The transition to post-quantum cryptography

Daniel J. Bernstein & Tanja Lange

University of Illinois at Chicago; Ruhr University Bochum; Academia Sinica
&
Eindhoven University of Technology; Academia Sinica

1 April 2022

Cryptography

Sender
“Alice”

Receiver
“Bob”

Tsai Ing-Wen picture credit: By 總統府, Attribution, Wikimedia. Joe Biden picture credit: By Adam Schultz - White House, Public Domain, Wikimedia.

Cryptography

Sender
"Alice"

Untrustworthy network
"Eve"

Receiver
"Bob"

- Motivation #1: Communication channels are spying on our data.
- Motivation #2: Communication channels are modifying our data.

Tsai Ing-Wen picture credit: By 總統府, Attribution, Wikimedia. Joe Biden picture credit: By Adam Schultz - White House, Public Domain, Wikimedia.

Cryptography

Sender
"Alice"

Untrustworthy network
"Eve"

Receiver
"Bob"

- Motivation #1: Communication channels are spying on our data.
- Motivation #2: Communication channels are modifying our data.
- Literal meaning of cryptography: "secret writing".
- Achieves various security goals by secretly transforming messages.
  - Confidentiality: Eve cannot infer information about the content
  - Integrity: Eve cannot modify the message without this being noticed
  - Authenticity: Bob is convinced that the message originated from Alice

Public-key vs. symmetric-key cryptography

Public-key cryptography
Each user has 2 keys:
a public key and a private key.

Public key can be posted online;
private key must be kept secret.

Often can compute public key from private key.
Other direction must be hard.

Can be used on Internet with unknown parties.
Requires mathematically hard problem.

Symmetric-key cryptography
Each pair of users shares a key.
This key is symmetric between both users.

This key must be kept secret.

Symmetric systems often faster than
public-key systems.
Use latter to get symmetric key.

Requires that users have securely shared this key.
Typically cheaper/faster than public-key crypto.

Cryptographic tools

Many factors influence the security and privacy of data:

- Secure storage, physical security; access control.
- Protection against alteration of data
  $\Rightarrow$ public-key signatures, message-authentication codes.
- Protection of sensitive content against reading
  $\Rightarrow$ encryption (public-key or symmetric-key).

Many more security goals studied in cryptography

- Protecting against denial of service.
- Stopping traffic analysis.
- Securely tallying votes.
- Searching in and computing on encrypted data.
- ...

Cryptanalysis

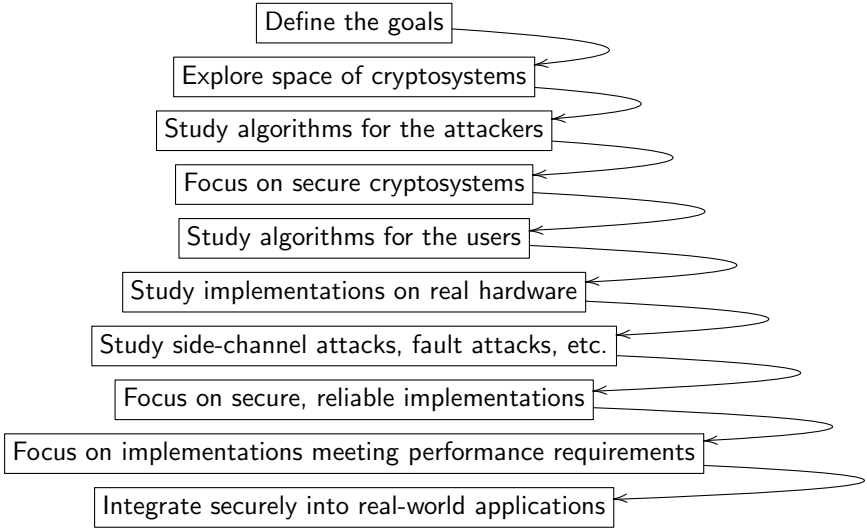
- Cryptanalysis is the study of security of cryptosystems.
- Breaking a system can mean that the hardness assumption was not hard or that it just was not as hard as previously assumed.
- Public cryptanalysis is ultimately constructive – ensure that secure systems get used, not insecure ones.
- Weakened crypto ultimately backfires – attacks today because of crypto wars in the 90s.
- Good arsenal of general approaches to cryptanalysis. There are some automated tools.
- This area is constantly under development; researchers revisit systems continuously.


神威
太湖之光

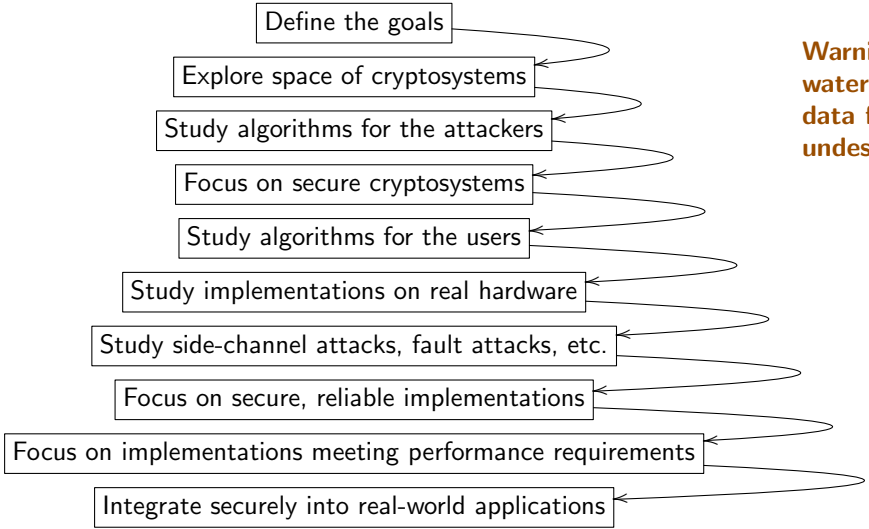
Security assumptions

- Hardness assumptions at the basis of all public-key and essentially all symmetric-key systems result from (failed) attempts at breaking systems.
- Security "proofs" are built only on top of those assumptions.
  These relate the hardness of breaking a bigger system to the hardness of these assumptions.
- A solid symmetric system is required to be as strong as exhaustive key search.
- For public-key systems the best attacks are faster than exhaustive key search.
  Parameters are chosen to ensure that the best attack is infeasible.

