalriskinstitute.org/publications/quantum-threat-timeline/>

## Open Quantum Safe project

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

## 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>

## CO 485 Mathematics of Public Key Cryptography

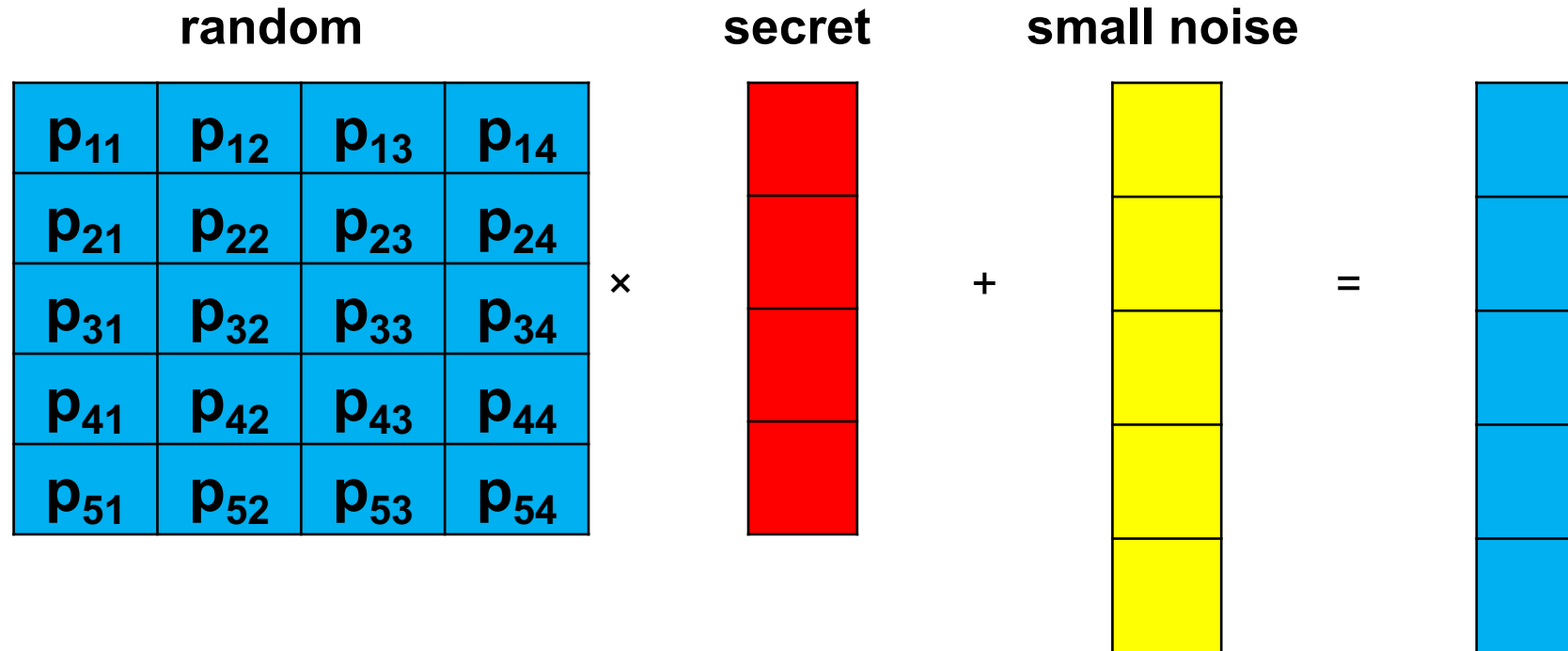
- Includes lattice-based cryptography and isogeny-based cryptography

## CO 487 Applied Cryptography

- Includes lattice-based cryptography and cryptographic protocols

# Appendix

# Module learning with errors problem



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

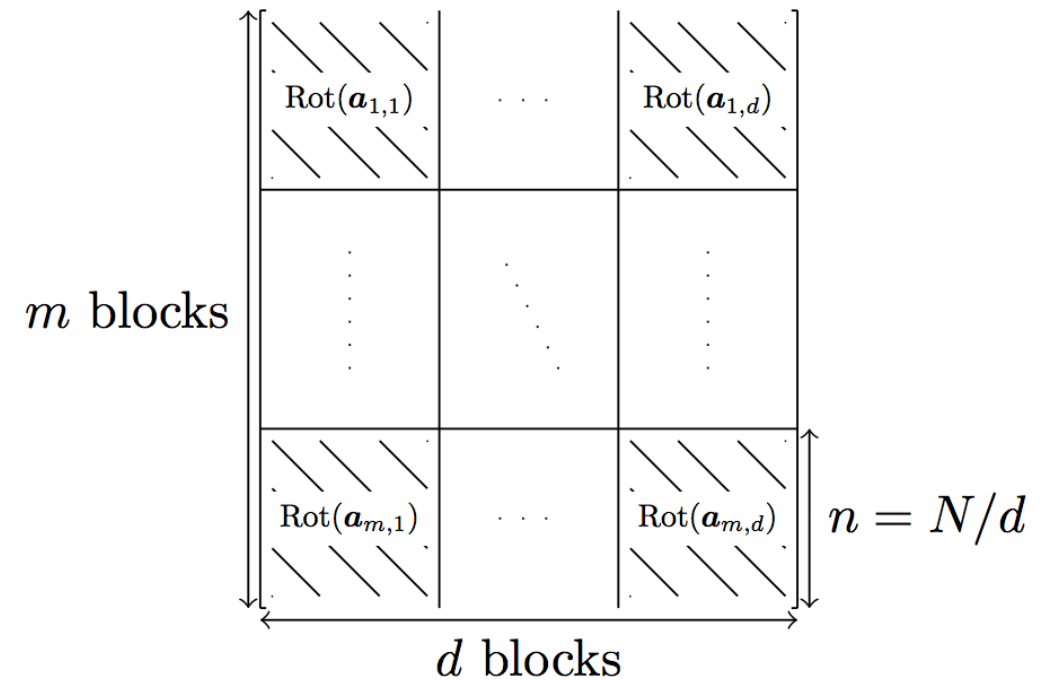
**Search Module-LWE problem:** given **blue**, find **red**

