

# Cybersecurity in Post-Quantum Era: On Post-Quantum Cryptography and Security Protocols

Lukas Malina

Brno University of Technology  
[malina@vut.cz](mailto:malina@vut.cz)

ESSCaSS 2023, Tartu



# About CHES project

**Cyber-security Excellence Hub in Estonia and South Moravia (CHES)** brings together leading R&I institutions in both regions to build connected innovation ecosystem in **Cybersecurity**.

6 Research Challenges:

- Internet of Safe Things
- Security Certification
- Verification of Trustworthy Software
- Security Preservation in Blockchain Technology
- **Post-Quantum Cryptography**
- Human Centric Aspects of Security

Website: <https://chess-eu.cs.ut.ee/>

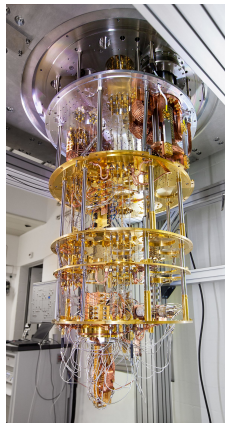


Funded by the European Union under Grant Agreement No. 101087529. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.



# Quantum Computers and Current Cryptography

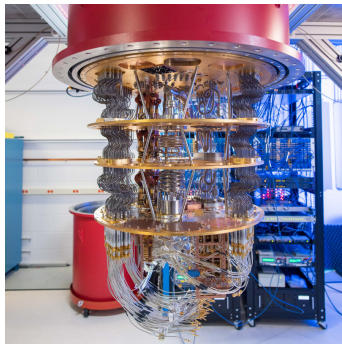
Functional **Quantum Computers QC** can break current asymmetric cryptosystems (e.g. RSA, DSA, ECC).



IBM Quantum Computer,  
Zurich

# Quantum Computers - Current State - Google

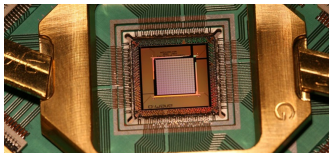
**Google** (+NASA and universities, 2019) - Sycamore processor (see Fig. below) - connected qubits create up to 53 qubits -  $10^{16}$  computational states in 1 dimension.<sup>1</sup>



<sup>1</sup><https://www.nature.com/articles/s41586-019-1666-5>

# Quantum Computers - Current State - D-Wave

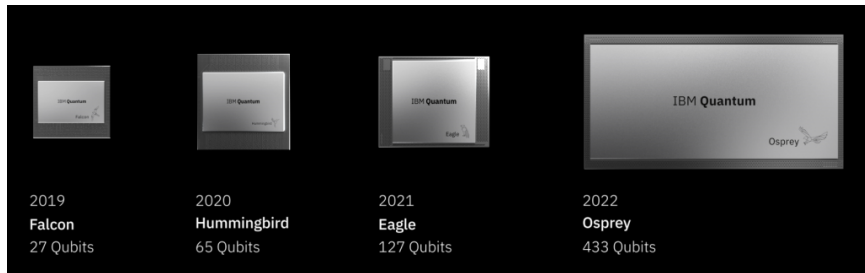
- D-Wave - single-purpose machines using quantum annealing - **cannot perform the Shor algorithm!**
- Quantum Computer D-Wave 2000Q s "2048" qubits (high error rate).
- Based on superconducting circuits (extreme cooling up to 15 mK / -273° C).
- Price approx. 15 million USD, the advantage is a lower consumption of 25 kW vs a similarly fast supercomputer of 2.5 MW.
- More info here: <https://www.dwavesys.com/d-wave-two-system>



# Quantum Computers - Current State - IBM



IBM - 2020 28-qubit QC (IBM Q - Raleigh), reduction of errors on gates.<sup>2</sup>, 11/2022  
433-qubit Osprey.



