The Quantum Countdown

Quantum Computing And The Future Of Smart Ledger Encryption

February 2018

Foreword

Civilisation relies on a nuanced balance between transparency and secrecy. Both transparency and secrecy are tools appropriate in different ways and combinations to achieve social goals. Computer-based trade, commerce, privacy, and defence all rely upon a combination of cryptographic techniques such as public key encryption, hashing, and digital signatures to defend data from unauthorised viewing, use, or alteration.

Quantum computers are a fascinating, nascent technology that offers significant promise in solving some intractable computing problems, including some cryptographic ones. The impending arrival of quantum computers raises the prospect of weakening, perhaps fatally, the usefulness of much current cryptography, in turn creating a post-quantum cryptography (PQC) problem.

Smart Ledgers are multi-organisational databases with a super audit trail (aka blockchains) with embedded programming and sensing, thus permitting semi-intelligent, autonomous transactions. Smart ledgers provide immutable records for the long-term and rely heavily on cryptographic techniques to function.

This report aims to explore the PQC problem from a Smart Ledger viewpoint. The intention is to help decision-makers understand when they need to act, and provide guidance on their choices. The report's conclusions are not alarmist. For certain long-term Smart Ledger applications with sensitive data, this report is a 'call to action'. Nevertheless, the conclusions do constitute a 'call to awareness' if not a 'call to action' for everyone else.

We are pleased to have sponsored this important research and do hope that the guidance herein is of help to business people, technologists, policy-makers, and regulators in considering the implications of the PQC problem.

Michael Parsons, FCA
Chairman, Cardano Foundation

Contents

Executive Summary

The post-quantum cryptography (PQC) problem will threaten the security of the world's computer networks if large-scale quantum computers become available. The problem exists because such quantum computers would be able to break the security of widely-used public key cryptography, which allows remote parties to communicate securely and authenticate transactions and data without sharing a secret key in advance. However, it is highly uncertain when (and if) such quantum computers will become available — the nearest estimates are within about 10 to 15 years.

This report focuses on how the PQC problem affects Smart Ledgers, and also addresses broader concerns for the overall internet ecosystem. Fortunately, there are good solutions to the PQC problem, and better ones are emerging. The hard questions for individual computer system operators involve when and how to address the PQC problem, given its uncertain timing and the evolving solutions.

This report explains the PQC problem in detail, with the aim of being understandable to non-technical readers, while including essential technical detail to support informed decisions on how to react to the problem. We start with the essentials of cryptography, quantum computing, and how quantum computing threatens public key cryptography. We then consider the available solutions to the PQC problem, and provide frameworks for deciding when and how to respond to it.

We conclude that the sky is not falling. However, action may be appropriate now for Smart Ledgers and other computer systems that (i) are new (to avoid later redesign), (ii) have large consequences associated with insecurity and/or (iii) require security of long duration.

"If the critical industries and government agencies don't start to pick up the pace of dealing with this problem right now, Congress and the Clinton Administration are going to have to ... deal with a true national emergency."
-- Senator Christopher J. Dodd, at the first hearings of the US Senate Special Committee on the Year 2000 Technology Problem, June 12, 1998

"Don't panic."
-- Douglas Adams, *The Hitchhiker's Guide to the Galaxy*, 1978

1. Introduction

Information technology design choices, made with all good intentions, can have unforeseen and often adverse consequences many years later. The Year 2000 (Y2K) problem, heralded by Senator Dodd's alarmist rhetoric above (such rhetoric about Y2K was common at the time), involved programming of computers with two-digit fields to hold the year. This was a perfectly reasonable design choice for software written in the 1960s or 1970s, or even the 1980s, when computer memory and storage were more expensive than they are now, and there was typically an expectation that systems would be upgraded or replaced within a decade or two (longer than we expect now). But many systems stayed around for longer, and the two-digit year fields would have been unable to cope with the turn of the millennium. The global costs of repairing systems before New Year's Eve 1999 was several hundred billion dollars[1], and the general conclusion was that most remediation was useful[2].

Other examples of such problems are legion. The Internet was originally intended for communication among US government and academic networks, resulting in an intentionally insecure design (because each connected network was trusted), which has led to many of today's cybersecurity problems. Even now, when the cybersecurity community knows better, companies around the

[1] *See* Jack Schofield, "Money we spent" (in The Millennium Bug: special report), *The Guardian* (4 January 2000).

[2] *See* Professor Martyn Thomas, "What Really Happened In Y2K?", Gresham College lecture (4 April 2017) - https://www.gresham.ac.uk/lectures-and-events/what-really-happened-in-y2k

world are rolling out a multitude of Internet of Things (IoT) devices whose limited upgradability means that there is no effective way to patch their inevitable security holes. The Mirai botnet was the first major global cybersecurity scare that exploited the vulnerability of IoT devices,[3] and it will not be the last.

This report addresses another potential widespread threat to network security — the vulnerability of public key cryptography to large-scale quantum computing — and the development post-quantum cryptography (PQC) to address this vulnerability. We call this the "PQC problem", and we focus on how it could affect Smart Ledgers. Smart Ledgers are based on a combination of mutual distributed ledgers — i.e. blockchain and related techniques — with embedded programming and sensing, thus permitting semi-intelligent, autonomous transactions.[4]

If large-scale quantum computing turns out to be viable, the PQC problem could have effects even more severe than were forecast for the Y2K problem. Although the PQC problem does not directly threaten that systems will just stop working (as many feared would occur on 1 January 2000), it poses the risk of widespread security breaches that could make it impossible to trust the many computer networks and services that depend on public key encryption for security. Over the two decades since the Y2K problem raised its head, the dependence of our society on ever-more-complex computer networks has increased steadily, and so has the risk of those networks not being available or trusted.

The potentially devastating consequences of failure of our pervasive networks is intimately linked to the 'network effect' — the phenomenon that a service becomes more valuable as more users join.[5] That is, as we become more dependent on networks, the risks of network failure increase. A familiar example involving network availability is mobile phones. The range of everyday services that people access via smartphones has grown rapidly, and

[3] *See* Lily Hay Newman, "The Botnet That Broke the Internet Isn't Going Away", *Wired* (12 September 2016).

[4] Substantial further information on Smart Ledgers and other mutual distributed ledgers is available on the Distributed Futures website at http://www.longfinance.net/programmes/distributed-futures-menu.html.

[5] Carl Shapiro & Hal R. Varian, *Information Rules: A Strategic Guide to the Network Economy* (Harvard Business Review Press: 1998).

most of us know the feeling of frustration when the mobile network goes down and it is suddenly impossible to access many services that we now take for granted. For networks that support critical societal functions — *e.g.* payments, critical infrastructure, emergency response, the global Internet — the consequences of failure are potentially much more severe.

As Smart Ledgers and similar distributed solutions become increasingly widespread — and many have suggested that such solutions are the next important stage in the evolution of the Internet — security issues for such solutions (including the PQC problem) will become an increasingly important aspect of overall cybersecurity concerns. Cybersecurity solution providers have already begun to focus on such issues for blockchain.[6]

We are not the first to draw an analogy between the PQC problem and Y2K,[7] but the analogy has limitations. Unlike Y2K, there is no hard deadline for the PQC problem. The advance of large-scale quantum computing is far from certain — indeed some believe that it will never happen. Perhaps a more appropriate analogy for the PQC problem is climate change: it is apparent that a large shift is underway, but there is great uncertainty as to the timing and nature of impacts.[8]

Fortunately, as for both Y2K and climate change, there are good available solutions for the PQC problem, and work is underway on even more robust solutions. The hard questions are about when such fixes should be applied to specific networks and systems, taking into account the uncertain timing, the value of data and other resources on the systems, and the cost of protecting them. These are questions that must be asked now, because rapid action may be required for some systems, particular under optimistic estimates of when large-scale quantum computers will be available. Indeed, it could be too late

[6] *See* Deloitte (Eric Piscini, David Dalton & Lory Kehoe), *Blockchain and Cyber Security. Let's Discuss* (2017). Others have noted the similar issues for blockchain and data protection, which is closely linked to cybersecurity. *See, e.g.,* Jakob Nielsen & Omar Hamidi, "The role of blockchain in helping organizations meet GDPR compliance", *Information Management* (22 January 2018), https://www.information-management.com/opinion/the-role-of-blockchain-in-helping-organizations-achieve-gdpr-compliance.

[7] *See, e.g.*, Alex Hutchison, "Hacking, Cryptography, and the Countdown to Quantum Computing", *The New Yorker* (26 September 2016). Our thanks to this article for prompting the idea of using "countdown" in the title of this report.

[8] We believe the PQC problem is significantly less certain in its impact than climate change.

to protect some legacy data that is stored with vulnerable encryption in a way that cannot be withdrawn from public availability. For example, transactions on existing blockchains are protected by public key encryption that is potentially vulnerable to quantum computing, and those blockchains are already widely distributed.

Each operator of a Smart Ledger or other system needs to decide what response is appropriate, and when. These issues are starting to see widespread public attention as the Y2K problem did, for example in a February 2018 article in *The Times* with the headline that quantum computers "will put every secret at risk".[9] Overall, we conclude that the sky is not falling — contrary to the conclusion of Chicken Little, and to many people as the deadline for the Y2K problem approached — and that "Don't panic" is more sensible advice.

Figure 1. Chicken Little and The Hitchhiker's Guide[10]

REMARKABLE STORY OF CHICKEN LITTLE
PUBLISHED BY
J. G. CHANDLER,
ROXBURY, Mass.
DON'T PANIC
The Hitchhiker's Guide to the Galaxy

[9] Tom Whipple, "Quantum leap for computers will put every secret at risk", *The Times* (3 February 2018).

[10] Sources https://howlingpixel.com/wiki/Henny_Penny (1840 Chicken Little title page); https://www.pinterest.co.uk/pin/210824826276657996/ (Hitchhiker's Guide).

Section 2 of this report sets the technical background for the PQC problem. Section 3 discusses the evolving options for developing quantum-resistant cryptography to address the problem. Section 4 explores the timeline for action, focusing on specific risks for Smart Ledgers. Section 5 offers a set of recommendations.

We have two overall goals in this report. First, we aim to provide a clear explanation of the PQC problem for non-technical decision makers. Second, we seek to summarise essential technical detail that makes clear the nature of the PQC problem and its solutions, in order to support informed decisions on how to react to the problem.

2. Defining the PQC Problem - Background and Key Concepts

Understanding the PQC problem requires background on three technical issues:

- how modern digital cryptography works, including its key building blocks;
- the progress of quantum computing, and the basics of how it works; and
- how quantum computing threatens the viability of certain types of cryptography, including the possible effect of these vulnerabilities for Smart Ledgers and other applications.

A. How Cryptography Works

First, some definitions. The Oxford Living Dictionary defines ‘cryptography’ as “the art of writing or solving codes”, and ‘encryption’ more narrowly as “the process of converting information or data into a code, especially to prevent unauthorized access” — although these two terms are often used interchangeably (and we do so in this report in contexts where such usage is common in computer security circles). Cryptography techniques are often called ‘encryption algorithms’, and a ‘cryptosystem’ is a combination of multiple algorithms to accomplish a specific security task. More generally, an ‘algorithm’ is “a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer”.[11] These definitions are set out in Table 1 for reference.

Table 1. Cryptography Terms

Term	Definition
algorithm	a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer
cryptography	the art of writing or solving codes
cryptosystem	a combination of multiple encryption algorithms to accomplish a specific security task
encryption	the process of converting information or data into a code, especially to prevent unauthorized access (sometimes used loosely as a synonym for

[11] *Oxford Living Dictionary* (Oxford University Press), https://en.oxforddictionaries.com/.

	'cryptography')
encryption algorithm	an algorithm used for cryptography
key	data (usually a small amount of data) used to ensure secrecy during a particular use of an encryption algorithm (see further explanations below)
post-quantum cryptography, PQC	cryptography that remains secure if and when large-scale quantum computers are available

Cryptography is an ancient art. An early example is Julius Caesar's "Caesar cipher", which involves shifting each letter of a message some number (between 1 and 25) of letters forward in the alphabet. Cryptography has evolved steadily over the centuries, driven by military and diplomatic uses, although there has also been widespread use of cryptography for commercial purposes and by amateur enthusiasts.[12]

Over time, code-writing techniques have become more complex, making it more and more difficult for encrypted messages to be read by people other than the intended recipients. The development of cryptography has involved a cat and mouse game between those developing algorithms, and those looking for ways to 'break' or 'attack' algorithms. The discipline of attacking cryptography is known as 'cryptanalysis'.