Many stages of cryptographic research from design to deployment



Many stages of cryptographic research from design to deployment



Warning: waterfall data flow, undesirable.

Key size recommendations

	Parameter	Legacy	Future System Use	
			Near Term	Long Term
Symmetric Key Size	k	80	128	256
Hash Function Output Size	m	160	256	512
MAC Output Size*	m	80	128	256
RSA Problem	$\ell(n) \geq$	1024	3072	15360
Finite Field DLP	$\ell(p^n) \geq$	1024	3072	15360
	$\ell(p), \ell(q) \geq$	160	256	512
ECDLP	$\ell(q) \geq$	160	256	512
Pairing	$\ell(p^{k \cdot n}) \geq$	1024	6144	15360
	$\ell(p), \ell(q) \geq$	160	256	512

- Source: ECRYPT-CSA "Algorithms, Key Size and Protocols Report" (2018).
- These recommendations take into account attacks known today.
- Use extrapolations to larger problem sizes.
- Attacker power typically limited to 2^{128} operations (less for legacy).
- More to come on long-term security ...

Summary: current state of the art

- Currently used crypto (check the lock icon in your browser) starts with RSA, Diffie-Hellman (DH) in finite fields, or elliptic-curve Diffie-Hellman (ECDH).
- Older standards are RSA or elliptic curves from NIST (or Brainpool), e.g. NIST P256 or ECDSA.
- Internet currently moving over to Curve25519 (Bernstein) and Ed25519 (Bernstein, Duif, Lange, Schwabe, and Yang).
- For symmetric crypto, TLS (the protocol behind https) uses AES or ChaCha20 and some MAC, e.g. AES-GCM or ChaCha20-Poly1305.
  High-end devices have support for AES-GCM, smaller ones do better with ChaCha20-Poly1305.
- Security is getting better. Some obstacles: bugs; untrustworthy hardware.

Algorithms for Quantum Computation: Discrete Logarithms and Factoring

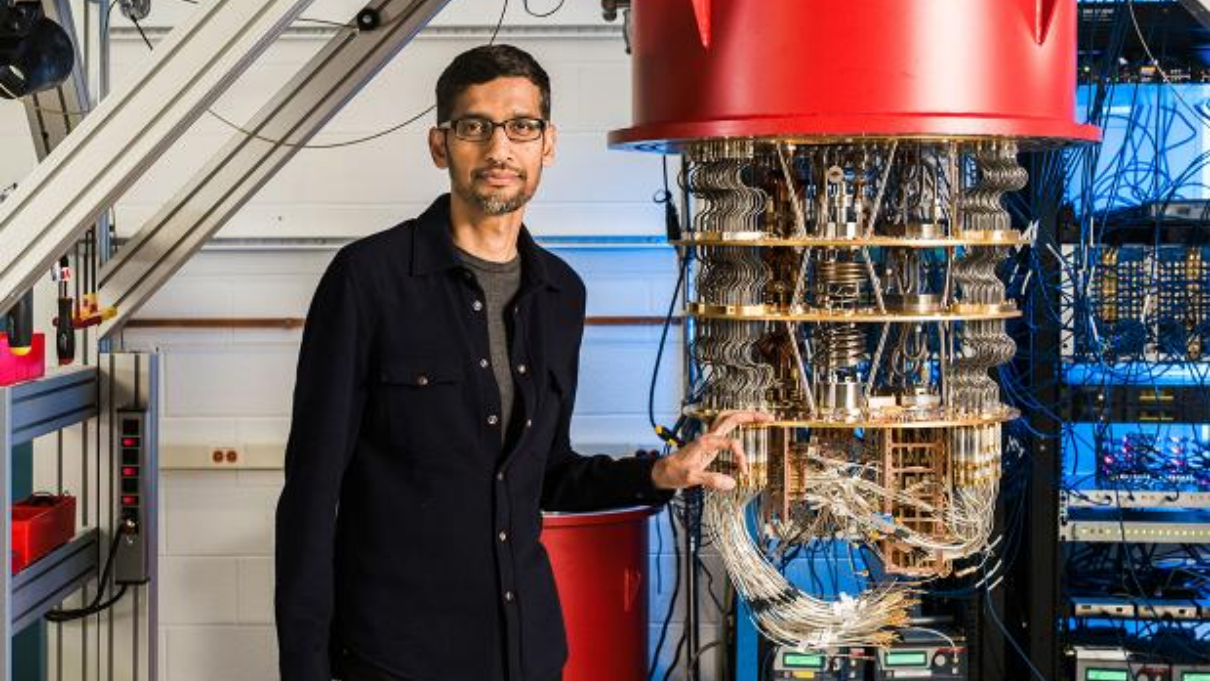
Peter W. Shor
AT&T Bell Labs
Room 2D-149
600 Mountain Ave.
Murray Hill, NJ 07974, USA

Abstract

A computer is generally considered to be a universal computational device; i.e., it is believed able to simulate any physical computational device with a cost in computation time of at most a polynomial factor. It is not clear whether this is still true when quantum mechanics is taken into consideration. Several researchers, starting with David Deutsch, have developed models for quantum mechanical computers and have investigated their compu-

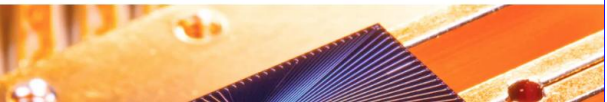
[1, 2]. Although he did not ask whether quantum mechanics conferred extra power to computation, he did show that a Turing machine could be simulated by the reversible unitary evolution of a quantum process, which is a necessary prerequisite for quantum computation. Deutsch [9, 10] was the first to give an explicit model of quantum computation. He defined both quantum Turing machines and quantum circuits and investigated some of their properties.

