Reading for Week 13:

Quantum computing: pages 139-167
Quantum communications: pages 189-193; 200-205; 217-223

Law and Policy for the Quantum Age

Draft: Do not circulate without permission

Chris Jay Hoofnagle and Simson Garfinkel

April 2, 2021

Note to reviewers

Thank you for taking the time to look through our draft!

We are most interested finding and correcting text that is *factually wrong*, *technically inaccurate*, or *misleading*.

Please do not concern yourself with typos or grammar errors. We will be focusing on those later when we have all of the text in its near-final form. (However, we are interest in word usage errors, or in misspelling that are the result of homophones, as they are much harder to detect and correct in editing.)

You can find the current draft of this book (as well as previous PDFs) here: https://simson.net/quantum-2021-slgcjh/

Thank you again! Please email us at:

choofnagle@berkeley.edu and simsong@acm.org

5

(NEAR FINAL) Quantum Computing Applications

> "A good way of pumping funding into the building of an actual quantum computer would be to find an efficient quantum factoring algorithm!"[1]

The risk of wide-scale cryptanalysis pervades narratives about quantum computing. We argue in this chapter that Feynman's vision for quantum computing will ultimately prevail, despite the discovery of Peter Shor's factoring algorithm that generated excitement about a use of quantum computers that people could understand—and dread. Feynman's vision of quantum devices that simulate complex quantum interactions is more exciting and strategically relevant, yet also more difficult to portray popular descriptions of technology. The Feynman vision for quantum computing will lead to applications that benefit humans in multifarious and unforeseen ways, just like the classical computing revolution improved our lives. Feynman's vision may also enable a "winner-take-all" outcome in building a large quantum computer.

To explain this outcome, we canvass the three primary applications that have been developed for quantum computing: Feynman's vision of simulating quantum mechanical systems, factoring, and search. The next chapter discusses today's quantum computing landscape.

For Feynman, a quantum computer was the only way that he could imagine to efficiently simulate the physics of quantum mechanical systems. Such systems are called *quantum simulators*.[2] Quantum simulation remains the likely first practical use of quantum computers. Oddly, this application is *not* responsible for most of the public interest in quantum computers, which has instead been fueled by the desire to make super-machines that can crack the world's strongest encryption algorithms. Since then, without dramatic demonstrations of other capabilities, and with the underlying complexity of achievements that have been made, many news articles cast quantum computing in a single, privacy-ending narrative.

We believe that prominence of cryptanalysis in public interest and government funding over the past two decades is because a working quantum computer that could run Shor's algorithm on today's code would give governments that owned it an incredible advantage to use over their adversaries: the ability to crack messages

[1] Berthiaume and Brassard, "Oracle Quantum Computing" (1994), written hours before Peter Shor discovered such an algorithm.

[2] The term *quantum simulators* is confusing, because it is also applied to programs running on conventional computers that simulate quantum physics. For this reason, some authors use the terms *Feynman simulators* or even *Schrödinger-Feynman simulators*.

that had been collected and archives going back decades. But while this advantage may be responsible for early funding of quantum computing, we believe that the cryptanalytic capabilities of initial quantum computers will be limited and outshone by the ability of these machines to realize Feynman's vision. And Fenyman's vision, unlike cryptanalysis, confers first-mover advantage, since a working quantum physics simulator can be used to build better quantum physics simulators. That is, quantum physics simulations are likely to create a virtuous circle, allowing the rate of technology change to increase over time.

The last section of this chapter turns to search, and explains the kinds of speedups quantum computers are likely to provide. Understanding those likely speedups further advances our prediction that the future of quantum computing will be Feynman's.

5.1 Simulating Physical Chemistry with Quantum Computers

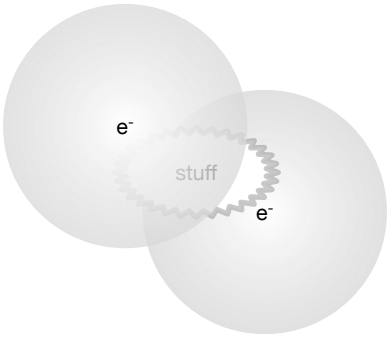
In this section we explore how one might actually go about simulating physics with quantum computers. Despite the similarity of titles, this section is not an extended discourse on Feynman's articles. Instead, it is a discussion of how chemists actually simulate the physics of chemical reactions with classical computers today, and how they might do so with quantum computers tomorrow.

Classical computers—like the computers used to write and typeset the book—are designed to execute predetermined sequences of instructions without error and as reliably as possible. Computer engineers have made these machines steadily faster over the past 80 years, which makes it possible to edit this book with graphical editors and typeset its hundreds of pages in less than a minute. Both of those activities are fundamentally a sequence of operations applied to a sequence of bits, starting with an input stream of `0`s and `1`s, and possibly a character typed on a computer keyboard) and deterministically creating a single output stream (the PDF file that is displayed on the computer's screen).

Modeling molecular interactions is fundamentally different from word processing and typesetting. When your computer is running a word processing program and you press the H key, there is typically only one thing that is supposed to happen: an "H" appears on the at the cursor on the screen. But many different things can happen when two molecules interact: they might stick together, they might bounce, or an atom might transfer from one molecule to the other. The probability of each of these outcomes is determined by quantum physics.

To explore how two molecules interact, the basic approach is to build a model of all the atomic nuclei and the electrons in the two-molecule system and then compute how the wave function for the system evolves over time. Such simulations quickly become unworkable, so scientists will consider a subset of the atoms and electrons, with the hope that others will stay more-or-less static. Other approximations exist, such as assuming that the nuclei are fixed in space and are point charges, rather than wave functions themselves. High school chemistry, which typically presents the electrons as little balls of charge spinning around the nuclei, is a further simplification.

We present such a simplified system in Figure 5.1. To keep things simple, we have assumed might assume that there are only two electrons of interest, and that each will end up in either a *low* or *high* energy state. Facing this system, a scientist



Energy State		outcome
electron 1	electron 2	probability
low	low	2%
low	high	5%
high	low	90%
high	high	3%

Figure 5.1: The possible energy states of two electrons in a hypothetical quantum system.

can use modeling software to determine the probably of each of outcomes. Here our hypothetical scientist has used a conventional computer to would run this experiment many times, tabulate the results, and report them in the rightmost column as an *outcome probability*.

Our scientist would take fundamentally different approach to solve this problem on a quantum computer. Instead of modeling the probabilities, the scientist designs a *quantum circuit* that directly represents (or simulates) the chemistry in question. With most quantum computers today, the scientist would then turn on the quantum computer, placing each of its quantum bits (called *qubits*) into a superposition state. The quantum circuit plays through the quantum computer, changing how the qubits interact with each other over time. This "playing" of the quantum circuit is performed by a second computer—a classical computer—that controls the quantum computer. When the circuit is finished playing, the second computer measures each qubit, collapsing the superposition wave function and revealing its quantum state. At this point each qubit is *either* a `0` or a `1`.

In this example, each qubit might directly represent an energy state of an electron that was previously modeled. So if our scientist designed a quantum circuit and ran it on our hypothetical quantum computer, the result might look like this:

Trial	qubit 1	qubit 2
#1	`1`	`0`

It looks like the quantum computer has found the right answer instantly!

Actually, no. Because if the scientist ran the experiment a second time, the answer might be different:

Trial	qubit 1	qubit 2
#2	`1`	`1`

In an actual quantum computer, the experiment would run multiple times:

Trial	qubit 1	qubit 2
#3	1	0
#4	1	0
#5	0	0
#6	1	0
#7	1	0
#8	1	0
#9	1	0
#10	1	0

After these trials, the results are tabulated to get a distribution of possible answers. The statistics that are similar to those produced by the classical computer, but a little different:

qubit 1	qubit 2	Trial #s	Count	Probability
0	0	#5	1	10%
0	1	–	0	0%
1	0	#1, #3, #4, #6, #7, #8, #9, #10	8	80%
1	1	#2	1	10%

Notice that the quantum computer does not generally produce the same results as the classical computer. This may be because we did not run sufficiently many trials to get results with the same statistical distribution as the results produced by the classical computer. It might also be because the model run on the classical computer is incomplete. More likely, both models are incomplete, but incomplete in different ways. (Even if they were identical models, it's unlikely that identical statistics would emerge with just ten runs.)

It is important to remember that in this simulation, as in real quantum systems, *there is no right answer*. Instead, there is a range of possible answers, with some more probable and some less probable. This is one of the reasons that there can be so many different combustion products when even relatively simple compounds burn in the open air.

In practice, efficient quantum computing algorithms are designed so that "correct" or desired answers tend to generate constructive interference on the quantum computing circuits, while answers that are not desired tend to cancel each other out with destructive interference. This is possible because what quantum computers actually do is to evolve carefully constructed probability waves in space and time. These waves "collapse" when the final measurement is made by the scientist (or, more specifically, by the classical computer that is controlling the quantum computer). For a discussion of quantum mechanics and probability, please see Chapter B.

The advantage of a quantum computer becomes clear as the scale increases. Exploring the interaction of 32 electrons, each of which could be in two states, requires exploring a maximum of 4 Gi[3]p combinations. A classical computer would need to explore *all* of those combinations one-by-one. Exponential growth is really something: simply printing out those 4 Gi combinations at 6 lines per inch would consume

[3]4 Gi means 4 Gigi, which is the SI prefix that denotes powers-of-two rather than powers-of-ten counting. 4 Gi is $4 \times 1024 \times 1024 \times 1024 = 2^{32} = 4,294,967,296$, or roughly 4.2 billion.

11,297 linear miles of paper. Today for certain problems, quantum computing scientists have discovered algorithms that run more efficiently on quantum computers than the equivalent classical algorithms that exist to solve the problems on conventional computers. Generally speaking, the more qubits a quantum computer has, the more complex a system it can simulate.

Approaches for programming quantum computers are still in their infancy. Because the machines are small—with dozens of qubits, rather than millions—programmers need to concern themselves with individual qubits and gates. In some notable cases quantum computers are being constructed to solve specific problems.[4] This is reminiscent of the way that the first computers were built and programmed in the 1940s, before the invention of stored programs and computer languages: in England the Colossus computers were built to crack the German's Lorentz code, while in the U.S. the ENIAC was created to print artillery tables. Programming quantum computers will get easier as scientists shift from single-purpose to general machines and as the machines themselves get larger.

In addition to the number of qubits, the second number that determines the usefulness of a modern quantum computer is the stability of its qubits. Stability is determined by many things, including the technology on which the qubits are based, the purity of the materials from which the qubits are manufactured, the degree of isolation between the qubits and the rest of the universe, and possibly other factors. Qubits that are exceedingly stable could be used to compute complex, lengthy quantum programs. Such qubits do not currently exist. In fact, an entire research field explores ways to shorten quantum algorithms so that they are compatible with short-lived qubits.

Quantum engineers use the word noise to describe the thing that makes qubits less stable. *Noise* is a technical term that engineers use to describe random signals. The reason we use this term is that random signals fed into a speaker literally sound like a burst of noise, like the crackle between stations on an AM radio, or the sound of crashing waves. Noise in the circuit does not help the quantum computer achieve the proper distributions of randomness and uncertainty described by quantum mechanics. Instead, noise collapses the wave functions and scrambles the quantum computations, similar to the way that jamming the relay contacts in the Harvard's Mark II computer caused it to compute the wrong numbers on September 9, 1947.[5] Early computers only became useful after computer engineers learned how to design circuits that reduced noise to the point of irrelevance. They did this using an engineering technique called *digital discipline* that is still used today (see page 63), but that approach won't work with quantum computers.

Instead, companies like Rigetti, IBM and Google have created machines that have noisy qubits. As a result, most quantum programs today are small and designed to run quickly. Looking towards the future, many noisy qubits can be combined to simulate cleaner qubits using an error-correcting technique called surface codes,[6] but today's machines do not have enough sufficient noisy qubits for this to be practical. Another approach is to use a quantum computing media that is largely immune to

[4] Zhong et al., "Quantum computational advantage using photons" (2020).

[5] A moth was found pinned between the contacts of Relay #70 Panel F. Grace Hopper, a developer and builder of the Mark II, taped the insect into her laboratory notebook with the notation "first actual case of bug being found."

[6] Fowler et al., "Surface codes: Towards practical large-scale quantum computation" (2012).

Quantum error correction

The quantum computing applications that we discuss in this chapter all assume the existence of a working, reliable quantum computer with sufficient qubits, able to run quantum circuits with sufficient size and complexity for a sufficiently long period of time.

Although an absolutely reliable quantum computer is a useful theoretical construct for thinking about quantum computing algorithms. Actual quantum computers will probably need to use some form of *quantum computers!quantum error correction*, in which multiple noisy qubits are used to simulate a smaller number of qubits that have less noise.

Although quantum error correction is powerful, today's techniques do not appear to be up to the task of sustaining a single quantum computation for time periods that would be sufficiently long enough to pose a threat to modern cryptographic systems.

noise; that's the approach being taken by Microsoft with its so-called *topological qubits*, although other approaches using photonic qubits or ion traps might produce similar noise-free results. But for today, noise significantly limits the complexity of computations that can be done on quantum computers, even if we could build machines with hundreds or thousands of noisy qubits.

Even so, some companies are eager to get a head start, and are having their scientists and engineers learn to program these machines today. As a result, IBM is able to generate revenue with its "quantum experience" by giving free access over the Internet to machines with only a few qubits, and renting time to institutions who want access to IBM's larger machines. Likewise, Amazon Web Services has started making small quantum computers built by other companies available through its "Bracket" cloud service. However, the power of these machines is dwarfed by Amazon's conventional computing infrastructure.

Finally, there is an important point that we need to make: there is no mathematical *proof* that a quantum computer will be able to simulate physics faster than a classical computer. The lack of such a proof reflects humanity's fundamental ignorance on one of the great mathematical problems of time, *NP completeness* (see Section 3.5.4, "NP-Complete" (p. 82)). What we do know is that *today's* quantum simulation algorithms get exponentially slower as the size of the problem being simulated increases in size, and the simulation algorithms that we have designed for quantum computers do not. But this may reflect the limits of our knowledge, rather than the limits of classical computers. It might be that work on quantum computing leads to a breakthrough in mathematics that allows us to create dramatically faster algorithms to run on today's classical computers. Or it may be that work on quantum computing allows us to prove that quantum computers really fundamentally more powerful than classical computers, which would help us to solve the great mathematical question of NP completeness. What we know today is that quantum computers can take advantage of quantum physics to run so-called *BQP algorithms*, and that today's BQP algorithms run more efficiently than the fastest algorithms that we know of to run on classical computers. (See Section 3.5.4 (p. 82) and Section 3.5.6 (p. 86) for a more in-depth discussion of these topics.)

5.1.1 Nitrogen Fixation, Without Simulation

To put efforts to develop a quantum computer into context, this section explores how such a machine might help developing more efficient approaches for "fixing" nitrogen.

Nitrogen, in the form of organic nitrates, is both vital for biological life and in surprisingly short supply. The productivity of pre-industrial agriculture was often limited by the lack of nitrogen, rather than limitations of water or sunlight. Industrial agriculture has solved this problem through the industrial production of nitrogen-based fertilizers.

What makes the need for added nitrogen so surprising is the fact that plants are surrounded by nitrogen in the form of air. Nearly 80% of dry air is nitrogen. The problem is that nitrogen in the air is N_2, also written $N{\equiv}N$, with a triple chemical bond between the two nitrogen atoms. This triple bond has the charge of six electrons, making it difficult to break. As a result, the nitrogen in air is inaccessible to most plants.

Nitrogen fixation is the process of taking N_2 and turning it into a more usable form, typically ammonia (NH_3). The overall chemical reaction is not very complex:

$$\mathit{Energy} + N_2 + 3\,H_2 \longrightarrow 2\,NH_3 \tag{1}$$

Most of the natural nitrogen fixation on Earth happens in the roots of alfalfa and other legumes, where nitrogen-fixing bacteria live in a symbiotic relationship with the plant host.[7] Instead of hydrogen gas, biological nitrogen fixation uses ATP (adenosine triphosphate) produced by photosynthesis, some spare electrons, and some hydrogen ions (present in acid) that just happen to be floating around. The products are ammonia (containing the fixed nitrogen), hydrogen gas, ADP (adenosine diphosphate), and inorganic potassium (written as Pi below):

$$N_2 + 16\,\mathrm{ATP} + 8\,e^- + 8\,H^+ \longrightarrow 2\,NH_3 + H_2 + 16\,\mathrm{ADP} + 16\,\mathrm{Pi} \tag{2}$$

The plant then uses photosynthesis and sunlight to turn the ADP back into ATP.

In 1909, the German chemist Fritz Haber discovered an inorganic approach to nitrogen fixation using high pressure and the chemical element osmium, which somehow helps the electrons to rearrange. Chemists say that osmium *catalyzes* the reaction. Haber was awarded the Nobel Prize in Chemistry in 1918, "for the synthesis of ammonia from its elements."[8]

Haber sold his discovery to the German chemical firm BASF, which assigned Carl Bosch the job of making the process commercially viable. Osmium has 76

[7] There is also a small amount of nitrogen fixation that results from lightning.

[8] Haber is also known as the "father of chemical warfare" for his work weaponizing the production and delivery of chlorine gas as part of Germany's efforts during World War I, and for his institute's development of Zyklon A. Despite this service to the country and the fact that he had converted from Judaism to Christianity, Haber was considered a Jew by the Nazi regime, and fled to England after the Nazis rose to power. "[S]cientists there shunned him for his work with chemical weapons. He traveled Europe, fruitlessly searching for a place to call home, then suffered heart failure in a hotel in Switzerland in 1934. He passed away shortly thereafter at the age of 65, but not before repenting for devoting his mind and his talents to wage war with poison gasses." (King, "Fritz Haber's Experiments in Life and Death" [2012]) Zyklon A ultimately led to the development and use of Zyklon B in the Nazi extermination camps.

electrons that are exquisitely arranged, which presumably is the reason for its catalytic prowess, but it is also one of the rarest chemicals on the planet, so Bosch and his colleague looked for a cheaper catalyst. They discovered that uranium also worked, but settled on catalyst made by treating iron with potassium. (Iron is in the same column of the periodic table as Osmium because they have same arrangement of "outer" electrons, with the result that they have some similar chemical properties.) Today modern industrial catalysts for nitrogen fixation include mixtures of aluminum oxide (Al_2O_3), potassium oxide (K_2O), zirconium dioxide (ZrO_2), and silicon oxide (SiO_2). For this work, Carl Bosch received the 1931 Nobel Prize in Chemistry, which he shared with Friedrich Bergius, another BASF employee.

Chemically, the modern Haber-Bosch process looks something like this:

$$Energy + N_2 + 3\,H_2 \xrightarrow{Fe,\,Fe_3O_4,\,Al_2O_3} NH_3 + H_2 \tag{3}$$

The energy comes from temperatures in the range from 750°F to 3000°F, with pressures as great as 350 times atmospheric pressure at sea-level, and the hydrogen comes from natural gas. Today the world is so hungry for nitrogen that the Haber-Bosch process is responsible for 3% of the world's carbon emissions and consumes roughly 3% of the world's natural gas. Not surprisingly, scientists are constantly looking for ways to improve nitrogen fixation. Areas of current research including finding better catalysts[9] and using researching how biological systems work.[10,11,12] After all, alfalfa is able to fix nitrogen at room temperature with just air, water, sunlight, and some clever microbes.

5.1.2 Modeling Chemical Reactions

One way for industry to develop improved nitrogen fixation catalysts would be to better understand what is happening at the atomic level when nitrogen gas becomes ammonia inside those microbes. Chemists think of this process in terms of some chemical bonds being broken while new chemical bonds are created. Much of modern chemistry is devoted to describing and predicting the behavior such chemical bonds.

Except there is really no such thing as a chemical bond! While students in high school chemistry class learn to visualize bonds as little black lines connecting letters (*e.g.*, N≡N), "bonds" and indeed our entire model of chemical reactions are really just approximations for Schrödinger wave equations that evolve over time and describe the probability that a collection of mass, charge and spin will interact with our measuring devices. It is just far too hard to write down such wave equations, let alone solve them. Meanwhile, the mental models of chemical bonds and other approximations developed over the past 150 years all work pretty well, especially with ongoing refinements, and so chemists continue to use these approximations.[13]

[9] Ashida et al., "Molybdenum-catalysed ammonia production with samarium diiodide and alcohols or water" (2019).

[10] Molteni, "With Designer Bacteria, Crops Could One Day Fertilize Themselves" (2017).

[11] *Biological Nitrogen Fixation: Research Challenges—A Review of Research Grants Funded by the U.S. Agency for International Development* (1994).

[12] Manglaviti, *Exploring Greener Approaches to Nitrogen Fixation* (2018).

[13] A current textbook about the chemical bond reminds its readers that there are no electrons spinning around the atoms, only a "charge wave surrounding the nucleus." (Brown, *The Chemical Bond in Inorganic Chemistry: The Bond Valence Model, 2nd edition* [2016], Chapter 2) (Figure 5.2.) Nevertheless, the author continues, "chemists have largely rejected this simple wave picture of the

Figure 5.2: McMaster University Professor Emeritus I. David Brown observes: "An electron is the smallest quantum of charge that can have an independent existence, but the free electrons that are attracted to a nucleus in order to form a neutral atom cease to exist the moment they are captured by the nucleus. They are absorbed into the charge wave and, like Lewis Carroll's (1865) Cheshire Cat that disappears leaving only its smile behind, the electron disappears bequeathing only its conserved properties: charge, mass and spin, to the charge wave surrounding the nucleus."Brown, *The Chemical Bond in Inorganic Chemistry: The Bond Valence Model, 2nd edition* (2016), chapter 2

More accurate models that do a better job incorporating the underlying quantum physics would let chemists create more accurate predictions of how these things we call atoms rearrange during the course of a chemical reaction. Highly accurate models would let chemists design and try out catalyst candidates in a computer, without having to go to the trouble of actually synthesizing them in a lab. This is the world of *computational chemistry*, also called quantum chemistry, or even computational quantum chemistry, which uses the math of quantum mechanics to answer questions about the chemical nature of the world around us.

Wave equations describe probabilities, so predicting the behavior of atoms at the quantum level requires programs that explore probability distributions. One way to do this is with a Monte Carlo simulation (See **Section 5.1.2, "The Monte Carlo Method"**). Simulations take exponentially longer to run as the number of electrons in the system increases—a good rule of thumb is that each additional electron doubles the simulation's running time.

In the Haber-Bosch nitrogen fixation equation presented above, there are 14 electrons among the two nitrogen atoms and 6 hydrogen electrons for a total of 20 electrons. But do not forget that all-important catalyst: that is where the chemical dance of the electrons is happening. Iron has 26 electrons per atom, while Fe_3O_4 has 110, and Al_2O_3 has 50. There must be some extraordinarily complex chemistry

atom in favor of a hybrid view in which the charge is composed of a collection of electrons that are not waves but small particles, [with the] density of the charge wave merely represent[ing] the probability that an electron will be found at a given location."

happening at the interface of the gaseous nitrogen and the solid catalyst.

To understand that complex chemistry, a computational chemist creates a simulation of the electrons and nuclei. Into the simulation the chemist programs physical constants that have been measured over the decades as well as mathematical functions that represents the laws of quantum mechanics. The more electrons and nuclei, the more complex the simulation.

The math of quantum physics is based on probability, so all of those probabilistic interactions—many coin flips—become inputs to the simulation. For example, some of the random draws might have less electron charge in a particular location between the two nitrogen nuclei and more charge between the nitrogen and an iron nuclei that is interacting with some oxygen. This might sometimes push the two nitrogen nuclei slightly further apart—their electrostatic charges repel, after all—which might sometimes cause the charge probability rearrange a little more, and then all of a sudden …*wham!* …the two nitrogen nuclei can now pick up some free floating protons, and the physics simulation has converted simulated nitrogen into simulated ammonia!

Running this simulation with a classical computer requires many random draws, many crunchings of quantum mathematics, and a lot of matrix mathematics. Remember, classical computers are *deterministic by design*. To explore what happens when 4 random variables encounter each other, the computer takes random draws on each four variables and crunches the math. One cannot simply explore what happens when the most-probable value of each variable happens, because there might be some important outcome when three of the variables are in a low-probability configuration.

If it takes 10 seconds to simulate a single random variable, it will take on the order of $10 \times 10 \times 10 \times 10 = 10^4 = 1,000$ seconds to simulate 4 random variables. With 10 random variables (and without any optimization), it will take 10^{10} seconds or 115,740 days—roughly 317 years.

These days, a computation that takes 317 years is not a big deal, provided that the computation consists of many individual problems that can be run in parallel. Good news: quantum simulations are such a problem! As we write this book in 2020, cloud providers will rent a computer with 96 cores for roughly $5/hour. One can rent 100 of those computers for $500/hour and solve the 317-year problem in 12 days for $6000. Alternatively, one can rent 1,000 of those computers and solve the problem in 29 hours—for the same price of $6000. (This demonstrates why cloud computing is so attractive for these so-called *embarrassingly parallel* workloads.)

Today's massive cloud computing data centers provide only *linear* speedup for these hard problems: if 1,000 computers will solve the problem in 29 hours, then 10,000 computers will solve the problem in 2.9 hours. And there's the rub: absent a more elegant algorithm, each additional electron in our hypothetical simulation increases the problem's difficulty *exponentially*. With 20 electron variables, the problem takes on the order of 10^{20} seconds or 3,168,808,781,402 years—3168 billion years!—which is more time than anyone has.[14] Even with a million 96-core computers (a speedup of 96 million), our hypothetical computation would take 33,008 years, which is still too long. Classical computers are simply not well-suited to simulating probabilistic quantum physics.

[14] Current estimates are that the universe is somewhere between 15 and 20 billion years old.

The Monte Carlo Method

Modeling nuclear reactions was one of the first uses of electronic computers in the 1940s. Stanislaw Ulam at Los Alamos was trying to create a mathematical model for the movement of neutrons through material. Ulam couldn't create an exact model, so instead he ran hundreds of individual mathematical experiments, with each experiment modeling a different path of probabilistic interactions between a neutron and the material. For each of these interactions, the program used a "random digit generator" to choose between more probable and less probable possibilities. Ulam called this the *Monte Carlo method*, named after the casino where his uncle frequently gambled.[a]

Ulam shared his idea with fellow scientist John von Neumann, who directed the team at University of Pennsylvania to program the ENIAC to carry out the computations.

One of the requirements of randomized algorithms like Monte Carlo is that the random numbers must be truly random. Generating such numbers requires an underlying source of physical randomness, something that the early computers didn't have. Instead, the computer systems of the day used a deterministic algorithm called a *pseudorandom number generator* to generate a sequence of numbers that *appeared* random, but which were actually determined from the starting "seed." Von Neumann later quipped: "Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin."[b]

It is necessary to use algorithms such as the *Monte Carlo method* when modeling quantum interactions, because it is not possible to solve the Schrödinger wave equation for even mildly complex systems.[c] The result was the successful fusion bomb test in November 1952 and decades of employment for physicists at weapons laboratories around the world. By the 1990s modeling had gotten so good that it was no longer necessary to even test the bombs, and the United States signed (but did not ratify) the Comprehensive Nuclear-Test-Ban Treaty.

[a] Metropolis, "The Beginning of the Monte Carlo Method" (1987).

[b] Neumann, "Various Techniques Used in Conneciton With Random Digits" (1951).

[c] Random sampling can also be used to find approximate integrals to complex mathematical functions: instead of attempting to find an exact solution, the approach is to evaluate the function at a number of randomly chosen locations and interpolate. This is similar to statistical sampling, except that what's being sampled is a *mathematical universe*, rather than a universe of people or objects.

It's widely believed that quantum computers will be able to efficiently solve problems involving quantum modeling of chemical reactions. Even the "quantum simulators" discussed here, special-purpose machines constructed to solve a specific problem, should be dramatically faster than all of the world's computers working forever…provided that we can scale the quantum simulators to be large enough. As such, quantum chemistry simulation is likely to be the first application for quantum computers in which they are used for something other than doing research and writing papers about quantum computers.

5.2 Quantum Factoring (Shor's Algorithm)

As we explained in Section 4.8, "Aftermath: The Quantum Computing Baby" (p. 122), Peter Shor's discovery of an algorithm that can rapidly break numbers down into their prime factors sparked the world's interest in quantum computing. In this section we will describe why Shor's algorithm was so important, how it became a driver of quantum computing, and why it is no longer a driver—at least, not in the public, commercial world. (See Section 3.5.6 (p. 86) for a discussion of what we mean by "rapidly.")

To understand why Shor's algorithm is such a big deal, we start with a discussion of public key cryptography. In Section 5.2.3 (p. 146) we discuss how a quantum computer makes factoring faster. We will then explore whether Shor's algorithm running on a quantum computer would truly be faster than anything that could ever run on a classical computer, or whether we just need better math.

5.2.1 An Introduction to Cryptography

In modern usage, we use the word "cryptography" to describe the body of knowledge involved in creating and solving secret codes. Here the word "code" means a system for representing information, while "secret" implies that something about the code allows people who know the secret to decode its meaning, while people who do not know the secret can not.

Secret Key Cryptography

One of the oldest know codes it the "Caesar cipher," which was reportedly used by Julius Caesar to messages to his generals. Messages are encrypted character-by-character by shifting each letter forward in the alphabet by three positions, so T becomes Q, H becomes E, E becomes B, the letter C wraps around to Z, and so on. To decrypt messages simply shift in the other direction. QEB ZXBPXO ZFMEBO FP KLQ SBOV PBZROB, that is, THE CAESAR CIPHER IS NOT VERY SECURE.

The Caesar cipher is called a *secret key algorithm* because the secrecy of the message depends upon the secrecy of the key, and the same key is used to encrypt and decrypt each message. It's not a very good secret key algorithm, because once you know the secret—shift by three—you can decrypt any encrypted message. We call this number three the *key* because it is the key to decrypting the message! You can think of the Caesar cipher as lock which fits over the hasp that is used to secure a wooden box, and the number *three* as a key that opens the lock.

We can make the algorithm marginally more complicated by allowing the shift to be any number between 1 and 25: that creates 25 possible encryption keys, so an attacker needs to figure out which one is in play. It's still not very hard to crack the code.

There are lots of ways to make this simple substitution cipher stronger, that is, to make it harder for someone to decrypt or "crack" a message without knowing the secret piece of information used to encrypt the message in advance. This is directly analogous to making the lock on the box stronger. For example, instead of shifting every letter by the same amount, you can make the encrypted alphabet a random permutation of the decrypted alphabet. Now you have a word puzzle called

a cryptogram. These can be easy or hard to solve depending on the length of the message, whether or not the message uses common words, and the number of times each letter is present in the message.

Humans solve these puzzles by looking for patterns in the encrypted message, called a *ciphertext*. We can eliminate such patterns by encrypting each letter with a different key. Now there are no patterns! This kind of encryption algorithm is sometimes called a Vernam cipher (named after its inventor, Gilbert Vernam) or more commonly a *one-time pad* (because spies of yore had encryption keys written on pads of paper, with instructions to use each key once and then destroy it). One-time pads are hard to use in practice, because the key needs to be both truly random and as long as the original message. We discuss them more in Section 7.4 (p. 203).