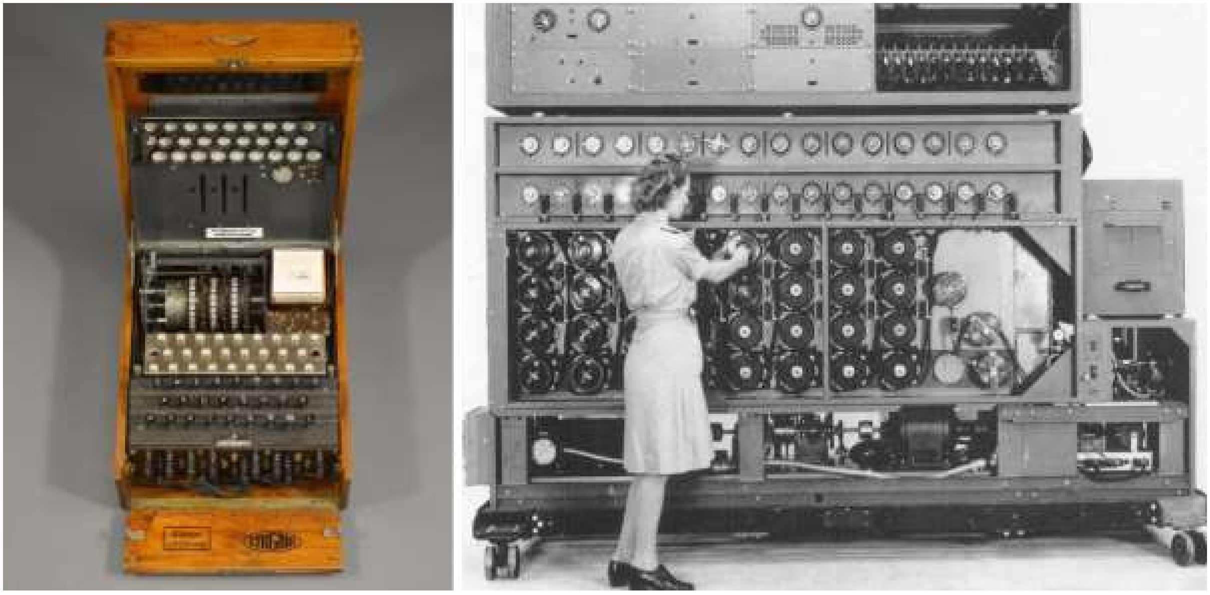
For most of its history, cryptography has involved 'symmetric algorithms' where two communicating parties use the same secret 'key'. With symmetric encryption, one party encrypts a message using the secret key with a known (and often publicly-available) mathematical technique, and the other party decrypts the message by reversing the technique using the same key. The secrecy of the message thus depends upon the secrecy of the key.

Breaking a symmetric algorithm generally requires an exhaustive search for the correct key, and attackers often use large-scale computing resources to conduct such searches. The goal of secure cryptography is to make such a search so expensive in computing power and time that it is infeasible or not cost-effective.

[12] *See, e.g.*, Katherine Ellison, *A Cultural History of Early Modern English Cryptography Manuals* (Routledge: 2017).

Perhaps the most famous examples of symmetric encryption are the Enigma machines used by the German military during World War II. The breaking of the Enigma encryption by a British team at Bletchley Park led by Alan Turing, recently popularised in the 2015 film *The Imitation Game*, was a major contribution to the success of the Allied war effort.

Figure 2. An Enigma machine, and a 'bombe' at Bletchley Park used to break the Enigma code[13]



An important challenge of symmetric encryption is how to secretly share keys between the communicating parties without the keys being intercepted. For Enigma, the Germans distributed sheets or books of daily keys, and a couple of captures of key sheets from a German submarine and a weather ship were instrumental to the Allied codebreaking effort.[14]

Cryptography took a huge leap beyond symmetric encryption in the late 20th century with the discovery of "public key cryptography". This invention was first made secretly in the early 1970s at UK Government Communications Headquarters (GCHQ), although the GCHQ invention may never have been

[13] https://pocketbookuk.files.wordpress.com/2015/02/bombeenigmamachines.jpg.

[14] *See* "How did the Germans distribute Enigma keys during WW2?", https://www.quora.com/How-did-the-Germans-distribute-Enigma-keys-during-WW2.

implemented at the time, due to the computing resources then available.[15] The widespread use of public-key cryptography was prompted by its public discovery, notably a 'key exchange' algorithm (*i.e.* for sharing secret keys) described by Whitfield Diffie and Martin Hellman in a 1976 paper.[16]

Public key cryptography involves use of a mathematically-related 'public key' (which can be known to anyone) and 'private key' (which must be kept secret by its owner, like a symmetric encryption key). A message encrypted with the public key can only be read by the holder of the private key, and vice versa, so that public key cryptography can be used for both secure communication and digital signature (as illustrated in Table 2).

When used for confidentiality, public key cryptography could be compared to the recipient (the holder of the private key) making an infinite number of open padlocks available around the world, where each padlock can be snapped onto a message to protect it in transit to the recipient, who is the only one who can open it (with the private key). For Smart Ledgers, the most common uses of public key cryptography are for digital signatures, which authenticate the right of a user to make transactions, or to access documents or other data or software stored on a ledger.

[15] *See* Christian Lawson-Perfect, "GCHQ has declassified James Ellis's papers on public key cryptography", *The Aperiodical* (20 March 2016), http://aperiodical.com/2016/03/gchq-has-declassified-james-elliss-papers-on-public-key-cryptography/; GCHQ, "GCHQ's Public Key Cryptography pioneer receives prestigious award" (6 May 2014), https://www.gchq.gov.uk/news-article/gchqs-public-key-cryptography-pioneer-receives-prestigious-award; GCHQ, "A Note on 'Non-secret Encryption'", https://www.gchq.gov.uk/note-non-secret-encryption.

[16] *See* "Public key cryptography", *Wikipedia,* https://en.wikipedia.org/wiki/Public-key_cryptography.

Table 2. How Public Key Cryptography Works

Technique	Sender Uses	Recipient Uses	Why It Works
Public key secure communication	Recipient's public key	Recipient's private key	Only recipient (using her private key) can read messages encrypted with her public key
Public key digital signature	Sender's private key	Sender's public key	Only sender can sign with her private key, and recipient can use her private key to confirm signature

Public key encryption works because the mathematical relationship between the public and private keys is based on a 'hard problem', which is relatively easy to calculate in one direction but very difficult in a different direction. For example, a hard problem based on common experience is a variant of the 'knapsack problem', which involves choosing from among a set of items with varying weights with the aim of filling a theoretical knapsack to its weight capacity. This is a very difficult optimisation challenge. However, determining the weight of the knapsack for a given selection of items is easy (you weigh it).[17]

The two purely mathematical hard problems that underpin the most widely used public key cryptography algorithms are *integer factorisation* and *calculation of discrete logarithms*. The integer factorisation problem involves that fact that it is relatively easy to multiply two large prime numbers A and B to calculate C, but much more difficult given C to determine the two factors A and B.

The discrete logarithm problem is based upon modular arithmetic, where "X mod Y" means remainder when X is divided by Y. A simple example of modular arithmetic is 'clock' arithmetic, where Y = 12 and the remainder is the hour shown when the hour hand has travelled X hours from midnight (and gone

[17] There is a knapsack cryptosystem, developed in 1978, based on this problem, but it proved to be insecure several years later. *See* Andrew Ellinor, Nathan Landman, Eddie The Head & Mahindra Jain, "Knapsack Cryptosystem", *Brilliant*, https://brilliant.org/wiki/knapsack-cryptosystem/.

through midnight multiple times). Specifically, the discrete logarithm problem is that, in properly-constructed cases:

- it is relatively easy to calculate a = b^x mod c (given b, x and c), but
- calculating x in the same equation (given a, b and c) is much more difficult.

A working, large-scale quantum computer would be able to substantially reduce the difficulty of solving these hard problems (as we discuss below). That is the reason that the PQC problem exists.

In addition to symmetric algorithms and public key algorithms, modern cryptography uses a third technique called a 'hash algorithm', which reduces data of arbitrary length (*e.g.* a password, a JPEG image, an executable computer program, *War and Peace*) to a fixed-length 'digest' (typically around 256 bits, or 32 characters). The security of hash algorithms depend on the fact that any change to the underlying data (*e.g.* a change of a single letter in *War and Peace* or a single pixel of an image) produces a change in the digest, so that the change in the data can be detected by the fact that the hash digest of the altered data will not match the digest of the original data.[18] Hash algorithms are crucial to Smart Ledgers, because they guarantee the authenticity of a blockchain by including the digest of each block within the next block. Fortunately, hash algorithms are not as vulnerable to the PQC problem as are public key algorithms.

[18] With a properly-designed hash algorithm, it is not feasible to find an alteration to the data that does not alter the digest — known as a 'collision' — given the very large number of possible digests.

Table 3. Main Algorithms Types Used for Cryptography, and Uses For Smart Ledgers[19]

Type of Algorithm	General Use	Example Algorithms of This Type	Example Uses for Smart Ledgers
Symmetric	Secret communications	AES, DES, 3DES, RC4	Protection of resources stored on ledger
Public key	Secret communications (including key exchange) or digital signature	RSA, Diffie-Hellman, El Gamal, ECDSA	User authentication; signature of transactions, data or software
Hash	Generating fixed-length digest of arbitrary-length text	SHA-256, SHA-512, SHA-3	Ensuring authenticity of blockchain

Table 3 provides a summary of the main types of algorithms used for modern cryptography.[20] Algorithms are linked together to form ‘encryption protocols’ that perform the wide variety of security-enabled tasks required by modern computer networks. For example, one very common encryption protocol that uses all of the algorithms in the table below is the Secure Sockets Layer / Transport Layer Security (SSL/TLS) protocol used to protect communications between an Internet browser and a website (when SSL/TLS is enabled, modern browsers typically display a padlock icon or similar symbol to show that the connection is secure — see Figure 3).

[19] This table does not include all basic building blocks of cryptography protocols — known as ‘primitives’. Another important class of primitives is pseudo-random number generators, which generate random “seed” values for other algorithms.

[20] This is not an exclusive list. Other types of algorithms that are important for modern cryptography include pseudo-random number generators (which provide random “seeds” for other algorithms), hash-based message authentication code algorithms (which combine a hash algorithm with a secret key) and others.

Figure 3. A Browser 'Padlock' Icon

Secure | https://www.google.co.uk

B. The Progress of Quantum Computing

Quantum mechanics, which is one of the key discoveries of 20th century physics, describes how atoms and sub-atomic particles behave at very small scales. The idea of a quantum computer was first articulated by Nobel prize-winning physicist Richard Feynman in a 1981 speech at MIT, in which he considered the ability of computers to simulate the phenomena of quantum mechanics.[21] Since Feynman's speech, there has been significant progress on the theory and practice of quantum computing. The first demonstration of a quantum computer took place at Oxford University in 1998, and this has been followed by steady, if slow, progress towards quantum computers that are useful in practice.[22]

Quantum computers depend on two phenomena of quantum mechanics: *superposition* and *entanglement*:

- Superposition means that a particle has multiple states at the same time. In a classical digital computer, each bit of data is either 0 or 1. By contrast, a quantum bit (known as a 'qubit') has both values 0 and 1 *at the same time*, with a probability distribution defining which value it will have when observed.
- Entanglement between two particles means that the quantum state of each particle cannot be described independently of the state of the other particle, even though the two particles are separated by a (potentially large) distance.[23]

[21] Richard P. Feynman, "Simulating Physics with Computers", *International Journal of Theoretical Physics*, Vol. 21, Nos. 6/7 (1982) (text of May 1981 keynote speech at First Conference on the Physics of Computation); *see* Emma Strubell, "An Introduction to Quantum Algorithms", p. 3 (2011), https://people.cs.umass.edu/~strubell/doc/quantum_tutorial.pdf (noting that Feynman originated the idea of a quantum computer).

[22] "Timeline of quantum computing", *Wikipedia*, https://en.wikipedia.org/wiki/Timeline_of_quantum_computing.

[23] *See* "Quantum Superposition and Entanglement Explained", https://www.clerro.com/guide/491/quantum-superposition-and-entanglement-explained.

These phenomena are strange even to scientists, and hard to understand intuitively, because they do not correspond to the way the world behaves in ordinary human experience. Indeed, Albert Einstein, perhaps the most celebrated scientist of the 20th century, never accepted quantum mechanics, famously commenting (with regard to superposition and related concepts) that "God does not play dice with the universe". Einstein called entanglement "spooky action at a distance". Nevertheless, quantum mechanics has achieved extensive experimental verification in recent decades, and the heavy majority of scientific opinion (with some exceptions[24]) concludes that these quantum phenomena do exist.

Together, superposition and entanglement mean that a quantum computer behaves very differently from a classical digital computer. In a classical computer, eight bits of memory (or one 'byte', often corresponding to one character) can hold any of $2^8 = 256$ different values. In a quantum computer, eight entangled qubits hold all 256 values at the same time, and a program running on the computer could theoretically determine in a single step which of the 256 states is most likely.

