An Update on Post-Quantum Cryptography and Standardisation

Newcastle Post-Quantum Security Workshop

Martin R. Albrecht

14 September 2023

Outline

Post-Quantum Era

Post-Quantum Standardisation

Post-Quantum Security

Post-Quantum Hedging

Post-Quantum PETS

Post-Quantum Era

Quantum Computers

- A quantum computer makes use of quantum effects (superpositions and entanglement) to perform computations.
- Quantum computers are not **faster** than classical computers, they are **different**.
- Some computations are easy on a quantum computer that are – as far as we know – hard on a classical computer.

- Small universal quantum computers exist.
- Key challenge is to scale them up by making them more stable.
- There is a critical point where we can scale up further using error correction.

FINANCIAL TIMES

HOME WORLD UK COMPANIES TECH MARKETS GRAPHICS OPINION WORK & CAREERS LIFE & ARTS HOW TO SPEND IT

Latest on Quantum technologies

A quantum future could be bright if managed well

Rivals rubbish Google's claim of quantum supremacy

Quantum computing builds on Turing's legacy

Quantum technologies + Add to myFT

Google claims to have reached quantum supremacy

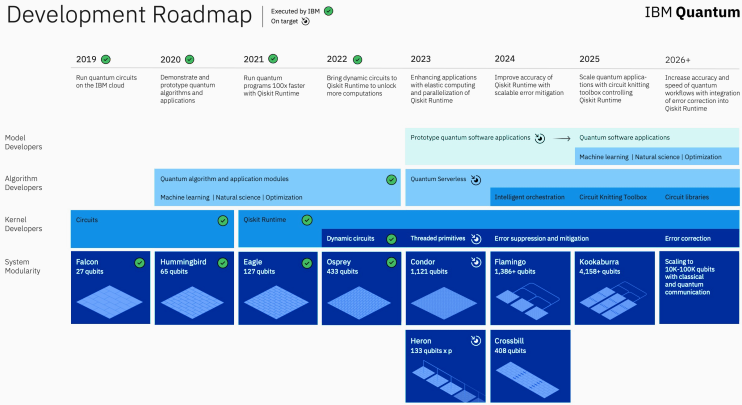
Researchers say their quantum computer has calculated an impossible problem for ordinary machines

Madhumita Murgia and Richard Waters SEPTEMBER 20 2019

162

Google claims to have built the first quantum computer that can carry out calculations beyond the ability of today's most powerful supercomputers, a landmark moment that has been hotly anticipated by researchers.

IBM Quantum Computing Timeline



https://www.ibm.com/quantum/roadmap

- NIST PQC Standards are expected some time before end of 2024
- Other standardisation bodies and authorities will follow suit[1]

⇒ Post-quantum cryptography is coming regardless of quantum computers

Do it right: We are ripping out the plumbing, might as well do it right (protocols, formal assurances, etc)

[1] "NCSC guidance for quantum-safe algorithms will follow the outcome of the NIST process by recommending specific algorithms for representative use cases." — NCSC: Preparing for Quantum-Safe Cryptography

Essential Cryptographic Primitives

Symmetric Primitives

- Block and stream ciphers (AES, ChaCha20, ...)
- Authentication codes (HMAC, Poly1305, ...)
- Hash functions (SHA-2, SHA-3, ...)

Asymmetric Primitives

- Key agreement and public-key encryption (RSA, Diffie-Hellman, ECDH, ...)
- Digital signatures (RSA, DSA, ECDSA, ...)

Applications

TLS, SSH, banking, smart cards, hard disk encryption . . .

Essential Cryptographic Primitives: Theoretical Perspective

Minicrypt

- Block and stream ciphers
- Hash functions
- Authentication codes
- **Digital signatures**

Cryptomania

- Key agreement and public-key encryption
- ...

Essential Cryptographic Primitives: Theoretical Perspective

Minicrypt

- Block and stream ciphers
- Hash functions
- Authentication codes
- **Digital signatures**

Cryptomania

- Key agreement and public-key encryption
- ...

Very slow one-time digital signatures from hash functions

- **KeyGen** $H(\cdot)$ is a hash function with 256 bits of output. Sample random numbers $(a_{0,0}, a_{0,1}), (a_{1,0}, a_{1,1}), \ldots, (a_{255,0}, a_{255,1})$. Publish $H(a_{i,j})$ for all $a_{i,j}$.
- **Sign** Let b_i be the bits of $H(m)$. For each bit b_i, publish a_{i,b_i}.
- **Verify** Check that a_{i,b_i} indeed hashes to $H(a_{i,j})$ in the public key.

