NTNU | Norwegian University of Science and Technology

QUANTUM-SAFE SIGNATURES

TTM4205 – Lecture 10

Tjerand Silde

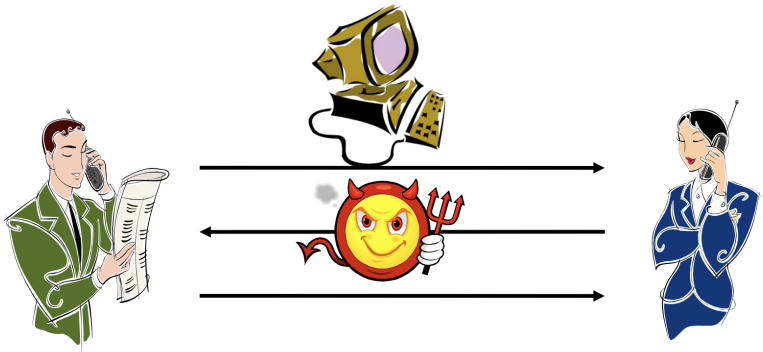
19.09.2025

Contents

Contents

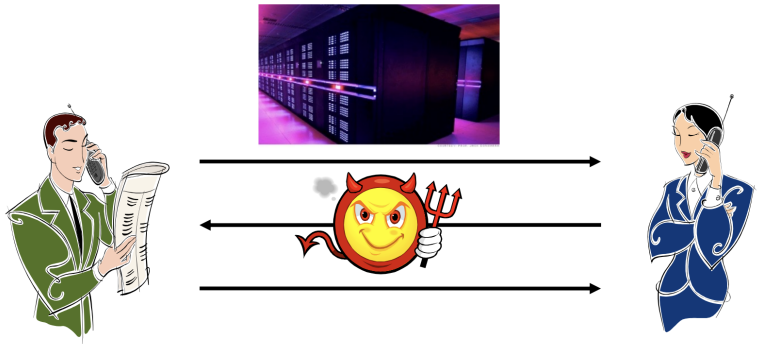
Cryptography Today

Allows for secure communication in the presence of malicious parties



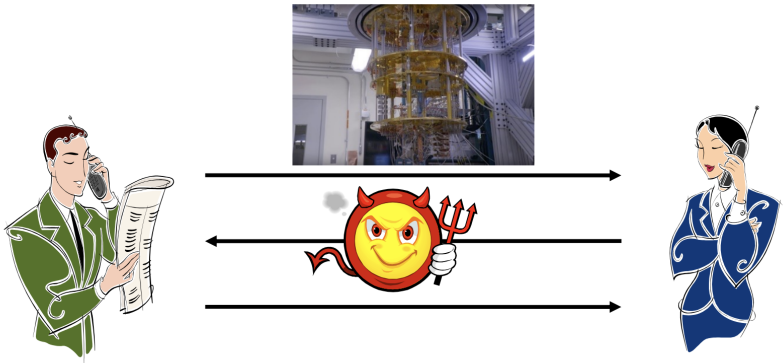
Cryptography Today

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



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

iMessage

Quantum-Secure Cryptography in Messaging Apps

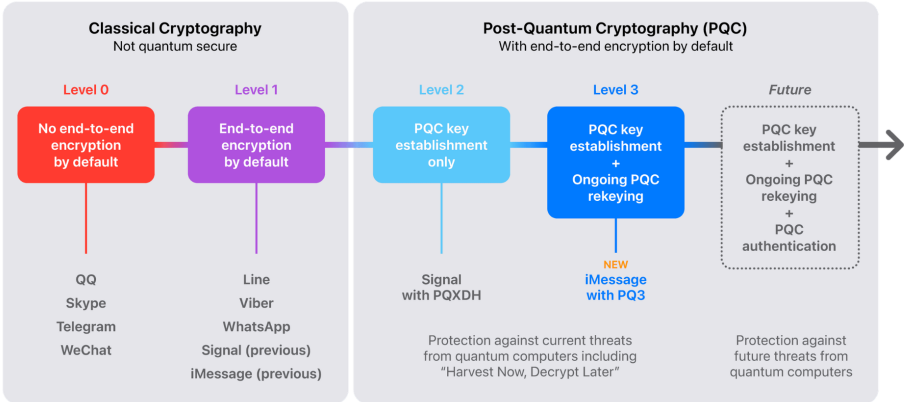


Figure: https://security.apple.com/blog/imessage-pq3

GitHub

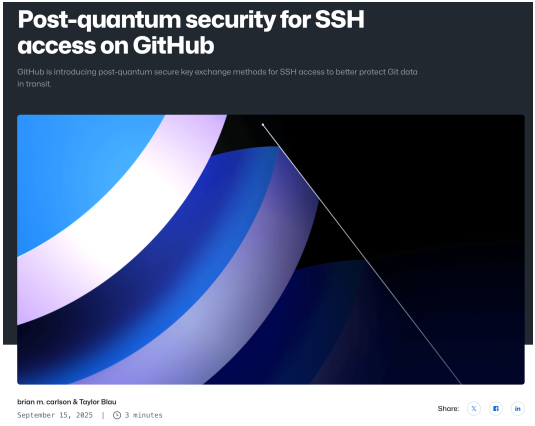


Figure: https://github.blog/engineering/platform-security/post-quantum-security-for-ssh-access-on-github

NIST Standards

The first Set of NIST PQC Standards

NIST

FIPS 203 Module-Lattice-Based Key-Encapsulation Mechanism Standard (Based on CRYSTALS-Kyber)	FIPS 204 Module-Lattice-Based Digital Signature Standard (Based on CRYSTALS-Dilithium)	FIPS 205 Stateless Hash-Based Digital Signature Standard (Based on SPHINCS+)	FIPS 206 FFT-Over-NTRU-Lattice-Based Digital Signature Standard (Based on FALCON, ***under development***)
• A module learning with errors (MLWE)-based key encapsulation mechanism (KEM) • Good performance in different platforms • An algorithm for key establishment in security protocols	• A lattice-based digital signature algorithm based on the Fiat-Shamir paradigm • Good performance, simple implementation, moderate public-key and signature size, suitable for general applications	• Not require to keep track of any state between signatures • Solid security, signatures are longer compared with ML-DSA	• Hash and sign paradigm • Smaller bandwidth and fast verification but more complicated implementation