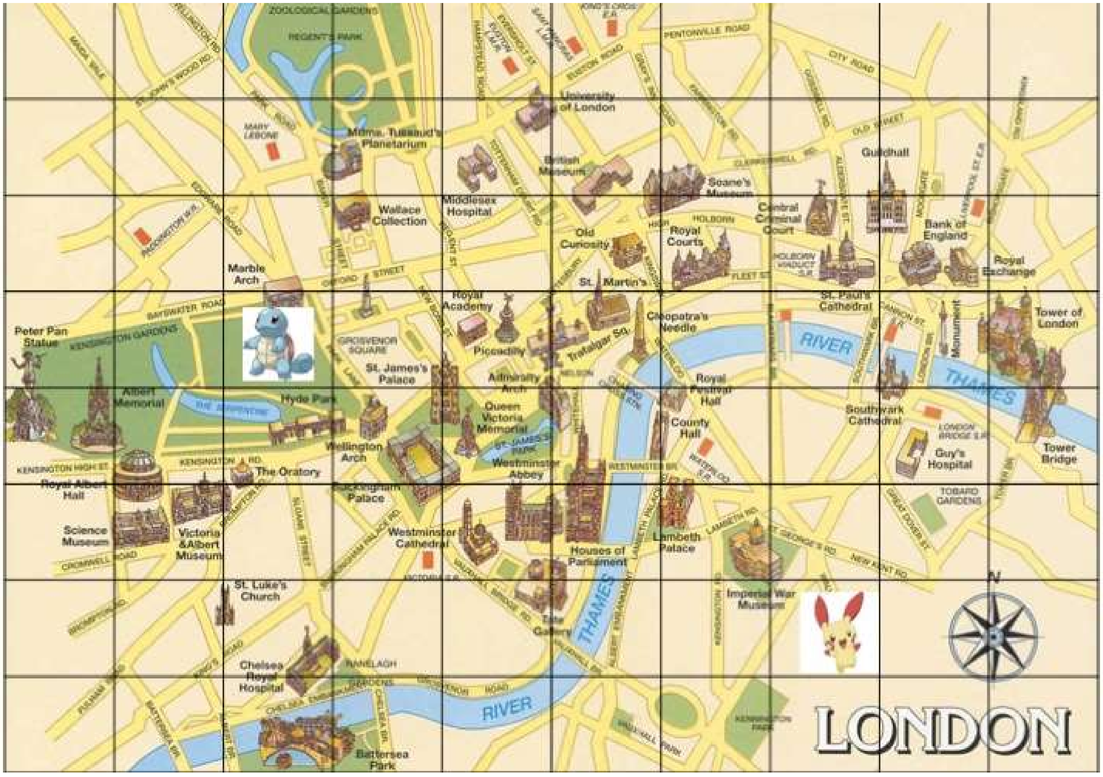
More generally, quantum computers can theoretically greatly accelerate the solution of certain problems of *exponential* complexity where the problem involves a distribution of probabilities across the potential solutions to the problem. Exponential complexity means that the difficulty of the problem increases with a variable exponent (*e.g.* 2^x, where x is the number of variable features of the problem). Such problems can become computationally intractable for conventional digital computers as complexity increases. Finding a solution becomes like finding the single best footpath through a forest so gigantic that it would take many human lifetimes to explore.

To take a concrete (and somewhat silly) example, imagine that a probability density function exists that gives the likelihood of finding characters in the popular Pokémon mobile game in any location in central London. The probability might be sampled at various locations in a grid covering the city, as

[24] The most prominent advocate of the position that quantum phenomena have classical origins is Professor Ross Anderson of the Cambridge Computer Laboratory. *See, e.g.,* Ross Anderson, "Emerging, fascinating, and disruptive views of quantum mechanics", *Light Blue Touchpaper* (blog) (28 October 2015), https://www.lightbluetouchpaper.org/2015/10/28/emerging-fascinating-and-disruptive-views-of-quantum-mechanics/.

in Figure 4. A viable quantum computer would be able to simultaneously assess probabilities across the entire grid, and quickly advise an eager player that the most promising hunting locations (in this example) are in Hyde Park and southeast of the Imperial War Museum in Lambeth.

Figure 4. Fictional Pokémon Game (as an illustration of quantum computing capabilities)



This Pokémon task could of course be solved with a traditional computer (your mobile phone, for example), but a large-scale quantum computer would have advantages solving similar problems over more complex data sets. For example, this technique could be useful for machine learning, since inference on large data sets is very computationally demanding. This has prompted interest in quantum computing from leading machine learning companies such as Google, IBM, and Intel.[25]

[25] *See, e.g.,* Greg Robinson, "AI and quantum algorithms together can computer a better world", *VentureBeat* (23 October 2017), https://venturebeat.com/2017/10/23/ai-and-quantum-algorithms-together-can-compute-a-better-world/; "Launching the Quantum Artificial Intelligence Lab", Google Research Blog (16 May 2013), https://research.googleblog.com/2013/05/launching-quantum-artificial.html.

To date, however, quantum computers cannot solve any problem more efficiently than a classical computer, and there remain huge engineering challenges to building quantum computers at a scale that would trigger the PQC problem[26] (which we refer to in this report as a 'large-scale' quantum computer). Probably the most difficult challenge is 'decoherence' — *i.e.* that qubits fall out of the state of quantum superposition when they interact with the surrounding environment[27] — and time to decoherence is very short with current technologies, making current quantum qubits too unstable for most practical applications. For example, IBM recently announced that it has built a 'prototype' 50-qubit quantum computer (an industry record) with a record-high decoherence time of 90 microseconds (less than one ten-thousandth of a second).[28] But these improved capabilities are still limited in practical terms.

The ultimate goal is to build quantum computers with a large number of stable 'logical qubits' that can be used to reliably run quantum programs — analogous to the way your laptop computer runs your Internet browser when you need it. Scientists are working toward this goal, through the problem of decoherence, by two main routes. First, they are engineering quantum computers whose qubits have a high degree of physical isolation (and therefore relative stability), using a wide variety of physical architectures generally involving ultra-low temperatures and some means to isolate a single atom or particle (the details of these architecture are beyond the scope of this report[29]). Second, they can use 'quantum error correction', which allows a large number of 'physical qubits' (*i.e.* individual atoms or particles in superposition) to underpin a single logical qubit, by discarding physical qubits as they decohere.

Figure 5 shows key milestones that have been reached in terms of numbers of physical qubits (no system has achieved a stable logical qubit). Progress from the first operating quantum computer in 1998 was slow until late 2017, when

[26] For a summary of these engineering challenges, see M. H. Devoret & R. J. Schoelkopf, "Superconducting Circuits for Quantum Information: An Outlook", *Science*, Vol. 339 (8 March 2013).

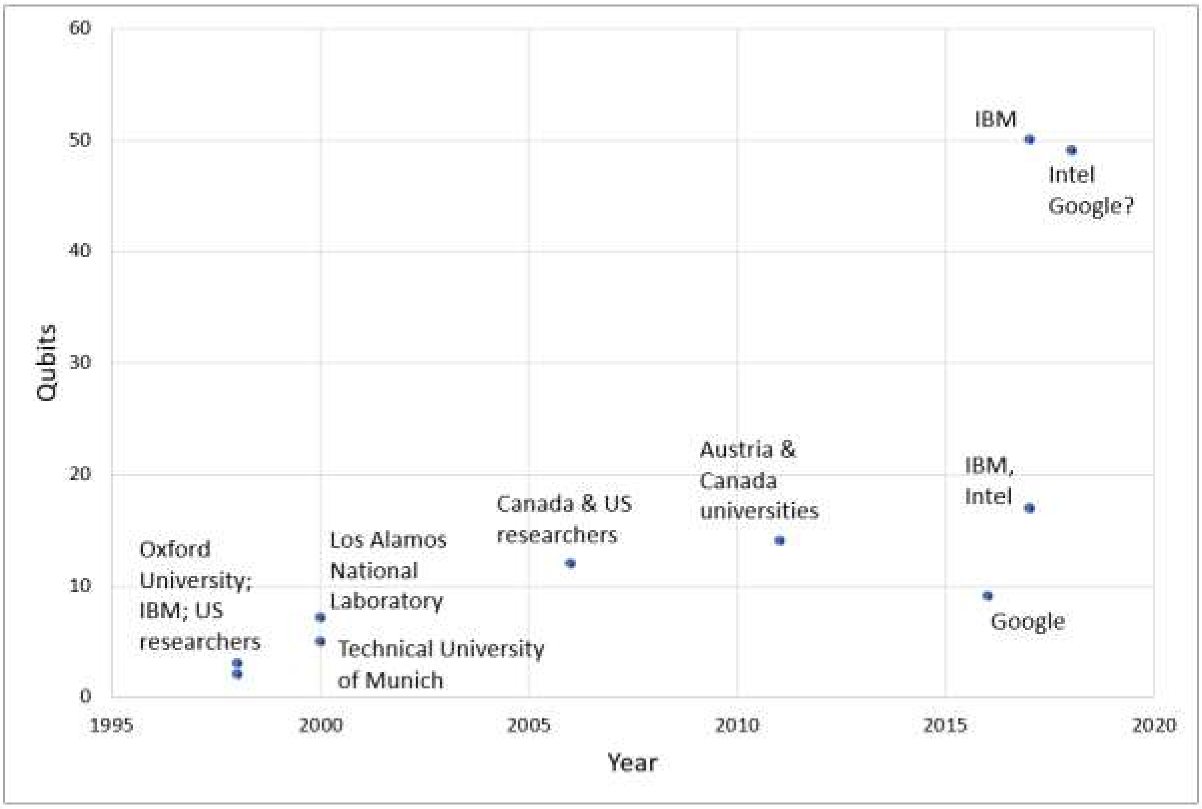
[27] Decoherence is related to Heisenberg's uncertainty principle, which establishes limits on the measurability of quantum states. *See* "Uncertainty principle", *Wikipedia*, https://en.wikipedia.org/wiki/Uncertainty_principle.

[28] Will Knight, "IBM Raises the Bar with a 50-Qubit Quantum Computer", *MIT Technology Review* (10 November 2017).

[29] For a list of such architectures, *see* "Quantum computing (Developments)", https://en.wikipedia.org/wiki/Quantum_computing#Developments.

IBM announced its 50-qubit computer and Google announced a plan to build a 49-qubit computer.[30] Then in January 2018 Intel joined the fray by announcing its own 49-qubit chip.[31] It is too soon to say whether this recent jump in physical qubit numbers is the beginning of an acceleration in quantum computing capacity, and we therefore do not attempt to project future trends from the data.

Figure 5. Progress of Quantum Computers[32]



This progress has been associated with some modest demonstrations of practical applications of quantum computing. For example, in 2001

[30] "Google Just Revealed How They'll Build Quantum Computers", *Futurism* (6 October 2017), https://futurism.com/google-just-revealed-how-theyll-build-quantum-computers/.

[31] Michael Feldman, "Intel Reveals 49-Qubit Quantum Computing Chip", *Top500* (11 January 2018), https://www.top500.org/news/intel-reveals-49-qubit-quantum-computing-chip/.

[32] Sources: "Timeline of quantum computing", *Wikipedia,* https://en.wikipedia.org/wiki/Timeline_of_quantum_computing. The chart does not include quantum annealing analog devices from D-Wave Systems, which have achieved much higher numbers of qubits (2000 qubits as of January 2017, with a larger device being built), but are not general-purpose digital quantum computers suitable for the type of computation that causes the PQC problem. There has also been controversy regarding whether the D-Wave computers are actually quantum devices.

researchers at IBM and Stanford for the first time used a quantum computer to factor an integer (one of the key tasks that raises the PQC problem), calculating that 15 = 3 x 5[33]; and in 2012 Chinese researchers claimed to have factored 143 (it's 13 x 11) using a 4-qubit quantum computer.[34] But a conventional computer can factor a number this size in a tiny fraction of a second. In 2017, IBM made an interface available that allows the public to experiment with running algorithms on IBM's quantum computers[35]; but such experiments cannot yet address any practically useful task.

Despite current enthusiasm for quantum computing progress, there remains doubt whether ongoing progress will ever lead to large-scale quantum computing. Beyond the considerable engineering challenges associated with decoherence, a small but vocal minority of scientists continue to share Einstein's doubts whether quantum mechanics expresses a complete model of reality,[36] with the implication that quantum computers are misunderstood simulators of natural probability distributions. Paul Davies, a researcher in Australia, has suggested that entanglement of more than 400 qubits would violate limits on the total amount of information in the universe, with the implication that above this level (which is too low to pose the PQC problem) decoherence would be inevitable and any benefit of quantum computing would necessarily disappear.[37]

In short, quantum computing has the promise to provide revolutionary computational benefits, but it is far from certain that these benefits will ever be realised. This uncertainty regarding the viability of large-scale quantum

[33] Lieven M. K. Vandersypen, "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance", *Nature*, Vol. 414 (20/27 December 2001).

[34] Stephen Battersby, "Controversial quantum computer beats factoring record", *New Scientist* (13 April 2012), https://www.newscientist.com/article/dn21699-controversial-quantum-computer-beats-factoring-record/. The Chinese computer was intended to be an "adiabatic" quantum computer that uses quantum effects to find the minimum value of a function, and did not run Shor's algorithm (which is described in section 1.C). There was also controversy whether the computer actually exhibited quantum effects.

[35] IBM Q Experience, https://quantumexperience.ng.bluemix.net/qx/experience.

[36] *See, e.g.*, David Hestenes, "Hunting for Snarks in Quantum Mechanics", Arizona State University (December 2009), http://geocalc.clas.asu.edu/pdf/SnarkPaper.pp.pdf. These challenges go to the root of quantum mechanics as a whole, not just the PQC problem.

[37] Paul Davies, "The Implications of a Cosmological Information Bound for Complexity, Quantum Information and the Nature of Physical Law", *arXiv* (6 March 2007), https://arxiv.org/abs/quant-ph/0703041/

computing is the primary uncertainty affecting decisions on how to react to the PQC problem — because the PQC problem will not become real if large-scale quantum computers never become viable.

C. Methods for Attacking Cryptography with Quantum Computers

Bell Laboratories, which was founded by AT&T and has passed through various other owners following the break-up of AT&T in the 1990s,[38] has played an outsized role in development of the ideas that have created the PQC problem. Both of the two key mathematical algorithms known to provide useful ways to break (or 'attack') the security of current cryptography using quantum computers were developed by Bell Laboratories researchers.

Shor's Algorithm

In 1994, Peter Shor (then a Bell Laboratories researcher and now a professor at MIT) discovered that there is a relationship between the hard problem of integer factorisation and a "period finding" problem that can be solved efficiently by a quantum computer.[39] The period finding problem involves the following function:

$f(x) = a^x \bmod N$

where N is the number being factored, and 'a' is a random integer smaller than N.