The next part of this paper discusses how quantum com-



Premium

Quantum computing could end encryption within five years, says Google boss



Mr Pichai said a combination of artificial intelligence and quantum would "help us tackle some of the biggest problems we see", but said it was important encryption evolved to match this.

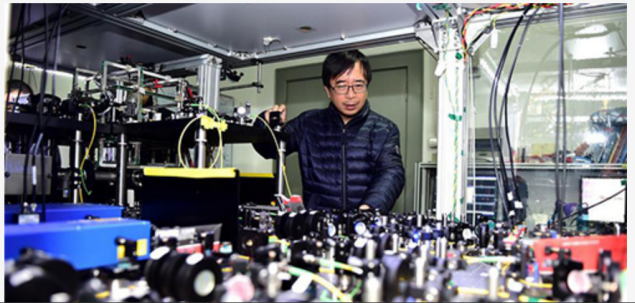
"In a five to ten year time frame, quantum computing will break encryption as we know it today."

This is because current encryption methods, by which information such as texts or passwords is turned into code to make it unreadable, rely upon the fact that classic computers would take billions of years to decipher that code.

Quantum computers, with their ability to be

Chinese researchers expect quantum leap in computing, challenging Google's supremacy

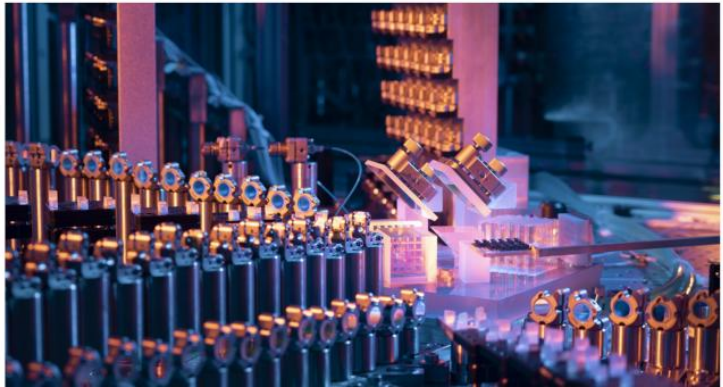
Source: Global Times Published: 2020/8/26 14:58:42

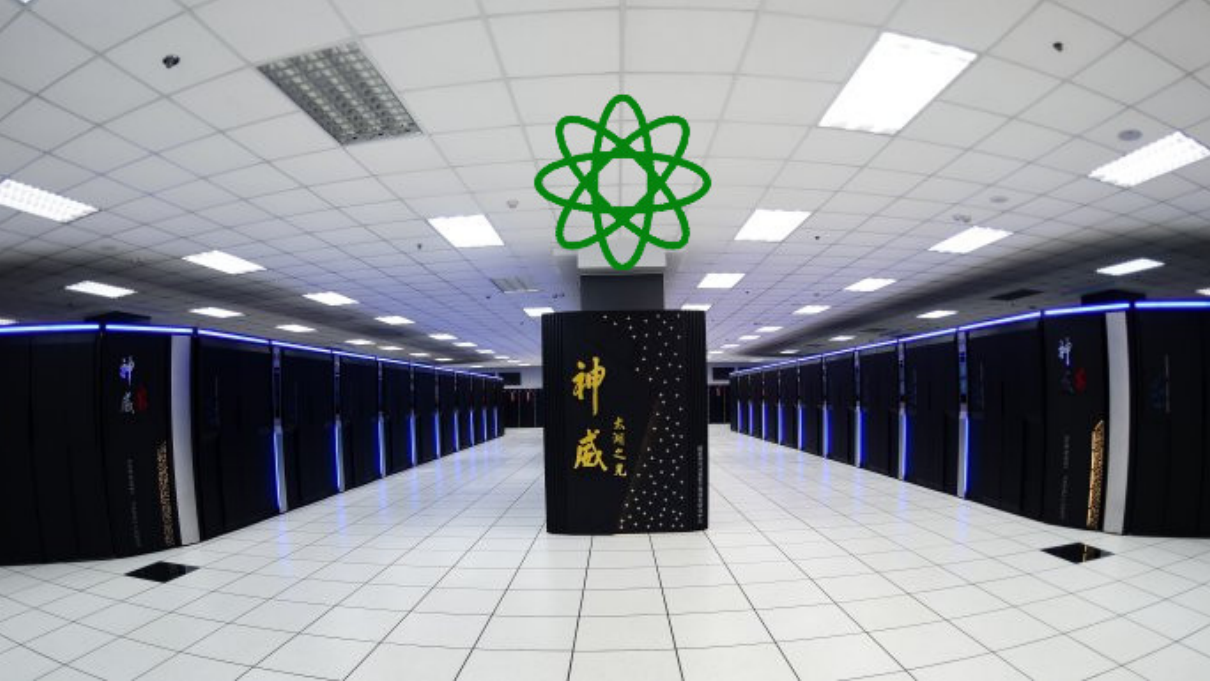


SOURCE / ECONOMY

Chinese researchers achieve quantum advantage in two mainstream routes

By Global Times
Published: Oct 26, 2021 01:18 PM




神威
太湖之光

National Academy of Sciences (US)

4 December 2018: Report on quantum computing

Don't panic. "Key Finding 1: Given the current state of quantum computing and recent rates of progress, it is highly unexpected that a quantum computer that can compromise RSA 2048 or comparable discrete logarithm-based public key cryptosystems will be built within the next decade."

National Academy of Sciences (US)

4 December 2018: Report on quantum computing

Don't panic. "Key Finding 1: Given the current state of quantum computing and recent rates of progress, it is highly unexpected that a quantum computer that can compromise RSA 2048 or comparable discrete logarithm-based public key cryptosystems will be built within the next decade."