Published August 2024!

Contents

Recall: Computational Diffie-Hellman

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

Sample a, b uniformly at random from $\mathbb{Z}_p$. The Computational Diffie-Hellman problem is, given g, g^a, and g^b, to find g^{ab} in $\mathbb{G}$.

Recall: Learning With Errors

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 $\mathbf{u} \leftarrow \mathbb{Z}_q^n$.

New: Short Integer Solution (SIS)

Definition 4. For positive integers m, n, q, and $\beta < q$, the $\mathsf{SIS}_{n,m,q,\beta}$ problem asks to find, for a randomly-chosen matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, vectors $\mathbf{s}_1 \in [\beta]^m$ and $\mathbf{s}_2 \in [\beta]^n$ such that $\mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 = \mathbf{0} \pmod{q}$.

SIS Hardness

The Short Integer Solution problem gets harder when...

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

This is opposite of LWE with respect to norms!

Hardness of LWE and SIS

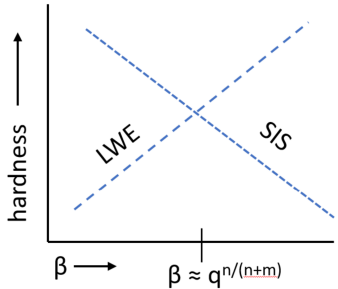


Figure 2: The hardness of $\mathsf{LWE}_{n,m,q,\beta}$ and $\mathsf{SIS}_{n,m,q,\beta}$ for fixed n, m, q, and varying β. The lines are not meant to describe the concrete hardness of these problems, but rather to illustrate the dependence of the hardness of these problems on β. The intersection point is approximately at $\beta = q^{n/(n+m)}$.