The Poverty of Public-Key Cryptography

Symmetric Primitives

Indeed, it seems that "you can't throw a rock without hitting a one-way function" in the sense that, once you cobble together a large number of simple computational operations then, unless the operations satisfy some special property such as linearity, you will typically get a function that is hard to invert.[a]

[a] Boaz Barak. The Complexity of Public-Key Cryptography. Cryptology ePrint Archive, Report 2017/365. https://eprint.iacr.org/2017/365. 2017.

Asymmetric Primitives

All widely deployed asymmetric cryptography relies on the hardness of **factoring:**

Given $N = p \cdot q$ find p, or

(elliptic-curve) discrete logarithms:

Given $g^a \bmod q$ and g find a.

Symmetric Primitives: Quantum Computing Perspective (Good News)

Best known quantum algorithms for attacking symmetric cryptography are based on Grover's algorithm.

- Search key space of size 2^n in $2^{n/2}$ operations: AES-256 $\rightarrow$ 128 "quantum bits of security".
- Taking all costs into account: $> 2^{152}$ classical operations for AES-256.[2]
- Assuming a max depth of 2^{96} for a quantum circuit: overall AES-256 cost is $\approx 2^{190}$.
- Does not parallelise: have to wait for 2^X steps, cannot buy 2^{32} quantum computers and wait $2^X/2^{32}$ steps.

[2] Samuel Jaques, Michael Naehrig, Martin Roetteler and Fernando Virdia. Implementing Grover Oracles for Quantum Key Search on AES and LowMC. In: *EUROCRYPT 2020, Part II*. ed. by Anne Canteaut and Yuval Ishai. Vol. 12106. LNCS. Springer, Heidelberg, May 2020, pp. 280–310. DOI: 10.1007/978-3-030-45724-2_10.

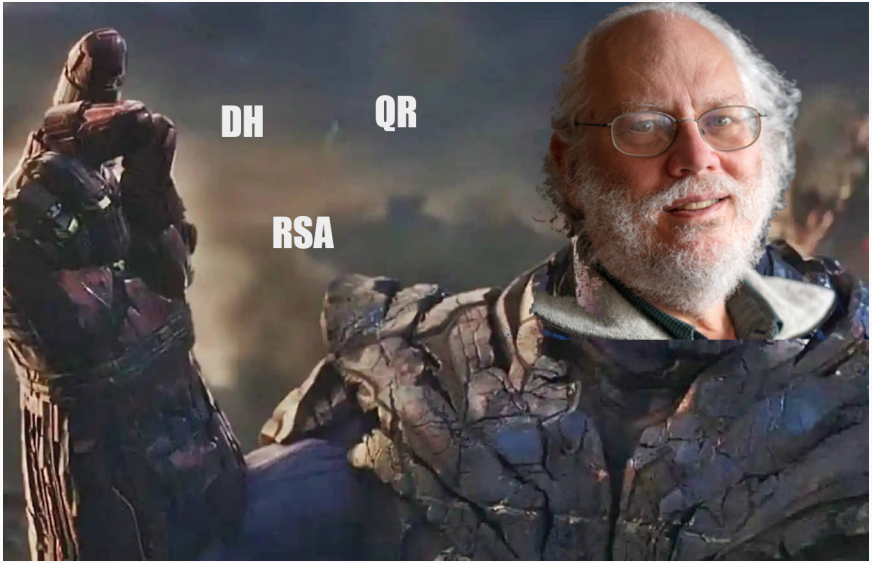
Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*

Peter W. Shor†

Abstract

A digital computer is generally believed to be an efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time by at most a polynomial factor. This may not be true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and which have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored.

Asymmetric Primitives: Quantum Computing Perspective



Post-Quantum Standardisation

Post-Quantum Standardisation

NIST **Post Quantum ~~Competition~~ Process**

ETSI Cyber Working Group for Quantum Safe Cryptography

ISO WG2 Standing Document 8 (SD8): Survey

IETF Standardisation of **stateful** hash-based signatures, nothing further

CSA Quantum-safe Security Working Group: position papers

NIST **Post Quantum Process: Digital Signatures**

Post-Quantum Standardisation

NIST **Post Quantum ~~Competition~~ Process**

ETSI Cyber Working Group for Quantum Safe Cryptography

ISO WG2 Standing Document 8 (SD8): Survey

IETF Standardisation of **stateful** hash-based signatures, nothing further

CSA Quantum-safe Security Working Group: position papers

NIST **Post Quantum Process: Digital Signatures**

Bottom Line

Essentially, everyone is/was waiting for NIST.