Public Key Cryptography

For all of human history until the 1970s, cryptography existed as a kind of mathematical deadbolt, in which each encrypted message was first locked and then later unlocked by the same key. There were thus four principle challenges in creating and deploying a working encryption system: 1) Assuring that the sender and the intended recipient of an encrypted message had the same key; 2) Assuring that no one else had a copy of the correct key; 3) Assuring that the correct key could not be guessed or otherwise discovered by chance; 4) Assuring that the message could not be decrypted without knowledge of the key. (See Figure 5.5)

All of this changed in the 1970s with the discovery of public key cryptography, a term used to describe encryption systems in which a message is encrypted with one key and decrypted with a second.

Originally called *non-secret* encryption, it is now generally believed that public key cryptography was discovered in 1973 by James Ellis, Clifford Cocks and Malcolm Williamson[15] in at the Government Communications Headquarters (GCHQ), the United Kingdom's signals intelligence and information assurance agency (roughly the UK's equivalent of the U.S. National Security Agency (NSA)). The UK intelligence agency reportedly shared the discovery with the NSA,[16] but neither sought to exploit the invention. The basic idea was then re-discovered at Stanford by professor Whitfield Diffie and professor Martin Hellman, whose paper "New Directions in Cryptography" inspired Ronald Rivest, Adi Shamir and Leonard Adleman at MIT to create a working public key system.[17,18]

The basic concept of public key cryptography is a mathematical lock that is locked with one key and unlocked with a second. The key that locks (encrypts) is

[15] Ellis, Cocks, and Williamson, *Public-key Cryptography* (1975).

[16] Levy, "The Open Secret" (0199).

[17] Rivest, Shamir, and Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems" (1978b).

[18] The RSA crypto system was published first in Martin Gardner's column in Scientific American (Gardner, "Mathematical Games: A new kind of cipher that would take millions of years to break" (1977b)), in which the RSA-129 number that we will discuss on 192 was first published. In that article, the MIT professors famously offered US$100 to anyone who could factor the 129-digit number or otherwise decrypt the message that they had encrypted with it. The professors also offered a copy of their technical paper to anyone who sent a self-addressed stamped envelope to their offices at MIT. Rivest discusses this in his Turing award lecture Rivest, "The Eary Days of RSA: History and Lessons" (2011), following Adleman's lecture Adleman, "Pre-RSA Days: History and Lessons" (2011), and followed by Shamir's Shamir, "Cryptography: State of the Science" (2011).

Figure 5.3: A locked suggestion box is a good metaphor for public key cryptography. To protect your message, just drop it through the slot. To retrieve your message, you must unlock the padlock and open the lid. Photograph by Hashir Milhan of a suggestion box in Sri Lanka, photographed on October 3, 2007. CC BY 2.0

called the *public key*, while the key that unlocks (decrypts) is the *private key*. The two keys are mathematically linked and need to be made at the same time.[19]

A locked suggestion box is a good mental model for how a public key cryptography works: to encrypt something, write it on a piece of paper and drop it into the locked box. Now the only way to get that message back is by unlocking the box and retrieving the message. In this example, the slot in the box represents the public key, and the key that unlocks the padlock represents the private key (Figure 5.3).

The great advantage of public key cryptography is that it dramatically simplifies the problem of key management. With public key cryptography, each person in an organization simply makes their own public/private keypair and then provides their public key to the organization's central registry, which then prints a phone book contain each employee's name and public key, then sends each employee their own copy. Now any employee can send an encrypted message to any other employee by simply looking up the intended recipient's key in the directory, using that key to encrypt a message, and then sending the message using the corporate email system. Nobody will be able to decrypt the message—not even the system administrators who run the corporate email system or the employee who printed the phone book.

Public key cryptography can also be used to create a kind of *digital signature*. In this case, the encrypting key is retained and the decrypting key is published. To sign a document, just encrypt it with your private key, then publish result as a *signature*. Anyone who has access to your public key (from the directory) can decrypt your signature and get back to the original document. If you practiced

[19] There is a more refined version of public key technology called *identity-based encryption* (IBE) that allows the keys to be made at separate times by a trusted third party. IBE was proposed by Adi Shamir in 1984 Shamir, "Identity-Based Cryptosystems and Signature Schemes." (1984). Two working IBE systems were developed in 2001, one by Dan Boneh and Matthew K. Franklin (Boneh and Franklin, "Identity-Based Encryption from the Weil Pairing" (2001)), the other by Clifford Cocks of GCHQ fame (Cocks, "An Identity Based Encryption Scheme Based on Quadratic Residues" (2001)).

good cryptographic hygiene and no one has obtained your private key, now called the *signing key*, then we now have good proof that you alone could have signed the document.

It is still possible for employees to send and receive messages within an organization without using public key cryptography, but the procedures are more involved. One possibility is for the central authority to create different secret key for every pair of employees that needs to communicate, then to send each pair of employees all of the keys that they need in a sealed envelope. This approach has the feature that individuals can only exchange encrypted email with other individuals with whom they are authorized to exchange messages. Another feature is that the central key-making authority can in theory decrypt any message exchanged by a pair of employees if it retains that pair's key, although the authority can choose to destroy its copy if it wishes to allow the pair to communicate without the possibility of eavesdropping. This is the sort of system that military organizations traditionally set up, and it is presumably what GCHQ and the NSA were using in the 1970s, which is why they saw no need to develop the non-secret encryption that Cocks and Ellis had invented: GCHQ and NSA already had a system that was well-developed and deployed to meet their organizational requirements, and the benefits of digital signatures were not immediately obvious.

For the academics at Stanford and MIT, however, the discovery of public key cryptography opened the door on a new area of intellectual pursuit that combined the fields of number theory and computation. It was an academic green field, full of wonder, possibility and low-hanging fruit. For example, in 1978, an MIT undergraduate named Loren Kohnfelder realized that digital signatures made it unnecessary for an organization to publish a directory of every employee's public key. Instead, the organization could have a single private/public keypair for the organization itself, and use the private key to sign each employee's public key. The employees could then distribute to each other their own public keys, signed by the organization's public key, to other employees as needed. As long as each employee had a copy of the organization's public key, they could verify each other's keys, and the organization would not need to send out a directory with every employee's public key. Today we call these signed public keys *digital certificates* and the central signing authority a *certificate authority*. With his 1978 undergraduate thesis, Kohnfelder had invented public key infrastructure (PKI).[20]

The following year, Ralph Merkle's PhD thesis[21] introduced the idea of cryptographic hash functions. A hash function is a mathematical function that takes an input of any size and produces an output of a fixed size. The basic concept was invented by IBM engineer Hans Peter Luhn in the 1950s.[22] Merkle's innovation was to have hash functions that produced an output that was both large—more than a hundred bits—and unpredictable, so that it would be computationally infeasible to find an input that produced a specific hash. Given such a function, you don't need to use to sign an entire document, you just need to sign a hash of the document. Today we call such things *cryptographic hash functions* and there are many, the most prominent being the U.S. Government's Secure Hash Algorithm version 3 (SHA-3).

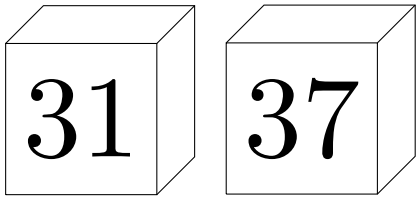
[20] Kohnfelder, "Towards a practical public-key cryptosystem" (1978).
[21] Merkle, *Secrecy, Authentication and Public Key Systems* (1979).
[22] Stevens, "Hans Peter Luhn and the Birth of the Hashing Algorithm" (2018).

In the end, the discovery catalyzed interest and innovation in cryptography. Academics and entrepreneurs were attracted to the field; they launched companies and ultimately set in motion the commercialization of the Internet, which was only possible because public key cryptography allowed consumers to send their credit card numbers securely over the Internet to buy things.

A demonstration of RSA public key cryptography

The most widely used public key encryption system today is RSA, named after its inventors Rivest, Sharmir and Adleman. The system is based on math that is beyond this book but which is easy to find if you have interest, and easy to understand if you understand basic number theory. For the purpose of this demonstration we will just assume that you have a set of magic dice that always roll prime numbers and a box that given these two prime numbers p and q outputs two sets of numbers: your public, encrypting key $\boxed{\text{e,n}}$ and your private, decrypting key $\boxed{\text{d,n}}$.

We roll the prime number dice and get two prime numbers:



We drop these into our key generator and get two keys:

public key	
e	7
n	1147

private key	
d	463
n	1147

To encrypt a plaintext message P (which is a number) to produce an encrypted message C (which is another number), we use this mathematical formula:

$$C = P^e \pmod{n} \tag{4}$$

This means multiply the number P by itself e times and then take the integer remainder after dividing the resultant by n. For example, the number 53 (which represents the letter "S") encrypts as 914:

$$C = 53^7 \pmod{1147} = 1,174,711,139,837 \pmod{1147} = 641 \tag{5}$$

To decrypt the number 914, we follow roughly the same procedure using the values for d and n:

$$P = C^d \pmod{n} = 641^{463} \pmod{1147} = 53 \tag{6}$$

We haven't expanded 641^{463} above; the number is 1300 digits long. RSA implementations use a variety of mathematical tricks to avoid naively computing these numbers—for example, you can apply the modulo after *each* multiplication to prevent the intermediate number from getting too large—but it's easy enough to do

the math directly using the Python programming language if you want to check our work.

The RSA algorithm is secure as long as you can't compute the number d knowing e and n (and provided that you follow some implementation guidance that was developed after the algorithm was first published[23]). It turns out that it's easy to compute d, however, if you can factor n. Not a lot was known about the difficulty of factoring numbers in 1977, although the best factoring numbers took exponentially more time as the length of the number being factored increases. That's still the case today. This may be something inherent in the nature of factoring, or it may reflect a limitation in our knowledge. After more than forty years of intensely studying the question, mathematicians, computer scientists and cryptographers still don't know.

5.2.2 Forty Years of Public Key Cryptography

Despite the fact that humanity is still unsure about the fundamental hardness of factoring, we have learned a lot about cryptography over the past forty years. Here we focus on three significant improvements: speed, algorithmic improvements, and key length.

Cryptographic Speed

The computers of the 1970s were too slow for public key cryptography to be practical: a single RSA encryption or decryption on a computer could take as long as 30 seconds. By the 1980s computers were fast enough that it took just a few seconds, and some companies developed and marketed *cryptographic co-processors* that could accelerate the math required to make RSA run fast as well as store the RSA private keys in tamper-proof hardware. By the 1990s general purpose microprocessors were fast enough that special purpose hardware was no longer needed, and these days most microprocessors include special instructions and dedicated silicon that can be used to accelerate both secret and public key cryptography.

As a result, cryptography has gone from being a technology that was only used occasionally, when it was absolutely needed, to a protection that is always enabled. For example, the early web used encryption just to send passwords and credit card numbers, sending everything else over the Internet in plaintext. These days encryption is the default, and web browsers warn when any page is downloaded without encryption.[24]

Algorithmic Improvements

Working together, cryptographers and security engineers have also made stunning improvements to cryptographic systems, making them both faster and more security.

Although the underlying math of RSA is sound, cryptographers developed that there were many subtle nuances to using it practical applications. For example, we simply encrypt letters one code at a time, as we did in the example above,

[23] For an example of up-to-date guidance, see Housley, *Use of the RSAES-OAEP Key Transport Algorithm in Cryptographic Message Syntax (CMS)* (2003).

[24] Our understanding of Internet security has also expanded, so now we know that a single advertisement, image or font downloaded without encryption over the Internet can be leveraged by an attacker to compromise your computer's interactions with a remote website.

Elliptic Curve Public Key Cryptography

In the 1980s cryptographers Neal Koblitz and Victor S. Miller suggested that mathematical constructs called "elliptic curves over finite fields" might provide sort of operations required for a working public key cryptography system.[a] "These elliptic curve cryptosystems may be more secure, because the analog of the discrete logarithm problem on elliptic curves is likely to be harder than the classical discrete logarithm problem," wrote Koblitz.

Over the following years elliptic curve cryptography (ECC) was developed and standardized in the 1990s; the American National Standards Institute (ANSI) adopted ANSI X9.62, the Elliptic Cure Digital Signature Algorithm (ECDSA), in 1999. The US National Security Agency aggressively promoted ECC over RSA. And why not? Compared with the RSA, ECC keys could be dramatically shorter and achieve similar security properties. ECC implementations were also dramatically faster than at encrypting and decrypting, and thus used less power. This made ECC especially popular for mobile computing such as cell phones, and for web servers that receive significant amounts of traffic from many different parties.

To date, the primary disadvantage of ECC has been the need to license patents from the Certicom, the Canadian company founded in 1985 to commercialize ECC technology. Whereas the RSA algorithm was protected by a single patent that was limited to the US and expired in 2000[b], Certicom aggressively patented many different aspects of both ECC math and efficient ECC implementations. ECC also took a hit when *The New York Times* reported in 2013 that a random number generator based on ECC had been intentionally weakened by the U.S. Government.

More recently, a second concern regarding the security of ECC is that the number theory of elliptic curves is less studied than the number theory that underlies the RSA algorithm. In 2015, cryptographers Neal Koblitz and Alfred Menezes noted that the NSA was moving away from elliptic curve cryptography after having previously been enthusiastic about the technology[c].

Like RSA, the math that underlies ECC is vulnerable to quantum computers. And since the ECC keys are significantly shorter than RSA keys, as quantum computers scale up they will be able to crack ECC keys currently in use before they are able to crack RSA keys that offer similar security. Just how much time would elapse between the cracking of and ECC key and the equivallent of RSA key is unknown. Assuming that there are no fundamental scientific limits to scaling up the quantum computer, "it's just a matter of money," observed Koblitz and Menezes.

[a] Koblitz, "Elliptic Curve Cryptosystems" (1987); Miller, "Use of Elliptic Curves in Cryptography" (1986).

[b] Adleman, Rivest, and Shamir, *Cryptographic Communications System and Method* (1983).

[c] Koblitz and Menezes, "A Riddle Wrapped in an Enigma" (2016).

an adversary has a straightforward method to attack the ciphertext. The adversary can encrypt all possible combinations of messages using the public key until a match emerges with the ciphertext. The attacker can do this because the attacker always has access to the target's public key—that's the core reason we are using public key cryptography. This approach of trying every possible combination is called a *brute-force attack* or a *key-search* attack. For this reason, whatever message that's encrypted is always combined with random string of bits, called a pad. With a long pad it's impossible for the attacker to try every combination; padding also assures that the same message will always encrypt differently, which makes cryptanalysis harder. RSA without a pad is called Textbook RSA: it's good enough for textbooks, but it doesn't actually protect your message.

Engineers developed clever encryption protocols that limit the number of public key operations that need to be computed. This is done by combining public key cryptography with traditional secret key cryptography. For example, an hour of HD video (roughly 10GB of data, with compression) can be encrypted with a single public key operation. This is done by first encrypting the video with randomly generated secret key, and then encrypting the secret key with a public key algorithm. This approach is sometimes called a *hybrid system*; it is the approach that is used by both the Trusted Layer Security (TLS) protocol and the Secure Shell (SSH) protocols used to send information over the Internet.

5.2.3 Cracking Public Key with Shor's Algorithm

Here is one measure of public key technology's success: today the vast majority of information sent over the Internet is encrypted with TLS, the hybrid system described above (p. 146) that uses public key technology to exchange a session key, and then uses the session key to encrypt the information itself. If you are viewing web pages, you are probably using TLS.

TLS is a sometimes called a *pluggable protocol*, meaning that it can be used with many different encryption algorithms—it's as simple as plugging-in a new algorithm implementation. When you type a web address into your browser, your browser opens a connection to the remote website and the remote website sends to your browser the website's public key certificate, which is used to establish the website's identity. The two computers then negotiate which set of algorithms to use based on which algorithmic plug-ins the web server and the web browser have in common. Today there are tools built into most web browsers to examine website certificates and the TLS connections, but these tools can be confusing because the same website can appear to provide different certificates at different times. This is typically because a single "website" might actually be a collection of several hundred computers, all configured with different certificates.

Because the public key certificate is sent over the Internet when a web page is downloaded, anyone who can eavesdrop upon and capture the Internet communications now has all of the information that they need to decrypt the communications, provided that they have sufficient computing power to derive the website's matching private key from its public key—that is, to "crack" the public key. In the case of RSA, this is the very factoring problem posed by decrypting the Scientific American message that was encrypted with RSA-129. In the case of elliptic curve algorithms, other mathematical approaches are used to crack the public key.

Before the invention of Shor's algorithm, the fastest factoring algorithms required exponentially more time to execute as number of bits in the public key increased. Shor's algorithm uses an approach for factoring that has only *polynominal complexity*: longer keys still take longer to factor, just not exponentially longer. The catch is that Shor's algorithm requires a working quantum computer with enough stable qubits to run a quantum algorithm that helps to factor the number in question: with perfect qubits, factoring the numbers used in modern cryptographic system would require thousands of qubits. But if the qubits have even the smallest amount of noise, then it will be necessary to use quantum error correction, increasing the number of qubits needed roughly a hundred million (see 151).[25] Of course, the the first computer to use transistors was built in 1953 at the Manchester University: it had just 92 point-contact transistors that had been constructed by hand. Today's Apple M1 microprocessor has 16 billion transistors, built with a feature size of just 5-nanometers.

Shor's algorithm contains a classical part and a quantum part. The classical part contains some of the same number theory that powers RSA encryption, which isn't terribly surprising since both are based prime numbers, factoring, and Euler's Theorem. To use RSA, the *code-maker* randomly chooses two prime numbers, p and q. These numbers are multiplied to compute N and also used to create the public key and private key. With Shor's algorithm, the attacker just has the public key, which contains N. The attacker also has access to a quantum computer that can perform two quantum functions: the quantum Fourier transform and quantum modular exponentiation. With this functions, the attacker can factor N, learning p and q, and re-generate the code-makers private-key. With this private key, the attacker can decrypt any message that was encrypted with the code-makers public key.

Alas, explaining either the classical *or* the quantum aspects of Shor's algorithm require more math and physics that we require for readers of this book, so we refer interested readers with sufficient skills to other publications, including the second version of Shor's 1997 paper[26] which can be downloaded from arXiv[27], as well as the Wikipedia article on Shor's algorithm.[28]

If you had a quantum computer with sufficiently many stable qubits to run Shor's algorithm, and *if* you had recorded the complete encrypted communication between a web server and a web browser at anytime from the dawn of the commercial Internet through today, *then* decrypting that communication would be straightforward.

For example, consider an unscrupulous internet service provider (ISP) that wants to eavesdrop on one of its user's email. Before 2008, the ISPmerely needed to capture the user's packets and reassemble them into web pages—a fairly trivial task.[29] But

[25] Mohseni et al., "Commercialize quantum technologies in five years" (2017).

[26] Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer" (1997).

[27] https://arxiv.org/abs/quant-ph/9508027v2

[28] With some amusement, we note that in January 2020 the quantum algorithm section of the Wikipedia article contained this note: "This section **may be too technical for most readers to understand**. Please *help improve it* and *make it understandable to non-experts*, without removing the technical details." We encourage any of our readers with sufficient skill to accept this challenge.

[29] Ohm, "The Rise and Fall of Invasive ISP Surveillance" (2009); Bellovin, "Wiretapping the Net" (2000).

since 2008 Google has allowed users to access the server using encryption[30], and in 2010 Google made encryption the default. Once the user started using encryption, the nosy ISP would be out-of-luck: the web pages would be encrypted using RSA cryptography. However, if the ISP had recorded these packets and later rented time on a sufficiently large quantum computer, all the ISP would n eed to do is to extract Gmail's public key certificate, factor N, apply the RSA key generation algorithm to compute the private key, use the private key to decrypt something the *master secret* that was used to encrypt the web pages, and then use the master secret to decrypt the individual pages. This is not hard to do—there exists software that readily performs all of the reassembly and decryption—provided that you have a copy of the server's private key.

If you had captured the packets and *didn't* have a quantum computer, there are still other ways to get that private key. You might be able to get it by hacking into Google's server and stealing it. Alternatively, you might be able to bribe someone at Google, or even obtain a court order against Google to force the company to produce its private key or use it to decrypt the captured transmission.

In 2011, Google made a change to its computers to remove the risk that a stolen private key could be used to compromise the privacy of its service users: Google implemented *forward secrecy* by default.[31] Also known as *perfect forward secrecy*, the term is applied to security protocols that use session keys that are not revealed even if long-term secrets used to create or protect those session keys are compromised. In the case of web protocol, forward secrecy is typically assured by using digital signatures to certify an ephemeral cryptographic key created using the Diffie-Hellman key agreement protocol, which is an interactive public key encryption algorithm that allows two parties to agree on a shared secret.[32]

Google's 2011 move to forward secrecy is a boon for privacy: it means that after the conclusion of communications between a user's web browser and the Gmail server, not even Google can use its own private key to decrypt communications that might have been covertly recorded. This is because Google's Gmail server destroys its copy of the ephemeral encryption key that was used to encrypt the session when the session concludes.

It turns out that the forward secrecy algorithm used by Google, the Diffie-Hellman key agreement protocol, is also vulnerable to an attacker that has a quantum computer. This is because the security of the Diffie-Hellman algorithm depends on the difficulty of computing something known as a *discrete logarithm*, and the quantum part of Shor's algorithm can do that as well. So those packets recorded by the ISP in our scenario are still vulnerable to some future attacker with a large-enough quantum computer.

[30] Rideout, "Making security easier" (2008).

[31] Langley, "Protecting data for the long term with forward secrecy" (2011).

[32] Diffie-Hellman is an *interactive* algorithm because performing the protocol requires the two parties to exchange information with each other and act upon the exchanged information. In this way it is different from RSA, which is a *non-interactive protocol*, because it is possible for one party to encrypt or decrypt information using RSA without the active participation of the other party.

5.2.4 Evaluating the quantum computer threat to public key cryptography

Factoring is clearly a problem that quantum computers will be able to solve faster than classical computers if they become sufficiently large. Will quantum computers ever actually be large enough to pose a threat to public key cryptography? *We don't know the answer to this question today.*

In 2001, a 7-qubit bespoke quantum computer constructed by Isaac Chuang's group at IBM Alamaden Research Center successfully factored the number 15 into its factors 3 and 5.[33] The number 15 is represented in binary by four bits: `1111`. The number 15 is also, not coincidentally, the smallest number that is not prime, not even, and not a perfect square. So realistically, it's the smallest number that the IBM team could have meaningfully factored.[34]

The quantum "computer" that IBM used doesn't look anything like our modern conception of a computer: it was a tube containing a chemical that IBM had synthesized especially for the experiment, a chemical called a "perfluorobutadienyl iron complex with the inner two carbons," and with chemical formula $F_2C{=}C(Fe(C_5H_5)(CO)(CO))CF{=}CF_2$ (Figure 5.4). The quantum circuit was played through the tube as a series of radio frequency pulses, and the qubits were measured using nuclear magnetic resonance (NMR), a procedure in which a material is placed in a strong magnetic field and probed with radio waves at at different frequencies. We discuss NMR-based quantum computers in Section 4.8.2 (p. 125).[35]

Since IBM's demonstration, other researchers have factored other numbers on quantum computers. None of these approaches have managed to factor a number out of reach of a conventional computer. Most of the numbers factored, in fact, can be factored with pen-and-paper. For example, in 2012 a team led by Nanyang Xu at the University of Science and Technology of China, Hefei, successfully factored the number 143 using "a liquid-crystal NMR quantum processor with dipole-dipole couplings."[36] The factors were 11 and 13, of course. What's exciting is that the researchers used a different factoring approach called *adiabatic quantum computation* (AQC), using only four qubits. In 2014, Nikesh Dattani at Kyoto University and Nathaniel Bryans at University of Calgary posted a follow-up article to the arXiv

[33] Vandersypen et al., "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance." (2001).

[34] Even numbers are easy to factor: just divide them by two. Numbers that are perfect squares are also easy to factor: just take their square root, which can be quickly computed using Newton's method. The number 15 is the smallest non-even number that is the product of two different primes: three and five.

[35] It may seem implausible that a tube containing a solution of a specially synthesized compound inside a scientific instrument is actually *computing*, at least in the way that we typically think of the term. But the IBM experiment demonstrated that the computational media responded in a way that was consistent with factoring the number 15, producing the numbers 3 and 5.

It turns out that computing is more fundamental than electronics, and there are many different media that can be used for computation. For example, in the 1970s Danny Hillis created a computer from Tinkertoy rods and wheels that could play the TicTacToe. "It could have been built by any six-year old with 500 boxes of tinker toys and a PDP-10," Hills wrote at the time (Hillis and Silverman, *Original Tinkertoy Computer* (1978)). Another improbable computing medium is the seemingly haphazard but highly structured collection of lipids, proteins, nucleic acids, small amine molecules, amino acids and neuropeptides that make up the human neurological system.

[36] Xu et al., "Quantum Factorization of 143 on a Dipolar-Coupling Nuclear Magnetic Resonance System" (2012).

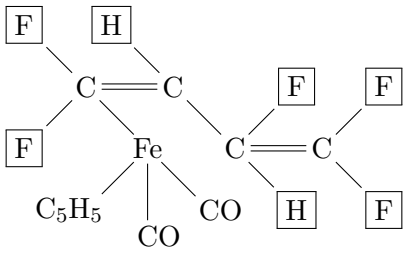


Figure 5.4: The perfluorobutadienyl iron complex with the inner two carbons that IBM scientists synthesized in 2001 for the purpose of factoring the number 15. The seven qubits are represented by the five fluorine (F) and two hydrogen (H) atoms shown surrounded by a box. For details, see Vandersypen et al., "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance." (2001).

open-access archive purportedly showing published that the results of the Chinese researchers could also be used to factor the numbers 3 599, 11 663, and 56 153.[37][38] The work on AQC factoring is exciting because it suggests that research in quantum computing may eventually lead researchers to make fundamental discoveries about factoring or even the nature of computation, with results that could then be applied to both quantum and classical computers. Although there have been no such discoveries to date, the field of quantum factoring is still quite young compared with other branches of number theory.

As of January 2019, the current record for factoring published in the peer-reviewed literature is held by Chinese scientists, who factored the 7-digit (20-bit) number 1 005 973 using 89 qubits on a D-Wave quantum annealing machine. The team noted that by using a factoring algorithm based on quadratic unconstrained binary optimization (QUBO), the team was able to constrain the factoring problem to the type of qubits that D-Wave provides. "Factoring 1 005 973 using Shor's algorithm would require about 41 universal qubits, which current universal quantum computers cannot reach with acceptable accuracy," the authors noted wryly.[39] This development was exciting because it demonstrated a new use for the D-Wave annealer, discussed further in Chapter 6, which is limited to certain kinds of applications. The scientists reasoned that because D-Wave scaled its annealer from just 128 bits to 2 000 in just seven years, perhaps a machine capable of factoring the kinds of numbers used to secure today's commercial Internet might soon be constructed.

We disagree: such a capacity would require a D-Wave computer with significantly more qubits than seems likely for the foreseeable future. As of January 2021, D-Wave's largest system, the Advantage, has just 5 000 qubits.[40]) To crack the RSA systems that are used to protect today's commercial Internet would require the ability to factor 2048 or 4096-bit numbers.[41]

[37] Dattani and Bryans, *Quantum factorization of 56153 with only 4 qubits* (2014).

[38] The Dattani/Bryans work was covered by the news site Phys.org (Zyga, "New largest number factored on a quantum device is 56,153" [2014]), but the work did not appear in the peer-reviewed literature.

[39] Peng et al., "Factoring larger integers with fewer qubits via quantum annealing with optimized parameters" (2019).

[40] D-Wave Systems Inc., *D-Wave Announces General Availability of First Quantum Computer Built for Business* (2020).

[41] For comparison, as of February 28, 2020, the largest RSA challenge number to be publicly factored is RSA-250, a 250-digit, 829-bit number. *Factorization of RSA-250* (2020) The total amount

Even with this work on factoring—perhaps because of it—there is still wide agreement in the scientific community that a practical application of quantum computing to factoring is far off. It is unclear whether the winning system will be a universal quantum computer with stable qubits that can also factor, or a special purpose device designed to perform factoring quickly. The advantage of the first machine is generality. The advantage of the second is that it could likely be developed years before a general-purpose quantum computer, and it could probably be developed for less money, and possibly in secret.

Google scientists have projected that factoring a conventional RSA public key in use on the commercial internet today "would take 100 million qubits, even if individual quantum operations failed just once in every 10 000 operations."[42] A National Academies group assessed in 2019 that "…to create a quantum computer that can run Shor's algorithm to find the private key in a 1024-bit RSA encrypted message requires building a machine that is more than five orders of magnitude larger and has error rates that are about two orders of magnitude better than current machines, as well as developing the software development environment to support this machine." The authors of the report stated that it is "highly unexpected" that a quantum computer that can break a 2 000-bit RSA key will be built before 2030.[43]

5.2.5 Post-quantum cryptography

Fully realized, large-scale, and sufficiently error-free, quantum computers will mean that public key encryption systems based on the RSA, Diffie-Hellman, and Elliptic Curve systems are no longer secure. But this will not mean the end of public-key cryptography.

Since the discovery of public key cryptography in the 1970s, dozens public key encryption algorithms have been devised. Of these, many do not depend on the difficulty of factoring or computing a discrete logarithm, and as such these algorithms would not be crushed by Shor's algorithm and a suitably large quantum computer. In fact there are so many choices and they are all so significantly different that is not immediately clear which is the best.

To help the world make the decision, in 2016 NIST embarked on the Post-Quantum Cryptography (PQC) Standardization effort. At the time, NIST stated that the competition for a PQC asymmetric algorithm would likely be more complex than its successful competitions to pick the Advanced Encryption Standard (AES) and the Secure Hash Algorithm 3 (SHA-3). "One reason is that the requirements for public-key encryption and digital signatures are more complicated. Another reason is that the current scientific understanding of the power of quantum computers is far from comprehensive. Finally, some of the candidate post-quantum cryptosystems may have completely different design attributes and mathematical foundations, so that a direct comparison of candidates would be difficult or impossible"[44]

NIST started with a field of 82 algorithm candidates, which was reduced to 26

of computer time required to perform the computation "was roughly 2 700 core-years, using Intel Xenon Gold 6t130 CPUs as a reference (2.1Ghz)," the authors reported.Peng et al., "Factoring larger integers with fewer qubits via quantum annealing with optimized parameters" (2019)

[42] Mohseni et al., "Commercialize quantum technologies in five years" (2017).

[43] Grumbling and Horowitz, *Quantum computing: progress and prospects* (2019).

[44] National Institute for Standards and Technology, *Post-Quantum Cryptography* (2017).

DNA-based computing

DNA (deoxyribonucleic acid) is the polymerized molecule inside cells which carries inheritance information and is used to synthesize proteins. It has been called "the building block of life."

Before the event of quantum computers, many researchers thought that DNA's ability to encode and to reproduce information might also make DNA a useful substrate for computing. One of the foremost proponents of DNA computing was Leonard Adleman (the "A" of RSA), who is frequently credited with inventing the field.

Adleman demonstrated that it was possible to compute with DNA by encoding a small graph into a DNA molecule and then using biomolecular reagents "to solve an instance of the directed Hamiltonian path problem."[a] This was highly significant, because the Hamiltonian Path problem is NP-Complete: if DNA computing could solve it efficiently, and if the system could be scaled up, then DNA could be used to solve any other problem contained within the NP class. In particular, a DNA computer would be able to factor efficiently.[b]

Work on DNA computing has continued, which researchers developing a variety of DNA-based algorithms,[c] and a recent review of "DNA-based Cryptanalysis"[d] found that the field remains promising. But it has been eclipsed by quantum computing.