Over the integers *x*, the result of this function is a series of integers that repeats with a period of *p* integers. A quantum computer can calculate *p* much more efficiently than a classical computer. This technique (which also can solve the hard problem of calculation of discrete logarithm) is now known as "Shor's algorithm", and it is at the root of the PQC problem. Indeed, Shor's discovery was a key factor in stimulating current interest in quantum

[38] *See* NOKIA Bell Labs, History, http://www.bell-labs.com/explore/history-bell-labs/.

[39] *See* P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", *SIAM Journal on Computing*, 26(5), 1484-26 (1997), http://dx.doi.org/10.1137/S0097539795293172; Stephanie Blanda, "Shor's algorithm — Breaking RSA Encryption", American Mathematical Society blog (30 April 2014), https://blogs.ams.org/mathgradblog/2014/04/30/shors-algorithm-breaking-rsa-encryption/; "Shor's algorithm", *Wikipedia*, https://en.wikipedia.org/wiki/Shor%27s_algorithm.

computing, because it demonstrated the utility of quantum computers to solve a practical problem.

The mathematics of Shor's algorithm has been proven robust — *i.e.* the algorithm would certainly work to attack public key cryptography if a large-scale quantum computer were available. So the central questions regarding use of Shor's algorithm are:

- How much quantum computing resource is required to implement Shor's algorithm to attack successfully modern public key cryptography?
- When could such resource be available?

On the first question, the US National Institute of Standards and Technology (NIST) estimates that 3,000 to 5,000 logical qubits would be required to defeat 2,048-bit RSA, the most frequently-used public key cryptography algorithm.[40] Michele Mosca, who is introduced more fully later, gives a similar estimate of 5,000 logical qubits, also noting that between 1 million and 1 billion physical qubits (with the top estimate likely to decrease over time) would be required to implement that number of logical qubits.[41] While these numbers of qubits are very large, technology tends to scale rapidly as it is perfected. For example, a two-terabyte disk drive can store over 16 *trillion* bits of information and sells at retail for about £50 — there has been more than a ten million fold decline in the cost of disk storage over the past 35 years.[42]

On the second question of when such resources could be available, there is substantial uncertainty (as discussed above). The most optimistic estimates are that such resources could be available as early as 10 to 15 years from now.

Grover's Algorithm

The other method using for quantum computers to speed up attacks on cryptography was developed in 1996 by Lov Grover, who like Peter Shor was a researcher at Bell Laboratories. The method, known as "Grover's algorithm",

[40] Interview with Dustin Moody, NIST PQC Lead (28 November 2017).

[41] Michele Mosca, "The quantum threat to payment systems", presentation at The Payments Summit Canada 2017 (May 2017) ("Mosca 2017 Presentation"), https://www.youtube.com/watch?v=3SkVJsGyUWc&t=5s.

[42] *See, e.g.,* "A history of storage cost (update)" (March 2014), http://www.mkomo.com/cost-per-gigabyte-update.

speeds up to attacks on cryptography techniques that do not rely on hard problems — specifically symmetric and hash algorithms. Attacking such algorithms usually requires a search through all possible solutions, known as a 'brute force' attack.

Grover's algorithm is an efficient technique for searching a database of N elements in random order (*e.g.* finding a specific telephone number in the phone book), which is mathematically equivalent to a brute force attack on cryptography.[43] Such a search using a traditional computer requires on average N/2 steps, while Grover's algorithm allows the search to be performed in a number of steps proportional to the square root of N.[44] This speed up, while significant, is not nearly as dramatic as Shor's algorithm provides for public key cryptography. As a result, there are relatively simple ways for symmetric and hash algorithms to avoid the PQC problem by using keys or digests of adequate length.

D. The PQC Problem: Threats to Smart Ledgers and Other Applications

We now explain the practical threats these attacks present for Smart Ledgers and other applications. There are two important caveats about the (lack of) completeness of our discussion of practical threats in this section. First, the specific threats associated with the PQC problem vary significantly by technology and business model and it is not feasible for us to identify the threats that will apply to all technologies or all business models. Second, the threat environment will continue to evolve along multiple dimensions *e.g.* changing Smart Ledger technologies, developing quantum computing platforms, and likely discovery of new quantum computing algorithms that expose new vulnerabilities. For both of these reasons, the threats identified in this section should be viewed as examples of significant threats, rather than anything approaching an exhaustive list.

[43] Lov K. Grover, "A fast quantum mechanical algorithm for database search", *arXiv* (29 May 1996), https://arxiv.org/abs/quant-ph/9605043.

[44] Grover's algorithm is related to a category of quantum algorithms known as "quantum random walks". To date, there does not appear to be any quantum random walk algorithm that provides a speed up to brute force attacks on cryptography that is more substantial than that of Grover's algorithm. *See, e.g.*, Andris Ambainis, "Quantum walk algorithm for element distinctiveness", *SIAM Journal on Computing*, 37(1) (2007), https://arxiv.org/abs/quant-ph/0311001; PQCRYPTO: Post-Quantum Cryptography for Long-Term Security, Initial recommendations of long-term secure post-quantum systems (7 September 2015) ("PQCRYPTO Initial Recommendations").

We consider four kinds of threats. The first three involve the structure, transactions and stored data/software of distributed ledger (aka blockchain) architectures, since the substantial majority of Smart Ledgers currently being used or developed are based on some form of blockchain architecture. However, there will be other important ledger technologies in the future,[45] which is both an additional threat (because different ledger technologies may present new vulnerabilities) and a genuine opportunity (because new ledger technologies can be designed from the outset to avoid or resist the PQC problem). We finally consider briefly the more general threats beyond blockchain.

Threats to Distributed Ledger (aka Blockchain) Architecture

The basic architecture of most blockchains uses encryption in two ways — a hash algorithm to ensure integrity of the overall blockchain, and a digital signature algorithm to authenticate new transactions. For example, Bitcoin uses the SHA-256 hash algorithm and the elliptic curve digital signature algorithm (ECDSA),[46] and Ethereum uses the KECCAK-3 hash algorithm and ECDSA.[47] ECDSA has become a *de facto* signature standard for popular public blockchains.[48]

Use of the SHA-256 and KECCAK-3 hash algorithms (and similar modern hash algorithms) should not be vulnerable to quantum computing. As a result, the PQC problem does not seem to include the risk that historical blockchains can be altered. However, this does not mean the historical blockchain is invulnerable to unexpected use, as we discuss later.

Cryptocurrency mining based on the 'proof of work' model used for Bitcoin also does not appear significantly vulnerable to the PQC problem, because it

[45] For example, the IOTA cryptocurrency developed in Germany relies on a mathematical "tangle" instead of blockchain. *See* Mike Orcutt, "A Cryptocurrency Without a Blockchain Has Been Built to Outperform Bitcoin", *MIT Technology Review* (14 December 2017).

[46] *See* Bitcoin Developer Reference, https://bitcoin.org/en/developer-reference; Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", https://bitcoin.org/bitcoin.pdf (2008).

[47] *See* Ethereum wiki, https://github.com/ethereum/wiki/wiki.

[48] *See* Lionello Lunesu, "A Tale of Two Curves", Enuma Technologies blog (1 November 2016), http://blog.enuma.io/update/2016/11/01/a-tale-of-two-curves-hardware-signing-for-ethereum.html.

involves repeated execution of hash algorithms[49] (it is hashing for mining that drives nearly all of the huge energy consumption of the Bitcoin ecosystem[50]). In theory, Grover's algorithm offers a speed-up for mining by using quantum computers — and bitcoin miners invest heavily in mining speed-up — but the extent of investment required would be very large, and even if made would not affect the security of the underlying blockchain (and would just make mining easier). The vulnerability of other methods for validating blockchain transactions (such as 'proof of stake') will need to be assessed for each specific approach.

Threats to Distributed Ledger Transactions

The critical aspect of the PQC problem for blockchain architectures involves signature based on public key cryptography. ECDSA is based upon the discrete logarithm problem, and is therefore vulnerable to Shor's algorithm. This means that it will be possible to spoof new blockchain transactions using ECDSA if sufficiently powerful quantum computers are available. The same problem would apply to accessing existing blockchain assets, such as amounts in cryptocurrency wallets, if new security measures are not introduced in time. For example, a Bitcoin transaction uses the sender's ECDSA signature to transfer bitcoins from one address to another, so vulnerability of the signature equals vulnerability to theft of the underlying assets. Solutions will certainly emerge — at least if bitcoins are still valuable in 10 years or so — to transfer the bitcoins or their value into systems that are resistant to the PQC problem. However, it is not clear what the solution would be for bitcoin wallets that are not easily transferred. For example, it is not clear what would happen to the bitcoins owned by Bitcoin founder Satoshi Nakamoto — worth approximately $7 billion (and falling) at the date of this report[51] — if Satoshi is dead (as many have speculated)?[52] If bitcoins retain their value in a couple of decades, a bad

[49] *See* Nakamoto, note 466, p. 3.

[50] *See, e.g.,* Chris Mooney & Steven Mufson, "The Bitcoin craze is using up so much energy", *Independent* (30 December 2017), http://www.independent.co.uk/life-style/gadgets-and-tech/the-bitcoin-craze-is-using-up-so-much-energy-a8118486.html.

[51] *See* "People Keep Sending Satoshi Nakamoto Bitcoin", Bitcoin.com (24 December 2017), https://news.bitcoin.com/people-keep-sending-satoshi-nakamoto-bitcoin; CoinMarketCap, https://coinmarketcap.com/ (current Bitcoin prices).

[52] *See* "If Satoshi Nakamoto is dead, then will his 1 million Bitcoin go out of circulation forever?", https://www.quora.com/If-Satoshi-Nakamoto-is-dead-then-will-his-1-million-Bitcoin-go-out-of-circulation-forever.

actor with access to a large-scale quantum computer might be able easily to deduce Satoshi's private key from his public key, and steal this buried treasure.

Threats to Data and Software Stored on Ledgers

Many Smart Ledger applications assume that data stored on a ledger will be protected from being accessed by those without authorisation, now and in the future. That assumed security extends to stored information, as well as rights to execute or access software (*e.g.* smart contracts on Ethereum).

The vulnerability of data and software on a ledger to the PQC problem will depend heavily on the specific security architecture. For example, it should remain possible to protect the confidentiality of blockchain data by encrypting it using a robust symmetric algorithm like AES-256. At the other end of the spectrum, applications that use digital signatures that are vulnerable to Shor's algorithm would face the same vulnerability noted earlier. For example, the 'smart contract' capability of Ethereum provides a flexible programming language that supports multi-signature capabilities,[53] so any Ethereum application using digital signatures for authentication would need to be scrutinised carefully to assess whether it is affected by the PQC problem.

In summary, some of the likely risks to blockchain-based Smart Ledger architectures once suitably powerful quantum computers are available (to the extent weaknesses associated with the PQC problem are not addressed or cannot be repaired) are shown in Table 4, with areas of modest risk shown in yellow and areas of primary risk shown in red.

[53] Ethereum White Paper, Code Execution (originally published 2013), https://github.com/ethereum/wiki/wiki/White-Paper#code-execution.

Table 4. Risks to Blockchain Architectures from Quantum Computing

	Transactions	Data on Blockchain	Software on Blockchain
Read historical records without authorization	No (blockchains are intended to allow access to transaction information)	No, unless confidential and secured with vulnerable cryptography	No, unless confidential and secured with vulnerable cryptography
Alter historical records	No	No	May be able to run software without authorisation if signature used
Spoof ongoing records	Yes, possibly	Yes, possibly	Yes, possibly

General Threats Beyond Smart Ledgers

It is important to remember that the PQC problem affects the entire Internet ecosystem, so is of potential concern to every entity operating online whether they use Smart Ledgers or not. Likewise, Smart Ledgers typically operate either over the Internet or over private networks that use cryptography to ensure their security, and ensuring the reliability of such networks is critical to the reliability of Smart Ledgers.[54]

For example, the SSL/TLS protocol used to secure Internet browser connections typically relies on the RSA algorithm for key exchange and/or signature, and RSA is vulnerable to Shor's algorithm. The risk of pervasive insecurity from vulnerabilities in SSL/TLS is huge, as illustrated by the widespread concern over the Heartbleed bug (a flaw in the OpenSSL library that is widely used to implement SSL/TLS).[55] A detailed study of other vulnerabilities of Internet communications protocols to the PQC problem can

[54] It is possible that some Smart Ledgers may operate over private networks that use physical isolation from the Internet and other networks to ensure security. But even in this unusual, high-security situation, it would be advisable not to ignore the PQC problem, given the heightened security requirements that ordinarily exist for such isolated networks. In practice, achieving genuinely isolated networks is extremely difficult.

[55] *See, e.g.* Jane Wakefield, "Heartbleed bug: what you need to know", *BBC News* (10 April 2014), http://www.bbc.co.uk/news/technology-26969629; http://heartbleed.com/.

be found in a June 2015 report by the European Telecommunications Standards Institute (ETSI).[56] Attention to such potential vulnerabilities will be crucial as the PQC problem become clearer and nearer.

[56] ETSI, “Quantum Safe Cryptography and Security: An introduction, benefits, enablers and challenges”, ETSI White Paper No. 8 (June 2015) (“ETSI PQC Paper”).