NIST PQC ~~Competition~~ Process

Timeline

Submission	November 2017
Round 2 Selection	January 2019
Round 3 Selection	July 2020
Winners and Round 4 Selection	July 2022
Final Standard Expectation	by 2024

“Key Establishment”/Key Encapsulation

- `(pk,sk) ← KeyGen()`
- `(c,k) ← Encaps(pk)`
- `k ← Decaps(c,sk)`

Digital Signature

- `(vk,sk) ← KeyGen()`
- `s ← Sig(m,sk)`
- `{0,1} ← Verify(s,m,vk)`

KEM: Security

What you get: "IND-CCA"

- We give the adversary either the real k or a random fake one.
- The adversary is allowed to ask for decryptions of any ciphertext but c
- The adversary wins if it guesses correctly which key we gave it

This implies the adversary cannot learn anything about an encrypted message (except its length) even when being allowed to decrypt anything else.

What you *do not* get

- Given a ciphertext it is unclear who it was encrypted too
  - Ciphertexts might reveal what keys can decrypt them
- If you can decrypt, only you can decrypt
  - It might be possible construct a ciphertext that decrypts correctly under two or more decryption keys
- ...

KEM: Kyber

RSA 2048

Key generation	≈ 130,000,000 cycles
Encapsulation	≈ 20,000 cycles
Decapsulation	≈ 2,700,000 cycles
Ciphertext	256 bytes
Public key	256 bytes

https://bench.cr.yp.to/results-kem.html

Curve25519

Key generation	≈ 60,000 cycles
Key agreement	≈ 160,000 cycles
Public key	32 bytes
Key Share	32 bytes

https://eprint.iacr.org/2015/943

Kyber-768

Key generation	≈ 38,000 cycles
Encapsulation	≈ 49,000 cycles
Decapsulation	≈ 39,000 cycles
Ciphertext	1,088 bytes
Public key	1,184 bytes

https://bench.cr.yp.to/results-kem.html

Lattice-based KEM: Learning with Errors

"KeyGen:"

```
A = random_matrix(GF(7681), 3*256, 3*256)
s = random_vector(ZZ, 3*256, x=-4, y=5)
```

"Encrypt:"

```
e = random_vector(ZZ, 3*256, x=-4, y=5) # this makes it hard!
m = random_vector(GF(2), 3*256).lift()
b = A*s + e + 7681//2 * m # encrypt
```

"Decrypt:"

```
r = (b - A*s).lift_centered() # this is == e + 7681//2 * m
vector(ZZ, [round(float(r_)/(7681//2)) for r_ in r]) == m # round and check
```

```
True
```

SIG: Security

What you get: "EUF-CMA"

- An adversary is allowed to ask us to sign any message it wants, as often as it likes
- The adversary wins if it then outputs a valid signature for a message it has not asked us for a signature before
- A valid signature is a signature that checks out given the verification key.

What you *do not* get

- Given a signature and it verifies under a given verification key then it was signed by the matching sender
  - There might be more than one verification key under which a signature validates
- Given a signature and message pair, there is only this one message for a given signature.
  - The same signature might be valid for multiple messages.
- ...

SIG: Lattic-based (Falcon, Dilithium) or Hash-based (SPHINCS+)

Scheme	PK	Sig	Verif	Sign
NIST P-256	64	64	1 (baseline)	1 (baseline)
RSA-2048	256	256	0.2	25
Dilithium2	1,320	2,420	0.3	2.5
Falcon-512	897	666	0.3	5
Falcon-512 FPEMU	897	666	0.3	100
SPHINCS+-128ss har.	32	7,856	1.7	3,000

https://blog.cloudflare.com/sizing-up-post-quantum-signatures/

Lattice-based SIG: Short Integer Solutions

Easy:

```
q = next_prime(2^13)
A = random_matrix(GF(q), 1024, 2048)
u = random_vector(ZZ, 2048, x=-ceil(sqrt(q)), y=ceil(sqrt(q)))
t = A*u # easy
assert max(u) < q//4
```

Hard:

```
v = A.solve_right(t).lift_centered()
assert A*v == t
max(v) < q//4
```

```
False
```

Post-Quantum Security

Security Notions

KEM **IND-CCA**: Given some challenge ciphertext $\mathbf{c}$ and some key $\mathbf{k}$, the adversary gets an oracle to decapsulate (“decrypt”) any other ciphertext but still cannot decide if $\mathbf{c}$ encapsulates (“encrypts”) the key $\mathbf{k}$.