In addition to DNA-based computing, there have been significant breakthroughs in using DNA to encode information directly. This approach has the advantage that DNA storage is incredibly dense. In June 2019, a Boston-based startup called Catalog announced that it hand encoded all 16GB of Wikipedia into a set of DNA strands the size of a pencil eraser.[e] DNA is also stable over long periods of time; DNA is now routinely recovered from humans that lived thousands of years ago. Since DNA is the basis of life, the ability to transcribe DNA is likely to be re-invented by any future biologically-based civilization on Earth, should the current technological society fail. DNA thus makes an excellent backup medium not just for organizations, but also for the intellectual heritage of our civilization.

[a] Adleman, "Molecular computation of solutions to combinatorial problems" (1994).

[b] Factoring is not NP-complete, but it is contained with in the class of NP problems.

[c] Weng-Long Chang, Minyi Guo, and Ho, "Fast parallel molecular algorithms for DNA-based computation: factoring integers" (2005).

[d] Sadkhan and Yaseen, "DNA-based Cryptanalysis: Challenges, and Future Trends" (2019).

[e] Shankland, "Startup packs all 16GB of Wikipedia onto DNA strands to demonstrate new storage tech" (2019); *Catalog*.

algorithms in early 2019. In July 2020 NIST announced the "Round 3 candidates" for the competition, with four public-key and key-establishment algorithms under consideration as "finalists:" Classic McEliece[45], CRYSTALS-KYBER[46], NTRUE[47], and SABER[48]. Another three algorithms are under consideration for digital signa-

[45] https://classic.mceliece.org

[46] https://pq-crystals.org

[47] https://ntru.org

[48] https://www.esat.kuleuven.be/cosic/pqcrypto/saber/

ture algorithms: CRYSTALS-DILITHIUM[49], FALCON[50] and Rainbow[51]. There are also give alternative public-key encryption and three digital signature "alternate" algorithms. Each algorithm is being presented in a web-based seminar open to the public, with the previous presentations and videos archived on the NIST website. It is unclear when the process will be finished, but it is likely that the scientific community will have standardized a new family of asymmetric algorithms long before the availability of quantum computers with sufficient power to crack the algorithms in use today.

In the meantime, all of the algorithms that NIST is evaluating are published, several with accompanying intellectual property statement stating that the authors do not hold patents on the algorithms, have not filed for patents, and have no intention to file for patents. This means that the algorithms are available for experimentation now! And indeed, July 2016, Google announced that it had deployed its experimental CECPQ1 key agreement protocol in "Chrome Canary," the experimental, nightly build version of its popular Chrome web browser.

"Quantum computers exist today but, for the moment, they are small and experimental, containing only a handful of quantum bits," Google's software Engineer wrote in the company's Security Blog.[52] "However, a hypothetical, future quantum computer would be able to retrospectively decrypt any internet communication that was recorded today, and many types of information need to remain confidential for decades. Thus even the possibility of a future quantum computer is something that we should be thinking about today."

Google's protocol uses the conventional and PQC algorithms in parallel, so that both must be successfully attacked together, during the same session, in order for the contents of a protected session to be compromised.

One of the reasons that Google decided to experiment with live with PQC is that the PQC data structures are significantly larger and slower to compute than the data structures used today. Thus, it makes sense to experiment with this technology now, on a limited scale.

In 2019 Google and the webhosting company Cloudflare continued the experiment, jointly deployed an improved algorithm called CECPQ2. "With Cloudflare's highly distributed network of access points and Google's Chrome browser, both companies are in a very good position to perform this experiment."[53]

If you are interested in learning more about the PQC algorithms, Kwiatkowski's illustrated blog post does a great job explaining them, although it would be useful to have first taken a course in college-level algebra.

5.3 Quantum Search (Grover's Algorithm)

Two years after Shor showed that a large enough quantum computer would be able to factor the numbers used to secure the Internet, Lov Grover (also at Bell Labs) made a startling discovery: a properly constructed quantum computer could speed up all sorts of computations that have a certain mathematical property. The speedup was

[49] https://pq-crystals.org
[50] https://falcon-sign.info
[51] https://www.pqcrainbow.org
[52] Braithwaite, "Experimenting with Post-Quantum Cryptography" (2016).
[53] Kwiatkowski, "Towards Post-Quantum Cryptography in TLS" (2019).

not as significant as Shor's: instead of turning a problem that is computationally intractable into one that can be solved in just a few hours, Grover's algorithm gives a square-root speedup: if solving a problem takes on order of N steps without Grover, typically abbreviated $O(N)$, it now takes on the order of the square root of N steps—that is, $O(\sqrt{N})$. On the other hand, whereas Shor's algorithm can only be applied to the relatively obscure domain of number theory, Grover's algorithm can be broadly applied to a wide range of practical problems. Grover's algorithm is the second major quantum computing algorithm.

Later in this section we will discuss how Grover's algorithm can be used to crack a version of one of world's most popular encryption algorithms. We'll show why this was such a big deal at the time, and then discuss why it's not really a big deal any more. After that, we'll discuss other applications for Grover's algorithm. To get started, though, we need to further explore the world of cryptography and code cracking.

5.3.1 Symmetric Ciphers: DES and AES

In 1977 the U.S. Government adopted a standard algorithm for encrypting data that it unceremoniously named the Data Encryption Standard. Before the adoption of the DES, the few companies that sold data security equipment to the generally made up their own encryption algorithms and asserted that they were secure. This created a difficult commercial environment, because most customers (including most government customers) were not equipped to evaluate the vendors' claims. The DES solved this problem: after it was adopted, vendors could simply follow Federal Information Processing Standard 46: no longer did they need to claim that the algorithm they had cooked up in their labs was mathematically secure. This is the function of standards, and with the DES the standardization process worked beautifully. Both inside and outside the U.S. government, the algorithm was rapidly adopted and deployed.

The adoption of the DES was not without controversy, however. In choosing the DES, the National Bureau of Standards did not use an existing military encryption algorithm. Instead, NBS (the precursor to today's National Institute of Standards and Technology) invited submissions from industry and academia. The first submission round was unsuccessful. For the second round, IBM submitted an algorithm it had developed called Lucifer, based on a novel construction created by the German-born mathematician Horst Feistel (1915–1990).[54]

Ideally, symmetric block cipher algorithms like DES and Lucifer have the property that the only way to decrypt an encrypted message is by knowing (or guessing) the correct key. Clearly, one way to attack such a cipher is to try all possible keys—the brute-force approach. In practice there are other kinds of attacks; such attacks make it possible to correctly guess the decryption key without explicitly trying all of them.

[54]Feistel's family fled Germany in 1934. He enrolled at MIT in Physics and graduated in 1937, then proceeded to earn a master's degree at Harvard. At the outbreak of World War II Feistel immediately came under suspicion because of his German citizenship, but his talents were well recognized by others in the U.S. government: Feistel was granted U.S. citizenship on January 31, 1944, and awarded a top secret security clearance the following day. He worked at the U.S. Air Force Cambridge Research Center, MIT's Lincoln Laboratory, and MITRE, before moving to IBM.

Key Length

The most visible change in cryptography over the past forty years is way that cryptographic keys have steadily increased.

Key length is traditionally expressed in bits. A key length of two means that there are four possible secret keys: `00`, `01`, `10` and `11`. With a key length of three, there are eight possible secret keys: `000`, `001`, `010`, `011`, `100`, `101`, `110` and `111`. With 4 bits there are 16 possible keys, and with 8 bits there are 256. Concisely, if there are n bits, there are 2^n possible secret keys—the number of keys grows *exponentially* as the number of bits increases. With a strong secret key algorithm, it is necessary to try every possible key in order to crack the message: there are not algorithmic short-cuts.

Whereas adversaries will attack a message encrypted with a secret-key algorithm by trying to decrypt the message, attacks against public-key algorithms typically involving attacking the public key itself. In the case of RSA, such attacks involving factoring the product of the two prime numbers p and q. Such factoring is harder with longer public keys. As a result, engineers have used longer and longer public keys as computers have gotten better at factoring.

In the early days of the commercial Internet, web browsers supported an intentionally weak 512-bit RSA algorithm and a stronger 1024-bit algorithm. The idea was that the weakened algorithm was to be used outside the U.S. and for non-commercial applications, and the 1024-bit version was to be used within the U.S. for commercial applications. Today there are no significant export restrictions on cryptographic software and 2048-bit RSA (617 decimal digits) is widely used, although 4096-bit RSA (1234 decimal digits) systems are increasingly being deployed. For comparison, the original RSA-129 number is 426 bits (129 decimal digits), and the number 1147 used the example on page 143 is 11 bits (4 decimal digits).

The original Lucifer algorithm had a 128-bit key length (see the **sidebar "Key Length"**), but after analysis by the National Security Agency, the algorithm's internals were changed somewhat and the key the shortened to 56 bits. It was widely assumed at the time that the U.S. Government had intentionally weakened Lucifer because U.S. intelligence agencies didn't want an encryption algorithm adopted as a national standard that was too difficult to be cracked. In fact, we now know that the final DES algorithm with its 56-bit keys was *stronger* than the 128-bit algorithm: unlike Lucifer, DES was resistant to a cryptanalysis technique called "differential cryptanalysis" that was not widely known in the 1970s and would not be discovered by academic cryptographers until the 1990s.[55])

When DES was adopted in 1977 it was not feasible for an attacker to try all $2^{56} = 72\,057\,594\,037\,927\,936$ possible keys to crack a message, but this proved to be possible by the 1990s. To make DES stronger, some organizations adopted a variant called *triple-DES* in which DES was used three times over,each time with a different key, to encrypt a message. This produced an effective key size of 168-bits, but it was also three times slower than a single encryption. There were also lingering doubts as to whether or not the DES had vulnerabilities that had been intentionally hidden

[55] Coppersmith, "The Data Encryption Standard (DES) and its strength against attacks" (1994).

by its creators which might make even triple-DES suspect.

In the late 1990s, NIST ran a second public competition to select a new national encryption standards. This time the vetting process was public as well. After two years, NIST adopted the Advanced Encryption Standard (AES), a symmetric block encryption algorithm developed in the 1990s that is better than DES in every possible way.

AES has three primary modes of operation: AES-128, AES-192 and AES-256, with 128-bit, 192-bit and 256-bit keys respectively. In practice, only AES-128 and AES-256 are widely used: AES-128 is the fastest, for applications that require the fastest possible algorithm, and AES-256 for the applications where speed is not the most important factor. Because the strength of the algorithm doubles with each additional bit, AES-256 is at least 2^{128} times stronger than the 128-bit version.

In fact, the number 2^{128} is so impossibly large that it is not possible to crack a message encrypted with AES-128 using brute-force search on a classical computer: there is simply not enough time. For example, if you had five billion computers that could each try 90 billion AES-128 keys per second, it would take 24 billion years—roughly the age of the Universe—to try all possible AES-128 keys. Without a functioning quantum computer running Grover's algorithm, the only way that an AES-128 message will be cracked will be if a significant underlying mathematical vulnerability is found in the AES algorithm itself. Today such a discovery does not seem likely.

However, it may be possible to crack such messages using Grover's algorithm running on a sufficiently large quantum computer. We discuss this below in Section 5.3.3 (p. 159).

5.3.2 Brute-Force Key Search Attacks

As we mentioned above, messages encrypted with symmetric encryption algorithms can forcibly decrypted, or "cracked," by trying all possible keys in sequence. In Table 5.1 we show how this works in practice. We have an 8-character message that has been encrypted with a key that was specially created for this text. The first few attempts fail, but eventually we find one that succeeds. In an actual brute force search, the computer stops when it finds a decryption succeeds, but in the table we keep going we've tried all 72 quadrillion possibilities.

There are two technical challenges to conducting a key search attack: the time it takes to try all possible keys, and the difficulty of recognizing a correct decryption.[56] The time is determined by how many keys per second your code-cracking machine can attempt, and how many code-cracking machines you happen to have. For example, at Bletchley Park during World War II, the Bombe (see 60) designed to crack the three-rotor version of the German's Enigma code could cycle through all 17,576 possible rotor combinations in 20 minutes. With two of these machines, the British could try half the combinations on one machine and one half on the other, and crack a message in 10 minutes. Or they could attack two messages with

[56] Many treatises on cryptography and code breaking ignore the challenge of detecting when text is correctly decrypted. In practice, this challenge is readily overcome, provided that the attacker knows something about the format of the decrypted messages. This is called a *known plaintext attack*. In some cases the attacker can arrange for a message of its choosing to be encrypted by the system under attack; this is called a *chosen plaintext attack*.

Figure 5.5: A safe with a combination lock on its door is a good metaphor for secret key cryptography and symmetric ciphers. To protect your message, just enter the combination lock on the panel, open the safe, put in your message, and close the door. To retrieve your message, enter the same combination on the panel, open the door, and retrieve your message. Photograph by Dave L. Jones (EEVBlog), Wikimedia Commons Account Binarysequence. CC BY-SA 4.0

Trial	Binary Key (56-bits)	Decrypted Output	Text
0	0000 ... 0000	BE 47 A1 7A 2E 81 0E 8C	¾G¡z.•••
1	0000 ... 0001	62 59 0B B1 CB 67 8F 3A	bY•±Ëg•:
2	0000 ... 0010	B3 9B 0D 12 1F C5 A9 7C	³••••Å©|
3	0000 ... 0011	84 19 9D C6 B0 F5 AD 75	•••Æ°õ•u
4	0000 ... 0100	D4 E6 90 8D 8F 77 EA 07	Ôæ•••wê•
	...		
38 326 038 678 974 151	1000 ... 0111	42 65 72 6B 65 6C 65 79	Berkeley
	...		
72 057 594 037 927 935	1111 ... 1111	FB 90 3D D5 99 A3 27 3D	û•=Õ•£'=

Table 5.1: Decrypting a message encrypted with the Data Encryption Standard by trying all possible keys. Each DES key is 56 bits long; there are roughly 72 quadrillion keys. Characters that are not printable are displayed with a bullet (•). Notice that when correct key is found, all of the decrypted characters are printable. In this case was found roughly half way through because the key starts 1000. The same approach can be used with AES, except that there are $2^{128} = 340\,282\,366\,920\,938\,463\,463\,374\,607\,431\,768\,211\,456$ possible keys in its weakest implementation.

the two machines, and use the full 20 minutes to crack each. Of course, 20 minutes to crack a message was the *worst case*; on average a message would be cracked after half of the rotor positions had been tried. It was also necessary to detect when the correct rotor position was found. The Germans made this easier by their tendency to begin their encrypted messages with the same sequence of characters.

When the U.S. Data Encryption Standard was adopted by the National Bureau of Standards (NBS) in 1977, Hellman wrote a letter to NBS arguing that the reduction of the DES keysize from 64 bits to 56 bits suggested that it was done "to intentionally reduce the cost of exhaustive key search by a factor of 256."[57] In a follow-up article, Diffie and Hellman hypothesized that it should be possible to create a special-purpose DES-cracking microchip that could try a million keys each second. With a million such chips, it would be possible to try all 2^{56} keys in a day. They estimated the cost of constructing such a machine at $20 million in 1977 dollars; assuming a five-year life of the machine and a daily operating cost of $10,000, the average cost of cracking a DES-encrypted message in 1977 would be just $5000, including the cost of developing the machine.[58] With expected improvements in microelectronics, the Stanford professors estimated that the cost of their hypothetical DES-cracking machine to just $200,000 by 1987. In fact, it actually took twenty years. In 1998 the Electronic Frontier Foundation (EFF) announced that it had spent $250,000 and constructed the fabled DES Cracker. The EFF machine tried 90 billion 56-bit DES key every second, and cracked its first challenge message after only 56 hours of work.[59] The project is widely credited with putting the last nail into the coffin of weak symmetric encryption schemes.

When cracking symmetric encryption systems with a brute force attack, each additional bit of key length doubles the difficulty of the attack, because each additional bit doubles the number of keys that need to be searched. With 4 bits, there are 16 keys to search; with 8 bits there are 256, and so on. For a while, the U.S. Government's proposed replacement for DES was the so-called "Clipper" chip, which supported an 80-bit key, making it 2^{24} or roughly 16 million times harder to crack—except that the each Clipper chip was gimmicked so that the government *didn't need* to perform such an attack to decrypt a message encrypted with Clipper. That's because the Clipper implemented the government's "Escrowed Encryption Standard" (FIPS-185), which meant that every Clipper had its own secret decryption key that could be used to decrypt any message that the chip encrypted, and the government kept copies of these keys so that messages could be decrypted for legal process or in the event of a national security emergency. To prevent companies from creating software-only Clipper chips that didn't implement key escrow, the government declared that the encryption algorithm used by the chip had to be kept secret in the interest of national security.

As might be expected, Clipper chip was a commercial failure.

When the National Institute for Standards and Technology initiated its efforts to create a replacement algorithm for the Data Encryption Standard in the late 1990s, it committed itself to an open, unclassified project. NIST invited submissions for

[57] Blanchette, *Burdens of Proof: Cryptographic Culture and Evidence Law in the Age of Electronic Documents* (2012).

[58] Diffie and Hellman, "Special Feature Exhaustive Cryptanalysis of the NBS Data Encryption Standard" (1977).

[59] Electronic Frontier Foundation, *Cracking DES* (1998).

the new algorithm, held two academic conferences to discuss the submissions, and ultimately adopted an algorithm invented outside the United States by a pair of Belgian cryptographers, Vincent Rijmen and Joan Daemen. The algorithm, originally named Rijndael, is faster than DES and supports key sizes of 128, 192 and 256 bits. It was adopted by the U.S. Government as the Advanced Encryption Standard in 2001.

For many years after it was adopted, AES-128 was the preferred use of AES because it ran significantly faster than the more secure AES-256. That extra security is in fact the reason that AES-256 was slower. The design of AES is based on a function that is repeated a certain number of "rounds" for every block of data that the algorithm encrypts. AES-128 has 10 rounds, AES-256 has 14.[60] Today those differences are less significant than they were in 2001, as computers are faster and many microprocessors now contain hardware support to make AES run faster still. In most modern computers, encrypting with AES-128 is essentially free. For example, the Apple iPhone contains a chip that automatically encrypts data with AES when it is written from the CPU out to phone's flash memory, and automatically decrypts the data when it is read back in.

However, absent quantum computing, the differences between AES-128 and AES-256 are inconsequential for most users. That's because 2^{128} is a really big number: in a world without quantum computers, a message encrypted with a 128-bit key will *never* be cracked using a brute-force, key search attack.

5.3.3 Cracking AES-128 with Grover's algorithm

Grover's algorithm makes it possible to use a quantum computer to guess the right key with fewer steps than it would take to try all possible keys. To understand why AES-128 is vulnerable to a quantum computer running Grover's algorithm but AES-256 is not, it is necessary to understand more about how Grover's algorithm works in practice.

Although Grover's discovery is frequently described as an algorithm for speeding up "database search," this gives a misleading impression as to what the algorithm actually does. The "database" is not the kind of database that most people are familiar with: it doesn't actually store data. Instead, the database is a database of guesses and whether or not each guess is correct.

In Table 5.3.3, we have recast the problem of cracking an encrypted message into a database search problem that could then be searched using Grover's algorithm. To perform a brute force search for the correct key, just start at the top and examine each row until the database value is a `1`. In this example, a little more than half of the rows need to be examined. If you have a computer that can examine 90 billion rows a second—on par with the speed of the EFF DES Cracker—then you will find the answer in roughly five days.

A key search attack is possible because 2^{56} is not such a fantastically large number after all—that's the point that Hellman making in his letter the NBS when he urged that 56 bits was just too small. If NBS had gone with a 64-bit key length, then an average search time of 20 hours would become 1 280 days. That's better, but

[60] AES-256 may in fact be more than 2^{128} times stronger than AES-128, as AES-256 has 14 internal "rounds" of comptuation, while AES-128 has only 10. If there is an algorithmic weakness in the underlying AES algorithm, that weakness should be easier to exploit if there are fewer rounds.

Row	Row number in binary	Database Value
0	0000 ... 0000	0
1	0000 ... 0001	0
2	0000 ... 0010	0
3	0000 ... 0011	0
4	0000 ... 0100	0
	...	
38 326 038 678 974 151	1000 ... 0111	1
	...	
72 057 594 037 927 935	1111 ... 1111	0

Table 5.2: To use Grover's algorithm to crack an encryption key, Table 5.1 is recast as a database search problem, where one row has the value of 1 stored and all of the other rows have the value of 0. In this example the keys are 56-bit DES keys. If this table instead used 128-bit AES keys, the last row would be number 340 282 366 920 938 463 463 374 607 431 768 211 455 ($2^{128} - 1$).

it's still not good enough for government work, which requires that national security secrets be declassified after 50 years,[61] unless they contain names of confidential intelligence sources, contain information on weapons of mass destruction technology, would "reveal information that would impair U.S. cryptologic systems or activities," or meet a few other specified requirements.[62] Clearly for U.S. government use, an encryption algorithm that might be crackable at any point in the foreseeable future due to the likely advance of computer technology is not acceptable.

As we have stated above, AES-128 doesn't have this problem, because 2^{128} is fantastically larger than 2^{56}—unless the attacker has a functioning quantum computer that's large enough to compute AES-128.

Cracking AES-128 with Grover's algorithm is surprisingly straightforward. First, it is necessary to construct an implementation of AES-128 on a quantum computer with at least 129 qubits, such that when the first 128 qubits have the correct decryption key, the 129^{th} qubit has the value of 1. Additional qubits are required to implement various details of Grover's algorithm and to properly implement AES-128 (we won't go into the details here).

AES-128 has 10 rounds, which means there is an inner algorithm that is repeated in a loop 10 times. Quantum computers don't have this kind of loop, so it is necessary to *unroll* the rounds, meaning that the circuits for the inner AES function need to be repeated 10 times. Additional circuitry is required to detect when the correct decryption key has been found.

It's relatively straightforward to imagine how the AES-128 circuit might be run on the kinds of superconducting quantum computers being developed by IBM and Google. On these computers, the qubits are "artificial atoms" made up of superconducting circuits operating at close to absolute zero, while the quantum gates and circuits and implemented by precisely timed and aimed pulses of radio waves. The speed of the quantum computation is determined by how quickly the quantum computer can cycle through a specific combination of radio waves that it sends into

[61] For an explanation of the origin of this phrase and its corruption, see Lerman, *Good Enough for Government Work: The public Reputation Crisis in America (And What We Can Do to Fix It)* (2019).

[62] Obama, *Executive Order 13526: Classified National Security Information* (2009).

the artificial atoms. When the computation is finished, the qubits are measured with other radio wave pulses.

To run Grover's algorithm, each of the unknown bits (here, the 128-bit AES key) starts off as a superposition of `0` and `1`. The algorithm is then cycled $\sqrt{2^N}$ times, where N is the number of unknown bits. At the end of these cycles, the unknown bits are measured, and they are overwhelmingly likely to have the answer to the problem. Superposition must be maintained for the entire time: if it lost, the computation is ruined.

It turns out that $\sqrt{2^N} = 2^{N/2}$. So when cracking AES-128, only 2^{64} iterations are required, rather than 2^{128}. Because 2^{64} is not a fantastically large number, the mere existence of Grover's algorithm and the possible future existence of large-enough quantum computers was enough for cryptography experts to recommend discontinuing the use of AES-128 when these results became generally understood. However, AES-256 is still fine, because even with Grover's algorithm reducing the security parameter from 2^{256} to 2^{128}, that's okay because 2^{128} is a fantastically large number. All of this was clear from the theory, without the need to create an actual working quantum implementation of AES to actually try out Grover's algorithm.

In 2016, quantum computing theoreticians in Germany and the U.S. carried out the hard work of actually building "working" quantum circuits of AES-128, AES-192 and AES-256—at least, in theory. They found that implementing cracking a single AES-128 encryption key with Grover's algorithm require *at most* 2 953 qubits and on order of 2^{86} gates. For AES-256 the estimate was 6 681 qubits and 2^{151} gates.

"One of our main findings is that the number of logical qubits required to implement a Grover attack on AES is relatively low, namely between around 3 000 and 7 000 logical qubits. However, due to the large circuit depth of unrolling the entire Grover iteration, it seems challenging to implement this algorithm on an actual physical quantum computer, even if the gates are not error corrected," the authors write. The authors conclude "It seems prudent to move away from 128-bit keys when expecting the availability of at least a moderate size quantum computer."

The word "prudent" requires additional explanation, as even a work factor of 2^{86} is likely to be beyond the limits of any human technology for the foreseeable future. For example, a quantum computer that could sequence quantum gates every *femtosecond* (that is, 10^{15} times per second) would still require 2 451 *years* to crack a single AES-128 key using the implementation described in the 2016 publication. And a femtosecond clock would be a big deal—it would be 250 times faster than the clock speed of today's 4GHz microprocessors. Chemical reactions take place at the femtosecond scale; the time is so short that light only travels 300 nanometers.

Of course, given a cluster of 1 024 quantum computers, each running with a femtosecond clock, each one attempting to crack AES-128 with a different 10-bit prefix, an AES-128 message could be cracked in less than a year. So if mass-produced femtosecond quantum computers with a thousand qubits that can compute a single calculation error-free for a year is a risk that you consider relevant, then you should not be using AES-128 to protect your data!

But remember—the 2016 article describes an *upper bound*: it might be possible to create AES-cracking quantum computing circuits that require fewer gates. In fact, two 2019 efforts[63] lowered the upper bound on the work factor to crack AES-

[63] Jaques et al., *Implementing Grover oracles for quantum key search on AES and LowMC* (2019);

128 to 2^{81} and 2^{79} respectively by developing better quantum gate implementations for the AES oracle (the quantum code that determines when the correct key has been guessed). It has long been the case that hand-tuning algorithms to squeeze out the last few cycles of performance has been something of a parlor game among computer scientists.[64] So instead of looking for upper bounds, it might be more productive to look for theoretical lower bounds.

The absolute lowest bound for a circuit that could crack AES using Grover's algorithm would be a circuit that executed a single gate over a large number of qubits: such a perfect implementation would require a minimum of 2^{64} cycles to crack AES-128, and 2^{128} to crack AES-256. We do not think that such a circuit is possible. However, this "perfect" quantum AES implementation would be able crack AES-128 in 5.12 hours with our fictional quantum computer with a femtosecond clock; even this perfect implementation would require 10 782 897 *billion years* to crack AES-256.

To push the hypothetical even explore, there's no fundamental reason why we should limit our fictional quantum computer to a femtosecond clock. What if we had a smaller, more compact quantum computer that could fit in a nanosphere—perhaps two thousand packed atoms in blob just 10 nm across. The maximum cycle time of this computer would be roughly $\frac{1}{30}$ of a femtosecond, the time it takes light to move from one side of the sphere to the other. With this computer and the (fictional) perfect Grover AES circuit, you could crack AES-128 in just 10 minutes, but it would still take 360 billion years to crack AES-256. Here parallelism finally begins to help: with a billion of these computers, you could crack an AES-256 in at most 3.6 years. Of course, if you have the kind of technology that can make and control a billion of these computers, there are probably far more productive things you would be able to do than go after AES-256 keys from the 2020s.

So to summarize, although it's conceivable that AES-128 might one day fall to a futuristic quantum computer, there is no conceivable technology that could crack AES-256. What's more, AES-128 is sufficiently close to the boundary of what a quantum computer might be able to crack over the next twenty of thirty years that it is indeed "prudent" to stop using AES-128 in favor of AES-256. In part, this is because the cost increase of using AES-256 instead of AES-128 is quite minor: on a 2018 Apple "Mac Mini" computer, encrypting a 7 GiB file took 7.1 s with AES-128 running in "cipher block chaining" mode; with AES-256 it took 9.1 s. For the vast majority of applications this 28% increase in encryption time is simply not significant.

But remember—all of the analysis above assumes that AES-256 is a perfect symmetric encryption algorithm. However, there might be underlying vulnerabilities that make it possible to crack with significantly less work than a full brute-force attack. To date no such attacks have been published that offer speedup greater than Grover's algorithm,[65] but there's always tomorrow. Certainly, if computer

Langenberg, Pham, and Steinwandt, *Reducing the Cost of Implementing AES as a Quantum Circuit* (2019).

[64] For example, in 2010, a group of researchers at the Naval Postgraduate School that included one of us published a high-speed implementation of AES for the Sony PlayStation.Dinolt et al., *Parallelizing SHA-256, SHA-1 MD5 and AES on the Cell Broadband Engine* (2010)

[65] There is one classical attack against AES-256 that lowers the work factor from 2^{256} to $2^{254.4}$; Grover's quantum algorithm lowers the work factor to 2^{128}.

scientists discover that P=NP (the **sidebar "??"**), then attacking AES-256 could become the stuff of high school science fairs shortly thereafter.

5.3.4 Grover's algorithm today

The impact of the square-root speedup offered by Grover's algorithm has been systematically misrepresented in the popular press over the past two decades. Recall that although Grover's algorithm speeds up *search*, it is not the kind of search that we do with Google looking for a web page or using an accounting system when we are looking for a specific transaction. Those kinds of searches involve the computer scanning through a database and looking for a matching record, as we discuss in Section 3.5.1 (p. 76). Although Grover's algorithm *could* be applied to such a search, it would require storing the entire database in some kind of quantum storage—a system that has only been well-specified in works of science fiction—playing the entire database through the quantum circuit, a process that would eliminate any speedup provided by Grover's algorithm in the first place.

To date, scientists have accomplished only limited demonstrations of Grover's algorithm. Beit, a quantum software company with a lab in Kraków, Poland, released two unpublished papers in 2020 reporting state of the science accomplishments in applications of Grover's search. A September 2020 paper from the group demonstrated a Grover implementation in IBM hardware, where the team performed an unstructured search among a list with just 16 elements. The goal of such a search is to identify one element in the list successfully, but the system was able to do so on average only 18—24 percent of the time.[66] A subsequent study employed Honeywell's 6-qubit Model H0 ion trap, which is commercially available. In June 2020, Honeywell hailed the device as the world's most powerful quantum computer, claiming that it has a quantum volume of 64.[67]The Beit team, using Honeywell's API, tested Grover's search in 4, 5, and 6-qubit implementations. Respectively, the team could select the right result 66 percent of the time with a 4-qubit circuit (selecting from a list with 16 elements), 25 percent of the time with a 5-qubit circuit (using a list with 32 elements), and just 6 percent of the time using all 6 qubits in a circuit (using a list with 64 elements).[68]

Some articles in the popular press incorrectly describe quantum computers as machines that use superposition to simultaneously consider all possible answers and select the one that is correct. Such machines do exist in the computer science literature, but they are called "non-deterministic Turing machines" (see Section 3.5.3 (p. 80)). And while such machines do exist *in theory*, they do not exist *in practice*: the conservation of mass and energy makes them impossible to build in this universe.[69]

[66]Gwinner et al., *Benchmarking 16-element quantum search algorithms on IBM quantum processors* (2020).

[67]*Quantum volume* (QV) is a metric that IBM created that measures the square of the number of quantum circuits that a quantum computer can implement. According to IBM, QV combines "many aspects of device performance," including "gate errors, measurement errors, the quality of the circuit compiler, and spectator errors."Jurcevic et al., *Demonstration of quantum volume 64 on a superconducting quantum computing system* (2020)

[68]Hlembotskyi et al., *Efficient unstructured search implementation on current ion-trap quantum processors* (2020).