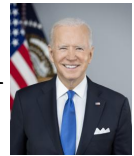
Panic. "Key Finding 10: Even if a quantum computer that can decrypt current cryptographic ciphers is more than a decade off, the hazard of such a machine is high enough—and the time frame for transitioning to a new security protocol is sufficiently long and uncertain—that prioritization of the development, standardization, and deployment of post-quantum cryptography is critical for minimizing the chance of a potential security and privacy disaster."

"[Section 4.4:] In particular, all encrypted data that is recorded today and stored for future use, will be cracked once a large-scale quantum computer is developed."

Commonly used systems

Sender
"Alice"

Untrustworthy network
"Eve"



Receiver
"Bob"

Cryptography with symmetric keys
AES-128. **AES-192**. **AES-256**. **AES-GCM**. **ChaCha20**. **HMAC-SHA-256**. **Poly1305**. **SHA-2**. **SHA-3**. **Salsa20**.

Cryptography with public keys
BN-254. **Curve25519**. **DH**. **DSA**. **ECDH**. **ECDSA**. **EdDSA**. **NIST P-256**. **NIST P-384**. **NIST P-521**. **RSA encrypt**. **RSA sign**. **secp256k1**.

Commonly used systems

Sender
"Alice"

Untrustworthy network
"Eve" with quantum computer

Receiver
"Bob"

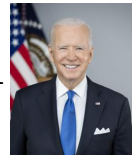
Cryptography with symmetric keys
AES-128. **AES-192**. **AES-256**. **AES-GCM**. **ChaCha20**. **HMAC-SHA-256**. **Poly1305**. **SHA-2**. **SHA-3**. **Salsa20**.

Cryptography with public keys
BN-254. **Curve25519**. **DH**. **DSA**. **ECDH**. **ECDSA**. **EdDSA**. **NIST P-256**. **NIST P-384**. **NIST P-521**. **RSA encrypt**. **RSA sign**. **secp256k1**.

Symmetric-key authenticated encryption

Sender
"Alice"

Untrustworthy network
"Eve" with quantum computer



Receiver
"Bob"

- Very easy solutions **if Alice and Bob already share long secret key k**:
  - "One-time pad" for confidentiality.
  - "Wegman–Carter MAC" for integrity and authenticity.
- AES-256: Standardized method to expand **short secret key** (256-bit k) into string indistinguishable from long secret key.
- AES introduced in 1998 by Daemen and Rijmen. Security analyzed in papers by dozens of cryptanalysts.
- No credible threat from quantum algorithms. Grover costs 2^{128}.
- Some results assume attacker has quantum access to computation, then some systems are weaker . . . but I'd know if my laptop had turned into a quantum computer.

Post-quantum cryptography

Cryptography under the assumption that the attacker has a quantum computer.

- 1994: Shor's quantum algorithm. 1996: Grover's quantum algorithm.
  Many subsequent papers on quantum algorithms: see quantumalgorithmzoo.org.
- 2003: Daniel J. Bernstein introduces term Post-quantum cryptography.
- 2006: First International Workshop on Post-Quantum Cryptography. PQCrypto 2006, 2008, 2010, 2011, 2013, 2014, 2016, 2017, 2018, 2019, 2020, 2021, (soon) 2022.
- 2015: NIST hosts its first workshop on post-quantum cryptography.
- 2016: NIST announces a standardization project for post-quantum systems.
- 2017: Deadline for submissions to the NIST competition.
- 2019: Second round of NIST competition begins.
- 2020: Third round of NIST competition begins.
- ~~2021~~ 2022 ~~"not later than the end of March"~~: NIST announces first selections.
- 2023/2024?: NIST issues post-quantum standards.

PQCRYPTO initial recommendations of long-term secure post-quantum systems (2015)

Daniel Augot, Lejla Batina, Daniel J. Bernstein, Joppe Bos,
Johannes Buchmann, Wouter Castryck, Orr Dunkelman,
Tim Güneysu, Shay Gueron, Andreas Hülsing,
Tanja Lange, Mohamed Saied Emam Mohamed,
Christian Rechberger, Peter Schwabe, Nicolas Sendrier,
Frederik Vercauteren, Bo-Yin Yang

Initial recommendations (PQCRYPTO, 2015)

- **Symmetric encryption** Thoroughly analyzed, 256-bit keys:
  - AES-256
  - Salsa20 with a 256-bit key

  Evaluating: Serpent-256, ...
- **Symmetric authentication** Information-theoretic MACs:
  - GCM using a 96-bit nonce and a 128-bit authenticator
  - Poly1305
- **Public-key encryption** McEliece with binary Goppa codes:
  - length $n = 6960$, dimension $k = 5413$, $t = 119$ errors

  Evaluating: QC-MDPC, Stehlé-Steinfeld NTRU, ...
- **Public-key signatures** Hash-based (minimal assumptions):
  - XMSS with any of the parameters specified in CFRG draft
  - SPHINCS-256

  Evaluating: HFEv-, ...

Major categories of public-key post-quantum systems

- **Code-based** encryption: McEliece cryptosystem has survived since 1978. Short ciphertexts and large public keys. Security relies on hardness of decoding error-correcting codes.
- **Hash-based** signatures: very solid security and small public keys. Require only a secure hash function (hard to find second preimages).
- **Isogeny-based** encryption: new kid on the block, promising short keys and ciphertexts and non-interactive key exchange. Security relies on hardness of finding isogenies between elliptic curves over finite fields.
- **Lattice-based** encryption and signatures: possibility for balanced sizes. Security relies on hardness of finding short vectors in some (typically special) lattice.
- **Multivariate-quadratic** signatures: short signatures and large public keys. Security relies on hardness of solving systems of multivariate equations over finite fields.

Warning: These are categories of mathematical problems;
individual systems may be totally insecure if the problem is not used correctly.