SIG **EUF-CMA**: Given access to some oracle that signs arbitrary messages, the adversary still cannot produce a valid signature of a message not previously submitted to the signing oracle.

Security Notions

KEM **IND-CCA**: Given some challenge ciphertext c and some key k, the adversary gets an oracle to decapsulate ("decrypt") any other ciphertext but still cannot decide if c encapsulates ("encrypts") the key k.

SIG **EUF-CMA**: Given access to some oracle that signs arbitrary messages, the adversary still cannot produce a valid signature of a message not previously submitted to the signing oracle.

Computational Security

"cannot" $\rightarrow$ "it takes too long even given access to a quantum computer."

Security Notions

KEM **IND-CCA**: Given some challenge ciphertext $\mathbf{c}$ and some key $\mathbf{k}$, the adversary gets an oracle to decapsulate ("decrypt") any other ciphertext but still cannot decide if $\mathbf{c}$ encapsulates ("encrypts") the key $\mathbf{k}$.

SIG **EUF-CMA**: Given access to some oracle that signs arbitrary messages, the adversary still cannot produce a valid signature of a message not previously submitted to the signing oracle.

Computational Security

"cannot" $\rightarrow$ "it takes too long even given access to a quantum computer."

Conditional Security

"cannot" $\rightarrow$ "... assuming some mathematical problem is hard on a quantum computer"

SIKE Attack

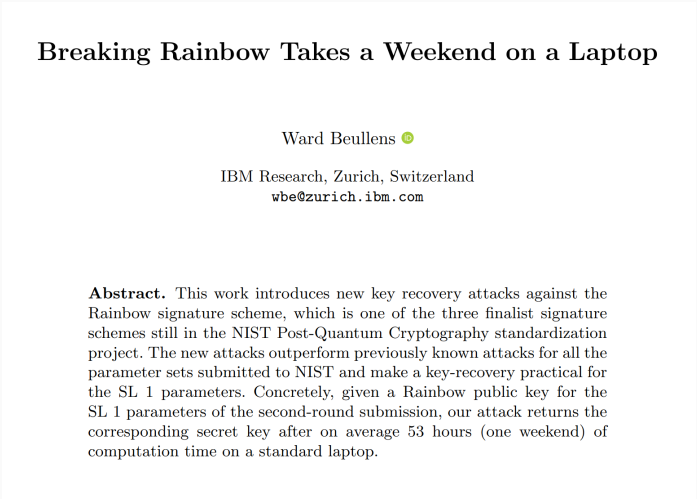
Wouter Castryck and Thomas Decru. **An efficient key recovery attack on SIDH (preliminary version).** Cryptology ePrint Archive, Report 2022/975. `https://eprint.iacr.org/2022/975`. 2022

- SIDH was “A decade unscathed” [Cos21]
- SIKE even *lowered* parameters during NIST PQC (following [JS19])
- qualified researchers tried to break it (e.g. [MP19])

Total Break

All SIKE parameters can be broken in about 2 hours on a single-core laptop now [OP22].

Rainbow Attack



Breaking Rainbow Takes a Weekend on a Laptop

Ward Beullens

IBM Research, Zurich, Switzerland
wbe@zurich.ibm.com

Abstract. This work introduces new key recovery attacks against the Rainbow signature scheme, which is one of the three finalist signature schemes still in the NIST Post-Quantum Cryptography standardization project. The new attacks outperform previously known attacks for all the parameter sets submitted to NIST and make a key-recovery practical for the SL 1 parameters. Concretely, given a Rainbow public key for the SL 1 parameters of the second-round submission, our attack returns the corresponding secret key after on average 53 hours (one weekend) of computation time on a standard laptop.

