NCC Group Cryptographic Services

Post-quantum cryptography overview

August 29, 2016 – Version 1.0

Prepared by
Steffan Karger – Developer Crypto & High Security at Fox-IT

Abstract
Organisations that need to keep long-term secrets, or which are designing systems that will be in use for ten or more years, need to plan for a post-quantum-computing world. This paper gives a short introduction and overview of post-quantum cryptography. We discuss why post-quantum crypto is needed, and provide handles to determine how to plan for migration. Furthermore, we provide an overview of promising post-quantum crypto directions, and provide references for further reading.

Table of Contents

Why do we need post-quantum crypto?

Physicists continue to make progress with developing quantum computers. Once a sufficiently large quantum computer is available, it will be capable of breaking most of our current uses of cryptography. Post-quantum cryptography will survive such a 'quantum crypto-apocalypse'.

A (qu)bit about quantum computers and algorithms

As the name suggests, quantum computing makes use of quantum-mechanical phenomena to 'extend' classical computers. Classical computers operate on *bits*, which can have one of two states; 0 or 1. Quantum computers, however, operate on *qubits*, which are in a superposition of states: 0, 1, or a little bit of both. Quantum algorithms can leverage this superposition of states to provide efficient solutions to a number of problems that classical computers cannot solve efficiently. Intuitively, quantum computers do this by evaluating many solutions at once, whereas a classical computer would have to evaluate each solution individually. Scientists believe that not every problem can be efficiently solved by a quantum computer, but unfortunately much of our current cryptography is based on problems that *are* efficiently solvable by a quantum computer.

Shor and Grover – Quantum Algorithms

Two of the most famous quantum algorithms are Shor's integer factorisation algorithm and Grover's algorithm for searching an unstructured database. Peter Shor developed a quantum algorithm that factors an integer N in polynomial time in the length of N [Sho99]. This means that breaking RSA becomes just as easy as using RSA. Additionally, Shor's algorithm can be adapted to break DH / DSA (discrete-logarithm problem in finite fields) and ECDH / ECDSA (elliptic curve cryptography).

Grover's algorithm is more generic, and can also be applied to symmetric algorithms such as AES, but has a less drastic impact [Gro96]. The algorithm was introduced by analogy: "Imagine a phone directory containing N names arranged in completely random order. In order to find someone's phone number with a probability of 1/2, any classical algorithm will need to look at a minimum of N/2 names." The quantum algorithm reduces the complexity to the square root of N. This effectively halves the security level of algorithms: AES-128 drops to 64-bit security, AES-256 drops to 128-bit security.

When do we need post-quantum crypto?

Most current cryptographic protocols rely on Diffie-Hellman ('DH') or elliptic-curve DH ('ECDH') for key exchange, and on RSA or ECDSA for authentication. All of these can be broken by quantum computers. But quantum computers will not be available for many years. How much time do we have to protect ourselves from quantum computers?

To decide if and when we need to protect ourselves against these hypothetical quantum computers, we need to ask ourselves the following questions:

- How long must our secrets remain secret, or our signatures remain trustworthy? (x years)
- How long will a full transition to post-quantum cryptography take? (y years)
- When do we expect a crypto-breaking quantum computer to exist? (z years)

Which leads us to Mosca's theorem: "if $x + y \geq z$, worry" [Mos15].

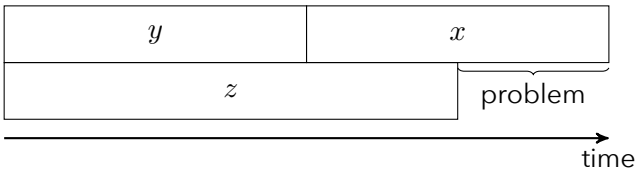

Figure 1: if $x + y \geq z$, worry.

The value of x, or how long secrets must remain secret, greatly depends on the application. The design of your yet-to-be-released product might be interesting for competitors today, but less so once you've released its successor. On the other side of the spectrum, state secrets have to be kept confidential for many years to come.

What's important to note here is that if an attacker records our (EC)DH key exchange today, she can decrypt our communications once she's acquired a quantum computer in the future. So to protect our communication, we will need to protect our key exchange well before quantum computers become available.

The argument for authentication is more subtle; if we're authenticating a key exchange, an attacker will have to break the authentication at the moment of the key exchange (i.e. we do not need post-quantum authentication of key exchanges before there are quantum computers that can break the authentication). If we're authenticating firmware images however, we'll not only want to accept fresh signatures, but also signatures from the past. So whether we'll need post-quantum signatures before quantum computers exist depends on the lifetime of the signatures.