We have a good understanding of what a quantum computer can do,
but new systems need more analysis.

More details of what NIST has said

2020.07.22 NIST: "NIST intends to select a small number of the finalists for standardization at the end of the third round. In addition, NIST expects to standardize a small number of the alternate candidates (most likely at a later date). . . . If NIST's confidence in better performing signature algorithms is shaken by new analysis , SPHINCS+ could provide an immediately available algorithm for standardization at the end of the third round."

2021.12.07 NIST: "NIST will be selecting the first post-quantum standards for KEMs (and digital signatures) around the end of December or sometime in early January."

2022.02.08 Nature article "The race to save the Internet from quantum hackers": "In the next few months, the institute will select two algorithms for each application. It will then begin to draft standards for one, while keeping the other as a reserve in case the first choice ends up being broken by an unexpected attack, quantum or otherwise."

2022.02.09 NIST: "We hope to be able to announce the results and report not later than the end of March."

2022.03.31 NIST: "We ask for a little bit more patience since we are not ready to make the announcement today. We still expect to make it very soon."

The lattice decisions: not easy to predict

2020.07.22 NIST: "As CRYSTALS-KYBER, NTRU, and SABER are all structured lattice schemes, NIST intends to select, at most, one of these finalists to be standardized. The same is true for the finalist signature schemes CRYSTALS-DILITHIUM and FALCON. In NIST's current view, these structured lattice schemes appear to be the most promising general-purpose algorithms for public-key encryption/KEM and digital signature schemes."

Hard for community to figure out what NIST is going to do:

- Kyber faster than SABER on big CPUs. SABER faster in hardware.
  Kyber more fashionable.
- NTRU: avoids several patent questions, but slower keygen and less fashionable.
- Falcon: much smaller sigs than Dilithium, but harder to implement and less fashionable.
- There are some claims regarding impact of performance on applications.
  Not clear what weight NIST will put on these claims.

How does PQC affect protocols?

- Length fields don't fit.

How does PQC affect protocols?

- Length fields don't fit.
  ⇒ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to PQC one,
  putting PQC part into the payload.

How does PQC affect protocols?

- Length fields don't fit.
  ⇒ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to PQC one,
  putting PQC part into the payload.
- Speed, resources.
  Combined schemes take about twice the time.

How does PQC affect protocols?

- Length fields don't fit.
  ⇒ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to PQC one,
  putting PQC part into the payload.
- Speed, resources.
  Combined schemes take about twice the time.
  Most experiments don't look so devastating.
- Interface mismatch – KEM instead of key exchange,
  stateful hash-based signatures fit some applications.

How does PQC affect protocols?

- Length fields don't fit.
  ⇒ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to PQC one,
  putting PQC part into the payload.
- Speed, resources.
  Combined schemes take about twice the time.
  Most experiments don't look so devastating.
- Interface mismatch – KEM instead of key exchange,
  stateful hash-based signatures fit some applications.
  ⇒ Shoehorning PQC into current systems may prioritize
  weaker systems; redesigning protocols takes time.

How does PQC affect protocols?

- Length fields don't fit.
  ⇒ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to PQC one,
  putting PQC part into the payload.
- Speed, resources.
  Combined schemes take about twice the time.
  Most experiments don't look so devastating.
- Interface mismatch – KEM instead of key exchange,
  stateful hash-based signatures fit some applications.
  ⇒ Shoehorning PQC into current systems may prioritize
  weaker systems; redesigning protocols takes time.
- Validation and certification schemes are not yet updated.

How does PQC affect protocols?

- Length fields don't fit.
  ⇒ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to PQC one,
  putting PQC part into the payload.
- Speed, resources.
  Combined schemes take about twice the time.
  Most experiments don't look so devastating.
- Interface mismatch – KEM instead of key exchange,
  stateful hash-based signatures fit some applications.
  ⇒ Shoehorning PQC into current systems may prioritize
  weaker systems; redesigning protocols takes time.
- Validation and certification schemes are not yet updated.
  ⇒ Combine pre-and post-quantum schemes,
  certification only applies to pre-quantum scheme.

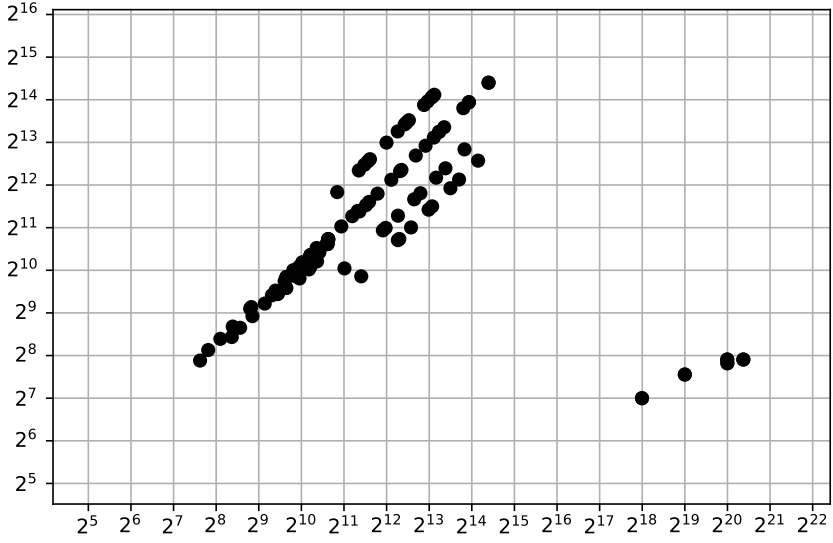
How does PQC affect protocols?

- Length fields don't fit.
  ⇒ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to PQC one,
  putting PQC part into the payload.
- Speed, resources.
  Combined schemes take about twice the time.
  Most experiments don't look so devastating.
- Interface mismatch – KEM instead of key exchange,
  stateful hash-based signatures fit some applications.
  ⇒ Shoehorning PQC into current systems may prioritize
  weaker systems; redesigning protocols takes time.
- Validation and certification schemes are not yet updated.
  ⇒ Combine pre-and post-quantum schemes,
  certification only applies to pre-quantum scheme.
  For such *hybrid* schemes, ensure that as strong as strongest not as weak as weakest.

