



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

DII
Dipartimento di Ingegneria
dell'Informazione

Venti anni di Ricerca in Crittografia Post-Quantum

Marco Baldi

Università Politecnica delle Marche, Dipartimento di Ingegneria dell'Informazione
`m.baldi@univpm.it`

8 Ottobre 2026

Venti anni di Ricerca in Crittografia Post-Quantum



Università Politecnica delle Marche

Faculty of Engineering

Research Doctorate Course in
Electronics, Informatics and Telecommunications Engineering

Quasi-Cyclic Low-Density Parity-Check Codes and their Application to Cryptography

Marco Baldi

Author: Thesis Supervisor:

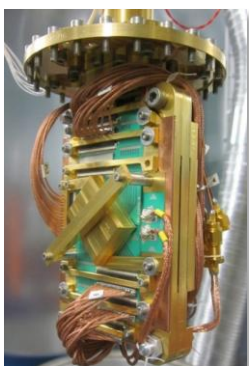
Prof. Franco Chiaraluce

Course Coordinator:

Prof. Francesco Piazza

November 2006

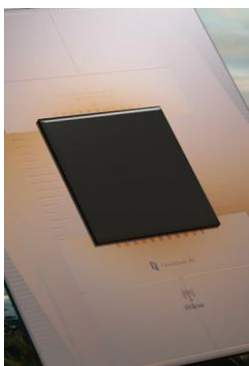
Venti anni di Ricerca in Quantum Computing



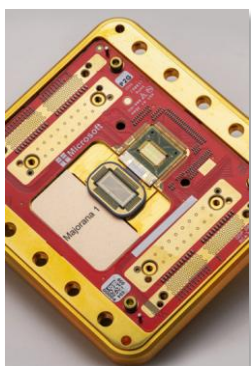
2007
D-Wave
Orion
16 qubits
(annealer)



2023
IBM
Condor
1121 qubits
(supercond.)



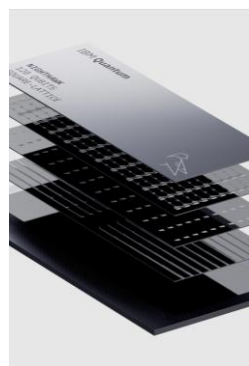
2024
Google
Willow
105 qubits
(scalable)



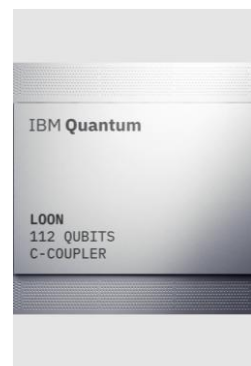
2025
Microsoft
Majorana 1
8 qubits
(topological)



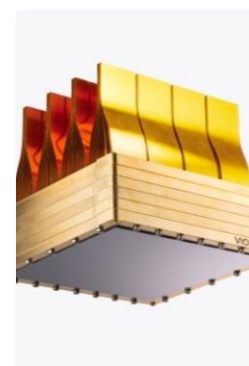
2025
Amazon
Ocelot
9 qubits
(fault tolerant)



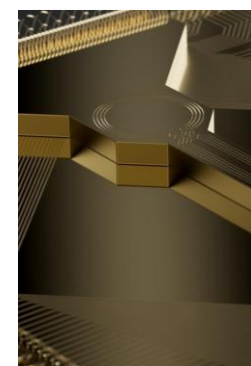
2025
IBM
Nighthawk
120 qubits
(connected)



2025
IBM
Loon
112 qubits
(error correcting)



2025
QuantWare
VIO-40K
10000 qubits
(3D stack)



2025
Quantinuum
Helios
98 qubits
(48 logical, 2:1)

Quantum Computer e Crittografia



ASYMMETRIC
CRYPTOGRAPHY

RSA,
ElGamal,
Diffie-Hellman,
DSA, ECDSA...

Livello di sicurezza
ridotto da
esponenziale a
polinomiale

Da sostituire con **crittografia
post-quantum** (PQC),
tenendo anche conto di
attacchi **harvest-now-
decrypt-later**



SYMMETRIC
CRYPTOGRAPHY

AES,
ChaCha20,
SHA2, SHA3...

Livello di sicurezza
che rimane
esponenziale, con
una riduzione
quadratica

Ancora **utilizzabile**,
possibilmente aumentando
la lunghezza di chiavi e
digest

Famiglie di crittosistemi post-quantum

- **Sistemi asimmetrici:**

- Basati su reticoli
- Basati su codici
- Basati su sistemi multivariati
- Basati su funzioni hash
- Basati su isogenie
- ...



Devono sostituire i loro predecessori quantum-vulnerable (RSA, ECC...)