<sup>2</sup><https://www.ibm.com/blogs/research/2020/01/quantum-volume-32/>

**Shor's Algorithm running on QC allows efficient solving math problems used in asymmetric cryptography (i.e. Integer Factorization Problem IFP and Discrete Logarithm Problem DLP (Elliptic Curves too!)).**

- **Shor's Algorithm (1994)** runs in **polynomial time**.
- **Shor's Algorithm** threatens current asymmetric cryptography schemes e.g. RSA, ECDSA, DH etc. **Asymmetric cryptography** currently (2023) **not yet compromised (!!)** because Shor's algorithm needs on thousands of **qubits** without errors (i.e. stable qubits).

**This is an algorithm for quantum computers searching with high probability for a unique input in a black box function that produces a certain output. It runs in  $O(\sqrt{N})$  time, where  $N$  is the size of the function domain.**

- Lov Grover, formulation in 1996.
- **Grover's Algorithm** can be used for cracking passwords, solving the collision problem, or finding a symmetric key by brute force (e.g., a 128-bit key in  $2^{64}$  iterations).
- Recommendation: **hash functions and symmetric cryptography** - double-size the key lengths.



## Quantum Cryptography

- ✓ Uses quantum mechanics and photonics for security.
- ✗ Expensive special equipment and infrastructure.
- ✗ Only key establishment.

## Post-Quantum Cryptography

- ✓ Based on **hard** math problems.
- ✓ Runnable on Classic Computers (higher memory requirements only).
- ✓ Key establishment, encryption, signatures.
- ✓ Standardized by NIST.

Combination of QKD and PQC? Check our project **NESPOQ** (<https://www.nespoq.cz/>).

# QKD (Quantum Key Distribution) at BUT

- ID Quantique (IDQ) rack QKD system - experimental testbed with Alice and Bob (depicted below) at Brno Uni. of Technology.
- Using COW protocol (modification of Bennett Brassard 1984 protocol).



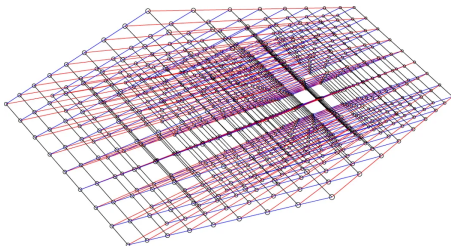
## Post-Quantum Cryptography (PQC):

- Represents **quantum-computer-attack-resistant cryptographic schemes**.
- Focuses on the **replacement** of the compromised **asymmetric cryptographic schemes** based on FP, DLP and ECDLP.
- **Asymmetric encryption, signature and key establishment** (some principles and systems are over 40 years old).
- Symmetric schemes are partly compromised due to Grover's algorithm, key lengths will need to be doubled.

- **Lattice**-based cryptography.
- **Code**-based cryptography.
- **Multivariate** cryptography.
- **Hash**-based cryptography.
- **Isogeny**-based cryptography - Supersingular isogeny Diffie-Hellman.

**LBC uses point hiding in a high-dimensional grid (mod  $q$ ) using a small change of all coordinates.**

Based on problems: Learning With Errors, Ring-LWE Problem.



3-dimension lattice <sup>3</sup>

---

<sup>3</sup> Bernstein, Daniel J., and Tanja Lange. "Post-quantum cryptography." *Nature* 549.7671 (2017): 188-194.

Examples of lattice-based cryptosystems for key establishment:

- BCNS15 - R-LWE problem.
- New Hope - improved version of BCNS15, R-LWE problem.
- Frodo - 2016, LWE,  $n \times n$  matrices, larger keys.
- KYBER - 2016, Kyber.AKE, Ring-LWE.

Examples of Lattice-based cryptosystems for encryption:

- NTRU - designed by Hoffstein, Pipher and Silverman, works with  $N$  polynomials.
- LP - proposed by Lindner, Peikert, LWE, works with matrices.
- KYBER - 2016, Kyber.CPA for encryption.
- Kyber.Hybrid = (KeyGen, Encaps, Decaps) - singryption.

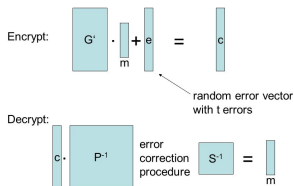
Examples of Lattice-based signature cryptosystems:

- Dilithium - uses the Fiat-Shamir design.
- qTESLA - uses the Fiat-Shamir design.

