Post-quantum cryptography

Tanja Lange

Eindhoven University of Technology; Academia Sinica

20 July 2022

Cryptography

Sender
“Alice”

Receiver
“Bob”

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.

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.

Separate designs for signature (authentication) and key establishment

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 are 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.

Achieve confidentiality, authenticity, and integrity for communicating parties.

How does TLS (`https`) work? (Example with Diffie-Hellman function)

Client | **Server**

Client: $(\mathsf{sk}_C, \mathsf{pk}_C) \leftarrow_{\$} \mathsf{KGen}$

$$\xrightarrow{\quad \mathsf{pk}_C \quad}$$

Server: $(\mathsf{sk}_S, \mathsf{pk}_S) \leftarrow_{\$} \mathsf{KGen}$

Server: $k \leftarrow \mathsf{DH}(\mathsf{sk}_S, \mathsf{pk}_C)$

$$\xleftarrow{\quad \mathsf{pk}_S \quad}$$

Client: $k \leftarrow \mathsf{DH}(\mathsf{sk}_C, \mathsf{pk}_S)$

$$\xrightarrow[\text{proves } C \text{ knows } k]{\text{stuff encrypted using } k}$$

Server: $\Sigma \leftarrow \mathsf{Sig}(\text{everything sent so far})$

$$\xleftarrow[\text{stuff encrypted using } k]{\Sigma}$$

this uses a long-term signing key

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.

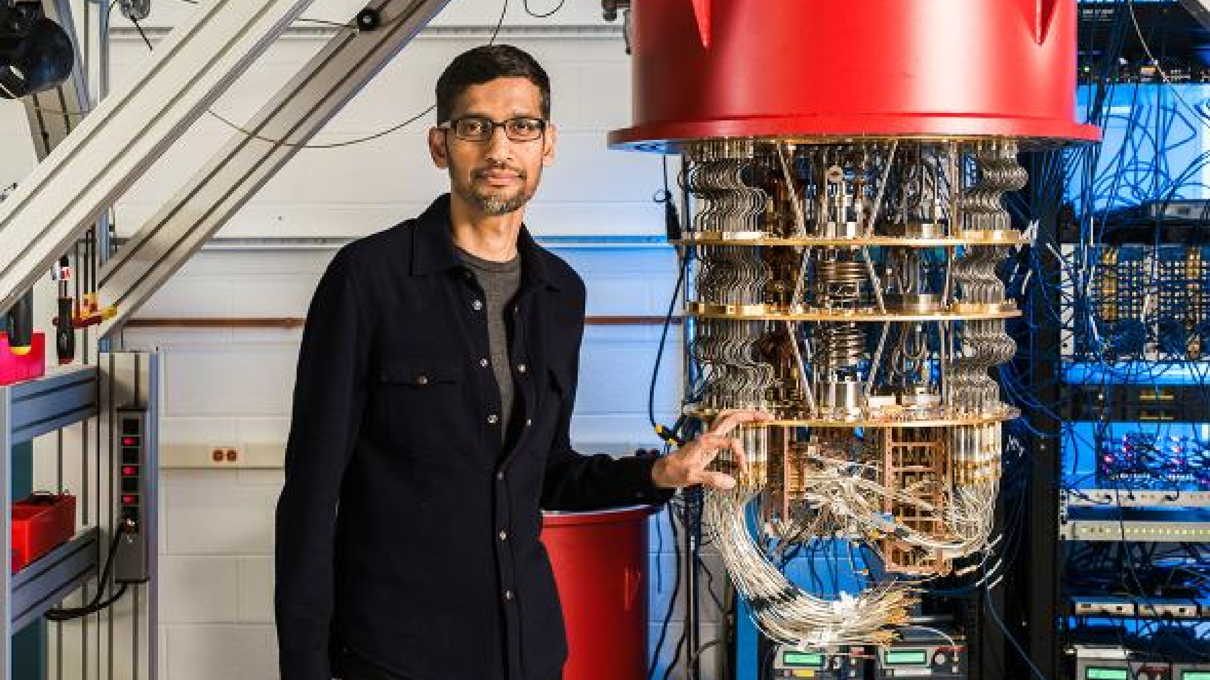
Key size recommendations

	Parameter	Legacy	Future System Use Near Term	Future System Use 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 ...

Current state of the art in applied cryptography

- Currently used crypto (check the lock icon in your browser) starts with RSA (can be broken by factoring large integers), Diffie-Hellman in finite fields, or elliptic-curve Diffie-Hellman (both require the attacker to compute discrete logarithms in some group).
- Older standards are RSA or elliptic curves from NIST (or Brainpool), e.g. NIST P256 or ECDSA.
- Internet currently moving over to Curve25519 and Ed25519
- 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.
- Some countries make ill-advised recommendations to weaken crypto.



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

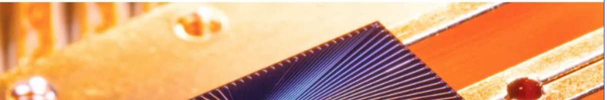
A computer is generally considered to be a universal computational device; i.e., it is believed able to simulate any physical computational device with a cost in computation time of at most a polynomial factor. It is not clear whether this is still true when quantum mechanics is taken into consideration. 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

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.

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."

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"~~: 05 Jul NIST announces first selections.
- 2022 $\rightarrow \infty$ NIST studies further systems
- 2023/2024?: NIST issues post-quantum standards.