[69]Such machines are not even possible if you subscribe to the many-worlds interpretation of quantum physics: it may be that a computer facing an NP-hard problem with a quantum-mechanical

The Limits of Quantum Computation

"The manipulation and transmission of information is today carried out by physical machines (computers, routers, scanners, etc.), in which the embodiment and transformations of this information can be described using the language of classical mechanics," wrote David P. DiVincenzo, then a theoretical physicist at the IBM T.J. Watson Research Center, in 2000.[a] "But the final physical theory of the world is not Newtonian mechanics, and there is no reason to suppose that machines following the laws of quantum mechanics should have the same computational power as classical machines; indeed, since Newtonian mechanics emerges as a special limit of quantum mechanics, quantum machines can only have greater computational power than classical ones."

"So, how much is gained by computing with quantum physics over computing with classical physics? We do not seem to be near to a final answer to this question, which is natural since even the ultimate computing power of classical machines remains unknown."

For example, DiVincenzo wrote, we know that quantum computing does not speed up some problems at all, while some are sped up "moderately" (in the example of Grover's algorithm), and others are "apparently sped up exponentially" (Shor's algorithm).

DiVincenzo notes that, on purely theoretical grounds, quantum computing also could result in a "quadratic reduction" in the amount of data required to be transmitted across a link between two parties to complete certain mathematical protocols. But such a reduction requires the data is transmitted as quantum states—over a quantum network—rather than as classical states. "The list of these tasks that have been considered in the light of quantum capabilities, and for which some advantage has been found in using quantum tools, is fairly long and diverse: it includes secret key distribution, multiparty function evaluation as in appointment scheduling, secret sharing, and game playing."

[a] DiVincenzo, "The Physical Implementation of Quantum Computation" (2000).

Quantum Algorithm Zoo

Stephen Jordan, a physicist at Microsoft Research who works on quantum computing, maintains a database of quantum algorithms—the Quantum Algorithm Zoo. Jordan categorizes today's quantum algorithms into four types:[a]

1. **Algebraic and number theoretic algorithms,** which use properties of quantum computers to solve number theory problems. An example is Shor's algorithm for factoring.

2. **Oracular algorithms,** which depend upon an *oracle* that can provide an answer to a question. An example is Grover's algorithm for speeding up search.

3. **Approximation and simulation algorithms**, such as would be used to simulate the process of nitrogen fixation as discussed in Nitrogen Fixation, Without Simulation.

4. **Optimization, numerics, and machine learning algorithms,** which could be used for improving systems based on so-called neural networks, including speech, vision, and machine translation.

[a] You can find the list of algorithms at Jordan's website, http://quantumalgorithmzoo.org/, which is based on his May 2008 MIT PhD Thesis Jordan, "Quantum computation beyond the circuit model" (2008).

Quantum computers use superposition to simultaneously consider a multitude of solutions, which *does* allow them to compute the answers to *some kinds of problems* faster than computers that are not based on superposition and entanglement. But they don't do this by coming up with the single, best answer to those problems. Instead, modern quantum computers attempt to solve a single problem many times over and come up with a distribution of possible answers, with more probable answers coming up more often and the less probable answers coming up less often. The trick to programming the machines is to set up the computer so that the answers that are significantly more probable and that incorrect answers are significantly less probable. This is done, ultimately, with constructive and destructive interference at the quantum level, in the machine's Schrödinger wave equation.

Another source of confusion might be that quantum computers can solve particular kinds of problems in polynomial time that are thought to be harder than the complexity class known as P (polynomial). The key example here is factoring. Because NP (nondeterministic polynomial) is the class that most people think is harder than P, and NP is the class solved by non-deterministic Turing machines, some people jump to the conclusion that quantum computers can solve NP-hard problems.

There are several problems with this line of thinking. First, just because mathematicians haven't found an algorithm that can factor in polynomial time doesn't mean that such an algorithm doesn't exist: it wasn't until 2002 that mathematicians had an algorithm for primality testing that ran in polynomial time. So factoring might be in P, and we just haven't found the algorithm yet. Or, more likely, factoring might be harder than P and still not in NP. Or, it might be that $P = NP$, which would make factoring in both P *and* NP, because they would be the same. As we discussed in Section 3.5.6 (p. 86), computer scientists use the complexity class called BQP to describe the class of decision problems solvable by a quantum computer in polynomial time. Just as we don't know if P is equal to NP, we don't know if BQP is the same or different from P or NP. This can be written as:

$$P \stackrel{?}{=} BQP \stackrel{?}{=} NP \tag{7}$$

For further discussion of this topic, we recommend Aaronson's article "The Limits of Quantum"[70].

Similar to the situation with the NP-hard and NP-complete problems, there is no proof that quantum computers would *definitely* be faster at solving these problems than classical computers. Such a mathematical proof would put theoreticians well on their way to solving the whole $P \neq NP$ conjecture, so it is either right around the corner or it is a long way off. It is simply the case that scientists have discovered efficient algorithms for solving these problems on quantum computers, and no such corresponding algorithms have been discovered for classical computers.

---

random number generator splits the universe 2^N times and that in one of those universes a computer immediately finds the correct answer. The problem is that in all of the *other* $2^N - 1$ universes the computers all discover that their answer is incorrect, and there is no inter-universe network to allows the computer that guessed correctly to inform its clones of the correct choice

[70] Aaronson, "The Limits of Quantum" (2008).

5.4 Conclusion

Whereas the electromechanical and early electronic computers of the 1940s were transformative, allowing the United Kingdom to crack the German Enigma code and the United States to create the hydrogen bomb, the main use of quantum computers today in 2021 is by researchers who are developing better quantum computers, better quantum algorithms, and students who are learning about quantum computers. The main output of today's quantum computers is not military intelligence and might, but papers published in prestigious journals.

Nevertheless, it would be a mistake to dismiss this research as quantum navel gazing. Unlike the limits that have impacted Silicon Valley's efforts to make increasingly faster electronic computers, we may be a far way off from hitting any fundamental limit or law of nature that will prevent researchers from making larger and faster quantum computers—provided that governments and industry continue to invest the necessary capital.[71]

This may be why some governments continue to pour money into quantum computing. Although promoters speak about the benefits in terms of simulation and optimization, they are surely also driven by that darker goal of being able to crack today's encryption schemes used to secure the vast majority of information transmitted over the Internet and through the air. And because information transmitted in secret today might be useful if decrypted many decades from today, *the mere possibility* that powerful, reliable quantum computers might exist several decades in the future is a powerful influencer today.

Today's quantum computers are not powerful enough to break the world's cryptography algorithms (or do anything else), but each year they improve, as quantum computing engineers become more adept at precisely controlling fundamental quantum processes. For this reason alone, our society should seek to rapidly transition from today's quantum-vulnerable encryption algorithms like RSA and AES-128 to the next generation of post-quantum encryption algorithms. If our understanding of quantum mechanics is correct, it is only a matter of time until the machines are sufficiently powerful.

We are still at the beginning of quantum computing, and very basic questions of technology and architecture still have yet to be worked out. The next chapter canvasses the research groups that are wrestling with different physical substrates for representing quantum information, different ways of organizing those physics packages into computing platforms, and different languages that programmers can use to express quantum algorithms. Much research in quantum computing is so preliminary and theoretical that an idea can have a major impact years before it's been reduced to practice and demonstrated. What's concerning is that there field hasn't had a mind-blowing discovery since the breakthroughs of Shor and Grover in the mid-1990s.

[71] If it turns out that we can never make machines that work at large scale, then it is likely that there is something fundamentally wrong about our understanding of quantum physics. Many advocates say that this alone is worth the study of quantum computers. And while some funding agencies might disagree, the amount of money spent on quantum computing to date appears to be significantly less than the $10-$20 billion that the U.S. high energy physics community proposed spending on the Superconducting Super Collider in the 1990s, or even the $4.75 billion that Europe spent on the Large Hadron Collider between 1994 and 2014.

6

(NEAR FINAL) Quantum Computing Today

At the 25th Solvay Conference on Physics in 2011, John Preskill asked a question about quantum computing for which we still have no answer:

> Is controlling large-scale quantum systems merely **really, really hard**, or is it **ridiculously hard**?[1]

Preskill, who (somewhat ironically) is the Richard P. Feynman Professor of Theoretical Physics at the California Institute of Technology, was asking if building ever larger quantum computers of the kind we envisioned in the last chapter is merely matter of better engineering, or if there are fundamental limits about the nature of physics, computation, and reality itself that will get in the way. That is, are we likely to have working quantum computers "going beyond what can be achieved with ordinary digital computers"—what Preskill called "quantum supremacy"—after "a few decades of very hard work"? Or are we likely to come up short after even centuries of effort?

Preskill didn't have an answer, but he was enthusiastic about the quest: even if efforts to build a working large-scale quantum computer failed, humanity would still learn important fundamental truths about the fundamental nature of the universe.

In the last chapter we discussed the first three great applications that have been envisioned for quantum computers: simulating quantum mechanical systems (Feynman), factoring large numbers (Shor), and speeding the search for solutions to any mathematical problem for which it is possible to construct a quantum oracle (Grover). All of these applications were developed by theoreticians working with nothing more than the metaphorical pencil and paper, and the ability to discuss ideas with their collaborators. Actually realizing these applications requires something more: a large-scale, reliable quantum computer.

Companies and research labs are racing to answer Preskill's question. Some are large, established technology powerhouses, like Google, IBM, and Microsoft. Others are well-funded emerging players, such as ColdQuanta, D-Wave and Rigetti. Most are building actual physics packages, with super-cooled superconductors and parts that are literally gold-plated. In most but not all cases, the results of these quantum computers can be reliably simulated using clusters of conventional computers. However, in a few cases, machines have been constructed that can solve problems beyond the capacity of today's digital computers—even when millions of those computers are networked together.

[1] Preskill, "Quantum computing and the entanglement frontier" (2012), emphasis in the original.

"I proposed the term 'quantum supremacy' to describe the point where quantum computers can do things that classical computers can't, regardless of whether those tasks are useful," Preskill wrote in 2019.[2] "With that new term, I wanted to emphasize that this is a privileged time in the history of our planet, when information technologies based on principles of quantum physics are ascendant."

After gaining traction, Preskill's term *quantum supremacy* has been somewhat supplanted by the term *quantum advantage*. Some researchers prefer this term, because rightfully implies that quantum computers will be working alongside classical computers to literally confer advantage, just as a modern computer might offload some computations to a graphics processing unit (GPU).

Quantum computers have not scaled up at the same rate as their electronic computing predecessors. We have yet to experience a quantum form of Moore's Law (see Section 3.5 (p. 70)), in part because quantum engineers have not found a suitable quantum mechanism to the digital discipline that allows creating ever-larger digital circuits without ever-increasing amounts of systemic error (see Section 3.3 (p. 63)). Although quantum error correction schemes exist, it is unclear if they can scale to allow for meaningfully complex computations, because these schemes themselves require higher quality qubits operational for longer timescales than are currently possible. Without resolving this issue, we will still likely be able to create analog *quantum simulators* for solving questions in physics, chemistry and biology, but the goal of using quantum computers to crack codes may remain forever out of reach. Nevertheless, researchers at both Google and the University of Science and Technology of China created quantum computing systems that clearly meet Preskill's requirement for quantum supremacy.

In this first section of this chapter we will describe in abstract the basics of how the current generation of quantum computers work. Next, in Section 6.2.2 (p. 174) we discuss the hardware efforts of today and the near future. We discuss what will need to be overcome in Section 6.3 (p. 178). Finally we conclude this chapter with Section 6.4 (p. 187).

6.1 How to Build A Quantum Computer

In Chapter 4 we introduced the basic idea of the Fredkin and Toffoli gates, and in chaprefchapter-quantum-computing-applications we discussed the two quantum algorithms that started serious money flowing into the creation of actual quantum computers. In this chapter we'll briefly look at a simple quantum circuit and discuss the barriers to creating quantum circuits of the size necessary to accomplish the computational goals set out in the previous chapter.

In a now classic article, David P. DiVincenzo, then at the IBM T.J. Watson Research Center, formulated five requirements for quantum computing:[3]

1. There needed to be something that could "hold data and perform computation." For simplicity, scientists have focused systems that have two precise states, which we call qubits. Whereas a classical bit can only have two values, `0` and `1`, quantum bits are a superposition of these two states. This superposition is typically written using Paul Dirac's Bra-ket notation as $a\,|0\rangle + b\,|1\rangle$,

[2] Preskill, "Why I called it 'Quantum Supremacy'" (2019).
[3] Divincenzo, "Topics in Quantum Computers" (1997).

where a and b are taken to be complex numbers such that $|a|^2+|b|^2 = 1$ during the course of the computation, but which become either `0` or `1` when they are measured at the end of the computation.[4] This measurement corresponds to "opening the box" in Schrödinger's famous thought experiment (see p. 380).[5]

2. The ability to initialize the qubits to a known "fiducial starting quantum state." This requirement is akin to resetting all of the bits in a classical computer to `0`. In his 1997 article, DiVincenzo wrote "I do not think that this "initial state preparation" requirement will be the most difficult one to achieve for quantum computation. Three years later in his follow-up article, DiVincenzo was less sanguine: "The problem of continuous initialization does not have to be solved very soon; still, experimentalists should be aware that the speed with which a qubit can be zeroed will eventually be a very important issue."[6]

3. The ability to interact with each other using some form of *quantum gate*. This is where the Feynman and Toffoli gates from Section 4.5 (p. 113) become relevant. Each gate mixes the quantum state of two, three or more qubits together to perform some sort of simple computation. The physical construction of the quantum computer determines which qubits can be connected together. Ideally, the quantum gates are *universal*, so that they can be used to describe any computation (provided that you have sufficient qubits and time.)

   As we will see in Chapter 3, this design makes the construction and programming of quantum computers fundamentally different from the way we have built classical computers. In classical computers the bits represented by the presence or absence of an electric charge move through the electronic circuits, which are fixed at the time the computer is manufactured. In a quantum computer, it is the qubits that are fixed when the computer is manufactured, and the system is programmed by playing a sequence of circuits through the qubits to perform the desired computation. Thus, the computing speed of the quantum computer fundamentally depends on the number of qubits that it has and the speed at which the circuits can be constructed; this speed is exactly analogous to the clock speed of a modern microprocessor.[7]

4. The ability to keep the qubits in their *coherent*, *entangled* state for an extended period of time. This period time is not measured in seconds, but in terms of how many gates can be played through the qubits. In article, DiVincenzo suggested that it would be necessary to execute between a thousand and ten thousand gates in order to be able to perform meaningful computations with sufficient quantum error correction.[8]

   An added complication is how error propagates as the quantum computer begins to lose its coherency: if errors are correlated rather than randomly

[4] With two qubits, the systems state is described by a four dimensional vector: $a\,|00\rangle + b\,|01\rangle + c\,|10\rangle + d\,|11\rangle$.

[5] Qubits must be physically isolated from the universe such that there is no external energy that would bias the qubit towards being `0` or `1` on measurement. This is why qubits do not need to be isolated from gravity: both the $|0\rangle$ and the $|1\rangle$ states have the same mass.

[6] DiVincenzo, "The Physical Implementation of Quantum Computation" (2000).

[7] In his 1997 and 2000 articles, the requirement of "a 'universal' set of quantum gates" is presented as the fourth DiVincenzo criteria.

[8] Long decoherence time was originally presented as the third DiVincenzo criteria.

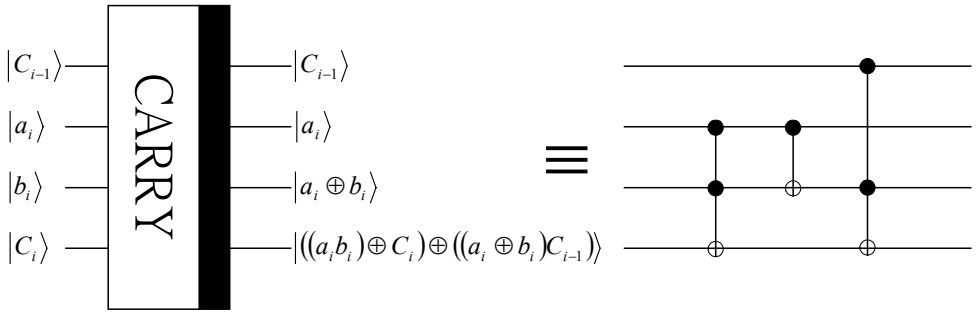


Figure 6.1: A 2-bit quantum carry gate, from Cheng and Tseng, *Quantum Plain and Carry Look-Ahead Adders* (2002), used with permission. The gate reversibly determines whether adding two bits produces a carry operation.

scattered through the system, it may adversely impact the ability to perform meaningful quantum error correction.

5. The ability to measure each qubit at the end of the computation.

We show what this looks like in figures 6.1 through 6.3. This adder, which would be a small part of much larger quantum circuit, takes two numbers between `0` and `15` and adds them together. The key difference between this adder and the 4-bit adder that you might find in a classical computer (such as Figure 3.5) is that this adder is reversible. The adder in Figure 6.3 uses 13 qubits and requires 30 gates. The design in Figure 6.3 also requires 30 cycles to operate because none of the gates execute at the same time. However, this algorithm can be optimized (Figure 6.4) by having many of the gates acting simultaneously. This optimized algorithm can run in just 7 cycles.

By *reversible*, we mean that this adder needs to be able to run in reverse. That is, it needs to be able to take the result of the addition, a single number between `0` and `15` and provide the two specific input numbers that was used to create it. This may seem like a magic trick! If we told you that the number `9` is the sum of two numbers and asked you what they were, you would be unable to tell us: the answer might be `0` and `9`, or `1` and `8`, or `2` and `7`, and so on. As a result, the quantum 4-bit adder needs more than 4 bits of output: besides the four-bit sum, it also preserves half of the input bits. The adder also has an additional input bit called z and an output bit that combines z with the carry bit. Such additional qubit are sometimes called an *ancillary* or *ancilla qubits*; designing efficient quantum circuits that use a minimum number of ancilla qubits is one of the current challenges of quantum computer programming, due to the small number of qubits and the short decoherence times. Programming quantum computers at the circuit level in this manner is exactly analogous to the way that computing's pioneers in the 1940s and 1950s modified the hardware of their computers to add new instructions and programmed the machines using machine code.

In summary, In order to compute at the quantum level , one must be able to generate, maintain, manipulate, and measure quantum states. Thus, quantum sensors are a precursor technology for quantum computing, and this is why this book presented quantum sensing first. In many ways, today's quantum computers are really just large-scale quantum sensor arrays.

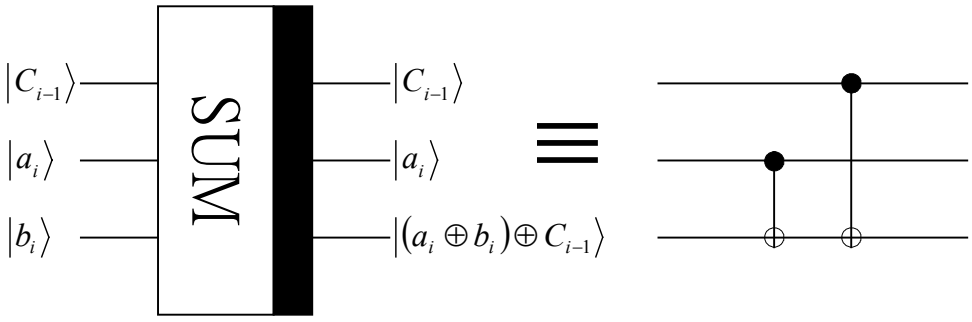


Figure 6.2: A 2-bit quantum sum gate, from Cheng and Tseng, *Quantum Plain and Carry Look-Ahead Adders* (2002), used with permission. The gate reversibly determines whether adding two bits produces a sum.

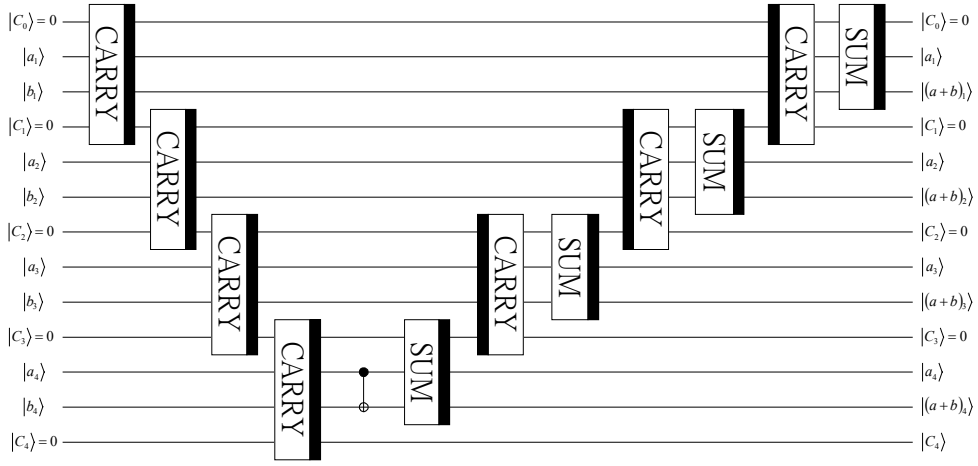


Figure 6.3: A 4-bit quantum adder circuit, from Cheng and Tseng, *Quantum Plain and Carry Look-Ahead Adders* (2002), used with permission. The inputs on the left are the nibbles $a_4a_3a_2a_1$ and $b_4b_3b_2b_1$ and the carry bit C_0. The output bits on the right are the sum $(a+b)_4(a+b)_3(a+b)2(a+b)_1$, the input value $a_4a_3a_2a_1$ and the carry bit C_4.Time flows from left to right. Compare this with Figure 3.5, the 4-bit classical adder.

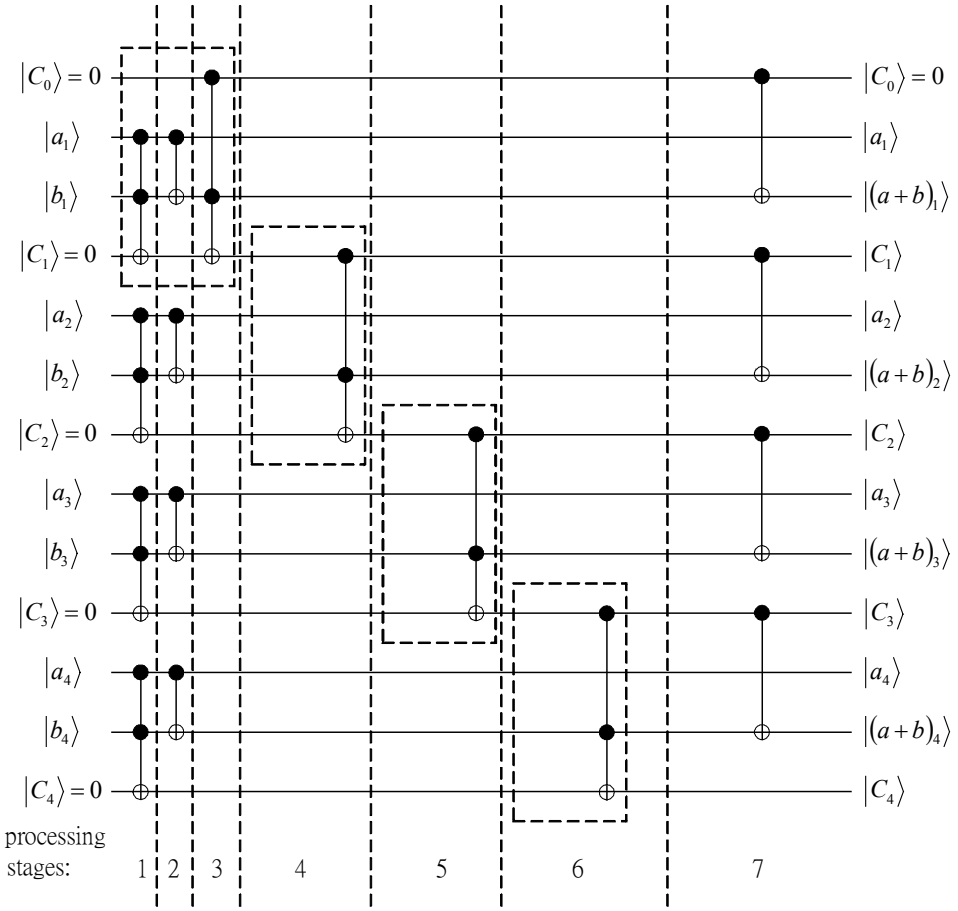


Figure 6.4: The 4-bit quantum adder from Figure 6.3, optimized to execute in fewer cycles. From Cheng and Tseng, *Quantum Plain and Carry Look-Ahead Adders* (2002), used with permission.

6.2 The Quantum Computer Landscape

Preskill's 2019 article argues that the question he posed in 2012 is all but answered, and that we have moved from the era of quantum computing's first steps and into the era of noisy intermediate scale quantum devices—NISQ—another term that he coined.

Unlike classical computers, which are nearly all based on silicon semiconductors, today's NISQ computers are not dominated by a single physical substrate. Instead, we are in a period of experimentation—one that might stretch out for decades. Today's quantum innovators are experimenting with different approaches to creating and managing the quantum states necessary for computation. To date, none has realized the scale required for solving meaningful problems outside the world of experimental physics. The different media are promising in different ways, with some offering longer coherence times and greater interconnection, while others lack the need for specialized cooling or have engineering characteristics that might make a large-scale possible. We don't know which will be the winner.

6.2.1 Comparing Quantum Media

Understanding the quantum computing landscape is challenging because virtually every device that's been produced has different characteristics and capabilities. Some competitors claim to have relatively large-scale qubit devices, yet these may not be as interconnected as smaller devices, and large devices' size and architecture may be noisier and less stable than smaller devices. One cannot evaluate today's quantum computers simply by comparing the number of qubits they possess.

Adding to the difficulty, companies claims' on quantum computers may be strategically shaped to capture para-hardware markets, such as software and services. Companies have created vocabularies and software frameworks that are explicitly helpful to them and their business model. Even when claimed to be neutral and universal, these vocabularies and frameworks cannot help but seek to establish a software ecosystem that is favorable to their creators.

Competitors in the field all seek the *logical qubit*, a qubit that can overcome the problems of gate errors, environmental noise, and decoherence long enough to perform quantum operations. Understandably, competitors have chosen different paths for the construction of a stable quantum computer. The paths chosen reflect a deeper design approach philosophy where some innovators are focused on small devices with high levels of inter-connectivity and stability, while others are focused on building the largest device possible. The philosophy of the large devices it that with many *physical qubits*, the device can manage its own error.[9]

We've seen this behavior before repeatedly over the 70-year history of computing. Computer engineers in the 1950s experimented with a variety of computing and storage media before settling on silicon for switching, core memory for short-term storage, and a combination of hard drives, magnetic tape and punch cards for long-term storage. Similar technology competitions and selections took place in the world of high-performance supercomputers in the 1970s and 1980s. This fight played out once again during the emergence of cloud computing in the 2000s, with the surprising discovery (to some) that vast computing clouds built from commodity hardware could outperform specialized high-performance systems on a wide variety of tasks, once companies like Google and Amazon developed approaches for overcoming the challenges with scale.

6.2.2 Five kinds of quantum computers

The word "quantum" is attached to a range of devices, and terminology in the field sometimes takes a functional approach. That is, the category of the device is cast by its use rather than its underlying architecture and capabilities. The lines between different categories of quantum computers blur. When it comes to computing, the word *quantum* can can describe:

- **Simulations of quantum computers.** On the most basic level, classical computers can be optimized to simulate quantum effects. The fundamental problem with using classical computer to simulate quantum systems it that

[9] Doug Finke, the publisher of the Quantum Computing Report, maintains the most comprehensive and up-to-date summary and categorization of hardware and software approaches by competitors. Finke's site carefully tracks claims of device size, quality, and construction.Finke, "Qubit Count"

today's algorithms require exponentially more steps to simulate a quantum system as the number of quantum particles increases; quantum computers do not have this problem (see **Section 5.1.2, "Modeling Chemical Reactions,"**). However, we do not know if this exponential scaling is fundamental or not; an answer to that question would likely also result in an answer to the question of whether or not $P = NP$ (see the **sidebar "??"**).

- **Quantum annealers.** Quantum annealers achieve quantum effects in specially-prepared materials. D-Wave System's quantum annealer is the most well-known device in this category. A quantum annealer uses a metal material that exhibits quantum properties as it is cooled to temperatures close to absolute zero. Unlike a general purpose quantum computer, which uses gates to process qubits, the annealer is analog. The annealing process directly manipulates qubits.

  Quantum annealers are limited in function. Although D-Wave's machines have literally thousands of qubits[10], the numbers cannot be compared with other kinds of quantum computers because the D-Wave qubits are not universal: they can only be used to solve a limited range of quantum problems. Specifically, the D-Wave can only solve problems phrased as quadratic unconstrained binary optimization (QUBO) calculations. When it comes to QUBO problems, D-Wave can solve problems that are significantly larger than almost all private companies in the field. D-Wave also hopes that it's ability to solve optimization problems will make the system commercially attractive today to companies not interested in learning about quantum computing, but interested in actually using quantum computing to solve other problems. At this point, however, there is no clear evidence that D-Wave's systems are more cost effective at optimizing that existing commercial optimizers such as CPLEX and Gurobi run on traditional electronic computers.

- **Quantum simulators.** Feynman's view that quantum computers would simulate quantum interactions is being pursued in the form of quantum simulators, devices that use, "entanglement and other many-particle quantum phenomena to explore and solve hard scientific, engineering, and computational problems," according to the report singed by 37 attendees of a 2019 workshop organized by the National Science Foundation.[11] According to the workshop report, there are now more than 300 quantum simulators operating around the world based on a wide variety of underlying platforms. Those working in the field are pursuing a two-phase strategy: in the first phase, early prototypes are built that are research curiosities in themselves. These early devices are intended to bridge to a second phase where a broader set of researchers can employ quantum simulation, with a goal of moving second-generation devices out of quantum computing applied research laboratories and into other fields such as botany, chemistry, materials science, astronomy, and in the creation of other

[10] D-Wave Systems scaled its annealer from 128 qubits, the D-Wave "One" released in 2011, to the D-Wave 2,000Q, a 2,000-qubit annealer in 2017. The 2,000Q has been commercially available since 2017 (popular reporting suggests a $15m price tag)Temperton, "Got a Spare $ 15 Million? Why Not Buy Your Very Own D-Wave Quantum Computer" (2017)

[11] Altman et al., "Quantum simulators: Architectures and opportunities" (2019).

quantum devices, including quantum internet technologies (discussed in Chapter 7). That is, the goal is to stop doing research on quantum simulators, and to start doing research *with* quantum simulators.

Quantum simulators are similar in design to quantum computers, but as with quantum annealers, quantum simulators are not universal: simulators are constructed with a single goal of simulating quantum mechanical systems, and often on a single scientific problem, such as understanding photosynthesis. By taking the complexities involved in the pursuit of universality off the table, some see quantum physics simulators as the most compelling near-term strategy for quantum computing. The NSF group predicted: "Scaling existing bottom-up quantum simulators to hundreds or even thousands of interacting, entangled, and well-controlled quantum elements is realistically within reach."[12]

- **Noisy Intermediate-Scale Quantum Devices (NISQ).** NISQs represent the state of the science in programmable digital quantum computing. Universities, research labs, and private companies are pouring untold sums of money into developing an "intermediate-scale" device that could lend insights into the building of larger devices. That is, a mid-scale quantum computer with 50–100 qubits might reveal characteristics of materials or engineering that makes creation of a 500 qubit device possible, and so on.

  NISQs are being built with several technology substrates, all familiar to readers of **Chapter 2, "(FINAL) Quantum Sensing and Metrology,"**. Several large companies such as Google and IBM are betting on the superconducting circuit approach, where Josephson junctions form the basis of the architecture. This is the same underlying approach as superconducting quantum interference devices discussed in Section 2.2 (p. 30).

