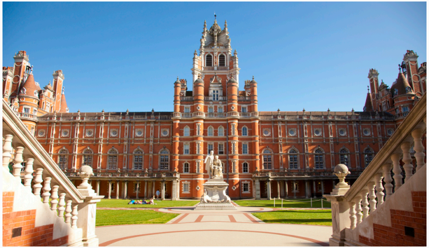
The Road to Post-Quantum Cryptography

Martin R. Albrecht

https://malb.io

Introduction

About Me



Reader in the Information Security Group, Royal Holloway, University of London

Teaching penetration testing

Research post-quantum cryptography with a focus on lattice-based cryptography [Alb17; Alb+19]
breaking cryptographic protocols/implementations such as SSH [APW09; Alb+16] and TLS [AP16; Alb+18]

Standards member of ETSI quantum-safe working group, submitter of two post-quantum candidates to the NIST process

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, secure chat, SSH, 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
- ...

A Personal View of Average-Case Complexity

RUSSELL IMPAGLIAZZO*
Computer Science and Engineering
UC, San Diego
9500 Gilman Drive
La Jolla, CA 92093-0114
russell@cs.ucsd.edu

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 $H(a_{i,b_i})$ in the public key.

Symmetric v Asymmetric Primitives

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. [Bar17]

Asymmetric Primitives

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

$$\text{Given } N = p \cdot q \text{ find } p, \text{ or}$$

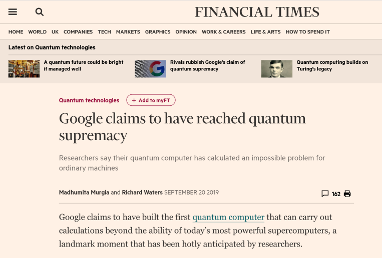
(elliptic-curve) discrete logarithms:

$$\text{Given } g^a \bmod q \text{ and } g \text{ find } a.$$

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.

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".
- This estimate is too optimistic, taking all costs into account: $> 2^{152}$ classical operations for AES-256.[1]
- Assuming a max depth of 2^{96} for a quantum circuit: overall (parallel) AES-256 cost is $\approx 2^{190}$.
- Grover's algorithm does not parallelise: have to wait for 2^X steps, cannot buy 2^{32} quantum computers and wait 2^{X-32} steps.

[1] Samuel Jaques, Michael Naehrig, Martin Roetteler, and Fernando Virdia. Implementing Grover oracles for quantum key search on AES and LowMC. Cryptology ePrint Archive, Report 2019/1146. `https://eprint.iacr.org/2019/1146`. 2019.

Symmetric Primitives: Quantum Computing Perspective (Point to Consider)

- Grover is optimal for unstructured search but block ciphers have structure.
- Consider the Even-Mansour construction:

$$y = k_0 \oplus F(x \oplus k_1)$$

  where $F(\cdot)$ is some public function and k_i have n bits.
- Optimal classical attack costs $2^{n/2}$ operations, best quantum attack takes $2^{n/3}$ quantum operations using Simon's period-finding algorithm.[2]

[2] Xavier Bonnetain, Akinori Hosoyamada, María Naya-Plasencia, Yu Sasaki, and André Schrottenloher. Quantum Attacks without Superposition Queries: the Offline Simon Algorithm. In: *IACR Cryptology ePrint Archive* 2019 (2019), p. 614. URL: `https://eprint.iacr.org/2019/614`.

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.

Post-Quantum Cryptography

Definition

Asymmetric cryptographic algorithms run on classical computers that resist attacks using classical and quantum computers.

Post-Quantum Cryptography

Definition

Asymmetric cryptographic algorithms run on classical computers that resist attacks using classical and quantum computers.

Note

Post-quantum cryptography is entirely distinct from quantum cryptography such as a quantum key exchange (QKD). The latter uses quantum effects to achieve security.

