Cryptography Apocalypse

Preparing for the Day When Quantum Computing Breaks Today's Crypto

Roger A. Grimes

WILEY

Cryptography Apocalypse

Cryptography Apocalypse

Preparing for the Day When Quantum Computing Breaks Today's Crypto

Roger A. Grimes

WILEY

Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto

Published by
John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030
www.wiley.com

Published simultaneously in Canada

ISBN: 978-1-119-61819-5
ISBN: 978-1-119-61821-8 (ebk)
ISBN: 978-1-119-61822-5 (ebk)

Manufactured in the United States of America

Library of Congress Control Number: 2019946679

I dedicate this book to my wife, Tricia. She is the woman behind the man in every sense of the saying.

About the Author

Roger A. Grimes has been fighting malicious computer hackers for more than three decades (since 1987). He's earned dozens of computer certifications (including CISSP, CISA, MCSE, CEH, and Security+), and he even passed the very tough Certified Public Accountant (CPA) exam, although it has nothing to do with computer security and he is the worst accountant ever. He's been paid as a professional penetration tester to break into companies and their websites for over 20 years, and it has never taken him more than three hours to do so. He has created and updated computer security classes, been an instructor, and taught thousands of students how to hack or defend. Roger is a frequent presenter at national computer security conferences. He's previously written or co-written 10 books on computer security and more than a thousand magazine articles. He's been the computer security columnist for *InfoWorld* and *CSO* magazines (`www.infoworld.com/blog/security-adviser/`) since August 2005, and he's been working as a full-time computer security consultant for more than two decades. Roger is frequently interviewed by magazines and television shows, and for the radio, including by *Newsweek* magazine and NPR's *All Things Considered*. Roger currently advises companies, large and small, around the world on how to stop malicious hackers and malware in the quickest and most efficient ways. He has been reading and studying quantum physics since 1983.

You can contact and read more from Roger at:

- Email: `roger@banneretcs.com`
- LinkedIn: `www.linkedin.com/in/rogeragrimes/`
- Twitter: `@rogeragrimes`
- CSOOnline: `www.csoonline.com/author/Roger-A.-Grimes/`

Acknowledgments

I would like to thank Wiley and Jim Mintel for greenlighting this book. I had been giving presentations on this topic for more than a year to enthusiastic crowds and didn't see the opportunity right in front of my face. Thanks to my employer, KnowBe4, Inc., and awesome CEO Stu Sjouwerman, Kathy Wattman, Kendra Irmie, and Mary Owens for letting me develop the original presentation and go around the county presenting it. Thanks to my core KnowBe4 quantum presentation support team: Amy Mitchell, Jessica Shelton, and Andy Reed.

I want to thank everyone whom I interviewed and sent emails back and forth with in my quest to better understand how quantum computers will impact our world, including cryptography. I don't think my head has ever hurt worse figuring out how to understand very complex concepts and trying to convey them in a more understandable way to lay audiences. This was further complicated by the fact that much of quantum physics and cryptography is defined in the world of advance mathematics. In a few areas I just gave up and just quoted what the experts wrote or said. Easily my favorite "give-up" quote appears in Chapter 6 when I'm trying to generally describe one of the quantum-resistant ciphers: "The NTRU Prime team describes their cipher as 'efficient implementation of high-security prime-degree large-Galois-group inert-modulus ideal-lattice-based cryptography' and which others describe as using 'irreducible, non-cyclotomic polynomials.'" I still can't stop laughing when I see that description because of everything I do not know involving it and all the advanced mathematics I would have to explain to basically say, "This is a really hard-to-solve math problem."

With that said, any factual errors made in this book are mine alone. I tried my best to make sure not a single mistake made it into the book. I pride myself on being factually correct above everything else. But in a book that covers so many advanced topics, I've bound to have made mistakes. There is going to be a quantum cryptographic expert somewhere mad at me for horribly messing up some key concept. Please know that I tried my best to be as accurate as possible, and that I'm only human. I apologize in advance for any mistakes.

I want to thank all the great teachers and writers who attempted to more simply explain quantum mechanics and computing to me and everyone else. In this book, I often repeated examples and allegories made by many others that I have read, listened, and watched over the last 20 years. I only understand these sometimes difficult subjects because of their prior work. I tried to give credit to any examples or explanations where I could remember or find the author. I apologize for any missed credit. I am simply humbled.

I want to thank all the submission teams who responded to my call for their help to correct and clarify my summaries of their NIST-submitted algorithms in Chapter 6. They tried their best to

get me see the facts of their cryptographic solution. Not all teams replied (or replied in time) to my queries. Here are the ones who did: Peter Schwabe with CRYSTAL-Kyber; Thomas Prest with FALCON; Douglas Stebila with FRODOKEM; Philippe Gaborit with HQC, Rollo, and RQC; Vadim Lyubashevsky with Dilithium; Xianhui Lu with LAC; Marco Baldi with LEDCrypt; Ward Beullens with LUOV; Joost Rijneveld with MQDSS & SPHINCS+; Simona Samardziska with MQDSS; Thomas Poeppelmann with NewHope; John Schanck with NTRU; Nina Bindel with qTESLA; Scott Fluhrer with SPHINCS+; and Mike Hamburg with ThreeBears. Thank you all.

I'd like to give special thanks to University of Texas Austin quantum professor Scott Aaronson; physical science writer Philip Bell; Ken Mafli of Townsend Security; and Daniel Burgarth. Last, a big thanks to the following Wiley folks who put up with my constant complete rewrites: Kim Wimpsett, Pete Gaughan, and Athiyappan Lalitkumar. They finally had to stop me from adding things and tell me to let them print it.

NOTE I often intentionally or unintentionally used the word *cipher* to describe any cryptographic algorithm. Technically, *cipher* refers only to encryption algorithms, and digital signature algorithms are *schemes*. I sometimes used the word *cipher* to refer to either to make writing about cryptography over nine chapters easier. Please forgive any technical misuse.

Contents

Contents

Introduction

In the late 1990s the world was consumed by a coming computer problem known as Y2K, which stood for the Year 2000. The difficulty was that most of the world's devices, computers, and programs to that point in time recorded dates using only the last two digits of the year. From a programmatic level, they couldn't tell the difference between 1850, 1950, and 2050.

When 1999 turned into 2000, many of those computers and programs would not have been able to correctly process any calculation involving two-digit dates in the new century. There had been many known failures by programs and devices that were already using dates in the future (such as scheduling and warranty programs). Symptoms of failed devices and programs ranged from visible errors to errors that happened but were not readily visible (which can be extremely dangerous) to complete device and program shutdowns.

The problem was that although we knew that a sizable percentage of devices and programs were impacted, no one knew which untested things were fine and didn't need to be updated and which had to be updated or replaced before January 1, 2000. There was a two- to three-year rush to find out what was broken and what was fine. As with many slow-moving potential catastrophes, most of the world did little to nothing to prepare until the last few months. The last-minute global rush created a bit of a worldwide panic about what would happen as clocks moved into the new century. There was even a fantastically bad 1999 disaster movie (`www.imdb.com/title/tt0215370`) that had planes dropping out of the sky along with other worldwide cataclysmic mayhem.

In the end, when Y2K rolled around, it was a bit of a dud if you wanted real life to be like the movies. There were issues, but for the most part the world continued as usual. There were devices and programs that failed to handle the newer dates appropriately, but most major systems worked correctly. There were no falling planes, fires, or burst dams. For many people who were expecting disaster outcomes, it was a bit of a letdown—so much so that, over time the term Y2K evolved to become a unofficial synonym for overly hyped events involving premature panic with little resulting damage.

What most people today don't realize is that Y2K was anticlimactic precisely because we had years of preparation and warning. Most major systems were checked for Y2K issues and replaced or updated as needed. Had the world not become aware of it and not done anything, Y2K would have certainly been far, far worse (albeit, I'm still not sure planes would be falling out of the sky). Y2K wasn't a premature panic dud. It was the foreseeable outcome from years of preparation, demonstrating the success of what humanity can do when faced with a looming digital problem.

The Coming Quantum Day of Reckoning

Most of the world doesn't know it yet, but we are in another even more momentous, looming Y2K moment, except this one is likely already causing serious problems and damage. Worse, we can't stop all the damage even if we begin preparing now. There are organizations sustaining harm today that will not be able to program their way out. Nation-states and corporate adversaries are likely already taking advantage of the problem.

Quantum computers will likely soon break traditional public key cryptography, including the ciphers protecting most of the world's digital secrets. These soon-to-be-broken protocols and components include HTTPS, TLS, SSH, PKI, digital certificates, RSA, DH, ECC, most Wi-Fi networks, most VPNs, smartcards, HSMs, most cryptocurrencies, and most multifactor authentication devices that rely on public key crypto. If the list just included HTTPS and TLS, it would cover most of the Internet. On the day that quantum computing breaks traditional public crypto, every captured secret protected by those protocols and mechanisms will be readable.

Even more important, anyone capturing and storing those (currently protected) secrets will be able to go back after the quantum crypto break and reveal them. How many secrets do you have or does your organization have that you want revealed to anyone within a few years? That's the new Y2K problem we are dealing with today.

There are many workable solutions you can implement today, although some are beyond the average company's means or, if implemented prematurely, can cause significant performance and operational disruption. Preparing for the coming quantum break requires education, critical choices, and planning. Individuals and organizations who clearly understand what is ahead can take the right steps now to be as prepared as possible. They can stop the unwarranted eavesdropping today and start to move their managed assets to a more quantum-resistant environment. This book has that knowledge and gives you the plan to help minimize your organization's risk from the coming quantum crypto break. If enough organizations prepare now, we can make the quantum break as inconsequential as the Y2K problem.

Who This Book Is For

This book is primarily aimed at anyone who is in charge of managing their organization's computer security and, in particular, computer cryptography. These are the people who will likely be in charge and leading the way for their post-quantum migration project. It is also for managers and other leaders who understand the importance of good cryptography and its impact on their organization. Last, anyone with a passing interest in quantum mechanics, quantum computers, and quantum cryptography will find many new facts to make this book a worthwhile read.

What Is Covered in This Book?

Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto contains nine chapters separated into two parts.

Part I, "Quantum Computing Primer," is a basic primer on quantum mechanics, computing, and how it can break today's cryptographic protection.

Chapter 1, "Introduction to Quantum Mechanics"

If you didn't understand quantum mechanics the first time you read about it, don't worry—quantum mechanics has vexed the most brilliant minds our planet has ever had for over a century. We mere mortals can be forgiven for not immediately grasping the central concepts. Chapter 1 explains the properties most important to our understanding of how it impacts our digital world. If I do my job right, you'll understand it better than 99 percent of everyone else in the computer world.

Chapter 2, "Introduction to Quantum Computers"

Quantum computers use quantum properties to provide capabilities, logic, and arithmetic outcomes that are simply not possible with traditional binary computers. Chapter 2 covers the different types of quantum computers, the various quantum properties they support, and where they are likely headed in the next decade as we become surrounded by them.

Chapter 3, "How Can Quantum Computing Break Today's Cryptography?"

The most common question asked when a person is told that quantum computers will likely break traditional public key cryptography is how. Chapter 3 tells why traditional binary computers can't easily break most public key crypto and how quantum computers likely will. It covers what quantum computers are likely to break and what is resistant to quantum computing power.

Chapter 4, "When Will the Quantum Crypto Break Happen?"

After explaining how quantum computers will likely break traditional public key crypto, the second most often asked question is when it will happen. Although no one (publicly) knows, it is likely to be sooner than later. Chapter 4 discusses the different possible timings and their possibilities.

Chapter 5, "What Will a Post-Quantum World Look Like?"

Like the invention of the Internet, there will be a world before and a world after quantum supremacy. Quantum will solve problems that have plagued us for centuries and will give us new problems that will vex us in the future. Chapter 5 will describe that post-quantum world and how it will impact you.

Part II, "Preparing for the Quantum Break," will help you and your organization most efficiently prepare for the coming quantum supremacy.

Chapter 6, "Quantum-Resistant Cryptography"

Chapter 6 covers over two dozen quantum-resistant ciphers and schemes, which the National Institute of Standards and Technology (NIST) is considering in the second round of its post-quantum

contest. Two or more of these quantum-resistant algorithms will become the next U.S. national cryptography standards. Read about the competitors and their strengths and weaknesses.

Chapter 7, "Quantum Cryptography"
Chapter 6 covered traditional binary quantum-resistant cryptography, which does not use quantum properties to provide protection. Chapter 7 covers ciphers and schemes, which do use quantum properties to provide their cryptographic strength. In the long run, you will likely be using quantum-based cryptography and not just quantum-resistant cryptography. Come learn what that looks like.

Chapter 8, "Quantum Networking"
Chapter 8 covers quantum-based networking devices, such as quantum repeaters, and the applications that are seeking quantum network protection. It covers the current state of quantum networking and where it will likely be over the near-term and long-term futures. One day the entire Internet will likely be quantum-based. Read about those networking parts and components and how we will get there.

Chapter 9, "Preparing Now"
Chapter 9 is a perfect reason to buy this book. It tells any organization how they can start preparing today for the coming quantum cryptographic break. It tells you what you can do today to protect your most critical long-term secrets, what cryptographic key sizes you need to increase, and what has to be replaced and when. The summarized plan has been used in previous global cryptographic updates and can be used to ward off a cryptographic apocalypse.

The appendix lists dozens of links to quantum information resources, including books, videos, blogs, white papers, and websites.

If I've done my job correctly, by the end of this book you will comprehend quantum physics better than ever before, understand how it will break today's traditional public key cryptography, and be able to appropriately prepare and better protect your critical digital secrets.

How to Contact Wiley or the Author

Wiley strives to keep you supplied with the latest tools and information you need for your work. Please check the website at `www.wiley.com/go/cryptographyapocalypse`, where I'll post additional content and updates that supplement this book should the need arise.

If you have any questions, suggestions, or corrections, feel free to email me at `roger@banneretcs.com`.

Quantum Computing Primer

Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto, First Edition. Roger A. Grimes.

1 Introduction to Quantum Mechanics

Those who are not shocked when they first come across quantum theory cannot possibly have understood it.

Niels Bohr, quantum physicist and 1922 Nobel Prize winner

Any sufficiently advanced technology is indistinguishable from magic.

Arthur C. Clarke, science-fiction author

Chapter 1 will discuss quantum mechanics basics, concentrating on the topics that relate particularly to quantum computing. This chapter is intentionally not completely inclusive as that would require a book and not just a chapter on the subject. It will not cover every particle, property, or possible interaction and will skip all the complicated math and equations.

This chapter will give you enough of an understanding of quantum physics to explain how quantum computers are capable of quickly answering previously considered impossible-to-solve math problems, which many common types of encryption are based on to provide protection. Understanding quantum mechanics and quantum computing perfectly is not required to prepare for the coming cryptographic breaks, but it does help to have some background basics when discussing the relevant issues with others.

What Is Quantum Mechanics?

In this section, I'll explain quantum mechanics, but I want to give a little caution if this is your first exposure to the topic. Quantum mechanics is incredibly cool, but at the same time we don't fully understand what is going on. Much of it seems so strange to our current understanding of how the world works that fully comprehending it for the first time isn't easy for most people. Even after nearly 30 years of trying to fully grasp the entirety of the field and its implications, my head still

Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto, First Edition. Roger A. Grimes.

gets mentally fatigued. I'm not alone. It's being gracious to simply say that at first glance quantum mechanics is counterintuitive and seemingly unnatural. It often beggars belief. It goes against many things we've been previously taught about how our world and the universe works. One plus one does not always equal two. It goes against much of what we can readily see, touch, and feel, even though all of reality is possible due to it.

Even though the top minds of our civilization have repeatedly proven the existence of quantum mechanics beyond a shadow of a doubt, what it entails sounds so strange to the average person that it often remains unbelievable and magical. Understanding the implications of quantum mechanics for the first time means questioning what reality even means.

A not uncommon first-time response from laypeople first exposed to quantum theory is to suppose that all believers must be under some sort of science fiction, mass delusion because what they are saying cannot possibly be true. Or as a friend once said to me after I did an obviously poor job of explaining it to her, "You can believe whatever you want to believe, but that's a bunch of bull!" except she didn't say the word *bull*.

Even Albert Einstein, who helped discover and participate in some of its most important underlying principles, didn't completely believe many of its other fundamental tenets. He spent decades trying to understand it and he understood it better than most. It was his strong understanding of its implications which caused him problems. He even created experiments to prove or disprove it. He just couldn't logically believe or explain its many strange properties and "spooky at a distance" outcomes. After decades of waiting for experiments to catch up with his propositions, he just moved on to other subjects of study. Apparently, his head tired of thinking about it. So lesser minds can be excused.

With that said, I wrote this quantum primer chapter in a way that I wish it had been explained to me when I first started studying it. It is my hope that this chapter can help shorten the learning curve.

Quantum Is Counterintuitive

Even though quantum mechanics underlies all of reality, it doesn't readily appear in a way that laypeople can easily discern in their everyday life. As examples, a single-colored dog can't both be white and black at the same time, a white dog stuck in a room doesn't suddenly become a black dog when it exits, and a dog can't split into two dogs in front of your very eyes and then merge together again. But at the atomic and subatomic levels, the peculiarities of quantum mechanics are equivalently strange.

What are the quantum properties I keep saying are so strange? Here are some examples:

- A single quantum particle can be in two places and be two distinctly different things at once.
- A single quantum particle can split in two and then later appear to run into or interfere with itself and recombine or cancel itself out.
- In a truly empty space with absolutely nothing (that scientists are aware of), quantum particles can just appear "out of thin air" and then vanish.

- A quantum particle will seem to behave one way when not being measured and another when being measured, as if nature absolutely cares about the action of measurement. It will seemingly even change its path or behavior back in time if you decide to measure it after it went through its original path.
- Two quantum particles can be "entangled" in such a way that when you change one, the other also instantly changes in the same way, every time, no matter how far apart they are, even across the universe.
- A quantum state is always all possible states (called a *superposition of states*), but the single, eventual resulting state can't be predicted with certainty.
- Every possible answer will be the answer at some point, although those answers may each be in their own separate universe. There may be a different universe for each possible combination of answer choices (called *multiverses*) at the atomic level.
- *Star Trek*–like teleportation is possible.

Here's the example I love to share with people to explain exactly how strange quantum mechanics can be. When we look up into the night sky and see stars, the light from those stars has traveled millions of miles and taken many years to reach your eye. The closest stars to Earth (besides our own Sun) are 4.2 light-years away. That means that it took at least 4.2 years or longer for the light from any star that you are looking at in the moment to reach your eye. That star isn't where you think you "see it," but where the star was when the light left it many years ago. This is a great astronomic fact to share on a romantic night or with kids and friends.

Quantum mechanics says that the path that any individual particle of light (known as a *photon*) travels from the star is changed simply because you decided to look up and see it at that particular moment. The path it started was adjusted, before you looked at it, because you looked at it. And if you decided to hesitate a millisecond before you looked up or not look up at all, the photon from that star would have taken a different overall path. If your friend looked up before you and saw that same photon instead, the path the photon took from the star would be different than what it took if you looked at it. And the path appears to change back in time based upon what happened now. Seems impossible, but events very similar to this story have been witnessed and repeated over and over. We don't know what is going on or how, but we know it is occurring. We don't even know enough to know if we are describing the event correctly, only that what our meager minds appear to be seeing can be described as a historic change based on a current event. Welcome to the world of quantum mechanics!

Quantum Mechanics Is Real

The "strange" properties of quantum particles can be hard to believe. But except for the multi-universe proclamations, not only have these quantum properties and outcomes been tested and proved, but they are among the most tested and accepted scientific theories in the world. They are continuously

being tested and challenged. All experiments that have been conducted to disprove the basic, accepted theories of quantum mechanics have failed. Many of the failures, including those by Einstein, only succeeded in proving quantum theory even more. Most of the Nobel Prizes in physics from the last 75 years have been awarded to scientists who improved our understanding of quantum mechanics. There has been a renewed focus on quantum mechanics the last few decades and our understanding is improving each year.

Although the facts listed in the previous section may appear unbelievable on first reading, the genuineness of quantum physics appears to us throughout our larger reality, including how the Sun gives life to our planet, the red hot glow of any superheated material, digital cameras, fiber-optic cables, lasers, computer chips, and even the majority of the Internet (storage and transmission media). The very likely reality is that every bit of our reality is based on quantum mechanics.

Quantum mechanics is giving us very powerful computers that were previously unthinkable. Quantum computers and devices are going to change our world in many incredible ways that we can and can't fathom now, just like the Internet, USB memory storage keys, and iPods did for the current generation. Critical quantum inventions will significantly change our lives for the better, and the most important ones are coming soon.

Interestingly, although much of quantum theory has been confirmed by repeated observations, experiments, and math, scientists still don't know why many quantum properties are the way they are or why particular results occur. Theoretical physicists often take guesses about why a quantum-something is the way it is. You'll hear these guesses talked about as *interpretations* or *views*, such as the *Copenhagen interpretation* or the *Many Worlds view* (covered in the "Observer Effect" section later in this chapter). There are well over a dozen interpretations, each trying to explain some part of quantum mechanics, without really knowing if their interpretation is the accurate one.

What's important to understand is that regardless of the guess of why or how some quantum action or result occurs, the action or result does occur, always occurs in an expected way, and is experimentally and mathematically proven regardless of the interpretation. There has never been a serious quantum prediction not backed up by well-formed experimentation. We may not always know why quantum behavior is, well, quantum-acting, but we know it is real. It may seem like magic, but it is real, even if we can't explain it or "see it" in a conventional sense.

This bothers some nonscientists. Asking someone to believe in something they can't see or feel and that is supercounterintuitive to everything they've previously been taught is asking a lot. It's not like how they previously learned to appreciate science. For example, they may not understand the physics and math behind gravity, but they can "see it" and its outcome every time they throw a ball, trip and fall, see a proverbial apple fall from a tree, or watch the Moon circle Earth. They may not understand the math, but they understand how and why gravity works . . . well, most of us, that is. Many people ask, how can we believe anything science says really exists without knowing how or why it occurred? How can we believe in something we can't readily see with our own eyes, especially something so incredible and counterintuitive sounding?

What skeptics usually don't know is that much, if not most, of the advancement in science for the last century—especially in physics and especially, especially in quantum physics—has almost always first been proven by experiments and/or math without understanding why or how. Many times, scientists have only the vaguest of theories to support what little they can tangentially observe and prove with math. This is where the term *theoretical physicist* comes from. They are often starting from the barest of real evidence and haphazard an intelligent supporting theory to explain what they are observing. If they (or someone else) can provide a math equation that consistently describes what they are observing, then most scientists will rely on the math as conclusive proof of the behavior. It doesn't take a picture of something to be believed by a physicist.

The math is even more important than a picture or direct whole observation to a physicist. Someone once said, "The only absolute truth in the world is math." What they meant is that anything else besides a well-supported math equation is subject to personal biases and interpretations. Either the math works consistently or it doesn't. Either it supports something or it doesn't. It isn't subject to the opinion of the observer. If a scientist sees some previously unexplained phenomenon and can consistently support its interactions with a math formula and if every experiment and outcome is accurately described by the math, then the scientific fact is considered proven. The math is the proof. Direct, conclusive, confirmative observation isn't necessarily needed.

The conclusive observable event that most nonscientists think of as proof often comes many decades, or even centuries, later. Usually by then the involved scientists and their successors had long believed and treated the earlier theory supported by mathematical proof as a trusted fact. In their mind, the final uncontestable, physical proof is considered an almost unneeded formality.

Many past scientific postulations, both very small and very large, including the discovery of atoms, electrons, and black holes, were first discovered by scientists creating theories and math around previously unexplained observed phenomena. In the previous examples of the black hole and newly discovered solar system planets, observers had noticed subtle deviations in orbiting bodies and light that they knew could be explained only by previously unknown third-party effects. Black holes were theorized beginning in 1784 (by John Mitchell), and mathematically supported by Einstein's theory of general relativity in 1915. Further related observations over the next half century supported the math and existence of black holes, even if we couldn't "see" them. From the 1970s on, scientists considered the reality of black holes as a given. The first picture and what many nonscientists would think of as the first "real proof" of black holes didn't occur until April 2019 (`https://phys.org/news/2019-04-scientists-unveil-picture-black-hole.html`).

The history of quantum mechanics follows a similar path. It involves hundreds of brilliant physicists observing behaviors on very small objects that they could not otherwise explain using traditional (i.e., classical) physics. They then began exploring the new, strange phenomena even more, figuring out math equations that appeared to support what they were seeing. They made guesses as to why and how something was happening and then created experiments to prove or disprove their guess. Over time, additional experiments and observation created the known facts of quantum

mechanics. Some brilliant minds, like Einstein's, were proven wrong on certain facts, and previously obscure physicists had their careers made (and won Nobel Prizes) proving others. All in all, the contributions of hundreds of individual scientists and their skepticism has created the field of quantum mechanics as we know it today, strange and unexplainable as it may be at times.

The Basic Properties of Quantum Mechanics

In this section, I will cover popular properties of quantum mechanics, such as the photoelectric effect, wave-particle duality, probabilities, the uncertainty principle, spin states, tunneling, superposition, the observer effect, and quantum entanglement.

NOTE So, what is the *quantum* in quantum physics? When physicists use the term *quantum* or *quanta* (from the Latin root *quantus*, which means the amount or how much), they are stating that whatever they are describing is the smallest possible unit of something (e.g., light or energy) and cannot be divided into smaller units. And any mathematical calculation involving a quanta cannot further subdivide the quanta into anything less than a whole number.

Quantum mechanics or *quantum physics* consists of the properties of and actions of quantum particles and interactions. It is also what the field of study involving quantum properties and particles is called. Everyone pretty much uses these words interchangeably.

Although our entire reality is made up of quantum particles and actions, quantum mechanics happens at the very microscopic level on very, very small elemental objects, such as photons, quarks, electrons, and atoms. If an elemental object displays quantum properties, it's known as a *quantum particle*. The smallest known particles usually display quantum properties. Quantum properties may occur on larger objects, on what is known as the *macroscopic level*, but science has not yet advanced to understand if it does or doesn't consistently, and if it does, how it does it. Understanding how the actions of very small objects transition and impact larger things is the ultimate goal of the much-sought-after, so-called unifying *Theory of Everything*.

NOTE The macroscopic level includes any object larger than the microscopic level of atomic and subatomic particles but is often interpreted as beginning with objects that can be detected by the naked human eye. Most scientists agree that the human eye can detect an object that is the width of a human hair (or 0.4mm), or about 100,000 atoms of an element.

Photons and Quantum Mechanics

You will often read about photons (originally called *energy quanta* by Einstein) being used in quantum mechanics experiments. A *photon* is the smallest possible divisible unit of light and is

quantum-behaving. They are very small. It would take at least a hundred photons, sent nearly instantaneously, for the average straining human eye to register even a faint flicker of light. Any beam of light or image we normally see involves millions to trillions of photons.

Quantum physicists often run experiments using single (or relatively small quantities of) photons or other elementary particles, because by using small quantities, the scientists can remove other unnecessary clutter that would otherwise only complicate their experiments, the results, and mathematical proofs. Early proof of quantum properties was first discovered in experiments using photons while investigating radiation, electromagnetic waves, and the photoelectric effect (for which Einstein was awarded his only Nobel Prize in 1921). Einstein's work was critical to establishing quantum theory. Even his work to disprove quantum mechanics only improved our understanding.

For a long time now, scientists have been able to generate single protons, send them along various pathways in experiments, and measure what happens using light-sensitive equipment called *photomultiplier tubes*. A photomultiplier is able to take one detected photon and multiply it into enough other photons that an electrical current can be triggered to register and confirm the initial detected single photon. Think of it like falling dominos. One falling domino can cause a lot of other dominos to fall. For all these reasons, when you read about quantum physic experiments, you will often read about photons (and similar elementary quantum particles). Experiments using individual electrons, atoms, and molecules are also common. Let's discuss what some of those experiments have proven.

Photoelectric Effect

Understanding and quantifying the photoelectric effect in the early 1900s (by Planck, Einstein, and others) was the beginning foundation to the formation of modern-day quantum mechanics. The visible light we see is just one type and range of *electromagnetic radiation* across what is called the *electromagnetic spectrum*. The electromagnetic spectrum describes all types of electromagnetic radiation, including the visible light we can see and all the types we can't (such as x-rays, microwaves, gamma waves, and radio waves). The different types of electromagnetic radiation differ primarily by wavelength (visible light has a wavelength of 400 to 700 nanometers (nm) and x-rays have 0.10 to 10nm, as examples), frequency (often measured in cycles per second, called Hertz [Hz]), intensity, direction, and other properties. All types of electromagnetic radiation move in a relatively straight line, if unobstructed (by an object, gravity, etc.), at the speed of light (which is 299,792,458 meters per second in a vacuum).

NOTE Frequency and wavelength can be converted into each other via the speed of light and are really the same variable.

Light has momentum and energy (but no mass). Planck and Einstein realized that when light (or other forms of electromagnetic radiation) hit other material, the material would often emit electrons (which are always negatively charged) from the resulting transfer of energy from the photon to the

material, as represented in Figure 1.1. The higher the intensity of light, the more electrons emitted. The photoelectric effect occurs when light hits most materials but is most readily observed when it hits metals and other highly conductive materials. The photoelectric effect is how the sun's energy is converted into electricity by solar cells. The photoelectric effect is also behind the fundamental way digital cameras work and record images.

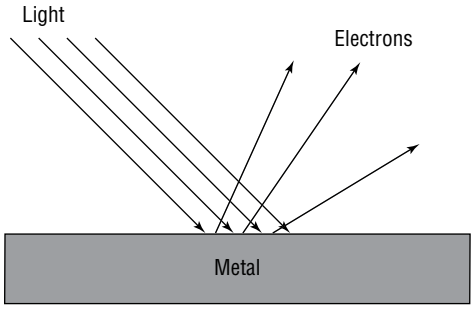

Figure 1.1: Photons hitting a material that then emits electrons
Steven Holzner, Figure 1.3 from *Quantum Physics For Dummies, Revised Edition*; Wiley, 2013

Wave-Particle Duality

For hundreds of years, everything scientists discovered at the microscopic level was classified as a particle or a wave. *Particles* are (microscopic) objects that follow well-defined, easy-to-see, physical laws, which we can see demonstrated by everyday macroscopic objects (such as rocks or balls). Particles travel along predictable paths, are subject to gravity, and have interactions with other particles and objects that are fairly easy to describe, predict, and mathematically model.

A *wave* is continuous disturbance in a field that oscillates between different points in space or some other variable. A wave may transmit energy through an underlying medium without greatly disturbing other objects in the medium, like what happens when a floating object, such as a boat on water, gets disturbed by a wave. The wave will lift the boat up and down as it passes, but not greatly disturb the boat's overall position in the water unless the wave is so big that it is cresting. Waves comprise not only things we can see (like ocean waves, ripples in a lake, or vibrations in a string), but also other waves we can't see (like sound, radio, radiation, and microwaves).

NOTE Waves can oscillate over variables other than space or position. For example, in electromagnetic waves it is the electric and magnetic fields that vary.

Waves oscillate in a continuous, repeated, connected pattern. The *wave form* of each particular type of wave has a top peak or crest followed by a bottom valley or trough, which then repeats over and over. The distance between the top and bottom of the wave form is called its *amplitude*. The

number of complete up-and-down wave oscillations in a particular time period determines its *frequency*.

Particles and waves are guided by very different physical properties, or so scientists thought. Particles function more like rocks or baseballs. They don't easily "bend" around objects. They strike with momentum and force. Their collision trajectories and resulting glancing bounces can be predetermined and calculated ahead of time. You can more easily see each discrete unit making up the large mass of particles, like seeing the individual rocks that make up a rock pile. A particle hitting a wall impacts it like a bug hitting a windshield. Waves have the opposite properties.

In the mid-1800s it was "settled science," after much theory and experimentation, that light and the photons that make it up traveled as waves. But starting in the early 1900s, when photons and other electromagnetic particles were observed and used in a greater number of subatomic experiments, different scientists started to notice that photons and other particles behaved as both a wave and a particle (i.e., *wave-particle duality*). At the time, this was considered scientifically blasphemous. Einstein, in particular, persisted with this new view and won his only Nobel Prize in physics for demonstrating that light also acted as a particle. Einstein wrote of his discovery:

> *It seems as though we must use sometimes the one theory and sometimes the other, while at times we may use either. We are faced with a new kind of difficulty. We have two contradictory pictures of reality; separately neither of them fully explains the phenomena of light, but together they do.*

One of the best ways to think of wave-particle duality is to imagine you have a rubber ball, which when behaving like a particle bounces all around, hitting other objects and bouncing back and forth, depending on its trajectory and what it bounces into. Then imagine that it falls into a lake and disappears (below the surface). Its energy is immediately transformed into waves and the resulting ripples. Then imagine that the wave ripples hit a dock post sitting in the water, and at that instant, a rubber ball reappears on a dock and the waves disappear. That's wave-particle duality. Depending on the situation, sometimes a photon is acting like a wave and sometimes like a particle. Thanks to Dominic Walliman for that excellent allegory.

It's a Particle

Scientists demonstrated wave-particle duality by using a simple experiment using a high-intensity (laser) light, a background, and an intervening blocking material with one or two cut slits in it. They shot photons, one a time, into the slit(s) of the blocking material and then checked to see where they landed on the background.

When one slit was used and the photon was fired, the photon went through the slit and landed somewhat directly on the background behind. When multiple photons were shot, one at a time, each landed fairly near each other, somewhat mimicking the shape of the slit. Picture a marksman firing a bullet from a rifle through the same slit. If the rifle was in the exact same position each time, you

could expect the bullet to land almost in the same place, with minor adjustments due to the rifleman's expertise, the gun's ability to accurately fire a bullet, the bullet's individual characteristics, and any other intervening factors. If the gun was shot from a bunch of different angles, the bullets could land in a more scattershot pattern. This is what happened when multiple photons were shot, one at a time. The photons were demonstrating characteristics of a particle.

Interfering Waves

Something surprising happened when they added a second nearby slit in the intervening blocking material. When they fired a single photon, it still landed on the background behind the slits, with the footprint of a single particle (i.e., like a bullet hole), but no longer directly behind the slots. Instead, as they shot more and more photons (one at a time), they seemed to land in areas not directly behind the slits. There were areas of distinct preferences, with clusters of areas with lots of aggregated landings interweaved with areas where the photons did not land much at all. It created banding—alternating areas of light and dark vertical bands (as represented by Figure 1.2).

The scientists immediately realized that what they were seeing was a result of the photons, shot one at a time, traveling as a wave (and landing as a particle). The bands are caused because when the photon, traveling as a wave, hits both slits, it creates two resulting waves, one on the other side of each slit, with each part of the original single wave going through the slit it interacted with. On the other side, the two resulting waves interfered with each other, creating the bands. But when the photon landed, it landed with the footprint of a particle (as represented by this Wikipedia video: `https://upload.wikimedia.org/wikipedia/commons/e/e4/Wave-particle_duality.ogv`). It was a remarkable finding.

The banding is created by the waves interacting with each other. If one light wave is at top of its crest and it meets another light wave at the top of its crest at the same moment, it will create the largest possible combined, synchronized, light wave possible, which makes the brightest light. It also

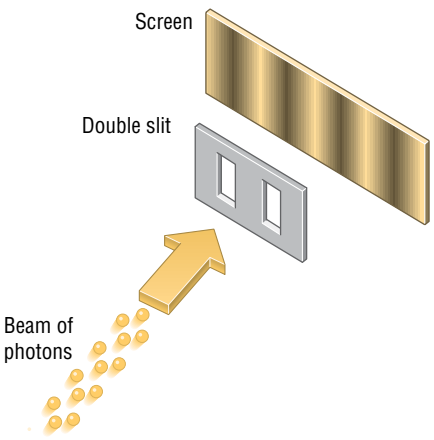

Figure 1.2: Wave-particle x duality experiment using light source and two slits
David Young and Shane Stadler, Figure 29-1 from *Cutnell & Johnson Physics, 11th Edition*; Wiley, 2018

means at their combined troughs, it creates the darkest extremes as well. Any other combination, other than two of the highest peaks (or lowest troughs) perfectly synchronizing up, will cause a smaller combined wave with less bright and dark outcomes.

Early on when this was being discovered, scientists simply couldn't believe the claims, math, or results. It took decades for light to be believed by most of science as acting as both a particle and a wave at the same time. Now we know without a doubt that all subatomic particles, which make up all matter, act with wave-particle duality. This finding strengthened scientists' resolve to more fully explore quantum mechanics and try to more fully "hook it" to the rest of our larger world. Today, anyone can perform a simple experiment to see that wave-particle duality of light.

Your Double-Slit Experiment

It's kind of cool to be able to re-create one of the early wave-particle duality experiments to see quantumness working in front of you. You can duplicate this experiment using a laser pointer, tinfoil, and a solid background such as a wall. Use a strong, solid-color (not white light) laser pointer. The stronger, the better. White light is all the colors of light and it makes the experiment harder to see because the individual colors that make up white light have different frequencies. Place the tinfoil against a cutting board surface and cut two equal-length vertical slits about 1 inch long as close together as possible (we are talking millimeters apart). Then in a darkened room shine the laser light in between the two slits from a foot or more away with the tinfoil a foot or more away from the background surface. You may have to experiment with the distances away that the laser pointer, intervening material, and wall are from each other, but if done correctly you will see the banding. It probably won't be as stark as you see in serious physics experiments with better lab equipment, but you'll get the banding.

The particle nature of light is proven in the same experiment, although we can't readily see this without special detection equipment, because each individual fired photon will be detected as a single particle right at the slits or upon landing on the background. When photon detectors are used, they confirm that each photon goes through a slit and lands as a particle. But when all the fired photons are measured over many, many experiments, the effect is that of light and dark interleaved bands, again reaffirming the wave properties of light. This one experiment proves that light (like all quantum particles and molecules) has wave-particle duality.

NOTE If you want to see real-life examples of this experiment, just go to YouTube and search for **wave double slit experiment** or something like that. You will usually find dozens of videos showing the experiment. One great, animated example is `https://www.youtube.com/watch?v=fwXQjRBLwsQ`.

Detection Strangeness

Now things get really strange. When scientists place photon detectors at one or both slits to see which slit the photon actually travels through, the photon acts as a particle and all wave-like behavior goes immediately away. Let me say that again. Before the detectors are put in front of or back of the two

slits, the photons act like waves. And after the detectors are placed and turned on, for reasons we cannot yet fully explain, the photons immediately begin acting like particles, as if there were only one slit. It's as if the particles, themselves, see the act of detection and change their behavior. Scientists have even done experiments where they don't turn on the detectors until after the photons have gone through a slit and when they turn on the detectors the photons appear to act as particles (when they should have gone through the slits as waves). It is as if the photon has retroactively adjusted its initial behavior in the past based upon the initiation of a future detection. We cannot say this (i.e., changing the past) is really happening or for that matter what time, the past, or reality really is. No one knows what is happening or how. Only that the behavior change happens anytime a detector is used, and we are having a hard time understanding what is going on. This known as part of the observer effect, which is covered in more detail below and is the explanation behind the star light path change story that started this chapter.

Probability Principle

Understanding of how electrons orbit around a nucleus led to better understanding of how our world works, especially at the quantum level. For example, as schoolchildren, we probably all learned that each atomic element is made up of electrons, protons, and neutrons. Every *atom* (the smallest unit of ordinary matter) is made up of a *nucleus* (which is made up of positively charged *protons*) and (no charge) *neutrons*, surrounded by negatively charged electrons. The electrons "orbit" the nucleus because of electromagnetic attraction. In elementary school, most of us learned that electrons circle the nucleus in orbital bands known as *shells*.

In elementary school, likely for simplistic reasons, these electron orbital shells were shown as perfect circles or maybe ovals, often conjuring up perfect planetary-like orbits, but at the atomic level (see Figure 1.3).

But quantum physics has shown us that electrons don't orbit in perfect circles or even ovals. Those perfect circle electron shells are a figment of somebody's early imagination, and today are used solely to demonstrate electron shells in an uncomplicated pattern. But that's not the way nature really works. Instead, electrons orbit the nucleus in more complex patterns dictated by quantum mechanics and the involved energy (Figure 1.4 shows a two-dimensional representative example for electron orbits around a nucleus at a particular energy level). These areas of probable orbit are known as *atomic orbits* or *electron clouds*. The probable part is very important in quantum mechanics and will be explained in more detail in the next section.

Complicating matters a bit, no one can guess ahead of time where a particular electron may be orbiting at any one time, only the *probability* of it to be in certain (predicted) atomic orbital areas. No math equation exists that can say with any certainty that any electron will be exactly in spot A at any time. The best quantum mechanics can say is that an electron has a particular percentage of likelihood of being in spot A when you try to measure it. And if you take that measurement many times, that electron will be in spot A the number of times indicated by its probability percentage.

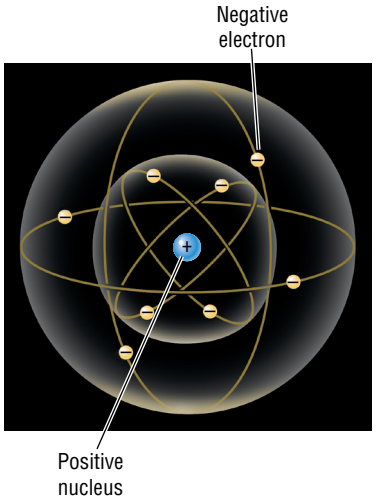

Figure 1.3: Atom nucleus surrounded by overly simplified electron shell orbits
David Young and Shane Stadler, Figure 30-1 from *Cutnell & Johnson Physics, 11th Edition*; Wiley, 2018

The probability principle applies to any property of a quantum particle, not just electrons. Not only can't a particular property state or position be guessed ahead of time, but the state or position when measured is absolutely random within the larger confines of the probability predictions during any single measurement. And this randomness of a specific answer or state isn't accidental; it's fundamental and inherent to quantum mechanics.

This is a key difference between quantum mechanics and traditional, classical physics in that the exact state or position of a quantum object or property cannot be precisely predicted ahead of time. In classical physics, A + B = C, and will always equal C. Not only that, but if I know A and C, I can

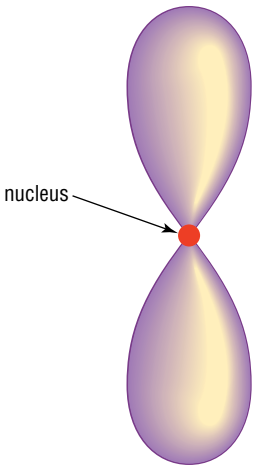

Figure 1.4: Two-dimensional atomic orbit for an electron orbiting a nucleus

predict B, and so on. But with quantum mechanics, an electron's exact place in orbit or a quantum property or placement of any quantum particle can only be described by likelihoods and probabilities. No one knows ahead of time what any single measurement answer will be. Only that it will be from a range of possible answers and if taken many times the measured outcomes will meet a predicted likelihood. That's the best anyone can do ahead of the final taken measurement.