- Rainbow was a NIST finalist [Beu22]
- Can be remedied by increasing parameters
- MQ signatures have a shaky history
- NIST is specifically looking to standardise UOV, a long-standing MQ signature scheme

MATZOV Attack

Report on the Security of LWE: Improved Dual Lattice Attack

The Center of Encryption and Information Security – MATZOV*†
IDF

Abstract

Many of the leading post-quantum key exchange and signature schemes rely on the conjectured hardness of the Learning With Errors (LWE) and Learning With Rounding (LWR) problems and their algebraic variants, including 3 of the 6 finalists in NIST's PQC process. The best known cryptanalysis techniques against these problems are primal and dual lattice attacks, where dual attacks are generally considered less practical.

In this report, we present several algorithmic improvements to the dual lattice attack, which allow it to exceed the efficiency of primal attacks. In the improved attack, we enumerate over more coordinates of the secret and use an improved distinguisher based on FFT. In addition, we incorporate improvements to the estimates of the cost of performing a lattice sieve in the RAM model, reducing the gate-count of random product code decoding and performing less inner product calculations.

Combining these improvements considerably reduces the security levels of Kyber, Saber and Dilithium, the LWE/LWR based finalists, bringing them below the thresholds defined by NIST.

- Made some waves, partly due to the authorship
- Ingredients:
  - Improvement in a lower-order "sieving" term
  - Generalisation of a technique from [GJ21]
- Precise impact a bit unclear[a]

[a] Léo Ducas and Ludo Pulles. **Does the Dual-Sieve Attack on Learning with Errors even Work?** Cryptology ePrint Archive, Report 2023/302. `https://eprint.iacr.org/2023/302`. 2023.

Lattices

$\frac{3}{4}$ selected NIST algorithms are based on structured lattices

- We have good evidence that lattice problems are hard asymptotically.
- We have a relatively good understanding of how know algorithms behave concretely.
  - Our estimates are conservative, ignoring e.g. the cost of memory access.
- Quantum computers seem to not help for lattices in any meaningful way.
- No known algorithms that perform better on structured lattices than on unstructured lattices.
  - Biggest potential for improvements here!

Post-Quantum Hedging

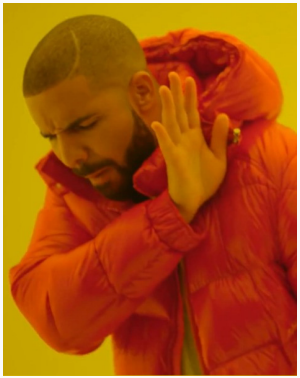
NIST Round 4

- BIKE
- Classic McEliece
- HQC
- SIKE[a]

[a] See above.

”Both BIKE and HQC are based on structured codes, and either would be suitable as a general-purpose KEM that is not based on lattices. NIST expects to select at most one of these two candidates for standardization at the conclusion of the fourth round. (. . .)
Classic McEliece was a finalist but is not being standardized by NIST at this time. Although Classic McEliece is widely regarded as secure, NIST does not anticipate it being widely used due to its large public key size. NIST may choose to standardize Classic McEliece at the end of the fourth round.”

Hybrids



PQ MIGRATION

CYBERSECURITY

GOALS

- Crypto agility
  - Rigorous, effective algorithm vetting is a must, NSA has confidence in the NIST PQC process
- NSA will not require a hybrid design for security purposes
- NSA only anticipates using hybrid solutions to maintain interoperability during the transition (or where direct drop-in is not feasible)
  - Any hybrid method adopted should allow for a quick transition to PQ-only solutions
- Ensure interoperability with PQ-only systems is included for forward compatibility and to allow for use of direct drop-in of PQ

https://datatracker.ietf.org/meeting/112/materials/slides-112-lamps-hybrid-non-composite-multi-certificate-00

Other Agencies

BSI (Germany) and ANSSI (France) recommend hybrid encryption.