- **Sistemi simmetrici:**

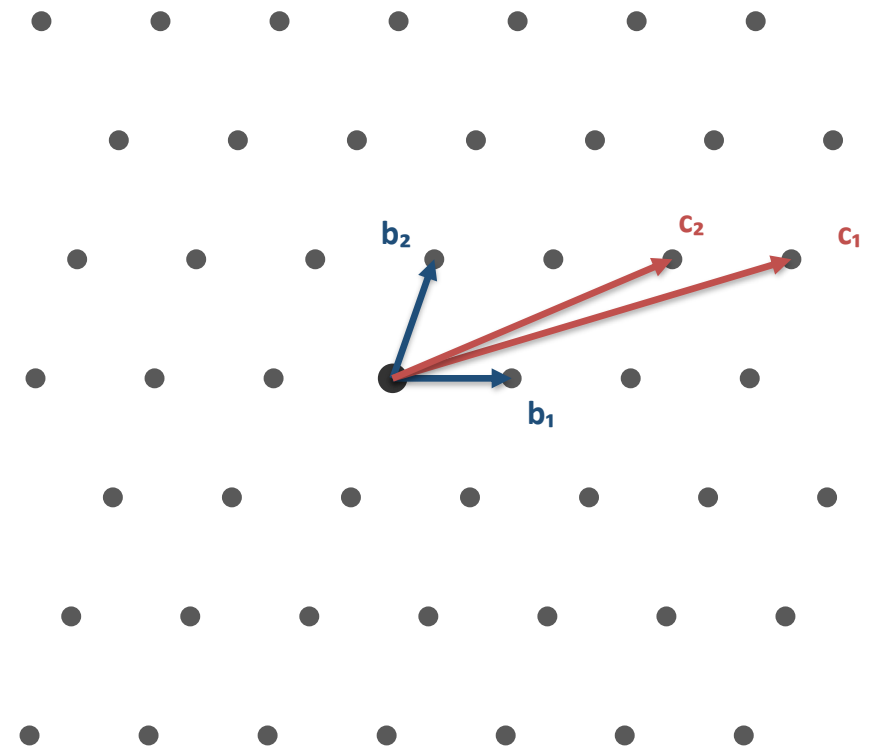
- Cifrari simmetrici (AES...)
- Funzioni hash (SHA...)



Ancora utilizzabili, tenendo conto dell'accelerazione quantum degli attacchi

Crittografia basata su Reticoli

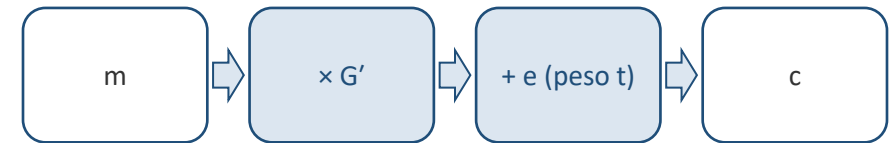
- Un **reticolo** è una griglia di punti descrivibile con infinite basi: “buone” (vettori corti, quasi ortogonali) e “cattive” (vettori lunghi, obliqui).
- Problemi difficili: trovare vettore più corto (**SVP**) e vettore più vicino (**CVP**), anche in versione approssimata. I migliori algoritmi che li risolvono hanno complessità **esponenziale**, anche con un quantum computer.
- Trapdoor: base buona = chiave privata, base cattiva = chiave pubblica (ad esempio in Falcon).
- Learning With Errors (Regev, **2005**): dati **A** e **b** = **As** + **e** mod q con **e** “rumore” piccolo, trovare **s**.
- Punti di forza: miglior compromesso **dimensioni/velocità**; copre cifratura, KEM, firme e costruzioni avanzate (cifratura omomorfa).



Crittografia basata su Codici

- **Codice lineare:** oggetto matematico descritto da una matrice generatrice **G** o di parità **H**, usato per codificare sequenze di dati per poi correggere gli errori che le corrompono.
- Problema difficile: decodificare un codice lineare generico, trovando il vettore di errori.
- NP-completo (Berlekamp, McEliece, van Tilborg **1978**); i migliori attacchi sono **esponenziali**.
- Trapdoor (McEliece, **1978**): un codice con decodifica efficiente (Goppa) mascherato come codice generico.
- Chiave pubblica **G'**; chiave privata (**S, G, P**); cifratura **c = mG' + e** con **e** vettore di *t* errori casuali.
- Il più antico schema post-quantum (coetaneo di RSA).

Cifratura (chiave pubblica $G' = S \cdot G \cdot P$)

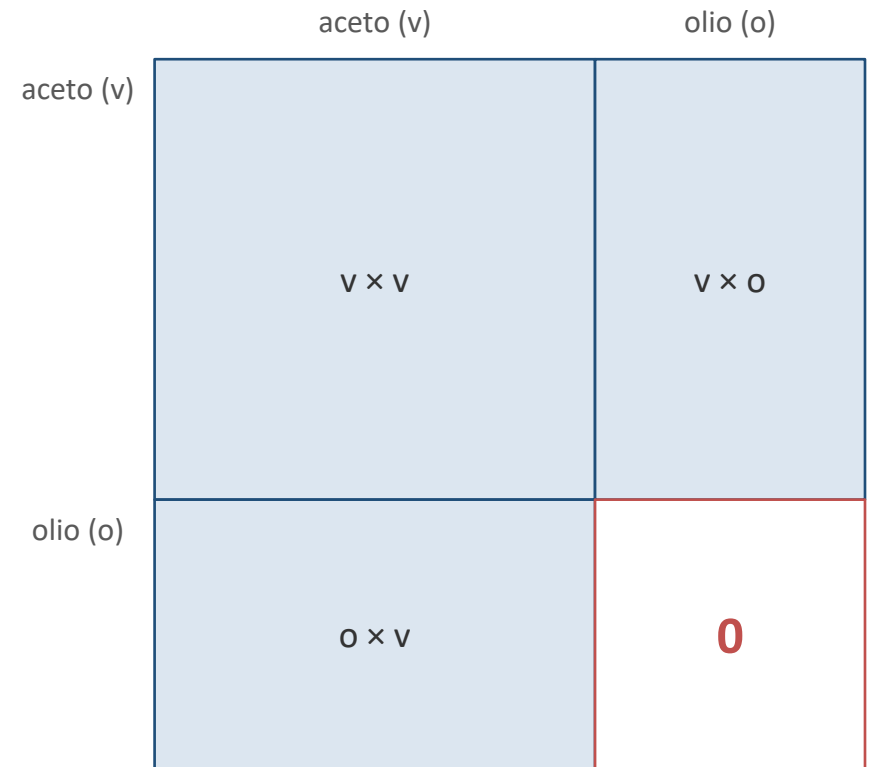


Decifratura (chiave privata **S, G, P**)