  Others, such as Honeywell, are experimenting with ion trap approaches (see Figure 6.2.2), where charged electronic particles are held in position with lasers, magnetic fields, or even in a physical substrate, such as the nitrogen-vacancy approach discussed in Section 2.2 (p. 31). Ion traps do not require supercooling and enjoy long coherence times, but to date have been very limited in their number of qubits.[13]

  Photons are another option for NISQs. Photonic approaches also avoid supercooling and have good stability, and can be implemented using existing materials, like silicon and optical devices from commercial providers such as ThorLabs. As of this writing, the largest quantum computer is a photonic interferometer in China, but the device is limited to a single scientific application (see Figure 6.2.2).

  Microsoft is pursing a cutting-edge approach known as "topological qubits," which involves splitting an electron in order to store information redundantly

[12] Altman et al., "Quantum simulators: Architectures and opportunities" (2019).

[13] In June 2020, Honeywell announced that it had created "the world's highest performing quantum computer," bench-marking it with IBM's notion of a "quantum volume" of 64. (Honeywell, *The World's Highest Performing Quantum Computer is Here* (2020)) The computer had only six qubits, yet its interconnection and low noise led the company to make dramatic performance claims. (Crane, "Honeywell claims it has built the most powerful quantum computer ever" (2020).)

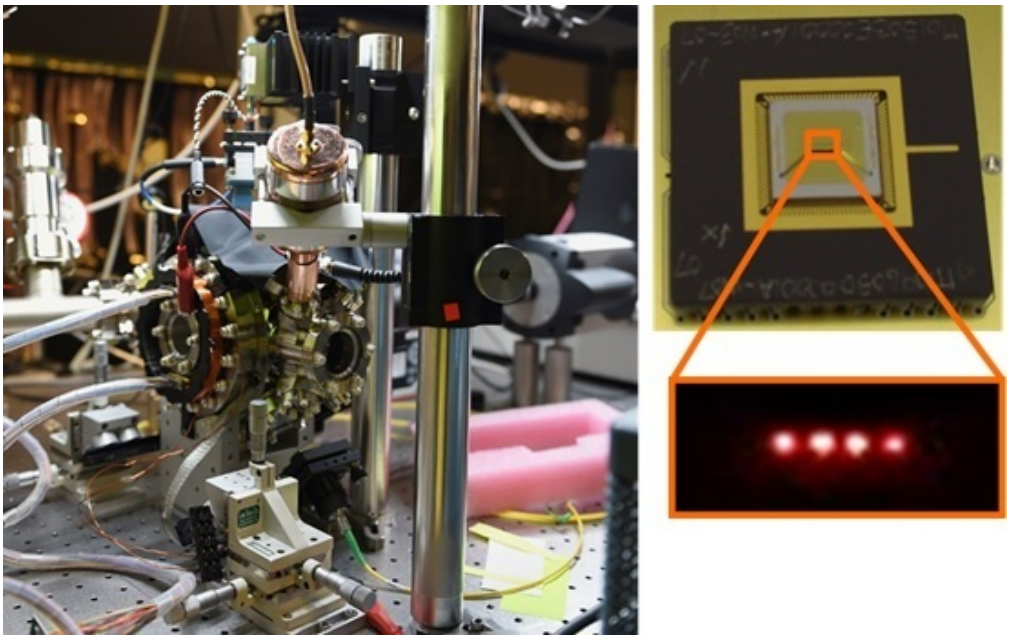


Figure 6.5: The device on the left is a vacuum chamber that houses four trapped ytterbium ions (on right) from Sandia National Laboratory. These ions can be measured using single-photon-sensitive media and are hoped to be a substrate for quantum computing and quantum memory. Photo courtesy U.S. Air Force.

and thus manage noise problems that cause decoherence. This approach is promising, but it is not nearly as developed as other approaches.

Despite their cutting-edge engineering, The National Academies of Sciences (NAS) characterizes NISQs as having "primitive" gate operations and as being plagued by error and decoherence.[14] A 2019 NAS report concluded that today's NISQs will never scale to become the large-scale, general purpose quantum machines so desired.

- **Large-scale quantum computers.** For many of the above-described efforts, the goal is to create a large, stable, universal digital quantum computer with millions of error-corrected qubits. Such a device would be similar to a modern high-performance computer. Stored in its creator's cloud warehouse, its universal functionality could be leased out to users to solve all manner of interesting problems. The question is now to realize that goal.

  One path is through fundamental discoveries in materials science, chemistry, or physics that can be applied to manage qubits. Indeed, while cryptanalysis grabs the news headlines, companies in quantum computing identify chemistry and materials science as their research focus. This is because with a mid-scale quantum computer, one might discover fundamental insights in materials design and in chemistry that elucidates strategies to build a larger quantum computer. Thus, like classical computers before it, quantum computer strategy is to trigger a virtuous cycle of growth. This insight also foreshadows an innovation policy issue: groups that can make those fundamental obser-

[14] Grumbling and Horowitz, *Quantum computing: progress and prospects* (2019).

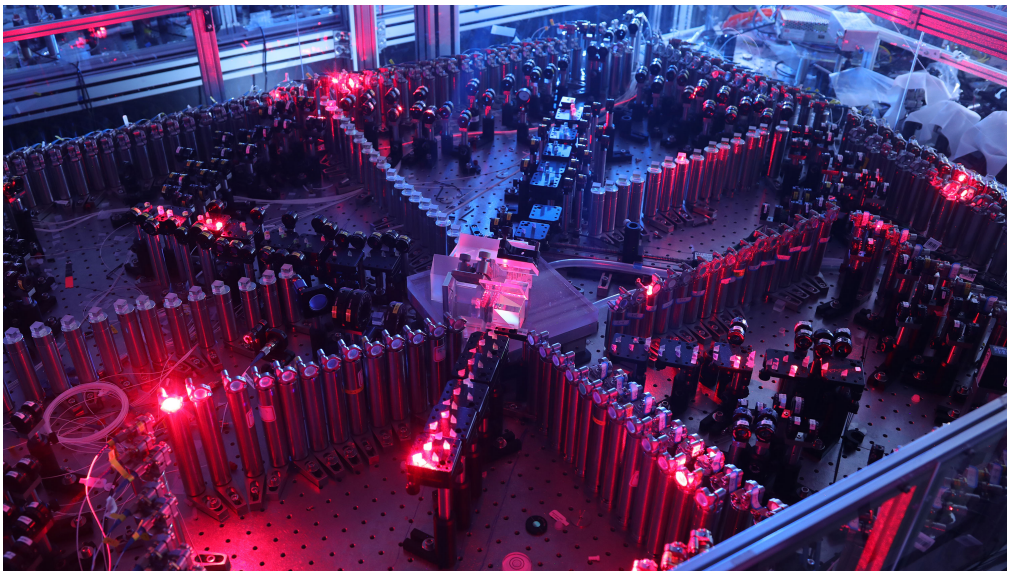


Figure 6.6: In 2020, Jian-Wei Pan and Chao-Yang Lu at the University of Science and Technology of China built a large-scale interferometer to solve the "boson sampling" problem, a task insoluble with classical computers. With 25 laser sources and 100 single-photon sensors, the Jiuzhang Device demonstrates the link between quantum sensing and computing. Image courtesy of Jian-Wei Pan.

vations are likely to pull ahead of the pack, building ever-larger computers with teams that were trained over decades, using discoveries that competitors cannot obtain. In this large-scale scenario, quantum computing could be a *winner-take-all technology*, suggesting that the first innovator might well become the most successful one.

Alternatively, the path to the large-scale quantum computer may be just a matter of scaling up existing approaches. This appears to be the strategy of several reputable companies in the quantum computing field that are creating ever-larger devices based on superconducting circuits. Perhaps the manufacture of densely-produced, well connected and controlled Josephson junctions will yield room-sized quantum computers with millions of qubits.

When will a large-scale quantum device be built? Even scientists at companies known to enthusiastically promote their technologies say that it will take a decade. Some say several decades. Others say this task is impossible. The next section turns to the reasons why building a quantum computer is so difficult.

6.3 Quantum Skeptics Present Quantum Computing's Challenges

Almost 20 years ago, physicists Jonathan P. Dowling and Gerard J. Milburn wrote that humankind had entered a new stage of quantum information science: the second quantum revolution. In the first quantum revolution, scientists used quantum mechanics to better understand our reality. Truly a scientific revolution, the first period of QIS started with theory and expanded over the century as more insights were made (see Chapter A and Chapter B). The second quantum revolution is a technological one, the focus of I, where scientists actively employ "quantum mechanics to alter the quantum face of our physical world."

Dowling and Milburn canvassed the exciting state-of-the-science developments of this second revolution. Finally they warned that, "A solid-state quantum computer is probably the most daunting quantum technological challenge of all and will require huge advances in almost all the areas of quantum technology we have discussed."[15]

Significant progress has been made since then. Nevertheless, quantum computing still depends on realizing a number of technical feats. Until now we've presented the challenges as significant but surmountable. However, a significant number of well-credentialed experts maintain that general purpose quantum computing is simply not achievable with physics as we understand it today. This section details those challenges.

6.3.1 Scientific Challenges

In 2018 the National Academies of Sciences characterized quantum computing as consisting of creating small, proof-of-concept, demonstration devices.[16] This is because quantum computing requires a mastery of quantum superposition and entanglement, development of software and control systems, and management of costly, difficult physical conditions. But more than that, breakthroughs in quantum computing may also require fundamental breakthroughs in basic physics—or at very least, transitioning phenomena that have only been observed in a laboratory setting (and only in the last decade) into engineering prototypes.

To get an idea of the gap between theoretical advance and engineering realization, consider that Microsoft's approach, the "topological qubit,"[17] is based on a 1937 theoretical prediction that single electrons can be split into sub particles.[18] Now Microsoft hopes to use the phenomena to create a working quantum computer. But it took 75 years between the theory's discovery and Microsoft's demonstrated in 2012, through a collaboration with the Delft University of Technology (TU Delft), the oldest and largest Dutch public technological university in the Netherlands.[19]

Some argue that quantum computing will never be achieved; in fact some claim that quantum computing as a field is near its end. Physicist Mikhail Dyakonov summarized the challenges in a 2018 piece: "Such a computer would have to be able to manipulate—on a microscopic level and with enormous precision—a physical system characterized by an unimaginably huge set of parameters, each of which can take on a continuous range of values. Could we ever learn to control the more than 10^{300} continuously variable parameters defining the quantum state of such a system? My answer is simple. *No, never*."[20]

A chorus of other commentators have downplayed quantum computing as an overhyped phenomenon. In 2015, a U.S. Air Force advisory board found that technology advocates "herald[ed]" imminent breakthroughs but nevertheless, "no compelling evidence exists that quantum computers can be usefully applied to computing

[15] Dowling and G. J. Milburn, "Quantum technology: The second quantum revolution." (2003).
[16] Grumbling and Horowitz, *Quantum computing: progress and prospects* (2019).
[17] Microsoft Corp., *Developing a topological qubit* (2018).
[18] Majorana and Maiani, "A symmetric theory of electrons and positrons" (2006).
[19] Mourik et al., "Signatures of Majorana fermions in hybrid superconductor-semiconductor nanowire devices" (2012).
[20] Dyakonov, "When will useful quantum computers be constructed? Not in the foreseeable future, this physicist argues. Here's why: The case against: Quantum computing" (2019a).

problems of interest to the Air Force."[21]

The most specific critique comes from a 2018 National Academy of Sciences (NAS) survey of the field that made both economic and technological assessments. On the economic front, the NAS group observed that there are essentially no economically advantaged uses for quantum computers for the foreseeable future (and obviously no consumer ones either).[22] This is directly different from the history of computing, in which spending money on computing was advantageous from the very first dollar spent. From the beginning, spending money on computing—be it mechanical, electromechanical, or electronic—made it possible to do something that wasn't otherwise possible, or to do it faster, or for less money overall. Although quantum computing might one day make it possible to train large-scale artificial intelligence machine learning models faster and with far less electricity than is currently the case, this does not seem to be a breakthrough that is plainly visible on the short-term horizon.

6.3.2 Engineering Challenges

Without uses that produce big savings or profits in the near term, funding for quantum computing is likely to be limited to governments and the largest technology companies. As such, quantum computing lacks the "virtuous cycle," like what was enjoyed with classical computers, with increasing commercial and consumer utility driving demands and willingness to pay for fantastic technological innovations.

The NAS survey's core technological critique is relates to the difficulty of scaling up today's quantum systems into larger systems that can be used to solve meaningful problems. As a result of these challenges, the survey found it too uncertain to predict when a scalable quantum computer would be invented and that existing devices could never scale into general-purpose machines.

Quantum computers are characterized by the integration of multiple qubits. Thus, for a quantum computer to work, one needs to be able to encode, entangle, manipulate, and maintain an array of qubits, raising the challenges visited in Chapter 2. The challenges inherent in quantum computing are thus different from the obstacles encountered by engineers building and then scaling digital computers. Classical computers went through an evolution of mechanical, to relay, to tube, and to discrete transistors, and finally to integrated circuits. Each improve produced systems that were smaller, faster, and required less overall energy to perform a computation. Semiconductors enjoyed their own virtuous cycle, providing chip makers with tools for designing and manufacturing make computers that were ever more complex yet less expensive. Quantum computing has not realized a scaling breakthrough on the level of the transistor. Perhaps more to the point, there is no such breakthrough lurking in the future of any realistic technology road map. In many ways this is similar to the days of mechanical, electromechanical and tube-based computing, when larger computers might be faster than smaller ones, but they were also dramatically more expensive and less reliable.

Different technologies can be used to create qubits, but for each, quantum scientists must be able to master and control events at quantum scales (see Chapter A). Some of the technologies used include ion traps (spins of ions), quantum dots (the

[21] Board, *Utility of Quantum Systems for the Air Force Study Abstract* (2015).
[22] Grumbling and Horowitz, *Quantum computing: progress and prospects* (2019).

spin of electrons or their energy level), photonic (the position or polarization of photons) and superconducting circuits (the magnetic spin of electrons in artificial atoms). These functions require substantial technical expertise, reflected in the multidisciplinary nature of quantum computing teams (engineers, physicists, mathematicians, computer scientists, chemists, materials science). This is also a difference from the last 70 years of computing, which generally required mastery of fewer technical domains, and were modularization and isolation between technical domains meant that there was a need for comparatively less requirement for interdisciplinary work.

Quantum computers require that their qubits be entangled, cohered into a group that can be operated upon. But at the same time, quantum computers must be shielded from the universe, lest noise in the environment cause those qubits to decohere. This makes the quantum computer challenge fundamentally different from the classical computer. The transistor allowed scale with intricately-managed stability. However, with quantum computers, scale requires the management of additional, exquisitely fragile quantum states.

When qubits decohere, they lose information. Thus, quantum algorithms have to be crafted to be efficient enough to execute before coherence is lost. As of this writing, some state-of-the-science devices have coherence in the hundreds of *micro*seconds, a time too short for the quantum gates of today to process significant numbers of qubits. This is a time period so short that human physical experience has no analogue for it. A blink of the eye takes about 100,000 microseconds.

The longer quantum computers run, the more performance degrades. In classical computing, extra bits are used to correct ordinary errors that occur in processing. This approach works because of all the engineering performed in classical computers to avoid quantum effects like tunneling. In quantum computing, many of the qubits employed are dedicated to error correction, so many that it creates significant overhead and degrades computing performance. Current thinking is that to emerge from the era of NISQ machines, as many as 90% of a quantum computer's qubits might have to be dedicated to error correction.[23] Initially, one might suggest just adding more qubits to achieve reliability, but as more qubits are added, system complexity increases, and quantum devices become more prone to both random environmental interference and to noise from the computer's own control system.

Quantum computers are not fault tolerant. In addition to temperature, vibration and electromagnetic interference can easily destabilize quantum computers. Conventional electronic computers rely on the digital discipline to smooth out errors so that they effectively do not matters.[24] In quantum devices, by contrast, errors and not rounded out, but instead compound until the conclusion of the computation.

To shield quantum computers from environmental noise that triggers decoherence, many quantum computer architectures require supercooling. This cooling is *super* because it is colder than even the background temperature of the universe. Extreme frigidity is needed both to elicit quantum properties from materials (for instance, in analog quantum annealers) but also because heat increases the chances that random energy collisions will generate noise that will interfere with quantum

[23] Möller and Vuik, "On the impact of quantum computing technology on future developments in high-performance scientific computing" (2017).

[24] In classical computing, bits of data are either a `0` or `1`. In that environment, error appears as a decimal value such as `0.1` or `0.9` that can be easily rounded to `0` or `1`. For more information, see p.63.

states or cause decoherence.

Keeping quantum devices at 15 millikelvin (−273 C, −459 F) means that quantum computer scientists need liquid helium, an increasingly rare and valuable element, of which there is a finite supply of on Earth. There are currently no limits on the usage of Earth's helium supply.[25] Unlike quantum computing, many other quantum technologies do not require supercooling. This means that some sensing and communications technologies can be miniaturized, commercialized, and deployed in many more challenging contexts (in outer space, underwater, in missiles) than quantum computers.

6.3.3 Validation Challenges

It will be necessary to validate quantum computers to make sure that the answers they produce are correct. Ironically (and annoyingly), validation is easy for many of the hard, long-term applications for quantum computing, but likely to be harder for the more likely, near-term applications.

For the algorithms like factoring with Shor's algorithm and search with Grover's, validation is easy: just try the answer provided by the quantum computer and see if it works. That is, if the quantum computer says that the `2227` are `131` and `17`, one need merely multiply 131×17 to determine if the factorization is correct or not. The same logic applies to using Grover's algorithm to crack an AES-128 key: just try to decrypt the encrypted message: if the message decrypts, the AES-128 key is correct.

On the other hand, approaches for both error-correction and validation are less developed for analog quantum simulators. One approach suggested in the 2019 NSF report is to run simulations forward and backwards (theoretically possible, since the computations should be reversible) to see if the simulator retraces its step. Another approach is to see if different systems that should have equivalent outcomes do indeed have similar outcomes.

6.3.4 Ecosystem Challenges

A final challenge is not technical, but organizational. Significant work still needs to be done to create a rich ecosystem of quantum software. Beyond basic programming languages and compilers, which exist today, there is need for documentation for people at multiple levels of expertise, programming courses, systems on which to run those programs, and finally organizations willing to pay for training and to hire quantum programmers.

On the software front, many teams are developing languages to make interaction with quantum computers more routine and standardized. As of 2021, a growing "zoo" of quantum algorithms included 430 papers.[26] But the overwhelming number of these algorithms are expressed as *papers* in *scientific journals* or on *preprint servers*; they are not code on sites like GitHub that can be downloaded, incorporated in to other, larger quantum programs, and run. Recalling that Ed Fredkin got

[25] Some hope that early quantum computers will solve fundamental challenges in fusion. If that happens, we could create helium via hydrogen fusion.

[26] Montanaro, "Quantum algorithms: an overview" (2016); Jordan, *Quantum Algorithm Zoo* (2021).

himself hired in 1956 without a college degree at BBN to write programs for the company's first computer (which he convinced BBN to purchase—see Section 4.4.1 (p. 110)), we have not yet reached the point where it is possible to teach yourself quantum programming and get a job at a company that needs someone to write quantum algorithms to run on their quantum computer.

6.3.5 Quantum Supremacy and Quantum Advantage

Quantum Supremacy is an awkward term. As Preskill defined it in 2012, the goal is to perform a computation—any computation—that cannot be performed with a classical computer. But the term is misleading, because quantum engineers in China and the US have clearly achieved "supremacy" as defined by Preskill, but quantum computers are not supreme: for the vast majority of computations performed on planet Earth, you would not be able to use one of today's quantum computers. And even if reliable, large-scale quantum computes are available in the future, it is hard to imagine that these machines will be used for more than a tiny fraction of the world's computing problems. And even in these applications, Quantum computers are likely to be co-processors that depend on classical computers for many functions. For these reasons, we prefer the term "quantum advantage" to describe the achievement of solving a problem with a quantum device that cannot be solved with a classical computer.

In December 2020, Jian-Wei Pan and Chao-Yang Lu made the most compelling claim of quantum advantage to date.[27] Their team built a large-scale interferometer to compute a specific problem, Gaussian Boson Sampling (GBS).The team named their device Jiuzhang, for the ancient Chinese manuscript focused upon applied mathematics, *Nine Chapters on the Mathematical Art*. But as exciting as the Jiuzhang development is, the device can perform just one computation. However, it's really fast!

Previously, Google researchers announced in October 2019 that they had achieved quantum supremacy using their 54 qubit Sycamore superconducting approach.[28] The Google researchers programmed their computer to create and then evaluate random quantum circuits. IBM, a chief rival to Google, quickly disputed the supremacy claim, arguing on its research blog that "ideal simulation of the same task can be performed on a classical system in 2.5 days and with far greater fidelity."[29] In March 2021, two Chinese scientists claimed that they replicated the Google approach with higher fidelity using classical GPUs.[30] The quick retorts to Google's claim demonstrates the value of quantum computing bragging rights, even if the bragging is only about the ability to solve otherwise meaningless random quantum puzzles.

While the Jiuzhang device is a clear demonstration of quantum advantage, the device has limitations in application. Whereas Google's claim of advantage stands on contested ground, the Sycamore device can be programmed to solve other problems

[27] Zhong et al., "Quantum computational advantage using photons" (2020).

[28] Arute et al., "Quantum supremacy using a programmable superconducting processor" (2019).

[29] Pednault et al., "On "Quantum Supremacy"" (2019).

[30] Pan and Zhang, *Simulating the Sycamore quantum supremacy circuits* (2021), The authors conclude with a humble brag that their "proposed algorithm can be used straightforwardly for simulating and verifying existing and near-future NISQ quantum circuits" and the authors helpfully posted their approach on Github.

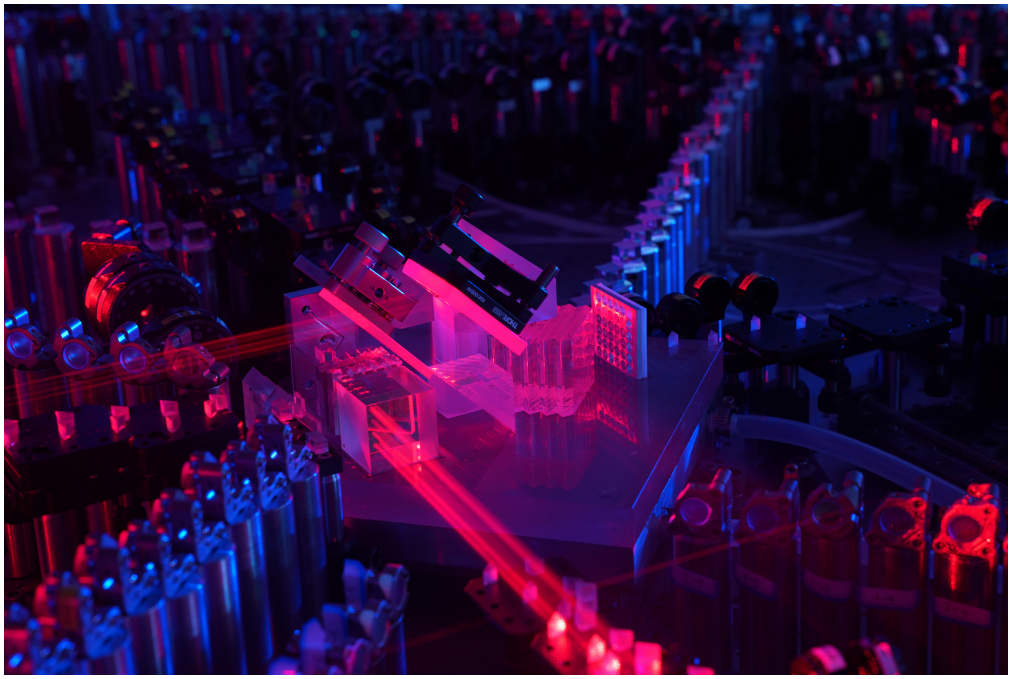


Figure 6.7: Computing a specific distribution of photons that would have taken 600 million years to solve on the fastest existing classical supercomputer in 2020, was computed in 200 seconds with a reported 99% fidelity by Jian-Wei Pan and Chao-Yang Lu at the Hefei National Laboratory, University of Science and Technology of China. However, turning the device into a "fault-tolerant universal quantum computer, is a very long-term goal and requires many more challenges to tackle, including ultra-high-efficiency quantum light sources and detectors, and ultra-fast and ultra-low-loss optical switch," Lu told us. Image courtesy of Jian-Wei Pan.

The Helium Challenge

Helium is a colorless, odorless, inert gas. Its stability, non-reactivity, and phase as a fluid at near absolute zero temperatures makes it useful for quantum computing and critical for variety of industrial applications, from welding to the cooling of nuclear reactors to the cooling of magnets in Magnetic Resonance Imaging machines. Helium is abundant in the universe but on Earth it collects underground as a result of radioactive decay and is typically rendered as a byproduct of natural gas. If it is not captured but instead released into the atmosphere is rapidly diluted (dry air at sea level is 5.24 parts-per-million helium) and no longer financially viable to collect. A small amount of helium escapes to the upper layers of the atmosphere, where it is it is torn away from the earth by the solar wind. As such, helium is a non-renewable resource.

A large portion of domestic U.S. demand for helium is provided by a storage and enrichment facility in Amarillo, Texas, run by the U.S. Bureau of Land Management. The United States and Qatar are the largest producers of helium. But Russia's Gazprom and China are building plants in order to reduce their reliance on U.S. sources. Because of helium's many uses, limited availability, and strategic relevance, conservationists have called for an international helium agency to preserve supply and prevent a crisis in availability, and to expand extraction of helium from existing natural gas plants.[a] But don't feel guilty about your kids' helium balloons. Such consumption is inconsequential compared to industrial and medical uses.

Different quantum technologies require more or less helium. The biggest consumers are MRI machines and devices that are used at border crossings to detect dirty bombs and other nuclear devices. Quantum computers use less helium and modern cryogenics equipment attempts to conserve and recycle it. D-Wave explicitly markets its annealer as recycling helium to avoid the need to continuously resupply the machine's local store of helium.

On the other hand, some quantum computers require light helium, Helium-3. This is extracted from nuclear reactors, and is somewhat controlled.

The complex web of nation-state conflict and the technological need for cooling is spawning different strategies. In the U.S., IBM's plans for a 1,000 qubit superconducting device caused the company to develop a custom dilution refrigerator. While others are building supercooling capacities that do not use a cryogen like helium or liquid nitrogen. These non-cryogen coolers have a major disadvantage: they require much more electricity for cooling. However, as nations signal an interest in decoupling their technology stacks, nations without access to helium sales may simply turn to electric cooling.

[a] Nuttall, Clarke, and Glowacki, "Stop squandering helium" (2012).

other than random puzzles, so it is probably more important from a commercial point of view.

For computer scientists, achieving quantum advantage was long seen as a kind of Rubicon. But for most organizations, the real quantum computing Rubicon will be the moment that quantum computing can perform some useful commercial, defense or intelligence application. And even that moment is likely to be linked to how understandable the commercial application is. If the first commercial applications are in advertising or business optimization, the public is likely to notice, but if instead the first applications are in chemistry simulation, few will realize a Rubicon has been crossed.

How can one make sense of quantum computers' power when they rely on different physical media (ranging from photonics to trapped ions to annealing) and when innovators claim to have more qubits than competing devices? Quantum computers cannot be evaluated simply by the number of qubits they have, otherwise D-Wave's 2000-qubit system would be leagues ahead of teams at IBM, Google, and Microsoft—even when those systems can clearly perform computations that the quantum annealer can't. To evaluate quantum devices, IBM created its own metric called *quantum volume.*[31] A computer's quantum volume is "the largest random circuit of equal width and depth that the computer successfully implements." Thus, quantum volumes are necessarily perfect squares: 2, 4, 9, 16 and so on. Unfortunately, the largest quantum volume that IBM measured was 16, on a machine with 4 qubits running a circuit with a depth of four gates. "We conjecture that systems with higher connectivity will have higher quantum volume given otherwise similar performance parameters," the authors state.

Despite all these challenges, governments and large technology companies (e.g. Fujitsu, Google, IBM, Microsoft, Toshiba), have devoted major resources to quantum computing and several startups (e.g. IonQ, Rigetti, Xanadu) are betting the company on it. Competition has produced wonderful resources to learn about and even experiment with quantum computing. For instance, IBM and others have made instructional videos, extensive, carefully curated explanatory material, and even made rudimentary quantum computers available through the cloud at https://quantum-computing.ibm.com for anyone who wants to try their hand at programming the machines.

Quantum computing efforts are either basic or applied research. Basic research projects, like the Large Hadron Collider (LHC) at the European Organization for Nuclear Research (CERN), can be huge impressive projects that reveal fundamental truths about the nature of the universe: at a cost of approximately $9 billion, the LHC is one of the most expensive scientific instruments ever built, and it is responsible for the "discovery" of the Higgs boson, but it is hard to draw a line from the LHC to improvements in day-to-day life of anyone except for several thousand construction workers, physicists and science journalists. On the other hand, nuclear fission was discovered in December 1938 by physicists Lise Meitner and Otto Frisch,[32] which led to the creation of a working nuclear bomb within just seven years and the first nuclear power plants in 1954. Such is the unpredictability of research.

[31] Cross et al., "Validating quantum computers using randomized model circuits" (2019).

[32] Tretkoff, "This Month in Physics History: December 1938: Discovery of Nuclear Fission" (2007).

6.4 The Outlook for Quantum Computing

The long-term outlook for quantum computing may be hazy, but the near-term outlook for quantum computing companies appears to be quite bright.

As we saw in the last chapter, although it was the potential for quantum computers to crack codes that led to the initial burst of enthusiasm, interest in quantum computing is likely being sustained by the promise of using quantum technology as an advanced scientific instrument for learning more about quantum physics and quantum chemistry. The payoffs may be directly in these fields, or they may simply be the development of superior quantum sensors that are usable throughout the military industrial complex.

As such, there are many practical regulatory implications at least in the short term:

1. Because of their expense and complexity, only large firms and governments are likely to be able to afford quantum computers for some time. This means that governments have a relatively small number of players to police in quantum computing, and that the technologies may be easier to monitor and control. This period of large-organization exclusivity may continue for decades. Consider that classical computers were the domain of universities, governments, and large companies until the personal computer revolution of the 1970s

2. Because of their complexity, quantum computers require teams of multidisciplinary experts. This means that one cannot simply sell a quantum computer and expect a user to make sense of it. Sellers will be on-the-premises of buyers and will probably know about the buyers' intended uses of the devices. The business model may be selling services as much as selling the device itself.

3. Because of their sensitivity to interference of all types, quantum computers are likely to be placed in low-noise environments. For instance, the D-Wave system occupies a 10x10x10 foot housing plus three auxiliary cabinets for control systems. The cabinet is part of a system to produce quantum effects in D-Wave's annealer, where the chip is the size of a thumbnail. This requires a vacuum environment, a low-vibration floor, shielding to 50,000 times less than the Earth's magnetic field, and cooling to 0.0012 Kelvin.[33] Such devices are unlikely to be installed in jets for forward-deployed use, although they might be deplorable in a suitably outfitted ship.

4. Finally and relatedly, larger firms are likely to offer quantum processing through the cloud until fundamental physical challenges are overcome and quantum devices reach a price point available even to medium-sized enterprises. Until then, quantum computing is likely to be offered as an enhanced service, one optimized for specific problems.[34][35]

Taken together, these limits will shape the trajectory and offerings of quantum computers.