# Code-Based Cryptography

- The earliest approach of post-quantum schemes (McEliece, 1978).
- **CBC uses point hiding in a very-high-dimensional grid (mod 2) by changing some coordinates.**
- **Based on correction codes**, when the so-called Syndrome Decoding Problem (SDP) - NP-hard is used.
- The advantage is speed (higher than, for example, RSA).
- The disadvantage is the size of the private and public keys (in the form of a matrix).

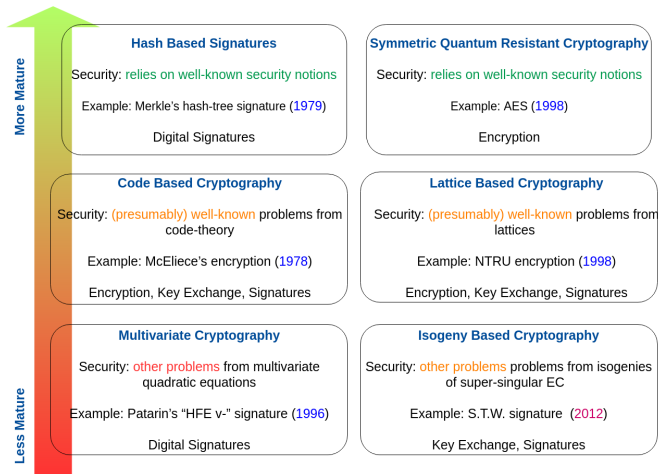
## The McEliece Cryptosystem



- **Based on the computational difficulty of finding a hash function input message collision**  $O(2^{n/2})$ .
- **Merkle signature** (1979) - MSS (Merkle Signature Scheme), hashing binary trees, applications in ZFS file systems, Bittorrent, Bitcoin.
- **XMSS scheme** (eXtended Merkle Signature Scheme, 2011) - long key generation time (tree structure) mil.ms - signature several tens of ms, unit of ms verification, on CPU 2.53 GHz, signature size 34-15 kb, public key size 14 kb, secret key 280 b.  
<http://www.dcs.fmph.uniba.sk/stanek/sighash.pdf>



# PQC Comparison



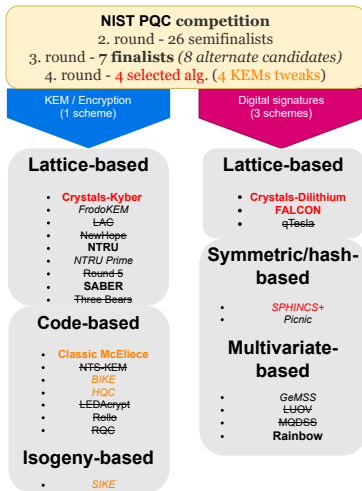


Fig. 4 : Selected Algorithms 2022 for Standardization.

Digital Signatures					
Scheme	Type	Sec. Level [b]	Private Key [B]	Public Key [B]	Signature [B]
Dilithium	lattice	125	-	1 472	2 701
Falcon	lattice	$\gg 128$	-	1 441	993,91
GeMSS	multivariate	128	14 208	417 408	48
LUOV	multivariate	128	32	7 300	1 700
MQDSS	multivariate	128	32	62	32 882
Picnic	symmetric/hash	128	32	64	195 458
qTESLA	lattice	$\gg 128$	12 320	39 712	6 176
Rainbow	multivariate	$\gg 128$	511 400	206 700	156
SPHINCS+	hash	128	64	32	16 976

# NIST PQC: Memory Assessment II.

Key Establishment (KEM)/Encryption Schemes					
Scheme	Type	Sec. Level [b]	Private Key [B]	Public Key [B]	Ciphertext [B]
BIKE	code	128	249	2 541	2 541
McEliece	code	128	6 452	261 120	128
Kyber	lattice	128	1 632 (or 32)	800	736
FrodoKEM	lattice	128	19 888	9 616	9 720
HQC	code	128	252	6 170	6 234
LAC	lattice	128	512	544	712
LEDAcrypt	code	128	452	1 872	1 872
NewHope	lattice	128	869	928	1 088
NTRU	lattice	128	1 452	1 138	1 138
NTRU Prime	lattice	128	1 125	897	1 025
NTS-KEM	code	128	9 248	319 488	1 024
ROLLO-I	code	128	40	465	465
Round5	lattice	128	16	634	682
RQC	code	128	40	853	1690
SABER	lattice	≫ 128	1 568	672	736
SIKE	isogeny	128	374	330	346
Three Bears	lattice	128	40	804	917
FrodoKEM	lattice	128	19872	9616	9736