Crittografia Multivariata

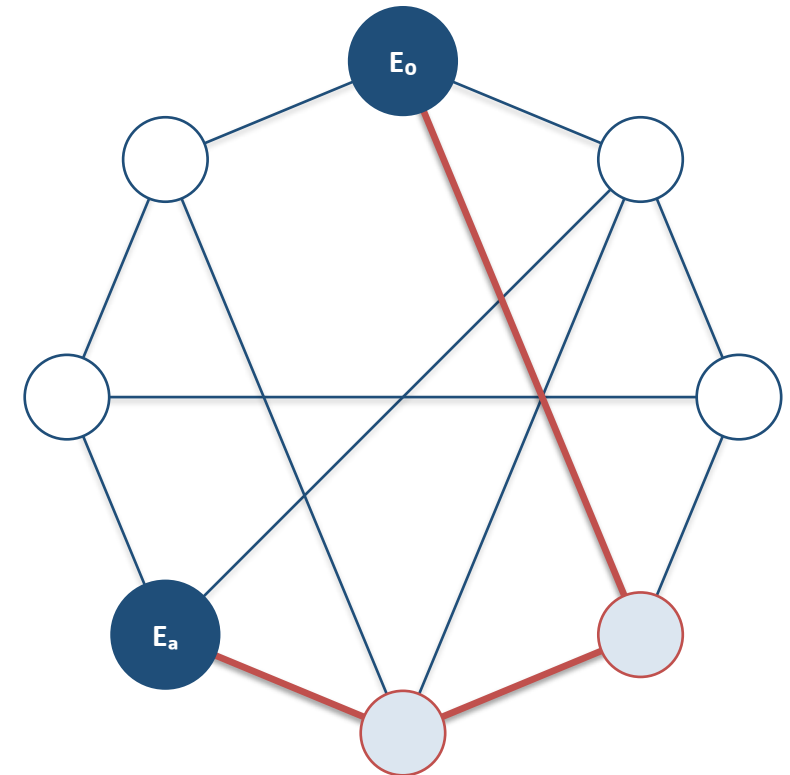
- Risolvere un sistema di m **equazioni polinomiali** (di solito quadratiche) in n variabili su un campo finito è un problema NP-difficile nel caso generale e nessun algoritmo quantistico noto lo risolve in tempo polinomiale.
- Chiave pubblica: la mappa $P = (p_1, \dots, p_m)$, ossia i coefficienti di m polinomi quadratici.
- Chiave privata: la struttura nascosta $P = S \circ F \circ T$, con F "mappa centrale" facile da invertire e S, T trasformazioni segrete che la mascherano.
- Adatta solo alla firma: le varianti per cifratura (Matsumoto–Imai **1988**, HFE **1996**) sono state rotte o restano poco pratiche.
- Sistemi più promettenti derivati da Unbalanced Oil and Vinegar (UOV, **1999**).



Struttura di un polinomio centrale UOV

Crittografia basata su Isogenie tra Curve Ellittiche

- **Isogenia**: mappa tra curve ellittiche che rispetta la struttura di gruppo.
- Trovare un'isogenia che colleghi due curve ellittiche è un problema difficile da risolvere, anche con un quantum computer.
- Il segreto non è un numero, ma un cammino in un **grafo** di curve ellittiche.
- Scambio di chiavi basato su isogenie (SIDH) introdotto nel **2011**, crittanalizzato a Luglio **2022** da Castryck, Decru ed altri.
- Schema di firma basato su isogenie (**SQIsign**) tuttora promettente.
- Chiavi e messaggi **piccoli**, ma calcoli lenti e matematica giovane, con storia recente di rotture.



Il Progetto NIST PQcrypto

- Nel **2016** il NIST ha avviato un processo per lo sviluppo e la standardizzazione di uno o più algoritmi crittografici a chiave pubblica per:
- **Public-key encryption (PKE)**
 - key generation, encryption and decryption
- **Key encapsulation mechanism (KEM)**
 - key generation, encapsulation and decapsulation
- **Digital signature**
 - key generation, signature generation and signature verification