Survey

Basic Lattice Cryptography

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

Vadim Lyubashevsky

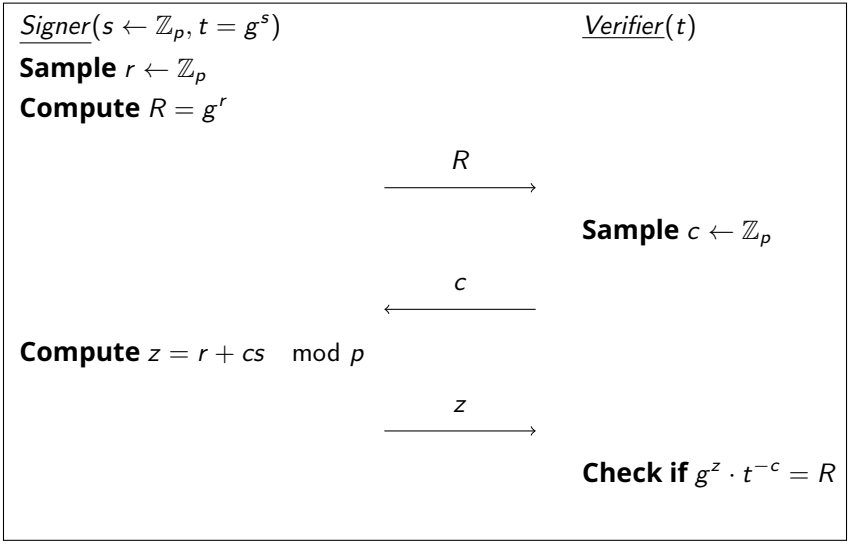
IBM Research Europe, Zurich
vad@zurich.ibm.com

(Last updated: June 18, 2025)

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

Contents

Recall: Schnorr Signatures



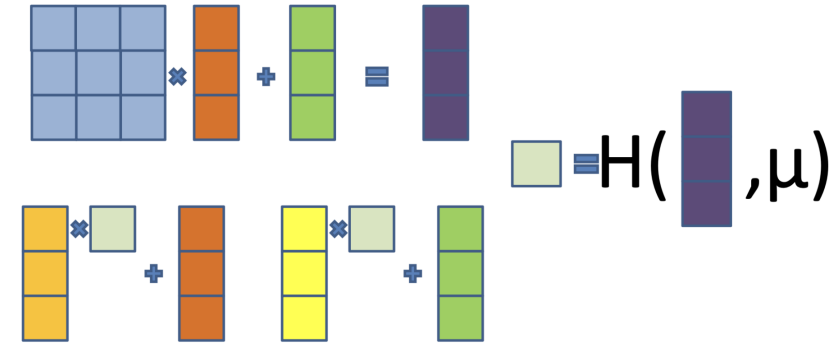
ML-DSA

Private information: $\mathbf{s}_1 \in [\beta]^m, \mathbf{s}_2 \in [\beta]^n$
Public information: $\mathbf{A} \in \mathcal{R}_{q,f}^{n \times m}, \mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 \in \mathcal{R}_{q,f}^n$

Prover		Verifier
$\mathbf{y}_1 \leftarrow [\gamma + \bar{\beta}]^m$ $\mathbf{y}_2 \leftarrow [\gamma + \bar{\beta}]^n$, $\mathbf{w} := \mathbf{A}\mathbf{y}_1 + \mathbf{y}_2$		
	$\xrightarrow{\mathbf{w}}$	
		$c \leftarrow \mathcal{C}$
	$\xleftarrow{c}$	
$\mathbf{z}_1 := c\mathbf{s}_1 + \mathbf{y}_1$ $\mathbf{z}_2 := c\mathbf{s}_2 + \mathbf{y}_2$ if $\mathbf{z}_1 \notin [\bar{\beta}]^m$ or $\mathbf{z}_2 \notin [\bar{\beta}]^n$ then $(\mathbf{z}_1, \mathbf{z}_2) := \perp$		
	$\xrightarrow{(\mathbf{z}_1, \mathbf{z}_2)}$	
		Accept iff $\mathbf{z}_1 \in [\bar{\beta}]^m$ and $\mathbf{z}_2 \in [\bar{\beta}]^n$ and $\mathbf{A}\mathbf{z}_1 + \mathbf{z}_2 - c\mathbf{t} = \mathbf{w}$