Post-Quantum Standardisation

NIST	Post Quantum Competition Process[3]
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

[3]“NIST believes that its post-quantum standards development process should not be treated as a competition; in some cases, it may not be possible to make a well-supported judgment that one candidate is ‘better’ than another.“

Post-Quantum Standardisation

- **NIST** Post Quantum Competition Process[3]
- **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

Bottom Line

Essentially, everyone is waiting for NIST.

[3] "NIST believes that its post-quantum standards development process should not be treated as a competition; in some cases, it may not be possible to make a well-supported judgment that one candidate is 'better' than another."

NIST PQC Competition Process

Timeline

- Submission deadline was November 2017.
- Round 2 selection announced January 2019.
- Final standard expected 2022-2024.

"Key Exchange"/Key Encapsulation

- `(pk,sk)` $\leftarrow$ `KeyGen()`
- `(c,k)` $\leftarrow$ `Encaps(pk)`
- `k` $\leftarrow$ `Decaps(c,sk)`

Digital Signature

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

NIST PQC Competition Process

Timeline

- Submission deadline was November 2017.
- Round 2 selection announced January 2019.
- Final standard expected 2022-2024.

"Key Exchange"/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)`

Public-key Encryption

NIST also asked for public-key encryption but this is less important as it can be built generically from a KEM and a block cipher.

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.

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 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."

Conditional Security

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

Post-Quantum Candidate Families

- Code-based (key encapsulation)
- Multivariate-based (signatures)
- OWF-based (signatures)
- Isogeny-based (key encapsulation)
- Lattice-based (key encapsulation, signatures)

NIST PQC 2nd Round

17 KEMs BIKE, Classic McEliece, CRYSTALS-KYBER, FrodoKEM, HQC, LAC, LEDAcrypt, NewHope, NTRU, NTRU Prime, NTS-KEM, ROLLO, Round5, RQC, SABER, SIKE, Three Bears.

9 SIGs CRYSTALS-DILITHIUM, FALCON, GeMSS, LUOV, MQDSS, Picnic, qTESLA, Rainbow, SPHINCS+.

Post-Quantum Candidate Families

- Code-based (key encapsulation)
- Multivariate-based (signatures)
- OWF-based (signatures)
- Isogeny-based (key encapsulation)
- Lattice-based (key encapsulation, signatures)

NIST PQC 2nd Round

17 KEMs BIKE, Classic McEliece, CRYSTALS-KYBER, FrodoKEM, HQC, LAC, LEDAcrypt, NewHope, NTRU, NTRU Prime, NTS-KEM, ROLLO, Round5, RQC, SABER, SIKE, Three Bears.

9 SIGs CRYSTALS-DILITHIUM, FALCON, GeMSS, LUOV, MQDSS, Picnic, qTESLA, Rainbow, SPHINCS+.

Post-Quantum Candidate Families

- Code-based (key encapsulation)
- Multivariate-based (signatures)
- OWF-based (signatures)
- Isogeny-based (key encapsulation)
- Lattice-based (key encapsulation, signatures)

NIST PQC 2nd Round

17 KEMs BIKE, Classic McEliece, CRYSTALS-KYBER, FrodoKEM, HQC, LAC, LEDAcrypt, NewHope, NTRU, NTRU Prime, NTS-KEM, ROLLO, Round5, RQC, SABER, SIKE, Three Bears.

9 SIGs CRYSTALS-DILITHIUM, FALCON, GeMSS, LUOV, MQDSS, Picnic, qTESLA, Rainbow, SPHINCS+.

Post-Quantum Candidate Families

- Code-based (key encapsulation)
- Multivariate-based (signatures)
- OWF-based (signatures)
- Isogeny-based (key encapsulation)
- Lattice-based (key encapsulation, signatures)

NIST PQC 2nd Round

17 KEMs BIKE, Classic McEliece, CRYSTALS-KYBER, FrodoKEM, HQC, LAC, LEDAcrypt, NewHope, NTRU, NTRU Prime, NTS-KEM, ROLLO, Round5, RQC, SABER, SIKE, Three Bears.