3. Changing Cryptography to Address the PQC Problem

An important and fortunate aspect of the PQC problem is that good solutions to the problem are known, and even better solutions are being developed. Because these solutions already exist, the PQC problem is not unavoidable, but rather requires careful analysis of whether and when existing solutions should be applied.

A. Developing and Testing Secure Cryptography

The solutions discussed below are intended to provide protection against the currently known attacks on cryptography. This does not mean that such attacks will remain the state of the art, or that the solutions described here will remain secure. There are many historical examples of cryptography that was believed to be secure turning out to have serious vulnerabilities.

For example, the MD5 and SHA-1 hash algorithms, which were once considered secure enough for widespread use on the Internet, have been found to have significant vulnerabilities to non-quantum attacks.[57] The severe weaknesses of MD5 played a key role in the 2012 Flame malware attacks. Even more spectacularly (and highly relevant to the PQC problem), the cryptography-based Estonian national ID card system was temporarily suspended in late 2017 when it was discovered that the computer chips on the ID cards generated insecure keys for the otherwise secure RSA algorithm.[58] The poor choice of keys meant that solving the integer factorisation problem for the cards (and stealing identities) would have been fairly easy for a knowledgeable attacker — much as the PQC problem would make possible for all uses of the RSA algorithm. The government and its consultants were able to re-secure the cards by using a different encryption algorithm based on

[57] MD5, which was first deployed in 1992, is now known to have severe vulnerabilities. *See* "MD5", *Wikipedia*, https://en.wikipedia.org/wiki/MD5. SHA-1, which was deployed in 1995 based upon a design by the US National Security Agency and standardisation by NIST, has less severe currently know vulnerabilities but has been broken at least once by Google, in a demonstration that produced two dissimilar PDF files with the same SHA-1 digest. *See* "SHA-1", *Wikipedia*, https://en.wikipedia.org/wiki/SHA-1.

[58] *See* "Security flaw force Estonia ID 'lockdown'", *BBC News* (3 November 2017), http://www.bbc.co.uk/news/technology-41858583; Cybernetica Case Study: Solving the Estonian ID-card Case (13 December 2017), https://cyber.ee/en/news/cybernetica-case-study-solving-the-estonian-id-card-case/.

elliptic curve cryptography that was supported by the defective chips.[59] But the event significantly damaged the credibility of the ID card system.

In short, the PQC problem is not a fundamentally new challenge. Securing encryption systems has long been an arms race between cryptography designers and attackers, and that arms race will certainly continue. However, what makes the PQC problem unusual is the scope of insecurity that it would exist if large-scale quantum computing is available. That is why, even though such quantum computing capabilities are very likely a decade or more away, it is important to plan responses to the PQC problem now. Early action is particularly crucial because the time required both (i) to develop and prove the security of new cryptosystems and (ii) to transition to those new systems is significant.

It is fundamental to modern cryptography that a new algorithm or cryptosystem is best tested by making its details public for thorough analysis by the cryptography community over a period of years (even with such scrutiny, vulnerabilities tend to slip through, as discussed above). This approach is grounded in "Kerckhoffs's principle" that cryptography methods must be secure even though their details are fully known.[60]

Because the specific mathematics used by different encryption algorithms varies widely, the security of algorithms is generally compared in terms of standardised 'security level', which is expressed in terms of bits (in the length of a symmetric encryption key) or the number of operations required to break the encryption. For example, a security level of 128 bits or 2^{128} (which are equivalent, using these two alternate formulations) is generally considered sufficient for at least a few decades, although this level may increase later as both traditional and quantum computers advance.[61] The calculation of security level differs by type of algorithm, as follows.

Symmetric Algorithms

[59] *See* Cybernetica Case Study.
[60] *See* "Kerckhoff's principle", *Crypto-IT*, http://www.crypto-it.net/eng/theory/kerckhoffs.html. This principle, formulated by 19th century Dutch cryptographer Auguste Kerckhoffs, requires that security of a cryptographic communication be guaranteed only by the security of the private or secret key.
[61] *See* NIST, Report on Post-Quantum Security, NISTIR 8105 (April 2016) ("NIST PQC Report") (noting recommended transition from 112-bit to 128-bit security by 2031); PQCRYPTO Initial Recommendations (recommending PQC algorithms based upon providing 2^{128} security).

The security level of a symmetric algorithm generally is the same as the key length. This assumes that there is no known faster way to guess the key than a brute force attack (*i.e.* random guessing). This is the case for any good symmetric algorithm.

Hash Algorithms

The security level calculation for hash algorithms depends largely on the probability of finding multiple input texts that produce the same hash (known as a 'collision'). This again assumes a brute force attack to produce a collision, with the likelihood of collisions affected by the "Birthday attack". The name of this attack refers to the counterintuitive phenomenon that in a group of 23 people, the chance that any two people have the same birthday exceeds 50% (and the probability continues to increase with more people — *e.g.* to about 95% with 50 people).[62] Similarly, the chance of a collision between two hash digests is equal to approximately $1.25 * 2^{d/2}$, where d is the digest length. This means that the security level is roughly half the digest size.[63]

Public Key Algorithms

Determining the security for public key algorithms is more complex. Essentially, it involves determining how difficult it is to solve the hard problem underlying the algorithm, given the length of the public and private keys. For example, the European Union Agency for Network and Information Security (ENISA) has estimated for the RSA algorithm that a key size of 3,072 bits provides a 128-bit security level. That is, although the problem of integer factorisation underlying RSA is a hard one, it is still much easier than a brute force attack for equivalent key size.

A key step in proving the security of a public key algorithm is to demonstrate that the hard problem is equivalent to some other mathematical problem that is known to be solvable only in non-polynomial time (this means that the time-to-solve is an exponential function of complexity). Such problems fall into

[62] *See* Daniel Miessler, "The Birthday Attack", Daniel Miessler blog (28 June 2014), https://danielmiessler.com/study/birthday_attack/.

[63] *See* Arjen K. Lenstra, "Key Lengths: Contribution to The Handbook of Information Security", at 12-14, https://infoscience.epfl.ch/record/164539/files/NPDF-32.pdf.

three main categories (in ascending order of difficulty): NP, NP-complete and NP-hard.[64]

Furthermore, even for NP problems, not every instance of the problem is equally hard — put differently, 'average-case hardness' may be significantly lower than 'worst-case hardness'. When implementing a public key algorithm, it is important to avoid key choices for which hardness is known to be significantly below worst-case hardness. For example, the failure to address this issue caused the weakness in the Estonia ID card system that is described above (*i.e.* that easy-to-attack RSA keys were chosen by the computer chip on the cards).

B. Quantum Resistant Symmetric and Hash Algorithms

Existing symmetric and hash algorithms that are in widespread use can provide good resistance to the PQC problem, provided that they are deployed with secret keys (for symmetric algorithms) or digests (for hash algorithms) of sufficient length.

For symmetric algorithms, the widely-used AES algorithm, which is defined in a NIST standard, is available with key lengths of 128, 192 and 256 bits.[65] If large-scale quantum computers become available, the security level will reduce to half of the key length — since Grover's algorithm reduces the number of steps for a brute force attack by the square root of the number of steps (and the square root of 2^x equals $2^{x/2}$). Accordingly, AES-256 will provide adequate 128-bit security even with availability of large-scale quantum computers, while AES-128 and AES-192 likely will not. Table 5 summarises the specific security levels.

[64] The complexity of NP problems is a difficult mathematical issue, the details of which are well beyond the scope of this report. For a simple explanation, see Larry Hardesty, "Explained: P vs. NP", *MIT News* (29 October 2009), http://news.mit.edu/2009/explainer-pnp. NP-complete problems are the overlap between NP problems and NP-hard problems. *See* "What are the differences between NP, NP-Complete and NP-Hard?", https://stackoverflow.com/questions/1857244/what-are-the-differences-between-np-np-complete-and-np-hard

[65] Specification for the Advanced Encryption Standard (AES), US Federal Information Processing Standards Publication 197 (26 November 2001).

For hash algorithms, the SHA-2 and SHA-3 algorithms are available with digest sizes of 224, 256, 384 or 512 bits.[66] Unlike a symmetric algorithm, the improved search speed provided by Grover's algorithm does not halve the security level (from one half to one quarter of the digest size), but instead reduces the security level to one third of the digest size.[67] Accordingly, in the presence of large-scale quantum computing, the SHA2-384 and SHA3-384 hash algorithms provide a 128-bit security level, while the SHA-256 algorithm (which is another name for SHA2-256) used by Bitcoin and KECCAK-256 algorithm (very similar to SHA3-256) used by Ethereum provide only an 85-bit security level.[68]

[66] Secure Hash Standard (SHS), US Federal Information Processing Standards Publication 180-4 (March 2012) (specifying SHA-2); SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, US Federal Information Processing Standards Publication 202 (August 2015).

[67] *See* Gilles Brassard, Peter HØyer & Alain Tapp, "Quantum cryptanalysis of hash and claw-free functions", Latin American Symposium on Theoretical Informatics (25 May 2006), https://link.springer.com/chapter/10.1007%2FBFb0054319.

[68] Bitcoin uses double invocation of SHA-256 to deal with the possibility that SHA-256 will in the future be found to have weaknesses similar to those that have affected the SHA-1 hash algorithm. *See for comparison* "Hashcash", Bitcoin wiki, https://en.bitcoin.it/wiki/Hashcash. However, use of double SHA-256 (compared to single SHA-256) does not increase the difficulty of the collision attack based on Grover's algorithm.

Table 5. Security Levels of Symmetric and Hash Algorithms

Symmetric Algorithm	**Security Level Now (= Key Length)**	**Security Level with Large-Scale Quantum Computing (= ½ Key Length)**
AES-128	128	64
AES-192	192	96
AES-256	256	128
Hash Algorithm	**Security Level Now (= ½ Digest Length)**	**Security Level with Large-Scale Quantum Computing (= ⅓ Digest Length)**
SHA2-256 or SHA3-256	128	85
SHA2-384 or SHA3-384	192	128
SHA2-512 or SHA3-512	256	171

An 85-bit security level for hash algorithms is likely be sufficient for many Smart Ledger purposes in the medium term. It would be much more difficult than suggested by this security level to generate collisions across multiple blockchain blocks, and multiple collisions would be necessary to alter an historical blockchain (at least with most standard implementations, such as those of Bitcoin and Ethereum). However, it would be more conservative for future blockchain implementations to consider using hash algorithms with at least 384-bit digests, particularly considering that blockchains are intended to provide a permanent immutable record. Further research on this issue would be appropriate.

An important consequence of the relative resistance of symmetric and hash algorithms to quantum computing is that these types of algorithms can often be used in combination with public key algorithms to reduce the overall vulnerability of a system to the PQC problem. This strategy of 'defence in depth' is discussed in more detail below.

C. Quantum-Resistant Public Key Cryptography

Public key cryptography, as we have repeated, is at the heart of the PQC problem. Any public key algorithm that is vulnerable to Shor's algorithm will not be secure if large-scale quantum computers are developed (as discussed in sections 1.A and 1.C above). Such vulnerabilities extend to the widely-used algorithms RSA, ECDSA, El Gamal, Diffie-Hellman and others. For these algorithms, security level will be essentially irrelevant (*i.e.* there will be little or no guarantee of security) in a PQC environment, because quantum computers would be able to solve the underlying hard problems in a small number of steps.

Fortunately, for the public key algorithms that are vulnerable to the PQC problem, there are good current alternatives and more are under development. We refer to such alternatives as 'quantum-resistant' rather than 'quantum-safe', since there is no guarantee that any public key algorithm will be entirely invulnerable to later advances in quantum computing, traditional computing, or mathematics.

In September 2015, the EU-funded Post-Quantum Cryptography for Long-Term Security (PQCRYPTO) project published initial recommendations for which public key algorithms should be favoured in current PQC implementations.[69] The PQCRYPTO report recommends that the Extended Merkle Signature Scheme (XMSS) or SPHINCS-256 (both of which are hash-based algorithms — see section 2.C below) be used for public key signature, and that McEliece (a code-based algorithm — see section 2.C.) be used for public key secure communications. For those sensitive applications that may wish to consider transition to PQC now, the PQCRYPTO recommendations are useful guidance.

Of much broader interest than PQCRYPTO, NIST in the United States is currently running a competition, which began in December 2016, to identify the most promising existing and new quantum-resistant algorithms.[70] NIST recently published 69 submissions from teams around the world (with some

[69] PQCRYPTO Initial Recommendation.

[70] NIST, Announcing Request for Nominations for Public-Key Post-Quantum Cryptographic Algorithms (20 December 2016), https://csrc.nist.gov/news/2016/public-key-post-quantum-cryptographic-algorithms.

contributors involved in multiple submissions).[71] The final output of this competition (which will likely take 3 to 5 more years to complete[72]) will be highly influential, particularly because NIST has been responsible for the most important previous standardisations of cryptography algorithms (including for AES and SHA-3 discussed above).

Another important consideration in choosing new public key algorithms is whether a particular algorithm or associated architecture is subject to patent protection. Patent-protected algorithms raise difficult issues of balancing between the interest in low-cost availability of encryption algorithms for widespread use, and the incentives for invention that patent rights provide. NIST has limited any broad assertion of patent rights for winning algorithms:

> *NIST has observed that royalty-free availability of cryptosystems and implementations has facilitated adoption of cryptographic standards in the past. For that reason, NIST believes it is critical that this process leads to cryptographic standards that can be freely implemented in security technologies and products. As part of its evaluation of a PQC cryptosystem for standardization, NIST will consider assurances made in the statements by the submitter(s) and any patent owner(s), with a strong preference for submissions as to which there are commitments to license, without compensation, under reasonable terms and conditions that are demonstrably free of unfair discrimination.*[73]

This bias toward publicly-available algorithms is certainly influenced by controversies in the 1980s and 1990s over the patent for the RSA algorithm, which significantly increased the cost for use of the algorithm until the patent expired in 2000.[74]

Against this background, the main 'families' of public key algorithms that are believed to be quantum-resistant are described below, with a summary in

[71] NIST, Post-Quantum Cryptography: Round 1 Submissions, https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions.