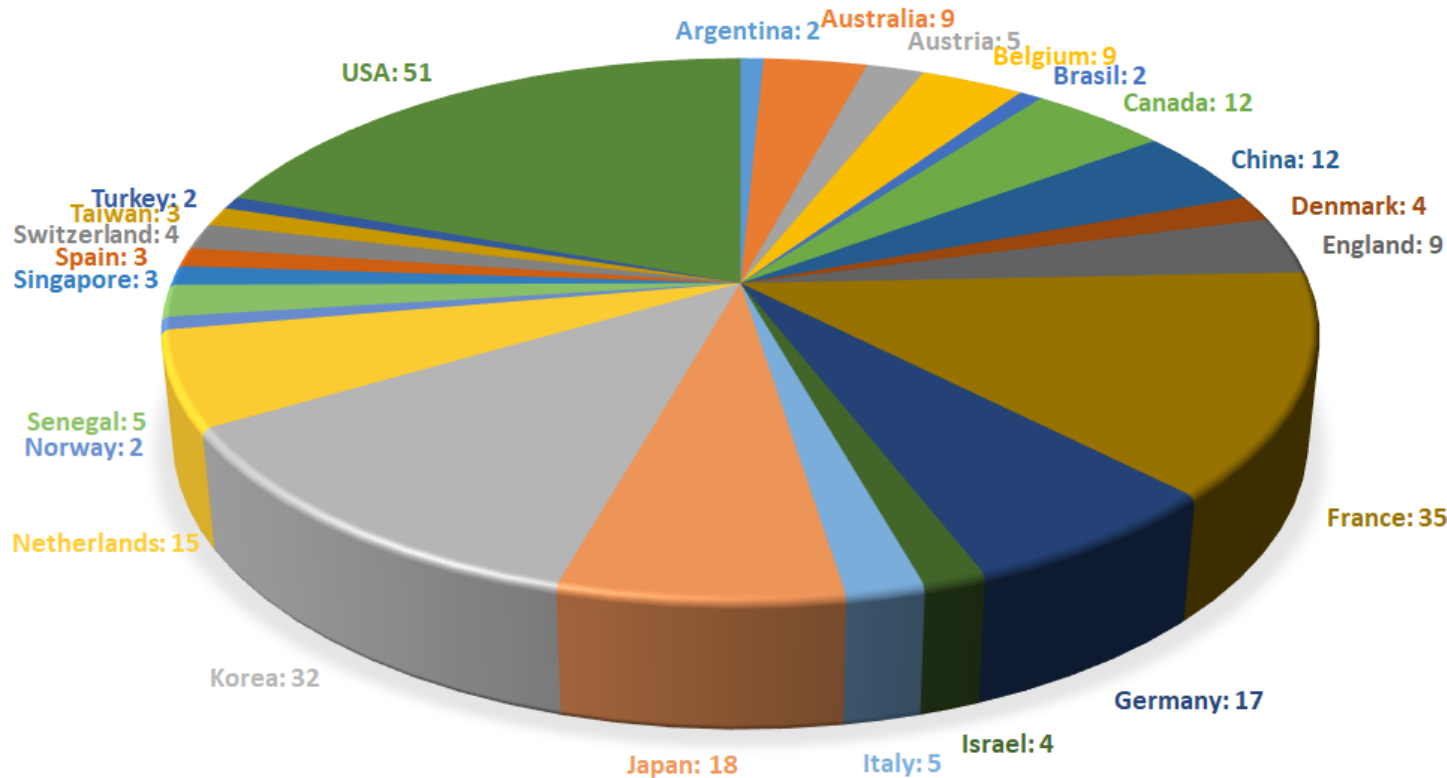
NIST – livelli di sicurezza



- “Any attack that breaks the relevant security definition must require computational resources comparable to or greater than those required for:”
 1. key search on a block cipher with a **128-bit** key (e.g. AES128)
 2. collision search on a **256-bit** hash function (e.g. SHA256/ SHA3-256)
 3. key search on a block cipher with a **192-bit** key (e.g. AES192)
 4. collision search on a **384-bit** hash function (e.g. SHA384/ SHA3-384)
 5. key search on a block cipher with a **256-bit** key (e.g. AES256)

Il Progetto NIST PQcrypto (2)

263 ricercatori da **24** Paesi



FINAL SUBMISSIONS RECEIVED

- The deadline is past – no more submissions
- 82 total submissions received
 - 23 signature schemes
 - 59 Encryption/KEM schemes

	Signatures	KEM/Encryption	Overall
Lattice-based	4	24	28
Code-based	5	19	24
Multi-variate	7	6	13
Hash-based	4		4
Other	3	10	13
Total	23	59	82

Il Progetto NIST PQcrypto – ingaggio della comunità



HQC and constant-time error correction 83 views

 **Mike Hamburg**
to pqc-...@list.nist.gov
Dec 31, 2017, 10:35:24 PM

Hello PQCers,

Looking at HQC, it appears to depend heavily on BCH codes (tensored with repetition codes) for correctness. These codes are configured to efficiently correct dozens of errors. If I am reading the implementation correctly, the decoding algorithm isn't constant-time: in fact, it uses branches and table lookups based on secret information. This seems like a serious side-channel risk to me.

There are actually several proposed systems that use public error-correcting codes — that is, their security isn't based on a hidden error-correcting code, but they use codes to overcome noise in the decryption process. At least Hila5 and ThreeBears also do this, and I expect there are others.

So I'm curious if the community has any references or ideas for error-correcting codes/implementations that are:

- * Efficient in terms of CPU and memory requirements.
- * Simple to implement efficiently in constant time. HQC's BCH code seems to fail this, but perhaps its authors or the community can point to a simple constant-time implementation?
- * Efficient in terms of errors corrected per bit. Hila5's XE5 is simple and constant-time but also pretty inefficient: it uses 48 payload bits per error corrected, compared to 9-10 for a comparable BCH code. This would make it unsuitable for HQC or even ThreeBears.
- * Flexible, so they can be configured to correct many errors or only a few. ThreeBears's Melas BCH code fails at this: it is simple and efficient, but as designed it can correct only 2 errors.
- * Ideally, support soft decoding or at least mixed error/erasure decoding. I expect that this would significantly improve performance for LWE schemes, probably superseding both ThreeBears' and Hila5's ECC mechanisms, and would probably also improve performance in a tensor application like HQC.

Would a binary Goppa code with fixed parameters (as used in a code-based system, but public) meet these desiderata better than BCH?

Cheers,
— Mike Hamburg

 **D. J. Bernstein**
to pqc-...@list.nist.gov
Jan 1, 2018, 3:48:11 AM

Mike Hamburg writes:
> Would a binary Goppa code with fixed parameters (as used in a
> code-based system, but public) meet these desiderata better than BCH?