Those possible areas and the *likelihood* of an electron being in a particular place or a specific quantum property once found and measured can be accurately predicted. If you were to run a particular electron experiment over and over (say a thousand times) to measure exactly where an electron was at a precise moment, you couldn't predict the position of any particular measured electron in a single measurement. Quantum mechanics says any single measured placement would be a random event. But all the locations taken together around the nucleus where the electron was eventually found and measured over many experiments would create a picture graph similar to predicted atomic orbits.

Even though we can't know the exact answer, the range of possible measurement answers is known. When the probability of a particular quantum state of a discrete quantum system is known, it is described by a mathematical formula known as a *wave function*. Physicists use wave functions to describe and predict what will happen within a broad range of probabilities within a particular quantum interaction or property. The specific answer for one measurement isn't known, but the range of likely answers is, along with the probabilities of each possible answer. The wave function mathematically describes everything that we know about a quantum particle, including all its properties, what values those properties can be, and their likelihood of appearing in a measurement. For physicists, the wave function is a complete map of the quantum particle. And using the wave function, scientists can predict what will happen when different interactions occur between particles.

This probability principle of quantum physics is important because it means that we cannot predict what any specific quantum property state will be before it is measured. For example, as a very simplistic (nonquantum) illustration, suppose we were trying to determine whether a dog was black or white. From a quantum perspective, we could not determine ahead of time, before the measurement, what color the dog was. We could state what the possible answers are (i.e., black or white), and even state the likelihood of each answer being the observed answer based on predetermined math (say 50 percent of the time the dog would be white and 50 percent it would be black if we conducted the experiment many times). But we would have to wait until the dog was revealed to see what color the dog was actually observed and measured. Not only that, but what color the dog was when measured at any particular time would be random. That's quantum.

NOTE This peculiar quantum property frustrates classically leaning physicists and makes quantum mechanics seem so unsettling. With traditional physics, once you know all the involved objects, their properties, and their interactions, you would always be able to figure out the answer or outcome ahead of time. When the experiment ended or the result was returned, it would match the predicted answer from the previously stated math. "That's science! That's how it works!" Quantum

mechanics, on the other hand, says that no matter how well you know the math, the objects, and how they interact, you will *never* be able to predict the specific answer of any single experiment or quantum outcome. The best you will be able to do is predict the probability of different answers.

To complicate things, the "answer" a quantum computer gives us may not be the right one, and often today (in 2019) is not the correct answer. Remember, quantum answers are only answers within a given range of probability. But if we can run the quantum scenario and get the quantum answer a large number of times, we can see the right answer as it is returned more frequently and consistently over successive runs. Essentially, to get the true, right answer, the computation is run again and again until the probabilistic answer returned has so much statistical confidence that it must be the right answer.

As macroscopic allegory, suppose a six-sided die is weighted so that it will roll a one more often than any other answer (i.e., a "loaded" die). You know the die is biased, but you don't know on what side of the die. When you roll the die any one time, it may or may not come up with the weighted biased side of the die. But when rolled many different times, the one is more likely to come up more often than any other number, confirming the bias with high confidence. Let's say you plan to roll the loaded die ten times. The first time you roll it comes up a two. Now you have an answer, but only with 10% confidence (1 out of 10 rolls). You roll it a second time and you get a one. Now you have another answer but only with 10% confidence (1 out of 10 rolls). On the third roll you get a one again to have 20% confidence (2 out of 10 rolls). On the fourth roll you get a five. Then on the next five rolls you get a one, which gives you 70% confidence (7 out of the 10 rolls were a one). And on the last roll you get a three. Overall you would have four different possible answers, but the one side of the die was the answer the highest number of times and so any reasonable person would conclude that the one side of the die was the biased side. In a quantum computer, the problem and answer is likely run a lot more than ten times to get the highest possibility of the right answer.

For any single measurement, though, there is always the chance that it is the wrong answer. And as unsettling as that may sound, it's how all quantum answers work. You're already living and surviving in that world and have been since birth.

Uncertainty Principle

The *Heisenberg uncertainty principle* states that the more precisely the position of a quantum particle is measured, the less precisely its momentum can be known, and vice versa. Uncertainty also applies to other dependent pairs of quantum properties (known as *conjugate variables*) but not all properties. There may be pairs of properties that you can perfectly measure at the same time, just not all pairs. Some pairs are dependently linked in a way that prevents perfectly accurate measuring of both properties at the same time.

This isn't due to some flaw in the way humanity can or can't precisely measure something; it's a quantum law of nature arising from a particle's wave-particle duality and the probability principle. Quantum mechanics has proven, due to those laws, that you cannot precisely measure both position

or momentum about a quantum particle at the same time, and as you attempt to more precisely measure one quantity, the other side of the dependent pair becomes less accurate.

Let's use another macroscopic allegory, that of trying to measure the speed of a car. In the macroscopic, classical world, a car's speed is simply a measurement of its distance traveled in a particular time period. If the car traveled 100 kilometers in exactly 1 hour, you would say it averaged 100km/hr. But in the quantum world, when looking at quantum properties of very small particles, both the time and distance variables are not fixed at all. They are changing across a range of probabilities, and any single measurement can result in a different answer out of those ranges. That makes measurement tougher out of the starting gate.

Although with our speeding car example, there is a very similar allegory. If at any time along the car's path you took a measurement, the car could be going faster and slower than 100km/hr. It's really highly unlikely that any complex self-powered object would be traveling at exactly the same speed at all moments. For a car you would have to factor in wind resistance, surface condition changes, temperature changes, and the hundreds of factors within the engine itself that determine how much power and torque it is generating at any one time to get the speed at which it was traveling at any one second. Although, if the car was ultimately measured at traveling 100km/hr over the whole course, it probably traveled exactly 100km/hr more of the time than any other speed.

This is representative of the probability principle. At any point along the course if someone had held a radar gun, the car could have been going any number of speeds, but odds are that a car finally measured as having traveled 100 kilometers in exactly 1 hour was going 100km/hr at more points along the course than any other (although there is always a chance it was going exactly 98km/hour for half the course and exactly 101km/hr for the other half, but less likely).

The uncertainty principle says that as you go to more accurately measure the time involved in the speed, the less accurately the distance can be measured at the same time, and there is no way to fix it. In the quantum world there is no such thing as highly accurate speed. As a concept, it does not exist. It's a law of nature. To continue our speeding car allegory, suppose our judges wanted to be superaccurate, and to do so they decided to get the world's best flash photography to take a picture of the car just as it crossed the finish line of the measuring contest. In order to get the exact instant the car crossed the line, the shutter of the camera would have to open and close extremely fast. At that exact microsecond, the car would be "frozen" in time. In the picture of the exact instant it crossed the line, the car would not appear to be moving at all. The finish line camera can capture the exact moment the car crosses the line, but in that exact instant the car would not be moving (or moving very much). The camera, trying to get the exact moment when the timing was over, would have to remove the speed out of the measurement. And if you had another camera that was measuring the true speed of the car, it would not be able to accurately capture the exact moment when the car crossed the line.

To complicate matters, what is the line? Any line at a macro level looks like a straight line. But magnify any painted or drawn line and its individual, tiny undulations come out under magnification.

To be the most accurate you would have to snap a picture or click the stopwatch exactly as the car crossed the first atom of the painted line. And your eye and the camera's eye would have to click exactly when the car crossed that first atom, knowing that we can't even see when the car actually crossed the first atom of the line until a photo, at that instant, comes back to our retina or camera lens. And those processes depend on photons and the speed of light.

By the time the first recording photon made it back, the car would actually be past the actual first atom for those measurements to be made. And we know the first atom is made up subatomic particles—electrons, protons, and neutrons. To be the most accurate, you would have to stop the stopwatch or trigger the camera when the car met the first electron in the outer electron shell orbit, and according to quantum theory we don't know where that first electron will be and at any single measure it can be anywhere, and may not be where its highest-probability location would be. Ultimately you can't make a truly accurate speed calculation because the very attributes you need to get the most accuracy (i.e., the electron) is moving and the whole particle is moving as a wave along a wave function of outputs. As you try to get more and more accurate, you realize that you simply can't get a truly accurate measure of anything, much less conjugate pairs whose very definition depends on the other. Everything is moving at all times (even a rock is made up of moving electrons), everything is both a particle and a wave, everything behaves differently when measured versus not measured, every answer during a particular measurement is random, and it may not even be the "right" (highest-probability) answer. And with conjugate pairs, the accurate measurement of one value depends on the other, which by definition must get less precisely measured. In our example, the concept of kilometers/hour (i.e., position and momentum) doesn't really exist down at the quantum level. It just doesn't. That's the uncertainty principle.

You have to understand that the uncertainty in the measurement pairs isn't due to a lack of measuring equipment capabilities. Many people when first hearing about the uncertainty principle think it has to do with problems with the measuring apparatus not being accurate enough. They think it has to do with a flaw in the measuring devices. It doesn't. We could have the most accurate measuring equipment that could very accurately (to our human senses) measure time and distance, and it wouldn't matter. It's not the measurement that is imprecise; it's due to the (quantum) laws of nature that govern how accurately we can measure any quantum state that relies on two dependent conjugate variables. As we measure one side of the dependent pair more accurately, it simply isn't possible to measure the other side of the equation as precisely. In fact, it's a guaranteed inverse relationship.

NOTE The probability and uncertainty principles should not be misconstrued to mean that quantum mechanics and quantum properties cannot be mathematically accurate. The exact opposite is true. The math and outcomes from quantum mechanics are incredibly accurate, and with a proven confidence level unsurpassed by most other sciences. The uncertainty principle should also not be confused with the observer effect, which is discussed below.

Spin States and Charges

There are 12 *fundamental* (also known as an *elemental*) quantum particles that make up all matter in the universe. Fundamental particles, as best we know, can't be broken down to into smaller, whole particles. Be prepared if you haven't been introduced to them before. Some of these particles have some strange-sounding names. The fundamental quantum particles are electron, *muon*, *tau*, *electron neutrino*, *muon neutrino*, *tau neutrino* (all part of the *lepton* family), and *up*, *down*, *top*, *bottom*, *charm*, *strange* (the last six are part of the *quark* family).

These fundamental quantum particles make up all other subatomic particles. For example, every proton is made up of two up quarks and one down quark. A neutron is made up of two down quarks and one up quark. The electron, as an elementary particle, is not made up of anything. It's an electron, with no further subatomic particles to add up or break it down into. But electrons, protons, and neutrons make up atoms, atoms make up elements and molecules, and so on.

NOTE We can never be sure that we have discovered every elementary particle, or even that the existing leptons and quarks are elemental, although current science is very adamant that they are the lowest-common-denominator particles. But in history we previously said that about cells, atoms, and protons. So, who knows what we will discover as we try to finish the grand jigsaw puzzle that is our reality?

Each elemental quantum particle has a mass, charge, and a spin. Everyone understands what mass is, so let's quickly discuss the other two. A *charge* is the amount of current as compared to an electron. For example, an up quark has two-thirds of an electron charge and a down quark has a negative one-third of an electron charge. Because a proton has two up quarks and one down quark, this means the proton has a 3/3 (2/3 + 2/3 – 1/3) charge of an electron, or exactly equal to one electron. In most stable atoms, the number of protons in the nucleus equals the number of orbiting electrons for this reason.

Elemental particles also have a *spin*, which relates inversely to the number of revolutions a particle must make to return to its original orientation. All elemental particles have a spin of one-half, which means they must rotate twice to return to their starting orientation. Why am I teaching you about quantum charges and spins? Because the answers that quantum computers give us are often a result of charges and spins. As covered in Chapter 2, "Introduction to Quantum Computers," different quantum properties and states are used to provide answers in different types of quantum computers.

Quantum Tunneling

Quantum tunneling is an unexplained ability for quantum particles to pass through barriers, which classical physics said could not happen. The common macroscopic similar example is a ball sitting at the bottom of a hill or wall. Suppose a person is trying to throw the ball over the wall, but they do not physically have enough strength to get the ball over the wall. They try again and again with

no success. Classical physics, looking at the person's arm and body strength, says the person will never be able to do it. But then, for reasons that cannot be explained, the thrown ball sometimes ends up on the other side of the wall. Some theories say that the ball unexplainably just rises over the wall. Others say that the wall lowers for that one throw or that the ball is allowed through the wall without leaving an entry or exit point.

We don't yet know how it works or exactly when a subatomic particle will have success using it versus all the previous unsuccessful tries, but it does exist and is the basis for all known life. Tunneling is how our Sun generates heat and light using *thermonuclear fusion*. Tunneling is how a radioactivity element decays. Tunneling is the basis for photosynthesis, which supports most plant life on Earth, which then supports human life. Quantum tunneling is also involved in some types of quantum computing.

Superposition

Superposition is a quantum property that says a particle can exist in all possible states, until the state is finally observed and measured to give a single answer. For example, let's say that a particular math problem that you don't know the answer to can possibly be answer A or B. Superposition says that while the answer is in its quantum state before being observed or measured, it is both A and B at the same time. It's not A *or* B. It's both.

This is because, as discussed above, at any particular measurement of a quantum property, the measured property can be any possible answer. And the actual measured answer from any single measurement can randomly be any of those possible answers. In the classical world, everything is what it is. An A is an A. A B is a B. A single letter can't randomly be A sometimes and B sometimes. But in the quantum world that is exactly what happens.

Perhaps you've heard of Erwin Schrödinger's famous quantum cat conundrum. Schrödinger created a (thought experiment only) scenario where a cat was placed in a closed box with a capped bottle of deadly poison, a radioactive element, and a Geiger counter. The radioactive element could decay or not decay. Radioactive decay is a quantum event, and the moment when any particular atom of the element decides to decay is a random event. If the Geiger counter detects radiation (from radioactive decay), the Geiger counter would trigger the shattering of the bottle containing the poison, which would kill the cat.

Schrödinger created this thought experiment, which is an example of a quantum superposition process leading to an observable, macroscopic event, to demonstrate how weird superposition would be if expanded to the macroscopic level. Schrödinger was trying to show how absurd quantum mechanics, as described in his day, was. He didn't make the thought experiment to back up quantum mechanics. He did it to show how absurd it was, and to say that we didn't really understand what was going on. If he were alive today he would probably chuckle that his purposefully absurd thought experiment is actually the most commonly used enduring example of how quantum mechanics truly works, because that was not what he was going for.

Prior to opening the box and observing the cat, the superposition principle states that the radioactive element has both decayed and not decayed. The cat is both alive and dead. In the classical physics (or real) world, the cat at any particular point in time would be either alive *or* dead—one or the other at a particular point in time. What quantum physics has proven, at the quantum level, is that the cat (by extension of the radioactive decay) is both alive and dead at the same time, before being observed by opening the box—and not in some half-state where the cat is somewhat poisoned but not completely dead or fully healthy. No, it means it's both 100 percent healthy and 100 percent dead at the same time. What seems nonsensical at the macroscopic level is the absolute reality at the quantum level.

If you're going to understand quantum mechanics and quantum computers, you have to understand the concept of superposition. You have to break how you otherwise see and understand the world, because at the quantum level, the world does not act like you think it would. It took me a long time to understand the ramifications of Schrödinger's thought experiment. I figured the cat was alive or dead, and when we opened the box, it was one or the other and had been since some previous point in time. That's not what superposition says. Superposition, which has been proven over and over again, says the cat is both alive and dead, in both states, until finally observed and measured. Once the cat's "state" is measured, the cat is either permanently alive or dead, and from that point forward, this will be the measured result for that observation. This reckoning has flummoxed the greatest scientific minds who ever existed and still does. Yet experiment after experiment supports superposition as a reality at the quantum level.

Quantum mechanics and, by extension, quantum computers are instantaneously generating all possible answers all at once, and until the answer is observed and measured, the "correct" answer is all possible answers. Once we observe or measure the answer, only one answer becomes our permanent reality.

To complicate matters, as discussed earlier, no one can predict what the final observed answer will be. No one can say, "Surely, the cat is dead!" or "Surely, the cat is alive!" and always be correct—only that the cat is both alive and dead before being measured, and that the cat will be alive *or* dead when measured, but only within a particular probability of likely outcomes, and the specific outcome when measured is random among the possible choices. If someone's guess is right, it is only because they were lucky (or played the probabilities).

If this is confusing or hurting your head, we haven't even gotten to the weirdest parts yet. Hold on.

Observer Effect

In the quantum world, merely observing a quantum system changes it, although quantum physicists don't know or agree why. Like all of the quantum properties discussed in this chapter, decades of experimentation have shown that this property is real and accurate. Scientists aren't wondering if is true, only why or how it is true. For example, in every double-slit experiment, when scientists place a photon detector to measure which of the two slits a photon goes through, the photon always behaves only as a particle (and the resulting wave bands do not occur). If they turn off the detector,

the wave bands come back. It's as if nature sees the measurement happening, cares about it, and changes what happens. This may not be what happens or why, but it's how we describe what is happening based on our experimental observations and outcomes because we don't have a lot of other ways to communicate what we are seeing. We don't know yet what is happening.

It has led to many different competing interpretations. One interpretation says that it's impossible to observe a system without somehow interfering with it. For example, to merely observe something often requires light (i.e., a photon) or some artificially inserted equipment to capture the result, and those additions impact the possible quantum outcomes. With the photon example, the photon must have "hit" the thing being measured and bounced back to the detector (or our eyeball) for us to detect it, and that "hit" *must* cause some sort of interaction.

Another popular interpretation (the *Copenhagen interpretation*) says that when a quantum wave function of the many probable possibilities is finally measured and observed, the wave function "breaks down" (known as *wave function collapse*) into a final state. The observation creating the resulting collapse is the interference. To understand the Copenhagen interpretation, you have to again make sure you understand and believe in superposition, that any quantum answer or state is all possible answers or states at the same time prior to being measured. The act of measuring the quantum scenario reduces all the states or answers into a single final answer or state. Measuring it collapses all the concurrent, possible answers into one final, permanent answer (which may or may not probabilistically be the "right" answer and may not have been the same answer if measured or observed differently in any way).

The Copenhagen interpretation has the largest amount of support in the quantum world for explaining why observing something changes it. Although its inherent strangeness is why Schrödinger created his cat-in-the-box paradox thought experiment, Schrödinger wanted to point out exactly how counterintuitive the Copenhagen interpretation was to what we previously believed. Little did scientists know that the Copenhagen interpretation wasn't even closest to the hardest-to-believe explanation.

Another, the *Many Worlds* interpretation, says that all the possible answers from before the wave function collapse are now in another universe, and that each quantum collapse creates a number of new universes equal to all the possible answers in the probabilistic wave function before the collapse. Yowser! Now considering that there are likely trillions and trillions of quantum results happening every second, this would make a lot of universes in a terrifically large sea of *multiverses*. As crazy as this seems, some basic experiments have been performed that support the idea that we can't rule out quantum multiverses, including the one that led to this news story in 2019 (`https://www.iflscience.com/physics/quantum-experiment-sees-two-versions-of-reality-existing-at-the-same-time/`). Most people do not believe that the multiverse explanation is the right answer, but until the math rules it out, who knows?

The observer effect has a huge impact on quantum computing. We want our quantum computers to give us wonderful answers for otherwise extremely hard-to-solve problems, but they have to be manufactured and operate in a way that minimizes or utilizes the observer effect so that we can get accurate answers when we want them.

No-Cloning Theorem

A related principle, which is incredibly important to quantum information science, is the *no-cloning theorem*, which says quantum states cannot be directly copied. Remember, measuring a quantum state changes it from its quantum state to its classical, permanent state. And according to the observer effect, merely observing or measuring a quantum state changes it. This is not to say that "copying" can't be done, but it must be done indirectly. More on this in later chapters.

The no-cloning theorem has many implications for quantum computing. On the negative side, it means you can't back up a quantum state in the middle of a quantum computation like you can with a classical computer. It makes copying and error correction more difficult on quantum computers and networking devices. On the positive side, it is a great property for quantum cryptography and prevents many eavesdropping scenarios that are far easier in the classical world.

Spooky Entanglement

Now it is time to discuss the quantum property that is often considered the weirdest, the one that vexed Einstein to his dying days. Quantum particles can get "entangled" in such a way that when a quantum property (such as polarization, spin, momentum, or charge) on one particle of the pair changes, the property on the other particle pair also changes immediately in a predictable way, even if the two particles are separated by very long distances. We don't know why or how, which is how Einstein came to call it "spooky action at a distance."

NOTE Entanglement is a read-only, measurement process. Scientists know that when they measure a property of one particle in the pair, the other particle in the pair will have the same reading. But if scientists try to manipulate the entangled particles in any way to get a particular desired new state—say, change a particle property from a "0" reading to "1"—it immediately breaks the entanglement. We can read information but not transmit it. Implementing a particular desired state requires measuring the state, and measuring the state breaks the quantum properties.

In nature, entanglement is a natural process. It happens any time any quantum particle interacts with another quantum particle. It happens every time. Entanglement grows with each particle encountered. It cannot easily be stopped. Entanglement ends up creating multiparticle entities that now depend on each other. From a physics perspective you can no longer talk about any entangled particle as a single particle anymore. Every observation must be made from the outcomes of all the entangled particles involved in the same entanglement. In the real world, entanglement happens a lot and very fast. A quantum particle can easily entangle itself with billions of other quantum particles in millionths of a second.

Although quantum particles are always entangling on their own, for quantum-testing purposes, scientists intentionally create or entangle only small amounts of quantum particles. That's because

when you're trying to get at the truth of something in an experiment, less is usually more. Having to figure out something that is a result from the interactions of billions of particles just muddies the waters.

So in experiments where entanglement is desired, scientists will work hard to isolate the experimental environment to prevent any unwanted entanglement and create their own entanglements on much smaller scales. Experimental entanglement can be done a bunch of different ways, although one of the most common methods is to take a single photon of higher energy and split it into two photons of lower energy. There are several other common entanglement methods, but they are too technically complex to describe than is fitting for this book.

So far experimental, the entanglement must involve two very nearby quantum particles. Scientists up to now have not been able to entangle two particles that are far away from each other, although the distance is lengthening all the time. But once entangled, these two particles can be moved very, very far away from each other and still keep their entanglement bonding. Although as distance increases, the chances of entangled particles interacting with other entangled particles increases, making it hard to impossible for the scientists to measure what they wanted from the original, intended entanglement.

Irish physicist John S. Bell strengthened the theory of quantum entanglement in a series of uncontestable experiments whose description he published in 1987 in his seminal white paper titled "Speakable and unspeakable in quantum mechanics" (`https://web.archive.org/web/20150412044550/http://philosophyfaculty.ucsd.edu/faculty/wuthrich/GSSPP09/Files/BellJohnS1981Speakable_BertlmannsSocks.pdf`). Bell ruled out "hidden local variables," which Einstein had postulated were another possible, more likely, explanation for entanglement. Bell proved there were no hidden local variables, which significantly strengthened entanglement theory and all of quantum physics.

Since then, his experiments have been repeated with the same success each time and on different quantum particles. Spooky entanglement has been demonstrated in photons, electrons, neutrinos, and even larger molecules such as "buckyballs." Quantum entanglement has even been demonstrated in macroscopic objects, like diamonds (`https://news.yahoo.com/two-diamonds-linked-strange-quantum-entanglement-190805281.html`). Not that quantum physicists need pictures to believe or prove anything, but in July 2019, scientists were able to capture the first picture of entangled particles (`https://phys.org/news/2019-07-scientists-unveil-first-ever-image-quantum.html`), which thrilled scientific and nonscientific minds alike.

Decoherence

The last quantum property we will discuss in this chapter is *decoherence*. It is extremely important in quantum physics and computing. It is something we both want and want to avoid (until the right time). When a quantum particle or system is in an easy-to-see set of quantum states we say that it is cohered or in *coherence*. We can easily see the results of its quantumness, which is operating along a

wave function with all the probable answers. Without extreme environment isolation, any quantum particle or system will begin to interact and entangle with other quantum particles. In fact, billions and billions of interactions within microseconds. This happens in even what we might think is a empty void. For example, when scientists create an artificial vacuum inside a box with no light or other intentional quantum particles in it, the apparatus used to create the vacuum will leach into the void. It's unavoidable. Again, without extreme conditions, this happens very often and very fast. With the best of conditions it still happens. It cannot be stopped from happening.

Each unwanted interaction causes entanglement, and now scientists trying to follow one or a few particles or properties must begin dealing with results that are from a more complex, multiparticle amalgam that they usually did not desire. Their original particle(s) are there but can easily be lost among a sea of other entangled particles, and in any case, they cannot easily figure out the impact or result of the original particle(s) they were watching and wanting to measure.

Imagine you wanted to follow a single drop of water and it dropped into an ocean. Or you wanted to follow a single photon out on a beach on a sunny day. The drop of water would still be in the ocean, but now immediately dispersed among trillions and trillions of other drops. You could possibly still follow the original drop, but it would be hard. You could possibly keep track of your original photon on the beach, but not only is it lost among a trillion other photons, but it is interacting with the other photons and other particles, both micro and macro (e.g., dust, air, wind). For all practical purposes, after just a few interactions, it would be difficult for any single particle to be tracked and to figure out what all the other entanglements caused or didn't cause.

Because of this, for quantum experiments and inside quantum computers and other devices, the internal structures must be highly isolated from the outside world. Quantum scientists want to prevent as much unwanted entanglement from happening as is humanly possible. Barren surfaces using a single stable element, cold temperatures, and shielding against the outside world are all commonly used. But when the scientists or machinery lose their ability to track the original particle(s) or their property/properties and figure out the originally desired outcome (which will always eventually happen no matter what), the quantum particle or system is considered decohered or in decoherence. It's important to note that the quantumness of the particle or system didn't change into something else. It didn't become nonquantum/classical. It just became too difficult for our meager minds and equipment to track and understand in a meaningful way.

Sometimes we need decoherence. In quantum information science, when we want to get a quantum answer we can write down and call a result, we have to measure it, and measuring it entangles and changes it. Whatever the measurement device is, it's also made up of quantum particles and properties and must interact with the particle or property being measured. Even if the measurement only involves light, light is made up of photons, and in order for the photon to capture the result and report it back, it must "hit" the particle and bounce back. Now that photon is entangled with the thing it measured. So, measurement alone will decohere a quantum system. It didn't suddenly change the quantum state into something not quantum, it just starts to immediately add measurement complexity.

But to record a quantum result of a particular experiment or computation, we must measure it. So, we want to measure it when and where the decoherence is controlled and minimized until our measurement apparatus is the thing decohering it. We need to measure and decohere it so we can get a final measurement and answer. We don't want the answer to be A sometimes and B sometimes. We need a permanent outcome value to record. Can you imagine if every time we needed an answer, we could just say "it's a range of all possible answers across a probability spectrum" and leave it at that? We couldn't just say the car is going 100km/hr. We'd have to say, well, it's going a speed somewhere between 0 and 200 km/hr (or whatever the maximum possible speed is) and here are the probabilities. It would be a nuts way of describing the world especially when everyone understands that a car recorded as going 100 km/hr was likely not going 100 km/hr at all times. In order for us to record answers we pretty much just want the most probable "right" answer, and not some spectrum of answers along a mathematical wave function. So, we want to intentionally decohere the system at only the time of needed measurement. We want to avoid decohering the system before the measurement, and once we have the measurement we need, it can decohere further all it likes. Although scientists would also like, and are trying, to get multiple measurements done without decohering a system. One of the biggest struggles, if not *the* biggest challenge, in quantum information sciences is to protect a system from premature decoherence until final measurement is needed.

There are many other central quantum mechanics properties, principles, and theories, such as contextuality, that we could cover, but what we have already discussed is a great base for discussing how quantum computers work in Chapter 2.

Quantum Examples in Our World Today

Although quantum mechanics mostly happens at a subatomic level, none of our reality could be possible except for its very real existence and impact at our real-life level. Quantum mechanics makes the Sun shine, is the reason all matter holds together, and is the basis for most of the things we see at the macroscopic level. When you look at a stove burner glowing red hot, that's only possible because of quantum effects. Quantum mechanics is responsible for our computer microprocessors, transistors, resistors, and all integrated circuits. Disk storage and network communications are only possible because of quantum mechanics. Your Wi-Fi connection works only because of quantum properties. Here are other macroscopic realities that are only possible directly due to quantum mechanics:

- Fiber-optic cables
- Lasers
- Superconductivity
- Superfluid liquids
- Atomic clocks
- Magnetic resonance imaging (MRI)
- And don't forget the whole reason for this book, quantum computers and quantum cryptography

All of these wonderful things, and all of reality, only work because of all the incredible and strange quirks of quantum mechanics. I'll cover more of how quantum mechanics will help us in Chapter 5, "What Will a Post-Quantum World Look Like?"

For Additional Information

The field of quantum physics is huge. The topics covered in this chapter truly represent the bare tip of the iceberg. Each summarized topic has been covered by dozens of white papers and books, and sometimes hundreds of papers and books. No one book, white paper, or online media tutorial can do quantum mechanics justice. Anyone interested in learning more should just pick a few resources to start with and dive right in. It will often take at least a few of these resources, well read or viewed, before even the basics begin to sink in.

With that said, here are some of my personal favorite resources any quantum physics newbie can start with:

Aaronson, Scott (2013). *Quantum Computing Since Democritus*. Cambridge: Cambridge University Press.

Bell, Philip (2018). *Beyond Weird: Why Everything You Knew About Quantum Physics Is Different*. Chicago: University of Chicago Press.

Orzel, Chad (2009). *How to Teach [Quantum] Physics to Your Dog*. New York: Scribner.

Orzel, Chad (2018). *Breakfast with Einstein: The Exotic Physics of Everyday Objects*. Dallas, TX: BenBella Books, Inc.

Dr. Mark G. Jackson's Articles for Popular Audiences. http://physicsjackson.com/articles/

Quantum Physics Blog. https://www.techbubble.info/blog/quantum-physics

Scott Aaronson Blog. https://www.scottaaronson.com

Dr. Scott Aaronson's Democritus online courses. https://www.scottaaronson.com/democritus/

YouTube. *Quantum Theory—Full Documentary HD*. https://www.youtube.com/watch?v=CBrsWPCp_rs

YouTube. *Quantum Physics for 7 Year Olds*. https://www.youtube.com/watch?v=ARWBdfWpDyc

YouTube. *Neil deGrasse Tyson Explains Quantum Entanglement*. https://www.youtube.com/watch?v=q8CQAOwi2RI

If you are interested in learning more about quantum mechanics, go to YouTube and/or Amazon and just type **quantum physics mechanics** to see hundreds of choices.

Summary

If this is your first introduction to quantum mechanics, I hope I have been successful in showing you its wondrous weirdness. Feel free to return to this chapter and reread it as you gain more understanding in later chapters of quantum mechanics. Quantum computers use these incredible quantum properties, including entanglement, uncertainty, and superposition, to provide us with answers that just aren't possible with traditional, binary computers. Chapter 2, "Introduction to Quantum Computers," will discuss how quantum computers and devices work to deliver us incredible answers and solutions we can rely on, along with what is currently the state-of-the-art.

2 Introduction to Quantum Computers

Quantum computers, devices, and software use the peculiar properties of quantum mechanics discussed in Chapter 1 to manipulate, create, and process data. All those strange and wondrous quantum properties, such as superposition and spooky entanglement, are on full display in quantum information science. Chapter 2 will cover quantum computers, including how they vary from traditional computers, the different types of quantum computer architectures, and many of the companies that manufacture them.

How Are Quantum Computers Different?

This section will discuss how quantum computers differ from traditional binary *classical* computers. It will begin by exploring the primary difference between bits and qubits.

Traditional Computers Use Bits

Traditional computers use binary digits (0s or 1s) to store, transfer, and manipulate data. A *bit* (binary digit) can possibly be only one of two states: it is either a one or a zero. It is either on or off. And it can only ever be one thing (i.e., state) at one time. The underlying binary nature occurs because the manipulated particles (e.g., usually electrons, but they can be photons and other particles) are being manipulated as whole particles. Since the beginning of digital computers up until the invention of quantum computers, it was the only way we could manipulate digital information. Quantum computers allow us to manipulate particles in a nonbinary way, using their quantum properties.

Traditional computer chips work only because of underlying quantum mechanics, but they can only manipulate and move whole electrons around in a binary way between the various logic gates and positions on the impacted doped (i.e., intentionally embedded with desired impurities) semiconductors. The measured underlying electrons they manipulate are in one of two whole states, which equate to a 1 or a 0. Traditional computers don't measure spin, polarization, or any other possible quantum properties.

Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto, First Edition. Roger A. Grimes.

Because a bit can represent only one of two states at any one time, we can easily calculate how many bits it takes to represent a particular amount of information. For example, 1 bit can possibly be two different bits of information (i.e., 0 or a 1), but it will represent only 1 bit of information before, during, and after being measured. Two bits can be four different possible pieces of information (i.e., 00 or 01 or 10 or 11), but will represent only 2 bits of information when measured. Three bits can be eight possible different pieces of information (i.e., 000 or 001 or 010 or 100 or 010 or 011 or 110 or 111), but will represent only 3 bits of information when measured, and so on. Each additional binary digit gives *exponential* growth of possibilities (2^4, 2^5, 2^6, etc.).

Early computers were directly programmed by turning individual bits on or off using a physical manipulation. They had physical, electronic "jumper" cables, which were or weren't plugged into the computer to make or not make a specific pathway connection. One of the longtime computer stories is that the term *computer bug* came about because of real bugs eating parts of the cables and causing programming bugs.

The long, floppy jumper cables were replaced with built-in mechanical switches and paper programming "punch cards," which essentially manipulated internal mechanical switches to change the computer's binary pathways. Even today, many computer devices have remnant "jumper" switches that a user can physically manipulate to determine binary choices such as "on" and "off" for particular computer pathways and decisions.

The mechanical switches were replaced by electronic switches, which eventually led to transistors, resistors, and microprocessors, which are at their simplest level simply cramming as many binary "logic gates" into the smallest spaces possible. But no matter how many binary switches we can cram onto a circuit board or into a piece of silicon, everything is being done in a binary way. It's just that more binary pathways fit into a smaller space.

The lowest-level computer languages, such as assembly languages, are only one abstraction layer removed from moving bits around in a computer's microprocessor. For example, the assembly language instruction `MOV AH, 1` is instructing a computer's microprocessor to move a binary value of 1 into the AH register (registers are microprocessor memory areas that store and help manipulate data).

Every binary computer programming language eventually gets broken down into binary instructions, which then physically manipulates the computer's microprocessor's electrons among predefined *Boolean* logic gates (AND, OR, NOT, etc.). The classical computer systems are built, from top to bottom, on binary manipulation and storage of binary data, and they have advanced the world in immeasurable, extraordinary ways. Underneath all the binary behavior was quantum particles and behavior, but it was not being used to store and transfer information or to compute.

There are, however, limits to what binary computers can do. There are things that binary computers simply cannot do at all—or not do well enough. Or they are not fast enough to be as useful as we need them to be. For example, binary computers are not fast at factoring mathematical equations involving large *prime* numbers. Large prime numbers are often used in digital cryptography (this is explained more in Chapter 3, "How Can Quantum Computing Break Today's Cryptography?"). Prime

numbers are any whole number larger than 1 that cannot be divided by any number other than itself or 1 and equal a whole number (with no remainder). Successive prime numbers starting above 1 are 2, 3, 5, 7, 11, 13, 17, 19, 23, and so on.

Binary computers can factor equations involving prime numbers and even factor equations involving large prime numbers that humans wouldn't consider doing. But to factor equations involving very large prime numbers, like the ones involved in computer cryptography, binary computers would need hundreds to millions of years.

Early on, to confirm a prime number candidate, binary computers essentially had to examine every possible number and then check it against every previous number before the candidate to confirm whether the prime number is prime. Some computer scientists call this "guess and check." It's not a very effective method for confirming prime numbers. There have been some great mathematical advances and algorithms that allow classical computers to confirm prime numbers with less effort than "guess and check," but even they aren't powerful enough to allow the traditional computers to easily factor an equation made up of two very large prime numbers.

Other times, binary computers simply can't do what is asked of them. Something seemingly as simple as generating a truly random number or string of characters is physically impossible for a classical computer (this is also discussed more in Chapter 3). The way they are constructed and the way they are able to compute make it impossible. Classical computers attempt to simulate very random numbers, but that's the best they can do . . . simulate. They aren't truly random, and that causes problems with programs that are reliant on truly random numbers. More on this in Chapter 7, "Quantum Cryptography."

There are myriad problems that traditional binary computers running at any possible speed cannot easily solve in a reasonable time frame or just cannot solve. It's just physics . . . well, physics before the promise of quantum mechanics and quantum computers.

Quantum Computers Use Qubits

Quantum computers use qubits instead of bits. *Quantum bits* (also shortened to *qubits* or *qbits*) have the near-miraculous quantum property of superposition. A single qubit is still a two-state system (1 and 0), but because of superposition it can be all possible states *at the same time* before measurement. In quantum computers, a qubit can be both a 0 or a 1, or a 0 and a 1. This is due to the wave function of a quantum particle and its inherent set of probable possibilities.

A common allegory is to liken a qubit to a coin that is being used to decide which team kicks off first in a football game. As the referee defines the coin for "heads" and "tails" sides of the coin, they tell the person "calling" the coin to call the result as heads if they see the heads side of the coin or to call the result tails if they see the tails side of the coin. But after the referee flips the coin into the air, instead of it flatly landing on a heads or tails side it lands exactly on its thin edge and stays there wobbling for a few seconds. There is a potential, before the coin is called (i.e., measured), for the calling person to see all three sides before it falls to one flat side. The coin's ultimate final answer

when it finally falls might be heads or tails, but at the moment of its edge-standing it could be read as both at the same time (i.e., superposition). This is not a perfect allegory, but it does communicate the idea of superposition versus the final, measured answer.

A bit is a 1 *or* a 0. A qubit is both a 1 *and* a 0 before the ultimate measurement, and because of this, its bit-state is exponential to itself and to every other additional qubit added to it. A single qubit can be two states at once (i.e., 0 and 1), and a 2-qubit system can represent four states at once (i.e., 00 and 01 and 10 and 11). A 3-qubit system can represent eight states at once (i.e., 000 and 001 and 010 and 100 and 010 and 011 and 110 and 111), and so on. A 3-qubit system is represented in Figure 2.1.

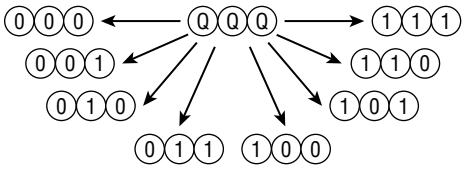

Figure 2.1: Representation of a 3-qubit system

An important point to understand about qubits is that any similar number of bits in a binary system has a similar number of possible states but is only one state at any one time. A binary coin can never land on the edge, so to speak. Thus, a 3-bit system can return only one state/answer at one time. A 3-qubit system can have all 8 states all at the same time involved in solving a problem. A 3-qubit system is an 8-fold state improvement over a 3-bit system. A 4-qubit system is a 16-fold state improvement over a 4-bit system, and so on. Now imagine you have thousands of qubits, each containing a simultaneous two-state system, and because of superposition they are all the possible states of thousands of qubits at the same time. The speed and logical improvements, as you can imagine, are pretty fantastic.

The best speed comparison example between bits and qubits I can relate is one that I heard once (but I don't remember from who). Imagine that we want to solve for every possible move on a chessboard. A chessboard has 64 squares (32 black and 32 white). If each possible move was represented by a grain of rice, the resulting number of grains of rice needed to represent all possible chess moves would result in a rice-mountain the size of Mount Everest. And the amount of rice needed to represent something like a 2048-bit prime number factorization would equate to 1,985 Mount Everests built of rice. Now, you can understand the magnitude of the issue. A traditional binary computer will either take a very, very long time to calculate the answer or never solve it. A quantum computer, with a few thousand qubits, can generate the right answer in under two minutes. That's the power of quantum computers.

Qubits are possible because the base objects being used by a quantum computer are the quantum states of quantum particles. A traditional binary computer may use electrons (or even photons), but they are created, manipulated, and measured as binary objects or states. They are either 0 or 1, either

on or off. A quantum computer, using quantum particles and measuring those particles' quantum states, sees all the represented quantum states. Thus, when measuring an electron's state, it sees all possible states of all possible quantum properties. When measuring a particle, it sees all possible states, like charge, spin, polarity, and so forth. The base state of a true quantum computer is that it can represent all possible particle states at once, not just a binary state. In a quantum computer, the logical comparative unit is known as a *quantum logic gate* (or *quantum gate*) versus a *classical gate*. Quantum gates are inherently capable of more options and complex problems than classical gates.

NOTE Even though superposition is all possible states at the same time, this doesn't mean quantum computers can show all possible answers for all possible problems instantaneously. All computers, including quantum computers, still have to compute and run programs to solve complex problems. They don't just get every answer immediately. There are necessary computations and algorithms that have to be followed to get the ultimate answer. But it is accurate to say that because of qubit superposition, quantum computers are likely to solve many types of problems far faster than traditional computers, and in some cases, the solutions will be provided so quickly as compared to traditional binary computers that it seems instantaneous.

Qubit Growth over Time

The first quantum computer with 1 qubit was created and demonstrated in 1998. Since then, we have seen the number of qubits represented in a quantum computer grow over time. Here is a basic qubit advance timeline by year as of this writing, based on various vendor claims:

- 2000: 5- and 7-qubit computers
- 2006: 12-qubit computer
- 2007: 28-qubit computer
- 2012: 84-qubit computer
- 2015: 1000-qubit computer
- 2017: 2000-qubit computer

NOTE As will be discussed in greater detail later in the chapter, not all qubits are alike. Some of these large qubit claims have some widespread doubt.

You can see a more complete list of the current number of known qubits by vendor and quantum computer type here: `https://quantumcomputingreport.com/scorecards/qubit-count/`. As you can see, we are growing the number of quantum states represented in a quantum computer somewhat exponentially over time. There is no reason not to believe that this exponential-like growth won't continue in the future just as the number of integrated circuits in a defined space doubled every two years (as stated and predicted by Moore's law) in traditional binary computers.