[33] Copeland, *The International Quantum Race* (2017).
[34] Ibid.
[35] Gibney, "Quantum gold rush: the private funding pouring into quantum start-ups" (2019).

Despite the lack of a practical demonstration, many scientists believe that sufficiently large quantum computers will be much more powerful than classical computers for solving certain kinds of problems. We lack *proof* that quantum computers will be innately more powerful for the same reason that we lack proof that factoring is fundamentally more difficult than primality testing, or that mixed integer linear programming is fundamentally harder than linear programming. That is, we don't have a proof that $P \neq NP$ (see Section **??** (p. **??**)).

7

(IN EDIT) Quantum Communication

"Quantum communications" refers to two related applications: first, the use of quantum states to ensure true randomness in number selection and to communicate encryption keys to other parties, known respectively as quantum random number generation and quantum key distribution; second, the use of quantum effects themselves, such as the the spin of photons, to encode a message, which is known as quantum internet or quantum networking.

There are four reasons to be excited by quantum communications and these three advantages are strategically relevant:

1. Properly implemented, quantum communications applications enjoy *information-theoretic security*, which means that no adversary, regardless of their computing resources or background knowledge, can decipher communications that have been covertly intercepted. Not even a quantum computer can decrypt such communications! This is because the security is a property of the underlying mathematics and quantum physics, rather than the putative "hardness" of a particular math problem.

   Quantum security guarantees protect institutions against the future. Those continuing to use *computationally-secure* post-quantum classical alternatives for distributing their keys rely on assumptions that may be proven incorrect. For instance, a mathematician may discover a new algorithm that unscrambles post-quantum encryption.

2. Quantum communications systems, unlike classical ones, reveal when a communication has been intercepted. That interception could be a surveilor, or it might be ordinary environmental interference, such as electronic noise or malfunctioning hardware. (Users of such systems typically cannot determine if the message failure was an accident of the environment or the actual presence of an eavesdropper.) The detection of interception capability results from the nature of quantum states. The act of interception interferes with quantum states, and this interference can be detected, unlike in classical communications, where interception is both easy and stealthy.

   For this reason, properly implemented quantum communications systems are not susceptible to proxying attacks, also called machine-in-the-middle or man-in-the-middle attacks. That's because if an attacker does intercept a photon carrying a particular quantum state, it is impossible for the attacker to both measure the photon's quantum state and retransmit a photon with the same quantum state.

3. In a fully-quantum network that uses quantum states themselves to communicate, communication security becomes end-to-end. Users no longer have to rely on network trust, and can shut out eavesdroppers from both the content of their communications *and the metadata about those conversations*. Because governments extensively use metadata to study adversaries, this meta-data-denying affordance of quantum internet schemes may be what is driving quantum network investments in Europe and China.

4. Just as Grover's algorithm speeds up some kinds of computations when performed on a quantum computer, some kinds of multi-party mathematical protocols enjoy a similar speedup when the parties communicate over a quantum network.

These benefits of quantum communications—information theoretic security, awareness of message interception, the possibility of metadata secrecy, and certain kinds of optimizations—are driving both interest in quantum communications and its early commercialization. Indeed, the first quantum key distribution systems reached the market in 2005.[1]

Although quantum communication was discovered before quantum computing, another way to think about quantum communications systems is as a quantum computer with a "flying qubit" that travels from one party to the second, or with two flying qubits that travel from a common sender to two different receiving parties.

Quantum communications builds upon the technologies of quantum sensing discussed in Chapter 2, including single-photon detectors, the ability to perform low-noise measurements of quantum states, and even superconducting quantum devices.[2]

This chapter sets the stage for interest in quantum communications by briefly explaining the rise of signals intelligence (SIGINT) (Section 7.2 (p. 193)) capabilities of governments and the proliferation of these powers to non-governmental actors. SIGINT is information derived from communications systems, radars, and weapons systems.[3] The chapter continues by explaining three quantum communications technologies, all of which can contribute to the confidentiality and integrity of communications.

First, quantum random number generation techniques use quantum uncertainty to create truly random numbers. Computer systems use high-quality random numbers in security, in simulations, and statistical models.

Second, quantum key distribution techniques use randomness to make secure encryption keys and ensure their confidentiality and integrity when they are transmitted to multiple parties. Although these protocols are called *quantum* key distribution, they are ultimately used to secure *classical* communications, for instance over the regular internet or even the telephone.

Finally, a quantum internet would preserve quantum and allow quantum computation between parties in different physical locations—possibly over great distances. This would provide both security against interception and secrecy of metadata. If

[1] Garfinkel, "Quantum Physics to the Rescue: Cryptographic systems can be cracked. And people make mistakes. Take those two factors out of the equation, and you have quantum cryptography and a new way to protect your data." (2005).

[2] Takemoto et al., "Quantum key distribution over 120 km using ultrahigh purity single-photon source and superconducting single-photon detectors" (2015).

[3] National Intelligence, *What is Intelligence?*

the quantum networking necessary to achieve the ideal of a quantum internet were achieved, one could likely use the technology to connect disparate, small quantum devices into a larger cluster computer, or connect multiple quantum computers together to create a larger quantum computer. Quantum networking can speed up certain protocols.

7.1 Information-theoretic Security

To understand the power of information theoretic security is to understand the sublime attraction of quantum methods for protecting communications. Because many readers will not be familiar with the concept of information-theoretic security, we present below three math problems: one that is easy, one that was hard in 1977 when it was posed but was solved in 1994, and one that is information-theoretic secure, which means that it cannot be solved with the information that we present, even by an attacker who has unlimited computer power.

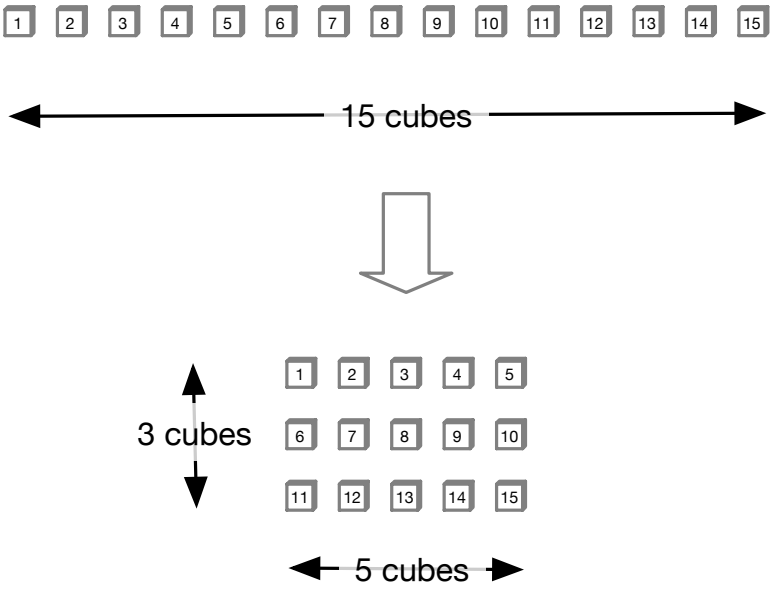
7.1.1 An easy math problem

Here is an easy math problem. The variables p and q are positive integers and p is less than q ($p < q$).

$$p \times q = 15 \tag{1}$$

That is, what two numbers multiplied by each other equal 15? The answer is 3 and 5. This is an easy problem.

Recall that 15 is the number factored by IBM's quantum computer factored in 2001 (Section 5.2 (p. 139)). A simple way to think about this problem is to imagine that you have 15 cubes in a single line and you want to arrange them into a rectangle. If you did that, what would be the dimensions of that rectangle be?



It turns out that there is only one way to make that rectangle, and that's with three rows of five cubes each.[4]

[4]Turning the rectangle 90 deg so that it's five rows of three cubes each doesn't count as another

7.1.2 A hard math problem

Here is a math problem that was posed in 1977 but was not solved until 1991, when it was cracked by an international team of 600 volunteers using more than a thousand computers. Instead of trying to factor the 2-digit number 15, try to break this number down to its prime factors p and q:

$$\begin{aligned} p \times q = 114381625757888867669235779976146612010218296721242 \\ 3625625618429357069352457338978305971235639587050589 \\ 8907514759929002687954354 1 \end{aligned} \tag{2}$$

This 129-digit number is called RSA-129. It was chosen by Ron Rivest in 1977 as a puzzle to accompany the publication of a Martin Gardner column in Scientific American[5]. Like the number 15 in equation 1, RSA-129 has two factors, here called p and q.[6] But what are p and q in this case? That was the problem posed by Rivest.

RSA-129 has a curious property: if you factor the number into its two primes, you can use the result to decrypt a secret message that Rivest wrote and encrypted back in 1977.

Factoring RSA-129 was computationally infeasible in 1977, Rivest didn't know how long it would be until computers were fast enough that it would be feasible. Gardner's column claims that Rivest estimated it would take "40 quadrillion years" to factor such a number. But that estimate was based on a single 1977 computer running with the best factoring algorithm of the day: in the following years computers got faster, factoring algorithms got better; it also became possible to connect many computers together to work on the same number at the same time. This is what we mean when we say that factoring RSA-129 was *computational infeasible* in 1977, or alternatively, that RSA-129 was *computationally-secure* (at least in 1977). Finding the factors of RSA-129 is left as an exercise for the reader.

7.1.3 An impossible math problem

Now here is a math problem that you can't solve no matter how much computational power you have:

> There is a line that passes through the points (x_1, y_1) and (x_2, y_2). Find the value of y where the line passes through the y-axis (that is, when $x = 0$), given that one of the points is (3,5).

That is, solve for y in this equation given $x = 0$, knowing that $x_1 = 3$ and $y_1 = 5$:

$$y = mx + b \tag{3}$$

This equation can't be solved to give a unique solution for y: you aren't provided with enough information. The equation $y = mx + b$ describes a line on a graph, where m is the slope of the line and b is y-intercept. It's the y-intercept that you

"way" in this situation, because we required that the first factor be less than the second.

[5] Gardner, "A new kind of cipher that would take millions of years to break" (1977a).

[6] Mathematicians frequently reuse variable names like p and q in different equations, just as lawyers reuse labels like "plaintiff," "defendant" and "the Court" in different lawsuits.

are trying to find. You can't find the y-intercept because you only have one point on the graph. This is an example of a problem that is information-theoretic secure (see the **sidebar "Secret Sharing"**).

Today nearly every use of encryption on the planet is protected using ciphers that are computationally secure. As we saw in Chapter 5, these algorithms can be cracked simply by trying every possible decryption key and recognizing the message when it is properly decrypted. Quantum computers promise to make this process faster. Even post-quantum encryption algorithms are still merely computationally secure: we know that with enough computer power, these algorithms can be cracked. There might also be short-cuts to cracking these algorithms that haven't yet been discovered, just as better approaches for factoring were discovered after 1977 that made it easier to factor RSA-129.

Adopters of a properly implemented quantum encryption system do not have to rely on *computationally-secure* algorithms for distributing their keys. Instead, they use qubits, safe with the knowledge that if the qubits are intercepted by an adversary, then the legitimate sender and recipient will be able to determine this fact.

There are actually two ways to use quantum cryptography, one that is secure given what we know about quantum computers today, and a second that is secure given our understanding of quantum physics and the physical laws of the universe:

1. With **Quantum Key Exchange**, flying qubits are used to exchange an encryption key that is then used with a conventional quantum-resistant symmetric encryption algorithm, such as AES-256. Because we believe that AES-256 cannot be cracked on a quantum computer, this approach is believed to be secure for the foreseeable future. That is, the key exchange is information theoretic secure, but the bulk encryption is only computationally secure.[7]

2. With **Quantum networking or "quantum internet"**, flying qubits are used to exchange *all* of the information end-to-end between the parties. This approach is information theoretic secure if the laws of quantum computing are correct. Put another way, it is secure as long as it is impossible to predict the future with absolute accuracy.

7.2 Golden Ages: SIGINT and Encryption Adoption

Signals Intelligence is one of the oldest intelligence gathering disciplines (Table 7.1). Many histories of SIGINT start with the use of wireless during World War I by both German and Allied forces: radio offered the advantage of instantaneous communications to troops in the field, potentially anywhere in the world, but suffered from risk that the enemy could be privy to the communications as well. Radio was too powerful to ignore, but too dangerous to use without some mechanism for protecting communications. Military users resolved this conflict by turning to encryption.[8]

[7] Note that AES-256 is only computationally secure against our current notions of quantum computing. It might not be secure against a computer based on quantum gravity, or strange matter, multiverse computation, or some kind of physics that we haven't yet imagined. Specifically,

HUMINT Human Intelligence Gathered from a person. Includes diplomatic reporting, espionage, interrogation, traveler debriefing, and other activities.

GEOINT Geospatial Intelligence Gathered from satellite, aerial photography, and maps.

IMINT Imagery Intelligence Analysis of images for their intelligence value. The National Geospatial-Intelligence Agency has primary responsibility for IMINT.

MASINT Measurement and signature intelligence Intelligence typically reviewed through the use of scientific measurement instruments. The Defense Intelligence Agency has primary responsibility for MASINT.

OSINT Open-source intelligence Analysis of information sources that are generally available, including news media and social media. The Director of National Intelligence's Open Source Center and the National Air and Space Intelligence Center are major contributors to OSINT.

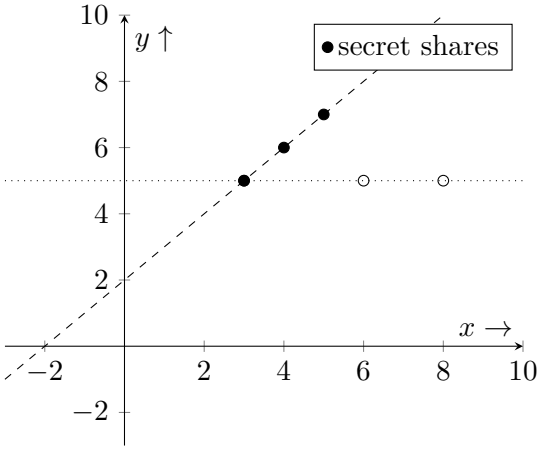
SIGINT Signals intelligence Intelligence gathered by analyzing "signals," which may include the analysis of intentional communications (COMINT—communications intelligence) and analysis of unintentional electronic emanations (ELINT—electronic intelligence). "The National Security Agency is responsible for collecting, processing and reporting SIGINT."

Table 7.1: A sampling of the intelligence gathering disciplines, from National Intelligence, *What is Intelligence?*

Secret Sharing

Secret sharing is an information-theoretic approach to splitting a secret into multiple parts. Invented independently in 1977 by G. R. Blakley[a] and Adi Shamir[b], one of the primary uses of secret sharing is splitting cryptographic keys used for data backups. Doing this renders the backup unusable unless multiple parties receiving the secret shares get together and reassemble the secret, allowing the backup to be decrypted.

Secret sharing works by representing the secret as a mathematical function that cannot be solved with the information present alone in each of the shares. In the example below, the secret is the y-intercept, which is where the straight line crosses the Y axis. Each share is a point on the line. Two points uniquely define a line, so without a second share, there is no way to identify the y-intercept.



Here we see an example of secret sharing at work. The secret is $y = 2$ (the dashed line). The shares are $x_1, y_1 = (3, 5)$, $x_2, y_2 = (4, 6)$ and $x_3, y_3 = (5, 7)$. Combining any two secrets allows reconstructing the line. Notice that if the shares had been $(3, 5)$, $(6, 5)$ and $(8, 5)$, then the secret would have been $y = 5$. Thus, there is no way for a person receiving the share of $(3, 5)$ to know the value of the secret without combining their share with a share that someone else received. Secret sharing can be used to split encryption keys between multiple parties in a way that is information-theoretic secure. A typical use of secret sharing would be for a company to distribute the encryption key for its offsite backups to three different individuals so that only by combining their shares could the company's backup be decrypted.

[a] Blakley, "Safeguarding cryptographic keys" (1979).
[b] Shamir, "How to Share a Secret" (1979).

In recent years events surely have altered balance between those who wish to eavesdrop on communications and those who wish to keep their communications private. However, there is no clear accounting as to which side is now ahead.

7.2.1 The Golden age of SIGINT

On the SIGINT side, many governments have developed audacious, comprehensive, systematic programs to capture communications and personal data in order to identify people, to attribute actions to parties and adversaries, to perform link analysis (the evaluation of relationships among people, adversaries, and others), and to capture communications content. For instance, it is alleged that in 2011 the Iranian government used compromised encryption certificates to access the email accounts of hundreds of thousands of Iranians who used Google's Gmail.[9]

In recent years, there have been repeated accounts in the U.S. media of both Chinese and Russian successes in exfiltrating data from both public and private U.S. information systems. With respect to China, the breach of the U.S. Office of Personnel Management database resulted in the theft of records on more than 20 million current and past federal employees, including fingerprint records and lengthy, detailed forms used when applying for a security clearance. Chinese hackers also reported to have stolen the credit reports on over a hundred million Americans. Between these two attacks, China can presumably identify and target people who are both likely involved in intelligence efforts and who are economically vulnerable. This data surveillance has real consequences for U.S. efforts and is believed to have enabled China to identify multiple CIA assets in Africa.[10] Turning to Russia, the former superpower has many satellites, terrestrial assets, and near-shore submarines, all of which can be used for collection of SIGINT. At the end of 2020, the U.S. intelligence stated that a supply chain attack on the U.S. company Solar Winds, which makes software to help organizations monitor their computer systems, was "likely Russian in origin."[11] More than ten thousand U.S. companies and government agencies were compromised as a result of the attack.

Books and reports that synthesize government programs into single readings, like Barton Gellman's *Dark Mirror*,[12] can seem like paranoid science fiction. In that book, for instance, Edward Snowden refuses to reveal whether he has a blender, for fear that the appliance's electrical signal would reveal his location to intelligence agencies. There is no way to know from public sources if Snowden's fears are justified.

---

it might not be secure against a device that could solve NP-hard problems in polynomial time.

[8] In fact, the use of both encryption and cryptanalysis by militaries predates the invention of radio by at least 2500 years. For a history of code making and code-breaking, we recommend David Kahn's updated classic Kahn, *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet* (1996), as well as the more manageable Singh, *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography* (2000). For a contemporaneous account of code breaking during World War I, we recommend Yardley, *The American Black Chamber* (1931).

[9] Hoogstraaten et al., *Black Tulip Report of the investigation into the DigiNotar Certificate Authority breach* (2012).

[10] Zach, *China Used Stolen Data to Expose CIA Operatives in Africa and Europe* (2020).

[11] Cybersecurity and Infrastructure Security Agency, *Joint Statement by the Federal Bureau of Investigation (FBI), the Cybersecurity and Infrastructure Security Agency (CISA), the Office of the Director of National Intelligence (ODNI), and the National Security Agency (NSA)* (2021).

[12] Gellman, *Dark mirror : Edward Snowden and the American surveillance state* (2020).

But we do know that in 2014 a smart refrigerator was taken over by hackers and used to send spam[13], and that in 2019 the FBI's Oregon office warned that hackers can take over the microphones and cameras in smart TVs and use them for surveillance.[14] More recently, New York Times cybersecurity reporter Nicole Perlroth published the bestseller *This is How They Tell Me The World Ends* which details decades of offensive hacking efforts by China, Iran, Israel, North Korea, Russia and the U.S. to access information and booby-trap information protection systems.[15]

Peter Swire, who served under two presidential administrations and was responsible for reviewing intelligence community activities after the Snowden documents were dumped, argues that we live in "The Golden Age of Surveillance"[16] Not only do nation states like China, Russia and the U.S. have well-funded institutions with technically-gifted employees searching for new ways to monitor, but important other factors have also begun to enhance surveillance powers.

As information traverses the Internet, operators of servers can log *metadata* about activity. Perhaps because the content/metadata distinction was in part driven from the days when a telephone's content was recorded with a pair of alligator clips onto a reel-to-reel tape recorder and metadata was captured with a *dialed number recorder* that literally recovered the numbers that a person dialed *and nothing else*), U.S. law currently makes it much easier for law enforcement to obtain metadata than content.

Metadata is commonly believed to be less sensitive than content. However, there is a good argument to be made that metadata is more revealing than content. Metadata is easier to structure in computer databases and analyze. Consider the act of watching and interacting with a YouTube video. The *content* of the session includes:

- The visual content of the video, including the individual frames, the images of the people in the frames, the images of the buildings, etc.
- The audio content of the video, including the sounds, music, and other information.
- The text of any comments left on the video.

But if you were an analyst, consider the knowledge that could be derived from the same video's metadata:

- The video's unique identifier and it's title.
- The time that the video was recorded, uploaded and edited.
- The unique identifiers of each person that watched the video, their geographic location, their internet protocol (IP) address, and the time that it was watched.
- Whether the viewers clicked "thumbs up" or "thumbs down" on the video.
- Whether the viewers shared the video with friends and, if so, whom.

[13] Starr, "Fridge caught sending spam emails in botnet attack" (2014).
[14] Steele, *Oregon FBI Tech Tuesday: Securing Smart TVs* (2019).
[15] Perlroth, *This Is How They Tell Me the World Ends: The Cyberweapons Arms Race* (2021).
[16] Swire, *The Golden Age of Surveillance* (2015).

- The identifiers of any individuals in the video found with face recognition software.

The additional information available from metadata—particularly surrounding the identity of the community of users interested in the video and the people to whom they send it, might be far more important than the video's actual content.

The lines between content and metadata are not sharp. A transcript of the video might be considered content, but keywords extracted from the transcript might be considered metadata. While we classify the comments as content, the timings between individual keystrokes when the comments were left might be considered metadata—even if software can recover the actual typed words using those timings.

Metadata can thus indicate location, the identities of friends, and provide many hints about the content of communications and actual activities online. In many cases, the metadata/content distinction is functionally irrelevant, because operators of servers and services directly examine the content of our email, photographs, and other communications in the dual interests of security (anti-spam) and commercialization (behavioral-based advertising). The private sector plays a critical role by assembling dossiers of both proprietary company data and open source information on people; such products can then be sold to both marketers and (even foreign) government agencies.

The move to the "cloud" means that governments can obtain troves of data about people by through legal process (or simply by guessing or otherwise obtaining the user's password) and accessing a trove of information that was previously confined to the home or a business. Individual users of technology also contribute by documenting their lives on social networks, and by carrying mobile trackers dutifully storing contact books in them, which give companies and intelligence agencies alike access to location data and fodder for link analysis.

As much as technological trends have benefited nation states, these capabilities have devolved to many private sector actors as well.[17]

Especially concerning to some is the use of state collection capabilities to support domestic industries and silence critics living abroad. In the 1990s, for example, France was accused of using its intelligence apparatus to spy against Boeing, Textron and Bell.[18] More recently businesses have raised concerns about intellectual property exfiltration by China, which then shares the information with its their commercial rivals in China. Businesses are concerned about China and other nations using a range of surveillance capabilities to collect information on dissidents, regime critics and refugees who live outside of the country. For example, in 2010 Google revealed that its Gmail system had been hacked by China and that information from the e-mail accounts of human rights activists had been pilfered.[19] Businesses are also concerned about the convergence of organized crime and government in Russia, which not only directly engages in financial fraud but also creates platforms and even a market for others to do so.[20]

[17] Weinbaum et al., *SIGINT for anyone : the growing availability of signals intelligence in the public domain* (2017).

[18] Doyle, "Business spy war erupts between US and France: Paris forced to come clean on hi-tech dirty tricks, writes Leonard Doyle, West Europe Editor" (1993); Greve, "Boeing Called A Target Of French Spy Effort" (1993).

[19] Zetter, "Google to Stop Censoring Search Results in China After Hack Attack" (2018).

[20] OCCRP, *The Russian Laundromat Exposed* (2017); Bureau for Africa, *Government Complicity*

Is your email encrypted?

Much email sent today is between two Gmail users. These messages are encrypted by the Transport Layer Security (TLS) as they travel from the sender's web browser to Google's web-mail service. Although the messages are not encrypted in the memory of Google's servers, they are encrypted when they are written to Google's disks where the messages are stored.Google LLC, *Encryption at Rest* (2021) Likewise, the email messages are encrypted when they are sent from Google's servers to the Gmail recipient.

Mail that gets sent from Gmail to other mail providers, such as Microsoft's Office 365 cloud platform, are frequently encrypted using the SMTP STARTTLS protocol Rose et al., *Trustworthy Email* (2019).

This kind of protection is not as strong as the so-called *end-to-end* encryption offered by the S/MIME and PGP encryption systems. However, it is significantly easier to use because each user does not need to create or otherwise obtain a public/private keypair.

7.2.2 The Golden Age of Encryption

The Golden Age of Surveillance is accompanied by a corresponding golden age of encryption *adoption by default*. Since 1991, users with significant technical ability have been able to use strong encryption in the form of Phil Zimmerman's Pretty Good Privacy,[21] although even later versions that were heralded as being easy to use were still too difficult for most people.[22] Since then, technologists have sought to change the security landscape by implementing encryption by default in seamless ways. Perhaps most notable is the shift of addresses on the World Wide Web from being prefixed by `http://` to `https://`, which seamlessly provides users greater confidentiality and integrity in their web browsing. Prior to this change, users' web browsing was sent over the Internet without encryption, allowing adversaries and telecommunications providers alike to monitor users' website visits or even change the content of web pages as they were being viewed.[23] Email likewise has moved from communications where most messages sent over the Internet backbone were sent entirely in plain-text to a system where such messages are largely encrypted (although email encryption is not generally end-to-end—see the **sidebar "Is your email encrypted?"**). Likewise, the popular messaging app WhatsApp offers end-to-end encryption. When WhatsApp was acquired by Facebook, the creators left and created Signal, another messaging application offering end-to-end encryption. Likewise, Apple's iPhone and its newest laptops and desktops use encryption for storage and for text messages sent between Apple users. Although such techniques can be defeated through the use of so-called 0-day attacks,[24] companies like Apple are typically quick to fix such vulnerabilities when they become public.

Central to this rise in encryption is that the user need not understand, configure,

in Organized Crime (2019).

[21] Garfinkel, *PGP: Pretty Good Privacy* (1994).

[22] Whitten and Tygar, "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0" (1999).

[23] The advent of free encryption certificate services and a policy from Google that sites with TLS would get higher rankings in search results caused a rush to adopt the `https://` prefix.

[24] Perlroth, *This Is How They Tell Me the World Ends: The Cyberweapons Arms Race* (2021).

or even activate it because encryption is on by default. This offers a lesson for the confidentiality and integrity gains possible in quantum communications: for these innovations to be realized, they must not only be easy to use, they must be secure and integrated into the fabric of communications systems and consumer-facing applications.

7.3 Quantum Random Number Generation (QRNG)

All of these encryption systems we discussed in the last section are based on more-or-less the same technology stack: the AES encryption algorithm to encrypt the messages, a secure random number generator to create the AES key, and public key cryptography to get the per-message key from the message sender to the recipient. Earlier in this book we discussed the role of the AES and public key cryptography algorithms. In this section we will discuss the role of random numbers.

Cryptography depends on strong random numbers. For instance, a RSA-2048 key is generated from prime numbers that are over 300 digits long: these prime numbers are found by guessing random numbers and checking them to see if they are prime. (Unlike factoring, there are mathematical tricks that are used to rapidly determine if a number is prime or not.) Likewise, the AES-256 keys are themselves random numbers.

Random numbers thus form the very basis of the security provided by encryption. If a 256-bit key is random, then that means every key is equally probable. But if an attacker can somehow interfere with the randomness of the number generation process, it can dramatically reduce the possible number of encryption keys. For such an attack, the strength of AES-256 with a key that is not very random might not be strong at all.

Modern computers generate random numbers by using an initial *random seed* which is then used with a deterministic random bit generator, also called a pseudo-random number generator (PRNG). Typically, the random seed is created by combining many events that, if not completely random, are at least unpredictable. For example, the early PGP program instructed users to type on the keyboard and used the inter-character timing as a source of randomness. Other sources of randomness include the arrival time of packets at a network interface, inputs to digital cameras, and even seismic sensors. In practice, the quality of random numbers is determined by the samples taken from the "random" source, the quality of the mixing, and the quality of the PRNG. If any of these produce output that is somewhat predictable, or for which there is correlation between successive values, then a knowledgeable adversary can gain advantage when attempting to decrypting a message that was encrypted with such "poor quality" randomness.

Concerns about the strength of random number generates has been raised many times in the past. One such case from the U.S. involves the Dual Elliptic Curve Deterministic Random Bit Generator (Dual_EC_DRBG)[25]. When Dual_EC_DRBG was proposed, security professional Bruce Schneier and others raised concerns that the algorithm might include a "secret backdoor" that would allow the U.S. government to predict the algorithm's "random" outputs.[26]. These concerns were con-

[25] Barker and Kelsey, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators (Revised)* (2007).

[26] Schneier, "Did NSA Put a Secret Backdoor in New Encryption Standard?" (2007).

The NIST Randomness Beacon

In 2013, the U.S. National Institute for Standards and Technology deployed announced its "Randomness Beacon," a web-based service that posted random numbers in blocks of 512 bits every minute. Like an electronic lottery machine, the bits posted to the NIST website are unpredictable.

The randomness service is an endless source of numbers that can be used in situations where a random choice needs to be made, and the person making the choice wants to demonstrate that they made the choice fairly. In football games, for example, the receiving team is chosen by a coin toss—but how do we know the coin is fair? Or consider the customer of a brewery who wants to test bottles of beer: if the customer simply opens and inspects every 1000th bottle, the brewery can predict which bottles will be inspected and make sure that every 1000th bottle is one of their best. But if the customer is allowed to choose which bottles to inspect, the brewery might allege that the customer is intentionally picking bottles that look bad with the intent of arguing for a lower price. In these and similar situations where a decision must be made on a random choice, the NIST service can be relied upon by both parties to ensure a selection that is unbiased. In our case, the customer and the brewery can agree to select the bottles specified by the Randomness Beacon.

Example applications that NIST proposed included selection for random screening at security checkpoints, selection of test and control groups in scientific trials, selection of people for random tax audits, assignment of judges to cases, and so forth. Because the beacon is public, and because each bitsream is added to a hash chain (or blockchain), the system can be audited by any party. Of course, being public comes with a risk as well: the bits should not be used in cases were both randomness and secrecy are required. To drive in this lesson, the NIST website states:[a]

WARNING:
DO NOT USE BEACON GENERATED VALUES
AS SECRET CRYPTOGRAPHIC KEYS.

[a] See https://beacon.nist.gov/home

firmed in 2013[27]. Following the disclosure, NIST issued guidance stating "NIST strongly recommends that, pending the resolution of the security concerns and the re-issuance of SP 800-90A, the Dual_EC_DRBG, as specified in the January 2012 version of SP 800-90A, no longer be used."[28] In 2015, the Director of Research at the National Security Agency said that the agency's "failure to drop support for the Dual_EC_DRBG" after vulnerabilities were identified in 2007 was "regrettable."[29].[30]

[27] Perlroth, "Government Announces Steps to Restore Confidence on Encryption Standards" (2013).

[28] Information Technology Laboratory, *Supplemental ITL Bulletin for September 2013* (2013).

[29] Wertheimer, "Encryption and the NSA Role in International Standards" (2015).

[30] This story and others surrounding the quest to produce high-quality random numbers at scale is discussed in Garfinkel and Leclerc, "Randomness Concerns When Deploying Differential Privacy" (2020), from which this story and its references are taken.