Visualization of ML-DSA

Public Key / Secret Key Generation



Sizes

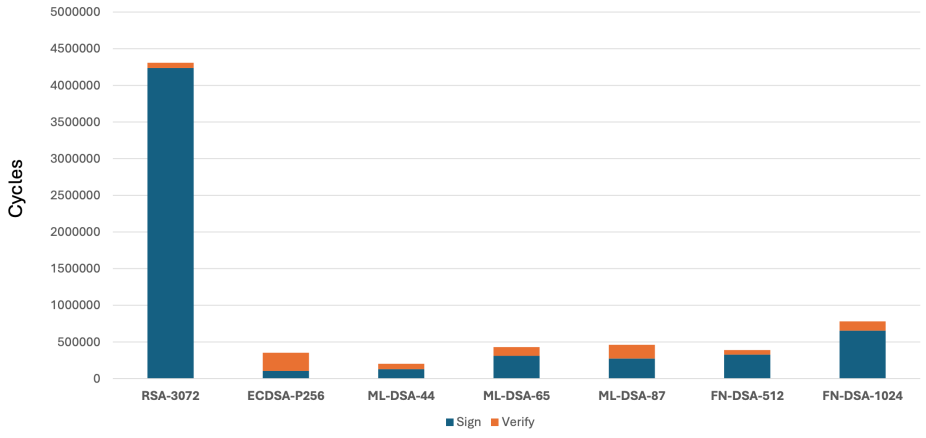
PQC Key and Signature Sizes

NIST

Scheme	Public Key (bytes)	Private Key (bytes)	Signature (bytes)	Security Level
RSA-3072	**384**	**384**	**384**	**Classical-128**
ECDSA-P256	**64**	**32**	**256**	**Classical-128**
ML-DSA-44 (Dilithium2)	**1312**	**2528**	**2420**	**PQC Category 2** *(SHA3-256)*
ML-DSA-65 (Dilithium3)	**1952**	**4000**	**3293**	**PQC Category 3** *(AES-192)*
ML-DSA-87 (Dilithium5)	**2592**	**4864**	**4595**	**PQC Category 5** *(AES-256)*
FN-DSA-512 (Falcon512)	**897**	**7553**	**666**	**PQC Category 1** *(AES-128)*
FN-DSA-1024 (Falcon1024)	**1793**	**13953**	**1280**	**PQC Category 5** *(AES-256)*

Performance

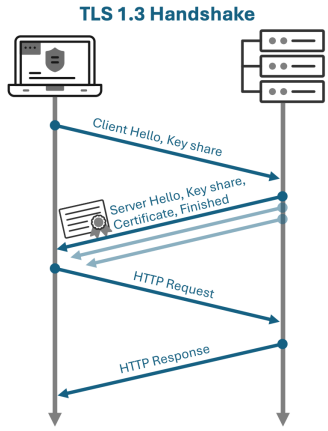
PQC Signatures– Performance



Signatures in TLS

A bit much to chew?

- TLS & WebPKI Certificate Signatures
  - *Server Certificate*: 1 public key and signature, 2 SCT signatures
  - *Intermediate CA Certificate*: 1 public key and signature
  - *TLS Handshake*: 1 signature
  - ML-DSA-44 → **14,724 bytes**
  - Current Quantum-Vulnerable → **1,248 bytes**
- ML-KEM-768 key shares
  - Client → Server: 1,184 bytes
  - Server → Client: **1,088 bytes**
- Why does this matter?
  - *TCP initial congestion window* limits the first wave of messages
  - Typical default: **~14,600 bytes**
- Without protocol/implementation changes, this could slow web connection establishment



Contents

FN-DSA (FALCON)

FN-DSA (FALCON)

- Based on a different lattice-assumption called NTRU
- Smaller keys and signatures compared to ML-DSA
- Much more complex key generation and signing algorithms
- Very difficult to secure against side-channel attacks
- The standard is not ready yet and the scheme might change

SLH-DSA (SPHINCS+)

FIPS 205

Federal Information Processing Standards Publication

Stateless Hash-Based Digital Signature Standard

Category: Computer Security **Subcategory: Cryptography**

SLH-DSA (SPHINCS+)

- Only based on the security of hash-functions
- The most conservative design for signatures
- Small keys but large and slow signatures
- Great fit for certificates or code signing
- There exists more efficient stateful versions
- Easy to implement securely in hardware

Contents

Additional Signatures

Table 2. First-round digital signature candidates organized by category, with the candidates selected to advance to the second round bolded and in blue. The starred signature schemes MIRA and MiRitH merged to form a new candidate Mirath.