The value of y, or how long a transition will take, is also greatly dependent on the application. In particular, it depends on:

- How long it will take until suitable post-quantum crypto is available
- How long you will need to implement and test your post-quantum crypto solution
- How long until products without post-quantum crypto are phased out (i.e. the lifetime of your product)

Estimates for z, or when a crypto-breaking quantum computer will be available, range from 'in a decade' [BWM+15] or 'at least a dozen years away'[Int15], to 'will never happen' [Kal11]. At the PQCRYPTO '14 conference, the quantum computer researcher Matteo Mariantoni of the University of Waterloo's Institute for Quantum Computing estimated that with a research budget of $1 billion, there could be a quantum

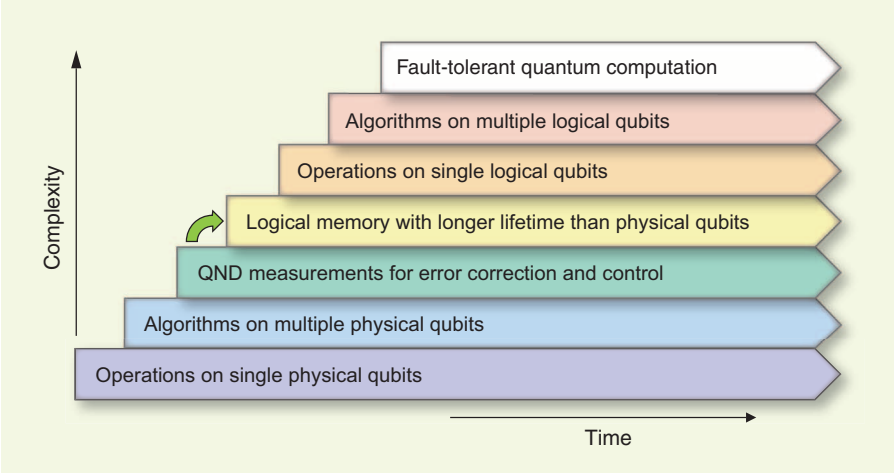

Figure 2: Seven steps towards a fault-tolerant quantum computer. The green arrow indicates the current research focus. Source: [DS13].

computer in fifteen years, breaking one RSA-2048 key per day[1](and such a quantum computer would need the equivalent of a dedicated nuclear power plant for power supply) [Mar14].

The available estimates for z are quite scarce, most likely because there are still a number of breakthroughs required, and predicting when a breakthrough will occur is hard. Figure 2 shows the steps/breakthroughs needed to build a fault-tolerant quantum computer, as identified by Devoret and Schöhlkopf[DS13]. Even though it is hard to predict when breakthroughs will happen, we do know that a lot of effort is being put into developing quantum computers. For example, intellegence agencies[IBM15], large corporations such as Intel[Int15] and Microsoft[Mic16], and coordinated research projects [EU16] are investing large sums into quantum computing (the EU project is said to invest € 1 billion; remember the figure from Mariantoni's prediction?).

So when (or even *if*) crypto-breaking quantum computers will be available is still uncertain, but we should make sure we're ready by then.

[1]Note that if we assume that current supercomputers can break one RSA-1024 per day right now, and we extrapolate computing power according to Moore's law, we can expect classical computers to break one RSA-2048 key per day in fifteen years. However, we can buy ourselves another ten years for attacks against classical computers by using RSA-3072, while that will (most likely) not suffice for attacks by quantum computers.

With the advent of quantum computers, cryptographers have been researching post-quantum cryptographic schemes that are resistant to attacks by quantum computers. Current symmetric ciphers with 256-bit keys, such as AES256 and ChaCha20, are believed to be quantum-resistant. Post-quantum cryptography research thus focuses on asymmetric algorithms, to replace RSA, (EC)DH and (EC)DSA. These post-quantum algorithms are based on mathematical problems that are believed to be difficult in the classical *and* quantum cases.[2] Leaders in the field of cryptography from both governments [CESG16] and academia [Ber16] argue that post-quantum algorithms are what we need to protect us against quantum computers.

The field of post-quantum cryptography is relatively young, and most post-quantum cryptography has not received adequate cryptanalysis to make it ready for production use; however, there are several promising areas of research that may develop into robust constructions. Several of these are detailed below.