This is not to say that qubit exponential growth is guaranteed. There are some big issues to overcome as quantum computers scale, over both the near term and long term. But it's likely that quantum computer scientists will solve them and continue to add more and more qubits over time. When I hear the qubit growth critics who think we have already reached some far smaller arbitrary limit or that state that future advances will take decades, it reminds me of all the critics who said that more and more classical gates could not possibly be added to traditional microprocessors. Each year, the "experts" would tell the world how we had finally reached technical limits for putting such and such a number of classical logic gates on a single computer chip, in a given space—and then the next year chips would have more. Chip fabricators developed or improved some technology that allowed them to cram more stuff in the same space. It was always something that the critics did not consider in their calculations. Today, we have extremely fast 32/64-CPU cores that fit on a single microprocessor die. Now, quantum critics might have more expertise and math for why adding increasing numbers of qubits in the same space will become impossible one day, but so far we're adding more on a fairly regular basis. We are not close to hitting the qubit wall yet. We are just starting to lay the first foundational layer.

Not All Qubits Are Equal

It is important to recognize that simply having more qubits does not directly correlate to faster and better quantum computers. Having more qubits is definitely a good have, but not all qubits are equal and quantum computers' ability to solve something is determined by more variables than just the sheer number of qubits. Some types of quantum computers, as we will cover later in the chapter, are better at solving certain types of problems. Some quantum computers, no matter how many qubits they will get, cannot solve some types of problems.

A similar comparison might be done with race cars. We can create really fast cars that can go nearly 500 mph but only in a straight line for a few minutes. Those cars, shaped like horizontal rockets, must be used on miles and miles of very flat, straight surfaces. They never could compete with NASCAR cars on an oval race track and last for hours and hours. Each type of car is designed to win a particular type of race.

It's the same with quantum computers. Each major type of quantum computer is maximized to solve particular types of problems. And quantum computers designed to solve a wide range of problems won't as quickly solve particular types of problems that certain types of quantum computers focus on to the exclusivity of other types.

Quantum Power Is More than Qubits

A higher number of qubits has the potential to make a specific quantum computer faster, but it doesn't guarantee it, just as higher horsepower doesn't always mean a car will win a race. A race car's success depends on everything that goes into making the car move forward faster than the other cars, including engine, fuel injection, tires, transmission, and torque conversion. It's the same thing with quantum computers. Increasing the number of qubits can rarely hurt, but many other factors can be speed-limiting.

IBM, long involved in and leading the quantum computing field, figured this out early on as it increased the number of qubits in its own quantum computers and as other competitors showed up on the scene. IBM recognized the need to develop a method anyone could use to independently compare the power and speed of various quantum computers. IBM's answer was a metric it calls *quantum volume*, which equates to the amount of quantum work a particular type of quantum computer can do in a given time period. According to IBM, the items involved in determining quantum volume are a variety of factors, including the number of qubits, connectivity (between qubits and other components), and coherence time, plus accounting for gate and measurement errors, device cross-talk, and circuit software compiler efficiency.

The Institute of Electrical and Electronics Engineers (IEEE) proposed an independent standard that attempts to measure, benchmark, and equate quantum performance: PAR 7131 (`https://standards.ieee.org/project/7131.html`), the Standard for Quantum Computing Performance Metrics & Performance Benchmarking. It includes many of the same quantum variables mentioned in IBM's quantum volume but also lists "gate times, generation and readout capabilities, and gate fidelities" among its criteria.

In a functioning computer, the hardware is only one part of the performance equation. Quantum computers have software just like traditional computers. The underlying software, whether it is represented as firmware or software, is required to find the solution to a problem but also adds overhead cycles to overall performance.

Some scientists have proposed creating "quantum puzzles," which each competing quantum computer could solve and then record the time it took to solve. The puzzle would have to be different for each major type of quantum computer (they are optimized to solve different types of problems). Unfortunately, I have no doubt that any vendor with a quantum computer coming up short in any benchmark comparison would have a million reasons why their quantum computer was unfairly shorted in the contest. The same thing has always occurred with traditional binary computer benchmarks. But speed benchmarks should not be completely discounted. They do serve a valuable service in some scenarios.

The key point to understand is that the number of qubits alone does not equate to how wonderful or fast a quantum computer is, although having more qubits is never a bad thing. Just like in our race car example, starting with a certain amount of horsepower is a requirement if you want to seriously compete with the other fast cars. But horsepower alone will not win a race.

Quantum Computers Are Not Ready for Prime Time Yet

As I write this in 2019, no existing quantum computer is faster than any traditional binary computer. It isn't even close. Your laptop likely has more raw performance. For now, the best that quantum computers can do is demonstrate quantum properties on a small scale that can possibly solve problems in the future that traditional binary computers cannot. Conversely, traditional binary computers can often emulate or simulate quantum solutions, often better than today's quantum computers. But it's key to understand that even though binary computers can simulate quantum mechanics,

they aren't quantum, and one day they will likely be bypassed by native quantum computers. The question is when.

Quantum Will Reign Supreme Soon

At some point, quantum computers will be able to solve problems that traditional computers cannot solve at all, as well as solve problems that traditional computers can solve but do it significantly faster. That moment is known as *quantum supremacy* (or *quantum advantage* as coined by IBM). We appear to be very close.

Many different companies have thought, and publicly announced, that they had or were near quantum supremacy. Google, Intel, and the Chinese government/companies have announced that they either have achieved quantum supremacy or are nearly there. IBM announced in 2017 that it thinks, based on its prediction of doubling quantum volume every year, it will be there by 2020. Perhaps some quantum computer in the world will have reached quantum supremacy before this book is published. Conversely, there is even a chance that some unforeseen technology blocker prevents quantum supremacy from ever being reached. Or perhaps binary microprocessors (the unstated denominator in the comparison) will have a stunning technological jump that deadens the advances made by quantum computers.

The smart money seems to be on quantum supremacy happening within a few years. For many years, the world's best computers couldn't beat the best human players at chess. That was until IBM's Deep Blue computer beat chess champion Gary Kasparov in 1996. For many years, a computer couldn't beat a human champion *Jeopardy* player. That was until IBM's Watson did it in 2011. Quantum supremacy is seen as happening the same way. There's a lot of marketing hype, but eventually it will happen. It's not if, but when.

ALTERNATE QUANTUM SUPREMACY SCENARIO

An alternative quantum supremacy scenario is where quantum computers begin to solve problems that cannot be logically solved by binary computers but are not necessarily more computational powerful. Quantum computers, in theory, can efficiently solve any classical problem (although not always as efficiently), but the converse is not true. Classical computers cannot solve all problems that quantum computers can, at least in any practical timeline. An alternative quantum supremacy scenario possibility emerges where quantum computers aren't "faster" than binary computers; they are simply capable of solving problems that binary computers can't. Even then, most quantum observers expect quantum computers to become more capable and faster in due time.

Quantum Computers Improve Qubits Using Error Correction

Theory-wise, many quantum computing experts say that we can achieve quantum supremacy with 40 to 50 perfect qubits, or at most 100 perfect qubits. One estimate says that quantum computers "could map all the information in the universe "from the Big Bang"? forward" using just 300 perfect qubits. Unfortunately, so far perfect qubits are eluding us. They are full of errors, especially at scale. This section will cover some of the ways quantum computing scientists are attempting to make better qubits, including improving coherence times, supercooling, check qubits, and increasing the performance of other components.

Premature Qubit Decoherence

Without a doubt, the single biggest challenge to quantum supremacy is premature qubit decoherence. As defined in Chapter 1, *decoherence* is a quantum particle's states going from its easy-to-see superposition (i.e., multistate) to its finalized, measured, single, classical state before all the eventual entangling makes getting useful information impossible. Once decoherence has happened, premature or not, it cannot easily be reversed. Try to pull a specific drop of water back out of the ocean or pick out a single photon on a bright, sunny day, and figure out all the impacts of its past entanglements.

With "perfect" quantum computers, decoherence would happen only and exactly at the point an "answer" is needed and measured. It would always be on purpose. A qubit would stay in its cohered state as long as is needed for the quantum calculation and then, and only then, decohere when measured.

NOTE Suppose a qubit has a 1 state, which the computer must hold to perform a calculation or return a result. How long the qubit will stably maintain that 1 is called its *coherence time*. It is often measured in milliseconds, but some quantum computer types can last seconds to many minutes. The first order of business for most quantum computer makers is to increase coherence time. Increased coherence time means fewer errors and more time to compute and return answers.

Today's quantum computers are full of premature quantum decoherence and just outright errors. Both can occur because of qubit construction, heat, radiation, noise, vibrations, faulty gates, faulty measurements, faulty initial state preparation, background nuclear spin, and myriad other events. Essentially, any interaction with the external world is a threat. Reducing errors and noise is the number-one quantum challenge and has spawned a field that has its own name, *quantum error correction*. Error correction is attempted using a bunch of different schemes, including both quantum and classical methods. No one has perfected it yet, but every quantum computing vendor is trying.

Error rates are usually reported as a ratio of quantum operating time to decoherence time. The *quantum error threshold theorem* holds that any quantum system that corrects errors faster than it creates them is usable. As the number of qubits increases, so too does the natural error rate. For quantum computers to be very useful, the error rates need to be below 1 percent—really below 0.001 percent. As a comparison, a classical CPU can do trillions of calculations without an error. In the

quantum world, we are just hoping to get the error rate down to one error per thousand calculations. Achieving that, along with some good error correction, will allow some serious quantum work. In 2019, we are not there yet.

NOTE Classical computers do make errors. The difference is that it usually takes larger events for the errors to occur in the classical world. As an allegory, in the classical world imagine how big of a wind gust it would take to flip a penny (i.e., bit) sitting flatly on the ground. But in the quantum world it takes only a small breeze to flip a penny sitting on its edge (i.e., qubit) to make it land flat on the ground.

Quantum computing scientists are trying to reduce quantum errors by identifying and fixing the most significant rate-limiting components or issues. Common solutions include improving coherence times, strictly isolating quantum components from the external world, using supercooling, using check qubits, increasing the performance of other components to outperform the errors, and using quantum entanglement as error correction.

Improving Coherence Times

One method of error correction is to improve the quality and control of each qubit's cohered state to be longer than the needed calculation time by improving whatever is the error-limiting component of the quantum computer, such as quantum gate noise or connectivity speed. The longer the qubit can stay cohered, the fewer errors it is likely to have.

Environmental Isolation

Since the beginning of classical computers, computer scientists have recognized the benefits of operating computers in controlled environments that isolate them from the extremes of the outside world. All computer rooms are temperature-controlled (heat is the enemy of all computer components), air-filtered, humidity-controlled, cleaner environments. You don't find a lot of computers running very long left in the outside world exposed to normal weather and events.

But most of today's classical computers have matured physically to a point where they can survive operating in normal weather environments, unless they are exposed to truly extreme conditions (or dropped in water, stepped on, etc.). For example, the most popular types of computing devices in existence work every day in the real world. Most laptops, pad devices, and personal computers operate just fine outside a very controlled computer room.

Quantum computers are not there yet. They are still very fragile machines and must be protected from not only weather extremes but even very normal conditions. In fact, most of them operate most efficiently by running (at least the quantum components) in particular types of weather extremes, such as very cold temperatures. They must be protected against radio waves, normal background radiation, electromagnetic interference, loud sounds, and vibrations. But most quantum scientists

do envision a day when, as is the case with classical computers, quantum computers will be built in a way so that they are much more resilient and in less need of special environmental isolation.

Supercooling

Most quantum computers must supercool their qubits (and other nearby components) to near zero degrees Kelvin (0K is near –460F) to minimize premature decoherence issues. Warmer temperatures are shown to allow more errors and to emit more unwanted stray quantum particles with almost all the quantum computer technologies, even with the few types that supposedly don't need ultra-cool temperatures. They may not "need" super cold temperatures, but even they seem to perform better with fewer errors in lower temperatures. To do this, most quantum computers (actually just their qubit chips and closely related apparatus) are supercooled using external *cryogenic* or *dilution refrigerators*.

You'll see most quantum computer manufacturers bragging about how cold their cooling is: "Our temperature is 200 times colder than the far reaches of the universe!" Many will state that their operating temperature is less than hundreds to thousandths of a single Kelvin (0.02K to 0.01K is often touted). Others, trying to humbly brag about their computers' ability to resist errors at relatively higher temperatures, will talk about their computers running at room temperature or a slightly higher value of Kelvin, such as 4K to 20K (20K is still –424F).

There has long been a race in physics to see who can create the coldest temperatures, but no one has yet reached *absolute zero* (0K), and it's likely impossible. But were it possible, at absolute zero all particles' energy and momentum would be stopped to the bare minimum physical possible (something called *zero point energy*). At absolute zero, most moving and even solid-state things would fail to function as usual. With that said, the quantum computer of the future is likely to be able to withstand higher temperatures, possibly equivalent to the requirements of today's classical computers, because requiring supercooling is expensive and limits where and how they can be used.

But for now, lower temperatures usually improve coherence times and decrease errors. Lower temperatures also create a quantum property called *superconductivity*, which is zero or near-zero electrical resistance in materials cooled below critical temperature thresholds. Superconductivity increases electron flows and allows stronger electromagnetic interactions. Many quantum computers use superconductivity to create their qubits, and it is used in many other applications, such as superfast maglev trains, medical equipment, and super-strong magnetics, to name a few.

Repetitive Calculations

One way to fight errors is to run the same calculation at least three times and store the results in the classical world. After running the same calculation multiple times, the computer will look at all the stored results and, if there is a disagreement, take the result that appears more than the others. However, this error correction method also slows the computer down in direct relation to how many duplicative operations are used, and there is no guarantee that the most represented answer is actually the right answer.

Using Quantum Entanglement for Error Correction

In the classical computer world, if a lot of errors are to be expected, a bit's value can be copied and stored to one or more "backup" bits at the same time. If there is a disagreement among bits, then the most popular value is taken. But in the quantum world, because of the no-cloning theorem and the observer principle, qubits cannot be copied directly while in their quantum states. Instead, entanglement can be used to create indirect copies, although entanglement bonds are sensitive and easy to lose track of as decoherence happens.

Check Qubits

Another error correction method uses additional qubits as *check qubits*, which is similar to how check bits are used in the classical binary world. The check qubits are implemented in some sort of logical checking method, which detects errors and helps correct them. For example, an additional check qubit is used to ensure that the resulting *qubyte* adds up in a particular way—for example, the computer can add a 0 or a 1 to the check qubit position of the qubyte to make sure the sum of all qubits ends up as an even value. If the quantum computer detects a negative sum returning from a qubyte sum, an error can be declared and the quantum operation can be repeated. You might be familiar with a similar popular error correction classical technology called Redundant Array of Independent Disks (RAID) in the binary computer world. The big problem with this sort of simple even-or-odd checking is that there is no way to ensure that errors don't happen in a way that aren't detected only by even or odd values. But it's better than nothing and more complex, but similar error-checking scenarios have been used to make our existing binary world incredibly reliable. Quantum computer manufacturers are using similar quantum error-checking methods to make quantum computers more reliable.

Error-checking qubits added to a system only to provide fault tolerance are known as *ancillary* (or *ancilla*) qubits. Quantum computer vendors are already struggling to maximize the number of qubits, so having to "waste" some to provide error checking isn't optimal. Right now, the current state of many quantum computers and devices is that it requires many, many ancillary qubits to ensure one stable qubit. The number of ancillary qubits needed to create one stable qubit has been measured from a few up to millions. That's an extraordinary range and scaling problem. Still, sometimes the best performance that can be achieved involves adding more qubits and using them as ancilla. But all vendors look forward to the day they can minimize ancillary qubits.

NOTE Repetitive calculations and ancilla qubit error-checking methods don't do anything to catch and correct the error until the very end, which is an inefficient process. Also, since quantum answers have a probabilistic nature, they might yield different answers each time you run them regardless of actual error rates.

Increase Performance of Other Components

Another practical way to defeat decoherence errors is to increase gate, connectivity, state preparation, and reading performance. By decreasing the time it takes to compute a quantum result and

read it, the desired computation and resulting value read can be performed before an error makes the qubit prematurely decohere. For an example, let's suppose that a quantum computer has a ton of decoherence errors that start happening around 100ms. If computations can be completed in under 100ms, then that quantum computer can avoid the worst effects of decoherence and get more accurate results. This website lists and compares the relative performance values of different quantum computers: `https://quantumcomputingreport.com/scorecards/qubit-quality/`.

CLASSICAL COMPUTING ADVANTAGES

Based on the challenges of quantum mechanics and quantum computers, traditional binary computers aren't going anywhere soon. We understand how binary computers work at a fundamental level (even the parts that only work because of quantum mechanics). Classical computers do not have to worry about decoherence, the observer effect, or the no-cloning theorem, and they are not nearly as sensitive to external influences as quantum devices. Heck, they actually work quite well in the real world. They do not require ultra-super environmental conditions to run. They are fairly inexpensive. Anyone can routinely get a new laptop for less than $300, a portable device for $100, and fully functioning mini-computers the size of a matchbox for $25.

We have been able to fit billions and billions of integrated circuits on very small pieces of doped silicon. We fit dozens of CPU cores into a single chip. The data and outcomes they provide are stable for long periods in memory and on the processors without us having to worry about how the external world will impact them. There are some types of raw, rote calculations that binary computers do very well. Quantum supremacy may bypass traditional computers one day soon, but it's hard to beat the bang-for-the-buck, workhorse-like, stable binary computers.

For this reason, we are likely to have classical computers in our lives for a long time to come. More on this in Chapter 5, "What Will a Post-Quantum World Look Like?"

The big takeaway from this section of the book is to understand that the problem-solving power of a particular quantum computer is not just a function of the number of its qubits. Today, many of the qubits are likely to be involved in error correction. The overall performance depends on a variety of other factors, such as gate preparation, connectivity between gates, error correction, and read performance. Even when number of qubits and error correction rates are identical between quantum computers, they may be different types of quantum computers made to solve different types of

problems, so be careful about any direct comparisons based on qubits alone. Don't get tricked into believing that a 2000-qubit computer has to be better than a 100-qubit computer, especially for all types of problems.

Types of Quantum Computers

There are dozens of different types of quantum computers, theoretical models, architectures, and implementations. There are so many that the nascent field of quantum computing can't even agree on what comprises the "main" models before another pops up. This is not necessarily a bad thing. It demonstrates that the field is competitively trying to find the best solutions and is open to any possible new one that solves the biggest challenges in any way it can.

As the quantum computing world matures and solves its challenges, you can expect the weaker candidates to drop out and a few of the stronger solutions (or only a single superior solution) to emerge. But for now, we have lots of types and a ton of competition. There is no "best" quantum computer at the moment, although many vendors will tell you the one they are working on is the best one.

NOTE When the term *quantum computer manufacturer* or *vendor* is used in this book, it's important to note that most quantum computer development projects are worked on by the actual vendor of the computer with a ton of external help. Most vendors work closely with one or more universities, commercial and private labs, companies, military divisions—and possibly in multiple countries. Oftentimes other vendors supply such critical components as the quantum chips, refrigeration systems, and other building blocks of the quantum computer. "Customers" are often the same organizations that help build the quantum computer being tested, and they provide critical feedback and suggestions for the vendor, if not some of the parts. In this promising period of quantum computing development, most projects are an "all-hands-on-deck" deal, with everyone striving to get to quantum supremacy and stability as soon as possible.

Let's look at some examples of quantum computers, some of the vendors that make them, and some of the advantages and disadvantages of each type.

Superconducting Quantum Computers

Quantum computers relying on superconducting architectures were among the earliest prototypes and are still among the most popular. Dozens of vendors, including Google, Microsoft, IBM, D-Wave Systems, Rigetti Computing, and Intel, have one or more superconducting quantum computers that rely on the peculiar properties of superconductors to create and manage qubits.

In superconducting quantum computers, two weakly coupled superconductors are placed end to end, separated by a very thin insulator. A paired or bound (not entangled) set of electrons or fermions

(called a *Cooper pair*) are transmitted between the ends of two superconductors (a location known as a *Josephson junction*), through the insulator, and to the other superconductor using quantum tunneling. Figure 2.2 is a representation of a Josephson junction being used to transmit two Cooper-paired electrons between two superconductors. Each Cooper pair is supercooled, which creates a *condensate wave function* as it is transmitted to the other superconductor. This forces the particles into their lowest quantum energies and allows their properties to be observed at the macroscopic level. Phase changes and other quantum property changes can be observed to create and use qubits. Many, if not most, of the quantum computers created today use some form of superconducting quantum circuits.

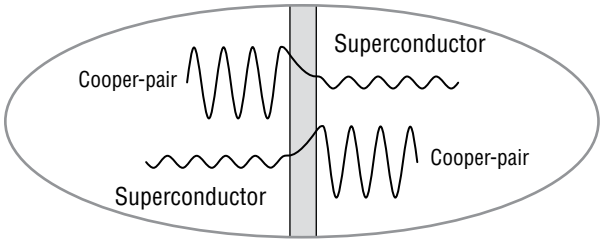

Figure 2.2: Representation of a Josephson junction with two Cooper-paired quantum particles being transmitted between two semiconductors

For more information on superconducting quantum computers, see the following sites:

`https://en.wikipedia.org/wiki/Superconducting_quantum_computing`

`https://web.physics.ucsb.edu/~martinisgroup/classnotes/finland/LesHouchesJunctionPhysics.pdf`

`www.nature.com/articles/s41534-016-0004-0`

`www.ncbi.nlm.nih.gov/pmc/articles/PMC3417795/`

`https://qudev.phys.ethz.ch/content/courses/ASC04_SCqubits_Review.pdf`

Quantum Annealing Computers

Annealing, in general, describes heating something to get to another desired state, such as heating glass to allow it to be molded into another shape or superheating a metal and allowing it to slowly cool to improve its strength or purity. Quantum annealing computers start with their qubits in a superposition of states, with each state having an equal probability of the eventual outcome. Then the computer applies a thermal-assisted (classical) and/or quantum tunneling annealing process to each qubit using an apparatus called an *electromagnetic coupler*. The coupler changes the states from an equal probability to an unequal probability (thus increasing the likelihood of particular states). Then the quantum states will try to minimize their energy to the lowest possible energy state (something that happens in the classical world as well). The lowest energy states have the highest probability of being the final answer. A good explanation of this process is explained in these videos:

www.youtube.com/watch?v=UV_RlCAc5Zs
www.youtube.com/watch?v=kq9VqROZGNc
www.youtube.com/watch?v=Yy93LMGQbpo

NOTE Annealing computers are closely related to *adiabatic* quantum computers.

If that technical explanation didn't make sense, a good way to think of the annealing process is having a ball (state) on one side of an undulating, uneven, sine wave–like hill (see Figure 2.3). The undulating shape varies based on the particular (math) problem involved. The ball, without any external influence, wants to stay still in its currently held, lowest area of surrounding ground. Without any external help, it's not going to be able to make it over any hill(s) to any other state, even if the other areas are lower than where the ball is now (i.e., the highest-probability answer). The annealing process helps the ball reach the lower areas. If the external help is applied thermally, the ball is given additional energy, which allows it to go over the initial and other successive hills until it reaches the lowest overall area (i.e., lowest energy and the final state). If quantum tunneling is used, the ball simply goes through each hill, like a train going through a tunnel, until it finds the overall lowest state.

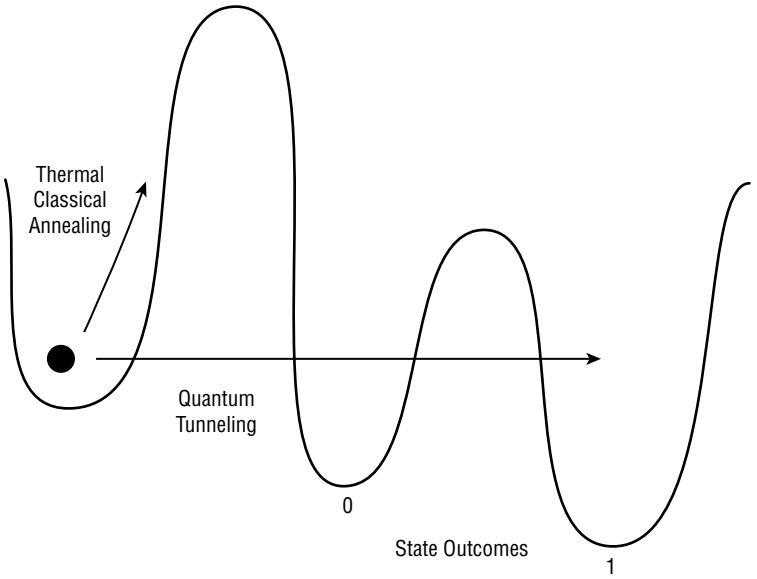

Figure 2.3: Graphical representation of the quantum annealing process

NOTE In nature and the larger world, the "correct answer" is often found by something seeking its lowest, less chaotic energy state. It's the way that mountains topple into sand and how water always wants to be at sea level. Many types of quantum computers also work by helping quantum particles get to their lowest natural energy states.

D-Wave Quantum Annealing Computers

An early and innovative adopter of annealing quantum computers is the company D-Wave Systems (`www.dwavesys.com/home`). The folks at D-Wave have created quantum computers with the highest number of claimed qubits of any quantum computing vendor. As of 2019, they have a 2048-qubit computer and will likely have more actual customers and applications than any competitor. D-Wave quantum processors are cooled to 0.015 Kelvin (0 Kelvin is nearly –460F), contain special shielding to block out all possible external electromagnetic interference, and have a relatively small footprint.

Advantages and Disadvantages of Quantum Annealing Computers

The advantages to quantum annealing computers are that they are fairly resistant to outside environmental noise, are easy to scale, and don't require near-absolute 0 Kelvin temperatures (even if they still perform better with lower temperatures). D-Wave proved that these types of quantum computers can be made—and made at scale.

Unfortunately, the disadvantages are probably the biggest of any quantum computer type. First, annealing computers can solve only one type of specific quantum problem known as *optimization problems*. This issue is reflected from the way they work, primarily relying on the lowest energy level to represent the optimal solution. For example, quantum annealing computers cannot factor equations involving large primes by using Shor's algorithm (discussed in more detail in Chapter 3).

Second, a large body of quantum physicists will not even consider annealing computers as truly, sufficiently quantum. They also question whether annealing computers can outperform classic computers in the long run or be useful enough to solve a wide range of nonclassical problems. There are a lot of back-and-forth arguments over these topics, specifically as they apply to D-Wave (especially since it was among the earliest and most prominent quantum computer manufacturers). However, evidence appears to be growing from various research papers in support of D-Wave's conjecture that its quantum computers use quantum tunnel annealing and that they can solve a larger range of problems than first theorized.

Universal Quantum Computers

As compared to the restricted use cases of quantum annealing, a universal quantum computer is the theoretical holy grail of quantum computers at the other end of the application-use spectrum. Universal quantum computers are not a specific type of quantum computer but generally describe a quantum computer that is not restricted to a group of limited-use cases.

Universal quantum is more of an outcome than a particular type of quantum computer or architecture as long as it can process any quantum algorithm. Some critics think it's more of a marketing term than anything else, but I disagree. The goal is to create a quantum computer that can accomplish any type of problem you throw at it, be it classical, quantum, or simulation. A universal quantum computer can not only accomplish the widest variety of scenario problems but can also simulate all the other types of more limited quantum computers.

The folks at IBM are working hard to make the first and best universal quantum computer, which is labeled IBM Q (www.research.ibm.com/ibm-q/). They have steadily been increasing the number of qubits IBM Q computers have, and since 2017 they have been predicting—and successfully meeting—a doubling of quantum volume each year. They are not only increasing the number of qubits, now at 50 qubits (see Figure 2.4), but also improving their stability and decreasing errors.

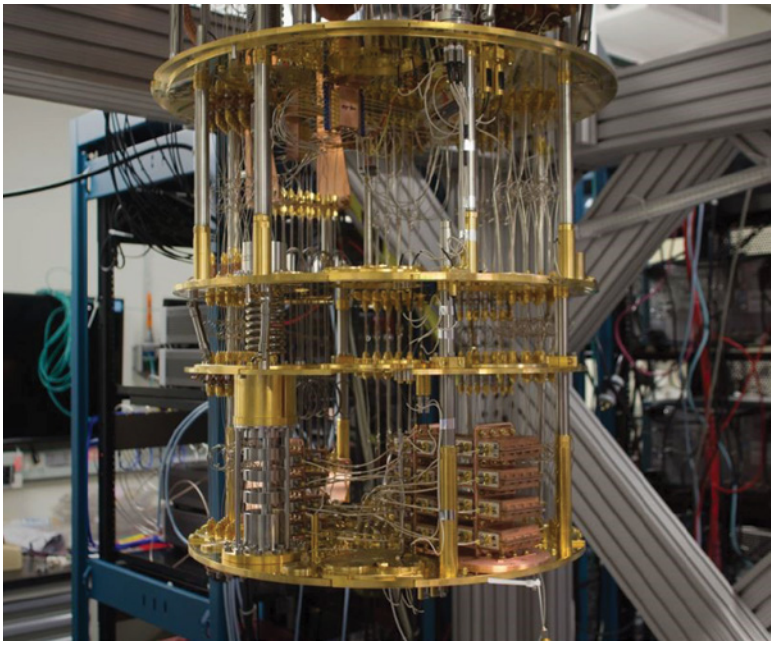

Figure 2.4: IBM's 50-qubit universal quantum computer
Image courtesy of IBM

Engineers at Google (https://ai.google/research/teams/applied-science/quantum-ai/) are also working on a universal quantum computer, currently at 72 qubits. Figure 2.5 shows Google's Bristlecone Quantum Processor, the backbone of its 72-qubit computer. Google has been predicting that it will reach quantum supremacy in 2019, although it made the same prediction in 2018. What that tells us is that Google engineers believe they are close and that they probably think the quantum supremacy threshold is around 100 qubits with today's error correction technology.

Advantages and Disadvantages of Universal Quantum Computers

Universal quantum computer manufacturers can use any technology they want to bring quantum processing to the largest and widest number of processing scenarios. The disadvantage is that these computers require the largest number of qubits and are among the hardest types to make.

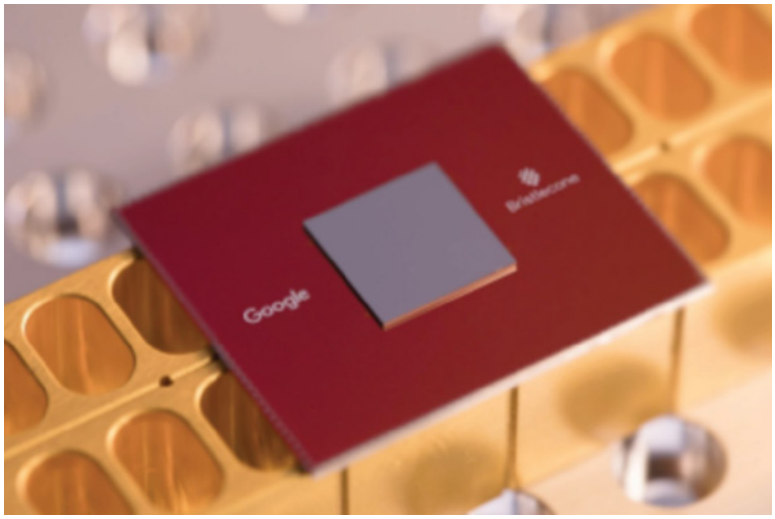

Figure 2.5: Google's Bristlecone quantum processor, the backbone of its 72-qubit computer

Topological Quantum Computers

Topology is a mathematical term used to describe transitioning properties of an object from one state to another without tearing or ruining the object. It's a study of how things are connected but it's not necessarily concerned with the distances. A common example you will see in quantum topological demonstrations is a donut-shaped circle morphing into a coffee mug, and vice versa. They look completely different except for the fact that the hole of the coffee handle mug came from the donut hole—and they can continuously be deformed into each other while preserving the same mathematical connectedness.

Quantum topological qubits are made up of relatively newly discovered, two-dimensional "quasiparticles" called *non-abelian anyons*. Anyons can be induced to create three-dimensional (time plus two spatial dimensions) quantum "braids." Individual anyons cannot be used to create qubits. It takes a collection of multiple anyons to create a qubit. Then, the collections of anyons can be moved around one another to create different types of quantum operations, and they can be moved together to create new particles. These movements and the new particles create wrapped braid chains.

NOTE If this type of quantum computer and the discussion of anyons seems difficult, you're not alone. Anyons were theorized for a long time before anyone could make them, and creating one just happened fairly recently. Even then, quantum physicists say making them takes "exotic phases of matter," which is not a term they throw around lightly. The 2016 Nobel Prize in Physics was given to three physicists who discovered quantum topology (`www.nobelprize.org/prizes/physics/2016/summary/`).

These wrapped anyon braids can be used to create strong quantum logic gates at the hardware level that fight decoherence and other quantum errors better than most of the other models. Whereas decoherence happens to most other quantum qubit types within tens of milliseconds or faster, anyon braids can last seconds. Only one type of quantum computer, ion trap, covered later, is thought to be able to have longer, uncorrected coherence times.

Topological quantum computers have one unique property that is especially interesting to physicists. Topological braids are often compared to strings with knots in them. You can move and change the string, but the "knots" are quantum information and remain no matter how you manipulate the string or what external influences bother it. Because the braid retains past quantum states (i.e., the history of the quantum information), observers can see from where the anyon braid state(s) started and how they changed over time. No other type of quantum computer has this property.

For a better understanding of anyons and quantum topological computers, check out the following videos:

```
www.youtube.com/watch?v=igPXzKjqrNg
www.youtube.com/watch?v=RW44rIrAZHY
www.youtube.com/watch?v=qj-w6ISQL5Y
www.youtube.com/watch?v=Xyfsr-coriQ
```

Microsoft Majorana Fermion Computers

Microsoft, Bell Labs, and several universities are heavily involved with quantum topology. In 2018, Microsoft created the first, very simple, 1-qubit, quantum topological computer, using Majorana fermions, which is related to, but not identical to, the anyons method. *Majorana fermions* are created by splitting electrons (which are elementary particles) into two smaller, entangled quasiparticles, which essentially form topological qubits that behave similar to anyons. Majorana fermions can act as their own *antiparticle*, meaning that if they meet each other they can annihilate each other. Every particle has an anti-particle (i.e., neutrons and electrons, for instance), but usually a particle type isn't also its own anti-particle. You can find an article on Majorana fermions at `www.sciencedaily.com/releases/2019/04/190401115906.htm`.

NOTE Splitting elections into smaller quasiparticles is known as *electron fractionalization*. The resulting, entangled quasiparticles each have half the charge of the original electron. Here's a great article on electron fractionalization: `https://phys.org/news/2015-05-electron.html`

Microsoft somewhat shocked the world when it delivered the first topological quantum computer, and even though it's only 1 qubit and independent developers have had no access to it (as of this writing), many quantum computing experts feel that, as this technology is scaled to far more qubits, it is likely to be a strong competitor for the quantum computers of the future. There are currently seven Microsoft quantum computing laboratories around the world working on quantum computers, and they have a full quantum stack (discussed in a moment).

Advantages and Disadvantages of Quantum Topological Computers

Topological quantum computers are relatively new and have a low number of demonstrated qubits (as compared to the other quantum computer types). But if Microsoft is successful at building more topological qubits to scale, the potential benefits are tremendous. The biggest advantages are that the qubits are more stable from the hardware level on up and the braids keep their quantum history.

Topological computers still need error correction and qubit control, but errors can be controlled by decreasing the temperatures and increasing the distance between topological particles. Because of this, fewer overall qubits are needed, which may also lower costs.

Ion Trap Quantum Computers

Ions are atomic particles with a net overall charge. Every *stable atom* has an equal number of protons (positively charged) and electrons (negatively charged), so there is no net overall charge in either direction. An ion is an atom with an unbalanced number of electrons and protons; thus it has a net positive or negative electrical charge. To create the ions, most of the ion trap computers superheat selected atoms (say calcium or ytterbium) to very high temperatures using a laser within a sealed vacuum. They then fire electrons at the superheated atoms, which causes the atoms to lose an electron and obtain a net positive charge.

Ion trap quantum computers use electromagnetic fields and a vacuum system to suspend and confine (i.e., trap) ions in free space above a room-temperature silicon chip. Lasers are then used to control the motion of the ions, including whether to entangle qubit pairs. Quantum information can be transferred through the collective motion of the ions or entangled pairs.

For more information on ion trap quantum computers, see the following resources:

```
www2.physics.ox.ac.uk/research/ion-trap-quantum-computing-group/intro-to-ion-trap-qc
www.youtube.com/watch?v=9aOLwjUZLm0
https://arxiv.org/pdf/quant-ph/9708050.pdf
www.youtube.com/watch?v=W0Q_jWe62EA
```

IonQ Ion Trap Quantum Computers

IonQ (`https://ionq.co/`), working with Sandia National Laboratories and other quantum users and vendors, is one of the leading proponents of ion trap technology. IonQ has been a leader in using room-temperature silicon chips in quantum computers, although some of its research has shown that it may need to cool the ion traps to 4 Kelvin (still well above the sub-1K temperatures most of the other quantum computer types require) as it increases the number of qubits beyond 32 qubits. Figure 2.6 shows an IonQ Ion Trap quantum processor unit with an artificially representative exploded cutout of the "trapped ions" above the QPU. The trapped ions are really inside that tiny slit in the center of the already tiny QPU.

Advantages and Disadvantages of Ion Trap Quantum Computers

There are a lot of advantages of trapped-ion systems, including that they can work at room temperature using somewhat traditional-looking silicon chips. They can have extremely long coherence

times, measured at 10 minutes and longer; they have high-fidelity entanglement; they can have all qubits coupled to one another (which is impossible for the other types of quantum computers); and they have very precise measurements as compared to other quantum technologies.

Disadvantages include increasing the number of simultaneously trapped ions while maintaining individual control and being able to measure them individually with high fidelity. Imagine someone holding a wooden 2×4 board with a straight line of marbles resting on the board. Early on, with fewer marbles, it's easier to keep them all in line on the board. But as you increase the number of marbles, it gets harder and harder to keep all the marbles from rolling off. Now imagine that someone is also individually moving the marbles back and forth and spinning them around. That's the central problem of trapped-ion quantum computing. Trapped-ion computers also have longer execution times, making them slower in comparison to other types.

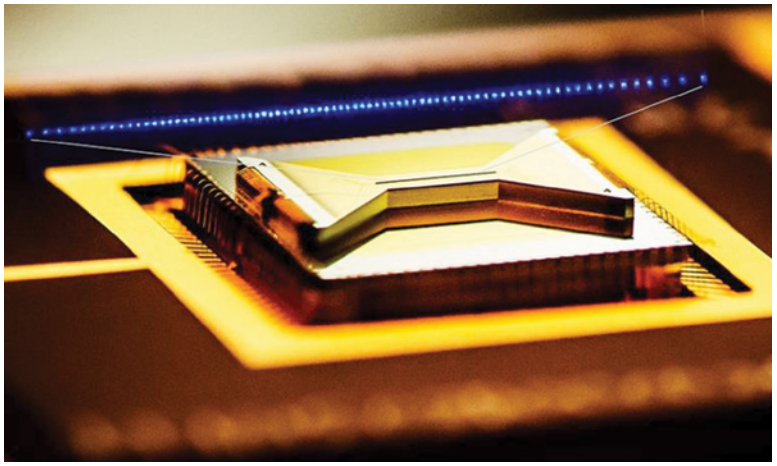

Figure 2.6: Example IonQ ion trap silicon chip with individual linear ions "exploded" out to reveal close up detail of individual ions
Image courtesy of Kai Hudek and Emily Edwards of IonQ, Inc.

With that said, there are research projects with over 100 trapped ions controlled with varying levels of success. But so far, the number of stable qubits has not approached the number of some of the other quantum computer types. But like the promise of Majorana fermion quantum computers, if IonQ and other ion trap quantum computer vendors can figure out how to scale up the trapped ions, this type of quantum computer could win out in the long run.

This ends our summary of some of the most important quantum computer architectures, models, and vendors. There are dozens of other real and theoretical quantum computer types, descriptions, architectures, and implementations that will not be covered in this book, including photonic, silicon quantum dots, diamond vacancy, one-way, quantum gate array, noisy-intermediate-scale-quantum, Turing, analog, and turning. I don't mean to slight any quantum computer architecture or indicate

that one is more important than another. They are not covered here simply to save space, as this chapter is already overly long. I covered enough different types to give you a flavor of the competitive world that is quantum computing and what some of the remaining challenges are. If you are interested in learning about other quantum computer types and architectures, a good place to start is `https://en.wikipedia.org/wiki/Quantum_computing#Quantum_computing_models`.

Quantum Computers in the Cloud

Currently, quantum computers are ultra-expensive, fairly large contraptions, requiring expert personnel to run and maintain, and they often require large refrigeration systems and other support services that the average company or person does not have access to. Still, this doesn't mean you or the average person can't quantum compute.

Many quantum computing vendors have long been offering access to their quantum computers or quantum simulators. Some vendors allow anyone to join and use the cloud computer for free for just about any legal reason. Others quantum cloud vendors require that users fill out detailed project forms, stating why the requestor is worthy of quantum computing time, and will be individually approved. Other times, the cloud service is 100 percent commercial or has private memberships. If you have a legitimate reason to play around with a quantum computer, you can probably find a quantum-based cloud service to help you, among them

```
www.research.ibm.com/ibm-q/
https://cloud.dwavesys.com
www.rigetti.com/qcs
www.huaweicloud.com/en-us
https://us.alibabacloud.com/
```

Many quantum computer observers think that quantum cloud computing is the model of the future, at least for the mid-term. Companies and individuals who can't afford their own quantum computers can take advantage of quantum cloud computing timesharing and pay only for what they need.

Non-U.S. Quantum Computers

Although previous sections of this book focused on U.S.-based and close ally quantum computer developers, many nations—including Australia, Austria, Belgium, Canada, China, Denmark, Finland, France, Germany, Italy, Japan, the Middle East (e.g., Saudi Arabia, Qatar, and the United Arab Emirates), the Netherlands, Poland, Russia, Singapore, South Korea, Spain, Sweden, Switzerland, and the United Kingdom—are pursuing quantum information science with varying levels of funding and participation. Out of this list of countries, most observers think the United States and China are the two biggest quantum competitors. Each is spending tens of billions on quantum computing.