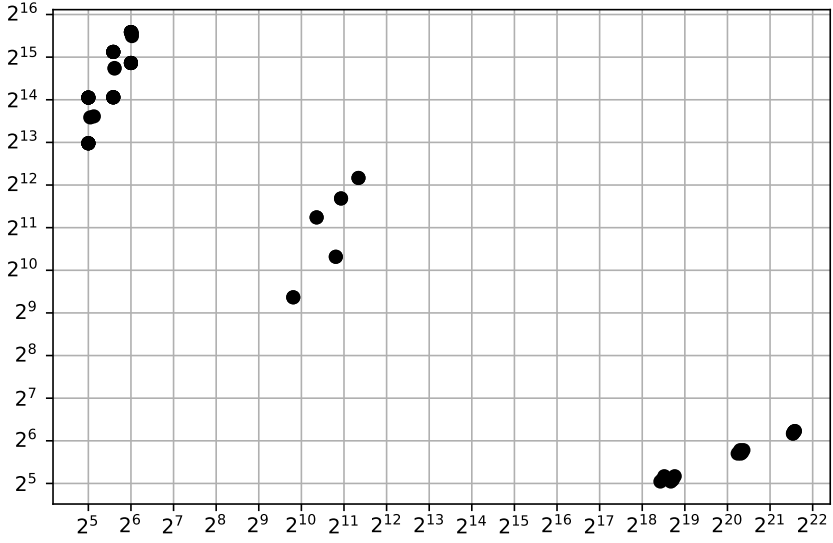
How does PQC affect protocols?

- Length fields don't fit.
  ⇒ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to PQC one,
  putting PQC part into the payload.
- Speed, resources.
  Combined schemes take about twice the time.
  Most experiments don't look so devastating.
- Interface mismatch – KEM instead of key exchange,
  stateful hash-based signatures fit some applications.
  ⇒ Shoehorning PQC into current systems may prioritize
  weaker systems; redesigning protocols takes time.
- Validation and certification schemes are not yet updated.
  ⇒ Combine pre-and post-quantum schemes,
  certification only applies to pre-quantum scheme.
  For such *hybrid* schemes, ensure that as strong as strongest not as weak as weakest.
- New security assumptions, new proofs, lots of new code.

Encryption (KEM): ciphertext size (vertical) vs. public-key size (horizontal)



Signatures: signature size (vertical) vs. public-key size (horizontal)



Deployment issues & solutions

- Different recommendations for rollout in different risk scenarios:
  - Use most efficient systems with ECC or RSA,
    to ease usage and gain familiarity.
  - Use most conservative systems (possibly with ECC),
    to ensure that data really remains secure.
- Protocol integration and implementation problems:
  - Key sizes or message sizes are larger for post-quantum systems,
    but IPv6 guarantees only delivery of $\leq$ 1280-byte packets,
    TLS software has length limits, etc.
  - Google experimented with larger keys and noticed delays and dropped connections.
  - Long-term keys require extra care (reaction attacks).
- Some libraries exist, quality is getting better.
- Google and Cloudflare are running some experiments of including post-quantum systems into TLS.

Timeline of widely used software adding post-quantum options

2016.07: Chrome adds `newhope1024` option.
Used for an experiment with Google servers.

Timeline of widely used software adding post-quantum options

2016.07: Chrome adds `newhope1024` option.
Used for an experiment with Google servers.
A patent holder then contacts Google and asks for money—oops!

Timeline of widely used software adding post-quantum options

2016.07: Chrome adds `newhope1024` option.
Used for an experiment with Google servers.
A patent holder then contacts Google and asks for money—oops!
2016.11: Chrome removes `newhope1024` option.

Timeline of widely used software adding post-quantum options

2016.07: Chrome adds `newhope1024` option.
Used for an experiment with Google servers.
A patent holder then contacts Google and asks for money—oops!
2016.11: Chrome removes `newhope1024` option.
2019.04: OpenSSH 8.0 adds `sntrup761` option.
Used if client and server configure it.

Timeline of widely used software adding post-quantum options

2016.07: Chrome adds `newhope1024` option.
Used for an experiment with Google servers.
A patent holder then contacts Google and asks for money—oops!

2016.11: Chrome removes `newhope1024` option.

2019.04: OpenSSH 8.0 adds `sntrup761` option.
Used if client and server configure it.

2019.07: Chrome adds `ntruhrss701` option.
Used for an experiment with Cloudflare servers.

Timeline of widely used software adding post-quantum options

2016.07: Chrome adds `newhope1024` option.
Used for an experiment with Google servers.
A patent holder then contacts Google and asks for money—oops!

2016.11: Chrome removes `newhope1024` option.

2019.04: OpenSSH 8.0 adds `sntrup761` option.
Used if client and server configure it.

2019.07: Chrome adds `ntruhrss701` option.
Used for an experiment with Cloudflare servers.

2021.05: OpenBSD adds `sntrup761` option for IPsec.
Used if client and server configure it.

Timeline of widely used software adding post-quantum options

2016.07: Chrome adds `newhope1024` option.
Used for an experiment with Google servers.
A patent holder then contacts Google and asks for money—oops!

2016.11: Chrome removes `newhope1024` option.

2019.04: OpenSSH 8.0 adds `sntrup761` option.
Used if client and server configure it.

2019.07: Chrome adds `ntruhrss701` option.
Used for an experiment with Cloudflare servers.

2021.05: OpenBSD adds `sntrup761` option for IPsec.
Used if client and server configure it.

2021.11: OpenSSH pre-8.9 enables `sntrup761` on server by default.
Used if client configures it.