No.

A BCH code is the following special case of a (classical) binary Goppa code: take the g polynomial as a power of x . Compared to the case of a random g , the BCH case allows the same error-correction methods, including soft-decision decoding, and it has slightly better code rate for the same number of errors, so it fits your desiderata better.

See <https://cr.yp.to/talks.html#2012.09.24> for discussion of BCH codes in the context of code-based crypto.

--Dan

 **Gaborit**
to pqc-...@list.nist.gov
Jan 1, 2018, 5:25:51 AM

hi,

that is a good question , mike, what I can say about HQC is that

we did not try to propose such an option in our basic implementation.

Now for HQC because of the repetition code, the BCH code is not too long (500-1000)

and because of that in our performances, the BCH decoding part

is less than 10% of the total cost of decryption, which I suppose, gives

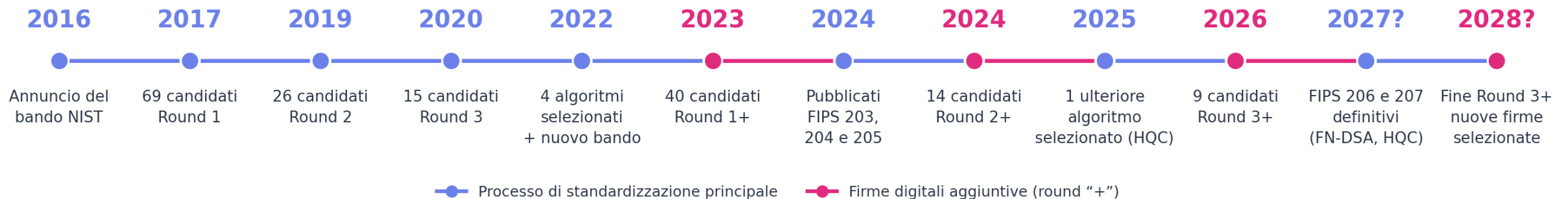
room to find an adapted algorithm in constant time.

best,

philippe

Il Progetto NIST PQcrypto (3)

- Il primo processo NIST per la standardizzazione di crittografia post-quantum si sta concludendo.
- Nel frattempo, nel 2022 il NIST ha aperto una **seconda competizione** per la standardizzazione di ulteriori sistemi di **firma digitale** post-quantum, che è ancora in corso.



PQC standard NIST



Schema (parametri liv. 1)	Standard e stato	Famiglia	Funzione	Chiave pubblica	Firma / cifrato	Tratto distintivo
ML-KEM-512	FIPS 203 finale (ago 2024)	Reticoli (Module-LWE)	KEM	800 B	768 B	Scelta primaria per lo scambio di chiavi; veloce, versione ML-KEM-768 già in TLS 1.3
ML-DSA-44 (liv. 2)	FIPS 204 finale (ago 2024)	Reticoli (Module-LWE/SIS)	Firma	1 312 B	2 420 B	Firma primaria; aritmetica intera, facile da rendere constant-time
SLH-DSA-SHA2-128s	FIPS 205 finale (ago 2024)	Funzioni hash	Firma	32 B	7 856 B (128f: 17 088 B)	Riserva conservativa; chiavi minuscole, firme enormi e firma lenta
FN-DSA-512	FIPS 206 bozza (finale 2026/27)	Reticoli NTRU	Firma	897 B	≈ 666 B	Le più compatte del portafoglio; campionamento in virgola mobile delicato
HQC-128	Selezionato mar 2025 bozza FIPS 2026, finale 2027	Codici (QCSD)	KEM	2 241 B	4 433 B	Riserva di ML-KEM su matematica diversa; ~3× la chiave, ~6× il cifrato

Competizione Addizionale per Firme Digitali

In [7], NIST indicated an intent to select at most one of **Dilithium** and **FALCON**, as both are based on structured lattices and could be used in most applications. Ultimately, however, NIST decided to select both schemes for standardization. As noted in Section 2.2.2, key and signature generation for FALCON appears to require more resources (gates and RAM) than Dilithium, which may make FALCON unsuitable for implementation on constrained devices, particularly in cases in which protection against side-channel attacks is required. In addition, NIST recognizes that the simpler design of Dilithium's key and signature generation will help ensure secure implementations. For these reasons, NIST selected **Dilithium as the primary signature algorithm** that it will recommend for general use and will prioritize its standardization.

NIST understands that some applications will not work as they are currently designed if the signature and the **data being signed cannot fit in a single internet packet**. For these applications, the **implementation complexity of FALCON's signature generation may not be a concern**, but the difficulty of modifying the applications to work with Dilithium's larger signature size may create a barrier to the transition to post-quantum signature schemes. For this reason, **NIST decided to standardize FALCON as well**. Given FALCON's overall better performance when signature generation does not need to be performed on constrained devices, many applications may prefer to use FALCON over Dilithium, even in cases in which Dilithium's signature size would not be a barrier to implementation.