- 09/2022 **NSA** (USA) recommends under Commercial National Security Algorithm Suite (CNSA 2.0) exchange **directly** for **PQC**, i.e., PQC Dilithium (signature) and Kyber (KEM) and SW/FW signatures using XMSS and LMS schemes.
- **ANSII** (FR) + **BSI** (GE) recommends a more cautious approach (**hybrid** combination of PQC with asym. cryp. for the period 2026 - 2030, then full transition).
- After 2030/2033, a full transition from asymmetric cryptography to QR solutions (PQC, possibly QKD) can be assumed.

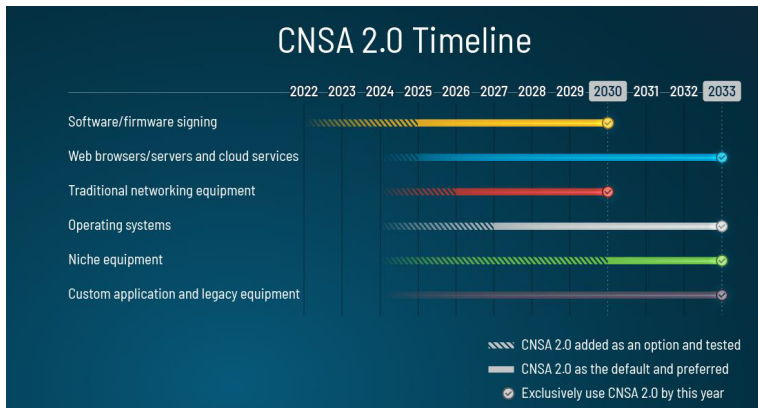
# Algorithm Suites (CNSA 1.0 and CNSA 2.0)



Function	Algorithms CNSA 1.0	Algorithms CNSA 2.0
Symmetric encryption	Advanced Encryption Standard (AES) with 256-bit keys	Advanced Encryption Standard (AES) with 256-bit keys
Key establishment	RSA with min. 3072-bit modulus, ECDH P-384, DH 3072-bit modulus	CRYSTALS-Kyber with sec. level V
General digital signatures	RSA with min. 3072-bit modulus, ECDSA P-384	CRYSTALS-Dilithium with sec. level V
Digital signatures of FW/SW	Not specified, could be used RSA with min. 3072-bit modulus, ECDSA P-384	<b>Leighton-Micali Signature (LMS)</b> with SHA256, <b>Xtended Merkle Signature Scheme (XMSS)</b>
Data hashing	SHA-384	SHA-384 or SHA512

# CNSA 2.0 Timeline

## CSNA 2.0 Timeline<sup>4</sup>:



<sup>4</sup>[https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA\\_CNSA\\_2.0\\_ALGORITHMS\\_.PDF](https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF)

- **MACsec** provides L2 security via Ethernet frames encryption and data integrity and authenticity by adding the Security TAG (SecTAG).
- AES-GCM encryption with 256 bits keys is quantum-safe.
- **MACsec Key Agreement** (MKA) protocol establishes keys for MACsec by using pre-shared keys or by using key exchange.
- PQ MKA variant is needed but PQ schemes exceed the Ethernet MTU (ca 1500 bytes).
- Current works (Gazdag *et al.*; Cho and Sergeev) test Classic McEliece, NTRU, CRYSTALS-Kyber, and SABER, together with classical Diffie-Hellman.
- PQ schemes did not significantly impact MACsec.