It isn't necessarily always a country-versus-country competition. Many companies based in one country are involved in one or more projects in other countries. For example, the independent

quantum application development firm of Cambridge Quantum Computing (https://cambridgequantum.com/), while officially headquartered in the United Kingdom, is involved in projects, commercial, government, and otherwise, in many countries around the world. And the U.S. national project to select a quantum-resistant cipher (covered in detail in Chapter 6) contains many non-U.S. and multinational teams.

This not to say that there definitely isn't a "race-to-the-moon" type of competition between countries going on. There is. This makes sense from just a competitive standpoint where the early countries will be able to start reaping the benefits sooner, and also because as it relates particularly to this book, quantum computing will both break many national secrets and protect new ones.

Components of a Quantum Computer

Regardless of what type or architecture a quantum computer uses, today they all have fairly similar involved components such as these:

- Support staff
- Environmentally controlled, secured, super-clean computer room
- Lots of supplied electrical power
- Cooling system
- Gas storage and delivery systems
- Wiring
- Piping
- External traditional, classical computers to monitor, control, and manipulate the quantum computer
- Supporting circuity
- Electromagnetic shielding
- Qubit quantum processor unit (QPU) physical packaging
- Quantum data plane (includes QPU and all other quantum components)
- Control and measurement area
- Hardware connections, remote connections, and interfaces
- Classical computer components to store outcomes and resulting data
- Networking
- External cabinetry (to give it a tidy look)
- Operating system (startup code, control, monitoring, compiler, etc.)
- Software interfaces
- Algorithms
- Application software

Today, in a workable quantum computer all the components directly attached to the quantum computer—the parts most laypeople would think of as the "quantum computer"—take up at least

one or more square yards of space. This can be compared to binary computers, which can be as small as a matchbox or even a single chip. As with most computer system components, over time each quantum component and the entire system of components is likely to get smaller and smaller and less resource intensive.

NOTE There is a question of whether we'll ever get quantum computers down to a very small size (like a desktop computer or laptop) and working in hot, noisy, heavily externally influenced environments like classical computers do today. Many quantum experts think that it can be done. Why? For one, because our brains work on quantum mechanics, and it's as hot, wet, and abused by the external environment as it comes. Nature figured out a way to do it. One day humans may be able to figure it out. Humans appear especially able to shrink things once we figure out to build them.

Two of the components, the application software and what is called the "stack," deserve additional coverage.

Quantum Software

It takes more than hardware and qubits to make a quantum computer able to solve difficult problems. Every quantum device comes with one or more operating systems, algorithms, interfaces, and application programs. At the very least, the quantum computer must have the prerequisite firmware or *control* software that allows the quantum computer to create, initialize, measure, control, error check, and decommission qubits.

Each quantum device must implement one or more quantum algorithms (some of which are covered in the next chapter), which handle the base layer computations and math based on qubit manipulation and natural laws. Not all quantum devices support all algorithms, although the universal gate quantum computers are supposed to.

Most quantum computers have compilers, programming languages, and script languages to allow developers to write their own quantum computer programs. Many quantum vendors provide their own private software to their customers, either for free (most commonly) or for a commercial fee. Some quantum device vendors have created or encourage their customers to work with open-source quantum software. Others offer their proprietary quantum software for free as a way to induce developers to learn and develop on their quantum computers. It's a model that has worked quite well in the classical computing world.

Quantum Stack

Many vendors provide a "network" of resources, including tutorials, programming tools, simulation tools, and access to their cloud resources. Many quantum computing vendors will talk about having "the full quantum stack." A *quantum stack* is the quantum-based collection of hardware, qubits,

quantum software development kit, APIs, and applications. Many vendors, including IBM, Google, Microsoft, D-Wave, IonQ, and many others, offer all or parts of the stack. Here are some websites for quantum software and stacks:

```
www.quantiki.org/wiki/list-qc-simulators
https://github.com/qosf/os_quantum_software
https://arxiv.org/abs/1812.09167
http://quantumalgorithmzoo.org/
https://qosf.org/
https://algassert.com/quirk
https://github.com/rigetti/pyquil
https://cambridgequantum.com/
https://marketplace.visualstudio.com/items?itemName=quantum.DevKit
https://quantumexperience.ng.bluemix.net/qx/editor
```

Quantum National Guidance

Most major countries have national agendas and funding involved in helping their country's government entities, industries, and companies to push ahead in quantum computing supremacy.

National Policy Guidance

For example, in the United States, the National Institute of Standards and Technology (NIST) has a national consortium dedicated to quantum computing topics (`www.nist.gov/news-events/news/2018/09/nist-launches-consortium-support-development-quantum-industry`), and the White House released a federal document titled the *National Strategic Overview for Quantum Information Science* (`www.whitehouse.gov/wp-content/uploads/2018/09/National-Strategic-Overview-for-Quantum-Information-Science.pdf`). NIST is also sponsoring a contest to determine the official post-quantum cipher (`https://csrc.nist.gov/Projects/Post-Quantum-Cryptography`) (covered in detail in Chapter 6). Many of the ciphers submitted to the NIST contest are likewise funded by other countries and their governments.

Money Grants and Investments

Most nations are putting their money where their mouths are by providing rich grants of funding. The top-tier countries, like the United States and China, are spending billions of dollars each year. Even the smallest involved countries are spending many millions to tens of millions of dollars. The grant funding goes to improve quantum research and computers at their universities, research labs, vendors, and government and military initiatives.

Here are some examples:

www.aip.org/fyi/2019/national-quantum-initiative-signed-law

www.nextbigfuture.com/2018/10/us-mobilizing-funding-for-quantum-ai-to-match-china-in-multi-billion-race.html

www.scmp.com/news/china/economy/article/2140860/china-winning-race-us-develop-quantum-computers

https://quantumcomputingreport.com/news/

www.executivegov.com/2019/04/energy-department-announces-quantum-computing-funding-opportunity/

Interested investors can commit some of their capital to quantum-related firms, using a quantum-dedicated exchange-traded fund (ETF) like this one: https://www.defianceetfs.com/qtum. Private investors are pouring hundreds of millions of dollars directly into private companies, including those noted here: www.nanalyze.com/2018/09/10-quantum-computing-startups/. In general, a lot of private investment is chasing quantum computing–related investment opportunities. As great as the promise of quantum information sciences is, as with anything else, investors beware. Many quantum industry observers think that all it takes to garner a huge inflow of investment capital is the word *quantum* in a company's name or prospectus, just like during the similar recent investment craze in the past if you included the word *bitcoin* or *cryptocurrency* in your company name. Many people lost fortunes in early bitcoin investments. There will be winners and losers in the quantum space as well.

Other Quantum Information Science Besides Computers

The field of quantum information sciences includes more than just quantum computers. It includes many devices and software components, such as

- Quantum random number generators
- Quantum networking
- Quantum cryptography
- Quantum applications

Many firms concentrate on these types of quantum devices instead of trying to compete in the hugely expensive field of quantum computers. Most of these items will be discussed in more detail in future chapters.

Quantum information sciences is promising to open up a range of applications that will significantly change our world. Here are some of the promised applications:

- Faster computation
- Faster optimized searches

- Better artificial intelligence
- Better cryptography
- More secure networking
- Military uses
- Improved weather forecasting
- Improved medicines and chemicals
- Improved understanding of the quantum world, astrophysics, and our universe
- Perfect privacy (using something known as *fully homomorphic cryptosystems*)
- Better financial modeling (stock trading, derivative trading, etc.)
- Better fraud detection
- Traffic management for autonomous vehicles
- Better, longer-lasting, lighter batteries
- Quantum money

We will cover most of these topics in more detail in Chapter 5.

For More Information

There are several good Internet articles and books on the history and state of quantum information science. If you are interested in more information, consider these sites:

```
https://mitpress.mit.edu/books/quantum-computing-everyone
www.irtf.org/mailman/listinfo/qirg
www.nist.gov/history-and-future-quantum-information
www.wired.com/story/wired-guide-to-quantum-computing/
https://en.wikipedia.org/wiki/Timeline_of_quantum_computing
https://en.wikipedia.org/wiki/Quantum_computing
https://towardsdatascience.com/the-need-promise-and-reality-of-quantum-computing-4264ce15c6c0
```

Summary

Chapter 2 covered quantum computers, their types and architectures, components, and other quantum information sciences. We discussed the main differences between quantum and traditional binary computers, and explained the technology, advantages, and disadvantages of each quantum computing architecture approach. We also covered the various major vendors and their quantum information science concentrations. Chapter 3 will discuss how quantum computers will likely break most forms of traditional public-key cryptography within a few years.

3 How Can Quantum Computing Break Today's Cryptography?

This chapter covers how quantum computing is likely capable of breaking most forms of traditional public key encryption. We begin by discussing cryptography basics, paying particular attention to how most of today's public key encryption schemes provide protection. Then you will build on your foundation from Chapter 2 and learn how quantum computers can break that protection and what cryptography is or isn't overly susceptible to quantum cracking.

Cryptography Basics

Cryptography is the science, study, and practice of securing and authenticating people, data, transactions, and other objects between authorized parties. It is done by using encryption, integrity checks, and algorithmic implementations. Cryptography allows confidentiality and integrity of data, communications, and participants to be maintained whenever desired between authorized, designated parties (or software or devices on their behalf). This section will cover digital encryption, authentication, and integrity hashing basics.

NOTE The term *subject* will be used throughout this chapter. It is used to refer to any identity that can be tied to a cryptographic action. A subject can be a user, group, computer, device, service, daemon, company, publisher, or any other identity object.

Encryption

Encryption is a popular method for subjects to keep something secret. A single subject may want to keep something secret to itself, or the secret may be shared between a selected group of people or devices. The secret can be any type of content, the identities of participating parties, and any involved transactions and objects.

Encryption in various forms has been used for thousands of years, beginning with spoken codes and encoded writing. A common example is that of simple *substitution ciphers*, where letters and

Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto, First Edition. Roger A. Grimes.

numbers of an alphabet are rearranged to create a coded message that only the intended parties understand. In its simplest application, the parties using the substitution cipher could have agreed to move every character of the unencrypted message forward one letter in the alphabet to encode and encrypt the message. Thus, the word FROG would become GSPH (F + 1 forward letter position = G, R + 1 = S, O + 1 = P, and G + 1 = H). All involved authorized receivers would need to be told that the resulting encoded message could be decoded by reversing the process (i.e., G – 1 = F, S – 1 = R, P – 1 = O, and H – 1 = G). See Figure 3.1 for a graphical example.

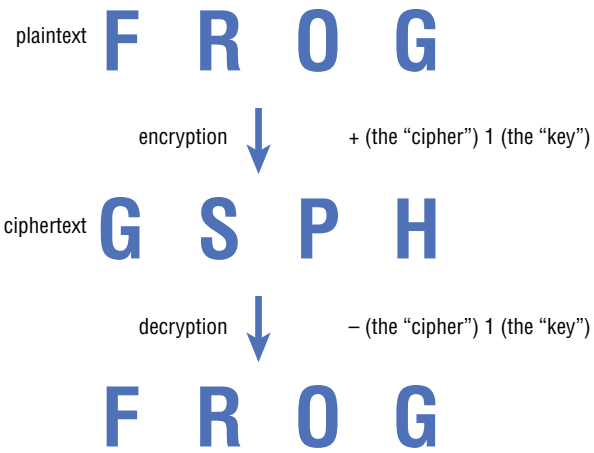

Figure 3.1: A simple substitution cipher

The original, unencoded message is known as the *plaintext message*. The encoded message is known as the *encrypted message*, or *ciphertext*. The process used to transform a plaintext text message to an encrypted form is called *encryption*. The process used to reverse an encrypted message back to its original, plaintext message form is called *decryption*. The documented process and steps used to encrypt or decrypt a message is known loosely as a *cipher*, or a *cipher algorithm*.

NOTE In the computer world, a message can be any type of digital content, including text, email, chat messages, data, sound, pictures, and videos.

Every cipher is essentially a more complex implementation of the basic encryption components covered earlier in the simple substitution example. There is always a plaintext message that is encoded and decoded using a cipher algorithm. In the simple substitution method earlier, the cipher algorithm is mathematically represented as $+X$ or $-X$ (i.e., + or $-X$, or $+-X$), where X is the number of positions in the alphabet to move forward or back for the encryption or decryption.

The + or – is the (simple) cipher algorithm. *X* is the *cipher key*. In today's world, the cipher algorithms are represented using fairly simple to complex math equations. Some cipher algorithms are made up of mathematical equations using traditional math operations such as addition, subtraction, multiplication, and division. Others use trigonometry, calculus, and other more advanced mathematical operations. In every case, the cipher algorithm allows a plaintext text message to be encrypted and decrypted in a predictable way if supplied with the relied-on algorithm and key(s).

Around the advent of wireless and radio communications, the analog transmission waves themselves were encoded to prevent unintended eavesdropping. When computers came into common use, digital encryption was used to protect sensitive digital communications. As covered in Chapter 2, "Introduction to Quantum Computers," traditional binary computers work with bits (the binary digits of 0s and 1s). All digital data and objects are stored as 0s and 1s on classical computers. When encryption is needed, those bits are rearranged in a predetermined, algorithmic way to provide the encryption obscurity.

NOTE Another term for a cipher is *cryptographic primitive*.

Encryption Keys

As previously explained in our simple substitution example, the cipher algorithm is represented as +–*X*, where *X* is the number of positions to move forward or back. The *X* is the cipher key, which is the number of bits used to encrypt the message using the cipher algorithm. A slightly more complex simple substitution example might be to use a key of 12. In this case, the word FROG becomes RDAS (F + 12 = R, R + 12 = D, O + 12 = A, and G + 12 = S). The cipher is the same, but the key changed. If the cipher is strong and the key long enough, it can be *nontrivial* (that's cryptospeak for very, very hard . . . or "you probably can't do it in your lifetime") for an unauthorized party without knowledge of the key to decrypt the protected message.

NOTE With good cryptography using strong and reliable ciphers, everyone can know the details of the cipher algorithm. The strength of the cryptography comes from the strength of the algorithmic conversion process and a long enough key. The key must be kept secret from unauthorized parties, but not the cipher. Cryptographic solutions requiring that the cipher also be kept secret are usually considered suspicious and likely weak by most observers.

All other things considered equal, the protection of a key increases as it becomes longer. As a key's length increases, it becomes harder for an unauthorized party to convert a protected encrypted message back to its plaintext state, even if they know the cipher algorithm. In our simple substitution examples, it's fairly easy even for a child to understand how to count and add or subtract one additional (+–1) alphabetic position to encode and decode a message, or even add or subtract twelve (+–12)

changed positions. But if our key suddenly became +–1,234,567,980 changed positions in the alphabet, the solution becomes harder for the average human to calculate (although still not impossible).

In the classical computer world, digital encryption keys are simply a long series of seemingly randomly generated 1s and 0s. A digital key looks something like 10101010101101000101010101010110 0111001010101. The digit key is applied to the plaintext message according to the cipher algorithm to produce an encrypted message. If done correctly, both the key and the encrypted message appear like a random set of unpredictable bits.

Today, digital encryption keys usually range in size from 128 to 4,096 bits, although they can be smaller and larger in less common scenarios. Whether a particular length of bits is considered secure depends on many factors, including the involved cipher algorithm, the speed at which all the possible key bit positions (called the *key space*) can be guessed, and any "tricks" that can be used to cut down the brute-force guessing methods. Strong cipher algorithms that are harder to "crack" can use smaller key bit sizes, whereas conversely weaker algorithms often require longer key sizes for equal periods of protection. Bit lengths for new keys using the same algorithms tend to grow over time to compensate for greater computational power and other cracking factors. Cryptographic attacks only get better over time, which weakens the protective power provided by the length of the key.

You can readily find and see cipher key examples by opening a digital certificate on any computer or device. Figure 3.2 shows the 2048-bit key extracted from a digital certificate.

```
30 82 01 0a 02 82 01 01 00 ad 0c 9f 7d 67 bc
70 6d 79 ba 25 05 3a 64 60 a0 e2 23 f3 ec 17
3b 6e 75 9e 88 50 fb d9 de 9c 62 2b de 19 a8
52 57 f0 09 62 2c 5e 64 45 9c 60 39 b5 14 48
2e 27 a4 db 82 c8 02 da ba 1d 91 51 fb 90 fa
bf f7 55 65 f1 cc 98 1a 3f 6b 0f 74 18 8f d4
cc 3b 44 ca 4d 53 df 95 94 72 20 d1 45 1a a5
9b 3b a8 f2 71 79 0e 6e ad 5b 87 ca 9e d1 7f
72 b8 2b 93 e0 36 69 31 7b 60 9a 44 f8 f4 a5
45 de 15 62 01 93 cd b3 ea e6 d1 d5 3c 1a 6b
cd ea a2 fd 7d 56 35 d0 c5 aa 5f 0e 6f 6e b2
c7 fa 8c 57 10 58 d3 0a 14 b4 2a fd 09 c6 ac
17 8e 3a ba 2c e8 dc 51 9f 29 a8 cb 39 e2 5a
8a 60 96 62 d7 64 05 94 d1 d7 8c 5b e3 0f fd
01 ed b4 5f 32 de b9 b1 b3 ea 3e 4c 6e d0 90
c4 82 eb 58 dc 6c 14 f0 4e 9f 1f 74 a3 76 26
30 bc 9a 97 91 fd 7c c8 c6 5a fd f8 54 ae 09
48 5a 50 b3 0c 3b 8f 43 f6 5f 02 03 01 00 01
```

Figure 3.2: 2048-bit cipher key from a digital certificate

NOTE Nearly all digital cryptographic keys are shown converted to their hexadecimal representations (e.g., base 16 numbering system) on most computers and devices instead of their underlying bits (i.e., 1s and 0s).

You can see the general recommended minimum key sizes for popular ciphers by visiting `www.keylength.com/`. Encryption ciphers are generally broken into two major types: symmetric and asymmetric.

Symmetric Ciphers

If a cipher algorithm uses the same key to both encrypt and decrypt a message, it is called a *symmetric cipher*. For example, in the earlier simple substitution examples, the keys used to encrypt the plaintext message (e.g., 1, 2, or 1,234,567,980) are the same keys used to encrypt or decrypt the encrypted message back to its original form.

All things considered equal, symmetric ciphers are stronger, faster, and easier to validate than asymmetric ciphers (covered later in this chapter), and they require smaller key sizes. From a cryptographic viewpoint, good symmetric ciphers are easier to prove as strong and reliable. They have less complicated math. They require less assumptions and guesswork. They are harder to attack. Accordingly, symmetric ciphers do most of the world's data encryption.

The world has used many different symmetric cipher standards since the 1970s, including Data Encryption Standard (DES), Triple DES (3DES), International Data Encryption Algorithm (IDEA), and Rivest Cipher 5 (RC5). All of these older symmetric ciphers are considered weak and broken today.

Since 2001, the most popular symmetric cipher is known as *Advanced Encryption Standard* (*AES*). Periodically, when needed, the National Institute of Standards and Technology (`www.nist.gov`) conducts public competitions to select new cryptographic cipher standards to replace aging and weakening cipher algorithms. For the AES competition, over a dozen different teams submitted their symmetric ciphers to NIST to be considered as the new national symmetric cipher standard. In a fairly open and deliberative process, NIST chose a cipher called *Rijndael* and renamed it Advanced Encryption Standard. Currently, AES uses key sizes of 128-, 192-, and 256-bit lengths, and its strength has held up very well under years of cryptographic scrutiny and attacks.

NOTE A cipher key that is only known to and used by a single subject and that is not intentionally shared with anyone else is known as a *private* or *secret key*. A cipher key that is intentionally shared between multiple subjects is known as a *shared key*. A key that can be known and used by anyone is known as a *public key*. A key created to be used only temporarily is known as a *session key*.

Symmetric Cipher Weaknesses This is not to say that symmetric ciphers don't have their weaknesses and shortcomings. They do. Common symmetric cipher weaknesses include a lack of authentication abilities and key exchange scaling problems.

Because the same key is used to encrypt and decrypt a message, any party with access to the key can both encrypt and decrypt messages, and possibly pretend to be any other involved party with the same keys. From a purely cryptographic standpoint, if someone accused one of the other participating parties of encrypting something, because everyone shares the same symmetric key, the accused

party cannot (again, from a cryptographic perspective) repudiate the accusation. This is known as *nonrepudiation*. It's not a desirable trait in the cryptographic world.

This also means that symmetric keys, by themselves, for the same reasons are more difficult to use in most authentication scenarios, especially where data integrity or subject authentication is desired. For example, suppose Fred, Wilma, and Dino share the same encryption key. Fred could encrypt some data and send it to Wilma but claim it came from or was originally created and encrypted by Dino. Because everyone shares the same key, from a purely cryptographic perspective Dino could not prove who really sent or encrypted it. Fred could even take a message originally encrypted and sent by Wilma, decrypt it, maliciously modify it, encrypt it again, and send it onto Dino claiming it was from Wilma. Dino would have no way to tell who the message came from and no way to tell if the original message was tampered with before being decrypted and opened by him.

A second big issue is that symmetric ciphers are not easy to use as the number of sharing participants grows. In a small group, say two or three people, it's relatively easy to securely exchange the shared symmetric key, although even then all participating parties need to make sure the shared key is accurately and securely communicated to all participants. It would be difficult for any two people to read, write, or say a 256-bit key accurately. Many of us have a hard time communicating a 16-digit credit card number to another person.

Say you had a shared symmetric key scenario that had a thousand participants. One or more participants would have to find a way to securely transmit the agreed-upon key to all the other authorized parties. How would it be done: writing, calling, emailing, etc.? If writing is used, how is the writing sent securely to the other parties? Can the mailing system and its workers be trusted? Could it be guaranteed that only the intended recipient opened a mailed message on the other end? If calling is used, how trusted is the telephone system? Is it possible that someone could be eavesdropping? Probably. What is the sender to do if the receiver is not available? Do they leave the key on voicemail? If the receiver does hear the key, can they accurately transcribe it? If email is used, how trustworthy is the email system and all the points of transit between the sender and the receiver? In any email system, there are one or more email administrators who can read everyone else's email. In any case, no matter how communicated, can you imagine a thousand different people trying to securely and accurately share a 256-bit symmetric key without making a single mistake? The difficulty of securely exchanging a shared, symmetric key increases exponentially in proportion to the number of involved participants.

Now suppose some of the thousand participants wanted additional, smaller groups, where they used different shared keys for each subgroup. They would be responsible for keeping track of which keys were used by which people and groups. Going further, suppose every participating user needed guaranteed encryption between each and every party that no other person or party could see. This would require that each of the thousand participants have a separate shared key for each possible union of other participants. Each user wishing to send a confidential message to all other users would need to send the message using 999 different symmetric keys and keep track of which keys belonged

to which unions. This would require 499,500 (1,000 participants × 999 other participants / 2) symmetric keys.

Clearly, this would be an onerous undertaking, especially if participants were periodically required to change keys to ensure continued, strong privacy against ongoing attacks. Philosophers and cryptographers for centuries sought a better way to securely exchange private information and/or symmetric keys (the latter of which is called *key exchange)*.

Asymmetric Ciphers

The holy grail in encryption was to find a method that allowed two or more parties to exchange symmetric keys across an untrusted (even knowingly malicious) communications channel without having to first establish ahead of time a private communication method to exchange the symmetric keys for each participant. In the mid-1970s, several different parties, unbeknownst to each other, developed nearly the same solution within a few years of one another.

Integer Factorization Workload Effort The solutions all used a polynomial math problem (e.g., A × B = C) that was so inherently difficult to solve back to its individual constituent parts (i.e., *factor*) that the workload effort needed to solve it was the protection. The math problem needed to be so hard to factor that if anyone learned C (the result of A × B), no one could easily figure A or B. That polynomial workload effort (also known as *integer factorization problem*) is the core protection behind most of today's *public key* cryptography.

NOTE Workload efforts similar to integer factorization but used in different yet related types of asymmetric ciphers include the *discrete logarithm problem* and the *elliptic-curve discrete logarithm problem*. They use different types of very hard-to-solve math, but they employ fundamentally different approaches.

Today, the most popular asymmetric cryptographic solutions use two large prime numbers (A and B), which when multiplied (or algorithmically applied) together gave a much larger result (e.g., C). As first covered in Chapter 2, a prime number is a whole number above 1, which can only be divided by itself and 1 to get a whole number (2, 3, 5, 7, 11, 13, 17, 19, 23, etc.). Any other combination results in a remainder or fraction. Prime numbers are inherently hard for traditional binary computers to create on demand and verify. If very large prime numbers are the values of A and B, even if someone knew C, they would have a very hard time factoring the result back to its underlying prime constituents (i.e., A and B).

To explain better, let's start with a simple example. Let's use a common cryptographic math equation that is representative of the integer factorization protection method: $p * q = n$, where p and q are prime numbers, and n is the resulting mathematical result and is the public key of the key pair (explained in a moment). If p and q are sufficiently large, p and q can be very hard to figure out if given only n.

For our simplest possible example, let's suppose that $n = 15$. What two prime numbers if multiplied together would give a result of 15? This is pretty easy to figure out, especially since the only possible prime numbers below 15 are 2, 3, 5, 7, 11, and 13. It wouldn't take anyone too long to figure out that p or q must be 3 or 5, because $3 \times 5 = 15$ and no other combination of prime numbers multiplied together equals 15.

Now, let's add just a bit of complexity to the problem. Suppose $n = 187$. What two primes when multiplied together will give you 187? Now, the mental effort it takes to factor 187 into two multiplied primes grows. The average human being can still figure this out, but doing so isn't nearly as easy. The answer is p or q is 17 or 11, since $17 \times 11 = 187$.

But supposing that $n = 84773093$, what are p and q? Now we are talking real mental effort. You would have to figure out all the primes below 84773093 and multiply them in different combinations to see which ones resulted in 84773093. Most humans could not do this quickly. It could be done—just not quickly without a computer. If you are interested, the answer is that p or $q = 9539$ or 8887. Computers can still do this one very quickly.

But now imagine that n equals a number represented by 4096 bits. This number is so large that most calculators cannot show it. They will error out or show an infinity symbol. A 4096-bit number is a number represented by 1234 decimal digits. It can be represented as 2^{4096}–1 possible numbers. To brute-force guess at a number that big is an impossible task, much less to actually figure out the two super huge primes multiplied together that would make up that number as a result.

When cryptographers try to explain how many brute-force guesses it would take to guess the right two prime numbers used to generate a 4,096-bit number, they start creating hilarious, absurd comparisons because it's the only way to possibly communicate to the average person how inherently difficult factoring super huge primes would be. All the possibilities for a 4096-bit number are more than all the atoms in the known universe. For perspective, there are more than 125 million atoms in the period at end of this sentence. Another comparison is to say that if you had a million of something, say pennies, for every star in the universe, and there are 100 billion stars in each of the 10 trillion galaxies of our universe, you still would have only enough pennies to represent 1 percent of the possible numbers a 4096-bit number could be, much less figure out the two larger prime numbers that were used to create it.

Some naïve observers, new to numbers this big, think that all we need is a lot more computing power—perhaps all the computing power on Earth would do it? They would be wrong. Not only are there not enough classical computers, processing power, memory, and storage space in the entire world now and forever, there aren't enough atoms of energy to power those items if they were to try—at least using traditional classical binary computers. So, the workload (and time) required to factor large prime number equations is what provides the protection.

NOTE All digital cipher protection is provided because of the hardness to brute-force guess the keys out of all the possible combinations or to factor some sort of math equation back to its original component parts. Various cipher creators have come up with a math problem that isn't easy to

figure out if you don't have some part of it. You may know that A + B = C, but even if you know A and C you can't easily figure out B. The difficulty of solving for the unknown value is what gives the cipher its protective capabilities.

Public-Private Key Pairs With this asymmetric cryptographic method of using large prime numbers, each participating party generates (or is given) a key pair, where the two keys of the pair are cryptographically related to each other. One key is kept private and shared with no one else (the private key). The other key can be distributed to the whole world (and is known as a *public key*). Whatever one key encrypts the other can decrypt, and vice versa. This is *hugely* important to the concept of asymmetric cryptography and everything it can do, so you must understand these two points if you want to understand asymmetric cryptography. Because one key is used to do the encryption and another key is used to decrypt, this type of cipher is known as *asymmetric*.

Even though both keys of the key pair can be used to encrypt a message to the other, and vice versa, the nature of who has the private and public key classifications is important. Remember, the private key is never shared with anyone else. Because of this, if someone wants to send a confidential message to another person, they must use that person's public key to encrypt the message. This will keep the message confidential until the receiver uses their related private key to decrypt it. Since no one else has the receiver's private key, no one else can decrypt the message.

NOTE With asymmetric encryption, we must use the public key of the receiver to encrypt messages to them.

With asymmetric encryption, each participating party needs their own private-public key pair but only one key pair per person to securely communicate with each other. Instead of needing 499,500 different symmetric keys to securely communicate with each other, an asymmetric system would need only 1,000 private-public key pairs (or 2,000 keys in total).

Digital Signing Asymmetric cryptography users can also use their key pairs to authenticate and digitally sign content. *Digital signing* is the act of providing proof that the signed content is still as it was at the moment of the signing. To sign content, a user uses their private key to "encrypt" the content (or a hash result, covered in a moment). Although we don't really call the process "encryption" since anyone who has the related public key (which theoretically could be the entire world) could decrypt and read it. It can't be considered confidential or encrypted if everyone in the world can see it.

Instead, we call it digital signing. Any content signed by the private key can be revealed only by using the related public key. If the content can be verified ("decrypted") by the related public key, it must have been signed by the related private key because the only thing the related public key can "decrypt" is something signed by the related private key. Similar processes can be used to authenticate user identities involved in cryptographic operations, some of which will be covered later in this chapter. Common digital signature ciphers include Digital Signature Algorithm (DSA) and Elliptic Curve DSA (ECDSA).

A message can be encrypted and signed if both protections are needed. If Fred needs to send a signed and encrypted message to Wilma, then he signs his message using his own private key and then encrypts it using Wilma's public key.

NOTE Digitally signing and verification is a bit more complex than indicated in the previous description. We will cover that later.

Because each party has their own, unique key pair, and only that key pair can encrypt and decrypt messages between each other, asymmetric cryptography also allows subject and message authentication. Each involved key pair can be tied to a particular subject. This allows *repudiation*.

Key Exchange Because symmetric keys are more secure at smaller key sizes, they are used to do most of the world's message encryption, and asymmetric ciphers are often used just to securely transmit shared symmetric keys between two parties. Asymmetric cryptography allows symmetric key exchange across untrusted networks without having to previously establish a secure, trusted channel. A very basic summary of the key exchange process looks similar to this:

1. The client and server connect to each other.
2. The server sends the client the server's public key of its asymmetric key pair.
3. The client uses the server's public key to encrypt the client's newly generated "*session*" symmetric key back to the server.
4. Both the server and the client now use the shared, session symmetric key to send encrypted content back and forth to each other.

In the real world, using asymmetric key exchange to securely transmit a shared symmetric key between the client and server has a few more steps and complexity (which will be covered later), but this is a good summary of the basic steps of asymmetric key exchange for now.

Common types of asymmetric cryptography include Rivest, Shamir, Adleman (RSA), Diffie–Hellman (DH), Elliptic Curve Cryptography (ECC), and ElGamal. RSA is easily the most popularly used asymmetric encryption cipher, perhaps accounting for 95 percent of all asymmetric cipher uses. Although all asymmetric ciphers are used to perform key exchange, Diffie–Hellman, also known as Diffie-Hellman-Merkle, is often associated with key exchange-only implementations, as is the lesser used Elliptic Curve Diffie Hellman (ECDH). RSA and DH key sizes typically range from 2048 to 4096 bits today, and they have doubled in length about every 7 to 10 years to fight off progressing cryptographic attacks.

NOTE RSA Security, the company behind the RSA cipher, used to offer an ongoing cash prize to cryptographers who broke increasingly bigger RSA key sizes. The biggest RSA key publicly broken to date using factorization is 768 bits (`https://eprint.iacr.org/2010/006.pdf`), accomplished in 2010. It involved a 232-digit number—big, but not anything close to as big as 2048-bit or 4096-bit

keys as is usually recommended today. Still, even before the 768-bit key break, RSA stopped offering the contest without explanation. Many observers think the coming implementation of quantum computers was a major factor (excuse the pun).

Key Trust and Public Key Infrastructure In order for asymmetric cipher systems to work, the people communicating with them must have trust that everyone's public key is valid and belongs to who they think it belongs to. In the early days of asymmetric cipher communications, it was enough for one person to send another person they already knew their public key, and the receiving person would trust that the person who sent it to them was the correct, valid person with the correct, valid, related private key.

But as the number of people in an asymmetric channel increases, not every participant may know and trust every other participant. One way to get public key trust in a person you don't know is to have someone you already trust vouch for the other person. For example, suppose Wilma wanted to communicate asymmetrically with Dino but didn't know or trust Dino ahead of time. But she knew that Fred knew and trusted Dino and could vouch for Dino and Dino's valid public key. Fred could sign Dino's public key with his own private key, which Wilma could then validate using Fred's public key. This is called a *peer-to-peer trust* (or *web trust*). This is how the very popular Pretty Good Privacy (PGP) encryption program works. But peer-to-peer trust systems don't work as well as the number of participants scale, particularly in global asymmetric systems where most of the participants don't know one another. Enter public key infrastructure.

Public key infrastructure (PKI) is a commonly used cryptographic framework and family of protocols used in the computer world to provide identity trust between unrelated parties. You may read or hear of many different descriptions of what PKI is and why it is needed, but at its base requirement, PKI primarily exists to authenticate subject identities and their asymmetric cryptographic keys involved in cryptographic transactions. Without this requirement you would not need PKI.

PKIs issue verified subjects "digital certificates," which are cryptographically protected documents attesting to the validity of a subject's identity and their associated asymmetric key pair. In practice, the subject (or something on their behalf) generates an asymmetric key pair for the subject to use. The subject submits their public key to the PKI (remember we don't share private keys). The PKI's *certification authority* service is then supposed to verify the identity of the subject submitting the public key.

The level of identity proof required of the subject by the PKI determines the level of *assurance* (or trust) the PKI can attest to. If the level of proof is very low (say just a valid email address), the issued digital certificate is considered to have low assurance. If the level of identity ownership and proof is substantial, such as the subject being required to appear in person and hand over a validated copy of their birth certificate and a national identification card to another human validator for verification, the assurance is considered high.

In any case, the primary job of a PKI is to verify the identity of the subject submitting their public key. If the subject's identity is validated, the PKI adds some additional information (such as validity

dates, subject name[s], certificate serial number, and the certification authority's name and identifier) and signs the subject's public key (and other information) with the PKI's private key. This creates a *digital certificate*. Figure 3.3 shows a partial example of a digital certificate highlighting the public key field. Theoretically, any entity who trusts the PKI (who issued a particular digital certificate) will trust any digital certificate created by the PKI and presented by the subject. The subject presenting the digital certificate is essentially saying, "I am who I say I am and a person who you trust verified it."

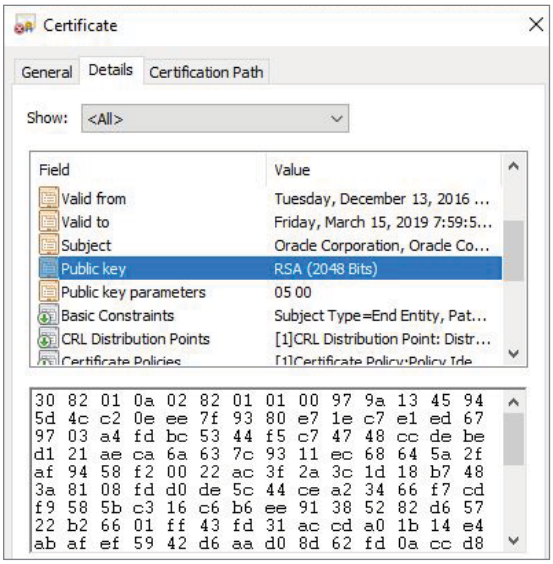

Figure 3.3: Details of a digital certificate

A PKI can be likened to the Department of Motor Vehicles (DMV) used in the United States. DMV license holders must substantially prove their identity to the DMV in order to get a driver's license. After the driver's (i.e., subject's) identity is successfully verified (assured), the DMV will take the subject's picture, add other information, and issue a DMV license, sealed along with the state emblem (somewhat like a real-world digital certificate). If the driver is stopped by law enforcement or goes to purchase something requiring age verification, they will often be required to present their DMV license. The officer and sales clerk trust the DMV license to be accurate and thus will rely on the information printed on the license during their verification process.

Much of the Internet works on PKI. Every time you connect to a website using Hypertext Transfer Protocol Secure (HTTPS), that website has an HTTPS/TLS digital certificate signed and issued by a trusted PKI. You may not personally trust that PKI, but your operating system or involved software does. When you connect to the website with your browser using the HTTPS protocol, the website sends you (or actually your browser) a copy of its digital certificate. The digital certificate, signed by

a PKI, attests to the website's name (often by URL), the website's public key, and other related important information. Once verified, your browser will then generate a brand-new shared session symmetric key, which it then securely sends to the website (using the website's public key). Then both the server and client can begin communicating securely using symmetric key communications (see Figure 3.4).

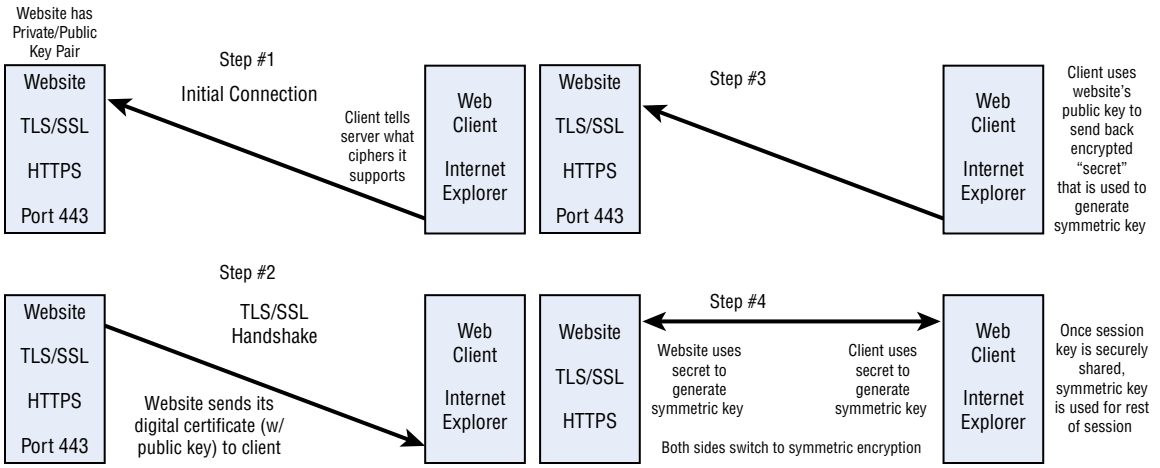

Figure 3.4: Web server and client using HTTPS and digital certificates to communicate with each other

In another popular example of PKI use, when you're downloading new software from popular vendors, the software will come with a digital certificate validating who signed the software (or the related integrity hash covered in more detail later) that allows the downloader (or more realistically usually a browser on their behalf) to validate that the software has not changed since the signer signed the software or hash. It doesn't matter where that software traveled between the signer and the receiver, whether it traveled over trusted or untrusted channels, how many intermediaries were involved, or how long ago the signing occurred (within reason). If validated, the digital certificate and accompanying validated hash tells users they can rely on the software to be as it was the moment it was signed and be from who it says signed it.

Asymmetric Cipher Weaknesses Asymmetric ciphers allow encryption and key exchange across untrusted channels and can be used to do authentication. Notwithstanding the recent impending threat posed by quantum computers, asymmetric ciphers have held up fairly well to decades of cryptographic attack. Still, they have their issues.

The biggest drawback is that asymmetric ciphers are inherently more mathematically complex than symmetric ciphers, and in the computer security world complexity is often an enemy of security. Most asymmetric ciphers use two cryptographically related keys separated only by a mathematical equation deemed difficult to factor. As compared to symmetric ciphers, there is an increased chance that someone will discover how to "shortcut the math" and more quickly factor the underlying math

equation or primes. And reality shows this is true (more on this later). All other things equal, asymmetric keys are usually longer than symmetric keys (but not always) and increase in size faster and more over time than symmetric keys to compensate for cipher attack advances.

Integrity Hashing

Another integral major cryptographic function is integrity hashing. *Hash algorithms* (also known as *hash functions* or simply *hashes*) are used to create unique output results for unique content inputs. They use "*one-way*" cryptographic functions that create/output a unique representative set of characters or bits (known as the hash, hash result, digital signature, or message digest) for examined unique content. Hash functions create cryptographic "digital fingerprints" of the content that they hash. Hashing functions can be used to cryptographically sign and verify the integrity of content, subjects, and other cryptographic objects.

When the integrity hash result (often known simply as a *hash* or *message digest*) is cryptographically tied to a particular cryptographic subject identity (e.g., user, device, or service), it is known as a *digital signature*. A verified digital signature allows the receiver of *signed* content to have confidence that the signed contents have not been altered since the signing of the content by the authenticated signer.

Secure, trusted hash functions have four important traits:

- For every unique input, a unique output result must be generated. This type of protection is known as *collision resistance*.
- Every time the same input is hashed, it should result in the same hash output.
- No two different inputs should ever result in the same hash output. This type of protection is known as *second preimage resistance*.
- If given the hash output, it should be *nontrivial* for anyone to derive the original content input. This type of protection is officially known as *preimage resistance*.

A good hash has all these attributes and even under sustained attack retains these protective hash capabilities. Collision resistance is related to and is similar to second preimage resistance, but they are not the same. And being good at both doesn't guarantee preimage resistance as they are unrelated attributes. If a hash falls susceptible to any of these attributes it is considered weak and should no longer be used.

Hash algorithms usually result in fixed-length hash results regardless of the input. Common hash lengths range from 128 to 256 bits. There have been many different generally accepted hash standards over the years, including Message Digest 5 (MD5), Windows LANManager (LM), Windows NT (NT), and Secure Hash Algorithm-1 (SHA-1). All of these previous standards, except for NT, are considered weak and broken.

Today, the most popular hashing algorithm is Secure Hash Algorithm-2 (SHA-2 or SHA2), although in 2015 NIST recommended that SHA-2's successor, Secure Hash Algorithm-3 (SHA-3 or SHA3), be

used instead as SHA-2 weakens over time from cryptographic attack improvements. So far, most people are still using SHA-2. SHA-2 has many different output sizes including 224, 384, 256, and 512 bits.