[72] Interview with Dustin Moody, NIST PQC Lead (28 November 2017).

[73] NIST, "Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process", p. 9 (December 2016), https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf.

[74] *See* Ann Harrison, "RSA encryption patent released", *Computerworld* (18 September 2000), https://www.computerworld.com/article/2588444/security0/rsa-encryption-patent-released.html.

Table 6. All of these algorithms use some version of the public/private key approach. For each category, the table indicates whether the family of algorithms is suitable for each of (a) signature, (b) secure communication and (c) key exchange. Signature and secure communication were described earlier. Key exchange is a sub-category of secure communication, in which the encrypted information is a secret key to be used for further communications with a symmetric algorithm (since a symmetric algorithm is nearly always much computationally less demanding and therefore faster than a public key algorithm). The NIST competition should provide substantial further clarity on which of these types of algorithms are most promising for PQC.

Table 6. Quantum-Resistant Public Key Cryptography Algorithms[75]

Cryptography Method	Suitable for		
	Signature	Secure Communication	Key exchange
Lattice – *e.g.* NTRU, BLISS, GGH, LWE, Ring-LWE	Yes	Yes	Yes
Hash-based signature – *e.g.* Lamport, Merkle	Yes	No	No
Code-based – *e.g.* McEliece, Niederreiter	Possibly	Yes	Yes
Multivariate – *e.g.* UOV, Rainbow	Yes	Maybe (efforts to date broken)	Maybe (efforts to date broken)
Supersingular elliptic curve isogeny	Maybe (new)	Maybe (slow)	Yes

Below are more detailed explanations of these families of algorithms. These explanations are necessarily technical, so readers without technical background may wish to skip the remainder of this section. These details do not add significantly to the key message for purposes of addressing the PQC problem: that there are many possible public key algorithms that can provide

[75] Sources: NIST PQC Report, Interview with Dustin Moody, NIST PQC Lead (28 November 2017); "Post-quantum cryptography", *Wikipedia*, https://en.wikipedia.org/wiki/Post-quantum_cryptography.

replacements for current public key algorithms that are vulnerable to quantum computing.

Lattice-Based Cryptography

Lattice-based algorithms involve calculations over a 'lattice' – *i.e.* an n-dimensional group of vectors that are the sum of integer multiples of the n 'basis' vectors that define the lattice. A two-dimensional lattice can be viewed as a regular tiling of points on a plane, and higher-dimension lattices take the same form, with additional dimensions added.[76]

There are several lattice-based hard problems that can be used for cryptography; for example, the shortest vector problem involves finding the shortest vector in the lattice, given the basis vectors. Several lattice problems have been shown to be NP-hard, and in addition have the advantage that average-case hardness is substantial.[77]

Lattice-based algorithms are potentially useful for signature, confidentiality and key exchange, and the flexibility and proven security of lattice-based algorithms make them a leading candidate for post-quantum cryptography. The oldest (and best-tested) lattice-based algorithm, NTRU, was subject to a US patent,[78] which appears to have impeded its usage; however, this patent expired in August 2017,[79] and several variations of NTRU have been submitted in the NIST competition.

Hash-Based Signature

Hash-based signatures rely on the security of hash algorithms (rather than on a mathematical hard problem),[80] and the public key is the digest produced as the final output of a 'tree' of hash operations. This tree-based approach was

[76] *See generally* PQCRYPTO, "Lattice-based public-key cryptography", https://pqcrypto.org/lattice.html.
[77] *See* "Lattice problem", *Wikipedia*, https://en.wikipedia.org/wiki/Lattice_problem.
[78] "Public key cryptosystem method and apparatus", US patent 6081597 (published 27 June 2000), https://www.google.com/patents/US6081597.
[79] *See* Tweet by Daniel J. Bernstein (@hashbreaker) (19 August 2017), https://twitter.com/hashbreaker/status/898997506410938369.
[80] More specifically, hard problems are more difficult in one direction than the other, while hash functions are "one-way" functions that are not reversible.

invented in 1979 by Ralph Merkle, and has been proven to be secure.[81] The major drawback of this approach is that the number of signatures that may be associated with a given public key is limited by the size of the tree, and increasing the size of the tree also increases the size of the public and private keys.

Because of the one-way nature of hash functions, hash-based public key cryptography is useful for signature, but not for confidentiality or key exchanges. As noted above, PQCRYPTO has recommended hash-based algorithms as the best currently-available option for quantum-resistant signature.

Code-Based Cryptography

Code-based cryptography is based upon the use of forward error correction codes for detection of noise-induced errors in communication channels. This technique has been used for communications since 1950.[82] Application of the approach to cryptography, first proposed in 1978, is well-studied and believed to be secure.[83] To date, code-based cryptography has been used primarily for secure communications and key exchange, although it is also considered a possible candidate for digital signature.[84]

Code-based cryptography does have the significant disadvantage that key sizes must be very large to provide adequate security. For example, the McEliece algorithm (which is recommended by PQCRYPTO for secure communications) requires keys 277 kilobytes in length to provide a 120-bit security level.[85] Efforts are being made to reduce such key lengths, although such reductions also can reduce security.[86]

[81] *See* PQCRYPTO, "Hash-based post-quantum cryptography", https://pqcrypto.org/hash.html; "Hash-based cryptography", *Wikipedia*, https://en.wikipedia.org/wiki/Hash-based_cryptography.
[82] "Forward error correction", *Wikipedia*, https://en.wikipedia.org/wiki/Forward_error_correction.
[83] *See* NIST PQC Report, p. 9; PQCRYPTO, "Code-based public-key cryptography", https://pqcrypto.org/code.html.
[84] Interview with Dustin Moody, NIST PQC Lead (28 November 2017).
[85] *See* Nicolas Sendrier, "Code-based cryptography", PQCRYPTO presentation to ECRYPT-CSA Executive School on Post-Quantum Cryptography, slide 18 (2017). Forward error correction codes (which do not need to be secure against intentional attack) are much shorter than keys for code-based cryptography.
[86] *See* NIST PQC Report, p. 9.

Multivariate Cryptography

Multivariate cryptography is based upon the hard problem of solving multivariate systems of equations. Various multivariate signature schemes have withstood substantial scrutiny, although security proofs so far appear less robust than for lattice-based and hash-based cryptosystems.[87] Furthermore, there are several patents protecting multivariate cryptosystems.[88] Multivariate encryption has not yet been found effective for secure communication or key exchange.

Supersingular Elliptic Curve Isogeny

This family of quantum resistant encryption algorithms is based upon the hard problem of calculating relationships between mathematical functions called 'supersingular' elliptic curves.[89] The approach is so far useful only for key exchange (and not signature or general secure communications). Like the widely-used Diffie-Hellman key exchange algorithm (which is vulnerable to quantum computing), supersingular elliptic curve isogeny has the significant advantage of providing 'perfect forward secrecy' — which means that compromise of the security of the private key does not affect the secrecy of communications made after the compromise.[90]

D. Quantum Key Distribution

Another solution that has been proposed to the PQC problem seeks to use quantum effects themselves. Quantum key distribution (QKD) involves hardware that uses quantum entanglement to transmit encryption keys between parties at distance, replacing key exchange using public key cryptography. The parties then use the keys to communicate using symmetric

[87] *See* PQCRYPTO, "Multivariate-quadratic-equations post-quantum cryptography", https://pqcrypto.org/mq.html; "Multivariate cryptography", *Wikipedia*, https://en.wikipedia.org/wiki/Multivariate_cryptography.

[88] *See, e.g.,* "Method to produce new multivariate public key cryptosystems", US patent 7961876 (published 17 January 2008), https://www.google.com/patents/US20080013716.

[89] Encryption algorithms based on calculation of discrete logarithms on elliptic curves are vulnerable to Shor's algorithm, but supersingular elliptic curve isogeny is not.

[90] Luca de Feo, David Jao & Jérôme Plût, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies", PQCRYPTO (2011), http://eprint.iacr.org/2011/506.pdf; "Supersingular isogency key exchange", *Wikipedia*, http://eprint.iacr.org/2011/506.pdf.

cryptography. However, QKD has substantial limitations of range, security, and cost, and it does not address the PQC problem for public key digital signature (which is the primary use of public key cryptography for Smart Ledgers).[91] Accordingly, we do not consider QKD to be a serious contender for addressing the PQC problem for Smart Ledgers.

[91] *See* Kenny Patterson, "Post-Quantum Cryptography", presentation for Internet Engineering Task Force (2017), https://datatracker.ietf.org/meeting/99/materials/slides-99-saag-post-quantum-cryptography/.

4. Deciding the Timing for Action

There are two key strategic questions for any system potentially threatened with the PQC problem:

- When is it appropriate, or necessary, to take action?
- What action should be taken?
-

For both questions, the short answer is, "It depends". Specifically, the right answer depends heavily on the nature of the system, including the hardware, software and processes used, the sensitivity of data being stored, and the possible migration paths to a quantum-resistant system.

A. Decision Drivers – Uncertainty, Cost and Risk

The decision whether to make a computer system quantum-resistant is a challenging one, with three key decision drivers: ***uncertainty***, ***cost*** and ***risk.*** We consider each of these in turn.

As discussed above, there is substantial uncertainty as to when and if large-scale quantum computing will be available. In current times of rapid (some say exponential) technological change, this is a common dilemma. For example, the physicist Max Tegmark wrote in his recent book *Life 3.0*, about artificial intelligence:

> *There have been a number of surveys asking AI researchers how many years from now they think we'll have human level AGI [artificial general intelligence] with at least 50% probability, and all these surveys have the same conclusion: the world's leading experts disagree, so we simply don't know.*[92]

Disagreement about the likelihood and timing of large-scale quantum computing is equally severe. Although this uncertainty can provide a good reason to choose delay, it should not be a cause for decision paralysis. It remains possible to consider the likely range of timing for availability of large-scale quantum computing and then to use that range to consider courses of action.

[92] Max Tegmark, *Life 3.0*, p. 42 (Allen Lane: 2017).

The issue of cost has two sides. On the one hand, changing a typical system to be resistant to the PQC problem is likely to be an expensive (and perhaps very expensive) effort. Furthermore, there are risks in getting the solution wrong by being a first-mover, and implementing before there are settled and mature standards for PQC solutions.

On the other hand, the old expression that "a stitch in time saves nine" has force here. Failure to act could be associated with two types of costs: (1) the potentially devastating cost of having a vulnerable system if action is taken too late and (2) a possibly significant increase in remediation costs for implementing a solution close to the ultimate deadline (as occurred for many companies that waited to deal with the Y2K problem[93]). These competing concerns can be difficult to balance. ETSI, for example, has sought to do so by proposing a gradual, standards-based response to the PQC problem as the least costly approach.[94]

The costs of addressing the PQC problem are also a market opportunity — as the Y2K problem was a temporary opportunity for those who had programming skills in legacy languages like COBOL and FORTRAN.[95] The global encryption software market is estimated to have exceeded $3 billion in 2017, and is forecast to grow at well over 20% per year over the coming years.[96] Growth in this market is likely to be particularly fast if the PQC problem becomes more urgent, with associated opportunities in adjacent markets such as systems development and implementation.

The final decision driver has both objective and subjective components. Objectively, some types of computer systems present higher intrinsic risk — *e.g.* sensitive government systems, critical infrastructure systems, and high-

[93] Interview with Michele Mosca (8 January 2018).

[94] ETSI PQC Paper, section 6.4.

[95] *See* Larry Holyoke, "COBOL programmers log off jobs", *St. Louis Business Journal* (2 January 2000); "COBOL: Everywhere and Nowhere", Coding Horror blog (9 August 2009), https://blog.codinghorror.com/cobol-everywhere-and-nowhere/.

[96] *See* "Encryption Software Market worth 12.96 Billion USD by 2022", Markets and Markets, https://www.marketsandmarkets.com/PressReleases/encryption-software.asp (forecasting 27.4% growth from $3.87 billion in 2017 to $12.96 billion in 2022); "Encryption Software Market Size & Share Will Reach USD 7.17 Billion by 2021: Zion Market Research, https://globenewswire.com/news-release/2017/11/03/1174544/0/en/Encryption-Software-Market-Size-Share-Will-Reach-USD-7-17-Billion-by-2021-Zion-Market-Research.html (forecasting 21.7% growth from $2.20 billion in 2015 to $7.17 billion in 2021).

value financial systems. Subjectively, systems operators have different risk tolerances. For example, a company that has a high chance of business failure is likely to be substantially more risk-tolerant than an established blue-chip company.

Risk analysis for the PQC problem cannot happen in isolation. Every entity facing the PQC problem will face a variety of other risks, both related and unrelated. Related risks include other attacks on current cryptosystems, including the possibility of new mathematical attacks on the hard problems underlying public key cryptography (without requiring quantum computing) and the ongoing steady performance improvement of conventional computers (which requires gradual increase in cryptography security levels over times). There are also constantly emerging cybersecurity risks not related to cryptography, such as the major Meltdown and Spectre processor bugs revealed in January 2018.[97] And of course every organisation faces a variety of other commercial, compliance, political and other risks that compete for attention and expenditure.

