NTNU | Norwegian University of Science and Technology

QUANTUM-SAFE ENCRYPTION

TTM4205 – Lecture 9

Tjerand Silde

15.09.2025

Contents

Quantum-Safe Cryptography

New Hardness Assumption

ML-KEM (CRYSTALS-Kyber)

Contents

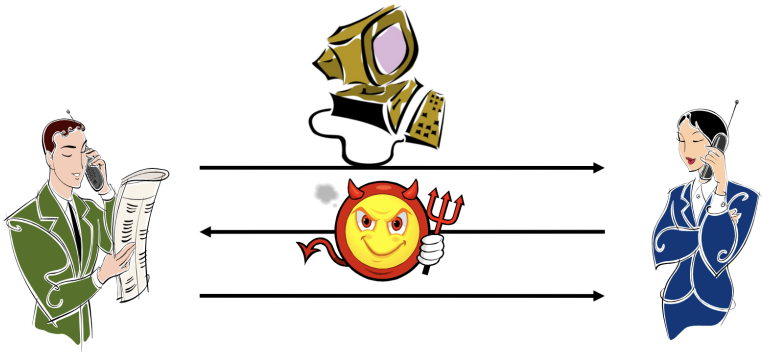
Quantum-Safe Cryptography

New Hardness Assumption

ML-KEM (CRYSTALS-Kyber)

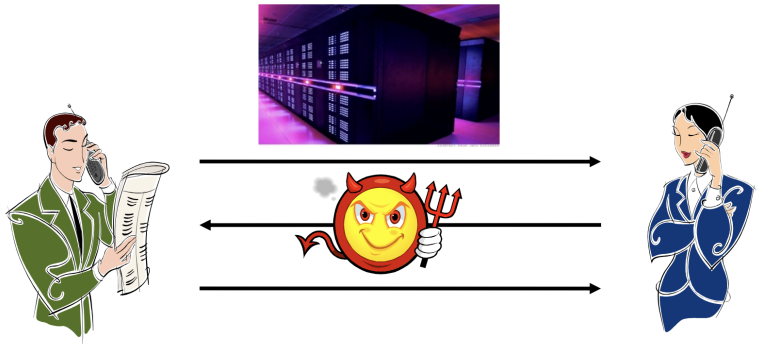
Cryptography Today

Allows for secure communication in the presence of malicious parties



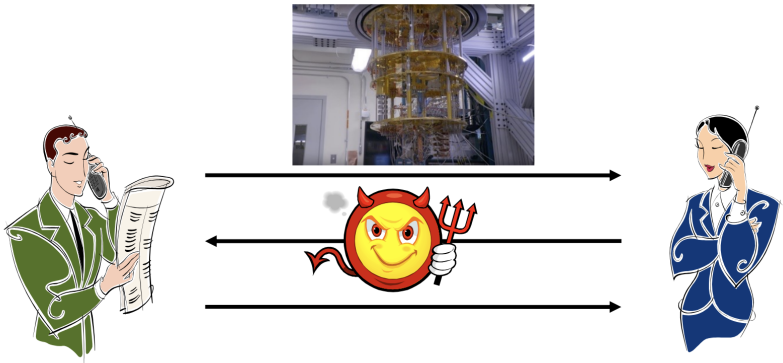
Cryptography Today

Large increase in the adversary's computing power
requires only a small increase in the key size