NIST Digital Signatures

"NIST also plans to issue a new Call for Proposals for public-key (quantum-resistant) digital signature algorithms by the end of summer 2022. NIST is primarily looking to diversify its signature portfolio, so signature schemes that are ***not based on structured lattices*** *are of greatest interest. NIST would like submissions for signature schemes that have short signatures and fast verification (e.g.,* ***UOV****)."*
— Dustin Moody (NIST) on PQC mailinglist, my emphasis

NIST Digital Signature Submissions

- Lattices
- Codes
- MPC-in-the-Head
- Multivariate
- Isogenies
- Symmetric
- Other

40 Submissions

3WISE, AIMer, ALTEQ, Ascon-Sign, Biscuit, CROSS, DME-Sign, EHT, EagleSign, Enhanced pqsigRM, FAEST, FuLeeca, HAETAE, HAWK, HPPC, HuFu, KAZ-SIGN, LESS, MAYO, MEDS, MIRA, MQOM, MiRitH, PERK, PROV, Preon, QR-UOV, RYDE, Raccoon, SDitH, SNOVA, SPHINCS-alpha, SQIsign, SQUIRRELS, TUOV, UOV, VOX, Wave, Xifrat1-Sign.I, eMLE-Sig 2.0

NIST Digital Signature Submissions

- Lattices
- Codes
- MPC-in-the-Head
- Multivariate
- Isogenies
- Symmetric
- Other

40 Submissions

3WISE, AIMer, ALTEQ, Ascon-Sign, Biscuit, CROSS, DME-Sign, EHT, EagleSign, Enhanced pqsigRM, FAEST, FuLeeca, HAETAE, HAWK, HPPC, HuFu, KAZ-SIGN, LESS, MAYO, MEDS, MIRA, MQOM, MiRitH, PERK, PROV, Preon, QR-UOV, RYDE, Raccoon, SDitH, SNOVA, SPHINCS-alpha, SQIsign, SQUIRRELS, TUOV, UOV, VOX, Wave, Xifrat1-Sign.I, eMLE-Sig 2.0

NIST Digital Signature Submissions

- Lattices
- Codes
- MPC-in-the-Head
- Multivariate
- Isogenies
- Symmetric
- Other

40 Submissions

3WISE, AIMer, ALTEQ, Ascon-Sign, Biscuit, CROSS, DME-Sign, EHT, EagleSign, Enhanced pqsigRM, FAEST, FuLeeca, HAETAE, HAWK, HPPC, HuFu, KAZ-SIGN, LESS, MAYO, MEDS, MIRA, MQOM, MiRitH, PERK, PROV, Preon, QR-UOV, RYDE, Raccoon, SDitH, SNOVA, SPHINCS-alpha, SQIsign, SQUIRRELS, TUOV, UOV, VOX, Wave, Xifrat1-Sign.I, eMLE-Sig 2.0

NIST Digital Signature Submissions

- Lattices
- Codes
- MPC-in-the-Head
- Multivariate
- Isogenies
- Symmetric
- Other

40 Submissions

3WISE, AIMer, ALTEQ, Ascon-Sign, Biscuit, CROSS, DME-Sign, EHT, EagleSign, Enhanced pqsigRM, FAEST, FuLeeca, HAETAE, HAWK, HPPC, HuFu, KAZ-SIGN, LESS, MAYO, MEDS, MIRA, MQOM, MiRitH, PERK, PROV, Preon, QR-UOV, RYDE, Raccoon, SDitH, SNOVA, SPHINCS-alpha, SQIsign, SQUIRRELS, TUOV, UOV, VOX, Wave, Xifrat1-Sign.I, eMLE-Sig 2.0

NIST Digital Signature Submissions

- Lattices
- Codes
- MPC-in-the-Head
- Multivariate
- Isogenies
- Symmetric
- Other

40 Submissions

3WISE, AIMer, ALTEQ, Ascon-Sign, Biscuit, CROSS, DME-Sign, EHT, EagleSign, Enhanced pqsigRM, FAEST, FuLeeca, HAETAE, HAWK, HPPC, HuFu, KAZ-SIGN, LESS, MAYO, MEDS, MIRA, MQOM, MiRitH, PERK, PROV, Preon, QR-UOV, RYDE, Raccoon, SDitH, SNOVA, SPHINCS-alpha, SQIsign, SQUIRRELS, TUOV, UOV, VOX, Wave, Xifrat1-Sign.I, eMLE-Sig 2.0

NIST Digital Signature Submissions

- Lattices
- Codes
- MPC-in-the-Head
- Multivariate
- Isogenies
- Symmetric
- Other

40 Submissions

3WISE, AIMer, ALTEQ, Ascon-Sign, Biscuit, CROSS, DME-Sign, EHT, EagleSign, Enhanced pqsigRM, FAEST, FuLeeca, HAETAE, HAWK, HPPC, HuFu, KAZ-SIGN, LESS, MAYO, MEDS, MIRA, MQOM, MiRitH, PERK, PROV, Preon, QR-UOV, RYDE, Raccoon, SDitH, SNOVA, SPHINCS-alpha, SQIsign, SQUIRRELS, TUOV, UOV, VOX, Wave, Xifrat1-Sign.I, eMLE-Sig 2.0