Hash-based signatures (XMSS and SPHINCS)

Hash-based signatures are the most mature among the algorithms we discuss. Hash-based signatures are built around hashes, and have security proofs that only rely on the security of a hash function. Cryptography has used such hash functions for decades, and there is a lot of confidence in their security.

There are currently two popular hash-based signature schemes: XMSS [HBGM16] and SPHINCS [BHH+15]. XMSS is faster and has smaller signatures (around 20kB), but requires the signer to keep a state, which is very error-prone. Not properly managing the state can result in leaking the private key – something we have learned hard lessons from with ECDSA. SPHINCS on the other hand is stateless, but has larger signatures.

XMSS is currently undergoing standardisation by the IETF[HBGM16].

McEliece / Niederreiter

Introduced in 1978 [McE78], Robert McEliece created a new public-key cryptosystem based on error-correcting codes, which harnesses the "Syndrome Decoding" problem, a known hard quantum problem. McEliece uses Goppa codes, which when used with the Patterson decoding algorithm are capable of encrypting and decrypting messages. In this scheme, a large matrix and an "error margin" form a public key (analogous to a large composite number and exponent in RSA). The Niederreiter cryptosystem[Nie86] is proven to be equivalent[LDW94] (when used with Goppa codes), but slightly more efficient. McEliece/Niederreiter with Goppa codes has been quite well studied, and is considered one of the most trustworthy post-quantum algorithms. Even with Niederreiter's optimisations though, it suffers from relatively large public keys (around 1 MB). The PQCRYPTO group has recommended McEliece as a post-quantum public-key encryption replacement[ABB+15].

Over time, many variants of McEliece have been proposed, but only the variant using Goppa codes has stood the test of time. Several promising optimisations for both speed and key size, such as QC-MDPC McEliece[MTSB13], are actively researched.

(Ring) Learning with Errors

Ring Learning with Errors (RLWE) is a relatively new public-key cryptosystem introduced by Lyubashkevky et al [LPR13]. This scheme is an efficient case of the "learning with errors" problem that can be reduced to the "shortest vector problem", a hard problem for quantum computers. This scheme has spurred research in many areas, [ZZD+15, Lyu12, GSW13]. Particularly interesting is that this scheme has been extended to form an efficient Diffie-Hellman-like key exchange scheme [Pei14], with only moderately higher data transfers than

[2] Post-quantum algorithms can be executed with today's classical computers, so you don't need a quantum computer to protect you from quantum computers.

traditional Diffie-Hellman key exchange. RLWE becomes efficient by adding special structure to the private key. Cryptographers worry that adding such structure might also introduce weaknesses, but at least up to today no such weaknesses have been found.

Researchers Bos et al [BCNS15] have experimented with adding a RLWE as a post-quantum key-exchange mechanism to TLS. They modified OpenSSL and added RWLE-based key exchange. Additionally, they included a hybrid scheme that combines a RLWE and ECDH, to achieve at least the security level that either one of those offers independently. They showed that using RLWE on average took 2.1 milliseconds, whereas ECDH took 1.4. Finally, a server using a RLWE key-exchange on average would handle 20% fewer connections than a server using ECDH. These results show that while RLWE key exchange is less efficient, it's not prohibitively expensive to use.

NTRU

The NTRU algorithm can be used for encryption and signing. At a (conjectured) 128-bit post-quantum security level, NTRU uses 12881-bit keys[HPS+15]. NTRU has been available for a considerable time, and has been studied by many cryptographers, but in contrast with other post-quantum encryption schemes it lacks security proofs. Instead, it relies on not being broken after years of attempts to break NTRU (as is the case with RSA – but RSA has received a *lot* more attention from researchers). Several proposals to change NTRU to provide better security proofs exist[SS11, BCLvV16].

NTRU is standardised for use in the financial sector (IEEE Std 1363.1 and X9.98), and is a designed, implemented, and deployed algorithm, putting it far ahead of most other algorithms. But that standardisation has been driven by Security Innovations, who hold patents for NTRU. In recent years they have opened up patent-free licensing for open source applications, released an open source version of their library, and partnered with wolfSSL to offer a post-quantum-enhanced TLS library that can be used today. For proprietary uses, patent licensing is still required, but as NIST is looking to standardise post-quantum algorithms it may make sense for SI to drop that requirement to push for universal adoption of NTRU, so keep an eye out.