Cryptography Tomorrow

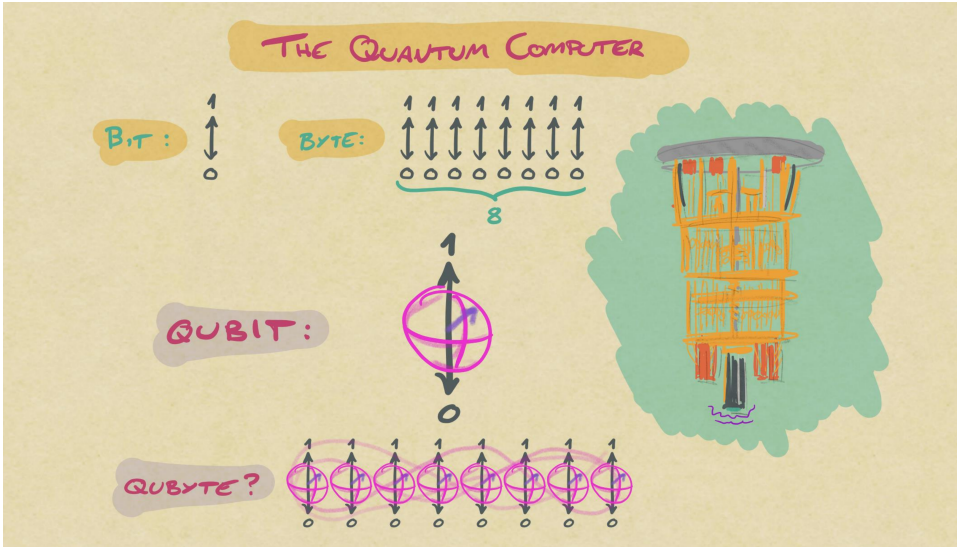
A quantum computer is outside the classical model of computation for efficiency purposes



Cryptography Tomorrow

Shor's quantum algorithm can factorize integers and compute discrete logs essentially as fast as using them, given a large quantum computer. This will break RSA, (EC)DH, (EC)DSA schemes and others relying these assumptions. To achieve future secrecy, there is an urgent need to replace those algorithms.

Quantum Computers



Quantum Computers

- Quantum computers are not better; they are different
- They will generally be worse, but do specific things better
- In theory, they can break public key encryption and digital signatures based on factoring and discrete log assumptions
- There are many recent developments in quantum computing

Quantum Computers

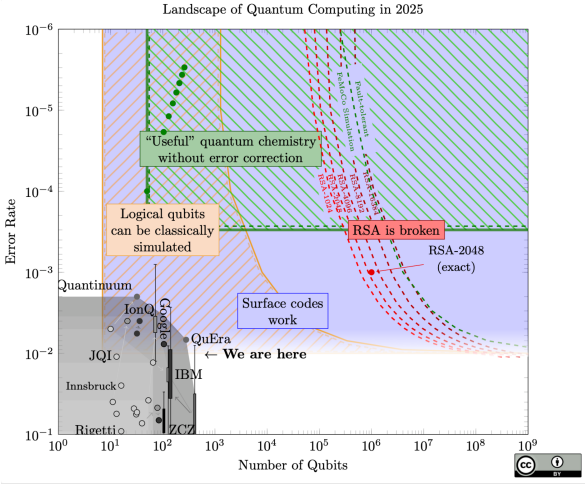


Figure: https://sam-jaques.appspot.com/quantum_landscape_2025

Factoring RSA

How to factor 2048 bit RSA integers with less than a million noisy qubits

Craig Gidney

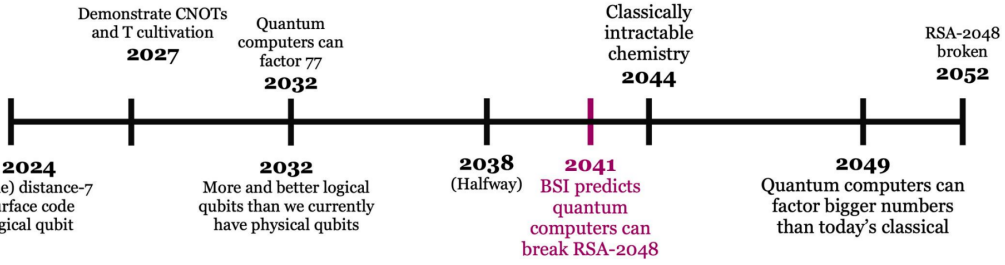
Google Quantum AI, Santa Barbara, California 93117, USA