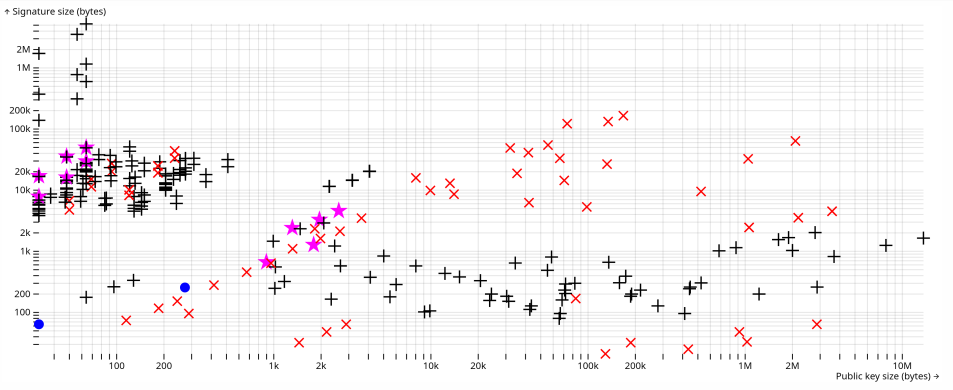
NIST Digital Signature Submissions

- Lattices
- Codes
- MPC-in-the-Head
- Multivariate
- Isogenies
- Symmetric
- Other

40 Submissions

3WISE, AIMer, ALTEQ, Ascon-Sign, Biscuit, CROSS, DME-Sign, EHT, EagleSign, Enhanced pqsigRM, FAEST, FuLeeca, HAETAE, HAWK, HPPC, HuFu, KAZ-SIGN, LESS, MAYO, MEDS, MIRA, MQOM, MiRitH, PERK, PROV, Preon, QR-UOV, RYDE, Raccoon, SDitH, SNOVA, SPHINCS-alpha, SQIsign, SQUIRRELS, TUOV, UOV, VOX, Wave, Xifrat1-Sign.I, eMLE-Sig 2.0

Performance



source: https://pqshield.github.io/nist-sigs-zoo/wide.html

Vulnerabilities in Specification

3WISE,[3] AIMer, ALTEQ, Ascon-Sign, Biscuit,[4] CROSS, DME-Sign,[5] EHT,[6] EagleSign,[7] Enhanced pqsigRM,[8] FAEST, FuLeeca,[9] HAETAE, HAWK, HPPC,[10] HuFu,[11] KAZ-SIGN,[12] LESS,[13] MAYO, MEDS,[14] MIRA, MQOM, MiRitH, PERK, PROV, Preon, QR-UOV, RYDE, Raccoon, SDitH,[15] SNOVA, SPHINCS-alpha, SQIsign, SQUIRRELS, TUOV, UOV, VOX, Wave, Xifrat1-Sign.I,[16] eMLE-Sig 2.0

[3] https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/fsfGqHCgGvY
[4] https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/sw8NueiNek0
[5] https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/E0mMMGI5eWE
[6] https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/mFl_5Rq6-RU
[7] https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/zas5PLiBe6A
[8] https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/yQ1CKOLbGng
[9] https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/KvIege2EbuM
[10] https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/KRh8w03PW4E
[11] https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/Hq-wRFDbIaU
[12] https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/aCbi4BMDeUs
[13] https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/Z36SPZJI8Ok
[14] https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/CtCe8WXUoXI
[15] https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/d_BcUfFGl5o
[16] https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/9FXtBZKWueA

Interpretation

This is to be expected.

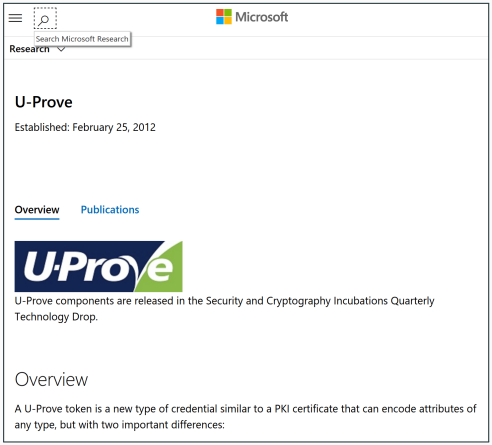
QKD?