Table 3.1 shows some hash outputs for the word "frog" using common example hashes.

Table 3.1: Example hash outputs for the word "frog"

Hash Algorithm	Hash result for "frog"
MD5	938C2CC0DCC05F2B68C4287040CFCF71
SHA-1	B3E0F62FA1046AC6A8559C68D231B6BD11345F36
SHA-2	74FA5327CC0F4E947789DD5E989A61A8242986A596F170640AC9033 7B1DA1EE4
SHA-3 (512)	6EB693784D6128476291A3BBBF799d287F77E1816b05C611CE114AF 239BE2DEE734B5Df71B21AC74A36BE12CD629890CE63EE87E0F53BE987 D938D39E8D52B62

Hashing Weaknesses Like asymmetric ciphers, hash algorithms are considered a bit mysterious. They seem to do the job they are intended to do, but no one is completely assured that any particular hash algorithm can meet all four of the above requirements, or if they appear to do so currently, for how long until someone finds a mistake. Most of the previous hashing standards were at their time of use also considered secure and strong, until they were weakened over time by various cryptographic attacks. Cryptographers find hash algorithms among the most difficult cryptographic functions to accurately prove or disprove.

Cryptographic Uses

The major cryptographic functions of symmetric ciphers, asymmetric ciphers, and integrity hashing functions provide a wide range of services to the computer world, and by extension, to the real world. Without them most of the Internet and the real world as we know it would not be possible. Common cryptographic uses include the following:

- Encryption
- Authentication
- Digital signing
- HTTPS/TLS
- Cryptocurrencies
- Smartcards, virtual smartcards
- Disk encryption
- Network encryption

- Email encryption
- Virtual private networks
- Wireless security
- Code and document signing
- Steganography
- Anonymity
- Tokenization
- Data obscurity/erasure

Cryptography protects the world's networks, computers, vehicles, governments, currencies, and digital identities, and helps authenticate and protect all digital content. A world without good, reliable cryptography would look closer to the 1860s than the 1960s. The incredible reliance on digital cryptography is why anything that can easily and suddenly break it causes a shudder throughout the world. Quantum computing is the biggest threat to today's most popular digital cryptography we have ever faced.

You may be wondering about how any computer can break today's cryptography, especially when I wrote earlier that there wasn't even enough energy in the known universe to accomplish it. Well, that was when we had only classical binary computers and relied only on brute-force guessing to accomplish an attack. The invention of new quantum algorithms and of real, working quantum computers changed all of that.

How Quantum Computers Can Break Cryptography

Quantum computers are capable of breaking many forms of traditional cryptography because of their inherent quantum properties, covered in Chapter 1 (such as superposition and entanglement), coupled with quantum algorithms, which take advantage of those properties and shortcut the math. How quantum computers can break many forms of today's cryptography is the focus of this section. It will be followed by describing what traditional cryptography quantum computers can and can't easily break.

Cutting Time

There is a popular saying that the only thing we cannot get back in this world is time. This isn't always true, especially in the quantum world. Part of why we love and use computers is their ability to do something very fast. However, there are many potential solutions to problems that not even the fastest computers can solve. As previously covered, such is the case with many cryptographic math problems. The inability of a computer, even a network of millions of very fast computers, to solve some of today's known math problems is what gives much of today's relied-on cryptography its protective capabilities.

That's not to say that humanity doesn't try. Defenders and attackers both try to come up with problems and solutions that try to shortcut or extend the normal time to do something. These solutions and problems are categorized by how much they increase or decrease a particular (worst case) solution versus normal time scales.

If we add additional resources to solve a problem, such as more memory, a faster CPU, more hard drive space, or even more computers, if the addition results in no faster solution we call the solution *constant time*. For example, if it takes one person a day to make 100 widgets and we add a person and they still only make 100 widgets a day both together, the additional resource resulted in a constant-time solution. If you are trying to solve a problem faster, constant-time solutions do not help you and are counterproductive. If you are trying to defend against an attacker, if all they have is constant-time attacks, it's a benefit.

If adding resources speeds up the solution, it's good for the attacker and less good for the defender. If adding resources results in the same number of widgets being made by each individual added, it's called *linear time* (or *direct* time). For example, 1 person makes 10 widgets, 2 people make 20 widgets, and 3 people make 30 widgets (each person is making only 10 widgets, but together they make more widgets).

If adding each additional resource adds double the speed of each previous resource collection, this is called *exponential time*. For example, 1 person makes 100 widgets a day, 2 people make 200 widgets, 3 people make 400 widgets, and 4 people make 800 widgets, and so on. This is how binary computers (i.e., 2^n) inherently work. Each bit added doubles the power of the previous bit(s). Any resource addition, be it a computer resource or algorithm that can complete solutions faster than exponential time, is considered a threat to problems that are defended using exponential time or less.

Mathematicians and cryptographers have created problems and solutions that require far more work effort than exponential time. Time scale solutions known as polynomial, square root, quadratic, and factorial are all huge improvements on exponential time and are known as *superexponential time* scale solutions. Any resource providing these types of time solution improvements are a threat to things protecting themselves by relying on exponential time defenses. In particular, any cryptographic attack that exceeds exponential time is a threat to cryptographic solutions relying on exponential time protection. Qubits and quantum algorithms often give superexponential problems and solutions. If you read of a cryptographic problem or solution working only in exponential time or less, usually no one cares. But if you read of a solution working in one of the superexponential time scales, especially one of the fastest methods, such as factorial, everyone in the cryptography world cares. It means adding each additional resource gives a tremendous benefit over "normal," exponential time scale problems and solutions.

For more information on time solutions, see `https://rob-bell.net/2009/06/a-beginners-guide-to-big-o-notation/` and `https://stackoverflow.com/questions/4317414/polynomial-time-and-exponential-time`.

Quantum Algorithms

Quantum algorithms are a series of (mathematical) steps relying on quantum theories and properties, which if followed on a quantum device, will give a particular outcome. For decades, much of what quantum computers could possibly do was described only in theoretical papers. Having actual, working quantum computers and devices to try things out on has moved the world of theoretical quantum mechanics into the real world.

With a working quantum computer, scientists can take a problem that is addressed by a particular quantum algorithm, apply the algorithm, and see the results. Most quantum algorithms are considered revolutionary in the speed increases they give over traditional computers or in the types of once-thought-unsolvable problems they can answer. Ultimately, much of today's cryptography can be broken by a combination of quantum properties, computers, and algorithms.

There are dozens of well-known quantum algorithms. You can see a fairly inclusive list of the main ones here: `https://en.wikipedia.org/wiki/Quantum_algorithm` or `https://quantumalgorithmzoo.org/`. Many prove, at least in theory, that a quantum computer can do something better than a classical computer. Others move beyond theory and can be applied to solve real problems using quantum computers far faster than what is possible with traditional computers. A handful of algorithms have become essential enough to quantum computing and the promise of quantum cryptography that they are discussed thousands of times every day in online quantum and crypto circles. The following are three of the most important quantum algorithms as related to breaking today's traditional cryptography.

Grover's Algorithm

After Shor's algorithm (discussed shortly), Lov Grover's algorithm is probably the most discussed and beloved quantum algorithm. Grover's algorithm essentially proved that discovering the answer to any unstructured/unordered search (or math) problem can be done far quicker with quantum computers than traditional classic binary computers. Grover said that instead of needing to calculate all possible N solutions, one at a time, linearly, as was needed on classical computers, it could be done in the square root of N on quantum computers with $\log(N) + 1$ qubits. Grover's algorithm provides a quadratic workload speedup.

Suppose a math answer (or search) can be any of 1,000,000 possible answers (i.e., $N = 1000000$). A traditional computer would possibly need to complete 1,000,000 operations, in the worst-case scenario, to find that answer. Grover's algorithm proved that quantum computers, with 7 qubits ($\log(1000000) + 1$ qubits) could find the same answer in the square root of that same number of operations, or 1,000 operations at most. Square root solutions essentially halve (remember, every single-digit increase in an exponent doubles the previous base amount) exponential problem workloads.

Grover's algorithm can help crack symmetric (and asymmetric to a far lesser extent) cryptographic keys and solve some types of cryptographic hash functions far faster on quantum computers than on classical computers. Experts recommend that symmetric keys and hashes be doubled in size to keep their relative protection in the post-quantum world.

Fourier Transform

Joseph Fourier, who died in 1830, created a series of physics insights known today as the Fourier series. The Fourier Transform algorithm takes a wave (or wave function) and converts it to its constituent parts, much like following a recipe can help someone break down and re-create the same meal.

NOTE Thanks to `https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/` for the recipe allegory.

The Fourier Transform analyzes a wave and breaks out the discrete values for the wave's peak, value, amplitude (i.e., angles), frequency, and offsets. It essentially allows any wave function to be broken down and reconstituted as a sum of its frequency components. It is a way of linking and converting quantum particles across their wave-particle duality spectrum.

With the recipe allegory, imagine you have a delicious vegetable soup. The wave would be the completed soup. The Fourier Transform would let anyone following the same soup recipe (e.g., 1 cup chicken broth, 2 cups diced carrots, 1 cup diced onions, etc., cooking at 300F for one hour) to create the same exact soup, and vice versa. Many other quantum solutions and algorithms, such as Shor's algorithm (explained next), depend on a quantum-based Fourier Transform for their own success. Fourier allows calculations to move from quantum's particle-based properties to their wave-based properties and back again where each has their bigger benefits.

Shor's Algorithm

Mathematician Peter Shor is perhaps the most well-known figure associated with quantum computing and cracking traditional asymmetric cryptography in the modern era. That's because in 1994 he published an algorithm (in his paper titled "Algorithms for Quantum Computation: Discrete Logarithms and Factoring") that essentially provided a way for quantum computers to factor very large prime number equations very quickly (`https://pdfs.semanticscholar.org/6902/cb196ec032852ff31cc178ca822a5f67b2f2.pdf`). Shor's algorithm provided at least an exponential improvement, and likely *polynomial time* improvement, for factoring large primes. Using quantum computers with enough stable qubits, Shor's algorithm can factor very large prime number equations in seconds to minutes. His algorithm was and is still revolutionary. Upon its publishing, even before there was a practical application to test the theory, the computer world immediately understood its implications: that quantum computers could and likely would eventually become more powerful than classical computers. Crypto experts knew immediately that most of today's traditional public key crypto could be toast! It's the key reason why this book is being published at all. Everyone in the crypto world has been worrying about the coming day when quantum computers would start breaking traditional asymmetric encryption ever since.

Without going into all the math involved (it's not especially complicated—there's just lots of it), Shor's algorithm allows quantum computers to factor prime numbers faster by using an equation that takes a purely random guess at one of the prime numbers and turns it into a much closer guess,

which then quickly finds the actual prime numbers. Shor's algorithm uses the mathematical relationship of the two involved prime numbers in a way that dramatically cuts the number of guesses needed as compared to a classical brute-force method. A very large number of guesses is still needed, but when those guesses are done using the quantum property of superposition, they can be generated nearly instantaneously on a quantum gate computer. Within all those guesses are the right two prime numbers. All of the guesses are considered "Stage 1" or "Part I" of Shor's algorithm.

NOTE You will be hard-pressed to find a better explanation of Shor's algorithm with all underlying math and equations than this video: `www.youtube.com/watch?v=lvTqbM5Dq4Q`.

In the second stage, Peter Shor also figured out how to quickly determine which of the many created guesses are the correct two prime numbers. Although this is done mathematically, it's far easier to explain it using a visual allegory (see Figure 3.5). Each prime number guess is transformed into a sine wave (using the Fourier Transform). Then each guess's sine wave is added to the other possible guesses' sine wave. The two correct answers create the combined sine wave with the tallest peaks and lowest valleys. All the other incorrect guesses' combined sine waves interfere with one another more, causing smaller peaks and valleys, and thus a smaller overall combined sine wave. In the end, all the quantum computer has to do to find the correct two very large prime numbers is to find the tallest sine wave. This last step of Shor's algorithm is much like any of us being able to quickly identify the tallest person in a group photo. Doing so takes longer than a few seconds, but in time that can be measured in minutes (theoretically). Compared to the billions of years it might take a classical computer to figure out the two correct numbers, quantum computers are much more usable.

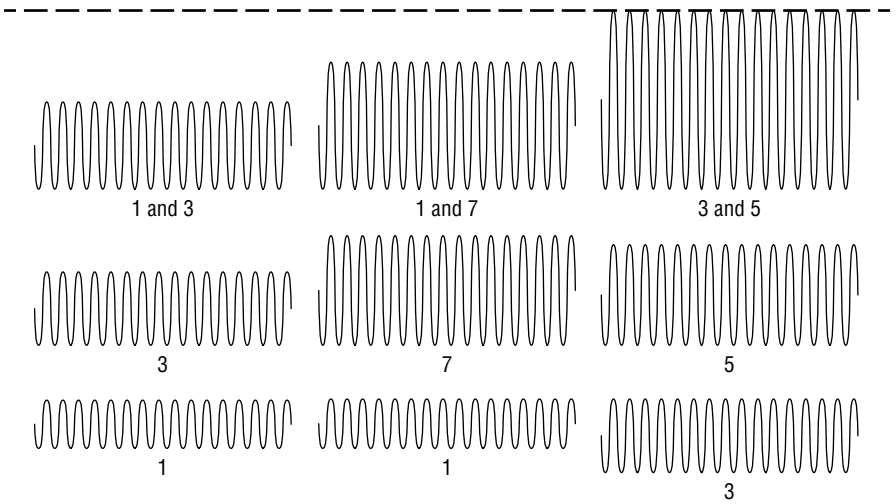

Figure 3.5: The right two prime numbers create the tallest sine wave peak when solving for an easy-to-understand small prime number equation

Beyond Shor's Algorithm

Other algorithms created since are touted to be even faster than Shor's, including GEECM (https://en.wikipedia.org/wiki/Lenstra_elliptic-curve_factorization#Quantum_version_(GEECM)) and this new announced advance, https://www.technologyreview.com/s/613596/how-a-quantum-computer-could-break-2048-bit-rsa-encryption-in-8-hours/. This means that Shor's algorithm is essentially a "floor" in how fast large prime number equations can be factored, and it's very likely that they can be solved even faster and/or with less qubits than Shor's predicted.

Also, adiabatic computers, which are a subclass of quantum annealing computers (covered in Chapter 2), have used adiabatic computation and related adiabatic theorems to factor large primes. In fact, as of 2019, adiabatic computers have factored far larger prime number equations than universal quantum computers using Shor's algorithm. But most observers believe that as universal quantum computers continue to be improved, they, along with Shor's algorithm, will far surpass the early progress made so far by adiabatic computers. Sort of a tortoise-versus-the-hare dilemma. In any case, there is a lot of computational progress being made by many different types of quantum computers and algorithms, and much of that progress points to the ability to break different types of cryptography.

So, the question of how quantum computers can likely break much of today's traditional crypto is answered by two near realities. First, quantum computers, in general, are likely to achieve quantum supremacy within the next year or two and will be capable of performing things that classical computers cannot easily do. Cipher algorithms that depend on the relative "slowness" of classical computers for their secret protection will be less strong.

Second, any cipher that relies on large prime number factorization (or the discrete logarithm problem or the elliptic-curve discrete logarithm problem) for their protection will be broken when quantum computers achieve enough stable qubits to perform more efficient quantum factoring algorithms against today's related cryptography. In short, quantum computers are faster and their quantum properties using quantum algorithms can "shortcut the math" that provided so much protection in a classical-computer-only world.

What Quantum Can and Can't Break

Quantum computers and quantum properties cannot magically break every known cryptography cipher. They can break only ciphers that rely on particular functions that are susceptible to quantum properties and quantum algorithms for their protections. This section will discuss what ciphers quantum computers are and aren't likely to break.

What Quantum Computing Can Break

As previously discussed, quantum computers can likely break any cipher algorithm whose security relies on the integer factorization problem, the discrete logarithm problem, the elliptic-curve discrete logarithm problem, or any other closely related mathematical problems. At a bare minimum, this

means the following ciphers and common applications (using those ciphers) are likely to be broken in the near future:

- Rivest, Shamir, Adleman (RSA)
- Diffie–Hellman (Key Exchange)
- Digital Signature Algorithm (DSA), also known as Finite Field Cryptography
- Elliptic Curve Cryptography (also known as Elliptic Curve Digital Signature Algorithm [ECDSA])
- ElGamal
- PKI (including digital certificates, digital signatures)
- HTTPS/TLS
- Most VPNs
- Hardware Security Modules (HSMs)
- Smartcards
- Most Wi-Fi security
- Cryptocurrencies
- Most two-factor authentication that relies on digital certificates (e.g., FIDO [Fast Identity Online] keys, Google security keys)
- Classical random number generators (RNGs)

Just including HTTPS/TLS means that most of the Internet's encryption will be broken. Adding in PKI-related cryptography means most business-related cryptography will be broken. Not all traditional cryptography is broken, but on a per-use basis, it includes much of the world.

NOTE It goes without saying that this assumption of what crypto can be broken applies to all those applications as they are commonly applied today, and they may not be easily broken if they use (or can be ported to) quantum-resistant ciphers.

What Quantum Computing Can't Break

Learning what quantum computers are likely to break is an impressive list of ciphers and applications. But not all of today's cryptography is susceptible to quantum computers (at least as far as we know today). Cryptography that is not known to be susceptible to quantum computers and algorithms is known as *quantum resistant*, or *quantum-safe*, or *post-quantum*. All three terms are used interchangeably by most cryptographers.

The following cryptography is known as being quantum resistant:

- Symmetric ciphers like AES (and applications and protocols relying solely on symmetric ciphers, like Kerberos and Network Switching Subsystem used by GSM cell phones) when used with "safe" key sizes
- Newer integrity hashes, like SHA-2, SHA-3, etc. when used with "safe" hash sizes

- SHAKE, a stream cipher
- Quantum key distribution (QKD), such as BB84, BBM, B92, COW, DPS, E91, and SARG04
- SNOW 3G, a word-based synchronous stream cipher
- Supersingular isogeny Diffie–Hellman key exchange (SIDH)
- Lattice-based ciphers
- Multivariate-based cryptography
- Code-based cryptography
- Some forms of zero-knowledge proof cryptography
- Quantum-based random number generators (covered in Chapter 7)
- Quantum-based ciphers

Many of these quantum-resistant ciphers will be covered again in the coming chapters.

All quantum-resistant cryptography and applications are currently assumed to be quantum-resistant but may become susceptible to quantum computing or new algorithms in the future if surprising new advances are made. Suffice to say, there are lots of quantum-safe ciphers and mechanisms to protect us in the future, but even those have risk. Part 2 of this book, "Preparing for the Quantum Break," discusses these quantum-safe ciphers and implementations.

NOTE Nothing is unhackable. Not even quantum-safe ciphers, not even if the most brilliant quantum scientist in the world tells you so. You will read and hear that quantum ciphers and other cryptographic devices and functions are "unhackable." And though that may be true at the theoretical level, things are a bit different in the real world. Humans have yet to be able to make something unhackable, even when starting with an "unhackable" theory or property. The quantum world does not change this, although quantum properties could make hacking much harder to pull off.

Why Symmetric Ciphers and Hashes Are Quantum Resistant Before we move on, I want to discuss why symmetric ciphers and hashes are quantum resistant in particular. Modern digital cryptography deals with protection bit sizes, which make classical brute-force attacks nontrivial to accomplish. There just isn't enough computational power to break the cipher keys generated and used by modern-day cipher algorithms. This is true even if we get an incredible speedup from quantum computers (even using Grover's algorithm) and even with quantum mechanics' incredible properties.

For example, just because a quantum computer can use superposition to generate every possible answer at once, that doesn't mean a quantum computer can just pick out the right answer and hand it over to the classical world. There has to be a way to pick out the correct answer out of the trillions that a quantum computer can generate for a particular answer. That's the glory of quantum algorithms. These algorithms use the quantum properties to essentially "shortcut" the brute-force math to enable us to find the right answer with fewer guesses or find the right answer out of trillions of answers.

Shor's algorithm helps quantum computers factor large prime number equations because it includes math logic that allows solutions to be found much faster than using simple brute-force methods and to find the right answer out of many guesses. Part of the reason that quantum computers can break most traditional public key crypto is that the math that public key crypto relies on has a "weakness" that quantum computers and algorithms can take advantage of. The brilliance of Shor is that he was able to create a faster math solution that could happen only because of quantum computers.

But not all problems have characteristics that are overly susceptible to quantum solutions. This is true of traditional symmetric ciphers and hashes. Grover's algorithm uses a square root time reduction to cut the protection of symmetric ciphers in half. This is a significant reduction in protection but not fatal (as it would be if the time reduction were polynomial, quadratic, or factorial). Any quantum computer attacking those types of cryptography may be a lot faster than classical computers, but the number of key bits is still so overwhelmingly large that the "lot faster" doesn't significantly weaken the protection power of those ciphers or hashes.

In general, it is believed that simply doubling the key size of traditional symmetric keys and hashes will allow them to remain quantum-safe for the foreseeable future—unless some new, unforeseen, related quantum break gets discovered. So, simply moving from AES-128 and SHA-256 to AES-256 and SHA-512 is considered a long-term solution to those concerns, and you should be doing that today. For more reading on this subject; consider the following excellent paper: `https://arxiv.org/pdf/1804.00200.pdf`.

NOTE Not all cryptographers think that all attacks against hashes are improved using quantum computers. Some of the brightest cryptographers think that quantum computers can actually be worse at some forms of hash attacks (i.e., collision finding) than classical computers. See `https://cr.yp.to/hash/collisioncost-20090517.pdf` as a good example.

NOTE Not all hashes are quantum safe. Using Grover's algorithm, some weaker hashes can be broken faster. But SHA-2, SHA-3, and the other modern-day hashing algorithms are considered strong and safe against known coming quantum attacks when using the appropriate key or hash sizes.

Still Theoretical

It's important to remember that until quantum supremacy is achieved (which many vendors are saying will be as soon as 2019), all of the feared crypto breaks are mostly theoretical. Shor's algorithm has been proven on quantum computers. It works as predicted. But so far, the largest factored prime number equation using Shor's algorithm on a quantum computer has been $7 \times 3 = 21$, which can easily be solved by a child. Your smartphone has more raw computational power than most quantum computers.

NOTE The largest primes factored by Shor's algorithm are super small, but there have been far larger prime number equations factored by quantum computers not using Shor's. See `https://crypto.stackexchange.com/questions/59795/largest-integer-factored-by-shors-algorithm`. Although once enough stable qubits have been created to allow Shor's to attack the bigger primes, these other records should quickly fall. As an additional note, remember that Shor's is just a floor (or ceiling depending on how you look at it) for qubits needed. It was created in 1994 and the world is already full of other algorithms that claim to make it old school.

But the entirety of quantum mechanics and quantum computing has always been a cycle of a theoretical beginning that then moved into the real world. Quantum mechanics were theorized for decades before proven beyond a shadow of a doubt by Einstein in the 1930s. The first model of a quantum computer was theorized in 1985 (by David Deutsch) followed by the first working quantum computer in 1998. Shor's algorithm was theorized in 1994 and then proven on a quantum computer in 2001 by IBM. Quantum computers are growing qubit by qubit. Qubit stability and error correction are getting stronger every day. Now we have several vendors with various quantum computers feeling confident that quantum supremacy is just around the corner. Once quantum supremacy is reached, breaking traditional cryptography won't be too far behind. Chapter 4, "When Will the Quantum Crypto Break Happen?" will cover when quantum computers will break most traditional cryptography in far more detail.

Summary

This chapter covered how quantum computing is likely capable of breaking most forms of traditional public key encryption. It started by discussing cryptography basics, paying particular attention to how most of today's public key encryption schemes provide protection. We then explored how quantum computers can break that protection and what type of cryptography is overly susceptible to quantum cracking. Chapter 4 discusses when the theoretical attacks of quantum computers will likely become a reality.

4 When Will the Quantum Crypto Break Happen?

This chapter will cover the main factors involved in determining when the quantum crypto break will happen, when it might likely happen, and how it will likely play out. It will give an answer so many other quantum computing experts want to avoid: exactly when will the quantum crypto break happen?

It Was Always "10 Years from Now"

Since 1994 with the release of Peter Shor's quantum algorithm, computer scientists and cryptographers around the globe have understood that quantum computers, if we could even make them, were likely to break existing public key ciphers and other cryptography. And if we could make them, it was just a matter of how many years afterward until the cryptographic breaks would start happening.

And for over two decades, when most of us were asked for our opinion of when it might happen, we answered, "Within the next 10 years!" I said the same thing whenever anyone asked me. Saying "Within the next 10 years!" was basically saying, "We really don't know. Maybe soon. Maybe decades away!" It was a fudge answer.

Let me cut to the chase. No one really knows when it will happen. Anyone telling you different is taking their best guess or is part of some secret group who has already broken it but is sworn to secrecy. There is even a chance it will never happen. We could find some startling technological wall that prevents it from ever happening! There are those who believe that "never" is the likely outcome. There are those who believe that everything we are describing as quantum computers really is something else and we are all mistaken. But in general, the average person following quantum physics, if asked to guess when the quantum crypto break would happen, would often say, "Within the next 10 years."

Once, nearly a decade ago, when I was stepping down from one of my frequent talks on quantum computing and the likely future quantum break, someone from the audience asked me when I thought it was likely to happen. I said, "In the next 10 years!" as I always did. Famed cryptography and industry luminary Bruce Schneier, to whom I've looked as an informal mentor, was going up on stage

Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto, First Edition. Roger A. Grimes.

after me to give a keynote speech. As he passed by me, he asked me in a quiet aside, "Roger, how long have you been saying that?" I realized that I had been saying "10 years" as an answer to that question for nearly 20 years. Time had moved on and my answer hadn't changed. It made me question if I really understood how far away the quantum crypto break was. It made me realize that I didn't really know. No one knows. And that was nearly 10 years ago.

But now if you asked me for my best guess today, I would tell you the quantum crypto break either has already happened, or more likely is going to happen in the next few years. I believe there is a strong chance that not only will the crypto break happen in the next one to three years, but most of the world will be unprepared for it. Hence, the whole reason I wrote this book: to help you better prepare for the coming reality as best as I can. I'm not a quantum physicist and I don't make quantum computers for a living. I'm knowingly risking my professional reputation by proclaiming my prediction of a quantum break sooner than later.

So, what changed to make me put a stake in the ground and predict a particular timeline when no one really knows for sure? Aren't I being like all of the hundreds, if not thousands, of previous failed doomsday soothsayers? They have predicted biblical catastrophes, asteroid collisions, particle-accelerating-antimatter-eating vortexes, and end-of-the-world scenarios of every possible cause. What changed to make me stop saying "10 years from now" and start to think it might happen within a few years? I cover that in the next section.

Quantum Crypto Break Factors

When people discuss whether they think quantum computers will break traditional cryptography, what they really are debating is the reality of quantum mechanics, its implementation in real-world quantum computers, and how capable those quantum computers are of setting about breaking traditional cryptography. In this section, we will discuss many of those factors and take a reading on where we are currently and where we will be in the near future. I'll tackle these factors one by one.

Is Quantum Mechanics Real?

Yes, quantum mechanics is real. As covered in Chapter 1, quantum mechanics and most of the involved quantum properties (photoelectric effect, wave-particle duality, entanglement, uncertainty, tunneling, etc.) have been proven over and over to exist. Quantum mechanics is one of the most proven and accurate sciences in the world. We don't often know how or why a quantum property works—and that is very disturbing for all involved—but quantum mechanics is not imaginary. Not only is it real, but how scientists have predicted it would work has been proven, over and over. Sometimes the world's best minds have attempted to prove that "weird properties" of quantum mechanics were due to something else that we were missing. And in every case the experiments to prove that quantum mechanics was not quantum mechanics has failed. Every experiment created to show that quantum properties exist and will react a certain way have been proven. It's quite the experiment record.

Are Quantum Computers Real?

Yes, quantum computers are real. Although most of today's quantum computers are not very powerful (annealing quantum computers are the notable exception), they still use—and are verified to use—quantum properties to do their computations. This is important. It's yet another way to prove that quantum properties are real. Until the first real working quantum computers came into existence, there was a chance that we mere humans would not be able to capture, harness, and manage quantum properties for our own bidding. Up until the first quantum computer was created, all we had were a lot of theories about how a quantum computer would look and operate. And then, in 1998 the first real-world quantum computer was created and we no longer had to worry. And it's been a helluva exciting ride in the 21 years since.

The central hurdle to worrying about quantum computers breaking today's traditional public key cryptography, especially after Peter Shor's prime number factoring algorithm was released in 1994, was *if* we could build a quantum computer. There were many naysayers. And then four years later we did it.

Without that single pinnacle achievement, none of the rest is possible. But we did indeed do it. We now have over 80 different quantum hardware groups in the Western Hemisphere alone that we know about—and likely many, many more worldwide we don't know about. The biggest hurdle was whether we could build a single quantum computer at all and we did it. The most difficult, supposedly "impossible to achieve" part was done. The first qubit was the hardest. It seems to me that going from 1 qubit to a million qubits is far less difficult of a problem.

NOTE Interestingly, a few very smart scientists are saying we still haven't truly achieved quantum computing, and they make scientifically or logically supported arguments to support their case. But with each new type of quantum computer and successive proof of successfully demonstrated quantum algorithms and solutions, those claims get harder to even consider.

Is Superposition Real?

Yes, superposition is real. You can't solve all the hard-to-solve classical computer problems like large prime number equation factoring without being able to generate a lot of answers at once . . . actually, *all* the possible answers instantaneously. There have been hundreds if not thousands of experiments that have proven that superposition is real. In 1996, a single atom was shown to have superposition of states (`https://quantumsciencephilippines.com/seminar/seminar-topics/SchrodingerCatAtom.pdf`). Since then, trillions of atoms (`https://arxiv.org/abs/1310.8343`) and tens of thousands of molecules in aggregate have been shown to demonstrate superposition. More importantly, all quantum computers use superposition as one of their key quantum properties. They could not exist and function as they do without superposition.

Is Peter Shor's Algorithm Real?

Yes, Peter Shor's algorithm is real. Without Shor's algorithm being applied in the real world and factoring large prime number equations fast, traditional public key crypto and other ciphers would remain safe for the foreseeable future. Even though so far the quantum computers that have used Shor's algorithm have not factored large prime number equations, they have used Shor's algorithm and showed it worked exactly the way Shor predicted it would work. IBM's early quantum computer in 2001 used Shor's algorithm to factor a prime number equation. This question is answered; we just need more stable qubits to solve the very large prime number equations. Shor's algorithm being verified as true also means that the other quantum algorithms it relies on, such as the Fourier Transform, are also accurate and true.

Do We Have Enough Stable Qubits?

No, we don't have enough stable qubits. This is the current holy grail of quantum computing. Running Shor's algorithm to factor any prime number equation requires $(2 \times n) + 3$ stable qubits, where n is the number of key bits to crack. Thus, to crack a 2048-bit RSA key, we need 4099 stable qubits, and to crack a 4096-bit RSA key, we need 8195 stable qubits. And we need the stable qubits to be on the right kind of computer. Shor's algorithm isn't as useful on a quantum annealing computer (as discussed in Chapter 2). So far (as of this writing), we have only a number of universal quantum qubits ranging under 100, and even these aren't as stable as we need them to be. That's a far cry from the over 4000 stable qubits needed (or potentially 4,000,000,000 overall qubits including ancillary qubits by some calculations). The question is how fast will the number of stable qubits rise?

I will say this about human ingenuity. Once we figure out a very difficult hardware thing, we are really, really good about making lots of them. Like back in World War II, Alan Turing and his team (based on the previous work done by hundreds of allies) finally figured out what it would take to crack the German Enigma codes. Turing would have to basically invent the first real working computers, which he did. And then he figured out that he needed hundreds of them. And he got them. The first radio and television were super hard to make. The next million not so hard. You could say it took hundreds or thousands of years to get to the place where the first of something was created. Usually the next million doesn't take half a decade. The first super stable qubit is the hardest. Going from one to a billion of them just isn't as hard.

At the same time, many scientists are working on factorization optimizations over Shor's algorithm to reduce the number of needed qubits. Shor's algorithm is a ceiling (i.e., the maximum number of qubits needed), not a floor. It is likely that Shor's algorithm will be significantly improved over time as the number of stable quantum qubits is also improved, so the number needed will decrease. So today, at least publicly, we don't have the necessary number of stable qubits, but we are getting there using two synergistic approaches meeting in the middle. One is adding more stable qubits and the other is requiring fewer stable qubits in the first place.

Qubit Stability and Error Correction

Are qubit coherence and decoherence currently where we need them to be for quantum supremacy and the quantum crypto break? No. But like the sheer number of qubits, stability and error correction are on the rise. There does appear to be solid improvement of both stability and error correction every quarter. There are even very solid chances for the most stable quantum computer types we know of right now, the Majorana fermion (Microsoft) and ion traps, to add more qubits. And each of their qubits are very stable qubits. These vendors can spend more time, money, and other resources on just adding qubits than trying to stabilize and error-correct what they have. It's a race to see which quantum technology wins out: lots of qubits with lots of error correction or fewer stable qubits period.

Quantum Resources and Competition

One very strong sign that we are likely to solve the existing quantum issues sooner than later is the sheer amount of resources being thrown against the remaining problems. All the major countries are spending tens of billions of dollars. It's becoming a top government priority for many of them, and even the smaller countries are partnering with the larger ones. All of the biggest technology and computer companies, along with the top universities, in each of the countries are involved. It reminds me of another global project just over a half decade ago.

In the 1950s, few people thought we would have humans on the moon by 1969. What's even more startling is that the United States' major push to get astronauts on the moon didn't happen until John F. Kennedy's famous 1961 declaration. Eight years later, after numerous mistakes and catastrophes, U.S. astronauts landed on the moon. I cannot think of another project with the same amount of focused global resources competing to be the first. It certainly feels like a moonshot project to me.

Do We Have Steady Improvement?

Yes, we have steady improvement. All of those global resources and competition are pushing a steady improvement in all things quantum computing. The number of qubits is increasing. Qubit stability is increasing. Error correction is getting better. Quantum logic gate speed is increasing. The number of quantum computer types is increasing, and improved quantum inventions happen on almost a weekly basis.

More quantum algorithms are being invented and older ones proven in use on working quantum computers. Multiple quantum processors are available. Over a dozen quantum programming languages, scripting languages, and compilers now exist. Quantum networking is no longer a dream. Quantum random number generators are in use. Multiple vendors feel that quantum supremacy is just around the corner. There don't appear to be any major setbacks or hurdles that people think cannot be solved.

Some critics liken quantum computing and the quantum break to nuclear fusion. Nuclear fusion is when the nuclei of two or more atoms combine to become one. Fusion produces large amounts of energy. It's how the sun generates its heat, energy, and light, and has long been thought to be the

energy technology that will power Earth. But after over 80 years of research and development, and billions of dollars spent, we appear no closer than we were in the beginning. Critics call quantum computing the next fusion cash call. They believe that quantum-based groups are overhyping the promise of quantum computing to get large amounts of funding dollars.

But there is a huge difference between fusion research and quantum research. Fusion research barely moves decade to decade. Most experiments are practical failures. We still don't have a working fusion reactor. In the quantum world we have working devices. We have constant improvement. We have constant progress. This does not appear to be a science that is getting stymied or slowed down. It's the opposite.

Expert Opinions

Lastly, for a long time nearly all quantum computing experts agreed that quantum supremacy was at least a decade off. Now the opinions are starting to conflict, and more and more voices are starting to think that quantum supremacy is just a year or two away. Many quantum computing experts believe that the quantum crypto break is only a few years away.

Mark Jackson, Scientific Lead of Business Development at Cambridge Quantum Computing, is one of those voices. He and Cambridge Quantum Computing are helping several different quantum computing projects. He is in the thick of quantum computer technology, and he understands where it is currently and where it is going over the next few years. He has publicly predicted that the quantum crypto break is likely within the next few years and certainly under 10 years. This isn't the same "10 years from now" fudge we use to say. Back then we didn't (at one point) even have quantum computers, much less a hundred working quantum computers, teams all over the world, and tens of billions of dollars tackling the problem. Now it's not a "we don't really know" answer, but one based on the steady, methodical advances in existing quantum computing sciences. And Jackson's not alone.

When the Quantum Cyber Break Will Happen

When all the factors needed to break traditional cryptography are considered, a strong argument can be made that it will happen sooner than later. Some experts have said they will be more surprised if it doesn't happen within the next half decade than if it does. With that said, no one knows when it will happen until it is done. A reasonable person should look at all the possible timing scenarios and consider which scenario appears more reasonable on a risk-adjusted basis.

Timing Scenarios

There are four possible broad time periods when the quantum crypto break is going to happen (or not happen): it's already happened, but we don't know about it; it will happen in the next few years; it will not happen in the next few years, but eventually will; or it will never happen. These are the only four possible outcomes, and each will be covered in more detail next.

It's Already Happened

There is a very real possibility that quantum supremacy and the quantum crypto break have already happened within a private entity and the rest of the public world simply does not know about it. It is generally believed that if a major country's government was able to obtain quantum supremacy and, in particular, perform the quantum crypto break first, they would have every incentive to keep the accomplishments silent.

NOTE The world's governments are good at keeping crypto secrets. Clifford Cocks of the UK's Government Communications Headquarters (GCHQ) created what we now call the RSA cipher in 1973, and Malcolm J. Williamson discovered what we would later call Diffie–Hellman in 1974. Both were passed along to the U.S. National Security Agency (NSA) soon after. Their separate publicly announced re-creations by Diffie, Hellman, and Merkle in 1976 and by RSA in 1977 did not make either the UK or U.S. governments acknowledge their previous existence for decades. In fact, the UK government did not acknowledge its role as the first creator of public key cryptography until 1997 (24 years later).

Most of the world's spy agencies would love to have obtained the quantum crypto break and kept it secret for as long as possible. Then they could spy on many entities and other governments that were still relying on traditional public key crypto, thinking it was still safe to use. I believe most crypto experts expect their governments to keep it a secret if their government obtains access to it first, before someone or some entity in the public sector does.

This theory can be entertained even more so because several countries and companies have already (prior to 2019) claimed that they have already obtained quantum supremacy or were very, very close to doing so. This includes Google, IBM, and Alibaba. And then all of a sudden many of those public voices went silent. Many are wondering why. Personally, just as my best guess without any particular real knowledge of its likelihood, I give this scenario a 15 percent chance of being true.

In the Next Few Years

Your author and many others think that quantum supremacy and the quantum crypto break are only a few years away, or sooner. It's definitely not a majority view, but it has growing support every day. Quantum supremacy is likely to be obtained in the next year or so (Google, IBM, and others have been vocal about that fact). I don't think it is a huge risk to agree with what Google and IBM are predicting.

What quantum supremacy means when it is obtained is another matter. Although quantum supremacy will be the historic tipping point moment when quantum computers can do things that classical computers cannot, it is unlikely to mean the world will be immediately different on a measurable level. The day after quantum supremacy is reached is the beginning of a lot of work—work that will no doubt lead to many great discoveries. But they all won't be realized in a day. It will take years and many more incremental advances for the fruits of most of those labors to be accomplished,

just as it did when electricity, the lightbulb, radio, the telephone, television, and the Internet were invented.

The quantum crypto break will absolutely follow quantum supremacy, but its timing afterward is also impossible to predict at this time. The central question is how long will it take universal quantum computer vendors to move from under 100 stable qubits to over 4000? There's a good chance that once qubit stabilization and error correction gets "figured out," the numbers will move pretty fast.

In the classical world, once we figured out how to fit a lot of transistors on a piece of silicon, the number of them in the same space doubled about every 18 months to two years (following a prediction known as Moore's law). The number of qubits created and able to be used with a particular type of quantum computer has so far risen in a less predictable manner, although it might be likened to the very early days of microprocessors. Table 4.1 shows the number of qubits used by various quantum computers by year comparing non-annealing and annealing types (just as a comparison).

Table 4.1: Number of qubits used by various quantum computer types by year

Year	Non-annealing	Annealing
1998	3	
2000	7	
2006	12	
2007		28
2012		84
2015		1000
2017	50	2000
2018	72	

It's important to remember, as it was covered in Chapter 2, that a quantum computer, its speed, and what it can accomplish is more than just the number of qubits. If you look at the history of quantum computing so far, nearly all of the components are being improved, as well as new components and combinations that didn't exist before. We only know that steady improvement is being made on all quantum manufacturing fronts. I put the likelihood of this scenario at 30 percent.

After the Next Few Years

This is essentially the standard "within the next 10 years" response. It won't likely happen in the next few years, but quantum supremacy and the quantum crypto break will happen sometime in the future. The vast majority of quantum scientists fall into this category. One of the notable differences

between this timing scenario and the former ("in the next few years") is that quantum scientists actually working directly on quantum computer products think it will happen sooner than 10 years. This is a notable event. The scientists aren't sure when it will happen, but they are starting to feel comfortable that it will happen, let's say, in the next five to seven years and don't believe we have to wait 10 years. That's a big shift in thinking. There is, of course, always the chance that quantum supremacy and the quantum crypto break will be off decades. After all, no one knows exactly when it will happen. I put the likelihood of this scenario at 50 percent.

It's Never Going to Happen

A smaller percentage of quantum computing experts think it will never happen. They see the remaining issues of large-scale quantum computing as insurmountable. Some even contend that the quantum computers we have today are not truly quantum. They argue that we are seeing what we want to see in a world that we really still don't know enough about. They believe that each type of quantum computer we have created will eventually encounter an issue that prevents them from truly progressing beyond the crude abacus-like contraptions we have created today. Their beliefs cannot be dismissed out of hand. Some of these believers are among the most brilliant minds in our world. They know far more about quantum computing than most of us.

Still, I would not put my money on this timing scenario. Never is a long time, and a lot of previous brilliant minds, including Einstein, went to their graves questioning the completeness of quantum mechanics, even though quantum properties they discounted were eventually proven to exist. I would put the likelihood of this scenario at 5 percent or less. Even more important, the vast majority of the most knowledgeable quantum scientists and the U.S. government do not put much faith in this last timing scenario (more on this in the next section).

One of my favorite quotes on the matter comes from University of Texas Austin quantum professor Scott Aaronson, who writes in his book *Quantum Computing Since Democritus*:

> *[I]f scalable quantum computing were proved to be impossible, that would excite me a thousand times more than if it were proved to be possible. For such a failure would imply something wrong or incomplete with our understanding of quantum mechanics itself: a revolution in physics!*

When Should You Prepare?