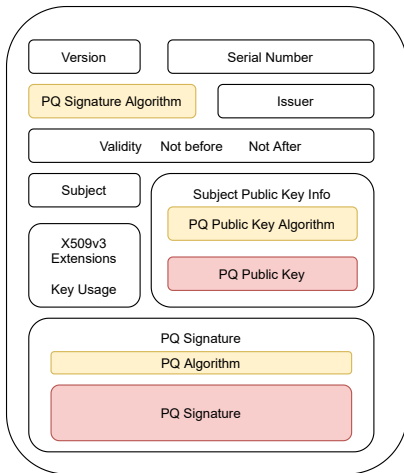
- **IPsec** secures connections at L3 by encrypting packets and ensuring their authenticity and integrity.
- Internet Key Exchange Version 2 (IKEv2) (component of IPsec) provides mutual authentication and key exchange.
- Quantum-safe IKEv2 must support PQC schemes.
- Strongswan library offers PQ schemes in the strongSwan 6.0beta Post-Quantum IKEv2 Daemon (Kyber, NTRU, Saber, Frodo, HQC, Dilithium, Falcon), using liboqs.
- Classic McEliece is not supported due to huge public key sizes (more than 100 kB).

- **TLS** as an essential security protocol at L4.
- PQ KEMs and PQ signatures in TLS versions 1.2 and 1.3 on various platforms.
- In 2016, Braithwaite experimented with a hybrid KEM (New Hope combined with elliptic curves) and ECDSA certs in TLS 1.2 in Google Chrome.
- In 2020, Paquin *et al.* assessed PQ performance in TLS (variants: ECDH + SIKE/Kyber/Frodo, and PQ digital signatures, i.e., ECDSA, Dilithium, qTESLA, Picnic).
- Using non-standard jumbo frames (from 1500 to 9000 B) may positively affect PQ-TLS performance.

- **SSH** (v2) is similar to TLS in negotiation, key establishment via KEMs with digital signatures, and symmetric encryption.
- SSH message lengths defined by 4-byte fields,  $2^{32}$ -byte messages, **large enough for PQC**.
- But OpenSSH has packet size limit of  $2^{18}$  (262,144) bytes, complication for Classic McEliece with large keys.
- Several works, e.g., Sikeridis *et al.*, measured the handshake of SSH PQC variants (Kyber512+SPHINCS128 takes 755 ms vs classic ECDH384+RSA2048 takes 584 ms on common servers with PCs using OQS-OpenSSH library).

- The size of **X.509 certificates** usually depends on the length of attributes, and extensions and mainly on the used algorithms' signatures and their public keys stored inside the certificate.
- Typical certificate sizes vary from 500 to 1500 bytes.
- PKI uses trusted root certificates and Intermediate Certification Authority (ICA) certificates.
- Certificate chains are often composed of two to four certificates with a total size of a few kilobytes (KB).

PQ signatures have usually **larger sizes** of public **keys** and **signatures**, deploying them into current X.509 and PKI standards has been already investigated by several works.



# PQC in Security Libraries 1/2



Name	PQC supported	Platforms/licenses/languages	Website
liboqs	Kyber, Classic McEliece, BIKE, HQC, FrodoKEM, NTRU-Prime, Dilithium, Falcon, SPHINCS+	open source C library for x86_64 and ARM architectures for prototyping; bindings/wrappers in Python, C#, C++, JAVA, Go, Rust	<a href="https://openquantumsafe.org/liboqs/">https://openquantumsafe.org/liboqs/</a>
libpqcrypto	19 PQC schemes (BIG QUAKE, Classic McEliece, DILITHIUM, KYBER, DAGS, FrodoKEM, Gui, KINDI, LUOV, MQDSS, NewHope, NTRU-HRSS-KEM, NTRU Prime, Picnic, qTESLA, Rainbow, Ramstake, SABER, SPHINCS+)	Experimental C library for Debian/Ubuntu systems, Python/C API	<a href="https://libpqcrypto.org/">https://libpqcrypto.org/</a>
CIRCL	Dilithium, Kyber, FrodoKEM, SIDH/SIKE	Cloudflare Interoperable, Reusable Cryptographic Library written in Go	<a href="https://github.com/cloudflare/circl">https://github.com/cloudflare/circl</a>
ISARA Radiate™ Quantum-safe Library V3.1	Classic McEliece, DILITHIUM, KYBER, HSS, XMSS, SPHINCS+	Multiplatform (Windows, Linux, macOS, iOS, Android) licensed C-based library via gcc or clang compilers	<a href="https://www.isara.com/toolkit/3/doc/library/index.html">https://www.isara.com/toolkit/3/doc/library/index.html</a>
Bouncy Castle	Kyber, Dilithium, Falcon, SPHINCS+, BIKE, HQC, NTRU, NTRU Prime, Picnic, FrodoKEM, GeMSS, LMS, Newhope, Rainbow, Saber, XMSS, SIKE	MIT licensed Java library with cryptographic algorithms and security protocols	<a href="https://www.bouncycastle.org/releasesnotes.html">https://www.bouncycastle.org/releasesnotes.html</a>