9 SIGs CRYSTALS-DILITHIUM, FALCON, GeMSS, LUOV, MQDSS, Picnic, qTESLA, Rainbow, SPHINCS+.

Post-Quantum Candidate Families

- Code-based (key encapsulation)
- Multivariate-based (signatures)
- OWF-based (signatures)
- Isogeny-based (key encapsulation)
- Lattice-based (key encapsulation, signatures)

NIST PQC 2nd Round

17 KEMs BIKE, Classic McEliece, CRYSTALS-KYBER, FrodoKEM, HQC, LAC, LEDAcrypt, NewHope, NTRU, NTRU Prime, NTS-KEM, ROLLO, Round5, RQC, SABER, SIKE, Three Bears.

9 SIGs CRYSTALS-DILITHIUM, FALCON, GeMSS, LUOV, MQDSS, Picnic, qTESLA, Rainbow, SPHINCS+.

Post-Quantum Candidate Families

- Code-based (key encapsulation)
- Multivariate-based (signatures)
- OWF-based (signatures)
- Isogeny-based (key encapsulation)
- Lattice-based (key encapsulation, signatures)

NIST PQC 2nd Round

17 KEMs BIKE, Classic McEliece, CRYSTALS-KYBER, FrodoKEM, HQC, LAC, LEDAcrypt, NewHope, NTRU, NTRU Prime, NTS-KEM, ROLLO, Round5, RQC, SABER, SIKE, Three Bears.

9 SIGs CRYSTALS-DILITHIUM, FALCON, GeMSS, LUOV, MQDSS, Picnic, qTESLA, Rainbow, SPHINCS+.

Post-Quantum Candidate Families

- Code-based (key encapsulation)
- Multivariate-based (signatures)
- OWF-based (signatures)
- Isogeny-based (key encapsulation)
- Lattice-based (key encapsulation, signatures)

NIST PQC 2nd Round

17 KEMs BIKE, Classic McEliece, CRYSTALS-KYBER, FrodoKEM, HQC, LAC, LEDAcrypt, NewHope, NTRU, NTRU Prime, NTS-KEM, ROLLO, Round5, RQC, SABER, SIKE, Three Bears.

9 SIGs CRYSTALS-DILITHIUM, FALCON, GeMSS, LUOV, MQDSS, Picnic, qTESLA, Rainbow, SPHINCS+.

Baseline: Pre Quantum Cryptography

RSA 2048

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

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

Curve25519

Key generation	$\approx$ 60,000 cycles
Key agreement	$\approx$ 160,000 cycles
Public key	32 bytes
Key Share	32 bytes

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

Interpretation

- A CPU running at 2Ghz has 2,000,000,000 cycles per second.
- An Ethernet frame can hold up to 1518 bytes.

KEM: Code-based

Idea: Take error-correcting code for up to t errors. Keep decoding algorithm secret, hide structure of the code.

- Encapsulated key: error vector with t error indices
- Most prominent example: McEliece (1978), uses binary Goppa codes
- Alternatives: QCMDPC codes (e.g. BIKE)
  - Less studied, less conservative, problems with CCA security

NTS-KEM(13, 136) NIST submission:

Key generation	$\approx$ 240,000,000 cycles
Encapsulation	$\approx$ 280,000 cycles
Decapsulation	$\approx$ 2,000,000 cycles
Ciphertext	253 bytes
Public key	1,419,704 bytes

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

KEM: Lattice-based

Idea: Noisy linear algebra mod q is hard and equivalent to finding short vectors in lattices.