So, you may be asking yourself, "If no one among the quantum computing experts agrees when it will happen, should we even begin preparing now for the coming quantum crypto break?" You might be fearful that you would be wasting valuable time and resources to begin focusing on something that might be many years, if not decades, off. "There are a lot things to worry about in the world of computer security, many of which are far more pressing than some theoretical, pie-in-the-sky, Y2K

problem!" you may be thinking. And you could be forgiven for thinking this, especially because the vast majority of your fellow computer security practitioners are currently completely unaware and doing nothing. You might even take the "conservative position" and think you are maximizing resources by waiting until you hear about quantum supremacy and the quantum crypto break actually happening. You might feel there is more safety in waiting for the masses to respond when the actual threat is finally realized, like fish moving in a synchronized agreement when fleeing a predator. You may believe that waiting for the break to be announced is both efficient and most cost-effective.

Well, unlike the timing scenarios where no one really knows what the answer is, the answer for when you and your organization should start preparing for quantum supremacy and the coming quantum crypto break is now! There are many things you should be doing now (covered in Part II of this book) that will be cheaper and easier if you start to do them now. And there is the very real possibility that waiting until the quantum break happens is too late to protect your organization's secrets. If your competitors or interested nation-states care enough about your most sensitive, protective data, they may already be siphoning your encrypted data, waiting for the day when they can see into it using quantum computing. Even if you think you have zero sensitive data that might be useful to an adversary, it is clearly cheaper and more efficient for you to start preparing for the post-quantum world now. I'm not alone in this recommendation.

NSA Says Now

In 2016, the United States National Institute of Standards and Technology (NIST), the National Security Agency (NSA), and Central Security Service (CSS) said "now" is the time to start preparing for a "post-quantum" world. They said so in in the *NSA/CSS Information Assurance Directorate Commercial National Security Algorithm Suite and Quantum Computing FAQ* (`https://cryptome.org/2016/01/CNSA-Suite-and-Quantum-Computing-FAQ.pdf`). It's very plainly stated that, regarding post-quantum preparation, "NSA believes the time is now right—consistent with advances in quantum computing."

If someone in your organization is asking whether you should be preparing for quantum supremacy and the coming quantum crypto break, show them this document and that section in particular. It's pretty clear. And it's at least three years old (as of 2019). So regardless of the possible timing scenarios and their likelihood, the best scientific minds of our country and the ones most likely intimately familiar with quantum computing advances and the remaining technological hurdles are telling the world it needs to prepare now. It sounds like they aren't seeing the cost–benefit argument of the last possible timing scenario of never.

National Academy of Sciences Says Now

In 2018, the U.S. National Academy of Sciences published a consensus study report titled *Quantum Computing: Progress and Prospects* (`http://cs.brown.edu/courses/csci1800/sources/2018_NAE_QuantumComputing_ProgressAndProspects.pdf`). Key Finding number 1 says that the prospect of quantum computing breaking even RSA-2048-bit is off by at least a decade. This is one of the most relaxed conclusions regarding when the quantum break will happen that I have read. But it is followed by Finding 10, which says

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

That is to say, start preparing now.

Mosca's Inequality

In 2015, University of Waterloo's Michele Mosca, stated that, we need to start worrying about the impact of quantum computers when the amount of time that we wish our data to be secure is added to the time it will take for our computer systems to transition from classical to post-quantum greater than the time it will take for quantum computers to start breaking existing quantum-susceptible encryption protocols. For example, if you need your critical data to be secure for 10 future years and it will take you 5 years to transition, then you need to start moving to post-quantum systems before 15 years out from the post-quantum world. When Mosca was first stating his conclusion, he picked 2020 as that future point in time, but general consensus is that for most organizations needing high data security we passed it in 2017. Here is a great article on the Mosca Inequality by Cambridge Quantum Computing co-founder Ilyas Khan: `www.linkedin.com/pulse/moscas-inequality-why-matters-ilyas-khan-ksg/`. I'll re-cover Mosca's Inequality in Chapter 9 with more detail.

Breakout Scenarios

To complicate matters a bit, no one is sure what the "breakout" scenarios will look like when we do reach quantum supremacy and the quantum crypto break. Will the power of what can be achieved and broken stay in the hands of a few, or will it be like the discovery of public key cryptography where the entire world ends up using it within a few years? Let's look at some of the likely breakout scenarios if and when it is achieved.

Stays in the Realm of Nation-States for a Long Time

Many nations are spending billions of dollars to be the first to achieve quantum supremacy and the quantum crypto break. Building quantum computers at scale requires tens if not hundreds of billions of dollars in investment. One possible breakout scenario is that one or more nation-states achieves these quantum goals, likely within a few years of each other, and that is where the power stays. The governments will guard and limit quantum power, because of its ability to break today's digital codes, in as few hands as is possible. They will want to limit the possible damage and ensure that their own nation's secrets are adequately protected before releasing quantum power to the general public.

This scenario might be seen as how most nations treat atomic weapons. They were achieved at great cost at the nation-state level and, once achieved, are highly protected. Each nation that enters the "nuclear club" is fought against, and once they are in the club, most of the other club members try to prevent future membership. Laws are put in place, nationally and globally, to prevent nuclear weapons from falling into the hands of anyone outside authorized top-secret government agencies.

You might be skeptical of the equating of quantum computing to nuclear weapons, but remember that many governments consider strong cryptography to be a top national secret. The United Kingdom and the United States didn't release the fact that the UK invented public key crypto for decades after it was used in the public domain. Even today, individuals and companies in many large nations, including the United States, are prevented from exporting strong cryptography to other nations. It may be a crime to even post strong cryptography on the Internet that would allow people in other nations to download it. Strong cryptography is considered "munitions" and covered by the U.S. Arms Export Control Act. Failure to follow national cryptography export laws may be considered treason and can result in the death penalty.

In the past, many citizens have been threatened with the conviction of treason for even sharing commonly used cipher algorithms in software products and on the Internet. In the early 1990s, Phillip Zimmerman (`https://en.wikipedia.org/wiki/Phil_Zimmermann`) was made an international cryptographic martyr threatened by the U.S. government for his use of common cipher standards in his freely downloadable Pretty Good Privacy (PGP) software program.

Suffice to say, if nation-states thought they could keep quantum computing to themselves they would. It is not out of the realm of possibility that governments may allow quantum computing but make the cracking of traditional cryptography illegal, even if it were possible by the general public. Governments might even create laws that prevent quantum computing manufacturers from allowing their computers to be used to crack traditional public key crypto. Could Shor's algorithm be outlawed? Could someone be arrested for making their own implementation of Shor's algorithm if it was outlawed?

Again, you might be skeptical of the possibility, but there is already a similar situation regarding printers, scanners, and copy machines (and other devices like fax machines and photo editing software) being forced by the U.S. government to prevent the printing of realistic U.S. currency. Unbeknownst to most people, most copiers, printers, and scanners (and software) contain coding that prevents the realistic printing of money. I would tell you to try copying and printing currency, but it's highly illegal to even try.

NOTE There are many, many U.S. laws that prevent the copying and printing of legal currency. Many people believe that devices contain code to prevent the printing of currency, but that is a widespread faux rumor. A "friend" of mine recently attempted to copy and print different dominations of U.S. currency across a wide range of devices and found that he could copy fairly realistic copies of currency and even save them as document files, but any attempts to print the currency at 100 percent scale were prevented. He could print very large copies of the money, but all attempts to print at the regular scale resulted in printing errors, refusals to print, and cutoff copies. This may not apply to all devices, but it did occur on the devices my friend tried in his limited test. Here is a YouTube video with more details: `www.youtube.com/watch?v=1c-jBfZPVv4`.

Used by Biggest Companies

There is the argument that the cost of developing quantum computers will keep quantum crypto cracking in the realm of nation-states and only available to the biggest companies with enough resources to make, buy, or rent time on large-scale quantum computers. Shor's algorithm requires quantum computers with over 4,000 very stable qubits to break today's most common public key sizes. Once that type of scale is reached, it is likely to be among the most expensive computers, costing far more than room-sized mainframes and even millions of hosted cloud virtual machines. The sheer economics of anything new, amazing, and very difficult and rare to build says that large-scale quantum computing will be super expensive for many years.

Sure, one day we may get a quantum computer (or processor) on everyone's desk, but that particular reality is definitely decades away. The first traditional computer microprocessor (at least what the vendor called a microprocessor) was created in 1968. Although microprocessors were in widespread use in expensive computers and somewhat expensive calculators, the idea of a computer microprocessor on most of the world's desktops didn't happen until the 1990s (and many people will still say that they are not worldwide).

So, for purely economic reasons alone, this might be a realistic breakout scenario, where the biggest companies use large-scale quantum computers but they are rarely available to companies and people with fewer resources for decades. It's also quite possible that the largest companies may limit the ability of quantum crypto cracking to preauthorized companies due to national laws (such has been done with printing currencies).

Mass Proliferation

The most reasonable mid-term breakout scenario is one of mass proliferation. Already, even around a hundred, small-scale quantum computers, limited quantum computing power is being offered to small companies and individuals for free or for a time-sharing fee. It would seem unlikely that if it isn't prevented by law, the companies making and sharing their existing, limited quantum computing resources wouldn't make them available for the masses as they become even more common.

Most Likely Breakout Scenario

If history can be used as a guide, then the most likely breakout scenario will be one where the nation-states (and their supporting agencies, companies, and involved universities) will be the first entities to use large-scale quantum computing. Perhaps the largest corporations (like Google, IBM, Alibaba, or Microsoft) might get there first—although if they do beat the governments to it, the governments will probably be the primary initial customers, followed by large organizations.

Very quickly you will then see companies of all sizes using quantum computers for hundreds of different applications. Use of time-sharing quantum computers will be common at large organizations and universities. I think all of this happens within a few years of quantum supremacy. Within a decade or so, we will all be running some sort of quantum computing functions on our own devices.

Either they will be part of the devices or our devices will be linked to a quantum computing service, which handles and delivers quantum computations when our devices and problems need them.

We've been here before, on the verge of a technology jumping point. It happened with the Internet. It happened with traditional public key crypto. All of that effort first benefited the government and large organizations, and in short order, moved to the rest of the world. The government tried (and still tries) to keep strong crypto locked up away from the rest of the world. So far that strategy has not worked for long—mostly because once a particularly strong cryptographic implementation is known in theory, it becomes a practical reality soon thereafter. Trying to stop strong crypto from being used by the world is like trying to stop communication. They are often the same thing, especially in the digital world.

Summary

To summarize this chapter, although we don't know when quantum supremacy and the quantum crypto break will happen, it is the general consensus of the most knowledgeable computer scientists and the U.S. government that you should start preparing now. Chapter 5 will cover what a post-quantum world will likely look like so that you can have a comprehensive understanding as you begin your preparation.

5 What Will a Post-Quantum World Look Like?

When quantum supremacy and the forthcoming quantum crypto break happens, the world will change forever. There will be the world's history before and the world's incredible future after. Most changes will not happen instantly but instead will happen across a multitude of timelines based on different uses and applications. Some will happen within timelines measured in weeks and months and others over years and decades. But far-reaching, momentous change is coming.

This chapter covers the likely changes starting by concentrating on what applications quantum computing is likely to break in the near term (the focus of this book), followed by all the new or improved devices and applications that we will see because of quantum properties. Like most other previous significant technological advances, the changes can and will be used for both good and evil. Quantum computing will impact us in many ways, not just by breaking cryptographic secrets. After covering the breaks and improvements related directly to cryptography, this chapter will explore all the wonderful new inventions and improvements we will see beyond the crypto issues.

NOTE The word *application* in this chapter is used to denote any type of implementation of technologies and not just software programs.

Broken Applications

The primary reason for this book is all the computing applications using current technology, algorithms, protocols, and ciphers, which will be weakened or completely broken by quantum computing. This includes any application that has protection based on something that newly harnessed quantum properties can defeat. The post-quantum world will be full of weakened and utterly broken cryptography (and with defenders' help, plenty of quantum-resistant cryptography, as will be discussed in Chapter 6, "Quantum-Resistant Cryptography," and Chapter 7, "Quantum Cryptography").

As covered in the previous chapters, this includes any protection that depends on the inability of traditional binary computers to do superfast calculations (which Grover's algorithm can overcome) or factoring mathematical formulas involving large prime numbers, using Shor's algorithm.

Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto, First Edition. Roger A. Grimes.

Weakened Hashes and Symmetric Ciphers

Grover's algorithm essentially means quantum computers will be able to weaken most existing traditional symmetric ciphers and hashes, especially when they are used with smaller key sizes. Grover's algorithm on quantum computers essentially cuts the protection of most symmetric ciphers by half. 128-bit ciphers will have only 64 bits of equivalent protection, 256-bit ciphers will have only 128 bits of equivalent protection, and so on. When the quantum crypto break happens, 128 bits of symmetric key protection will still be considered strong enough that it won't fall immediately, but it does place breaks within the near-term realm of possibility. Given the traditional improvements in computer processors (i.e., Moore's law), 128-bit symmetric keys are likely to provide protection only for years, not decades.

Any symmetric cipher or hash using key sizes or hash outputs smaller than 256 bits will be of questionable long-run protection value (i.e., *quantum-susceptible*). Symmetric cryptography using larger key sizes is considered *quantum-resistant* and, on the upper end of the scale, using key sizes larger than the bare minimums is considered *quantum safe*. Most cryptographic authorities recommend that 256-bit and larger symmetric keys (e.g., quantum-safe) be used *now* to fight the threats of quantum-based attacks for the long run. Conventional thinking is that organizations needing only a moderate security requirement or needing to protect secrets for a few years only can use 192-bit symmetric keys but that organizations needing high security or protection beyond a few years should use 512-bit keys. Although 192-to-256-bit keys can be used as a "bridge" until the larger key sizes can replace them, ultimately everyone should be striving to use larger key sizes for their most critical and sensitive data that they want to keep secure for a long time. Essentially, you want to let the Mosca Inequality guide your plan; more on this in Chapter 9, "Preparing Now."

NOTE Cryptographic hashes should also use digest outputs along the same recommended lengths as are recommended for symmetric ciphers, even though they are not related cryptographic types.

It's very important to note that even widely trusted, otherwise "cryptographically strong" symmetric ciphers and hashes are considered threatened by quantum computing if they use 192-bit and smaller key sizes and digests. For example, even currently trusted and accepted SHA-2 and SHA-3 hashes are not considered quantum-resistant if they use key sizes smaller than 192 bits. This is because the primary weakness in the quantum break scenarios is the inherent protection of key size itself and not a weakness in the underlying algorithm. The key size determines the "bits of protection" provided by the cipher or hash. For example, a cipher using a 128-bit key provides 2^{128} number of bits that a cracker would have to guess (unless there was an underlying cryptographic flaw in the algorithm that weakened the total bits of protection) to be assured of finding the correct answer.

In most real-world scenarios, the average number of guesses would be half of the 128 bits (which is 2^{127} bits), because on average half the time the number of real-world guesses would be less than 2^{127} guesses and half the time it would take more than 2^{127} guesses (equaling 2^{127} guesses over a large

number of tries on average). When doing pure brute-force guessing at keys or hash digest results, all that matters is the number of bits of protection that have to be guessed at. The underlying algorithm's mathematical prowess is not a factor when doing pure guessing attacks.

NOTE The National Institute of Standards and Technology (NIST) and other organizations consider 128-bit symmetric ciphers to be "weakly" quantum-resistant. I don't know of anyone who wants to move to use safe quantum-resistant cryptography who wants "weak" protection. Thus, I don't consider 128-bit keys to be truly quantum-safe and I don't write about them that way.

Table 5.1 lists examples of various traditional hashes and ciphers that are or aren't considered resilient (i.e., susceptible or resistant) in the post-quantum (PQ) world.

Table 5.1: Weak and quantum-resistant traditional hashes and ciphers in the post-quantum world

Hashes		Symmetric Ciphers	
Quantum-susceptible	**Quantum-resistant (when using 192-bit and larger digest options)**	**Quantum-susceptible**	**Quantum-resistant (when using 192-bit and larger key sizes)**
MD-4, MD-5, SHA-1, LM, NT, SHAKE-128, RIPEMD (when used with key sizes smaller than 192), PBKDF1, PBKDF2 (when using key sizes smaller than 192), BCRYPT	SHA-2, SHAKE, SHA-3, PBKDF2, RIPEMD, Argon2, Blake2	DES, 3DES, DESX, CAST, IDEA, SAFER Kuznyechik, Serpent-128 and -192, AES-128, Twofish (less than 192 bits)	AES, Blowfish, Twofish, Serpent-256 bits, Chacha/Salsa20

NOTE Any organization interested in being quantum-resistant in the PQ world needs to begin using traditional symmetric ciphers and hashes with key sizes equal to or larger than 256 bits, with 512 bits for best assurance.

If a symmetric key is broken, an adversary can read the content it was protecting. This has happened in the past. Earlier symmetric ciphers, such as DES (with 64-bit keys but only 56 bits of protection), were considered strong enough to protect confidential information. Over time, increased computational power made DES nonprotective. Today, information protected by DES can be broken in minutes. Accordingly, the current symmetric cipher recommendation is AES with 256 bits or more of protection and 512 bits if you want to be quantum-safe for the long term. AES-192 is acceptable for quantum protection only for the very short term, maybe a few years.

If a hash is broken, it can be possible for an adversary to create other, rogue content that has the same identical hash digest (which the adversary claims is the original legitimate content). This is

known as a second preimage attack. Two different contents creating the same hash result completely invalidates the hash algorithm for any use. It has happened a few times in the past, as was most recently proven by Google researchers in 2017 with the SHA-1 hash. Google was able to create two different documents that resulted in the same identical SHA-1 hashes (see Figure 5.1). You can read more about the first successful SHA-1 hash "collision" here: `http://shattered.io/`.

Figure 5.1: Example hashes and other content document attributes revealing identical SHA-1 hashes from two different documents

NOTE Interestingly, as you can see in Figure 5.1, the supposedly weaker hash algorithms of MD-5 and CRC32 correctly show two different hash values, while the purportedly stronger SHA-1 hash does not. This is because the two different documents were specifically constructed to take advantage of a flaw in the SHA-1 hash algorithm and the researchers did not care about impacts against other algorithms. In reality, exploiting flaws in MD-5 and CRC32 is exponentially easier, and such flaws were exploited many years ago. However, making two documents that would result in all three hashes showing the same hashes for both documents would be even more difficult. If the "document" was a more complex executable instead of a simple document, it would be exponentially harder (if not more) to create a malicious executable that had the same hash as the original, nonmalicious executable. But in cryptography, if your cipher fails even a simple test, it fails completely.

NOTE CRC32 is not a cryptographic hash. Cyclic redundancy checks (CRCs) are error-detecting codes that attempt to allow two different contents to be quickly summarized and compared to see if they differ, like a true cryptographic hash. But CRCs don't have any of the required properties of a good hash, such as guaranteeing that no two different contents will ever have the same digest output. CRCs were a popular "poor man's hash" for decades but have now been replaced by legitimate cryptographic hashes in most applications.

The lesson history teaches is that weak symmetric ciphers and hashes can be used by adversaries for malicious purposes. Weak symmetric ciphers can be used to read unauthorized content, and weak hashes can be used to unfairly claim that two different contents are identical, which can then be used to fool unsuspecting users. Weakened asymmetric ciphers have been used in several high-profile attacks to compromise unsuspecting victims.

Broken Asymmetric Ciphers

As covered in previous chapters, any asymmetric cipher that relies on one of the three following mathematical problems will be considered unusable once Shor's algorithm (and other algorithms which improve on it) is run on a quantum computer with a sufficient number of stable qubits: the integer factorization problem, the discrete logarithm problem, or the elliptic-curve discrete logarithm problem. This includes the following traditional asymmetric cipher algorithms:

- Rivest, Shamir, Adleman (RSA)
- Diffie–Hellman (DH) and related primitives
- Elliptic Curve Cryptography (ECC) and related primitives
- ElGamal

Like symmetric algorithms, the key size of the asymmetric cipher does play a role in whether it is considered weak or broken. Running Shor's algorithm to factor any prime number equation requires $(2 \times n) + 3$ stable qubits, where n is the number of asymmetric key bits to crack. Thus, to crack a 2,048-bit RSA key, you need 4,099 stable qubits and to crack a 4,096-bit RSA key, you need 8,195 stable qubits. Theoretically, you could keep incrementally increasing your asymmetric key sizes in order to keep ahead of the number of qubits that quantum computers are gaining. Yet most quantum experts think it's a far better strategy to move to a quantum-resistant asymmetric cipher so that your defense does not depend on staying ahead of another figure you have no control over.

NOTE Quantum-susceptible key exchanges are also weak or broken. Accordingly, most traditional key exchanges such as Diffie–Hellman (DH) and Elliptic Curve Diffie-Hellman (ECDH) are as well.

Weakened and Broken Random Number Generators

Computer security frequently relies on randomly generated numbers for much of their operations and security (this is explained in more detail in Chapter 7). Because of this dependence, most computers have built-in, hardware-level random number generators (RNGs), as do most operating systems and many applications at the software level. Unfortunately, it is impossible for a nonquantum computer to be truly random about anything, much less able to generate truly random numbers. And even if traditional computers could, they couldn't provide proof that any particular generated number was truly randomly selected. Instead, nonquantum computer RNGs try their best to approximate true randomness (something called *pseudo-randomness*), which to the average person and application appears to be perfectly random, even if it's not. The problem is that any number that is required to be truly randomly generated creates a potential vulnerability when the number is not. This is a problem that has flummoxed the computer security industry since the early days of computers.

Many if not most RNGs over decades have been found to contain one or more vulnerabilities, vulnerabilities that were found using traditional methods and in *standard computational time* (i.e., exponential or logarithmic solution speed-ups are not required to arrive at solutions in a reasonable time frame). The history of failed computer security solutions is replete with examples of flawed random number generators. Basically, if adversaries can find how the pseudo-random number generator is flawed (it will always be a repeatable pattern, which can be used to predict future generated numbers), they can use it to weaken or break the higher-level cipher or application.

Because of this, the most popular and dependent RNGs have steadily improved their pseudo-randomness over time. The really bad RNGs stopped being used and the existing ones worked harder to have less obvious flaws. Today, many non-quantum RNGs *appear* to be nearly truly random, even if they are not. Finding the flaws and predictable patterns is nontrivial. Still, many cryptographic researchers focus on finding RNG flaws. Looking for nonrandom, repeatable patterns is a bit like trying to factor large prime number equations or crack symmetric keys. The more computing power you have, the easier it will be to find RNG flaws.

Quantum properties and algorithms (such as superposition and Grover's algorithm) will improve the chances that a flaw in a traditional RNG will be found faster. There is even the strong chance that quantum computers will be able to help find every predictable, repeatable classical computer RNG pattern, revealing all their true flaws. Suffice to say, quantum computing may be able to weaken, if not break, traditional RNGs for good, and everything that relies on them. The best solution to weakened and broken traditional random number generators is to use quantum-based random number generators (covered in Chapter 7).

Weakened or Broken Dependent Applications

Obviously, any application relying on a quantum-susceptible hash, cipher, or RNG is also considered quantum-susceptible. Today, there are far more vulnerable computer security applications than nonvulnerable ones. Here are some common examples of such applications.

TLS

Just breaking the cryptography involved with Transport Layer Security (TLS), which much of the Internet relies on, shows how big a threat the quantum computing break is. TLS relies on quantum-susceptible public key infrastructure (PKI), digital certificates, digital signatures, asymmetric ciphers, symmetric keys, and hashes. TLS uses asymmetric ciphers and digital certificates to allow computer hosts and users to authenticate themselves to others. It also allows communicating participants to securely generate shared session symmetric keys so that they can encrypt traffic (using the independently generated, shared session symmetric keys) between authorized parties.

In 2019, over 70 percent of Internet websites are using HTTPS relying on TLS (`https://etherealmind.com/percentage-of-https-tls-encrypted-traffic-on-the-internet/`). TLS is also being adopted at a dizzying rate in nearly every virtual private networking (VPN) implementation as part of the VPN's security. For decades, the majority of VPNs made up their own proprietary

security algorithms and methods, but now most of them, including the largest and most popular VPNs (Cisco, Palo Alto, Microsoft, etc.), all rely on TLS for at least part of their base security.

There are several different TLS versions (version 1.3 is the most current version as of this writing), and although TLS can be updated to use quantum-resistant cryptography, nearly every current implementation uses a quantum-susceptible version. Once the world is told to upgrade their TLS implementation to a quantum-resistant form, if history is any guide it will take many years for most of the Internet to be upgraded. TLS has suffered many past critical vulnerabilities, and it usually takes three to five years for the majority of TLS implementers to move over to the less flawed versions. Let's hope it doesn't take that long when the quantum break happens. Or let's hope, even better, that implementers are able to move to quantum-resistant versions ahead of time.

PKI and Digital Certificate Applications

Like TLS, the number of PKI and digital certificate–consuming applications have exploded in popularity over the last decade. PKI has been around and in popular use for decades, although usually implemented in a way that is behind the scenes. Most end users don't realize how much they rely on PKI every day. The last 10 years have seen a rapid rise in the number of internal applications at organizations that use PKI. It's almost impossible to find an organization that doesn't rely on PKI for its daily business-critical functions.

Once quantum computers are able to factor public-private key pairs of 4,096 bits or less, most of the world's current PKI implementations will have been utterly broken—from the root certification authority (CA) to every digital certificate the CA and its subordinate dependent CAs have ever issued. Most currently issued digital certificates use only 2,048 bits of protection, and a substantial percentage of all digital certificates still use only 1,024 bits. A 4,096-bit crypto break will get them all, although the 1,024- and 2,048-bit certs will fall first.

PKI CAs and many PKI-enabled applications can be updated to use less susceptible forms of cryptography. But like TLS, almost every one of these applications is currently using quantum-susceptible forms, and moving them to quantum-resistant forms will likely take years. These PKI-enabled applications include the following:

- Host identity and authentication solutions using digital certificates
- Password and authentication solutions using public key cryptography
- TLS-secured versions of Secure Copy Protocol (SCP), Post Office Protocol (POP3), Network News Transfer Protocol (NNTP), Simple Mail Transfer Protocol (SMTP), Internet Message Access Protocol (IMAP), File Transfer Protocol (FTP), Telnet, Hypertext Transfer Protocol (HTTP), Secure Internet Live Conference Protocol (SILC), etc.
- Smartcards/virtual smartcards
- Multifactor authentication (MFA) that uses asymmetric ciphers
- Secure Shell (SSH)
- Pretty Good Privacy (PGP)

- Secure/Multipurpose Internet Mail Extensions (S/MIME)
- Unified Extensible Firmware Interface (UEFI), which is a computing device boot protocol used by most computers
- Domain Name System Security Extensions (DNSSEC), used to secure DNS transactions
- DomainKeys Identified Mail (DKIM), used to help prevent email domain spoofing
- Hardware security modules (HSMs)
- 802.1X port security (when using digital certificates)
- Paillier crypto systems (and its historical antecedents)
- YAK (a public key authenticated key agreement protocol)
- Vehicular computer systems that use asymmetric ciphers (most do)
- Nearly any other application using PKI and digital certificates

Suffice to say, quantum computing will likely break most of what the Internet relies on for its security. It's easier to find what isn't likely to be broken than what is.

Digital Signatures

Digital signing authenticates content (documents, programs, data, identities, etc.) using PKI, asymmetric ciphers, and hashing. All current popular implementations, including Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (EDSA), use quantum-susceptible cryptography. One of the biggest uses for digital signatures is for signing content and downloads from open-source and commercial vendors. The quantum crypto break could allow adversaries to generate identical public-private key pairs, and then allow them to sign newly modified or purely malicious content and send to unsuspecting consumers.

Historical Real-World Asymmetric Attack An attack happened in 2012 with public cryptographic digital signing certificates that had become significantly weakened over time. An advanced malware program known as Flame (`https://en.wikipedia.org/wiki/Flame_(malware)`) created a faked Microsoft digital signing certificate (see Figure 5.2) to sign the malware program. The forging succeeded because of a few weaknesses, including these:

- The involved asymmetric key was only 512 bits long.
- It was hashed by the vulnerable MD-5 hash.
- The parent CA allowed child certificates to include the purpose of digital signing, even though there was no reason for that specific purpose to be allowed for any certificate issued by the CA. This allowed attackers to create new digital signature signing certificates once the original asymmetric key pair was cracked.

All of these cryptographic weaknesses allowed a malicious adversary to re-create the legitimate public-private key pair used by a real-world Microsoft CA to sign additional (rogue) digital certificates. The adversary created a new rogue digital signature certificate and then used it to sign their

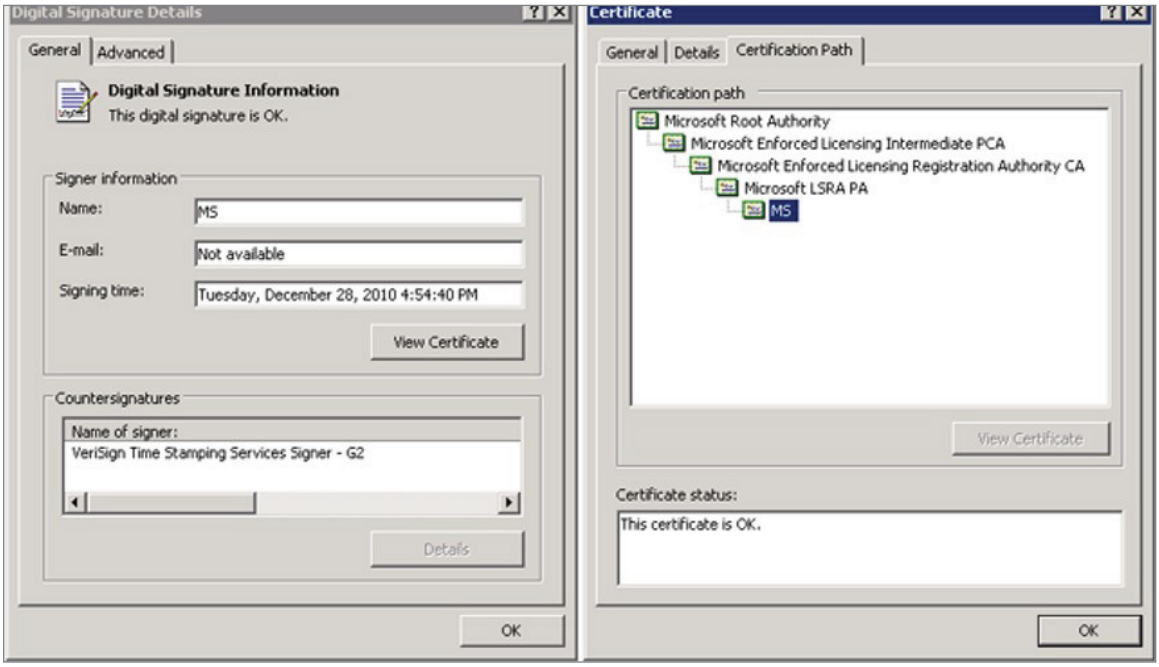

Figure 5.2: Forged digital certificate purporting to be from Microsoft used to sign malware

malware. The malware could then be sent to possible victims, who could easily be tricked into believing the involved program was a legitimate program from Microsoft. In some cases, the digital signature would allow the malware to be installed without the user having to consent to the install.

This was the first known instance of a popular vendor's CA public-private key pair being cracked by an adversary to allow malicious signing of unauthorized content. Malware creators had previously stolen public-private key pairs from legitimate vendors and then used the stolen certificate to sign their malware creations (as was done in the Stuxnet malware program), but this was the first time that cryptographic cracking was done to create a brand-new (rogue) digital signing certificate. See `https://arstechnica.com/information-technology/2012/06/flame-malware-hijacks-windows-update-to-propogate/` for an excellent discussion of the Flame malware program, and visit `https://blogs.technet.microsoft.com/msrc/2012/06/03/microsoft-releases-security-advisory-2718704/` if you want to read Microsoft's official warning about the issue.

Microsoft responded by revoking the rogue digital certificate (in many different ways) and did a complete review of every Microsoft digital certificate ever publicly issued that was not already expired or revoked. Several other weak digital certificates were found, removed, and revoked before they could be used maliciously. Microsoft also updated Microsoft Windows and other related software so that they would no longer accept any digital certificates with asymmetric keys below 1,024 bits. The lesson is that weakened or broken asymmetric ciphers can easily be used maliciously by an adversary.

Wi-Fi Network Security

Most Wi-Fi wireless networks are protected using a wireless security protocol known as Wi-Fi Protected Access (WPA). There are currently three versions: WPA, WPA2, and WPA3. Most Wi-Fi networks currently use WPA2; the WPA3 version, used for the first time in 2018, is still not deployed widely.

In many enterprise scenarios, WPA2 wireless security uses digital certificates, 802.1X port security, and symmetric encryption. In most implementations (home or enterprise), even if asymmetric ciphers are not used, symmetric ciphers are. And in most instances, the symmetric cipher is AES with a key size of 128 bits. Some newer implementations (using WPA3) use 192-bit symmetric cipher keys, still far below the 256-bit minimum symmetric key size recommended to be long-term quantum-resistant.

Many Wi-Fi routers use a pre-shared key (PSK) as the initialization value to allow new nodes to join the network. This is the "Wi-Fi password" that most people give to guests so they can join their Wi-Fi network. Today, PSKs should be randomly generated and should be at least 16 to 20 characters or longer, although in practice few Wi-Fi implementations follow that security advice. Traditional Wi-Fi network cracking tools often guess at the PSK and attempt to join and rejoin the network over and over until they hit the right PSK.

NOTE Most Wi-Fi wireless hubs allow a PSK to be up to 63 characters.

After a client supplies the correct PSK (or 802.1X digital certificate), WPA2 creates a shared session secret key called the pairwise master key (PMK) and hashes it using the PBKDF2-SHA1 hashing algorithm with quantum-susceptible key sizes. Quantum computer speed-ups could allow a quantum computer to more quickly crack the hash and guess the PMK.

So, whether unauthorized access is gained by attacking the PSK, PMK, hashes, or symmetric/asymmetric algorithms and keys, today's traditional Wi-Fi networks afford quantum computers many opportunities to obtain network access or to eavesdrop on otherwise cryptographically protected communications.

NOTE Motivated adversaries may already be recording currently protected Wi-Fi network traffic of their opponents, waiting for the day when they can decrypt that traffic using quantum computing. The threat of the quantum crypto break against wireless networks and other types of eavesdropping is already a risk.

Microsoft Windows

Like most of today's popular operating systems, Microsoft Windows incorporates a plethora of quantum-susceptible cryptography, including hashes, symmetric ciphers, asymmetric ciphers, and RNGs. Windows' main authentication protocols (Kerberos and NT [New Technology] LAN Manager [NTLM]) are quantum-susceptible. Both use the NT hash, which uses a 128-bit MD-5 hash value. NT hashes

are not only used in network and local logon authentication scenarios but also for password hash storage, both locally and on Active Directory domain controllers. Locally cached passwords use the PBKDF2 hash, which is more resistant but still susceptible.

Microsoft's new authentication protocol, Windows Hello for Business, uses hardware or software public key encryption and/or digital certificates behind the scenes to support allowed authentication mechanisms. Windows 10 (and later) supports the FIDO (Fast ID Online) 2.0 standard (`https://fidoalliance.org/fido2/`), which is based on digital certificates and public key ciphers.

Microsoft does use SHA-2 (128-bit) for hashing, although many files are also (or only) hashed by SHA-1 for backward compatibility purposes. Windows currently uses AES 128-bit ciphers for symmetric encryption and 2,048-bit RSA keys by default for asymmetric encryption, although larger key sizes are supported and can easily be enabled.

Most Microsoft applications, including its PKI flagship product, Active Directory Certificate Services (ADCS), use quantum-susceptible ciphers by default. Microsoft has already successfully tested ADCS using quantum-resistant ciphers to ensure ADCS can use them when needed.

Because Windows does not include quantum-resistant asymmetric ciphers by default, any Microsoft application using an asymmetric cipher can also be considered quantum-susceptible. Any Microsoft feature or application using smaller symmetric keys and hashes (especially if not using larger key sizes, such as 192 bits or larger) is also susceptible over the long-term. As covered in Chapter 2, "Introduction to Quantum Computers," Microsoft is a main quantum researcher and is already heavily researching and investing in learning how to move all of their products to quantum-resistant forms when needed, more so than any other popular operating system vendor. Pay attention to what Microsoft says about its cipher recommendations. When Microsoft says it's time to move, move!

Cryptocurrencies

A common question in quantum circles is whether cryptocurrencies are quantum-susceptible. Yes, most of them, including bitcoin and all the most popular implementations, are to some extent. Some of the underlying cryptography involved is at the very least somewhat quantum-susceptible, with many of the most critical components absolutely being quantum-susceptible. Most cryptocurrencies involve at least four major areas of concern: the blockchain, the individual user's public-private keys, the security of the network involved, and the individual user's cryptocurrency wallet.

Blockchain Susceptibility Let's start with the blockchain, which is probably the least quantum-susceptible of all the involved components. The *blockchain* is a distributed, decentralized ledger (i.e., records database) for tracking and verifying individual transactions. Each individual tracked transaction may be stored in a separate transaction "block," or multiple transactions may be stored together within a single block. The number of transactions stored per block depends on the implementation. An individual block contains the transaction information (it can be any information as defined by the application, including just a hash of the required transaction information), and at least one cryptographic hash, along with any other required information.

A common blockchain block format is represented by Figure 5.3. The "chain" of the blockchain refers to the fact that the hash of the previous block is stored in the next block, which is then hashed and stored in the next block, and so on. This makes each subsequent block "hooked" by hashing to the previous block in such a way that all blocks in the blockchain are cryptographically linked to each other. You cannot easily tamper with any block without also modifying every subsequent block (because the hash of the tampered block would change). It's a pretty strong protection—as long as you can protect the hashes.

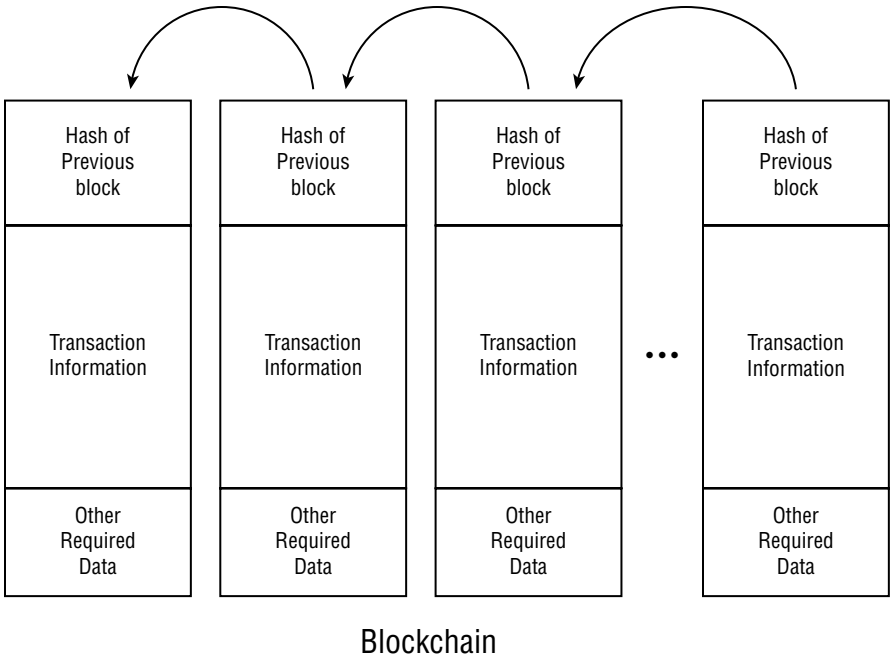

Figure 5.3: Format of a block in the block chain

In most blockchains, the hash is quantum-resistant and uses 256 bits of hash digest protection. Although 512 bits would be quantum safer and better in the long term, at least most of them are not 128 bit. There are also additional inherent projections. First, as covered earlier, in order to maliciously manipulate any single block in the blockchain, you would need to modify the information and hashes of all subsequent blocks and do so in a way that would not be detected and recovered by all (or at least half) the underlying participants. That's a very strong, underlying inherent protection. That's why blockchains are becoming so popular for transactions needing long-term integrity protection.

NOTE So-called "51% Attacks" have been accomplished in the real world against less popular cryptocurrencies. See `https://www.ccn.com/ethereum-classic-51-attack-blockchain-security-researchers-reveal-full-implications/` as an example.

Second, even though the hash digest may be only 256 bits long, it is often used multiple times at once and/or in conjunction with additional hashes. For example, bitcoin uses SHA-256 and RIPEMD-160. RIPEMD-160 individually is certainly considered weakly quantum-susceptible, but when combined with SHA-256, and especially with multiple rounds of SHA-256, it becomes less so.

Other Cryptocurrency Susceptibilities Where cryptocurrencies become more quantum-susceptible is where and when they use quantum-susceptible public ciphers. The Internet connections between individual participants and the cryptocurrency blockchain are protected by TLS, and each individual user uses quantum-susceptible public key cryptography to modify the blockchain and to protect their individual wallets. Anyone learning a user's public-private key pair could steal from the user's wallet or maliciously manipulate a user's transaction en route to the blockchain.

Individual users' wallets have been badly hacked over and over since bitcoin made a bunch of people instant millionaires. Hundreds of millions of dollars, and likely billions of dollars, have been stolen, even before the quantum crypto break was considered part of the potential risk picture. Blockchains and cryptocurrencies are under attack not only by individual hackers and groups but also by nation-states. Some rogue nations such as North Korea are known for funding their countries by stealing hundreds of millions of dollars in cryptocurrencies.

There are two big saving graces related to the quantum hacking of cryptocurrencies. First, once the quantum break happens, the entire world's monetary system will also be under attack (most of it is protected by TLS and other quantum-susceptible ciphers), so cryptocurrencies will be just one of our many concerns. Second, most cryptocurrencies can "fork" (i.e., split) their implementations to a more quantum-resistant set of requirements. This approach creates its own issues, but it has been done many times in the past for other security problems and issues.

There are also a few already existing quantum-resistant cryptocurrencies, but they are in the minority. Most existing cryptocurrency guiding bodies feel that the cryptographic overhead for switching to quantum-resistant ciphers ahead of time is not worth a premature move. Most of the popular cryptocurrencies plan to migrate to less quantum-susceptible cryptography when they hear news of the imminent quantum cryptographic break.