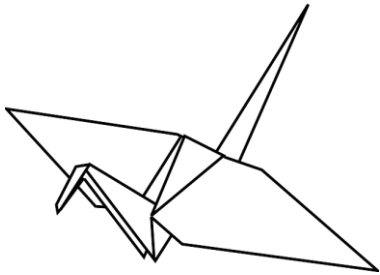
In order **to not rely entirely on the security of lattices**, NIST is also **standardizing SPHINCS+**. The security of SPHINCS+ is well-understood, although it is **much larger and slower than the lattice signatures**. SPHINCS+ is a mature scheme, and standardizing it creates a fallback option that helps minimize the risk that a single breakthrough in cryptanalysis would leave NIST without a viable signature. NIST recognizes that SPHINCS+ may not be suitable for many applications, given its performance profile. NIST made the choice to select SPHINCS+ now instead of perhaps including it in the fourth round. As such, this means the end of the current process for signature schemes. All signature candidates have either been selected for standardization or removed from consideration for standardization. NIST may standardize more signatures in the future (NIST plans to issue a new **Call for Proposals** for post-quantum signatures later in 2022), but this will take several years and there is no guarantee of better algorithms.

Competizione Addizionale per Firme Digitali (2)



Famiglia	Candidati al 1° round (17 luglio 2023): 40	2° round (24 ott. 2024): 14	3° round (14 mag. 2026): 9
Multivariata (10)	3WISE, DME-Sign, HPPC, MAYO, PROV, QR-UOV, SNOVA, TUOV, UOV, VOX	MAYO, QR-UOV, SNOVA, UOV	MAYO, QR-UOV, SNOVA, UOV
MPC-in-the-Head (7)	Biscuit, MIRA, MiRitH, MQOM, PERK, RYDE, SDitH	Mirath (fusione MIRA+MiRitH), MQOM, PERK, RYDE, SDitH	MQOM, SDitH
Reticoli (7)	EagleSign, EHTv3/v4, HAETAE, HAWK, HuFu, Raccoon, SQUIRRELS	HAWK	HAWK (ritirato dagli autori il 29 lug. 2026)
Codici (6)	CROSS, Enhanced pqsigRM, FuLeeca, LESS, MEDS, Wave	CROSS, LESS	—
Simmetrica / hash (4)	AIMer, Ascon-Sign, FAEST, SPHINCS-alpha	FAEST	FAEST
Isogenie (1)	SQIsign	SQIsign	SQIsign
Altro (5)	ALTEQ, eMLE-Sig 2.0, KAZ-SIGN, Preon, Xifrat1-Sign.I	—	—

LEDACrypt, LESS, CROSS e LEAST

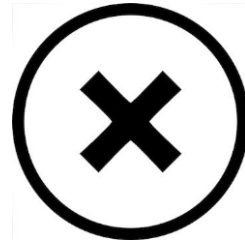


LEDACrypt

Low-density parity-check code-based cryptographic systems

www.ledacrypt.org

- Key Encapsulation Mechanism (KEM) e Public Key Cryptosystem (PKC).
- Basato su McEliece/Niederreiter con codici LDPC/MDPC.
- Non standardizzato, ma versione 3 sicura, compatta e veloce.

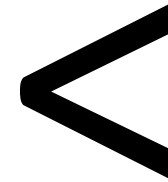


CROSS

Code and Restricted Object Signature Scheme

www.cross-crypto.com

- Schema di firma digitale basato sul problema di decodifica ristretta.
- Uno dei due basati su codici ammessi al secondo round della competizione addizionale NIST.
- Molto veloce e con firme ragionevolmente compatte.

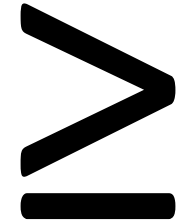


LESS

Linear Equivalence Signature Scheme

www.less-project.com

- Schema di firma digitale basato sul problema dell'equivalenza tra codici.
- Uno dei due basati su codici ammessi al secondo round della competizione addizionale NIST.
- Ragionevolmente veloce e con firme molto compatte.



LEAST

Linear Equivalence Action Threshold Signature

www.least-project.org

- Schema di firma digitale post-quantum per firme a soglia.
- Basato sul problema dell'equivalenza tra codici.
- Deriva da LESS ma supporta firme a soglia.
- Pronto per la prossima competizione NIST.

Transizione alla PQC

- La definizione di **algoritmi standard** è solo il primo passo della transizione.
- Ci servono **implementazioni** software e hardware che siano sicure e efficienti.
- Esse devono poi essere integrate nei sistemi e protocolli esistenti per la sicurezza dei dati e delle reti, considerando anche gli scenari di attacco "**harvest-now-decrypt-later**".

Algorithm	Private key (byte)	Public key (byte)	Signature (byte)
ML-DSA	2 560	1 312	2 420
FN-DSA	1 281	897	666
SLH-DSA	64	32	17 088
RSA-3072	384	384	384
ECDSA-256	32	64	64