June 9, 2025

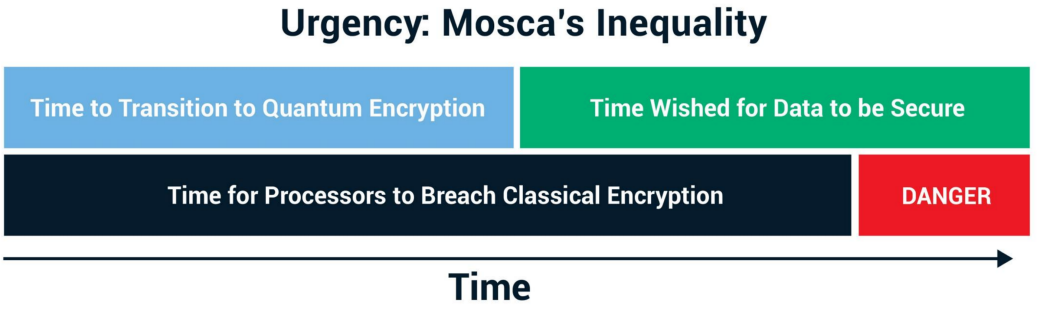
Planning the transition to quantum-safe cryptosystems requires understanding the cost of quantum attacks on vulnerable cryptosystems. In Gidney+Ekerå 2019, I co-published an estimate stating that 2048 bit RSA integers could be factored in eight hours by a quantum computer with 20 million noisy qubits. In this paper, I substantially reduce the number of qubits required. I estimate that a 2048 bit RSA integer could be factored in less than a week by a quantum computer with less than a million noisy qubits. I make the same assumptions as in 2019: a square grid of qubits with nearest neighbor connections, a uniform gate error rate of 0.1%, a surface code cycle time of 1 microsecond, and a control system reaction time of 10 microseconds.

Figure: https://arxiv.org/pdf/2505.15917

BSI Timeline

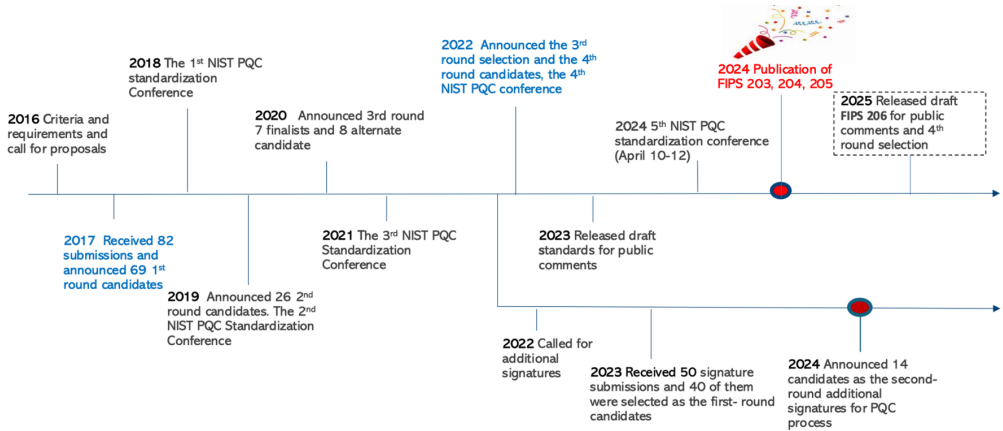


Harvest now, decrypt later

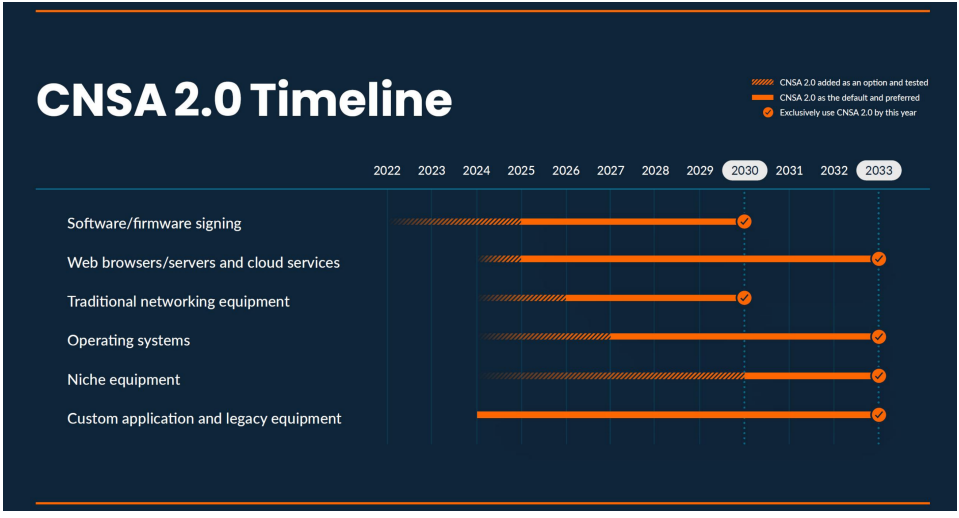


Don't wait - upgrade your encryption now!