In 2019 cryptographers stated that two Russian-designed encryption systems, Streebog and Kuznyechik, might also contain a secret backdoor that would give an advantage to a knowledgeable attacker trying to decrypt a message protected with the algorithm. In this case, the weakness was not in the random number generator, but in the algorithms' so-called "substitution boxes."[31]

Quantum states provide the best source for strong, unbiased randomness. Scientists have developed several different methods to derive strong randomness from quantum events, including the path that photons take when light is split, the polarization of individual photons, and the phase of quantum states and processes.[32] A notional device bears similarity to the dual-slit experiment discussed in Section B.1.3, "Light: it acts like a wave" (p. 355). The device works by cycling a particle or photon in and out of superposition. Measurement disturbs the superposition, causing decoherence and the production of a random bit. That bit is then used as a basis to generate random numbers. One way to think of these machines is as quantum computer with a single qubit that is constantly computing the answer to the question "is the qubit 0 or 1?"

Number generation in such a scheme faces two sets of challenges. The first is the cycle speed of the prepare-superposition process and the speed of the measurement-decoherence process, which together determines how fast these systems can produce random bits. These machines may also be impact by errors produced by classical noise and the reliability and tolerances of the quantum source and of the measurement mechanism, which can bias the results.

Properly implemented, QRNG produces strong randomness.[33] In fact, it probably produces the strongest possible random numbers, since modern physics holds that quantum processes are the ultimate source of all non-determinism that we observe in the universe. QRNG has also been commercially available for years. In fact, after scientists created a QRNG system at the Australian National University in 2011,[34] the investigators found they had more random numbers than they would ever need for experiments. So they created a free QRNG service on the web.[35] In 2020, IBM and Cambridge Quantum Computing offered QRNG as a cloud service. And NIST is deploying Entropy as a Service (EaaS), a public, quantum-based source of random numbers.

Using these remote, cloud-based services requires some reliance on the provider, but there are measures that can be taken to reduce the risk. Instead of using the source directly, it can be combined with a secret key and then used in a cryptographically strong PRNG—a CSPRNG! This approach works as long as the secret key is kept secret and as long the PRNG is really a CSPRNG. That's the use case that NIST envisions for its EaaS. The EaaS project is explicitly designed to serve Internet of Things (IoT) devices by providing random numbers that these devices can use to create strong encryption keys. The idea is that IoT devices will be small and inexpensive, so much so that even high-end brands will cut corners on security,

[31] Perrin, *Partitions in the S-Box of Streebog and Kuznyechik* (2019).

[32] Ma et al., "Quantum random number generation" (2016).

[33] Acin and Masanes, "Certified randomness in quantum physics" (2016); Bierhorst et al., *Experimentally generated randomness certified by the impossibility of superluminal signals* (2018).

[34] Symul, Assad, and Lam, "Real time demonstration of high bitrate quantum random number generation with coherent laser light" (2011).

[35] See https://qrng.anu.edu.au/

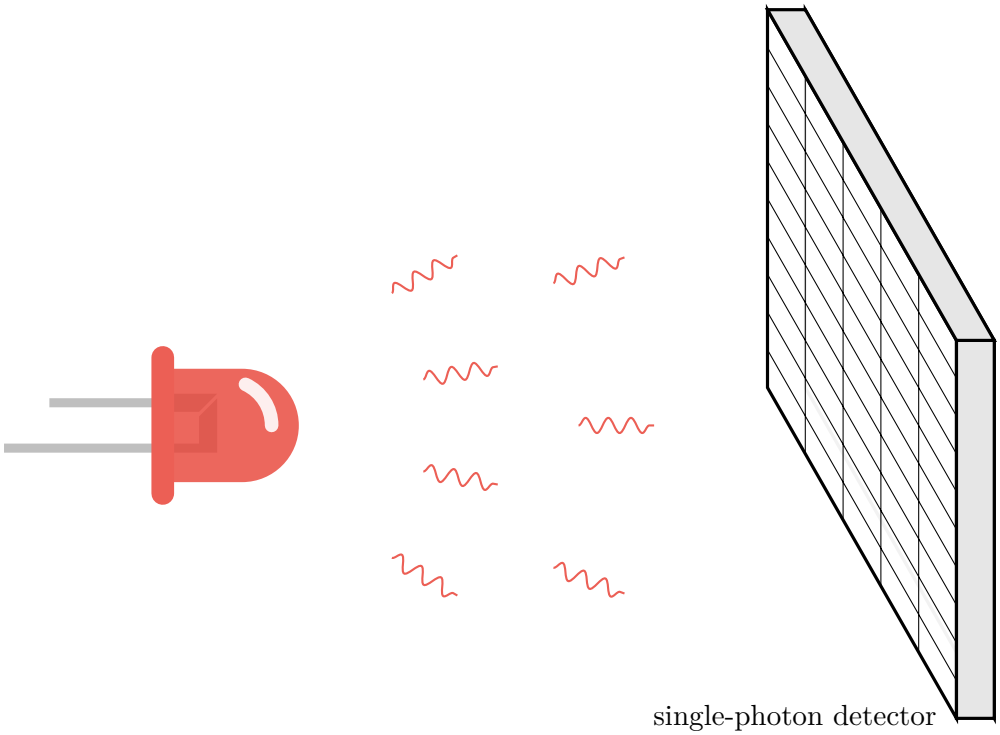


Figure 7.1: A mechanism for QRNG designed by ID Quantique fits into a mobile phone handset and pairs an LED and single-photon sensor array to derive randomness from photonic noise.

thus the chances that the market will produce QRNG for IoT devices is particularly unlikely. NIST is in effect substituting the market with security fundamentals for anyone to use. NIST is also upgrading its Randomness Beacon to use QRNG, as currently, it uses two classical generators to prevent guile.

Higher levels of assurance require implementing the QRNG locally, so that the high-quality random bits are generated where they are needed, and not by some third party. For instance, ID Quantique has long sold QRNG hardware that plugs into a standard personal computer or server. In 2020, the company announced a QRNG chip that could fit into mobile phone handsets.[36] This device uses the random "shot noise" from a light-emitting diode (LED) to generate numbers. Every time the LED fires, the number of photons emitted fluctuates randomly. A CMOS sensor array sensitive to single-photon events detects the number emitted and their positions. Random numbers are derived from the shot noise detection process, see Figure 7.1.

7.4 Quantum Key Distribution

When Rivest, Shamir and Adleman wrote their article introducing the RSA encryption system, they explained it with a woman "Alice" who wanted to send a secret

[36] Quantique, *Quantis QRNG Chip* (2020).

message to a man named "Bob."[37] Since then, Alice, Bob and a whole cast of other characters have been used to help scientists analyze and explain security protocols. There is Eve, the eavesdropper, who attempts to "intercept" (a strained metaphor) this conversation. And there is Mallory, a malicious attacker, who can modify the message or inject new ones.

Quantum Key Distribution (QKD) describes an approach where Alice and Bob can exchange an encryption key guaranteed to enjoy *unconditional* security. No computer available today or in the future can compromise this system, because the attacker does not have enough information to make sense of the ciphertext. Such systems are information theoretic secure.

Information theoretic approaches differs from the conditional, computationally secure approaches used today. Today's approaches depend on processes like large prime number factoring, which modern computers cannot do quickly. Security of today's systems are thus *conditional* on two assumptions: first, that factoring will stay hard, and that some clever person will not discover a way to factor more quickly using a conventional computer. Second, that factoring will remain computationally intractable, such that it is not possible to combine enough computers to solve these hard problems. If some revolution in engineering produced dramatically faster classical computers, these could be then be tasked with factoring numbers quickly using existing algorithms.

7.4.1 BB84

In 1984, Charles Bennett and Giles Brassard published the BB84 protocol, demonstrating how Alice and Bob could exchange encryption keys using quantum states.[38] Using the protocol, Alice and Bob get the same stream of `0` and `1` bits that they can use for any purpose. For example, they can use the sequence in 8-bit chunks as a *one-time pad* (see Figure 7.2), using each group of 8 bits to encrypt the next byte of the message. Alternatively, they the sequence in 256-bit chunks as AES-256 encryption keys.

The one-time pad is the gold standard for communications security because it is information-theoretic secure.[39] Even if the attacker tries every possible key, there is not enough information in the encrypted message to distinguish a correctly decrypted message from an incorrectly decrypted message. The reason is that the key is as long as the message thus every possible key makes the message decrypt a different way. This means that trying every possible key makes the encrypted message decrypt to every possible message.

One-time pads are the stuff of spy thrillers and history books, but they are not used much today because it is too difficult to distribute the pads in advance and then assure that each is used just once. The Soviet Union attempted to use one-time pads for its diplomatic communications after World War 2 and it failed; the NSA revealed its success in cracking the Soviet codes in 1995 (see Figure 7.4.4).[40].

[37] Rivest, Shamir, and Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems" (1978a).

[38] Bennett and Brassard, "Quantum cryptography: Public key distribution and coin tossing" (1984).

[39] Shannon, *Communication theory of secrecy systems* (1949).

[40] National Security Agency Central Security Service, *VENONA* (2021).

BB84 is revolutionary, because Bennett and Brassard's approach deals with two central challenges in communication: how to generate a secure, shared secret, and how to distribute it at a distance. Two other key challenges—usability and the time it takes to generate and transmit the key securely—are up to the companies that create applications using QKD protocols.

However, modern QKD systems cannot generate a stream of bits fast enough to encrypt modern data links. For this reason, QKD systems typically operate in a slightly less secure mode in which BB84 is used to exchange 256-bit encryption keys which are then used with conventional encryption algorithms such as AES-256. With a 256-bit key, each encrypted message will have only 2^{256} possible decryptions, and the likelihood is that all but one of them will be gibberish. As we discussed in Chapter 5, it isn't possible to try all 2^{256} keys, so using BB84 to exchange AES-256 keys is considered secure. However, it is only computationally secure, not information-theoretic secure. As a compromise, these systems might change their AES-256 keys every few seconds, to minimize the amount of ciphertext that has been encrypted with any given AES-256 key.

7.4.2 How QKD Works

Most QKD systems are based on the idea of sending a stream of photons from a sender (Alice) to a recipient (Bob). For more background on polarized light, see Appendix B.2.1, "Light: It's either polarized up-and-down, or it's not!,".

Here we provide a simplified explanation for how BB84 operates. The first thing to know is that actually using BB84 in a production system requires considerable mastery of the quantum realm and engineering cleverness not explained here.

In modern QKD systems, the photons either travel down a fiber optic strand, or they are created in pairs in a satellite and sent to two independent ground stations.[41] In the first case, Alice prepares a stream of photons by sending each through a polarizing filter that is either polarized horizontally (H), vertically (V), at a 45°angle, or at a 135°angle. Alice makes this choice at random, recording both the number of the photon and the orientation of her polarizing filter. Sending with a H or a 45°is tentatively sending a `0`, while sending with a V or a 135°is tentatively sending a `1`. (Alice can't actually number each photon, so instead she will encode each photon's value in the light stream itself.)

Let's say Alice sends 10 photons:

Photon #	Alice Filter orientation	Tentative bit
0	45°	0
1	45°	0
2	45°	0
3	H	0
4	V	1
5	135°	1
6	45°	0
7	45°	0
8	H	0
9	135°	1

[41] The protocol involving a pair of entangled photons is called E91, after its inventor Artur Ekert.Ekert, "Quantum cryptography based on Bell's theorem" (1991)

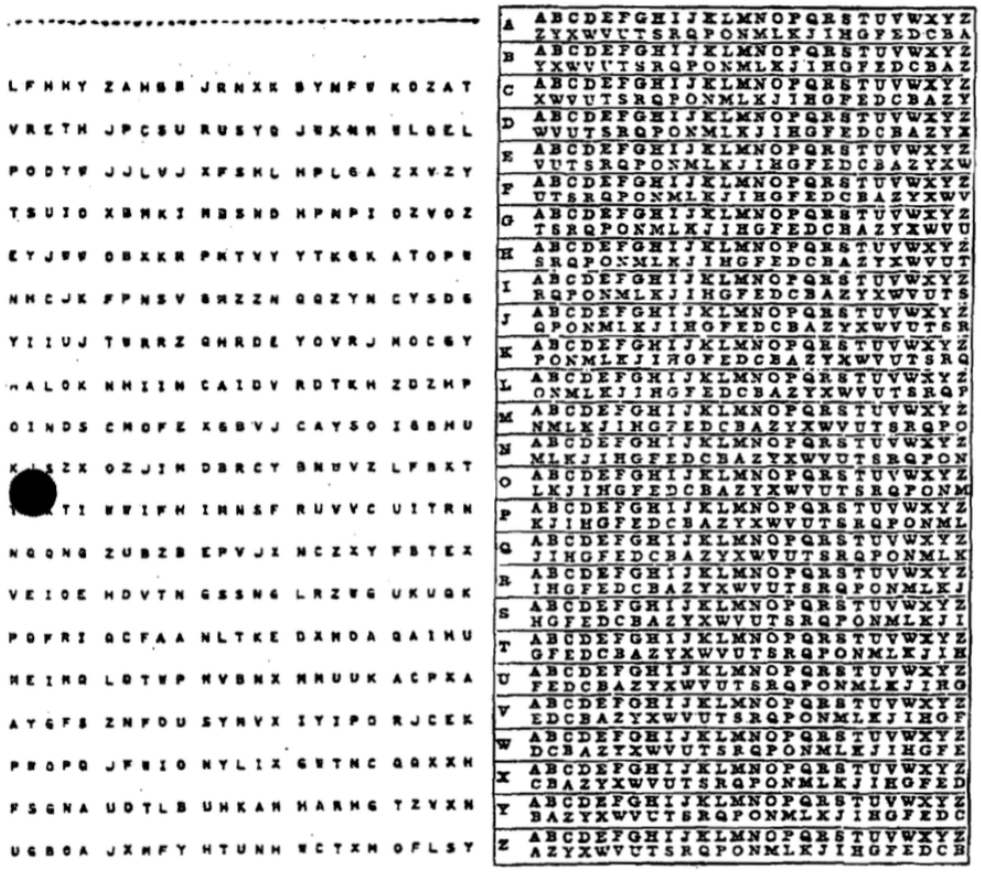


Figure 7.2: This table from the NSA's DIANA program illustrates how one-time pads produce messages with keys the same length of ciphertext. The key is on the left hand side. The right hand side is the table used to convert plain text to ciphertext (and vice versa). This key starts with the letter "L," thus, the user encrypting a message would use the L row on the table to choose the first letter of ciphertext. Assume that Alice wants to say "The Magic Words are Squeamish Ossifrage" to Bob. To encrypt, Alice notes the first letter from the key, left hand pane, which is L. Turning to the table, row L, and then to the letter T, the corresponding ciphertext underneath the T is a V. To encrypt the next letter, Alice would then use F from the key to locate the letter H and choose the ciphertext N, and so on. Alice must destroy her card when she is finished encrypting. Bob would have an identical card, and he must destroy his card when he finishes decrypting.

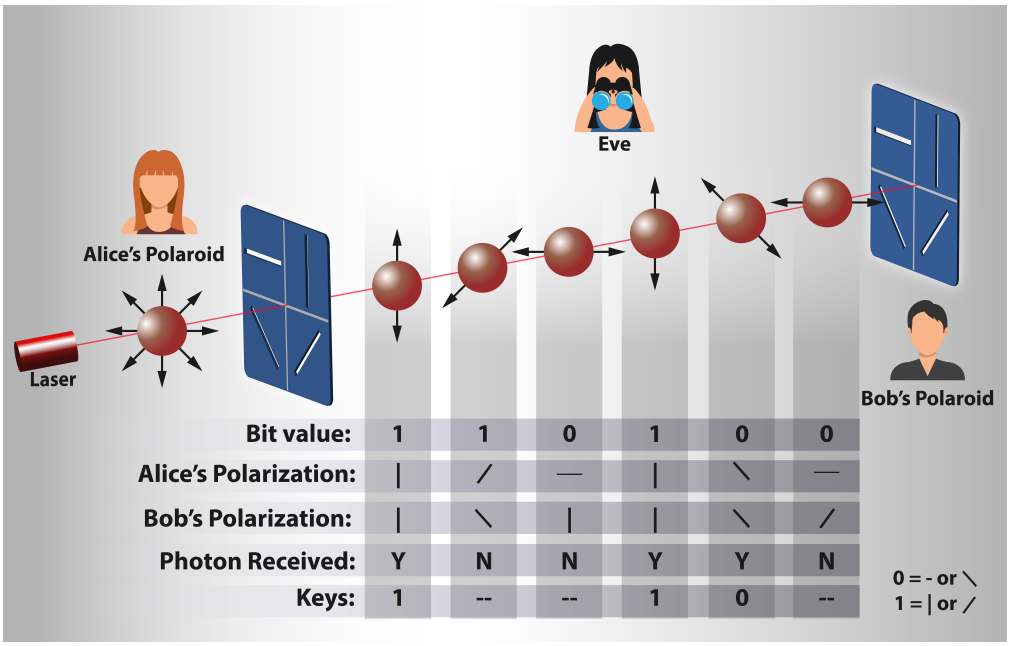


Figure 7.3: The BB84 protocol illustrated. Adapted from Aliberti and Bruen by Twitter user farooqumer89

When Bob receives the photons, he also passes them through a filter that is also randomly oriented at either V or at 135[a]. He then measures the presence or absence of the photon with a single photon detector:

Photon #	Bob Filter orientation	Photon detected?	tentative bit
0	135°	NO	0
1	135°	NO	0
2	V	YES	1
3	V	NO	0
4	V	YES	1
5	V	YES	1
6	135°	NO	0
7	V	NO	0
8	135°	NO	0
9	V	YES	1

Now Alice and Bob need to compare notes to see if the measurement that Bob made of the photon was compatible with the photon that Alice prepared and sent. If Bob measured with his V filter, then he will detect light if Alice sent the light with her V filter, but not if she used her H filter. But if Alice sent with her 45[a] or 135[a] filters, the measurement that Bob made is meaningless: there's a 50-50 chance that a photon polarized with the 45[a] filter will pass through a V filter.

To compare notes, Bob can reveal which filter he used to measure each photon. Alice then tells Bob which of his measurements he should keep and which he should throw out.

Photon #	Bob to Alice	Alice to Bob
0	135°	KEEP
1	135°	KEEP
2	V	–
3	V	KEEP
4	V	KEEP
5	V	–
6	135°	KEEP
7	V	–
8	135°	–
9	V	–

At this point, Alice and Bob knows that photons 0, 1, 3, 4 and 6 were sent and received with compatible polarizing filters. Alice looks at her table and discovers that the tentative bits corresponding to those numbers are `0 0 0 1 0`. Bob looks at his table and gets the same sequence of bits.

To determine that the system is operating properly, Alice and Bob can now decide to reveal every even bit of the resulting sequence. Alice says that even bits are `0`, `0` and `0`. Bob notes that his are the same. Alice and Bob then use the remaining bits (`0 1`) as their secret key.

If Alice and Bob do not reveal to each other the same bits, then either the system is not operating properly, or else an attacker is intercepting the beam and injecting a photon sequence of their own. In either case, Alice and Bob know not to use that key.

Because of measurement error, the sequence of bits that Alice and Bob recover are not exactly the same. A variety of error correction techniques exist that can be used to account for these errors, at the cost of using even more bits.

The two photon system is similar, except that a pair of entangled photons are sent from the satellite to both Alice and Bob, who then both measure the polarization and compare notes. In this design, the satellite cannot determine the key that Alice and Bob agree upon, nor can anything else in the universe: each photon can only be measured once. Of course, once Alice and Bob agree upon a key, a suitably skillful attacker might be able to steal it from either Alice or Bob if their QKD device does not properly protect the key after it has been created.

7.4.3 Why QKD is Secure

What makes QKD secure is the fact that the actions of Alice and Bob measuring the photon are independent, but the measurements are correlated *if and only if Alice and Bob choose compatible measurements.* If Alice measures the photon with a horizontal polarizing filter and Bob uses a filter that is polarized vertically, their measured results are linked and they have now agreed on a common bit. But if Bob uses a filter at 45°, the measures are incompatible and there is no correlation between them. This is the essence of Einstein's "spooky action at a distance," the paradox of entanglement. Because Alice and Bob chose their measurements at random, only 50% of them will be compatible: the remaining measurements will be thrown out.

Now let's say an attacker, Eve, tries to crash the party. Eve attempts the well-known "man-in-the-middle" attack: she catches the photons headed for Bob, measures them, and then prepares a new photon and sends it to Bob. Can Eve get

Quantum Computing and Bitcoin

Cryptocurrencies such as Bitcoin are speculative investment and value transfer mechanisms that are based on a *distributed ledger*, a kind of shared database, that is difficult to corrupt. BitCoin, the first cryptocurrency, relies on SHA-256 to build its ledger.

The Bitcoin ledger consists of many transactions, each of which is basically an electronic check that is signed with a private key. The check transfers some amount of Bitcoin from the user's corresponding public key (A bitcoin "address") to another public key. These transactions are grouped into blocks. In addition to these electronic checks, each block contains the hash of the previous block, a signature by the block's "miner," and block of random values placed there by the miner. The random values are manipulated such that the SHA-256 hash of the new block begins with a large number of zeros. To create a block that has a SHA-256 hash that begins with a large number of zeros, the Bitcoin "miner" takes the block of transactions and makes systematic changes to that random block until the hash has the correct form.

Because the hashes generated by SHA-256 appear random, with each bit having an equal chance of being a `0` or a `1`, finding hashes with a large number of leading zeros is computationally intensive. In March 2020, Bitcoin blocks had 76 leading binary `0`s, followed by 180 bits of `0`s and `1`s; the number leading `0`s is automatically adjusted to be longer and longer as more and faster Bitcoin miners join the network; each additional leading `0` requires roughly twice as much computational power to find.

In 2019, the National Academies estimated that a quantum computer could attack BitCoin's ledger system, but to do so, one would have to have a quantum computer with 2 403 qubits and 180 000 years to run the requisite quantum circuit. Given that the ledger gets a new block every 10 minutes, attacking the ledger itself in order to obtain free BitCoin appears unlikely. Perhaps over time the attack time estimate will lower, as quantum computers get faster at running quantum circuits, or as scientists discover clever quantum implementation of SHA-256.

But this does not mean that BitCoin holders are safe from quantum computing attacks. In the nearer term, a quantum computer could be tasked cracking the public key of an individual's Bitcoin user's wallet. This would let the attacker transfer the Bitcoin user's money to another address. Although the Bitcoin user could publicly complain, there would be no recourse, and other Bitcoin users would probably suspect that the cracked key had actually been stolen using traditional cyber approaches: breaking into the user's computer and stealing the private key.

Quantum Money

It was Stephen Wiesner's idea of using the entanglement of two particles to create unforgeable banknotes (see p. 102) that led Bennett and Brassard to come up with the idea of quantum cryptography in first place. Since then, many scientists have proposed systems that rely on quantum effects to store and transmit value, now broadly called *quantum money*. These schemes vary from implementation. Some provide information-theoretic security while others rely on public key systems.[a] But given current constraints in quantum memory, computing, and networking, hopes for quantum money systems are far off.

If they ever do arrive, some of the affordances promised will be contested by parties with interests in transactions. Crypto-currencies like Bitcoin and most if not all envisioned quantum currencies contain mechanisms to ensure that a purchaser actually has sufficient funds and to prevent "double spending." Beyond that, however, most of these mathematical monies are quite spartan.

Conventional value transfer mechanisms such as check, bank checks, bank wires, automated clearing house (ACH) and others are complex for many reasons. For instance, policy decisions must be made to reconcile the the different, conflicting interests held by ordinary consumers, merchants, banks, and governments in the governance of value transfer systems. A consumer might want the ability to repudiate a value transfer, in case of fraud, coercion, or perhaps even because of poor-quality goods received while merchants might want to block repudiation. Governments typically want the ability to unmask all parties in a transaction. Such mechanisms are missing—intentionally—from crypto-currencies like Bitcoin.

Crypto-libertarians, in their efforts to evade social contract and taxes, might want anonymous forms of value transfer, while governments will seek to prohibit this secrecy. Governments more broadly are concerned about counterfeiting and even the risk that a foreign adversary might circulate false bills in order to destabilize an economy. Most participants presumably want to minimize fraud and guile, but safety interests might compete with usability, universality of payment acceptance, and the speed of transactions.

The tensions of these requirements are reflected in regulation and custom surrounding the acceptance of cash, checks, and credit cards. In fact, as Bitcoin has become more mainstream, the original vision of a bank-free, anonymous, peer-to-peer payment system has ceded to something more akin to a commodities market, one mediated by exchanges that are regulated by governments and that follow taxation and anti-money-laundering rules to identify market participants.

[a] Hull et al., "Quantum Technology for Economists" (2020).

away with this deception? In a properly implemented QKD system, the answer is "no." That's because when Eve receives, measures, and retransmits the photon, she doesn't know how Bob is going to measure it. By chance, she will only measure the photon in a compatible manner 50% of the time. The other 50% of the time, she will measure the photon in a way that is incompatible. When she sends each of those incorrectly measured photons to Bob, Eve has a 50% chance of sending them in the correct state, and 50% chance of sending them in the wrong state.

When Bob compares notes with Alice, they first reveal how the photons were measured and throw out the photons for which Alice's and Bob's measurements were incompatible. But after this step, they intentionally reveal a certain percentage of the remaining photons. When Bob and Alice discuss these intentionally revealed photons, they will discover that their measurements disagree roughly half of the time. This indicates either that their equipment is not working properly, or that Eve is attempting to perform a man-in-the-middle attack.

Of course, Eve could go further, and pretend to be Bob to Alice and to be Alice to Bob. To prevent this, Alice and Bob need to have a way of authenticating the open messages that they send to each other. Today the easiest way to do this authentication is with public key cryptography. This use of public key cryptography is considered acceptable in QKD systems, because even if an attacker records the authentication messages and cracks the private keys behind them at some point in the future, that won't change the fact that the messages were properly authenticated when they were sent. No secret information is revealed if the authentication keys are cracked in the future.

Eve can prevent Alice and Bob from communicating securely even if the duo use a QKD system. Eve could also use electronic warfare approaches. Eve could inject noise to deny or degrade the quantum channel and cause Alice and Bob to have to revert to other, less secure communication, but she can't decipher the messages that it sends. (Indeed, risks of denial of service is among the reasons the NSA has spurned QKD in favor of quantum-resistant (or post-quantum) cryptography.[42]) And once the key is exchanged between Alice and Bob, the duo do not need a "quantum internet" or quantum states to talk securely. Alice and Bob can use the quantum key to communicate on existing classical channels, encrypting their communications with a conventional quantum-resistant symmetric algorithm such as AES-256.

7.4.4 QKD Gains Momentum

Since BB84 was proposed, new protocols and even implementations have emerged. For instance, in 1991, Arthur Ekert proposed a protocol that relies on entanglement.[43] Alice and Bob receive correlated photons from a split-beam laser. Using Bell tests, (see Section B.4 (p. 373)), Alice and Bob compare the correlations of their photons to ensure that Eve has not intercepted them. Under Ekert's proposal, even if Eve is operating the laser, she cannot determine the states of Alice and Bob's photons without interfering with the Bell correlations, thus revealing her attack. Ekert's proposal thus anticipates the possibility of a QKD-as-a-service approach—a satellite delivering entangled photons from space to the ground, allowing any two

[42] National Security Agency, *Quantum Key Distribution (QKD) and Quantum Cryptography (QC)* (2020).

[43] Ekert, "Quantum cryptography based on Bell's theorem" (1991).

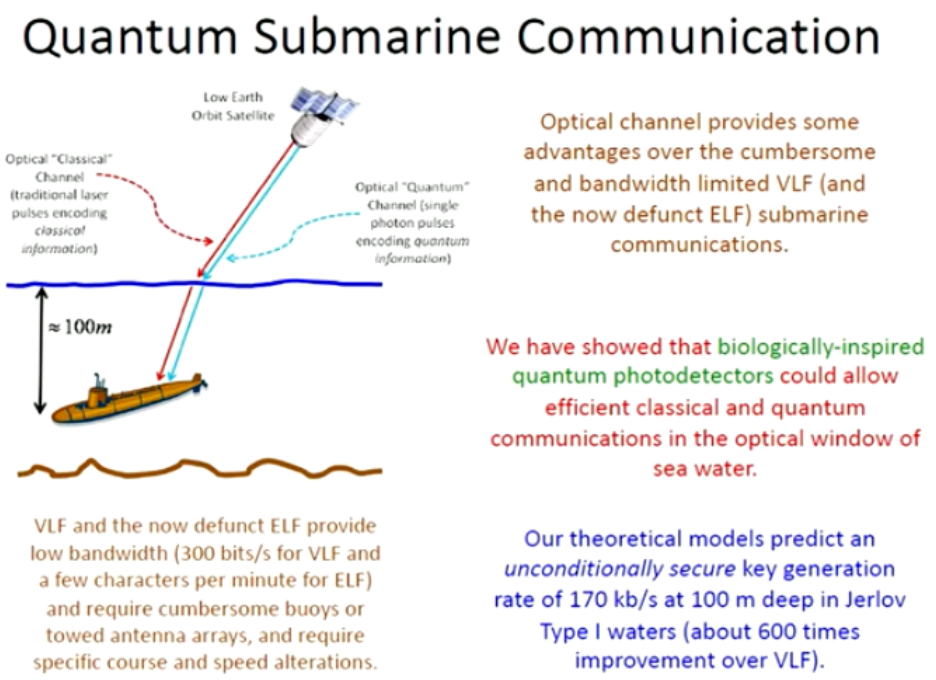


Figure 7.4: In a 2018 address to the National Academies, Dr. Marco Lanzagorta, explained how quantum communications might enable new forms of secure, satellite-to-submarine communication. Image courtesy U.S. Naval Research Laboratory.

parties to communicate securely, and not even the satellite can decipher their shared key.

Scientists have also proposed BB84 protocols to improve communications with satellites directly. In one scheme, a submarine equipped with a photosensor or towing a small buoy can exchange photons with a satellite, even while submerged (see Figure 7.4.3). The submarine would have to make speed versus depth tradeoffs, that is, at a depth of about 60 meters, data could be exchanged at 170 kilobits per second, but this drops in murky waters and at deeper levels. Nonetheless, the approach is stealthy and has advantages over existing submarine communication approaches.[44][45]

Long distance quantum channels for key distribution require special ingenuity to overcome a variety of technical challenges. Chinese scientists, led by that nation's "father of quantum," Jian-Wei Pan, demonstrated entanglement at 1,200 kilometers by using a satellite nicknamed Micius.[46] The satellite beamed photons between distant base stations what were in the coverage area of the Micius for just five minutes.[47] Pan's team pointed to the use of the entangled photons for an Ekert-protocol secure exchange, at a distance currently impossible to achieve with terrestrial, fiber

[44] Lanzagorta, *Envisioning the Future of Quantum Sensing and Communications* (2018).

[45] Lanzagorta, *Underwater communications* (2013).

[46] Launched in 2016 at the low-earth orbit of 500 km, Micius travels in a Sun-synchronous path. Micius is named for the Fifth Century BCE Chinese philosopher Mozi, founder of Moism, who wrote original works on optics.

[47] Yin et al., "Satellite-Based Entanglement Distribution over 1200 Kilometers" (2017).

optic connections (the quantum states degrade in the glass fiber after a distance of around 100 km without taking special measures). Yet, the approach still faces many challenges as revealed in the paper's methods. Pan's team had to beam millions of photons a second to maintain the link, and only a handful reached the base stations because of atmospheric and other interference.