# PQC in Security Libraries 2/2



Name	PQC supported	Platforms/licenses/languages	Website
OpenSSH	NTRU (hybrid mode with EC method), XMSS (only for experimental use)	BSD-style licensed SSH library with the support of the LibreSSL library for some cryptography methods	<a href="https://www.openssh.com/">https://www.openssh.com/</a>
OQS-OpenSSH	alone or in hybrid mode with ECDH: BIKE, ClassicMcEliece, FrodoKEM, HQC, Kyber; Dilithium, Falcon, SPHINCS-Haraka/SHA256/SHAKE256	A fork based on OpenSSH version 8.9 with liboqs, tested on Ubuntu 20.04.1	<a href="https://www.openssh.com/">https://www.openssh.com/</a>
OQS-OpenSSL	BIKE, Kyber, HQC, FrodoKEM, Dilithium, Falcon, SPHINCS-Haraka/SHA256/SHAKE256	A fork of multi-platform robust OpenSSL library (1.1.1) written in C; PQC is provided via oqsprovider, depended on liboqs	<a href="https://github.com/open-quantum-safe/oqs-provider">https://github.com/open-quantum-safe/oqs-provider</a>
Wolf SSL	Dilithium, FALCON, SPHINCS+, Kyber KEM	Lightweight SSL/TLS library written in ANSI C for embedded, RTOS, and resource-constrained environments	<a href="https://www.wolfssl.com/">https://www.wolfssl.com/</a>

- In the next decades **quantum computers may break asymmetric** cryptographic schemes (RSA, DH, ECDH, DSA, BBS, BS04, etc.), which are used in IPsec, SSH, TLS/HTTPS, etc.
- **PQC can help** but usually works with longer parameters (keys, signatures) compared to classic cryptography.
- **Symmetric** cryptography is **OK** (compromised only when using low key lengths  $< 160$  bits).
- 2022 **NIST** announced **4** candidates for standardization (KEM - **Kyber**, signatures - **Dilithium, Falcon, SPHINCS+**), launched the 4th round for KEM alternatives (3x Code-based, 1x supersing. EC).
- Many institutions (NSA, NIST, ENISA, BSI) release recommendations with **transaction** period ca. **2025 - 2030**.
- **Hybrid** vs pure PQC substitution is still open topic.
- PQC have started to be included in variants of IPsec, TLS or SSH libraries.



-  Editors: Daniel Bernstein, Johannes Buchman, Erik Dahmen  
*Post-quantum cryptography*.  
Springer, Berlin, Heidelberg, 2009. 1-249.
-  Buchmann, Johannes, and Jintai Ding.  
*Post-quantum cryptography*.  
second international workshop, PQCrypto. 2008.
-  Lukas Malina and Sara Ricci and Petr Dzurenda and David Smekal and Jan Hajny and Tomas Gerlich.  
*Towards Practical Deployment of Post-quantum Cryptography on Constrained Platforms and Hardware-Accelerated Platforms*.  
SecITC 2019.
-  Lukas Malina and Patrik Dobias and Jan Hajny and Kim-Kwang Raymond Choo.  
*On Deploying Quantum-Resistant Cybersecurity in Intelligent Infrastructures*.  
ARES/SP2I 2023 (in print).