Vulnerabilità quantum

Transizione alla PQC

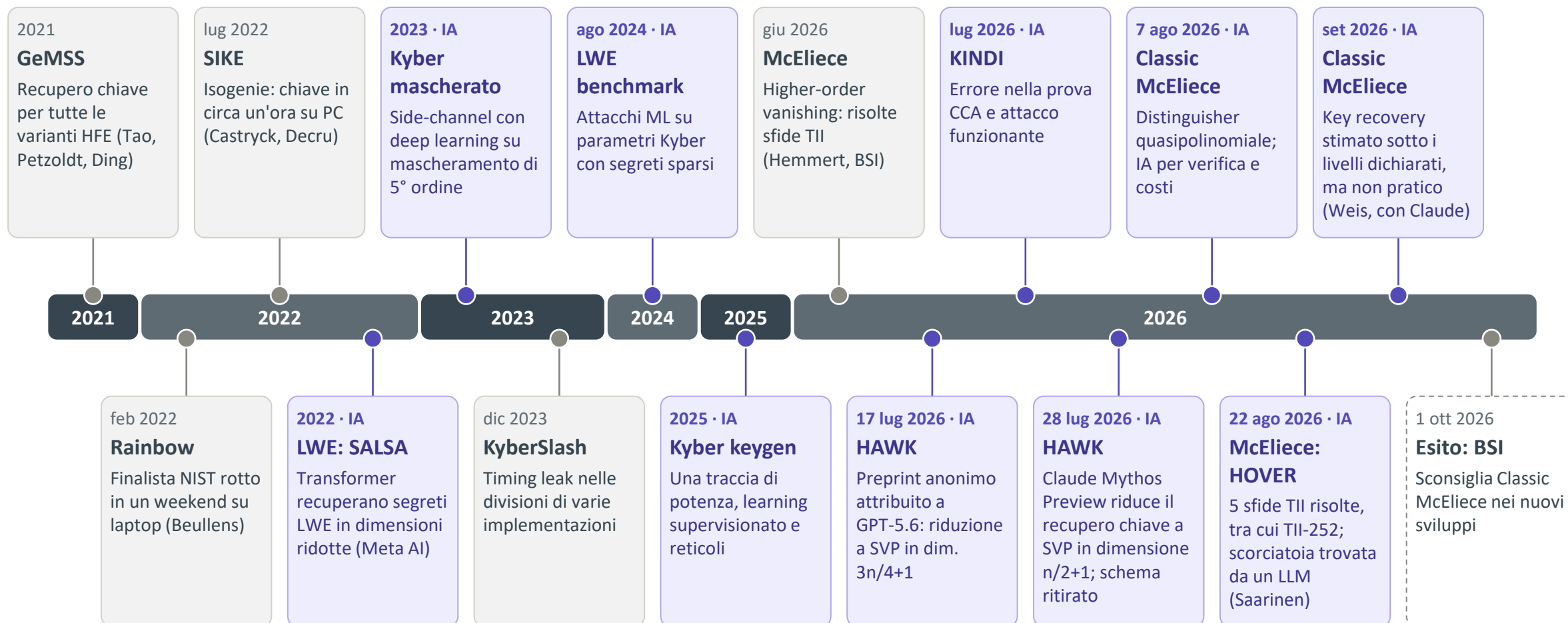
Sicurezza quantum



Crittanalisi, una storia che non finisce



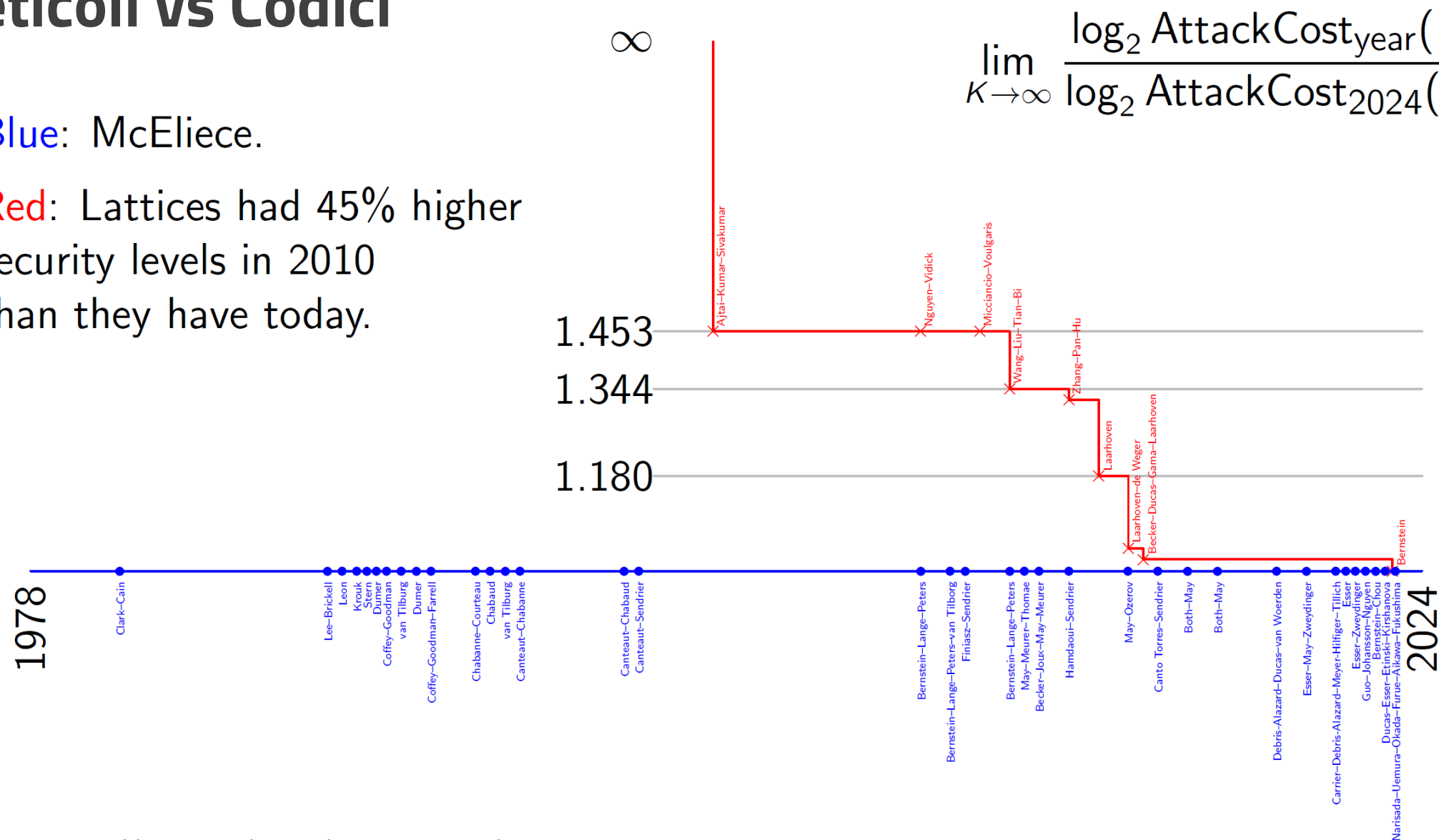
● Senza IA
● Con IA



Reticoli vs Codici

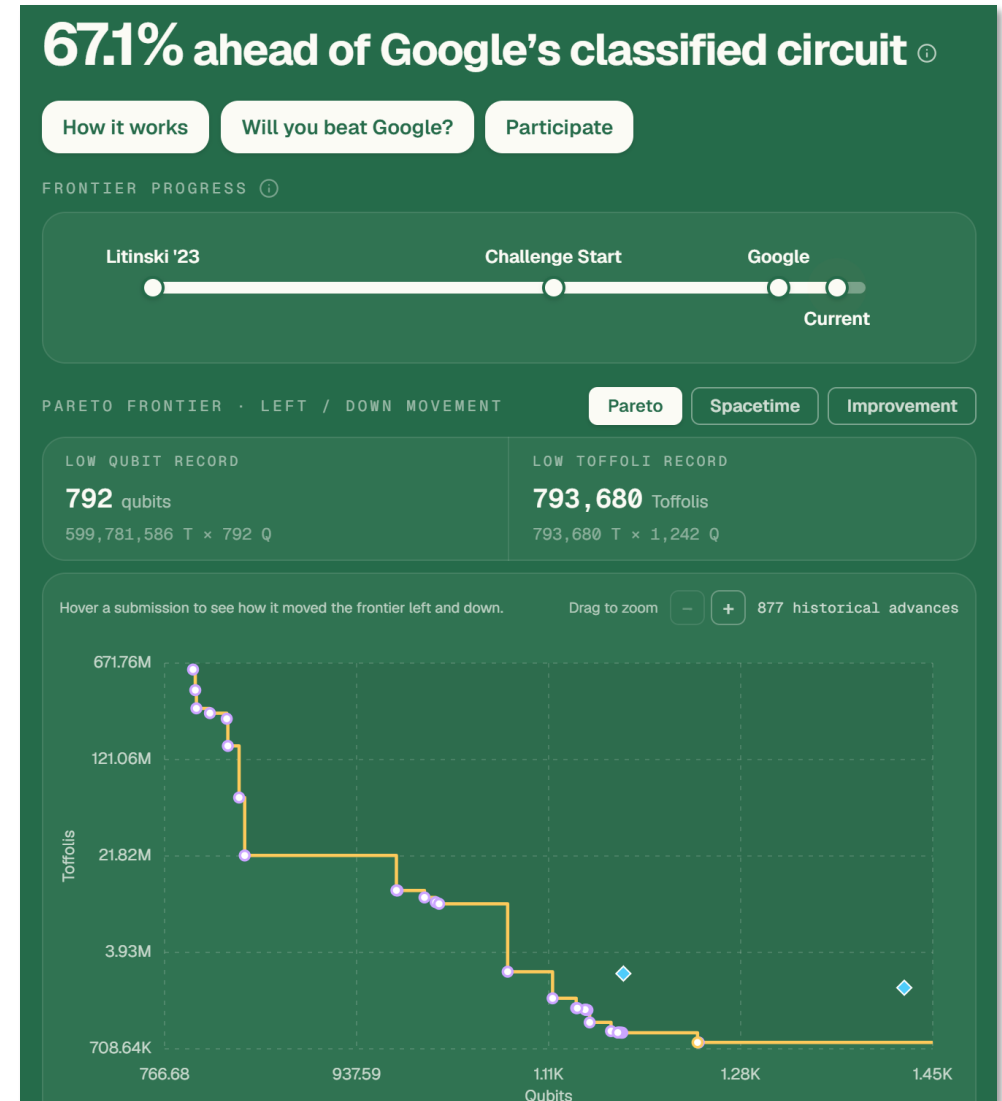
Blue: McEliece.

Red: Lattices had 45% higher security levels in 2010 than they have today.



Crittanalisi Quantum: ECDSA.fail

- Gara pubblica tramite cui chiunque può provare a ottimizzare un **circuito quantistico** che rompe **ECDSA**.
- Molti hanno iniziato a farlo con l'aiuto dell'**intelligenza artificiale**.
- Nel giro di poche settimane i contributi della comunità hanno ridotto di circa il **40%** il circuito segreto annunciato da Google a marzo 2026.
- Ad oggi si è raggiunta una riduzione di oltre il **67%** rispetto al circuito di Google.



Il progetto QSAFEIT



- Progetto di ricerca avviato a ottobre 2024, finanziato dal **Ministero dell'Università e della Ricerca** con circa un milione di euro su quattro anni.
- Obiettivo: accrescere la competenza nazionale nella crittografia post-quantum, con particolare attenzione ai sistemi basati sui codici.
- Sfide affrontate:
 - progettare nuovi schemi per scambio di chiavi e firme,
 - individuare nuovi attacchi per renderli più robusti,
 - realizzare implementazioni compatte per dispositivi con risorse limitate,
 - sperimentare la PQC in comunicazioni reali, satellitari e terrestri.



<https://qsafeit.dii.univpm.it/>