- Learning with Errors: given

$$(\mathsf{A}, \mathsf{b}) \equiv (\mathsf{A} \cdot \mathsf{s} + \mathsf{e} \bmod q)$$

  where e is a vector with small entries, find s
- Most submissions use structured A
  - Faster, but less conservative
- Frodo uses plain, unstructured LWE

Kyber-768 NIST submission:

Key generation	$\approx$ 50,000 cycles
Encapsulation	$\approx$ 70,000 cycles
Decapsulation	$\approx$ 60,000 cycles
Ciphertext	1,088 bytes
Public key	1,184 bytes

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

KEM: SIKE

Idea: Hard problem is finding a rational map that preserves structure **between** elliptic curves.

- "Supersingular-Isogeny Diffie-Hellman" (SIDH) proposed in 2011
- Security related to claw/collision finding, but no reduction from it
- Rather young construction, more study needed
- But very promising

SIKE NIST submission:

Key generation	$\approx$ 11,000,000 cycles
Encapsulation	$\approx$ 18,000,000 cycles
Decapsulation	$\approx$ 20,000,000 cycles
Ciphertext	402 bytes
Public key	378 bytes

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

SIG: OWF-BASED

Idea: Start from one-time digital signature based on hash functions. Build Merkle trees on top to produce many-time signature schemes.

- Many tradeoffs possible
- Secure if there exist collision/pre-image resistant hash functions

SPHINCS256 NIST submission:

Key generation	$\approx$ 2,500,000 cycles
Signing	$\approx$ 42,000,000 cycles
Verifying	$\approx$ 1,300,050 cycles
Signature	41,000 bytes
Verification key	1,056 bytes

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

SIG: Lattice-based (Hash-and-Sign)

Idea: Verification key is matrix A. Hash message m to vector $H(m)$. Signature is a **short** vector $\mathbf{s}$ such that $H(m) = \mathsf{A} \cdot \mathbf{s}$.

- Can be instantiated from structured and unstructured A
- Typically uses structured lattices
- Falcon uses NTRU problem: Given $\mathsf{A} = f/g$ where both f, g are small. Find f

Falcon-1024 NIST submission:

Key generation	$\approx$ 66,000,000 cycles
Signing	$\approx$ 1,400,000 cycles
Verifying	$\approx$ 200,000 cycles
Signature	1263 bytes
Verification key	1793 bytes

`https://bench.cr.yp.to/results-sign.html`

SIG: MQ-based

Idea: Hard problem is to find solution to system of quadratic equations in many variables over a finite field.

- All but one submissions use structured systems and assume attacker cannot exploit structure
- No reduction from standard MQ problem
- MQDSS reduces to unstructured MQ

Rainbowbinary256181212 NIST submission:

Key generation	≈ 10,000,000 cycles
Signing	≈ 14,000 cycles
Verifying	≈ 10,000 cycles
Signature	42 bytes
Verification key	30,240 bytes

`https://bench.cr.yp.to/results-sign.html`

Summary

Post-quantum cryptographic schemes are

fast many are faster than RSA and competitive with/faster than ECC

larger 1.5x (SIKE) to 4x (Kyber) compared to RSA; $\approx$ 30x compared to ECC

Bonus: Post-Quantum can be easier than RSA

Approximate Greatest Common Divisors

Let $p \approx \lambda \cdot 2^{\lambda}$ be some random number. Given

$$x_i = q_i \cdot p + r_i,$$

where $q_i \approx 2^{\lambda \log \lambda}$ and $r_i \approx 2^{\lambda}$ are random numbers, find p.

This problem is assumed to be hard even on a quantum computer.

The Road Ahead

Parameters Matter

One cannot hope to simply "plug in" a key of 10^6 or 10^9 bits into a protocol designed to work for keys of 10^3 bits and expect it to work as is, and so such results could bring about significant changes to the way we do security over the Internet. For example, it could lead to a centralization of power, where key exchange will be so expensive that users would share public-keys with only a few large corporations and governments, and smaller companies would have to route their communication through these larger corporations.[4]

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

Parameters Matter

> *One cannot hope to simply "plug in" a key of 10^6 or 10^9 bits into a protocol designed to work for keys of 10^3 bits and expect it to work as is, and so such results could bring about significant changes to the way we do security over the Internet. For example, it could lead to a centralization of power, where key exchange will be so expensive that users would share public-keys with only a few large corporations and governments, and smaller companies would have to route their communication through these larger corporations.*[4]

Example: SSH has a packet size $<$ 32KB, McEliece public keys are $\approx$ 1MB (but ciphertexts are small).