Timeline of widely used software adding post-quantum options

2016.07: Chrome adds `newhope1024` option.
Used for an experiment with Google servers.
A patent holder then contacts Google and asks for money—oops!
2016.11: Chrome removes `newhope1024` option.
2019.04: OpenSSH 8.0 adds `sntrup761` option.
Used if client and server configure it.
2019.07: Chrome adds `ntruhrss701` option.
Used for an experiment with Cloudflare servers.
2021.05: OpenBSD adds `sntrup761` option for IPsec.
Used if client and server configure it.
2021.11: OpenSSH pre-8.9 enables `sntrup761` on server by default.
Used if client configures it.
2022.03: OpenSSH pre-8.10 enables `sntrup761` on client by default.

Timeline of widely used software adding post-quantum options

2016.07: Chrome adds `newhope1024` option.
Used for an experiment with Google servers.
A patent holder then contacts Google and asks for money—oops!
2016.11: Chrome removes `newhope1024` option.
2019.04: OpenSSH 8.0 adds `sntrup761` option.
Used if client and server configure it.
2019.07: Chrome adds `ntruhrss701` option.
Used for an experiment with Cloudflare servers.
2021.05: OpenBSD adds `sntrup761` option for IPsec.
Used if client and server configure it.
2021.11: OpenSSH pre-8.9 enables `sntrup761` on server by default.
Used if client configures it.
2022.03: OpenSSH pre-8.10 enables `sntrup761` on client by default.

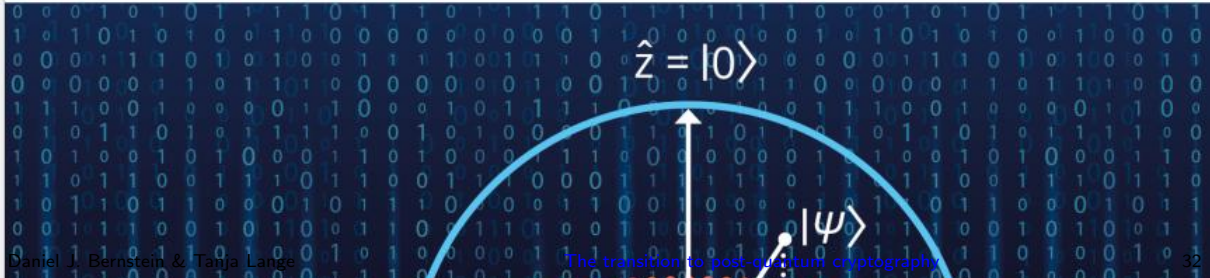
These encryption layers are *added* to X25519 encryption (ECC).
If lattices are completely broken, still have pre-quantum security.

Post-Quantum Cryptography: Current state and quantum mitigation

enisa

Ward Beullens, Jan-Pieter D'Anvers, Andreas Hülsing,
Tanja Lange, Lorenz Panny, Cyprien de Saint Guilhem, Nigel P. Smart.
Evangelos Rekleitis, Angeliki Aktypi, Athanasios-Vasileios Grammatopoulos.

EUROPEAN UNION AGE
FOR CYBERSECU



ENISA report: Current state and quantum mitigation

Chapters

1. Introduction
2. Families of Post-Quantum Algorithms
3. Security Notions and Generic Transforms
4. NIST Round 3 Finalists
5. Alternate Candidates
6. Quantum Mitigation
   - 6.1 Hybrid schemes
   - 6.2 Protective measures for pre-quantum cryptography

Report available from ENISA's website.

Hybrid schemes – implementation choices

Combine one (or more) pre-quantum schemes with one (or more) post-quantum schemes.

Signatures:
All individual signatures must be valid for the hybrid signature to be valid.

Hybrid schemes – implementation choices

Combine one (or more) pre-quantum schemes with one (or more) post-quantum schemes.

Signatures:
All individual signatures must be valid for the hybrid signature to be valid.

DH / KEM:
Use KDF (key-derivation function) on concatenation of keys or consume iteratively.

Different options to hybridize

- Execute pre- and post-quantum next to each other.
- Wrap PQC inside pre-quantum (benefit for length fields).
- Wrap pre-quantum inside PQC (limit the attack surface – quantum attacker cannot even break pre-quantum scheme).

US ANSI X9 on post-quantum hybrids

2021: "As we transition from classical cryptography to post-quantum cryptography (PQC), there is a need to understand the proper ways to use both methods simultaneously. PQC methods will not be able to be used as a direct replacement in all cases. And the confidence and broad acceptance of PQC methods will not be as great as classical cryptography. **Simultaneous use of both classical cryptography and PQC methods for both security and acceptance** is required during a transition and may be required long term as well. There are improper and insecure ways of implementing a hybrid of classical and PQC methods. Specifying the proper methods of using both are required." (emphasis added)

French ANSSI on post-quantum hybrids

2022: "Although this new post-quantum toolbox may seem handy for developers, the maturity level of the post-quantum algorithms presented to the NIST process should not be overestimated. Many aspects lack cryptanalytical hindsight or are still research topics, e.g. analysis of the difficulty of the underlying problem in the classical and quantum computation models, dimensioning, integration of schemes in protocols and more importantly the design of secure implementations. This situation will probably last some time after the publication of NIST standards. **Acknowledging the immaturity of PQC is important: ANSSI will not endorse any direct drop-in replacement of currently used algorithms in the short/medium term.** However, this immaturity should not serve as an argument for postponing the first deployments." (emphasis added)

US National Security Agency and Department of Homeland Security

2021 NSA: "NSA is waiting for the NIST process to be completed and for standards to be published. . . . NSS customers are reminded that NSA does not recommend and policy does not allow implementing or using unapproved, non-standard or experimental cryptographic algorithms. The field of quantum-resistant cryptography is no exception."

2021 DHS: **Do not use** "post-quantum cryptographic industry products until standardization, implementation, and testing of replacement products with approved algorithms are completed by NIST."