Code-Based	Lattice-Based	MPC-in-the-Head	Multivariate
CROSS	EagleSign	Biscuit	3WISE
Enhanced pqsigRM	EHTv4	**MIRA***	DME-Sign
FuLeeca	HAETAE	**MiRitH***	HPPC
LESS	**HAWK**	**MQOM**	**MAYO**
MEDS	HuFu	**PERK**	PROV
WAVE	Raccoon	**RYDE**	**QR-UOV**
	SQUIRRELS	**SDitH**	**SNOVA**
Other			TUOV
ALTEQ	Symmetric-Based	Isogeny-Based	**UOV**
eMLE-Sig 2.0	AIMer	**SQIsign**	VOX
KAZ-SIGN	Ascon-Sign		
PREON	**FAEST**		
Xifrat1-Sign.I	SPHINCS-alpha		

Figure: https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8528.pdf

Additional Signatures

On-Ramp Signatures

NIST

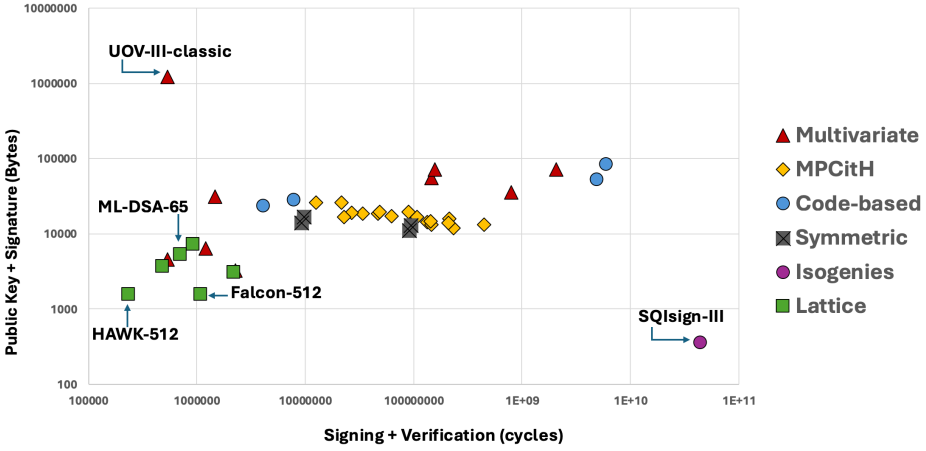
- Why NIST called for additional post-quantum signatures?
  - NIST is primarily interested in additional general-purpose signature schemes that are not based on structured lattices.
  - NIST may also be interested in signature schemes that have short signatures and fast verification.
  - Any lattice signature would need to significantly outperform CRYSTALS-Dilithium and FALCON and/or ensure substantial additional security properties.
- Received 50 submissions June 1, 2023 – 40 of them are accepted as the first-round candidates
- NIST announced 14 candidates to advance to the second round of the additional digital signatures for the PQC standardization process on October 24, 2024

Multivariate		MPC in-the-head			Lattice	Code	Symmetric	Isogeny
UOV	MinRank	SD/Rank-SD	PKP	MQ				
Mayo	Mirath	Ryde	Perk	MQOM	Hawk	Cross	FAEST	SQIsign
QR-UOV		SDitH				LESS		
SNOVA								
UOV								

Figure:
https://csrc.nist.gov/projects/pqc-dig-sig/round-2-additional-signatures

Additional Signatures

Performance Summary (log scale)



Contents

The Next Steps

- The new FALCON and HQC standards must be completed
- The additional signature competition must be completed
- Standards for KEM and DSA must be included in TLS etc.
- Extensive research into side-channels and implementation
- New research designing other schemes from same assumptions
- Update and upgrade existing protocols and systems to PQC

Questions?