One clear implication is that organisations implementing new systems now should strongly favour quantum-resistant solutions — in order to minimise uncertainty and risk, without the cost of changing or abandoning a legacy system. However, this is in fact *not* what is generally happening in the market for Smart Ledgers. For instance, most of the many cryptographic tokens being launched today use the ECDSA signature algorithm (which is vulnerable to Shor's algorithm), and anecdotally it appears that many other, less visible Smart Ledger projects are taking a similar approach. However, some are beginning to address this issue. For example, IOHK — which implements cryptocurrency Ada (Cardano) — announced "a long-term research agenda to gradually harden all algorithms used in Cardano's protocol stack against an adversary who possesses a quantum computer."[98] The extent to which the broader blockchain community will take a similar approach remains to be seen.

[97] *See* Devin Coldewey, "What Are Meltdown and Spectre, the bugs affecting nearly every computer and device?", *TechCrunch* (4 January 2018), https://techcrunch.com/2018/01/03/kernel-panic-what-are-meltdown-and-spectre-the-bugs-affecting-nearly-every-computer-and-device/.

[98] Charles Hoskinson, "Research program to work on hardening Cardano against quantum computers", *IOHK Blog* (1 February 2018), https://iohk.io/blog/research-program-to-work-on-hardening-cardano-against-quantum-computers/. IOHK is associated with the Cardano Foundation, the main sponsor of the Distributed Futures research programme of which this report is a part.

B. The Mosca Inequality – A Framework for Timing Decisions

Michele Mosca, a mathematics professor and cryptographer at Canada's University of Waterloo whose career has focused on quantum computing and who has has over the past decade devoted substantial attention to post-quantum cryptography, has proposed an excellent framework for assessing the timing for transition to PQC.[99] The so-called *Mosca inequality* involves three time periods:

- X – the desired duration of security of data on the system (X may effectively be zero where a system can be replaced with a quantum-resistant system in a way that avoids exposure of legacy data);
- Y – the time required to replace the cryptography in a system with quantum-resistant cryptography; and
- Z – the time until availability to quantum computing sufficient to break current encryption.

The X and Y periods are system-specific; and the Z period can be viewed as a probability distribution over the possible timings and outcomes in the development of large-scale quantum computing. Applying these figures to a given computer system or network:

> if $X + Y < Z$, then there is still time to implement quantum-resistant encryption; and
>
> if $X + Y > Z$, then it is too late to implement quantum-resistant encryption without risk to network security.

In the remainder of this section, we consider the values of X, Y and Z in more detail.

X — Desired Duration of Security

The appropriate and desired duration of security is highly system-, organisation- and data-specific. For example, for an e-commerce company that does not store credit card information, there may be relatively limited reason to protect commercial transaction information beyond a year or two.

[99] Professor Mosca is also CEO and Co-founder of evolutionQ (http://www.evolutionq.com/), a company consulting on the PQC problem.

There are certainly countervailing considerations, such as the value of customer lists and the fact that data protection law (*e.g.* the General Data Protection Regulation taking effect in May 2018) requires adequate security for personal data. But overall, such a company could reasonably decide that long-term security is not a sufficient strong interest to justify major current investments in quantum-resistant cryptography. Put differently, they could reasonably decide to be a follower if and when the e-commerce sector generally makes a transition to quantum-resistant cryptography, rather than being a leader.

Likewise, and perhaps counter-intuitively, X may be relatively short period for critical infrastructure systems. For example, the potential consequences of a real-time penetration of a power network are very high, but most utilities presumably care much less about hackers learning about how much power was produced yesterday, or last week.

Systems that require fairly lengthy security may include ones that store more sensitive personal data (*e.g.* health data) or high-value transaction data (*e.g.* bank systems or some Smart Ledgers). And the longest duration of security may be appropriate for government systems that are intended to maintain secrets for many decades.

Furthermore, for all systems, X is of limited relevance — and can be treated as effectively equal to zero[100] — where, after availability of large-scale quantum computing, either (1) a particular threat is relevant only to new transactions (or other new actions) or (2) the ultimate solution chosen for the PQC problem secures vulnerable data from access to relevant threat actors. As an example of the latter situation, consider a government database system that is accessed using vulnerable public key cryptography. Since such a database is usually not widely distributed, it may be possible to secure the legacy data with an appropriate degree of confidence as part of the transition to a quantum-resistant solution.

This approach of securing legacy data is closely related to the idea of 'defence in depth'. Securing legacy data may be difficult for Smart Ledgers due to the

[100] Michele Mosca agrees that such situations can exist (subject to certain conditions), and suggests treating them as changing his inequality to $Y < Z$ or $Y > Z$. Interview with Michele Mosca (8 January 2018).

distributed nature of the ledgers, but it could be possible to do so through changes to the associated software and protocols. For example, the Ethereum community took such an approach to deal with insecurity in the Distributed Autonomous Organization (DAO) protocol.[101]

Y — Time to Replace Cryptography

The time required to upgrade an existing system to avoid the PQC problem has two sub-components: the time required (a) to choose a solution and (b) to implement the chosen solution. The latter component is highly system- and organisation-specific, like nearly any IT project.

The time required to choose a solution depends in large part upon whether one waits for PQC solutions to mature further — for which by far the leading initiative is the NIST competition. Stanford's Professor Dan Boneh suggested doing exactly that at a panel at the 2017 RSA Conference on cryptography that addressed PQC and the question "Is time running out?" Boneh recommended: "Do nothing now, just wait for the NIST process."[102] This 'wait and see' approach also has the advantage that more information will be available over time about the prospects and limitations of quantum computers.

On the other hand, there is already good quantum-resistant cryptography available, as PQCRYPTO has recommended. And as we explain below, some organisations have already decided to begin the transition to quantum-resistant cryptography. For example, the US National Security Agency stated in August 2015 that it "will initiate a transition to quantum resistant algorithms in the not too distant future"[103]; and in 2016 Google announced a two-year experiment with some implementations of the Chrome browser and Google infrastructure that it accesses, in which vulnerable elliptic curve cryptography

[101] *See* "The DAO, The Hack, The Soft Fork and The Hard Fork", *CryptoCompare*, https://www.cryptocompare.com/coins/guides/the-dao-the-hack-the-soft-fork-and-the-hard-fork/.

[102] Matthew J. Schwartz, "Post-Quantum Crypto: Don't Do Anything", *BankInfo Security* (22 February 2017), https://www.bankinfosecurity.com/quantum-crypto-dont-do-anything-a-9737.

[103] *See* "NSA acknowledges need for quantum-safe crypto", *IDQ* (14 August 2015), https://www.idquantique.com/nsa-quantum-safe-crypto/; NSA Information Assurance Directorate, Data at Rest Capability Package, p. 4 (March 2016).

will be replaced with Ring-LWE (a lattice-based, quantum-resistant algorithm).[104]

Michele Mosca himself charts an intermediate course, stating that it can be a mistake to 'bake in' a PQC solution today given the extensive ongoing research on quantum-resistant algorithms. He advocates a quantum risk assessment working backward from the security goal to the mitigation plan for the PQC problem — designing, testing and implementing components of the plan as appropriate in parallel with the NIST standardisation process.[105]

Z — Timeline for Quantum Computing

The timeline for quantum computing is the great uncertainty associated with the PQC problem. To attempt to penetrate this uncertainty, we begin by specifying that we are interested in knowing the minimum time until quantum computers (a) have adequate capacity to efficiently attack current public key cryptography (*i.e.* about 3,000 to 5,000 logical qubits for 2,048-bit RSA) and (b) are available at a cost that attackers are willing to bear. However, these two constraints are probably not especially limiting, since the historical progress of various technologies (*e.g.* computers, storage devices, mobile phones) indicates that once reliable quantum computers with a significant number of logical qubits are available at any cost, further progress in increasing the number of qubits and reducing cost will probably be fairly rapid. The hard question is whether and when quantum computers with a significant number of entangled logical qubits (say over a few hundred) will be available at all.

Here are a few data points regarding these questions:

- April 2016 – NIST stated:
  While in the past it was less clear that large quantum computers are a physical possibility, many scientists now believe it to be merely a significant engineering challenge. Some experts even predict that within the next 20 or so years, sufficiently large quantum computers

[104] *See* Andy Greenberg, "Google Tests New Crypto in Chrome to Fend Off Quantum Attacks", *Wired* (2 July 2016), https://www.wired.com/2016/07/google-tests-new-crypto-chrome-fend-off-quantum-attacks/.

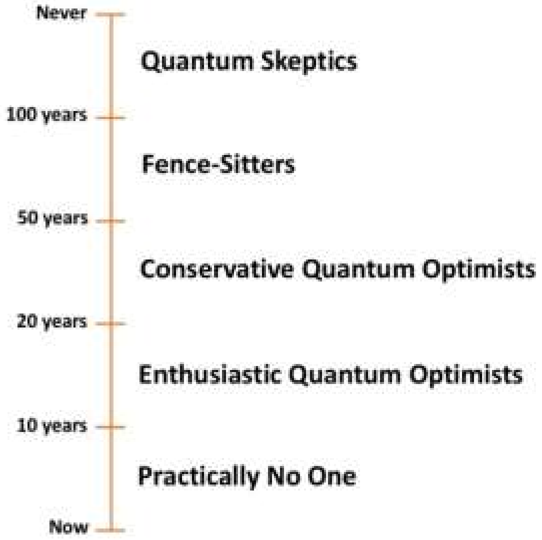
[105] Interview with Michele Mosca (8 January 2018). Mosca indicates that his company evolutionQ plans to release a paper on quantum-resistant blockchains within the next several months.

will be built to break essentially all public key schemes currently in use.[106]

- July 2016 – A Google executive responsibility for testing quantum-resistant cryptography stated: "We sometimes joke that practical quantum computers are always 20 years in the future, and have been for a very long time … ."[107]
- February 2017 – On the same RSA Conference panel mentioned in above, renowned cryptographer Adi Shamir (one of the authors of the RSA algorithm) stated: "I wouldn't lose too much sleep over quantum computers." Other scientists believe that there are insurmountable physical barriers to large-scale quantum computing.
- May 2017 – Michele Mosca suggested that there is a 1 in 6 chance that there will be large-scale quantum computing within 10 years, and that this is 'likely' within 10-15 years.[108]

In short, a consensus view of timing simply does not exist. One might depict the spectrum of expert viewpoints as in Figure 6.

Figure 6. Views on Timing of Large-Scale Quantum Computing



In the face of lack of expert consensus, some have suggested that governments should provide guidance on the likely timing of availability of large-scale

[106] NIST PQC Report, p. 2.
[107] *See* Andy Greenberg (note 1044 above).
[108] Mosca 2017 Presentation.

quantum computing — including because governments have historically had privileged access to a variety of stakeholders.[109] It is difficult to expect governments to develop a clear view where scientific and industry experts cannot (and any government with knowledge of substantial progress in quantum computing might choose to keep that information secret for its own strategic advantage). However, there may be opportunities for targeted guidance: for example, data protection authorities might at some future point opine on circumstances in which non-quantum-safe cryptography is inadequate for data protection purposes.

At least for the time being, it appears that companies and other organisation facing potential threats from the PQC problem will need to reach their own decision about what view to take on timing, taking into account their risk tolerance and other factors. In sum, perceived system and organisational risk is a crucial factor in the decision of when to take action on the PQC problem, because increased risk perception tends both to increase Y and to reduce perceptions of Z — both of which encourage earlier action. The incentive for early action may be particularly strong for organisations that are vulnerable to failures of public and/or investor confidence, because such organisations will likely face damage from increased likelihood of the PQC problem (*e.g.* through proof of concept of viable large-scale quantum computers) well before their systems actually become vulnerable.

C. Applying the Mosca Inequality to Smart Ledgers

Despite the inherent uncertainties in the values of X, Y and Z, the Mosca inequality provides a very useful tool for assessing the timeline for action to address the PQC problem. We illustrate this in Table 7 using the two primary categories of threats to Smart Ledgers from quantum computing:

- unauthorised running of software on historical blockchains; and
- spoofing of new transactions, data or software.

[109] Interview with Prof. Fred Piper, University of London, Royal Holloway (4 December 2017).

Table 7. Application of Mosca Inequality to Quantum Threats to Smart Ledgers

Threat	Timeline (years)			X + Y > Z	Conclusion
	X	Y	Z		
unauthorised running of historical software	indefinite (as long as software has value)	5	15 to 20	yes, possibly	Smart Ledgers with software that may be valuable for a long period should consider quantum-resistant solutions now.
spoofing new transactions, data or software	0 (except where existing assets cannot be moved)	5	15 to 20	no	There is plenty of time to watch PQC developments and decide how to act (with certain exceptions).

We have made two primary assumptions in Table 6:

- Y = 5 years on the basis that 5 years should be sufficient to redesign and implement a blockchain protocol under serious pressure from the PQC problem; and
- Z = 15 to 20 years on the basis that it's prudent to fall decidedly on the optimistic side of the spectrum about the timeline for quantum computing, unless costs of change are relatively high and risks are relatively low (neither of which is generally the case for Smart Ledgers).