NIST Timeline



NSA Timeline



Hybrid

Quantum-resistant schemes

- Establishing a shared secret between two parties:

Scheme	*Status*
ML-KEM	D

ML-KEM must be used in hybrid mode with a quantum-vulnerable key establishment scheme using an appropriate KEM combiner. The recommended parameter sets are ML-KEM-768 and ML-KEM-1024.

Figure: `https://nsm.no/regelverk-og-hjelp/veiledere-og-handboker/kryptografiske-anbefalinger-en-veileder-fra-nsm`

Cloudflare

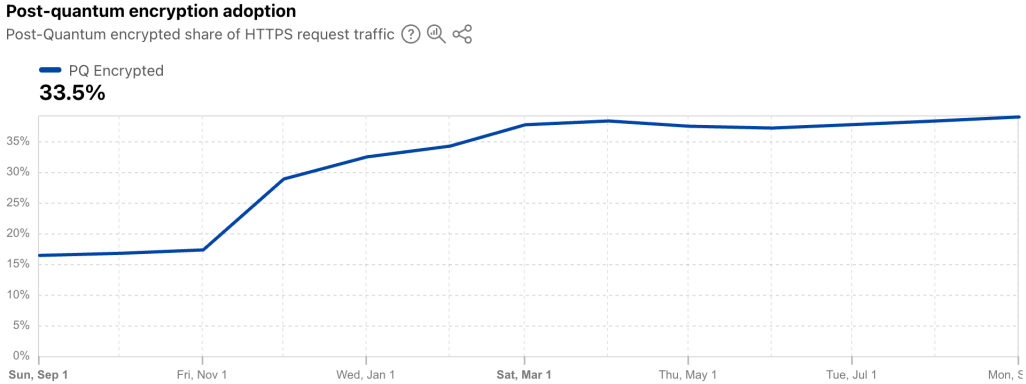
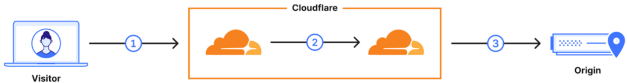


Figure: `https://radar.cloudflare.com/adoption-and-usage#post-quantum-encryption-adoption`

Cloudflare



On essentially all domains served (1) through Cloudflare, including this one, we have enabled hybrid post-quantum key agreement. We are also rolling out support for post-quantum key agreement for connection from Cloudflare to origins (3). Check out our blog post the state of the post-quantum Internet for more context.

You are using *X25519MLKEM768* which is **post-quantum secure**.

Deployed key agreements

Available with TLSv1.3 including HTTP/3 (QUIC)

Key agreement	TLS identifier
X25519MLKEM768	`0x11ec` (recommended)
X25519Kyber768Draft00	`0x6399` (obsolete), ~~`0xfe31`~~