If you are interested in reading more on cryptocurrencies and their quantum susceptibility, there are many great resources, including

`https://en.bitcoin.it/wiki/Quantum_computing_and_Bitcoin`

`www.youtube.com/watch?v=Uy5zHAwo43o`

`http://diyhpl.us/~bryan/papers2/bitcoin/On%20Bitcoin%20security%20in%20the%20presence%20of%20broken%20crypto%20primitives%20-%202016.pdf`

Bluetooth and NFC

Bluetooth is a very common short-distance (usually 15 feet or less) wireless standard, often used between two devices to transmit information and to connect wireless headphones and speakers using UHF radio frequencies. Near field communication (NFC) is used to transmit information over very short distances, usually measured in a few inches. NFC is often used for wireless payment systems, contactless authentication, and transmission of information between two devices, like cell phones.

Both Bluetooth and most NFC protocols are based on weakly quantum-susceptible ciphers. For example, there are various levels and modes of Bluetooth security depending on the version of the protocol being used. But even the best and highest levels of security are quantum-susceptible. Bluetooth security level 2 supports AES 128-bit key. Level 4, the highest security level, supports Elliptic Curve Diffie–Hellman P-256. Sadly, most Bluetooth users have no idea what versions or security features are used in their Bluetooth-using products. For more information on Bluetooth security, read https://en.wikipedia.org/wiki/Bluetooth#Security_concerns and https://duo.com/decipher/understanding-bluetooth-security.

Security feature-wise, NFC is far worse. Its creators mostly assumed the short distances it was created to be used over would be the defining security protection. And it is. But as with any wireless technology, hackers will likely learn how to interface with its wireless transactions many orders of magnitude further than the creator's intention. Most NFC implementations have no security beyond that built into the applications using NFC as a wireless transmission means. Those that do have some transmission security often use ciphers on the weak end of the scale such as AES-128 or quantum-susceptible implementations of public key cryptography. In summary, NFC is quantum-susceptible.

NOTE Radio-frequency identification (RFID) is a type of NFC. Although RFID is often used by credit cards for wireless transactions, it has no built-in transmission security. Anyone with a reader who can get within the appropriate distance can read what is transmitted between two nodes. You can find RFID eavesdropping videos all over the Internet. With that said, however, the risk of RFID crime is very low and getting lower all the time. For more information read the author's article on the subject: www.csoonline.com/article/3243089/cyber-attacks-espionage/the-truth-about-rfid-credit-card-fraud.html.

IoT and Hardware Devices

Most existing Internet of Things (IoT) and other computing hardware devices (such as phones, televisions, cameras, and appliances) contain quantum-susceptible forms of cryptography. What makes many IoT and other mainstream hardware devices a higher risk than average is that most consumers don't know what security they use and their devices are often hard to upgrade. So, once the quantum crypto break happens, there will likely be billions of quantum-susceptible IoT and hardware devices, many of which will be in our homes.

Many hardware devices (e.g., Blu-ray players, stereos, and speakers) contain no way for their owners to upgrade. They likely contain security flaws today, and every future vulnerability that they contain will never be fixed. Some devices contain built-in ways to update, but for a multitude of reasons many device owners will never update them. Many owners are unaware that the devices they have may need security updating. Most consumers never go into the device's management console after the initial install to check whether patches need applying. Their devices are waiting for their

owners to check for and apply new patches, but their owners will never see the prompts. A large percentage of consumers are aware that the devices they buy may need security updating in the future, but they do not care about the risks enough to update them. Such is a sad fact of life.

NOTE The percentage of devices containing upgrade capabilities that will be updated by their owners varies significantly by type of device. Computing devices and phones are on the high end of upgrade percentages (likely 70 percent and higher). Almost every other type of device has tremendously smaller upgrade compliance. Wi-Fi routers and Internet-connected security cameras have compliance rates often at less than 10 percent. Some models are around 1 percent patched. The world would be a safer place if every hardware device routinely updated itself without owner interaction.

Beware of Easy Promises

Be wary of claims from organizations that tell you updating their current quantum-susceptible standard and applications will not be difficult when needed. I frequently read these types of claims in cryptocurrency forums and from IoT vendors. Undoubtedly most of the participants in these organizations making these statements have never been involved a massive update endeavor. They are speaking from a place of inexperience. They think the upgrading process will be as simple as offering updated code and having the masses apply it. They don't truly understand the real-world challenges, such as the following:

- How much of their user base will not even hear of the need to update
- How much of their user base is using old, unsupported forms of their product
- How their "heavily tested" patch will not apply correctly to some percentage of impacted devices even when users try to do the right thing
- How many owners of their products cannot simply apply an update right away even if it is needed
- How difficult the upgrades will be and the power and challenges of the human psychology involved

No one involved in a previous massive update endeavor would ever claim that the next one is going to be smooth and orderly. Beware of anyone claiming that the upgrade process to quantum-resistant software will be easy. That statement alone is enough reason to disregard their supposed "expertise." Anyone who has been through a massive upgrade project is humbled by the experience and lowers their grand expectations. Upgrading is always harder than we imagine.

Much of our computing world and most of our intelligent devices contain quantum-susceptible cryptography. More computing devices and services are susceptible to future quantum attacks than those that are quantum-resistant. Many of them can be upgraded to quantum-resistant algorithms when the time comes. Others will remain forever quantum-susceptible. Chapter 9 will show you how you and your organization should prepare and plan for this eventuality.

Quantum Computing

I do not want this book to be all gloom and doom. Even though the focus on this book is on the threats to our computer security enabled by quantum computing, quantum computing will give us far more positive things than we can currently imagine or what we covered in Chapter 2. Here are some more detailed predictions.

Quantum Computers

We already have many dozens, if not over a hundred, quantum computers (as of 2019). That number is steadily headed north, even before quantum supremacy. Once quantum supremacy has been reached, the number of quantum computers will explode exponentially. Every big company that was sitting on the fence waiting to see if quantum supremacy would really happen will buy in. No big company wants to have its interests surpassed by its competitors with better computing power. No one wants to have the slower, "older" technology computers. Even companies and vendors who do not truly understand what quantum is and what the benefits are will want it. It will be a "buzzword" that drives marketing just like cloud computing and artificial intelligence did before it.

As covered in Chapter 2, there are over a dozen major types of quantum computers. Expect the number of types to decrease over time as the industry settles into the types with the most efficient benefits. Back in the early days of binary personal computers (PCs), there were dozens of distinct PC vendors (among them Apple, IBM, Altair, Micral, Wang, Tandy, Sinclair, Nippon Electric Company [NEC], Digital Equipment Corporation [DEC], Commodore, and Sun), many of which contained personalized vendor chips. Eventually, Apple and IBM-style and Linux PCs became the dominant models (Sun Microsystems was a major player for decades as well). The same consolidation is likely to happen to quantum computers as they mature.

Whatever types and vendors win out, quantum computers are likely to get smaller and cheaper over time. The average PC in the 1980s cost several thousands of dollars, and that was for a monochrome screen, two floppy disks, and a very small hard drive (10 to 20 megabytes), with less than 1 megabyte of RAM. They often weighed 15 to 30 pounds and took up most of the room of the desktop they were placed on. Early external hard drives weighed over a hundred pounds and were the size of a filing cabinet. Today, you can find dozens of computers weighing less than 2 pounds for a few hundred bucks with performance parameters that would have been equated to supercomputers back in the 1980s.

The same type of physical and performance consolidation is likely to happen in the quantum computing field. Humans excel at making things tinier. Expect quantum computers to get faster, cheaper, and in smaller form factors. One of the biggest form-factor limitations is the requirement for most quantum computers to be cooled to near 0 degrees Kelvin. Expect even super-cooled quantum computers to get smaller, although there are at least a few quantum computer designs (such as trapped ion) that do not need those extremely low temperatures. Perhaps one of those models will win out, allowing the form factor of quantum computers to shrink significantly right away.

It remains to be seen if quantum computers can be shrunk down to the form factors we are used to today (e.g., desktop computers, laptops, pad devices, and smartphones). But we've decreased the size and expense of nearly every other computing device, while at the same time significantly decreasing the cost. Why should quantum computing be any different?

Quantum Processors

We already have dozens of different types of quantum processors. The types, availability, and affordability will increase over time. Many prediction models point to the idea of a quantum coprocessor. Back in the early days of PCs, most computers could be upgraded by adding a separate math "coprocessor" chip to a slot on the motherboard. The math coprocessor was specifically created to do complex math faster than a PC's regular processor could. Programs requiring the use of floating-point, complex math could "offload" that math to the coprocessor, which would perform the needed calculations and then hand off the result to the main processor so that the program could complete faster than without a math coprocessor.

Performance tests routinely showed that computers using math coprocessors greatly outperformed computers without them. Pretty soon consumers insisted that any computer they bought have the "optional" math coprocessor installed. From a marketing perspective, it began to be not so optional. Demand for math coprocessors was so great that eventually the main PC processor manufacturers just added the advanced math routines into the regular processors. Around the time Intel introduced its 486 line of processors, the idea of needing a separate math coprocessor died a natural death. Today, everyone who buys a computer gets one with built-in advanced math components.

Many quantum computer scientists expect the same thing to happen with quantum computers. In the future, it is likely that the majority of computer applications will not require quantum calculations to do all of their work. Quantum experts expect that, for a time, quantum coprocessors will become a thing. Our computers will do normal (e.g., binary) computing as is needed for the majority of their functioning and offload the complex quantum calculations to a quantum coprocessor. The quantum coprocessor will take input from the computer's main processor, perform its quantum magic, and then decohere the result and hand it off to the computer's main processor. Then over time, the concept of a separate quantum coprocessor might fade into history just like the math coprocessors of yesterday. Many quantum experts see a day when we have a quantum computer on every desktop and in every device we own, although that is likely a decade or many decades off into the future.

Quantum Clouds

There are already nearly a dozen quantum-based clouds in existence, some of which are available for (free) public use. Expect the number of free and commercial quantum clouds to exponentially multiply when quantum supremacy happens. Quantum clouds are likely to be used as "virtual quantum coprocessors," where any heavy quantum computing is offloaded from the main computer's processor,

and the quantum answer returned when completed. This model makes a lot of sense, especially early on when quantum computers are expensive and require significantly environmental controls (i.e., near 0 degrees Kelvin).

Perhaps the mid-term model for quantum computing is the majority of traditional computers to be linked (programmatically) to cloud-based quantum computers, making quantum computing more cost-efficient for those who cannot afford their own quantum computers outright. Even organizations that need lots of quantum computers can adjunct additional quantum computing resources on the fly, as needed, when they don't have enough of their own individual quantum resources. Either way, both quantum computers and clouds will be with us and available to everyone for a long time to come.

Quantum Cryptography Will Be Used

The widescale implementation of quantum computers will bring many new quantum-resistant cryptographic algorithms as well as quantum-based cryptographic algorithms. Organizations will move to quantum-resistant and quantum-based algorithms over the next half decade. Quantum-resistant algorithms will be covered in detail in Chapter 6 and quantum-based ciphers and devices will be covered in detail in Chapters 7 and 8.

Quantum Perfect Privacy

One special quantum crypto gain is how quantum cryptography will better allow the creation of more fully homomorphic encryption (FHE) systems (`https://en.wikipedia.org/wiki/Homomorphic_encryption`), which promise perfect privacy. FHE is the idea that an organization can send encrypted content to a third party and allow the third-party systems to purposefully manipulate the encrypted content in some authorized and intended manner, without the encrypted text being decrypted by the third party.

As a simple example, suppose a company wanted to send a large set of sales lead records to a "cleanup" clearinghouse to find and remove duplicates and other types of invalid records. This is something many organizations with tens of thousands of sales leads do on a regular basis. Today, even though the original organization may have a security requirement that all records be encrypted, they may have to decrypt them (or share the decryption key) at some point so that the processor can search the data and remove the appropriate records.

In a perfect privacy system, the records could remain encrypted and still be successfully processed by the lead cleanup processor without revealing the plaintext content or sharing the original cipher keys. Another good future example use of homomorphic cryptosystems is allowing medical data to be shared, potentially globally with anyone doing research (e.g., Google is crowdsourcing medical information in order to solve existing challenging diseases), without divulging any personal data information to the researchers.

Most homomorphic cryptosystems involve an additional *evaluation algorithm* that is cryptographically linked to the original cipher, which can be used by the third-party processor to do their work. Homomorphic cryptosystems would allow the necessary transactions to occur without revealing private information. This would better protect the original host company, the processor company, and customers from unauthorized data leaks.

Homomorphic cryptosystems have been postulated and created since the invention of public key crypto back in the 1970s with varying results. Most of the attempts have resulted in halfway solutions, which could not be used in all situations and are known as *partially homomorphic*. There have been a dozen or so FHE systems proposed in the pre-quantum world, but they have been more about the theory than implementation. Quantum computing and especially the quantum property of entanglement promises to allow more, better, and practically implemented solutions. Quantum entanglement is the key component that FHE was waiting for. There is a subset of quantum cryptographers around the world who focus exclusively on this problem, using what they call *quantum homomorphic encryption* (or QHE). The net result of QHE is likely to be fewer data compromises. It's hard to have a malicious data leak happen on your watch if the data is never decrypted.

Quantum Networking Arrives

Right now, the quantum networking industry is nascent. Vendors are testing and successfully producing first-generation devices. Quantum networking has great potential because of its ability to be used across great distances with strong privacy. Quantum properties make it more difficult for an unauthorized actor to eavesdrop on a protected network communications stream without alerting the authorized involved parties. Expect to see quantum networking used in high-security networks followed by more general use in normal privacy requirement environments. Quantum networking is discussed in detail in Chapter 8, "Quantum Networking."

The wide availability of quantum computing and cryptography is going to threaten or break existing quantum-susceptible cryptography and usher in a new era of quantum protections that provide better security.

Quantum Applications

Beyond quantum cryptography, quantum computers are getting ready to generate entirely new industries and radically transform existing technologies. Applications that can benefit from using quantum properties and algorithms, such as Grover's algorithm, entanglement, and superposition, will be enhanced by quantum computing. There are thousands of computer problems that cannot be answered or optimized because classical computers do not have the speed or capability to answer them. Here are some of those applications that will be improved by quantum computing.

Better Chemicals and Medicines

High on the list for quantum advancement is the improvement of chemicals and medicines. It is already one of the top reasons why quantum computers are being developed. We know that atoms are made up of electrons, protons, and neutrons. And protons and neutrons are made up of other elemental particles, called quarks. All work and react on the quantum level. Individual elemental atoms often chemically bond with other atoms of the same type and different types to form larger molecules. For example, two hydrogen atoms combine with one oxygen atom to create H_2O, or water.

Nearly all of the matter that we interact with every second of every day is made up of molecules. Oftentimes those molecules are made up of hundreds to hundreds of billions of atoms and chemical bonds. Each bond can interact with other atoms and molecules in myriad ways. Understanding and predicting how atoms and molecules interact is the cornerstone of chemical and medicine research. The better chemistry and medicine can predict molecular reactions, the better resulting chemical and medicinal compounds can be.

Traditional classical computers can properly track two handfuls of molecular bonds before they begin losing information and predictive capability. Since most molecules that make our lives better consist of more bonds than that, it means that our current understanding and ability to make better chemicals and medicines is limited. Quantum computing will allow us not only to track, understand, and predict significantly more molecular interactions, but to do so at the quantum level (far easier than classical computers using quantum simulation), across longer timescales. This means better chemicals and medicine. It is likely that quantum computers will allow us to have better medicines that have fewer side effects and are customized to work with our particular medical or DNA structure (i.e., biomedicals).

Quantum effects are already used in some of the best diagnostic healthcare equipment, such as MRIs, but it will likely allow us to diagnose bad genetic markers earlier and better identify diseases. There is a greater likelihood that we will be able to better figure out how human memory and consciousness works, how it fails, and how to mitigate the associated weaknesses and illnesses. Applied therapies, such as radiation treatments, can be more focused and of less duration. The bad side effects of drug interactions can be better predicted. All in all, quantum computing will likely lead to significantly improved chemicals and medicines that will benefit humankind. If you've heard that tired diatribe of "Better living through chemistry," it certainly applies to quantum computing.

Better Batteries

The science of energy stored on batteries has not had a cosmic shift in decades. The battery life of our laptops and cell phones still progressively worsens over time. What time improvements we get come more from device energy-use improvements than from the batteries. Batteries never have enough charge to account for our natural use, they cause fires, and they contain dangerous chemicals. In electric cars, the battery is the heaviest and most expensive component of the car and lessens the overall carbon-benefit reason for having an electric car in the first place. Hundreds of companies are working to make better batteries—longer-lasting and lighter.

To that goal, many car companies and battery manufacturers are already using today's nascent quantum computers to better understand batteries at the molecular level, in particular how lithium-hydrogen and carbon molecular chains work and deplete (see an example article here: `https://insideevs.com/news/338440/volkswagen-turns-to-quantum-computing-for-electric-car-batteries/`). Other researchers are using quantum entanglement to make faster-charging batteries (`www.extremetech.com/extreme/211580-quantum-batteries-could-allow-for-super-fast-charging-thanks-to-entanglement`). It is likely that quantum computing will be central to us finding new chemical molecule interactions that will store more energy in smaller spaces, which benefits all battery applications.

True Artificial Intelligence

One of the most overused phrases in the computer world is artificial intelligence (and the related field of machine learning). The concept of AI is that computers can be programmed in such a way that they can become self-learning, much like a human being is, and with that capability and their inherent ability to do things super-fast, solve problems that human beings are not capable of. AI is the holy grail of computers. There is much debate over whether human beings' ultra-complex thought processes could ever be computer simulated.

Today, we are not close to true AI, although that does not stop thousands of computer vendors from claiming they have achieved some level of it. Quantum computing, which more accurately predicts all matter at a quantum level, may be able to give us a closer approximation to true AI. Better AI is supposed to improve our world in as many ways as quantum (without AI) will do. True artificial intelligence, where a computer supposedly can think with all the complexity of the human brain, is supposed to be a game changer. Purportedly, nearly everything we do will be improved and optimized.

One commonly cited example is of autonomous driving vehicles. Cars can already drive themselves. We are probably less than 10 years away from there being more autonomous vehicles than human-driven cars. Our kids and certainly our grandkids will not have driver's licenses or own cars. Society will simply let people rent vehicles when they need them to go to work, run errands, and take trips.

Quantum computing is likely to help make traffic management involving autonomous cars better. Quantum computers can consume all the involved autonomous cars, their positions, and speed, and figure out how to modify their speed and turns to maximize the speed and direction of all cars. The goal is to make it so that no autonomous vehicle ever needs to stop at an intersection. They will just modify their speed and paths so that they can just keep driving through the intersections and not have to stop at stop signs or traffic lights, which will become a thing of the past. This will save energy, decrease pollution, and save time for everyone involved.

Cybersecurity will also be greatly impacted by AI. We are already seeing early implementations of machine learning used to shift attackers past defenses or stop bad actors when they shift attacks. The future of computer security will likely be driven by autonomous AI-learning security and attack bots that attack and defend based on advanced algorithms. I'm not sure if we'll ever see the sophistication of the *Terminator* movies' Skynet becoming self-aware and attacking its human creators, but

many brilliant people are worried about that scenario, including Elon Musk (www.theguardian.com/technology/2014/oct/27/elon-musk-artificial-intelligence-ai-biggest-existential-threat). Where ever AI ends up, quantum is likely to be right beside it.

Supply Chain Management

Businesses have long wanted to optimize supply chains so that they get the goods exactly when they need them, no sooner and certainly not later than they need them to conduct business with customers. Inventory is just wasted money, space, and other resources. Today, we have many companies that appear to be nearing perfect supply chain management, including Amazon, FedEx, and the United States Postal Service. Every retailer is following their lead, trying to optimize and better secure their supply chain. Even nonretailers are trying to optimize collection and distribution. For example, next-generation energy grid management is key to the world reducing energy waste, reducing costs, preventing blackouts, and making consumers happy. Many of the world's largest retailers and energy companies are investing in quantum computing for just that purpose.

Quantum Finance

It goes without saying that if someone can make a buck off quantum computing they will. Quantum computing will supposedly allow investors to better manage their investment portfolios and better understand the multitude of factors that impact finance. It will impact stock trading, derivative creating and trading, market predictions, commodity trading, and anything else that is traded in the world today. And like the cybersecurity prediction, much of it will be done by automated algorithmic bots trying to find a technical advantage that the other company's bot hasn't already found and acted on. We have the precursors of this with automated *high-frequency trading*, which accounts for up to 40 percent of all stock trading. Quantum computing will only accelerate those trends. There's already a quantum-focused financial website called Quantum for Quants (http://www.quantumforquants.org/).

Improved Risk Management

Financial investing is really all about portfolio risk management—what to invest in and when. Quantum computers will improve risk management calculations for all industries. They will help the insurance industry better determine actuarial odds, detect more fraud, and decrease false positives. They will help computer security defenders better determine what they should and shouldn't be concentrating on. Solving risk management problems is like moving pieces across a chessboard, and quantum computers are good at solving complex problems with lots of factors.

Quantum Marketing

Many of the best technological inventions of our time were largely driven and funded by advertising and marketing. Radio, television, and cable TV were driven by advertising. The Internet has stepped up the ad game and allows advertisers to market to specific groups of people who are more likely

to buy a particular product. The joke goes that Google knows more about our true selves than our spouses or what we ourselves would be willing to admit to ourselves. The Internet knows all. Quantum computing will allow more specific marketing to even smaller subgroups of people and enable us to better understand the complex interaction of seemingly unrelated consumer choices. Here's an example: say dog lovers are found to buy more spaghetti, and so someone buying dog products might be shown a coupon for a new spaghetti sauce. Just like quantum computing did with helping us to get a better understanding of molecule interactions, so too will quantum computing power be put to use for marketing. I'm not sure whether to classify this is as a good or an evil use.

Better Weather Prediction

Better weather prediction is another top reason for the early funding in quantum computing. Being able to more accurately predict the weather impacts everyone in the world. Not only does it help protect people from severe weather, but it also improves criticality, likelihood, and pathway predictions. It will help farmers determine what to plant and when. It will help monitor and mitigate the impacts of climate change by better modeling the global climate environment, how it changes and due to what, and humans' impact on it.

Quantum Money

Quantum money is essentially a cryptocurrency with very protective, quantum-based, anti-forgery features that make traditional blockchains seem quaint. There have been several proposals for quantum money systems, some going back 40 years, which recognized that the protective quantum properties could be used to create a universal, unforgeable, un-stealable cryptocurrency. A few types of quantum currency have been proposed with different theories, but most contain the concept of a unique serial number that has embedded nonshared quantum properties that only a central bank/verifier would know. Thus, criminals might be able to create a new duplicate version of the currency using the same serial numbers, but because they would not know the states of the additional embedded quantum properties, they could not create perfectly duplicated currency that would pass the bank's verification process. For more information, visit

```
https://futurism.com/the-byte/virtual-money-quantum-galactic-commerce
https://en.wikipedia.org/wiki/Quantum_money
www.scottaaronson.com/papers/noclone-ccc.pdf
```

Of course, today's modern cryptocurrency systems could use quantum-resistant ciphers instead of quantum-susceptible ciphers and be considered "quantum money" as well. Using a quantum-based cryptocurrency, which uses a distributed (and sometimes anonymous) blockchain instead of a centralized verifier, is considered a key requirement by many cryptocurrency users. With quantum, you can have the best of both worlds and choose the system that works best for you and your purposes.

Quantum Simulation

Using quantum computers based on quantum properties will allow us to explore our natural universe far better. Not only will we be able to determine the how's and why's of quantum mechanics, but we'll also be able to figure out all the quantum interactions of all the things we don't necessarily associate with quantum mechanics but that nevertheless rely on them. Quantum computers will likely help us answer many of the huge physics questions facing us today, such as how many dimensions are in our physical universe and what dark matter is. Quantum computing will allow us to understand the universe (or even multiverses) around us in a way that simply is not possible using classical computers.

More Precise Military and Weapons

It also goes without saying that many of the cool new things we invent and improve using quantum computers will go to improving the world's various militaries. That is what happens with many technological improvements. It's how we got the Internet. And simple, harmless things, such as better weather prediction, will be used in battle planning. It's just a given.

Quantum Teleportation

Probably one of the most talked about and interesting quantum inventions is *quantum teleportation*. Using quantum teleportation, it is possible to exactly re-create an object (or its state[s]) at a second (receiving) location, no matter how far apart those two locations are, including separated by the large voids of space.

Many people, including this author, have started the discussion about quantum teleportation by unsophisticatedly referring to the science fiction television and movie series *Star Trek* and its fictional "transporter" (`https://en.wikipedia.org/wiki/Transporter_%28Star_Trek%29`). This is usually done just to quickly convey the overall concept of teleportation, although quantum teleportation is more akin to copying or faxing than teleportation. Many physicists have bemoaned the use of the word *teleportation*, because it often denotes something quite different in people's minds.

Several key differences exist between the fictional *Star Trek* teleporter and quantum teleportation. First, the *Star Trek* teleporter sends an object's particles themselves between two locations, whereas quantum teleportation sends only information about an object's particles (so that they can be mimicked) and not the particles themselves. Second, and most importantly, quantum teleportation actually works, although so far with nothing bigger than a macro molecule.

NOTE The *Star Trek* teleporter's ability to seamlessly send an entire human being is something that is likely to be beyond our real-world capabilities for a long, long time. To send an entire human using either the fictional teleporter or real quantum teleportation would require understanding, identifying, and encoding every iota of what it means to be a human being—not only every cell, atom, and quark property, but whatever it takes to represent active memory and consciousness (and

unconscious thoughts and behaviors). We are not even sure that it can be done using any known physical representation, and the number of particles and states that would have to be transmitted would be more than all the stars in the known universe.

Teleportation Protocol

Quantum teleportation was theorized back in the early 1990s, and more recently it has been successfully demonstrated (using photons, atoms, and molecules) in labs and between Earth and space (using quantum-based satellites) in dozens of experiments. There are many practical difficulties in making a simple particle-transmission device like the fictional *Star Trek* teleporter, such as the no-cloning theorem, which prevents direct copying of quantum particles. Because of this, quantum teleporters use an indirect logic method represented by a proven protocol to accomplish real-world teleportation. But it's important to fully understand how quantum teleportation works because it is behind many of the coming quantum developments and devices. It is how quantum-based network devices will work. We already have them today, and they will be discussed more in Chapter 8. So, it isn't just for science fiction fans. It is the way things will work in the not too distant future.

Quantum teleportation requires at least five things: an object to be teleported, two entangled qubits, and two binary bits for each qubit of information about the object you want to teleport. A simplified version of the teleportation protocol looks like this (see Figure 5.4 for a graphical representation):

1. Create two entangled qubits (A and B) for each qubit of information needed about the object to be teleported (X1). The entanglement is crucial because we need both qubits to stay synchronized until the final state change.
2. Transport one side of the entangled qubit(s) to the location where the object is to be teleported to (i.e., the sending station).
3. Transport the other side of the entangled qubit(s) to the receiving location (however you want to accomplish this).
4. At the sending station, allow the object's qubit state and the sending side of the entangled qubits to interact with each other and observe and record the difference between the object's qubit property state and the present entangled qubit's state.
   The difference can be accurately represented by one of four possible answers (given two qubits compared to each other), say, 1, 2, 3, or 4.
5. Use the two binary digits (i.e., $2^2 = 4$) to represent the comparison difference answer.
6. Transmit the difference answer now represented by the binary digits, using any classical means, to the destination location. Alternately, you could use any communication method to communicate the difference answer, including writing or calling, but the quickest method would likely be a basic digital bit transmission of some type.
7. Use the received binary different answer information to modify the destination qubit in a way that accurately reflects the qubit (state) of the original object as measured in step 4.
8. Repeat as needed to correctly re-create the object (X2) at the destination qubit by qubit.

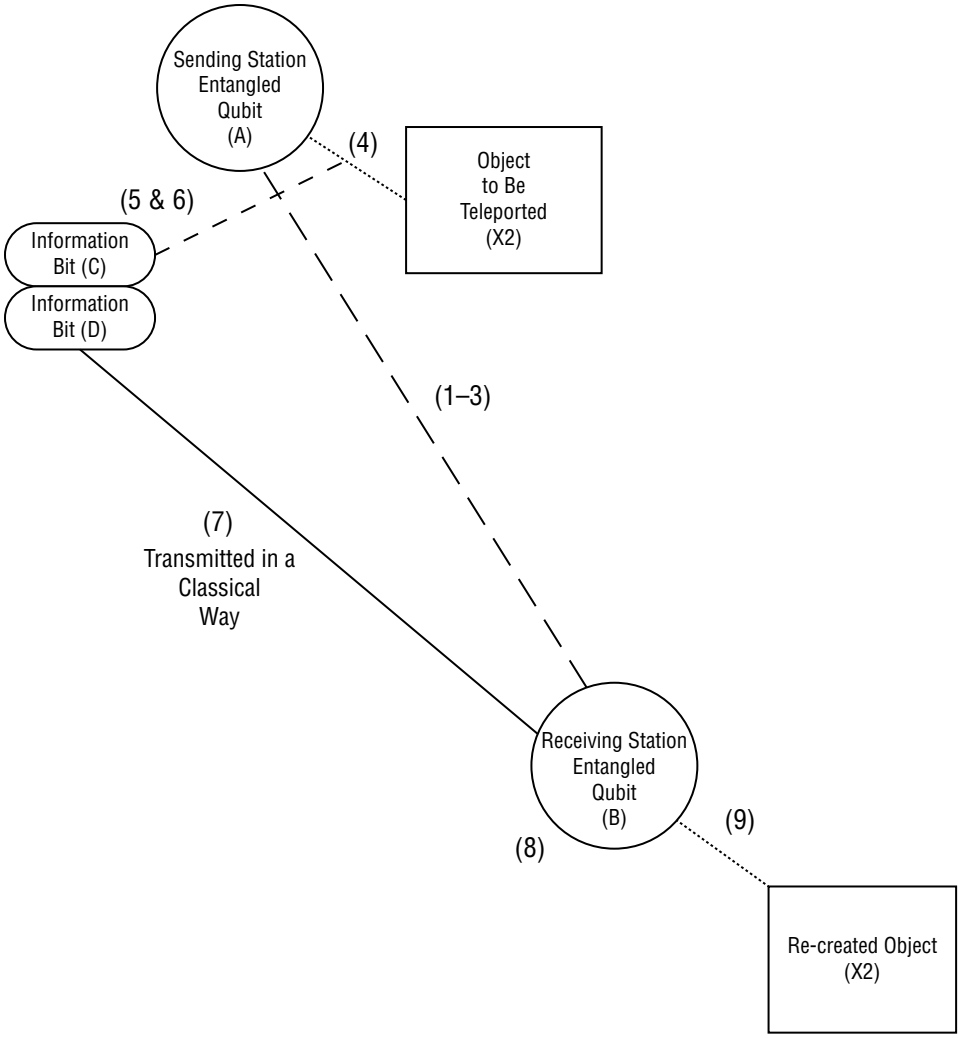

Figure 5.4: The basic quantum teleportation objects and steps

Now on the negative side, the act of teleportation breaks the original entanglement and destroys the original object (or it might even have to be destroyed for ethical reasons). For instance, suppose you teleport a living human. Now you have an identical copy of a human, with the same thoughts and memories, in two different places. They both would think they are the real, original human. One of those copies needs to be destroyed, or all sorts of negative strangeness can start occurring. For instance, both copies would think they were married to the same person, had the same parents, worked the same jobs, and so on. How would you like to be the human using quantum teleportation knowing that you're likely to be destroyed or killed in the process while the new copy of you gets to live a full life (or until it is next teleported)? Luckily we don't have these types of ethical issues when we are just trying to teleport regular digital data, which is what most quantum teleportation will be used for.

NOTE The *Star Trek* teleporter examples on television and in the movies just show a bunch of slowly disappearing colorful particles left in place of the teleported human. What they don't show is that the process destroys the original copy of the human. They make murder seem so pleasant and colorful.

If you are interested in learning more about quantum teleportation, visit

```
https://en.wikipedia.org/wiki/Quantum_teleportation
www.youtube.com/watch?v=Czi5elPLfvA
www.youtube.com/watch?v=hTe2PYwnEpc
www.scottaaronson.com/qclec/10.pdf
```

It should be noted that quantum teleportation isn't some insanely unique method of teleporting objects. We already do it all the time using traditional technologies. We frequently encode information about objects and then send that representative information to another destination, where the original object is re-created. Facsimile (fax), copy machines, scanners, and Internet file transfers do it every time. Quantum teleportation is a way to ensure that the quantum-level bits of information get correctly transmitted and/or to super-securely transfer the bits. Nonquantum teleportation would have a harder time transmitting the quantum-level information.

One other important caveat about quantum teleportation. Because people are often visualizing the fake science fiction version, they imagine being able to transfer objects and people to faraway places, even galaxies, in a blink of an eye. Step #6 (shown above) of quantum teleportation requires a classical information transfer, which means faster-than-light teleportation is not going to happen. This should not be surprising. Nothing can be transmitted faster than the speed of light (although we have yet to figure out how quantum entanglement appears to do it), but the transporting of the differencing information must be done in a classical way. This isn't terrible. Many classical methods, even those that use simple electricity, approach the speed of light, so having to involve them means it can still be done very fast. Just not faster than the speed of light.

If all of these quantum application uses and improvements seem a little "pie-in-the-sky" and you are skeptical, think about how binary computers and memory storage devices changed our pre–personal computer world within a few decades. Early computers were huge and took up entire floors of buildings. They were limited to military uses and only the largest corporations. Computers eventually became ubiquitous, fairly cheap, and very small. Today, we wear them on our wrists, have more computing power in our smartphones than a Cray supercomputer just a few decades ago, and can fit every document we have ever made along with our entire music collection on a memory storage device the size of a fingertip. You can prove a fact or learn about anything within a few seconds. You can watch a video of how to do almost anything as an expert would do it. News travels around our globe in seconds. Quantum computers are getting ready to show us what the next generation of fantastical inventions and improvements will bring. And it will no doubt be wondrous.

Summary

Quantum computers and properties will weaken or break most traditional cryptography, including hashes, symmetric keys, and asymmetric ciphers. In some cases, using longer key sizes will be the answer to becoming quantum-resistant. In other scenarios, only complete replacement of the quantum-susceptible cipher with a quantum-resistant or quantum-safe cipher will be acceptable. We will all be moving to more quantum-resistant cryptography in the next few years.

Quantum computing will also bring about new, better cryptography, networking, perfect privacy, and an array of new or improved applications. We will have better chemicals, medicines, batteries, weather predictions, and military weapons. Every quantum leap forward in technology brings both the good and the bad. Quantum computing is no different. Welcome to a post-quantum world.

Chapter 6, "Quantum-Resistant Cryptography," discusses dozens of quantum-resistant ciphers, some of which you should be soon utilizing in your organization's applications.

II Preparing for the Quantum Break

Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto, First Edition. Roger A. Grimes.

6 Quantum-Resistant Cryptography

The cryptography we will be using in a post-quantum world is a combination of quantum-resistant and quantum-based cryptography. Quantum-resistant cryptography is traditional, binary-based, cryptographic algorithms that are resistant to known quantum attacks. A quantum cryptographic algorithm is cryptography that uses quantum computing and properties to protect information. This chapter will cover quantum-resistant cryptography, and Chapter 7 will cover quantum-based cryptography.

This chapter is full of cryptographic technical and advanced mathematical jargon. General computer security readers might wonder why they should be interested in all the technical details behind particular algorithms. They might feel that all they really need to know to do their job are the names of the post-quantum algorithms . . . and this is perhaps true.

But it can be extremely helpful to anyone involved with implementing cryptography to understand the basics of the cryptography involved. This chapter gives a basic overview of over two dozen quantum-resistant algorithms so that you can understand them much in the same way you likely already understand that large prime numbers give RSA and Diffie–Hellman ciphers their inherent protection and why that reliance on prime numbers makes them susceptible to quantum attacks. Knowing more than the name of a cryptographic algorithm can only help when someone, be it an end user or a boss, asks more specific questions about your particular post-quantum implementation plan. Plus, you'll be more confident when discussing the post-quantum plan with your peers. You don't have to understand every detail about a particular algorithm, but it helps to have a general idea about how it works.

NIST Post-Quantum Contest

There are several dozen existing quantum-resistant ciphers and digital signatures, most of which have been evaluated by various cryptographic experts and groups around the world for years. In 2015, the European Telecommunications Standards Institute (ETSI), along with scientists and researchers around the world, was the first large public group to seriously look at quantum-resistant ciphers and was then followed by many other groups in other countries, including the United States.

Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto, First Edition. Roger A. Grimes.

The U.S. National Institute of Standards and Technology (NIST; see www.nist.gov) has conducted public, "competition-like" contests for years to evaluate various, newly proposed cryptography to replace existing, weakening cryptography. The winning algorithms become the newer U.S. cryptographic standards, and their creators agree to allow them to be used royalty free.

Previous NIST contests, with heavy coordination and participation by the U.S. National Security Agency (NSA), have selected SHA-3 as the new hash standard and AES as the new symmetric key cipher standard. Overall, these previous public cryptographic contests were seen as huge successes. Not only did many qualified candidates get submitted and publicly evaluated, but most people felt that the contests and contest winners were appropriate (although this has not always been the case with every NIST/NSA cryptographic evaluation; see the sidebar "Dubious NIST/NSA Contests").

DUBIOUS NIST/NSA CONTESTS

Although the NIST/NSA contests today are relatively trusted by most stakeholders, there have been at least two previous instances where the NSA seems to have proactively weakened cryptographic standards to make them easier for the NSA to break. This was first done decades ago in selecting the *Data Encryption Standard* (DES) symmetric cipher in 1977. The NSA convinced IBM (the creator of DES, then known as the Lucifer cipher) to shorten the proposed protective key length from 64 to 56 bits, and they were trying to get it shortened to 48 bits. IBM compromised by making DES use a 64-bit key, but most of the protection was really in the first 56 bits.

The NSA again invited criticism in 2006 when they required all computer manufacturers (that sold to the U.S. government, usually the largest buyer of computers) to include a new, but ultra-vulnerable, random number generator (RNG) known as *Dual_EC_DRBG*. It contained a mathematical flaw that created a secret backdoor into any cryptography that relied on it. Of course, NIST and the NSA didn't advertise that it had a backdoor, and no one has ever been able to prove the flaw was intentional. But even after the flaw was found, NIST said the involved RNG had to be included in any computer sold to the U.S. government as part of a cryptographic collection known as Suite B, and by extension (because it's easier to make just one version of a computer) would be included on nearly every computer sold to even nongovernment customers. Even though vendors included it on their computers, most of the world did not use it.

Not to be stopped, the NSA even secretly paid very popular computer device vendors to use the buggy RNG, meaning that any customer using them was (likely unknowingly) using severely weakened protection. It was a cryptographic nightmare scenario come true. One of the great mysteries in the applied cryptography world is why those vendors, when caught red-handed accepting a bribe from the U.S. government to hoist a backdoored-RNG on their customers, didn't get more public backlash. Bruce Schneier created an excellent summary of the issue as it was known in 2007: www.schneier.com/blog/archives/2007/11/the_strange_sto.html. In 2013, all the previous suspicions about the intentionally buggy RNG and the NSA's plot to proactively push it on people was revealed by CIA leaker Edward Snowden.

Both incidents, rightly so, severely damaged the confidence many parties have with trusting the U.S. government to pick truly secure cryptography standards. It showed that government entities with the dual responsibilities of both protecting and spying on their citizens will often allow weak cryptography to win over better protection. Despite these two issues of serious and consequential mistrust, most observers feel that the NIST/NSA contests to pick recent new cryptography standards (i.e., SHA-2, SHA-3, AES, and post-quantum crypto) can be trusted.

Starting in February 2016, NIST began a new contest called the *NIST Post-Quantum Cryptography Standardization Process* (`https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf`) to select post-quantum (i.e., quantum-resistant, quantum-safe) cryptographic standards for public key exchange and digital signatures. First-round candidates had to be submitted for consideration by November 2017. NIST received 82 unique submissions and allowed 69 to continue as official "First Round" candidates. Of these, 17 asymmetric encryption ciphers and 9 digital signature schemes were selected in January 2019 to continue to the "Second Round" (`https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf`). Third Round candidates are expected to be formally announced in 2020 or 2021. The final new post-quantum cryptography standards are expected to be announced between 2022 and 2024. This chapter will summarize all of the official Round 2 candidates.

The NIST/NSA contest winners usually become official U.S. federal government standards through NIST Federal Information Processing Standards (FIPS) publications documents. These standards must be followed by all government computers and devices sold and used by the U.S. government, as well as by any government subcontractors. This has the impact of causing all devices and computers sold in the United States to contain and use those standards, because the U.S. government is the biggest single buyer of computing devices. Computing device and software vendors find it easier and more cost-effective to incorporate the U.S. federal standards in all the devices they make rather than to make government and nongovernment versions. Because the United States has the biggest economy in the world, U.S. standards often become de facto global standards (although many larger countries such as Russia and China create and use their own country standards). For this reason, much importance is placed on the NIST/NSA cryptographic standard contests. Their outcomes impact much of the computers and software used in our world.

If you're interested in the details of each NIST cryptographic submission, I strongly recommend that you download the algorithm submitters submission documents, located at `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions`. Most of the best details are contained in a PDF document under the `Supporting_Documentation` folder of the team's zip file submission. NIST requires that each cryptographic submission have a ton of relevant information, including how the algorithm works, weaknesses, strengths, attack resiliency, demonstrated performance stats, code examples, NIST security strength-level recommendations (further detail on this in a moment), and more. You can read the review comments of supporters and critics on the

NIST contest submission website as well to get a general sense of how the algorithm is faring under review by cryptographic experts.

NIST Security Strength Classifications

In the NIST Post-Quantum contest, all submitted algorithms had to include specific implementations aligned to particular protection strengths as represented by currently existing quantum-resistant symmetric ciphers and hashes, as shown in Table 6.1. Increasing protection is indicated by increasing NIST security level numbers (e.g., NIST security level 4 provides more protection than security level 3).