It turns out on these fairly conservative assumptions that most of the game is in the value of X. And for these specific threats, the values of X are quite clear:

- For unauthorised running of software, anything that is put on a blockchain now is there indefinitely, and is potentially at risk if it still has validity/value when the PQC problem bites. For example, consider a Smart Ledger application designed to execute a last will and testament (or part of it), as developing 'smart contracts' technology could permit.[110] Such software would need to be secure for 50 years or more — and the period would be even longer if the application involved a multi-generational trust.

[110] *See, e.g.,* Ameer Rosic, "Smart Contracts: The Blockchain Technology That Will Replace Lawyers", *Blockgeeks*, https://blockgeeks.com/guides/smart-contracts/; Internet of Agreements project, http://internetofagreements.com/.

- For spoofing of new transactions, data or software, the main requirement is that a solution be in place before large-scale quantum computing is actually available — *i.e.* this is one of the situations in which X can be treated as zero for most purposes. However, this is subject to the important caveat that this reasoning does not apply where existing resources cannot be moved in a way that protects them from new transactions (as may be the case for Satoshi Nakamoto's bitcoins if he or she is dead).

In summary, and perhaps counterintuitively, the urgency for action is much greater for the narrower, long-term threat to certain current transactions than it is for the ultimate potential of the PQC problem to render current Smart Ledger architectures entirely insecure.

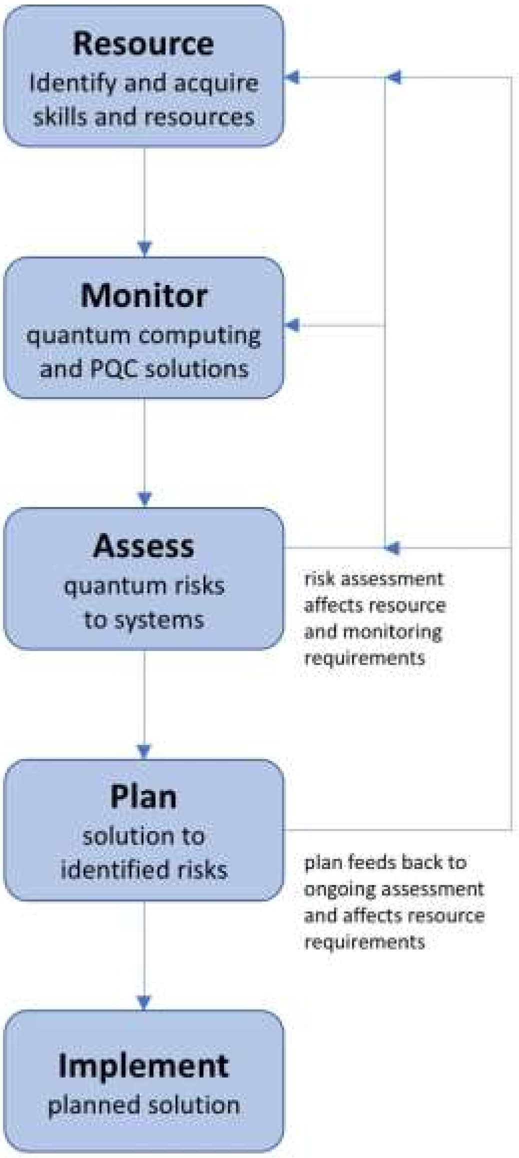
5. Recommendations

Grounded in the above tour of the PQC problem, we offer recommendations in two areas. First, we propose a framework for decision-making on the PQC problem by potentially affected organisation. Second, we suggest three areas for further related activity.

A. Organisational Decision-Making

Returning to our introductory metaphors of Chicken Little and the “Don’t panic” message of the *Hitchhiker’s Guide to the Galaxy*, our decided preference is for the latter. We suggest that organisations following the framework below (illustrated in Figure 7, with further description below) will avoid panic — the sky is not falling. There are likely to be robust responses to the PQC problem even for organisations with high-risk, long-term digital assets that are potentially vulnerable to quantum computing.

Figure 7. Framework for Addressing the PQC Problem


Resource
Identify and acquire skills and resources
Monitor
quantum computing and PQC solutions
Assess
quantum risks to systems
risk assessment affects resource and monitoring requirements
Plan
solution to identified risks
plan feeds back to ongoing assessment and affects resource requirements
Implement
planned solution

Resource. The PQC problem is complex, and it is important for any organisation confronting to have skills and resources sufficient to understand it fully, including to cut through market disinformation from vendors promoting particular solutions. Given the specialist nature of the skills required and the rapidly-emerging problem, there is likely to be significant competition for people with the right skills, raising their cost. For many organisations, outsourcing to specialists may be a good solution, particularly at early stages of engagement with the PQC problem. Resources requirements are likely to change as responses to the PQC problem are assessed and planned (as the flowchart illustrates).

Monitor. With the great uncertainty associated with quantum computing, it is crucial to understand the progress of quantum computing, the evolving likelihood and timeline of the PQC problem, and the evolving solutions (via the NIST competition and otherwise).

Assess. Along with the next step of planning, assessment is at the heart of the framework set out in this section. It involves determining and evaluating quantum risks to the organisation's systems, services and overall business, applying the general approaches set out in this report and paying particular implementation to specific technical issues (including interactions with technology vendors on available solutions). Evaluating the response timing for identified risks is a crucial component of the assessment. More generally, defining the scope of the assessment at an early stage is important. A further framework that provides useful additional detail on the assessment step (and to a lesser extent the other steps) of our framework has been proposed by Michele Mosca and John Mulholland, who are both associated with evolutionQ (a consulting firm focusing on the PQC problem).[111]

Plan. Planning a specific response to identified quantum risks is where the rubber hits the road. This may involve immediate replacement of vulnerable cryptography with quantum-resistant cryptography, or it could involve intermediate steps of building resilience by using a 'defence in depth' approach (such as introducing additional quantum-resistant protections to an existing system). This latter approach could also involve a standards-based evolution of existing systems (as ETSI has proposed) or a gradual system redesign that allows cryptography algorithms to be swapped at a later stage as the nature of the PQC problem and its solutions become clearer.

Implement. This should ideally be a straightforward implementation of the plan from the previous step. But many IT projects go wrong, especially where specifications are not clear, or change. Careful attention should be given to these issues, and resource requirements may change significantly at this stage.

[111] Michele Mosca & John Mulholland, "A Methodology for Quantum Risk Assessment", Global Risk Institute (5 January 2017), https://globalriskinstitute.org/publications/3423-2/.

B. Areas for Further Research

We would suggest at least three related areas for further research.

Modelling of Response Timing for the PQC Problem

Significant further modelling of the timeline for action on the PQC problem could be done. The modelling could build on the Mosca inequality, provide more 'learning curve' insights, and explore technology-dependencies. A particular powerful tool might be to define the problem as a Bayesian network (a flow of actions and decisions based upon inputs that are defined as probability distributions), and to assess the likelihood of particular outcomes on that Bayesian network using Monte Carlo simulations (simulations using multiple iterations with random variation of assumptions and/or data) or other tools.

Policy-Making and Advocacy

There has been useful analysis in recent years about the potential for government policy to push cybersecurity decisions in the right direction, including by establishing cybersecurity standards and liability rules.[112] Most major governments have a cybersecurity strategy,[113] and responsible advocacy will be an important part of ensuring that these strategies include an appropriate approach to the evolving PQC problem. A significant amount of work could be done on standards, benchmarking, use of Smart Ledgers in

[112] *See, e.g.,* Bruce Schneier, *Beyond Fear: Thinking Sensibly About Security in an Uncertain World* (Springer: 2006). On the general idea of using government policy to "nudge" individuals and industry in the right direction, *see* Richard H. Thaler & Cass R. Sunstein, *Nudge: Improving Decisions About Health, Wealth and Happiness* (Penguin: 2009). The ideas behind this book led in 2010 to the formation of a Behavioural Insights Team (known as the "Nudge Unit") at the UK Cabinet Office. *See* Felicity Lawrence, "First goal of David Cameron's 'nudge unit' is to encourage healthy living", *The Guardian* (12 November 2010).

[113] *See, e.g.,* UK National Cyber Security Strategy 2016 to 2021 (1 November 2016), https://www.gov.uk/government/publications/national-cyber-security-strategy-2016-to-2021; US Presidential Executive Order on Strengthening the Cybersecurity of Federal Networks and Critical Infrastructure (11 May 2017), https://www.whitehouse.gov/presidential-actions/presidential-executive-order-strengthening-cybersecurity-federal-networks-critical-infrastructure/; KPMG, "Overview of China's Cybersecurity Law" (February 2017), https://assets.kpmg.com/content/dam/kpmg/cn/pdf/en/2017/02/overview-of-cybersecurity-law.pdf.

critical national infrastructure, appropriate government sharing of information on encryption, and other government-business relations.

Insurance

Insurance is a rich topic for PQC and Smart Ledgers, with at least three large areas for consideration:

- long-tail data risk — Ledgers built today that are not quantum-resistant could pose long-term liability risks for directors' errors & omissions, data privacy, and cyber risk. These risks should be considered and priced today.
- long-term opportunities — There are four generic approaches to managing risk: accept, avoid, transfer, and reduce. PQC risks appear unacceptable, and there are few viable market mechanisms for transferring PQC risks. So for those PQC risks that are unavoidable, there is only one option: reduce PQC risks, for example through co-operative pooling approaches to risk reduction (*e.g.* based on mutual insurance models, such as Protection & Indemnity mutuals used in shipping).
- government-sponsored cyber-catastrophe reinsurance — A PQC event could constitute a cyber-catastrophe event that could seriously affect the economy. To reduce such impacts, governments could consider promoting public-private cyber-catastrophe reinsurance schemes, with the government as an insurer of last resort if the scheme's reserves have been exhausted.

Principal Author

This report was written by Maury Shenk. Maury's experience focuses at the intersection of technology, law and business. He is founder and managing director of Lily Innovation, through which he handles a portfolio of activities including private equity and corporate finance, legal advisory, directorships, start-up investing and teaching/writing. Maury is a dual-qualified US/UK lawyer and former managing partner of the London office of global law firm Steptoe & Johnson, where he remains an advisor; general counsel of China-based private equity fund Spring Capital Asia; and director of testing and certification company PeopleCert and recycling compliance company Valpak. He has a deep practical understanding of technology, especially IT and telecommunications, artificial intelligence, information security and green technology. Maury is a graduate of Harvard College and Stanford Law School. He is a lover of languages – a native speaker of English (the American version), proficient in French and Russian, comfortable in Mandarin Chinese and Spanish, and dilettante in German, Italian and Norwegian. He is also an avid competitive and recreational sailor.

Acknowledgements

We thank all those who have contributed time and expertise to this research. Any misinterpretations or errors are, of course, our responsibility. The following people provided extremely valuable insights and assistance:

- Professor Ross Anderson, Computer Laboratory, University of Cambridge (email exchange)
- Dave Askey, Chief Technology Officer, Post-Quantum
- David Birch, Global Ambassador, Consult Hyperion
- Gary Glennon-Alty, Blockchain Lead, UK Defence Science and Technology Laboratory
- Professor Michael Mainelli, Executive Chairman, Z/Yen Group
- Bob McDowall, Special Advisor, Cardano Foundation
- Dustin Moody, Post-Quantum Cryptography Lead, NIST: US National Institute of Standards and Technology
- Professor Michele Mosca, Co-Founder, Institute of Quantum Computing, University of Waterloo (Canada) and CEO and Co-Founder, evolutionQ
- Professor Fred Piper, Royal Holloway, University of London
- Mike St John-Green, independent cyber security consultant

We would also like to thank the production team:

- Alexandra Karathodorou
- James Pitcher

Distributed Futures is a significant part of the Long Finance research programme managed by Z/Yen Group. The programme includes a wide variety of activities ranging from developing new technologies, proofs-of-concept demonstrators and pilots, through research papers and commissioned reports, events, seminars, lectures and online fora.

Distributed Futures topics include the social, technical, economic, and political implications of smart ledgers, such as identity, trade, artificial intelligence, cryptography, digital money, provenance, FinTech, RegTech, and the internet-of-things.
www.distributedfutures.net

Cardano Foundation is a blockchain and cryptocurrency organisation based in Zug, Switzerland. The Foundation is dedicated to act as an objective, supervisory and educational body for the Cardano Protocol and its associated ecosystem and serve the Cardano community by creating an environment where advocates can aggregate and collaborate.

The Foundation aims to influence and progress the emerging commercial and legislative landscape for blockchain technology and cryptocurrencies. Its strategy is to pro-actively approach government and regulatory bodies and to form strategic partnerships with businesses, enterprises and other open-source projects. The Foundation's mission is the promotion of developments of new technologies and applications, especially in the field of new open and decentralised software architectures.
www.cardanofoundation.org

"When would we know our financial system is working?" is the question underlying Long Finance's goal to improve society's understanding and use of finance over the long term. Long Finance aims to:

- expand frontiers - developing methodologies to solve financial system problems;
- change systems - provide evidence-based examples of how financing methods work and don't work;
- deliver services - including conferences and training using collaborative tools;
- build communities - through meetings, networking and events.

www.longfinance.net

Z/Yen is the City of London's leading commercial think-tank, founded to promote societal advance through better finance and technology. Z/Yen 'asks, solves, and acts' on strategy, finance, systems, marketing and intelligence projects in a wide variety of fields. Z/Yen manages the Long Finance initiative.

Z/Yen Group Limited
41 Lothbury, London EC2R 7HG, United Kingdom
+44 (20) 7562-9562 (telephone)
hub@zyen.com (email)
www.zyen.com