2022 NSA: "NSA does not expect to approve post-quantum algorithms with any kind of 'but just to be safe, combine with an older algorithm' guidance"—in other words, to sell post-quantum products to U.S. government, companies have to turn off ECC.

Protective measures for pre-quantum cryptography

Aka poor-man's PQC

Premise: Known/slowly changing set of peers.
This fits email, messaging (Signal, etc.), most enterprise setups.
Does not fit web servers.

Protective measures for pre-quantum cryptography

Aka poor-man's PQC

Premise: Known/slowly changing set of peers.
This fits email, messaging (Signal, etc.), most enterprise setups.
Does not fit web servers.

Requires: Adjust protocol to have user keep state per peer.

Protective measures for pre-quantum cryptography

Aka poor-man's PQC

Premise: Known/slowly changing set of peers.
This fits email, messaging (Signal, etc.), most enterprise setups.
Does not fit web servers.

Requires: Adjust protocol to have user keep state per peer.

Option 1: Have fixed secret per peer, include this in KDF.
Secret exchanged out of band, or exchange is not observed.
Provided in WireGuard VPN as option.

Protective measures for pre-quantum cryptography

Aka poor-man's PQC

Premise: Known/slowly changing set of peers.
This fits email, messaging (Signal, etc.), most enterprise setups.
Does not fit web servers.

Requires: Adjust protocol to have user keep state per peer.

Option 1: Have fixed secret per peer, include this in KDF.
Secret exchanged out of band, or exchange is not observed.
Provided in WireGuard VPN as option.

Option 2: Have updatable secret per peer, include this in KDF.
Update per-peer secret with each new public-key operation.
Initial secret exchanged out of band, or exchange is not observed.

More complicated dataflow, e.g., do not overwrite without confirmation that peer can update, but full forward secrecy.

Details worked out in RFC 6189 on ZRTP, see also section 6.2 of the ENISA report.

PQConnect: An Automated Boring Protocol for Quantum-Secure Tunnels

Daniel J. Bernstein, Tung Chou, Kai-Min Chung, Tanja Lange, Jonathan Levin, Lorenz Panny, Jon A. Solworth, Bo-Yin Yang.

Different deployment strategy

- Do not patch PQC onto existing network protocols, but add a new layer with superior security.

Different deployment strategy

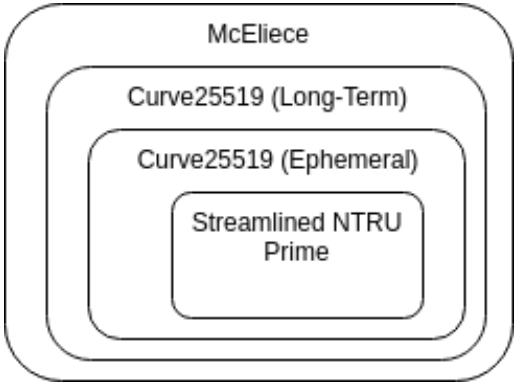
- Do not patch PQC onto existing network protocols, but add a new layer with superior security.
- Can be gradually deployed.
- Add support for VPN-like tunnels to clients and servers.

Different deployment strategy

- Do not patch PQC onto existing network protocols, but add a new layer with superior security.
- Can be gradually deployed.
- Add support for VPN-like tunnels to clients and servers.
- PQConnect is designed for security, handshake and ratcheting proven using Tamarin prover (formal verification tool).
- Use Curve25519 (pre-quantum) and Classic McEliece (conservative PQC) for long-term identity keys.
- Use Curve25519 (pre-quantum) and Streamlined NTRU Prime (PQC) for ephemeral keys.

PQConnect handshake: Nesting schemes

Most conservative system on the outside.



Attacker can see long-term Curve25519 identity key,
can break it with a quantum computer,
but cannot obtain DH value as client's share is wrapped.

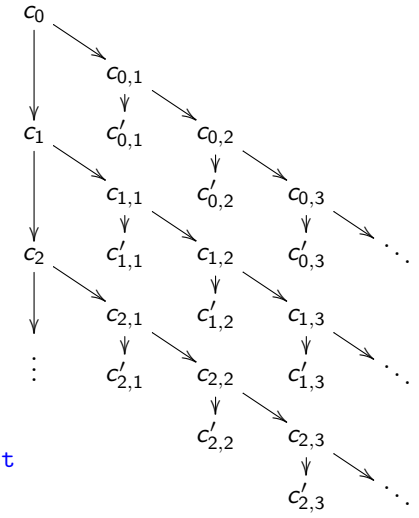
Key ratchet advances by message and time

Complete protocol follows picture on previous slide.
All systems linked together to generate initial key c_0.
Keys are updated (ratcheted) to protect against later decryption by theft of computer equipment.
Immediately advance ratchet in 3 ways:

- New epoch master key: c_1.
- New branch keys: $c_{0,1}, c_{0,2}$.
- New message key: $c'_{0,1}$.

Delete key as soon as no longer needed.
Message keys can deal with delayed transmissions.

More information on protocol:
https://research.tue.nl/en/studentTheses/pqconnect



Further information

- https://pqcrypto.org our overview page.
- PQCrypto 2016, PQCrypto 2017, PQCrypto 2018, PQCrypto 2019, PQCrypto 2020, PQCrypto 2021 with many slides and videos online.
- https://pqcrypto.eu.org: PQCRYPTO EU Project.
  - PQCRYPTO recommendations.
  - Free software libraries (libpqcrypto, pqm4, pqhw).
  - Many reports, scientific articles, (overview) talks.
- YouTube channel Tanja Lange: Post-quantum cryptography.
- https://2017.pqcrypto.org/school: PQCRYPTO summer school with 21 lectures on video, slides, and exercises.
- https://2017.pqcrypto.org/exec and https://pqcschool.org/index.html: Executive school (less math, more perspective).
- Quantum Threat Timeline from Global Risk Institute, 2019; 2021 update.
- Status of quantum computer development (by German BSI).
- NIST PQC competition.