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 algorithmic abstraction of what a quantum computer can do,
but new systems need more analysis. Any extra structure offers more attack surface.

Lorentz center

Online Workshop

Post-Quantum Cryptography for Embedded Systems

5 - 9 October 2020, Leiden, the Netherlands

How does PQC affect protocols?

- Length fields don't fit.

How does PQC affect protocols?

- Length fields don't fit.
  $\Rightarrow$ 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.
  $\Rightarrow$ 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.
  $\Rightarrow$ 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 DH,

How does PQC affect protocols?

- Length fields don't fit.
  $\Rightarrow$ 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 DH,
  $\Rightarrow$ Shoehorning PQC into current systems may
  prioritize weaker systems.

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 DH,
  ⇒ Shoehorning PQC into current systems may
  prioritize weaker systems.
- Validation and certification schemes are not updated.

How does PQC affect protocols?

- Length fields don't fit.
  $\Rightarrow$ 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 DH,
  $\Rightarrow$ Shoehorning PQC into current systems may
  prioritize weaker systems.
- Validation and certification schemes are not updated.
  $\Rightarrow$ Combine pre-and post-quantum schemes,
  certification only applies to pre-quantum scheme.

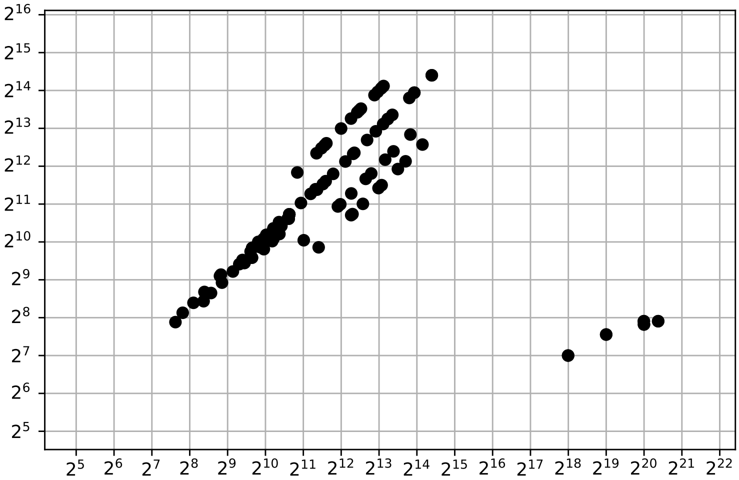
How does PQC affect protocols?

- Length fields don't fit.
  $\Rightarrow$ 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 DH,
  $\Rightarrow$ Shoehorning PQC into current systems may
  prioritize weaker systems.
- Validation and certification schemes are not updated.
  $\Rightarrow$ 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?

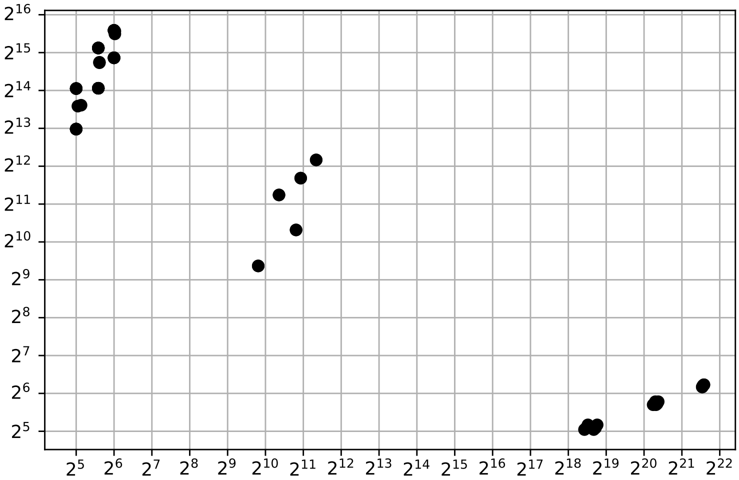
- Length fields don't fit.
  ⇒ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to PQC one,
  putting PQC part into the payload.
- Speed, resources.
  Combined schemes take about twice the time.
  Most experiments don't look so devastating.
- Interface mismatch – KEM instead of DH,
  ⇒ Shoehorning PQC into current systems may
  prioritize weaker systems.
- Validation and certification schemes are not 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)



For more graphs incl. speed comparison on many CPUs see `http://bench.cr.yp.to/results-kem.html`.
Graphs linked with every CPU.

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



For more graphs incl. speed comparison on many CPUs see http://bench.cr.yp.to/results-sign.html.
Graphs linked with every CPU.

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. item Google and Cloudflare are running some experiments of including post-quantum systems into TLS.

Further information

- 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).
- https://pqcrypto.org our overview page.
- ENISA report on PQC, co-authored.
- https://pqcrypto.eu.org: PQCRYPTO EU Project.
  - PQCRYPTO recommendations.
  - Free software libraries (libpqcrypto, pqm4, pqhw).
  - Many reports, scientific articles, (overview) talks.
- Quantum Threat Timeline from Global Risk Institute, 2019; 2021 update.
- Status of quantum computer development (by German BSI).
- NIST PQC competition.
- PQCrypto 2016, PQCrypto 2017, PQCrypto 2018, PQCrypto 2019, PQCrypto 2020, PQCrypto 2021 with many slides and videos online.