Pan's demonstration is part of a $100 million project in China, the Quantum Experiments at Space Scale program (QuESS). The entangled distribution over such a great distance demonstrated a substantial goal of the program. Key exchange was realized later the same year, using a mixed fiber-optic/satellite path of over 7 000 km.[48] Pan's team demonstrated the key exchange by holding a videoconference between Beijing and Austria. However, this demonstration did not use end-to-end entanglement between Alice and Bob, as described by Ekert. In this initial experiment, Pan's team used the BB84 protocol, and the satellite operated as a trusted relay. Micius exchanged separate keys with each of the different ground stations.

With a relay, the implementation is not fully quantum—it's not a quantum internet—and the parties must trust the satellite's security. That's a concern. Governments will probably trust their own satellites, but this trust should not be absolute, as the computers in satellites are vulnerable to cyber attack just as computers down here on the ground. Nevertheless, the trusted repeater approach is likely to be operational before systems that provides end-to-end quantum security, for the simple reason that China has such a system today: in 2020, Pan's team announced a satellite-terrestrial quantum network covering 4,600 km. The network has over 150 users, and achieved a transfer rate of 47 kilobytes a second, more than sufficient for exchanging 256-bit AES keys.[49]

In the U.S., fewer than ten QKD networks have been implemented in recent years. The first, DARPA's QKD network, was implemented by Raytheon BBN, at Harvard and Boston Universities in 2003.[50] The team used dark fiber (unused fiber optic cables) in Cambridge, Massachusetts to connect the almost 30 km long network. The network, which had trusted optical point-to-point systems and untrusted, relaying infrastructure, operated for four years. Here "untrusted" means that the relaying infrastructure could not impact the security of the data sent over the fiber.

At Los Alamos National Laboratory, scientists created a hub-and-spoke quantum network.[51] In the implementation, a central, trusted server performs the key exchange, which then enables nodes in the spokes to communicate among each other with authenticated quantum encryption. This sort of trust model works when all of the networks have a some reason to trust the central node; in the LANL demonstration, their model was a power distribution network.

Major challenges still exist for QKD implementation. The point-to-point nature required to preserve quantum states between Alice and Bob makes QKD networks more like the early telegraph than the telephone or internet. Quantum states decohere in long fiber runs, thus some networks require repeating, which, like the Micius satellite demonstration, requires trusting the repeater. Alice and Bob also need so-

[48] Liao et al., "Satellite-Relayed Intercontinental Quantum Network" (2018).

[49] Chen et al., "An integrated space-to-ground quantum communication network over 4,600 kilometres" (2021).

[50] Elliott and Yeh, *DARPA Quantum Network Testbed* (2007).

[51] Hughes et al., *Network-Centric Quantum Communications with Application to Critical Infrastructure Protection* (2013).

Figure 7.5: In 2019, Air Force Research Laboratory scientists demonstrated daylight QKD using this rig at the Starfire Optical Range, located at Kirtland Air Force Base in Albuquerque, New Mexico. This is important because stray daylight entering the collector causes substantial noise that interferes with the measurement, limiting long-distance QKD during the daytime. (The Air Force's Directed Energy Directorate, which developers lasers and optics, was identified for transfer to the U.S. Space Force in 2020.) Image by U.S. Air Force photographer Todd Berenger.)

phisticated equipment: lasers, single-photon detectors, interferometers and the like. These are now packaged in commodity QKD systems that communicate over fiber optics, although systems that communicate in free space or using satellites are still basic science endeavors. Even so, QKD is among the most mature quantum technologies and solving these limitations is receiving significant attention. The next section turns to such commercialization.

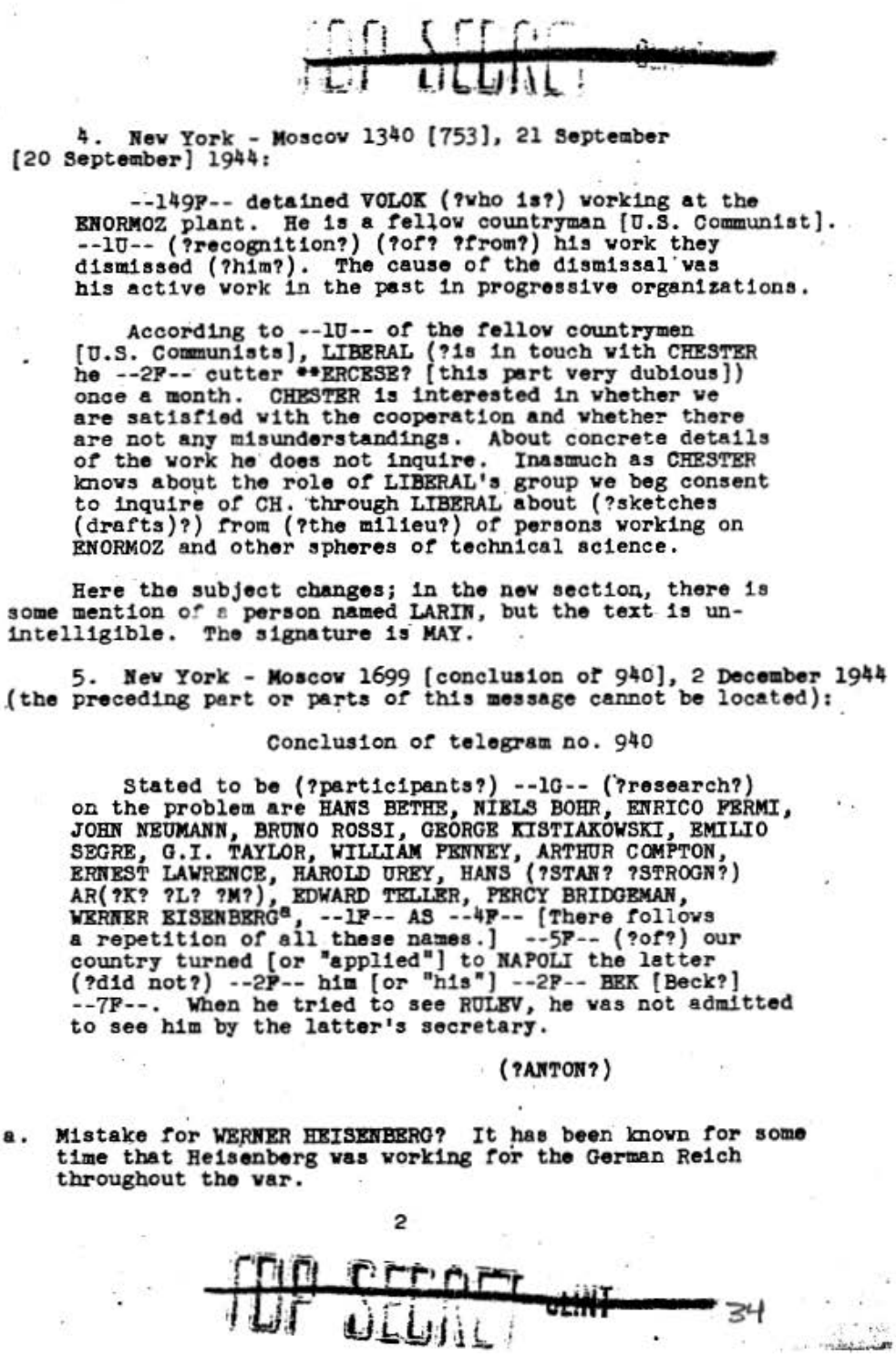
7.4.5 QKD Commercialized, Miniaturized

As early as 2009, three companies (ID Quantique, Switzerland; MagiQ Technologies, U.S; and Smartquantum, France) offered working QKD devices.[52] According to the Quantum Computing Report, at least a dozen private firms are working on QKD offerings, along with a few large public companies.[53]

Despite the growing competition in QKD, adoption of QKD has been weak. For starters, without large, encryption-breaking quantum computers, there is no demonstrated need for the technology. In 2015, an unclassified summary of U.S. Air Force advisory board report threw cold water on QKD, apparently finding that QKD significantly increases system complexity while providing "little advantage over

[52] Scarani et al., "The security of practical quantum key distribution" (2009).

[53] ArQit, InfiniQuant, KETS Quantum Security, Phase Space Computing, QEYnet, Qrate Quantum Communications, Quantropi, Quantum Xchange, Qubit Reset LLC, Quintessence Labs, QuNu Labs, SeQureNet and VeriQloud; larger firms include Nippon Telegraph and Telephone Corporation (NTT), Raytheon BBN Technologies and Toshiba.



4. New York - Moscow 1340 [753], 21 September [20 September] 1944:

--149F-- detained VOLOK (?who is?) working at the ENORMOZ plant. He is a fellow countryman [U.S. Communist]. --1U-- (?recognition?) (?of? ?from?) his work they dismissed (?him?). The cause of the dismissal was his active work in the past in progressive organizations.

According to --1U-- of the fellow countrymen [U.S. Communists], LIBERAL (?is in touch with CHESTER he --2F-- cutter **ERCESE? [this part very dubious]) once a month. CHESTER is interested in whether we are satisfied with the cooperation and whether there are not any misunderstandings. About concrete details of the work he does not inquire. Inasmuch as CHESTER knows about the role of LIBERAL's group we beg consent to inquire of CH. through LIBERAL about (?sketches (drafts)?) from (?the milieu?) of persons working on ENORMOZ and other spheres of technical science.

Here the subject changes; in the new section, there is some mention of a person named LARIN, but the text is unintelligible. The signature is MAY.

5. New York - Moscow 1699 [conclusion of 940], 2 December 1944 (the preceding part or parts of this message cannot be located):

Conclusion of telegram no. 940

Stated to be (?participants?) --1G-- (?research?) on the problem are HANS BETHE, NIELS BOHR, ENRICO FERMI, JOHN NEUMANN, BRUNO ROSSI, GEORGE KISTIAKOWSKI, EMILIO SEGRE, G.I. TAYLOR, WILLIAM PENNEY, ARTHUR COMPTON, ERNEST LAWRENCE, HAROLD UREY, HANS (?STAN? ?STROGN?) AR(?K? ?L? ?M?), EDWARD TELLER, PERCY BRIDGEMAN, WERNER EISENBERG[a], --1F-- AS --4F-- [There follows a repetition of all these names.] --5F-- (?of?) our country turned [or "applied"] to NAPOLI the latter (?did not?) --2F-- him [or "his"] --2F-- BEK [Beck?] --7F--. When he tried to see RULEV, he was not admitted to see him by the latter's secretary.

(?ANTON?)

a. Mistake for WERNER HEISENBERG? It has been known for some time that Heisenberg was working for the German Reich throughout the war.

2

34

Figure 7.6: Richard Hallock, an analyst at the U.S. Army's Signal Intelligence Service, discovered that Soviet spies had taken a major shortcut in the implementation of their cryptosystem—they were reusing portions of one-time pads. The revelation allowed the agency, a forerunner to the National Security Agency, to decrypt important Soviet communications. This summary of intercepted communications, partially decoded, shows that the Soviets had identified the main scientists involved in the Manhattan Project (the Soviet cryptonym for it was ENORMOZ; LIBERAL is Julius Rosenberg). The American analysts also ponder whether the Russians thought that Werner Heisenberg was working on the American fission project, alas he was working for the Germans. The decryption project, code name VENONA, ran from 1943 through 1980; it was revealed by the U.S. National Security Agency in 1995.

the best classical alternatives."[54] The USAF's full report is not publicly available, but perhaps the board meant that as system complexity increases, so do attack surfaces. A more complex system gives attackers more opportunities to interfere with communications, and perhaps the side channel attacks possible on quantum devices will be more difficult for network operators to understand. Aside from device problems, there remains the old problem that users can be fooled into granting access. Perhaps the USAF report's skepticism reflects that the U.S. government has a decades-old system of using trusted human couriers to transport high-value key material.

In October 2020, the NSA released a statement clarifying that it would not use QKD to secure the classified and sensitive-level networks it is responsible for protecting, and this NSA statement articulated the likely reasons why QKD has not been more commercially successful. Calling out the hype, the NSA statement recognized that QKD advocates "occasionally state bold claims based on theory" but that in reality, the technology is "highly implementation-dependent rather than assured by laws of physics." The NSA's specific objections related to the need to install new, more complex and expensive infrastructure that itself may have vulnerabilities.[55] Indeed, Russian scientist Vadim Marakov has elucidated a series of attacks on QKD *systems* (but not the underlying BB84 protocol).[56] The NSA concluded that whatever confidentiality QKD offers "can be provided by quantum-resistant cryptography, which is typically less expensive with a better understood risk profile."[57] As with the NSA, many companies probably see little reason to adopt a technology that will require infrastructure changes, require more training, introduce new complexities, and all for limited benefits against attackers many years in the future.

Nevertheless, QKD vendors are trying to overcome the skepticism. Four recent developments paint a path for greater QKD adoption in both the private sector and in governments. First, QKD devices have been miniaturized. ID Quantique and MagiQ both market rack-mounted QKD systems. Second, the general upset caused by the Snowden documents caused policymakers in other regions to make stronger communications security a priority and to make large vertical industrial policy investments in quantum technologies. This policy commitment may overcome the natural resistance to a switch to QKD. For instance, the European Union's quantum technologies strategy makes wide dispersal of QKD (and QRNG) a priority, even for consumer devices. The European Union's OpenQKD project, a three-year €15 million program (2019-2022), explicitly seeks standardization and other objectives to kick start a Continental QKD industry. Third, progress is being made on technical challenges, such as increasing the length of fiber over which QKD can operator: in 2018 scientists demonstrated QKD over a 400 km fiber run.[58] These ultra-long runs cause signal attenuation and key acquisition slows to a crawl (as much as 24 hours

[54] U.S. Air Force Scientific Advisory Board, *Utility of Quantum Systems for the Air Force Study Abstract* (2016).

[55] Scarani and Kurtsiefer, "The black paper of quantum cryptography: Real implementation problems" (2014).

[56] Anqi et al., "Implementation vulnerabilities in general quantum cryptography" (2018).

[57] National Security Agency, *Quantum Key Distribution (QKD) and Quantum Cryptography (QC)* (2020).

[58] Boaron et al., "Secure quantum key distribution over 421 km of optical fiber" (2018).

for a key block), but improvements are steady. Finally, concerns about the privacy and security of 5G telecommunications networks is driving international concern and an unprecedented search for technical security measures.

On this last point, the security of 5G, consider the activity of South Korea Telecom (SK Telecom). Operating in the shadow of North Korea, with its active, audacious intelligence activities, SK Telecom officials must contemplate that their own employees might be forced into revealing telecommunications data to North Korea. In 2016, SK Telecom started implementing QKD in some back-haul operations of their LTE network. This effort expanded in later years to 5G infrastructure. As QKD is implemented in SK Telecom's stack, the number of employees who could be coerced into revealing information to North Korea presumably winnows.

QKD or quantum networking to a consumer handset will probably never be a reality, but it is likely that QRNG will make it there: In May 2020, ID Quantique announced that its system-on-a-chip QRNG had be implemented in a handset offered by SK Telecom. In September 2020, as part of South Korea's $133 billion "digital new deal" program, the country will pilot QKD implementations in several critical infrastructures.

7.5 Quantum "Internet"

What's colloquially called "quantum internet" could be thought of the attempt to bring quantum computing to an infrastructure reminiscent of the internet. With a quantum internet, any two parties on a large network could communicate over some kind of quantum circuit made up of flying qubits, just as the conventional internet allows two parties to communicate using a virtual circuit built using packet switching. With a quantum network, Alice and Bob could communicate using quantum states, allowing them to enjoy both enjoy the protection of quantum cryptography, and also give them the ability to engage in quantum protocols or compute with quantum algorithms.

There are three non-obvious advances that follow from the resilient management of quantum states across distance and devices: first, mastery of quantum networking would make it possible to assemble a quantum computing cluster. Thus quantum networking could change the strategy by which organizations plan to build large quantum computers. Instead of mastering the management of single device with many qubits, a quantum network would allow organization to connect together several smaller, perhaps less expensive and easier to manage devices, into a cluster that has more qubits and volume than any competitor. Such a quantum network might reside within a single building. But while companies such as IBM, with its research lab full of quantum devices, seems well poised to do this, there (as of yet) no public evidence that IBM or others are taking this tack.

Second, a quantum network could enable *blind* quantum computing. Recall that quantum computing, because of its expense and complexity, is likely to be available as a cloud service rather than as on-premises devices. Currently, users of cloud-based quantum computers offered by Amazon and its competitors access those devices through classical communication and control computers. In a world with a functioning quantum internet, that cloud access could become end-to-end quantum intermediated. At that point, the owner of the cloud-based quantum computer would be blind to the user's action. Being blinded would limit policy

options because the quantum computing owner might not be able to detect and deter unwanted uses of the device, such as cryptanalysis or currently unimagined noisome behavior.

Depending on how it is implemented, a quantum internet might deny adversaries the ability to spy on metadata. Currently metadata, the data about data in the communications network, such as who calls whom and when, is a key tool of intelligence agencies. Metadata is well structured and relatively easy to analyze. Most people can be identified by their metadata (because most people do not constantly obtain new, clean communications devices) and even though metadata lacks information about the content of communications, metadata often hints at individuals' activities. If a quantum internet is used to set up quantum circuits between the endpoints so that the flying qubits properly travel from Alice to Bob, then such setup might be susceptible to surveillance. But if the quantum internet is itself controlled *inband* with its own quantum signaling, then it will be difficult to track who is talking to whom. Although this would be a real "going dark" problem that might have intelligence agencies and advertising agencies alike worried, such a possible network seems decades in the future.

Indeed, the challenge of realizing a large-scale quantum network is related to the very attributes that give quantum communications so much privacy: the no-cloning property. Jian-wei Pan's team demonstrated quantum communication over short distances, extending networks on optical fiber over a distance of about 100 kilometers in 2008.[59] In traditional fiber optic networks, light becomes diffused from the twists and turns of the fiber and needs to be periodically "repeated," or boosted, to travel to its final destination.[60] But the act of repeating requires copying, which is something that quantum networks can't do. Thus, a repeater on a quantum network breaks the end-to-end guarantees that users of a quantum network would want the network to provide. Although an approach may be developed to address this problem, in the near-term quantum networks will likely involve some sort of trusted repeater that catches the flying qubit, performs a classical computation, and then transmits a brand new flying qubit down the fiber.

Repeater node trust could be seen as a blessing or a curse—depending on one's perspective, it either can enable lawful access to otherwise unbreakable key exchange, or it represents a problematic security loophole. Still, even a classically-relayed quantum network is advantageous, in that if one controls the relay points, one could detect interception and still enjoy lawful access when needed. For instance, the political attributes of China probably fit neatly with the limits of classical repeaters. Those nodes could be operated by state-controlled companies, and surveilled when desired by domestic law enforcement and intelligence, while denying that same ability to foreign adversaries. Jian-wei Pan himself boasted, "China is completely capable of making full use of quantum communications in a regional war…The direction of development in the future calls for using relay satellites to realize quantum communications and control that covers the entire army."

A *quantum repeater* or *quantum memory router* can overcome the trust problem. The first re-transmits the flying qubit, and the second allows the flying qubit to

[59] Yuan et al., "Experimental demonstration of a BDCZ quantum repeater node" (2008).

[60] Briegel et al., "Quantum Repeaters: The Role of Imperfect Local Operations in Quantum Communication" (1998).

fly off in one of several possible directions. Such devices are still in their infancy.[61] Quantum internet routers are in effect small quantum computers. One approach uses atomic vapor technologies, specifically Electromagnetically Induced Transparency (EIT), introduced in Section 2.2, "Modern quantum sensing approaches" (p. 31). Scientists are working on the fidelity of copying and storage time; as of 2019, EIT memory loses fidelity in just microseconds.[62]

Quantum "teleportation" is a mechanism being explored to build quantum networks. Teleportation in science fiction is as unexplained as it is exciting. What exactly do teleporters do? How they work seems to change from season to season and among different series. The most well-developed fictional teleportation system appears in Star Trek, but the fictional "transporter" was originally created by the series writers to save the cost (in terms of special effects and screen time) of needing to use the ship's shuttle craft to send the crew down to the planet.[63]. Over time, the transporter became a useful plot device for creating and then exploring psychological situations, but similar to the show's "warp drive," the underlying physics were never satisfactorily explained.[64]

In contract to mythical teleportation devices, *quantum teleportation* is in effect that is well understood and has even been demonstrated. Quantum teleportation moves the *quantum state* from one particle to a second, irrevocably changing the state of the first particle in the process. Because the state is moved and not copied, quantum teleportation violates neither the Heisenberg uncertainty principle nor the "No Cloning" theorem, which holds that quantum states cannot be precisely copied.

One possible way to construct a quantum router is to use quantum teleportation to transmit data to some point in the distance, in effect creating a point-to-point communication between Alice and Bob. Teams at TU-Delft led by Stephanie Wehner and Ronald Hanson have impressive accomplishments in advancing entanglement and in teleportation. In a TU-Delft demonstration of quantum teleportation, Alice and Bob share a classical communication channel and an entangled particle. The entangled particle is a nitrogen-14 spin inside a diamond. Known as a "nitrogen-vacancy" chamber, this imperfection in a synthetic diamond isolates and insulates the nitrogen atom from the outside environment (see Chapter 2, Section 2.2, "Modern quantum sensing approaches" (p. 31)). That isolation makes the nitrogen spin more resilient to unwanted interference. With the nitrogen atoms entangled over a distance, Alice takes a second atom, the information bit, and performs a so-called "Bell measurement" between her entangled atom and the second atom. The measurement causes a corresponding change to Bob's entangled qubit. Bob can then

[61] Yan and Fan, "Single-photon quantum router with multiple output ports" (2014); Pant et al., "Routing entanglement in the quantum internet" (2019); Korzeczek and Braun, *Quantum-router: Storing and redirecting light at the photon level* (2020).

[62] Wang et al., "Efficient quantum memory for single-photon polarization qubits" (2019b).

[63] Whitfield and Roddenberry, *The Making of Star Trek* (1968).

[64] In both the original and Next Generation Star Trek series, transporters caused accidents and created doppelgangers: a good and evil Captain Kirk, and a copy of Commander Riker. In Star Trek Voyager, a teleporter accident fused a Vulcan (Tuvok) with a Talaxian (Neelix), creating the unfortunate Tuvix. In Spaceballs (1987), President Skroob's head materialized backwards, so that he faced his posterior, to the delight of the crew. An earlier transporter appeared in the movie "The Fly" (1958), in which a teleporter affixed a fly's head atop a smart scientist's body. The scientist keeps his mind, but is under siege from the fly's entomic instincts. See Rzetenly, "Is beaming down in Star Trek a death sentence?" (2017) for contemporary examination regarding the philosophical implications of creating a perfect copy of a person while destroying the original.

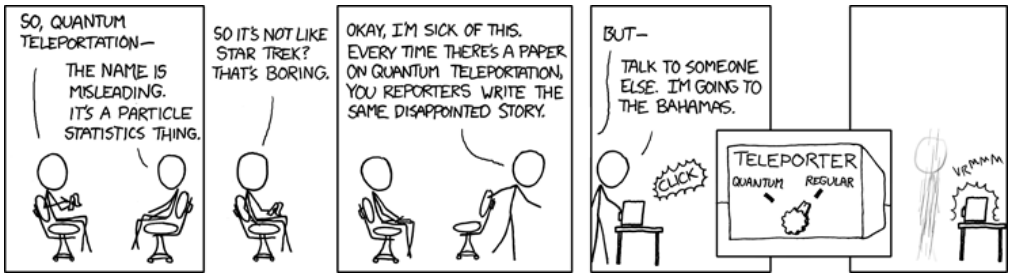


Figure 7.7: xkcd #465: Quantum Teleportation. Used with permission. https://xkcd.com/465/

extract the information—the state that Alice sent—by communicating with Alice over a classical channel. Alice tells Bob the transformations she made; by performing these same steps, Bob can extract the value of the original state.[65] Because this process uses both quantum entanglement and classical channels as a medium, teleportation protocols do not support faster-than-light communication, as is sometimes claimed (See the **sidebar "Quantum "Internet""**)

[66] Quantum teleportation was first conceived by an international team that included Charles Bennett and Gilles Brassard.[67] In 1997, scientists at the Austrian Institut für Experimentalphysik demonstrated teleportation in a laboratory setting using photons and their spins. Jian-Wei Pan was part of that team, then training under Austrian physicist Anton Zeilinger. Since then, teleportation has been demonstrated at greater distances. The TU-Delft team demonstrated teleportation at 3 meters in 2014 and by 2017, Jian-Wei Pan's team demonstrated teleportation at 1 400 km using entangled photons between a base station in Ngari, Tibet (elevation 4 500 m) and the Micius satellite.

To enable teleportation over greater distances, and indeed in a quantum internet, scientists are experimenting with entanglement *swapping*. In entanglement swapping, communication between Alice and Bob is made possible even if they lack a point-to-point path. The process works with a device, operated by a third party (here called Faythe), close enough to Alice and Bob to receive an entangled photon separately from each of them.[68]

The European Union has identified a quantum internet as a central goal in its 1 billion Euro investment in quantum technologies,[69] and scientists there have already achieved several key steps towards the creation of a quantum internet. The most synoptic expression of this vision is written by the german physicist Stephanie Wehner and it makes it clear that a quantum internet is seen as a special purpose network to exist alongside the classical internet.[70] The quantum internet is intended to maintain a channel capable of special functions, such as quantum key distribution, secure identification and others.

[65] Pfaff et al., "Unconditional quantum teleportation between distant solid-state quantum bits" (2014).

[66] Ren et al., "Ground-to-satellite quantum teleportation" (2017).

[67] Bennett et al., "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels" (1993).

[68] Halder et al., "Entangling independent photons by time measurement" (2007).

[69] High Level Steering Committee DG Connect, *Quantum Technologies Flagship Intermediate Report* (2017b).

[70] Wehner, Elkouss, and Hanson, "Quantum internet: A vision for the road ahead" (2018).

If nations decided to invest in creating a quantum internet, network paths would become a key focus. From a technical perspective, all paths would have to be fully quantum mechanical, or the quantum state would collapse and the technology would fail. Strategically, adversaries along those paths could easily interfere with the quantum state, causing it to collapse. These attacks on availability need not be at the router or even that sophisticated. Anything that degrades the light will work, meaning that these attacks might be easily deniable, and attributable to accident and so on.

Going back to the time of the telegraph, communications find their way along wires on specified routes. If a telegraph pole fell in a storm, that path would be interrupted, and the pole would have to be replaced or a new path set into place. One major advance of the internet was packet switching, the conversion of communications into datagrams that could take multiple routes. The sender and recipient need not specify these routes. But this lack of specificity comes with a downside: because the communications' paths change dynamically, an attack can intentional interfere with one route and force the communications to travel over another route with lower legal or technical protections.[71] Recently, the risk that Internet communications takes unnecessarily circuitous routes through other legal jurisdictions has become a concern of some nations. A 2019 study focusing on path-based risks studied tens of thousands of likely paths a user's browser might take when visiting popular sites. The group found that 33% "unnecessarily expose network traffic to at least one nation state, often more"[72] Some nations are building local internet exchange points to keep more communications domestic, and out of paths that traverse China, Russia, the U.S. or its "five-eyes" allies.

A quantum internet would almost certainly require that nations and sophisticated companies are likely to create dedicated fiber links for a quantum network, making it more like a separate, dedicated private network. The infrastructure for communication is likely to become much more state-specific. Already, sophisticated users are able to choose the paths that their conventional internet communications travel; the same will likely be true of quantum networks, if they are ever created. Already the Dutch telecom provider KPN has built a fiber optic, quantum channel network backbone between Leiden, Delft, Amsterdam, and the Hague. (The KPN network does not require repeating because of the short distances among these cities.[73])

Another option comes from satellites. It seems less likely that a satellite could be manipulated by an adversary than an underwater repeater. At least a half a dozen countries are pursuing satellite-based QKD programs.[74] Either physical or cyber manipulations could be impactful. Thus, initiatives such as Elon Musk's SpaceX/Starlink satellite network, which intends to populate the sky with internet-providing satellites, could also form the backbone of a tamper-resistant network that is mostly classical but could include quantum elements: perhaps two quantum-enabled ground-stations on opposite sides of the planet would communicate with a

[71] Woo, Swire, and Desai, "The Important, Justifiable, and Constrained Role of Nationality in Foreign Intelligence Surveillance" (2019).

[72] Holland, Smith, and Schuchard, "Measuring irregular geographic exposure on the internet" (2019).

[73] Baloo, "KPN's Quantum Journey" (2019).

[74] Khan et al., "Satellite-based QKD" (2018).

Sorry, Faster-than-light Communication is not Possible

Experiments in entanglement show that entangled particles somehow "know" the quantum state of their twin. One might think of entangled particles as parts of a connected system. Scientists do not know how they are connected, but scientists can show through Bell tests (see Section B.4 (p. 373)) that they are.

Quantum teleportation takes advantage of the linkage between distant particles to teleport a state from Alice's entangled particle to Bob's. Because Bob's particle reacts instantly, even when separated by great distances, some have speculated that teleportation could somehow enable faster-than-light (superluminal) communication. Alas, quantum teleportation does not enable faster-than-light communication.

Superluminal communication is impossible because quantum teleportation protocols depend on classical channels to extract the meaning from the entangled qubits. After teleporting a state to Bob, Alice and Bob communicate over a classical channel. Bob determines the teleported state by applying transformations that correspond to Alice's instructions.[a] This is the basis of the BB84 and E91 protocols.

So as one can see, the reversion to a classical channel, and the complexity of the information exchange and discovery, makes it impossible to communicate faster than light speed.

[a] Pfaff et al., "Unconditional quantum teleportation between distant solid-state quantum bits" (2014).

message passed from satellite-to-satellite.

Similarly, one might imagine businesses that place point-to-point servers connected by quantum channels in physically inaccessible places, for instance submerged in containers that if opened would fail.

7.6 Conclusion

Quantum communications can be binned into two categories: first, the related applications of quantum random number generation and key distribution, and second, technologies that enable a quantum network or quantum internet. While quantum random number generation and key distribution are both maturing technologies, early systems have been commercialized and are in use today. These technologies meet two central requirements for secure communications technologies: they are information theoretically secure and enable distribution of keys at a distance. Those who adopt QKD will never have to be worry that the keys they use today in encryption systems based on the RSA or Elliptic Curve public key cryptography systems might be cracked by some powerful quantum computer in the future—although adopters of today's QKD systems still need to verify that the QKD systems themselves are still secure against traditional vulnerabilities, such as electromagnetic radiation or cyber-attack.

Yet, if experience with other privacy-enhancing technologies holds, only entities with the most to lose will affirmatively adopt them. Banks, militaries, intelligence

agencies, and other entities with the awareness and budget are likely adopters. But for everyone else, three other requirements must be met: the system has to be fast, and it has to be usable by anyone, and it has to be on by default. The coming availability of classical encryption that is quantum resistant will be satisfactory for many actors. Unless some economic interest arises and militates strongly in favor of quantum encryption, most consumers and businesses will rely on classical alternatives.

The quantum internet's best use in the future—aside from its ability to procure funding for prestigious science projects—seems to be the interconnection of existing, small quantum computers into a cluster of unprecedented power. The other benefits, relating to time synchronization and astronomy, seem so tethered to scientific and technical users that it is difficult to see how they would inspire a commitment to outlay the money to make a quantum internet happen. In the nearer-term, the quantum internet's potential to make communications end-to-end secure and eliminate metadata surveillance may be the driving factor for nation states to invest in the technology.