A note on Quantum Key Distribution

Quantum Key Distribution (QKD) does not fall into the post-quantum-crypto category, but is often mentioned as an alternative solution to the 'quantum crypto-apocalypse'. QKD promises to be able to check whether an adversary listened in on your key distribution. If no-one listened in, the transferred key can be used. QKD is more about physics, lasers, and optics than math and cryptography; and physical attacks on the system have been demonstrated. These side channel attacks[Ber16] are particularly relevant for key distribution. There is no final verdict on the future of QKD yet. Switzerland has used QKD to secure elections[Mar07], the NSA does not encourage QKD but seems to keep the option open[NSA16], and the CESG discourages QKD[CESG16]:

> QKD has fundamental practical limitations, does not address large parts of the security problem, and 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.

Comparison

Tables 1, 2 and 3 give an overview of key and signature sizes at a (conjectured) 128-bit post-quantum security level for key exchange, signature schemes, and public-key encryption schemes respectively. The schemes in **boldface** are included in the initial recommendations of the PQCRYPTO project[ABB+15].

	Public key size	Signature size	Ref.
XMSS	1732 B	3 kB	[HBGM16]
SPHINCS	1 kB	41 kB	[BHH+15]
RLWE	25 kB	19 kB	[GLP12]

Table 1: Post-quantum signatures comparison (schemes recommended by the PQCRYPTO project in boldface)

	Public key size	Ref
McEliece/Niederreiter	1 MB	[BCS13]
QC-MDPC McEliece	8 kB	[MTSB13]
NTRU	1.5 kB	[Inn16]

Table 2: Post-quantum public-key encryption (schemes recommended by the PQCRYPTO project in boldface)

	Bytes transferred	Ref.
McEliece/Niederreiter[a]	1 + 1 MB	[BCS13]
RLWE (NewHope)	2 + 2 kB	[ADPS15]

Table 3: Post-quantum key exchange comparison

[a] Based on two peers exchanging random encrypted under public keys

PQCRYPTO recommendations

The PQCRYPTO consortium is a group of universities and companies investigating post-quantum cryptographic systems. They aim to design a portfolio of algorithms that will be performant in a variety of settings (mobile, cloud, etc.). In September 2015 they released a set of conservative choices for post-quantum systems and will continue to investigate other existing schemes[ABB+15]. Their initial recommendations are:

- Symmetric Encryption: AES-256 or Salsa20 (256-bit key)
- Symmetric Authentication: GCM with typical 96-bit nonce and 128-bit authenticator or Poly1305
- Public-key Encryption: McEliece with binary Goppa codes
- Public-key Signatures: XMSS or SPHINCS-256

The PQCRYPTO recommendations are shown in boldface in the comparion tables of Section 3.

NIST

In April 2016, NIST announced the start of a standardisation effort for post-quantum algorithms:

> NIST is taking the following steps to initiate a standardization effort in post-quantum cryptography. NIST plans to specify preliminary evaluation criteria for quantum-resistant public key cryptography standards. The criteria will include security and performance requirements. The draft criteria will be released for public comments in 2016 and hopefully finalized by the end of the year. At that time NIST will begin accepting proposals for quantum-resistant public key encryption, digital signatures, and key exchange algorithms. NIST intends to select at least one algorithm providing each of these functionalities for standardization. NIST will establish a submission deadline late in 2017 for algorithms to be considered, allowing the proposals to be subject to 3 to 5 years of public scrutiny before they are standardized.

Given that there is no 'silver bullet', NIST expects to standardise multiple good choices, rather than to pick a single winner. Taking into account the three to five years of public scrutiny, and a typical standardisation track of around two years, NIST standards are expected to arrive in 2021 to 2023. Hash-based signatures might make it to a standard earlier, given that they have good security proofs and are already undergoing standardisation by the IETF.

NSA

In August 2015, the NSA announced that they prefer partners and vendors to invest in a transition to post-quantum cryptography soon, rather than to elliptic-curve cryptography now:

> For those partners and vendors that have not yet made the transition to Suite B elliptic curve algorithms, we recommend not making a significant expenditure to do so at this point but instead to prepare for the upcoming quantum resistant algorithm transition.