Figure: `https://pq.cloudflareresearch.com`

Signal

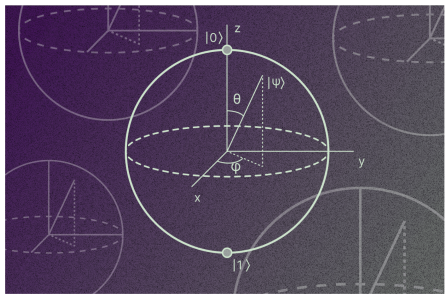


Figure: `https://signal.org/blog/pqxdh`

iMessage

Quantum-Secure Cryptography in Messaging Apps

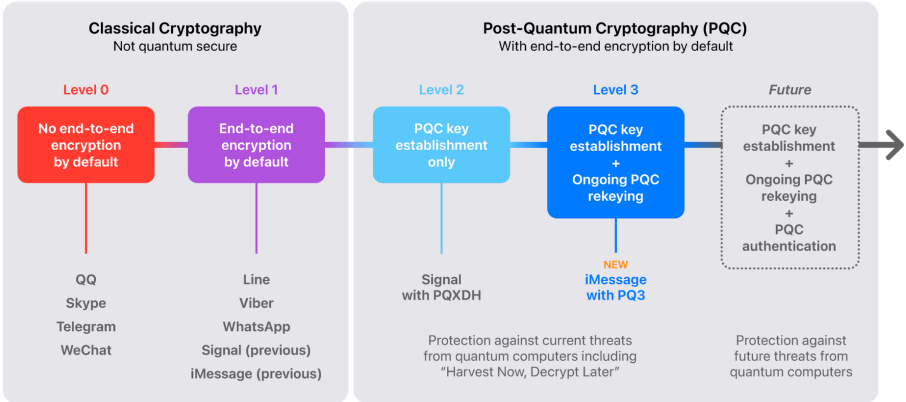


Figure: https://security.apple.com/blog/imessage-pq3

Crypto Categories

No Changes Necessary

Symmetric Cryptography:
- AES
- SHA-256 / SHA-3
- HMAC
- etc.

Done.

Almost Drop-in Replacements

NIST standardizations:
- Public Key Encryption
- Key Exchange
- Digital Signatures

A few other things:
- Identity-Based Encryption

Almost standards. Ready for deployment.

Serious Alterations of Protocols Required

Advanced Primitives:
- Zero-Knowledge Proofs
- Distributed Privacy
- Many blockchain privacy applications

Lots of recent progress on design. Near-optimality has just been achieved for certain primitives. Implementation starting at ZRL.

Can Only Be Done with Lattice Cryptography

- Fully-Homomorphic Encryption (FHE) - computation over encrypted data
- Some Obfuscation (still unclear if it can be efficient or have any useful applications)

Implementation / deployment of FHE at Haifa.

Contents

Quantum-Safe Cryptography

New Hardness Assumption

ML-KEM (CRYSTALS-Kyber)

Recall: Decisional Diffie-Hellman

Let $\mathbb{G}$ be a group of prime order p and g be a generator for $\mathbb{G}$.

Sample a, b, c uniformly at random from $\mathbb{Z}_p$. The Decisional Diffie-Hellman problem is to distinguish the two cases:

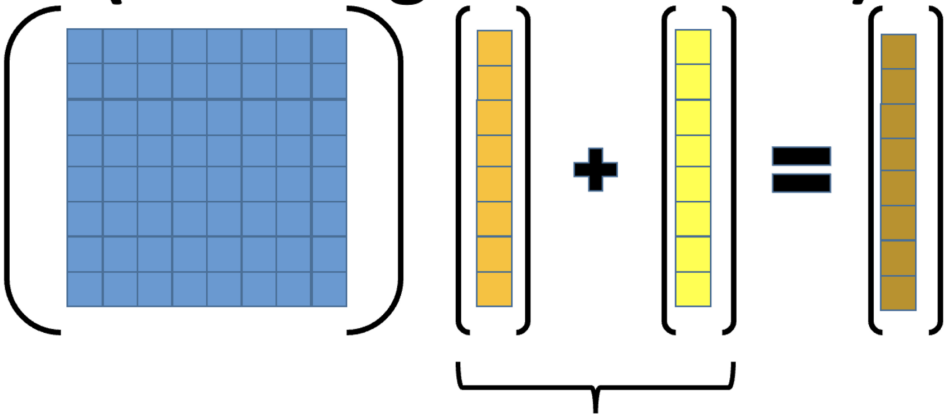
1. (g, g^a, g^b, g^{ab})
2. (g, g^a, g^b, g^c)

Learning With Errors (LWE)

Definition 1. For positive integers m, n, q, and $\beta < q$, the $\mathsf{LWE}_{n,m,q,\beta}$ problem asks to distinguish between the following two distributions:

1. $(\mathbf{A}, \mathbf{As} + \mathbf{e})$, where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{s} \leftarrow [\beta]^m, \mathbf{e} \leftarrow [\beta]^n$
2. $(\mathbf{A}, \mathbf{u})$, where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{u} \leftarrow \mathbb{Z}_q^n$.