"Given the specialised hardware requirements of QKD over classical cryptographic key agreement mechanisms and the requirement for authentication in all use cases, the NCSC does not endorse the use of QKD for any government or military applications, and cautions against sole reliance on QKD for business-critical networks, especially in Critical National Infrastructure sectors. [...] NCSC advice is that the best mitigation against the threat of quantum computers is quantum-safe cryptography."[17]

[17] https://www.ncsc.gov.uk/whitepaper/quantum-security-technologies

Post-Quantum PETS

Privacy-Preserving Computing



Pre-Quantum Applications:

- anonymous credentials (”I have an account with you and I am over 18”),
- central bank digital currency,
- privacy-preserving analytics (”Customers who liked . . .”)
- private contact discovery (”Alice is on WhatsApp”)

Privacy-Preserving Computing

Pre-Quantum Applications:

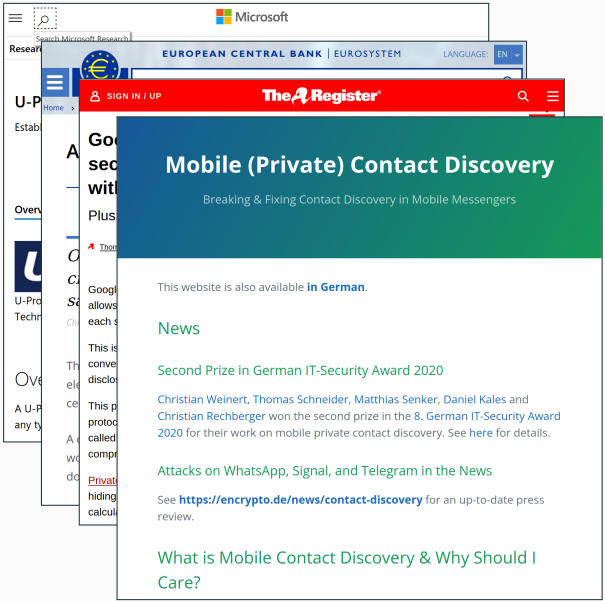
- anonymous credentials (”I have an account with you and I am over 18”),
- central bank digital currency,
- privacy-preserving analytics (”Customers who liked . . .”)
- private contact discovery (”Alice is on WhatsApp”)

Privacy-Preserving Computing

Pre-Quantum Applications:

- anonymous credentials (”I have an account with you and I am over 18”),
- central bank digital currency,
- privacy-preserving analytics (”Customers who liked . . .”)
- private contact discovery (”Alice is on WhatsApp”)

Privacy-Preserving Computing



Pre-Quantum Applications:

- anonymous credentials (”I have an account with you and I am over 18”),
- central bank digital currency,
- privacy-preserving analytics (”Customers who liked . . .”)
- private contact discovery (”Alice is on WhatsApp”)

VOPRF

(Verifiable) Oblivious Pseudorandom Functions allow two parties to compute a PRF $y = F_k(x)$ together, a server supplying k and a user supplying x. The server does not learn x or y, the user does not learn k.

- (V)OPRFs can be efficiently realised from the DH assumption and enable
  - anonymous credentials (e.g. Cloudflare's PrivacyPass),
  - Password-based Key Exchange (e.g. OPAQUE, in the process of IETF standardisation) or
  - Private Set Intersection (PSI), enabling e.g. privacy-preserving contact look-up [CHLR18].
- DH-based OPRFs are currently being standardised by the IETF.
- Post-quantum candidates still significantly less efficient[18]

[18] Martin R. Albrecht, Alex Davidson, Amit Deo and Daniel Gardham. Crypto Dark Matter on the Torus: Oblivious PRFs from shallow PRFs and FHE. Cryptology ePrint Archive, Report 2023/232. https://eprint.iacr.org/2023/232. 2023.

Lattices are rather versatile

- Fully-Homomorphic Encryption (FHE)
  - Computing on encrypted data
- Functional Encryption (FE)
  - Decryption keys correspond to $f(m)$
  - Not all function classes are currently realisable
- Identity-Based Encryption (IBE)
  - Names are the public keys
- Attribute-Based Encryption (ABE)
  - Encrypt to all doctors in an organisation etc.

Thank You