---

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

We will miss DH ...

Non-Interactive Key Exchange (NIKE):

- Bob knows Alice's long-term pk g^a
- Alice knows Bob's long-term pk g^b
- Agree on a shared key

$$(g^a)^b = (g^b)^a$$

  before exchanging any messages
- Expensive to instantiate post-quantum

Oblivious PRF:

- Alice sends h^r to Bob
- Bob computes

$$(h^r)^b$$

- Alice computes

$$(h^{r \cdot b})^{(1/r)} = h^b$$

- First, inefficient proposal from lattices very recent

. . . but Lattices are very versatile

- Fully-Homomorphic Encryption (FHE)
  - Computing on encrypted data
  - Only from lattices
- 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.

Signature Scheme != Signature Scheme

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.

- This does not imply an adversary cannot produce a new signature for a message already signed: non-malleability.
- This binds a message to known public key, but it does not bind a public-key to a message: conservative exclusive ownership.

In contrast, e.g. RFC 8032 (EdDSA) satisfies both non-malleability and conservative exclusive ownership.[5]

[5] Dennis Jackson, Cas Cremers, Katriel Cohn-Gordon, and Ralf Sasse. Seems Legit: Automated Analysis of Subtle Attacks on Protocols that Use Signatures. Cryptology ePrint Archive, Report 2019/779. `https://eprint.iacr.org/2019/779`. 2019.

Alternatives: QKD?

QKD: has fundamental practical limitations; does not address large parts of the security problem; is poorly understood in terms of potential attacks.
By contrast, post-quantum public key cryptography appears to offer much more effective mitigations for real-world communications systems from the threat of future quantum computers.[6]

- attacks on implementations/instantiations
- limited range, dedicated hardware
- limited speed → keys then used in AES
- authentication required: MAC or digital signature

[6] National Cyber Security Centre. Quantum Key Distribution. https://www.ncsc.gov.uk/information/quantum-key-distribution. 2016.

The Road Ahead

- We need to understand the underlying hard problems better to tune parameters
- Resistance to side-channel attacks
- Efficient, safe implementations
    - This is a real opportunity: we get to rip out the old piping and replace it by modern solutions[7]
- How fast is fast enough? How small is small enough?
    - Here your use cases can help!
- How do existing protocols interact with post-quantum primitives? Should we change protocols?
    - If you have bespoke protocols, this is something to check now.

---

[7] José Bacelar Almeida, Cécile Baritel-Ruet, Manuel Barbosa, Gilles Barthe, François Dupressoir, Benjamin Grégoire, Vincent Laporte, Tiago Oliveira, Alley Stoughton, and Pierre-Yves Strub. Machine-Checked Proofs for Cryptographic Standards. Cryptology ePrint Archive, Report 2019/1155. `https://eprint.iacr.org/2019/1155`. 2019.

Don't Jump the Gun!

- Temptation to pick one of the NIST candidates as drop-in replacement for deployment in existing protocols now
- This is a terrible idea!
    - mediocre performance
    - non-optimal security properties
- Bad cryptography is very hard to get rid of (think MD5)
- Will also need to think carefully about changes to protocols
- Let's get this one right!

Don't Jump the Gun!

- Temptation to pick one of the NIST candidates as drop-in replacement for deployment in existing protocols now
- This is a terrible idea!
  - mediocre performance
  - non-optimal security properties
- Bad cryptography is very hard to get rid of (think MD5)
- Will also need to think carefully about changes to protocols
- Let's get this one right!