(Learning With Errors)



LWE Hardness

The Learning With Errors problems gets harder when...

- the dimension gets larger
- the secret values gets larger
- the modulus gets smaller

Survey

Basic Lattice Cryptography

The concepts behind Kyber (ML-KEM) and Dilithium (ML-DSA)

Vadim Lyubashevsky

IBM Research Europe, Zurich
vad@zurich.ibm.com

(Last updated: June 18, 2025)

Figure: https://eprint.iacr.org/2024/1287.pdf

Contents

Quantum-Safe Cryptography

New Hardness Assumption

ML-KEM (CRYSTALS-Kyber)

FIPS 203

Federal Information Processing Standards Publication

Module-Lattice-Based Key-Encapsulation Mechanism Standard

Category: Computer Security Subcategory: Cryptography

Figure: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.203.pdf

Defining a KEM

Definition 1 (Key Encapsulation Mechanism (KEM))**.** A key encapsulation mechanism is a triple of algorithms KEM $= \{\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}\}$ with public keyspace $\mathcal{PK}$, private keyspace $\mathcal{SK}$, ciphertext space $\mathcal{C}$, and shared keyspace $\mathcal{K}$. The triple of algorithms is defined as:

- $\mathrm{KEM.}\mathsf{KeyGen}(\,) \xrightarrow{\$} (sk, pk)$ Randomized algorithm that outputs a secret (private) key $sk \in \mathcal{SK}$, and a public key $pk \in \mathcal{PK}$.
- $\mathrm{KEM.}\mathsf{Enc}(pk) \xrightarrow{\$} (k, c)$ Randomized algorithm that, given a public key $pk \in \mathcal{PK}$, outputs a shared key $k \in \mathcal{K}$, and a ciphertext $c \in \mathcal{C}$.
- $\mathrm{KEM.}\mathsf{Dec}(c, sk) \rightarrow y \in \{k, \perp\}$ Deterministic algorithm that, given a secret key, $sk \in \mathcal{SK}$ and a ciphertext $c \in \mathcal{C}$, returns the shared key $k \in \mathcal{K}$. In case of rejection, this algorithm returns $\perp$.

Figure: https://cic.iacr.org/p/1/1/21/pdf

Recall: ElGamal

Let $\mathbb{G}$ be a group of prime order p and g be a generator for $\mathbb{G}$. Denote by pp the public parameters $(\mathbb{G}, g, p)$.

Let the secret key $\mathsf{sk} \leftarrow\!\!\$\ \mathbb{Z}_p$ be sampled uniformly at random, and let the public key be $\mathsf{pk} = g^{\mathsf{sk}}$, where pk is made public.

The ElGamal encryption scheme, with message $m \in \mathbb{G}$, works as follows:

Enc : Sample uniform $x \leftarrow\!\!\$\ \mathbb{Z}_p$ and encrypt as $X = g^x$ and $Y = \mathsf{pk}^x \cdot m$.

Dec : Decrypt the ciphertext (X, Y) to get the message $m = Y \cdot X^{-\mathsf{sk}}$.

ML-KEM KGen and Enc

$$\mathsf{sk}: \mathbf{s} \leftarrow [\beta]^m, \ \mathsf{pk}: (\mathbf{A} \leftarrow \mathbb{Z}_q^{m \times m}, \mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}_1), \text{ where } \mathbf{e}_1 \leftarrow [\beta]^m. \tag{6}$$

To encrypt a message $\mu \in \{0,1\}$, the encryptor chooses $\mathbf{r}, \mathbf{e}_2 \leftarrow [\beta]^m$ and $e_3 \leftarrow [\beta]$, and outputs

$$\left(\mathbf{u}^T = \mathbf{r}^T\mathbf{A} + \mathbf{e}_2^T, v = \mathbf{r}^T\mathbf{t} + e_3 + \left\lceil \frac{q}{2} \right\rfloor \mu\right). \tag{7}$$