There is a wealth of speculation around cloak-and-dagger scenarios that may have caused the NSA to make this announcement, but the general opinion is to keep calm and carry on. The NSA later released an FAQ in which they discuss the motivations behind their announcement[NSA16]. The key take-aways are:

- The NSA waits for NIST to standardide post-quantum algorithms before requiring vendors to support such algorithms

- The NSA made this announcement in August 2015, because they previously announced that RSA would no longer be part of Suite B per October 2015. Recognising the cost of a crypto transition, they prefer vendors to spend their money on a transition towards post-quantum algorithms in a few years, rather than a transition to elliptic-curve cryptography now.

Transition period: hybrid schemes

So what if your product needs post-quantum crypto today? There are no standards yet, and for most use cases there is no silver bullet solution available. A viable solution could be to use a hybrid scheme: combine classical (standardised or trusted) crypto with post-quantum crypto to give you a solution that offers the best security of both. For example, adding both ECDSA and SPHINCS signatures to firmware images, or doing both ECDH and RLWE key exchanges (e.g. [BCNS15]). This might also be the way to add post-quantum security to products that require the use of approved algorithms, before NIST has finished standardising post-quantum schemes. Note that for such hybrid schemes it is very important that flaws in the post-quantum part do not affect the security of the classical part, and vice versa. As with any other crypto implementation, make sure to get an expert to evaluate your design and implementation.

Conclusions

Quantum computers will be able to efficiently break almost all public-key cryptography we use today. Organisations that need to keep long-term secrets or that are designing systems that will be in use for ten or more years need to plan for a post-quantum-computing world. Crypto-breaking quantum computers may still be decades away, but we need to start the transition to post-quantum cryptography well before that time. Secrets exchanged today may be exposed, and signatures of today may no longer be trustworthy when quantum computers are available. We should therefore consider which secrets of today we want to remain secret, and which signatures of today we want to remain trustworthy after quantum computers become available, and protect those now.

The field of post-quantum cryptography is relatively young, and there is no clear-cut post-quantum successor for RSA or DH yet. Typically, post-quantum algorithms have considerably larger (kilobytes to megabytes) public keys and signatures, and some require more computation than RSA/DH does. If you need post-quantum crypto today, keep an eye out for the European Union's PQCRYPTO project, which has started to provide recommendations[ABB+15]. Cryptographic consultants can help you combine post-quantum crypto with current best practices to create products that benefit from approved algorithms as well as post-quantum security. Otherwise, consider waiting for NIST to standardise post-quantum algorithms (expected around 2021-2023).

References

[ABB+15] Daniel Augot, Lejla Batina, Daniel J Bernstein, Joppe Bos, Johannes Buchmann, Wouter Castryck, Orr Dunkelman, Tim Güneysu, Shay Gueron, Andreas Hülsing, et al. Initial recommendations of long-term secure post-quantum systems. *Availab le at pqcrypto. eu. org/docs/initial-recommendations. pdf*, 2015. 6, 8, 9, 11

[ADPS15] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange-a new hope. Technical report, Cryptology ePrint Archive, Report 2015/1092, 201 5. http://eprint. iacr. org, 2015. 8

[BCLvV16] Daniel J Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. Ntru prime. 2016. 7

[BCNS15] Joppe W Bos, Craig Costello, Michael Naehrig, and Douglas Stebila. Post-quantum key exchange for the tls protocol from the ring learning with errors problem. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 553--570. IEEE, 2015. 7, 10

[BCS13] Daniel J Bernstein, Tung Chou, and Peter Schwabe. Mcbits: fast constant-time code-based cryptography. In *Cryptographic Hardware and Embedded Systems-CHES 2013*, pages 250--272. Springer, 2013. 8

[Ber16] Daniel J Bernstein. Is the security of quantum cryptography guaranteed by the laws of physics? 2016. https://sidechannels.cr.yp.to/qkd/holographic-20160326.pdf. 6, 7

[BHH+15] Daniel J Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. Sphincs: practical stateless hash-based signatures. In *Advances in Cryptology–EUROCRYPT 2015*, pages 368--397. Springer, 2015. 6, 8

[BWM+15] Bela Bauer, Dave Wecker, Andrew J Millis, Matthew B Hastings, and Matthias Troyer. Hybrid quantum-classical approach to correlated materials. *arXiv preprint arXiv:1510.03859*, 2015. 4

[CESG16] GCHQ Communications-Electronics Security Group. Quantum key distribution, 2016. https://www.cesg.gov.uk/white-papers/quantum-key-distribution. 6, 7

[DS13] Michel H Devoret and Robert J Schoelkopf. Superconducting circuits for quantum information: an outlook. *Science*, 339(6124):1169--1174, 2013. 5