# 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



# Learning with Rounding

## Learning with Errors

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

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

## Learning with Rounding

- Noise comes from rounding to a smaller interval

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

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



# NTRU problem

For an invertible  $s \in R_q^*$  and a distribution  $\chi$  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.

# NTRU

## Learning with Errors

- Noisy product

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

## NTRU

- Noisy quotient

$$e/s$$

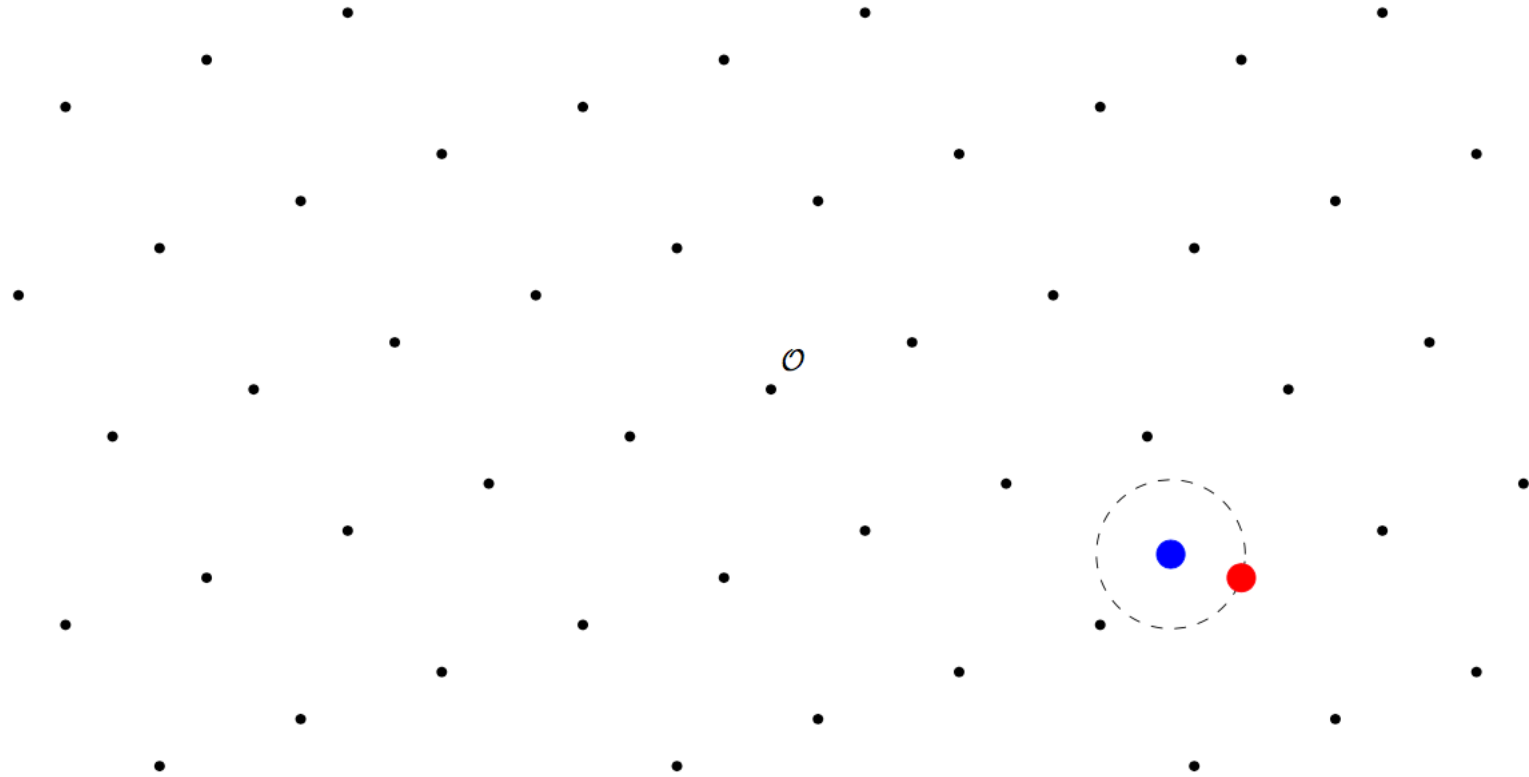
- Actually predates LWE

# 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  $H = AU$ .

# Closest vector problem



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

# 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 ( $\text{BDD}_\alpha$ ):** 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.)

# Strategies for solving LWE

SIS strategy

BDD strategy

Direct strategy

- See Albrecht, Player, Scott for a good survey

# 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.

# 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



# 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