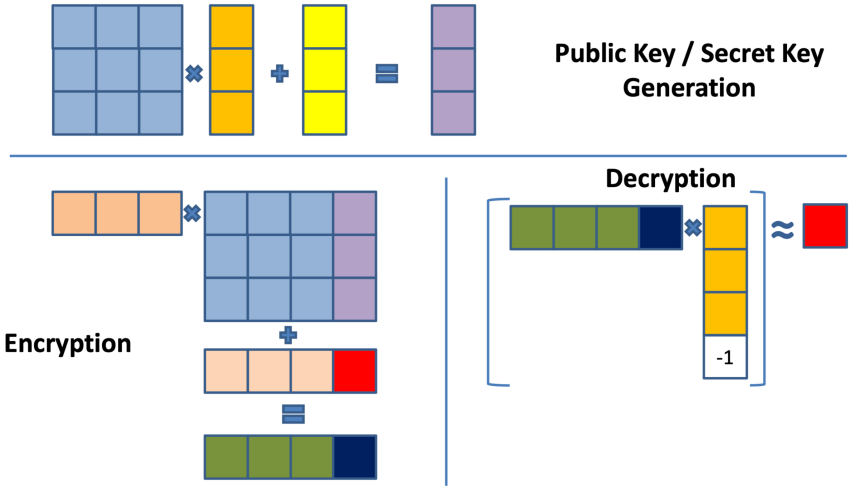
ML-KEM Dec

To decrypt, one computes $v - \mathbf{u}^{\tilde{T}}\mathbf{s}$. But rather than this cleanly giving us the message μ as in (4), we instead obtain

$$v - \mathbf{u}^T\mathbf{s} = \mathbf{r}^T(\mathbf{A}\mathbf{s} + \mathbf{e}_1) + e_3 + \frac{q}{2}\mu - \left(\mathbf{r}^T\mathbf{A} + \mathbf{e}_2^T\right)\mathbf{s} \tag{8}$$

$$= \mathbf{r}^T\mathbf{e}_1 + e_3 + \frac{q}{2}\mu - \mathbf{e}_2^T\mathbf{s} \tag{9}$$

Visualization of ML-KEM



Classical vs Quantum-Safe Key-Exchange

		Size keyshares(in bytes)		Ops/sec (higher is better)	
Algorithm	PQ	**Client**	**Server**	**Client**	**Server**
Kyber512	✅	800	768	50,000	100,000
Kyber768	✅	1,184	1,088	31,000	70,000
X25519	❌	32	32	17,000	17,000

X-Wing: Hybrid Key-Exchange

X-Wing private key (2464 bytes):

ML-KEM-768 private key (2400 bytes)	X25519 private key (32 bytes)	X25519 public key (32 bytes)

X-Wing public key (1216 bytes):

ML-KEM-768 public key (1184 bytes)	X25519 public key (32 bytes)

X-Wing ciphertext (1120 bytes):

ML-KEM-768 ciphertext (1088 bytes)	X25519 ciphertext (32 bytes)

X-Wing shared key (32 bytes):

SHA3-256 (

\./ /^\ (6 bytes)	ML-KEM-768 shared key (32 bytes)	X25519 shared key (32 bytes)	X25519 ciphertext (32 bytes)	X25519 public key (32 bytes)

)

Figure: https://cic.iacr.org/p/1/1/21/pdf

NIST Selects HQC as Fifth Algorithm for Post-Quantum Encryption

March 11, 2025

- NIST has chosen a new algorithm for post-quantum encryption called HQC, which will serve as a backup for ML-KEM, the main algorithm for general encryption.
- HQC is based on different math than ML-KEM, which could be important if a weakness were discovered in ML-KEM.
- NIST plans to issue a draft standard incorporating the HQC algorithm in about a year, with a finalized standard expected in 2027.

MEDIA CONTACT
301-975-2762

ORGANIZATIONS
Information Technology Laboratory
Computer Security Division
Cryptographic Technology Group

RELATED NEWS
NIST Releases First 3 Finalized Post-Quantum Encryption Standards

Figure: https://www.nist.gov/news-events/news/2025/03/nist-selects-hqc-fifth-algorithm-post-quantum-encryption

Questions?