[EU16] EU. European cloud initiative to give europe a global lead in the data-driven economy. Press Release, 2016. http://europa.eu/rapid/press-release_IP-16-1408_en.htm. 5

[GLP12] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In *Cryptographic Hardware and Embedded Systems–CHES 2012*, pages 530--547. Springer, 2012. 8

[Gro96] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212--219. ACM, 1996. 3

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology–CRYPTO 2013*, pages 75--92. Springer, 2013. 6

[HBGM16] A. Huelsing, D. Butin, S. Gazdag, and A. Mohaisen. Xmss: Extended hash-based signatures. IETF RFC, 2016. https://tools.ietf.org/html/draft-irtf-cfrg-xmss-hash-based-signatures-03. 6, 8

[HPS+15] Jeff Hoffstein, Jill Pipher, John M Schanck, Joseph H Silverman, William Whyte, and Zhenfei Zhang. Choosing parameters for ntruencrypt. Technical report, Cryptology ePrint Archive, Report 2015/708, 2015. 7

[IBM15] IBM. Ibm awarded iarpa grant to advance research towards a universal quantum computer. Press Release, 2015. https://www-03.ibm.com/press/us/en/pressrelease/48258.wss. 5

[Inn16] Security Innovation. Ntru open source project, 2016. https://github.com/NTRUOpenSourceProject/ntru-crypto. 8

[Int15] Intel. Intel invests us$50 million to advance quantum computing. Press Release, 2015. https://newsroom.intel.com/news-releases/intel-invests-us50-million-to-advance-quantum-computing/. 4, 5

[Kal11] Gil Kalai. How quantum computers fail: quantum codes, correlations in physical systems, and noise accumulation. *arXiv preprint arXiv:1106.0485*, 2011. 4

[LDW94] Yuan Xing Li, Robert H Deng, and Xin Mei Wang. On the equivalence of mceliece's and niederreiter's public-key cryptosystems. *IEEE Transactions on Information Theory*, 40(1):271--273, 1994. 6

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM (JACM)*, 60(6):43, 2013. 6

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In *Advances in Cryptology–EUROCRYPT 2012*, pages 738--755. Springer, 2012. 6

[Mar07] Paul Marks. Quantum cryptography to protect swiss election, 2007. https://www.newscientist.com/article/dn12786-quantum-cryptography-to-protect-swiss-election/. 7

[Mar14] Metteo Mariantoni. Building a superconducting quantum computer. Invited Talk, PQCRYPTO '14 conference, 2014. https://www.youtube.com/watch?v=wWHAs--HA1c. 5

[McE78] RJ McEliece. A public-key cryptosystem based on algebraic. *Coding Thv*, 4244:114--116, 1978. 6

[Mic16] Microsoft. Microsoft research station q. Microsoft Research website, 2016. http://research.microsoft.com/en-us/research-areas/quantum-computing.aspx. 5

[Mos15] Michele Mosca. Cybersecurity in an era with quantum computers: will we be ready? Cryptology ePrint Archive, Report 2015/1075, 2015. http://eprint.iacr.org/2015/1075. 4

[MTSB13] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo SLM Barreto. Mdpc-mceliece: New mceliece variants from moderate density parity-check codes. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 2069--2073. IEEE, 2013. 6, 8

[Nie86] Harald Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *PROBLEMS OF CONTROL AND INFORMATION THEORY-PROBLEMY UPRAVLENIYA I TEORII INFORMATSII*, 15(2):159--166, 1986. 6

[NSA16] NSA. Cnsa suite and quantum computing faq, 2016. https://www.iad.gov/iad/library/ia-guidance/ia-solutions-for-classified/algorithm-guidance/cnsa-suite-and-quantum-computing-faq.cfm. 7, 9

[Pei14] Chris Peikert. Lattice cryptography for the internet. In *Post-Quantum Cryptography*, pages 197--219. Springer, 2014. 6

[Sho99] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303--332, 1999. 3

[SS11] Damien Stehlé and Ron Steinfeld. Making ntru as secure as worst-case problems over ideal lattices. In *Advances in Cryptology–EUROCRYPT 2011*, pages 27--47. Springer, 2011. 7

[ZZD+15] Jiang Zhang, Zhenfeng Zhang, Jintai Ding, Michael Snook, and Özgür Dagdelen. Authenticated key exchange from ideal lattices. In *Advances in Cryptology-EUROCRYPT 2015*, pages 719--751. Springer, 2015. 6