Proof of Concept Code

...even worse idea: pick **source code** of one of the NIST candidates to deploy

Thank You

[Alb+16] Martin R. Albrecht, Jean Paul Degabriele, Torben Brandt Hansen, and Kenneth G. Paterson. A Surfeit of SSH Cipher Suites. In: *ACM CCS 2016*. Ed. by Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi. ACM Press, Oct. 2016, pp. 1480–1491. DOI: 10.1145/2976749.2978364.

[Alb17] Martin R. Albrecht. On Dual Lattice Attacks Against Small-Secret LWE and Parameter Choices in HElib and SEAL. In: *EUROCRYPT 2017, Part II*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10211. LNCS. Springer, Heidelberg, 2017, pp. 103–129. DOI: 10.1007/978-3-319-56614-6_4.

[Alb+18] Martin R. Albrecht, Jake Massimo, Kenneth G. Paterson, and Juraj Somorovsky. Prime and Prejudice: Primality Testing Under Adversarial Conditions. In: *ACM CCS 2018*. Ed. by David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang. ACM Press, Oct. 2018, pp. 281–298. DOI: 10.1145/3243734.3243787.

[Alb+19] Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The General Sieve Kernel and New Records in Lattice Reduction. In: *EUROCRYPT 2019, Part II*. Ed. by Yuval Ishai and Vincent Rijmen. Vol. 11477. LNCS. Springer, Heidelberg, May 2019, pp. 717–746. DOI: 10.1007/978-3-030-17656-3_25.

[Alm+19] José Bacelar Almeida, Cécile Baritel-Ruet, Manuel Barbosa, Gilles Barthe, François Dupressoir, Benjamin Grégoire, Vincent Laporte, Tiago Oliveira, Alley Stoughton, and Pierre-Yves Strub. Machine-Checked Proofs for Cryptographic Standards. Cryptology ePrint Archive, Report 2019/1155. https://eprint.iacr.org/2019/1155. 2019.

[AP16] Martin R. Albrecht and Kenneth G. Paterson. Lucky Microseconds: A Timing Attack on Amazon's s2n Implementation of TLS. In: *EUROCRYPT 2016, Part I*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9665. LNCS. Springer, Heidelberg, May 2016, pp. 622–643. DOI: 10.1007/978-3-662-49890-3_24.

[APW09] Martin R. Albrecht, Kenneth G. Paterson, and Gaven J. Watson. Plaintext Recovery Attacks against SSH. In: *2009 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2009, pp. 16–26. DOI: 10.1109/SP.2009.5.

[Bar17] Boaz Barak. The Complexity of Public-Key Cryptography. Cryptology ePrint Archive, Report 2017/365. http://eprint.iacr.org/2017/365. 2017.

[Bon+19] Xavier Bonnetain, Akinori Hosoyamada, María Naya-Plasencia, Yu Sasaki, and André Schrottenloher. Quantum Attacks without Superposition Queries: the Offline Simon Algorithm. In: *IACR Cryptology ePrint Archive* 2019 (2019), p. 614. URL: https://eprint.iacr.org/2019/614.

[Jac+19] Dennis Jackson, Cas Cremers, Katriel Cohn-Gordon, and Ralf Sasse. Seems Legit: Automated Analysis of Subtle Attacks on Protocols that Use Signatures. Cryptology ePrint Archive, Report 2019/779. https://eprint.iacr.org/2019/779. 2019.

[Jaq+19] Samuel Jaques, Michael Naehrig, Martin Roetteler, and Fernando Virdia. Implementing Grover oracles for quantum key search on AES and LowMC. Cryptology ePrint Archive, Report 2019/1146. https://eprint.iacr.org/2019/1146. 2019.

[NCSC:QKD16] National Cyber Security Centre. Quantum Key Distribution. https://www.ncsc.gov.uk/information/quantum-key-distribution. 2016.