Table 6.1: NIST security strength classifications and equivalent protection

NIST security level	Equivalent security	Quantum resistance
1	AES-128	Weak
2	SHA-256/SHA3-256	Strong
3	AES-192	Stronger
4	SHA-384/SHA3-384	Very Strong
5	AES-256	Strongest

NIST considers all five classifications quantum-resistant, although it describes security level 1 as "likely secure for the foreseeable future, unless quantum computers improve faster than is anticipated," which is to say fairly weak. This is because quantum computers using Grover's algorithm can cut AES-128's protective strength effectively to only 64 bits. There are currently no publicly known attacks that can break AES with 64 bits of strength, but future attacks that can are not that far away. Accordingly, many cryptographic experts do not consider cryptographic implementations barely meeting NIST security level 1 as being truly quantum-resistant for the long term. NIST, however, sees them as currently acceptable and views them as a "bridge" to using more resistant cryptography as time and resources allow.

NIST considers security levels 2 and 3 "probably secure for the foreseeable future" and levels 4 and 5 "likely excessive." Crypto people trust "likely excessive" resistant crypto, but performance and implementation considerations may prevent them from currently being deployed. For example, many current software programs and devices use AES-128 by default and cannot use AES-256 or larger (yet).

Most NIST competitors submitted implementations of their algorithms to meet NIST security levels 1, 3, and 5, and most skipped levels 2 and 4. There were exceptions. NTRU Prime did not submit NIST security levels 1 and 5. ThreeBears did not submit NIST security levels 1 and 3 ciphers but did submit levels 2 and 4. SIKE submitted NIST security level 2 examples, and NTRU Prime

submitted levels 2 and 4 as well. CRYSTALS-Dilithium and MQDSS did not submit a level 5 example. FALCON did not submit level 2, 3, or 4 implementations. LUOV did not submit level 1 or 3 samples.

For more information and detail on NIST security strength classifications, see section 4.A.5, Security Strength Categories, of the following NIST document:

`https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf`

There are excellent summary discussions of each algorithm on the NIST project website (`https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions`), at Privacy News Online (`www.privateinternetaccess.com/blog/2019/02/nist-round-2-and-post-quantum-cryptography-the-new-asymmetric-algorithms-part-2/`), and by many other sources, including Wikipedia (`https://en.wikipedia.org/wiki/Post-quantum_cryptography`).

Sections below summarize the NIST Round 2 candidates. Each cryptographic algorithm is described for what it does, what underlying principles are involved, sometimes a bit of history if warranted, and the national makeup of each submission team. The team makeup is included to show their often multinational makeup. Before the ciphers and schemes are discussed, related terminology and types will be defined.

NOTE Many algorithms covered in this chapter are also part of the Open Quantum Safe (`https://openquantumsafe.org/`) project. The Open Quantum Safe project is a code repository and organization helping to prepare us for the post-quantum world. When a cryptographic implementation is noted as participating with the project, it means that it is possible for an organization to implement that particular post-quantum algorithm in test and real-world scenarios today, use lessons and information from previous implementers, and share back challenges and new lessons learned.

PKE vs. KEM

Traditional public key cryptography, also known as *public key encryption* (PKE), is often used to transmit symmetric encryption keys, which are then used to encrypt the originally intended plaintext content needing the encryption protection. Symmetric keys are faster and stronger (for smaller key sizes) than asymmetric encryption, and so PKEs are often just used as a secure transport vehicle for the symmetric keys that really do all the direct encryption work. PKEs have worked great for decades, but they have at least one big inherent flaw.

When the public key is longer than the content being encrypted (such as is usually the case with the symmetric key in key exchanges), it allows attackers a very easy way to derive the original private key. To prevent this scenario, when the message content to be encrypted (e.g., the symmetric key) is shorter than the asymmetric private key used to do the encryption, PKEs will usually add additional "padding" to the message to be encrypted (e.g., the symmetric key) to remove the vulnerability.

Unfortunately, the random generation of the padding is often the weakest link in a PKE system. PKE attackers often focus on logic flaws in the padding and find vulnerabilities. Symmetric keys are likely only to get longer, especially to mitigate against improving quantum attacks. This presents a potential ongoing risk.

Key encapsulation methods (KEMs), also known as *key encapsulation schemes*, are a type of asymmetric encryption technique designed to improve the secure transmission (or generation) of symmetric keys because they don't need random padding added to short messages to stay secure. Many post-quantum cryptographic algorithms are especially conducive to creating KEMs and because post-quantum algorithms often have even longer asymmetric keys, you will see many quantum-resistant teams offering KEMs instead of PKEs. This is sometimes denoted by including KEM in the algorithm's name, such as is the case in FrodoKEM and NTS-KEM. Some post-quantum cryptographic algorithms will offer both PKE and KEM versions.

Formal Indistinguishability Assurances

The CPA and CCA letters you will see in some cipher names and descriptions are in reference to an always desired cryptographic property known as ciphertext indistinguishability. *Ciphertext indistinguishability* means the resulting ciphertext is so random-looking that an attacker cannot use the ciphertext to find easier attacks on the involved encryption keys. The *CPA* designation means the attacker can even have known, chosen plaintext submissions (i.e., *chosen plaintext attack*), see the resulting ciphertext, and still not get a clue about the involved secret encryption key. The *CCA* designation stands for *chosen ciphertext attack*, where an attacker can pick a particular ciphertext and have it decrypted to plaintext and still not get a hint toward the involved secret encryption keys. See `https://en.wikipedia.org/wiki/Ciphertext_indistinguishability` for more details.

The security of digital security systems is sometimes described as *EUF-CMA* and/or *SUF-CMA*, which can be somewhat seen as akin to the CPA and CCA descriptors used to summarize base asymmetric cipher system security. The EUF-CMA designation stands for *Existential Unforgeability under Chosen Message Attack*, and SUF-CMA stands for *Strong Existential Unforgeability under Chosen Message Attack*. With both designations, an attacker can ask for any content to be signed and be given the signature and still not be able to determine the private key used to sign the content. With SUF-CMA, which is slightly stronger security, the attacker cannot create a different digital signature, which still verifies under the original digital signing scheme as coming from the same content and private key (i.e., it would be bad form for the same unique content to generate two different valid signatures using the same digital signature scheme). Having these properties and proving that an algorithm absolutely has these properties, however, is the difference between being able to claim one or both of the descriptors. For more information on EUF-CMA and SUF-CMA, see `https://blog.cryptographyengineering.com/euf-cma-and-suf-cma/`.

All good PKE and KEMs will usually try to meet CPA- and CCA-secure requirements, and digital signatures will usually try to be EUF-CMA-secure. To use these designations, a cryptographic algorithm must first theoretically prove its security resistance and over time hold up to

sustained attacks. NIST required that successful cryptographic candidates be CPA- and CCA-secure or EUF-CMA-secure depending on the cryptographic algorithm type. Most submitted algorithms clearly state meeting these security assurance objectives in their submission documents, although NTRUPrime and NTS-KEM did not clearly call out meeting the CPA objective.

Key and Ciphertext Sizes

All asymmetric ciphers have two types of related cryptographic keys: secret and public. The *secret key* (also known as the *private key*) is used to sign content and should be known only to the key pair holder. In some post-quantum algorithms (such as CRYSTALS-Dilithium, SPHINCS+, and LUOV), the secret key is just a seed value used to generate other keys that do the actual work. In those implementations, the secret keys are usually small (16 to 64 bytes) and a constant size for any of the security strength implementations. In these cases, usually the related public key is very small as well. Quantum-resistant algorithms can have variable-length keys for different strength levels as well as different implementation intents (for example, a speedier, but less secure implementation) for different versions.

The *public key* is used to encrypt content and to verify content signed by the related private key. The public key can theoretically be known and used by the entire world and still the protected secret(s) would remain secret. The public key is meant to be used by everyone. It's the way asymmetric systems are intended to work. With asymmetric ciphers, the related public key is generated from the private key.

Ciphertext, in general, refers to any encrypted content, although in the context of comparing different submitted ciphers it refers to how big the smallest encrypted plaintext will become once encrypted. For example, if you encrypted only the letter *A* with a modern-day cipher, the resulting ciphertext will usually be much bigger than a single character.

A *digital signature* is the unique result outcome of hashed content. In digital signature schemes, the public key and digital signature sizes are inversely mathematically linked. If you decrease the size of one, the other grows, and vice versa. Thus, in most of the post-quantum digital signature submissions, if you see a small public key you will also see a large digital signature, and vice versa.

NIST required that all cryptographic algorithm submitters declare the sizes of each of these variables for each NIST security level declaration. Asymmetric encryption ciphers were also required to submit the size of the minimum ciphertext. Digital signature schemes were required to submit the size of the resulting digital signature. These sizes are important because very large sizes often have performance and storage issues (as compared to smaller competitors) and usually use up more memory space, storage space, and network bandwidth. The computational complexity of a particular algorithm, however, often has a far greater impact on overall cryptographic performance.

Many of the algorithms have more potential implementations with greater varying key sizes than just what was submitted to meet the specific NIST security strength levels. Some algorithms allow any key size to be used (within boundaries), depending on the implementer's desired security versus performance requirements.

Types of Post-Quantum Algorithms

When discussing different post-quantum algorithms, it's helpful to understand the major types of algorithms and summarize their method of protection against quantum-based attacks.

Code-Based Cryptography

Code-based cryptography (also known as *algebraic coding* or *error correcting codes [ECC]*) is a long-known and resilient-to-attack set of encryption and signature cryptography based on mathematical algorithms that intentionally induces "errors" (i.e., encryption) into plaintext content in such a way that it obscures/encrypts the original content. A corresponding "error-correcting" code/algorithm can be used to remove the "errors" and render the encrypted content back to its original, plaintext representation (i.e., decryption).

As a simple example, let's suppose the plaintext to be encrypted by the sender is 1111. "Errors" would be intentionally introduced into the content, say producing 011101, which would then be transmitted to the receiver. The "error correcting" process at the receiver's side would remove the "errors" and reliably reproduce the original 1111 content.

Code-based cryptography is based on ECC-like methods that are so complex that solving for the "errors" without knowing the involved key is very difficult (i.e., nontrivial) to break. In the 1970s and 1980s a Russian/Soviet mathematician, Valery Denisovich Goppa, linked geometric shapes and combinations to ECC. Today, these codes are widely known as *Goppa codes* and were adopted by cryptographers. One of the most successful code-based ciphers, McEliece (covered later in this chapter) is based on binary Goppa codes, as are most code-based ciphers in general. The second most popular type of asymmetric encryption cipher submitted to NIST was code-based ciphers. Code-based submitted ciphers include BIKE, Classic McEliece, HQC, LEDAcrypt, NTS-KEM, Rollo, and RQC.

There are two large technical challenges for code-based cryptography. First, ECC cryptography requires significantly more key bits than is usual (as compared to other cipher types) to encrypt data. Code-based cryptography cipher keys, especially the public keys, can easily range over 300,000 bits. This used to be a huge problem back in the 1970s and 1980s when McEliece was first introduced, but it isn't as big a computational obstacle today. Additionally, many code-based ciphers have to be able to significantly reduce their key size (for example, many use 40-byte secret keys). But if you see an absolutely huge key associated with an asymmetric encryption cipher, it is likely to be code-based.

Second, since ECC is correcting supposed "errors," there is always a chance, without appropriate design, that "errors" get by, which means legitimate decryption even with the correct decryption key could fail. This implies that decryption could have to be performed one or more additional times, and there is even the chance that a specific ECC decryption instance might not ever work or could be stuck in a temporary, self-induced, denial-of-service looping event. Most ECC ciphers attempt to prevent these sort of lockouts, and lockouts are extremely rare—near zero. But if you read of an ECC

cipher having a "nonzero" decryption failure rate (such as HQC does), then be aware of at least the theoretical possibility.

A great summary discussion on ECC and Goppa codes can be found here:
`https://surface.syr.edu/cgi/viewcontent.cgi?article=1846&context=honors_capstone`

Hash-Based Cryptography

Hash-based cryptography is based on cryptographic hashes, as the name implies, and usually applies to digital signature schemes (versus encryption). As covered in previous chapters, a hash is a one-way function that converts hashed content into a representative set of bits (called a hash, hash result, signature, or message digest) that is unique for unique content. XMSS (eXtended Merkle Signature Scheme), Leighton-Micali Signatures (LMS), Blockchained Post-Quantum Signatures (BPQS), SPHINCS, and SPHINCS+ digital signature schemes are based on hash-based cryptography. SPHINCS+ was the only hash-based digital signature submitted and accepted by NIST in Round 2 of the contest.

Ralph Merkle essentially invented the field of cryptographic hashes, and he also participated in the first publicly known implementation of public key encryption (along with Whitfield Diffie and Martin Hellman) in the late 1970s. For that reason, you'll often hear about Merkle trees (i.e., hash trees) and Merkle boxes and puzzles when discussing hash-based cryptography. Merkle trees are a hierarchical series of hashes that hash other hashes that hash the original content. If interested, see `https://en.wikipedia.org/wiki/Merkle_tree` for more detail on Merkle trees.

Hash-based cryptography is considered quantum-resistant since hashes are not susceptible to Shor's algorithm, although they are susceptible to Grover's algorithm. Grover's algorithm on quantum computers gives a square root improvement over binary computers when doing particular types of problems, like cracking hashes. This effectively halves the strength of any hash-based cryptography. It also means that doubling the key size of the hash offsets the attack benefits gained by Grover's algorithm and quantum computing.

NOTE It is critically important that the underlying hash conform to all the attributes (previously discussed) of a good, secure hash. If a hash fails over time to be a "good hash," then any cryptography based on that hash would become susceptible not only to quantum computing but also to binary computing, at levels well below its stated key strength.

All hashes are limited by the number of messages they can protect before the hash results become (prematurely) redundant for all the possible unique inputs they can hash. For example, as of this writing, all Microsoft Windows logon passwords are converted to NT hashes. You can create many quadrillions of unique possible passwords in Windows (nearly 2^{65535} different combinations), but the same NT hash would end up being identical for many different passwords (in hash attack theory this is known as an example of a *second preimage collision*), because of inherent limitations of the hash and its key space (i.e., all possible choices).

If hash-based cryptography "accidentally" repeats the same onetime key for two different inputs, it would give attackers strong insight into the private key. For this reason, hash and hash-based cryptography developers go to great lengths to prevent onetime key repeats. There are a few different methods to mitigating them.

One way this risk can be addressed is by increasing the accuracy of the hash algorithm to distinguish unique content. If the hash always makes unique hash outputs, then the repeating problem is gone. This can be difficult to do because the key space of the hash is always limited more than the potential things it is trying to hash. To offset this risk, hash developers can also increase the size of the resulting digital signature (i.e., to give more key space). The longer the digital signature, the more possible hashed outcomes. Thus, a hash with a 128-bit hash result is likely to be more accurate than one limited to a 64-bit result. Very large digital signatures can become overly large and unwieldy, causing performance and storage problems. Most cryptographic experts believe that a good hash, using its inherent algorithmic accuracy, should not lead to very large digital signatures. Others believe that large digital signatures are the only way to be assured of an accurate hash without built-in hash result redundancy. Either way, treat very small and very large digital signatures with extra consideration.

Another common way to prevent key repeats is to make the hash *stateful* (vs. *stateless*). A stateful hash keeps track of every onetime secret key it has used and makes sure that it doesn't reuse it again. Most traditional signature-based hashes are *stateful*. If a repeated key is detected, the algorithm is run again or a different part of the longer keystream is chosen to generate another, unique onetime key.

Both stateful and stateless hashes have their advantages and disadvantages. Stateless hashes cannot guarantee unique keys but in general have larger key sizes. Stateful hashes have smaller key sizes in general, but because they have to store a "state table," they are bulky to do from a resource, storage, and security perspective. Stateful hashes can also create a critical issue during data restore events. If handled with insufficient care, a restored stateful hash implementation might overwrite its previous state table, erasing evidence of a previously used key, and then the hash might accidentally reuse that same key again with a future cryptographic action. An attacker knowing that the state table has been overwritten might look for signs of a repeated key and get an advantage to use in a cryptographic attack. This is a fairly unlikely event with pretty low risk, but if cryptographers see any theoretical weakness, it's considered a big weakness. Thus, NIST would not allow stateful hash-based cryptographic algorithms to be submitted for consideration, knocking out several otherwise strong competitors.

Lattice-Based Cryptography

A *lattice* is a dimensional, distributed, repeating geometrical arrangement/pattern of something—for instance, objects or points—in a space. Lattices occur all throughout nature, such as in molecules or crystals, and are often used by people to create other much larger objects, including nets, fences, or weave patterns. Many mathematical formulas and algorithms create lattices. Lattice-based formulas and results have been created that are fundamentally difficult to factor (known as *computational*

lattice problems). The most common lattice problems used in cryptography are known as learning with errors (LWE), ring learning with errors (RLWE), module learning with errors (MLWE), learning with rounding (LWR), and dozens of variants. Each type of problem has its own advantages and disadvantages. In general, LWE involving rings tends to be faster and have smaller key sizes than classical LWE but also contains new mathematical constructs that have not been fully tested over time.

NOTE You will read a lot about "rings" when learning about cryptography, especially post-quantum cryptography. Rings refer to a fundamental and complex mathematical structure used in abstract algebra. See `https://en.wikipedia.org/wiki/Ring_(mathematics)` for more details.

These very hard-to-solve lattice problems have been used to create public key encryption and digital signature schemes, which are resistant to both binary and quantum computers. With lattice-based cryptography, a complex lattice function is created as the private key. The public key is generated as a modified version of the original lattice. Content is encrypted using the modified version (the public key), and only the holder of the original lattice version (the private key) can easily recover the encrypted message back to its original plaintext state.

Fundamentally, lattice-based cryptography has created a math workload problem that is somewhat equivalent to or greater than the workload effort needed to factor large prime numbers but does not rely on large prime numbers for their protection. Subsequently, lattice-based cryptography is not considered susceptible to Shor's algorithm or any quantum algorithm that factors primes.

On a negative, theoretical note, lattice-based cryptography could require relatively larger key sizes compared to other cipher types, although very importantly, this has not held true for most of the lattice-based candidate submissions to NIST, including CRYSTAL-Kyber, LAC, NewHope, NTRU, NTRUPrime, Round5, SABER, and ThreeBears. Only FRODO-KEM has a relatively large key size, although several code-based algorithms are much larger.

The first lattice-based cipher was NTRU, introduced in 1998, followed by multiple ciphers based on LWE and RLWE math problems. Today, lattice-based ciphers are the most popularly submitted type of post-quantum crypto submitted to NIST. Additionally, in 2009 Craig Gentry in his dissertation (`https://dl.acm.org/citation.cfm?id=1834954`) used lattice-based cryptography to create the first real-world fully homomorphic encryption system, which, as discussed in a previous chapter, allows a third party to correctly manipulate encrypted data without first decrypting it.

NOTE Most lattice-based ciphers and their related problems are based on *shortest vector problems* (SVPs), which require at least super-exponential time to solve. Unfortunately, the overall security provided by SVPs is not completely understood, and some theoretical attacks have significantly weakened their protection. For that reason, all lattice-based cryptography (and especially those based on SVPs without offsetting mitigations) are not completely trusted and could be found to be weaker than previously understood in the future. This could be problematic when coupled with the fact that most post-quantum ciphers (submitted to NIST) were lattice-based ciphers.

Multivariate Cryptography

Multivariate simply means "multiple variables." *Multivariate cryptography* refers to asymmetric encryption and signature schemes that rely on multivariate polynomial math equations, such as $x + y + z = n$, to form the cryptographic primitives. You will also hear cryptography based on multivariate polynomial math referred to as *multivariate quadratic (MQ) polynomial equation cryptography*. This refers to the fact that at least one of the variables is raised to the second power (e.g., $x^2 + y + z = n$). Correctly created multivariate cryptography cannot be solved in polynomial time and does not rely on large primes for protection. Hence, they are considered quantum-resistant. Their inherent characteristics also make them a good performance candidate for hardware implementations, such as application-specific integrated circuits (ASICs) and field-programmable gate arrays (FPGAs).

Multivariate cryptography includes HFE, Gui, Balanced Oil & Vinegar, Unbalanced Oil & Vinegar (some algorithm names are intentionally humorous), and Tame Transformation Signature. Submitted multivariate digital signature schemes include GeMSS, LUOV, MQDSS, and Rainbow. Rainbow is a multilayered implementation of Unbalanced Oil & Vinegar.

Supersingular Elliptic Curve Isogeny Cryptography

Supersingular elliptic curve isogeny cryptography (or isogeny cryptography for short) relies on math equations and algorithms that create supersingular elliptic curves and isogeny graphs for their encryption protection. Elliptic curves are created by mathematical formulas that represent algebraic curves that do not self-intersect (also known as *nonsingular*). All supersingular curves are nonsingular, and the "super" part refers to unusually large rings. *Isogeny* refers to separate algebraic groups that share an intersection of related values between each other. As a simple example, imagine you had numbers 1, 2, 3, and 4 in one group and the letters A, B, C, and D in another group, and each number was related to a corresponding letter. They are isogenic to each other. Isogenic curves would be two curves (represented by math, of course), which can be mapped to each other.

In the isogeny cryptography world, two different algorithm equations are creating an isogenic linkage that can be used to encrypt and decrypt. The public key is a pair of elliptic curves, and the private key is an isogeny between them. Finding this isogeny given only the pair of supersingular elliptic curves is believed to be a very hard problem to solve. If this sounds complicated, know that supersingular elliptic curve isogeny equations are among the most difficult math problems ever created but have been studied well enough to appreciate their strengths and weaknesses.

In 2012, Chinese researchers created the first quantum secure digital signatures based on supersingular elliptic curve isogenies and multivariate cryptography (`https://pdfs.semanticscholar.org/527a/4abe13ee6ce7858e040ceaa7cd0b983969d8.pdf`). Isogeny cryptography tends to have very small key sizes and also allows easy *perfect forward secrecy*. Perfect forward secrecy is a cryptographic protection involving frequently changing session keys so that a future key compromise cannot be used to more easily crack previous sessions because they were using different keys. Perfect forward secrecy is usually a desired cryptography trait, although it often cannot be attained. Isogeny is agreeable

with perfect forward secrecy. On the downside, isogeny cryptography is relatively new, so it hasn't been tested and attacked as much as the other post-quantum cryptography types. Although early isogenic implementations have been compromised, changes such as using supersingular isogeny instead of just nonsingular curves prevented the known attacks. The only isogeny cipher submitted and accepted by NIST for Round 2 evaluation was SIKE.

Zero-Knowledge Proof

A *zero-knowledge proof* (ZKP; also known as a *zero-knowledge protocol*) is a method by which one party (called the *prover*) can prove to another party (called the *verifier*) that they know value *x*, without having to convey or prove any information except for the fact that they really know the value *x*, without actually providing value *x* or leaking any extra, nonessential information.

An example of a common ZKP system is a logon password used in a modern-day "challenge-response" authentication system. Let's suppose a user wants to log on to a server using a valid password but at the same time not allow the server to know or store the plaintext password (so it can't be stolen or compromised). How can the user prove to the server that they have (and can use) the correct password without providing the correct password itself?

One ZKP answer is to use cryptographic challenge-response hashes based on the password but without using the actual password. For example, suppose the user's plaintext password is *frog*. When the user creates it for the first time, let's imagine that the plaintext password of *frog* is immediately hashed and the hashed result is *1234*. The hashed result is the only version of the password sent to and stored on the server. The server has no way of knowing the original plaintext password.

When the user wants to log on to the server, they initiate a connection to the server. The server creates a random value, say *9876*, and sends it to the user (called the *challenge*). The user subtracts the password hash of *1234* from *9876* to get a result *8642* and sends it back to the server (called the *response*). The server uses the user's stored password hash of *1234* to perform the same subtraction on the randomly generated number and, in doing so, will get the same result (8642) and compare it to the result the user sent back. Only a user with the correct original password of *frog* would have the correct hash of *1234* and be able to get the correct result when subtracting it from the randomly generated value of *9876*. So, the user can successfully prove they had the correct original password to the verifying server without revealing what the plaintext password is. Most modern-day authentication systems, including Microsoft Windows passwords, use a similar (although more complex) scheme.

ZKP implementations are claimed by many computer vendors. Like other computer security buzzwords before it (such as artificial intelligence [AI] or blockchain), ZKP is overused. It is incorrectly used by many vendors to inaccurately describe far more offerings than really use it. So, be initially skeptical whenever you see a vendor using the ZKP phrase. With that said, cryptographers are a fairly serious and truthful group of people. When a cryptography says their algorithm uses ZKP, they usually aren't saying it lightly or without support. ZKP cryptography typically involves proving knowledge of something cryptographic, such as a discrete logarithm function, without revealing the function

itself. In crypto circles, you may hear the "prove and verify" steps also referred to as *sigma protocols* or (three-step or three-message) proofs. The only submitted quantum-resistant algorithm that uses ZKP is the Picnic digital signature scheme.

Symmetric Key Quantum Resistance

This refers to the inherent ability of traditional symmetric key encryption and authentication algorithms to resist quantum attacks. As covered previously, symmetric ciphers are not susceptible to Shor's algorithm, and Grover's algorithm cuts their protection in half. So, in a sense, symmetric key ciphers are already quantum-resistant, as long as their key sizes are of sufficient size to fight off Grover algorithm attacks. Today, this means using symmetric ciphers with key sizes of 256 bits and longer for long-term protection. NIST and others will accept symmetric ciphers with 128-bit keys as being weakly quantum-resistant, with 192-bit symmetric keys considered moderately quantum-resistant. Accordingly, symmetric ciphers are not in NIST's latest post-quantum contest.

All the submitted algorithms use traditional symmetric key ciphers and hashes as part of their implementation. The most common symmetric cipher used in the world today is Advanced Encryption Standard (AES), and this also holds true for quantum-resistant ciphers. Most quantum-resistant ciphers use AES.

You will also see the SNOW 3G quantum-resistant symmetric cipher, although it is far less popular and is not used in any submitted cipher proposal. SNOW is a word-based, synchronous stream cipher, developed by Swedes Thomas Johansson and Patrik Ekdahl at Lund University. SNOW (version 1), SNOW 2.0, and SNOW 3G (adapted for cellular network use) are implemented in several products and applications. You can get more detail on SNOW and SNOW 3G here: `www.gsma.com/aboutus/wp-content/uploads/2014/12/uea2designevaluation.pdf`.

All submitted ciphers also use traditional hashes, which like symmetric key encryption are not susceptible to Shor's algorithm. Most use SHA-3 (another NIST standard), although many also use SHAKE, a stream cipher, which is used by SHA-3 as well. Quantum-resistant ciphers must use the appropriate key lengths of these traditional ciphers to remain quantum-resistant, and what key size a submitter used often changed relatively to the NIST security level they were trying to meet with a particular implementation.

Table 6.2 shows all the NIST Round 2 algorithm names along with their cryptography types.

All of these types of algorithms have been used to create quantum-resistant cryptography, and each will be summarized in the next section.

Table 6.2: NIST Round 2 cryptography types

Asymmetric encryption/KEMs	Type	Signatures	Type
CRYSTAL-Kyber	Lattice	CRYSTALS-Dilithium	Lattice
FrodoKEM	Lattice	FALCON	Lattice
LAC	Lattice	qTESLA	Lattice
NewHope	Lattice	SPHINCS+	Hash
Three Bears	Lattice	GeMSS	Multivariate
NTRU	Lattice	LUOV	Multivariate
NTRU Prime	Lattice	MQDSS	Multivariate
SABER	Lattice	Rainbow	Multivariate
Round5	Lattice	Picnic	Zero-knowledge proof
Classic McEliece	Code		
NTS-KEM	Code		
BIKE	Code		
HQC	Code		
LEDAcrypt	Code		
Rollo	Code		
RQC	Code		
SIKE	Isogeny		

Quantum-Resistant Asymmetric Encryption Ciphers

Quantum-resistant ciphers are cryptographic ciphers that are not overly susceptible to quantum computers running quantum-based algorithms and, in particular, Shor's algorithm (or any quantum algorithm that can factor large prime number equations very fast). They do not use quantum-based properties to defeat attacks. There are dozens of quantum-resistant ciphers, although one or more of the 17 Second Round NIST asymmetric encryption candidates are likely to be one of the eventual NIST federal standards. The NIST Second Round asymmetric PKE and KEM candidates (in alphabetic order) are as follows:

- BIKE
- Classic McEliece

- CRYSTALS-Kyber
- FrodoKEM
- HQC
- LAC
- LEDAcrypt
- NewHope
- NTRU
- NTRU Prime
- NTS-KEM
- ROLLO
- Round5
- RQC
- SABER
- SIKE
- ThreeBears

Some of these 17 ciphers are combinations of multiple ciphers submitted separately in the first contest round that were combined into a single cipher family with similar characteristics. For example, HILA5 and Round2 eventually became Round 5.

NOTE Algorithm names are all capitals or initial-capped according to their creator's original use. Names in all caps are usually abbreviations standing for longer whole-word names.

NOTE This list by no means includes all possibly strong post-quantum asymmetric algorithms. Many existing post-quantum cryptographic algorithms (asymmetric and digital signatures) were not submitted for a host of reasons, including that the creators did not want to give up their algorithm for free public use; many were attacked and broken after submission (22 of them); and many had weak submission papers or did not meet the NIST acceptance criteria (e.g. XMSS, LMS, and BPQS). Algorithms that were originally submitted for First Round consideration but did not make it to Round 2 include the following: BIG QUAKE, CFPKM, Compact LWE, DAGS, DME, DRS, DualModeMS, Edon-K, EMBLEM/R.EMBLEM, Giophantus, Guess Again, Gui, HiM-3, HK17, KCL, KINDI, Lepton, LIMA, Lizard, LOTUS, McNie, Mersenne-756839, pqNTRUSign, Odd Manhattan, Post-quantum RSA-Encryption, Post-quantum RSA-Signature, QC-MDPC KEM, RaCOSS, Ramstake, RankSign, RLCE-KEM, RVB, SRTPI, Titanium, and WalnutDSA. Additionally, there were dozens of quantum-resistant algorithms that were not submitted, including GGH, XMSS, and UOWHF. These cryptographic algorithms have been or could be accepted by other standards bodies and nations.

BIKE

Bit Flipping Key Encapsulation (BIKE) is a code-based KEM suite. Created by a multinational team (mostly from France, but with participants from Germany, Israel, and the United States), it has three different variants called BIKE-1, BIKE-2, and BIKE-3. It is based on McEliece encryption, QC-MDPC (Quasi-Cyclic Moderate Density Parity Check) codes, CAKE, Ouroboros (a Round 1 NIST candidate that didn't make it to Round 2 alone), and ephemeral keys. *Ephemeral keys* are cryptographic keys that are generated for each execution of a key establishment process, instead of using a single, static key. Ephemeral keys allow a cipher to use perfect forward secrecy.

BIKE has performance and key sizes similar to lattice-based cryptosystems. For the beginning of Round 2 for the NIST required security implementations, secret keys range from 1,988 to 4,110 bytes, public keys range from 20,326 to 65,498 bytes, and ciphertexts range from 20,326 to 65,498 bytes. BIKE has some of the largest public key and ciphertexts of any of the submitted candidates, although the size of its secret key gets an average ranking. One fairly interesting characteristic, shared by only a few other post-quantum ciphers, is that BIKE-encrypted data has a recognizable "signature" that could be used by adversaries and security controls to recognize and manipulate it. You can't usually tell what cipher was used to create most encryption, which complicates any attacks. Not so with BIKE. BIKE is part of the Open Quantum Safe project.

For more information on BIKE, visit: `https://bikesuite.org/`.

Classic McEliece

In 1978, Robert J. McEliece created a public key cipher that has withstood over 40 years of attacks. It is a code-based cipher using Goppa codes. McEliece's 1978 paper introducing McEliece public key encryption can be found here: `https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF`.

NOTE You will often see the related Niederreiter cryptosystem mentioned in the same places as McEliece. The Niederreiter cryptosystem is much faster and can be used for digital signatures.

Original McEliece is fast (as compared to RSA) and quantum-resistant but requires large key sizes (300 KB and longer, often greater than 1 MB). Over the years, several different teams have tried to modify it to reduce the required key sizes, but ultimately nearly all of the newer implementations were found to be far less secure than the original.

The NIST submission team, which includes highly noted cryptographer and secure coding software developer Daniel J. Bernstein, succeeded in modifying McEliece to allow it to use reduced key sizes while remaining quantum-resistant with zero decryption failures. For the beginning of Round 2 for the NIST required security implementations, secret key sizes range from 6,452 to 13,892 bytes, public keys range from 261,120 to 1,044,992 bytes, and ciphertexts range from 128 to 240 bytes.

Classic McEliece had the second largest public key size, surpassed only by NTS-KEM. Both had the smallest resulting ciphertext. While still significantly bigger than traditional public key ciphers and most of their post-quantum competition, they are very manageable using today's computers and networks. Classic McEliece has the added benefit of having very small ciphertext sizes and being very fast in hardware-based implementations.

For more information on Classic McEliece, visit https://classic.mceliece.org.

NOTE If you are interested in cryptography or writing secure code, anything Daniel J. Bernstein writes or releases is widely respected. He has given hundreds of talks, written nearly as many papers, and developed many different, very secure, low-bug-count programs. He invented the term *post-quantum cryptography* and was one of the early leaders making the rest of the world aware about the coming issue. Readers are encouraged to visit his personal website: https://cr.yp.to.

CRYSTALS-Kyber

CRYSTALS (Cryptographic Suite for Algebraic Lattices) encompasses two lattice-based cryptographic primitives: Kyber, a CCA-secure KEM, and Dilithium, a strongly EUF-CMA secure digital signature algorithm. CRYSTALS-Kyber is based on earlier MLWE-based encryption problems but uses square rather than rectangular matrixes as the public key along with polynomial rings (https://en.wikipedia.org/wiki/Polynomial_ring) rather than integers. It has good performance and can be easily scaled when larger key sizes are needed.

According to its multinational team of developers, Kyber-512 has security protection roughly equivalent to AES-128 (NIST Security Level Classification 1), Kyber-768 has security roughly equivalent to AES-192 (NIST Security Level Classification 3), and Kyber-1024 has security roughly equivalent to AES-256 (NIST Security Level Classification 5). For the beginning of Round 2 for the NIST required security implementations, secret keys range in size from 1,632 to 3,168 bytes, public keys range in size from 800 to 1,568 bytes, and ciphertexts range in size from 736 to 1,568 bytes. CRYSTAL-Kyber consistently ranks with average to smaller key sizes. It is part of the Open Quantum Safe project.

For more information on CRYSTALS, see https://pq-crystals.org/ and https://pq-crystals.org/kyber/index.shtml.

FrodoKEM

FrodoKEM is a CCA-secure and CPA-secure lattice-based cryptosystem that relies on LWE problem solving for its protection. It has slightly larger key sizes and slower performance as compared to other lattice-based models (based on LWE rings). It comes in three key sizes:

- FrodoKEM-640, which purports to have security equivalent to AES-128
- FrodoKEM-976, which purports to have security equivalent to AES-192
- FrodoKEM-1344, which purports to have security equivalent to AES-256

For the beginning of Round 2 for the NIST required security implementations, secret keys range in size from 19,888 to 43,088 bytes, public keys range in size from 9,616 to 21,520 bytes, and ciphertexts range in size from 9,720 to 21,632 bytes. FrodoKEM is among the largest public key and ciphertext sizes.

The FrodoKEM team has a lot of members from Microsoft and Google. The members took an earlier version of FrodoCCS, which is an ephemeral key exchange scheme, and improved it to make an IND-CCA-KEM. FrodoKEM is a simpler version, which involves less code. This makes it likely more reliable and resilient against attacks. And if a flaw is found, it might make it easier to fix. It is part of the Open Quantum Safe project.

Current FrodoKEM is also "constant-time" as built. It does not need to be reoptimized security-wise to prevent certain types of eavesdropping attacks. *Constant-time* is a cryptographic protection property designed to mitigate many types of side-channel *timing attacks*. In short, due to a variety of reasons, many ciphers and many early implementations of ciphers introduce CPU delays directly related to what the cipher is involved with (say, evaluating a cipher key). Any cipher process that changes processing time in relationship to the length of some examined variable can possibly create a measurable and predictable timing difference and give an attacker knowledgeable insight to otherwise secret information. This information might allow attackers to make assumptions to quicken their attack. In the cryptography world, these types of gained information are called *cribs*. You'll find a great basic discussion on timing side-channel attacks here: `www.chosenplaintext.ca/articles/beginners-guide-constant-time-cryptography.html`.

For more information on FrodoKEM, see `https://frodokem.org/`.

NOTE Joppe W. Bos, a cryptographic researcher at NXP Semiconductors, Leuven, Belgium, is a submitter of three post-quantum ciphers: FrodoKEM, CRYSTALS-Kyber, and NewHope.

HQC

QC (Hamming Quasi-Cyclic) is a code-based public key encryption scheme based on the difficulty of decoding random quasi-cyclic codes, which uses Bose–Chaudhuri–Hocquenghem (BCH) codes with a repetition code. BCH codes were invented in 1959 and 1960, and it's considered simple to control what "errors" they correct, thus making it easy to decode when you have the correct keys. BCH codes have been widely used in CDs, DVDs, barcoding, and computer storage devices for decades. Even though HQC uses "easy-to-decode" BCH codes, it is subject to "nonzero" (but still incredibly rare) decryption failures.

HQC has a 2^{-128} chance that any particular decryption round will not result in the original plaintext content. Computers, in general, have many other far more common errors that have a much higher chance of happening, and we accept using those computers just fine. A failure would mean it could take additional decryption rounds to be successful, but those additional rounds would happen very quickly and almost no one would notice. But the probability is nonzero, and so it's noted. Cryptographers are held to exacting reporting standards.

HQC has a secret key size of 40 bytes for all NIST required security implementations (which is tied for second smallest with three other submissions). For the beginning of Round 2 for the NIST required security implementations, HQC's public keys range from 3,125 to 8,897 bytes, with ciphertexts ranging from 6,234 to 17,777 bytes. Like BIKE, HQC has markers that can be used to identify HQC-encrypted traffic, and it uses ephemeral keys to allow perfect forward secrecy. The HQC team is multinational in its makeup, and many of them have submitted other algorithms to the NIST contest (for example, Frenchman Philippe Gaborit also worked on BIKE, HQC, RQC, and ROLLO).

For more information on HQC, see `https://pqc-hqc.org`.

LAC

The CPA- and CCA-secure LAC (Lattice-based Cryptosystems) cipher includes four different LAC-related primitives based on polynomial learning with errors (poly-LWE) problems over the ring. The LAC cipher primitives are

- LAC.CPA: A secure public key encryption scheme, which is the foundation for the other three implementations as well
- LAC.KE: A secure key exchange protocol that is directly converted from LAC.CPA
- LAC.CCA: A secure key encapsulation mechanism, which is related to LAC.CPA
- LAC.AKE: An authenticated key exchange protocol

LAC can run on both Intel and ARM processors (as do most of the other competitors), uses relatively smaller keys, and has good performance. For the beginning of Round 2 for the NIST required security implementations, secret key sizes range from 512 to 1,204 bytes, public key sizes range from 544 to 1,056 bytes, and ciphertexts range from 712 to 1,424 bytes. This gives LAC the fourth smallest combined key and ciphertext sizes of the NIST competitors.

There does seem to be a higher-than-average number of questions from NIST contest reviewers regarding LAC's security, including this comment: (`https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/official-comments/LAC-official-comment.pdf`), but so far none of them are claiming a significant break.

The LAC team is made up of Chinese cryptographers. Chinese cryptographers are involved in many quantum-resistant algorithms and submissions to NIST. This makes sense, because China is a leader in research on quantum computers, devices, quantum-based cryptography, and defenses.

For more information on LAC, download `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/LAC.zip`. The LAC team doesn't have a public-facing website, but this zip file has a lot of relevant information.

LEDAcrypt

LEDAcrypt (Low-dEnsity parity-check coDe-bAsed cryptographic systems) is an asymmetric cipher relying on Quasi-Cyclic Low Density Parity Check (QC-LDPC) codes and ephemeral keys. It was

created by a merger of LEDAkem/LEDApkc from the First Round, along with many improvements stemming from NIST suggestions.

LEDAcrypt is a modified version of the Niederreiter cryptosystem. QC-LDPC codes allow high-speed decoding and smaller key pairs. For the beginning of Round 2 for the NIST required security implementations, private key sizes range from 452 to 1,092 bytes, public key sizes range from 1,872 to 8,520 bytes, and ciphertexts range from 1,872 to 4,616 bytes. LEDAcrypt can allow perfect forward secrecy and uses SHA-3 (256 to 512 bits) for its hashing functions but, like many other code-based schemes, is susceptible to decryption failures. The LEDAcrypt team is Italy-based.

For more information on LEDAcrypt, see `www.ledacrypt.org`.

NewHope

NewHope is a CCA- and CPA-secure lattice-based key-exchange method based on the ring-learning-with-errors (ring-LWE) problem. It has four instantiations:

- NewHope512-CPA-KEM
- NewHope1024-CPA-KEM
- NewHope512-CCA-KEM
- NewHope1024-CCA-KEM

The 512 ring-dimension versions are purported to be equal to or greater than AES-128, and the 1024 ring-dimension versions are purported to be equal to or greater than AES-256. For the beginning of Round 2 for the NIST required security implementations, the CCA versions' secret key sizes range from 1,888 bytes to 3,680 bytes, public key sizes range from 928 to 1,824 bytes, and ciphertexts range from 1,120 to 2,208 bytes. NewHope has relatively good performance and has been explored quite a bit by Google. It is part of the Open Quantum Safe project.

For more information on NewHope, see `https://newhopecrypto.org`.

NTRU

NTRU (N-th degree Truncated Polynomial Ring) is a fast, lattice KEM based on the NTRU encryption scheme (which has been around since 1996 and is well studied). NTRU was one of the first public key cryptosystem not based on factorization or discrete logarithmic problems (after McEliece) and was the first asymmetric cipher identified as not being susceptible to Shor's algorithm. NTRU in NIST Round 2 is a merger of NTRUEncrypt (encryption) and NTRU-HRSS-KEM, which were separate submissions in Round 1. NTRU's underlying ciphers were patented but later released to the public domain in 2013.

In practical cryptography terms, NTRU uses lattices with more "structure" than the average lattice, which allows it to encrypt and generate secure keys significantly faster than traditional public key systems, such as RSA and ECC, and is faster than most Round 1 submissions (although not all).

For the beginning of Round 2 for the NIST required security implementations, secret key sizes range from 935 to 1,590 bytes; public key and ciphertext sizes range from 699 to 1,230 bytes. NTRU was submitted to the NIST contest by a multinational team. It is part of the Open Quantum Safe project.

For more information on NTRU, download https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/submissions/NTRU-Round2.zip.

NTRU Prime

NTRU Prime, a public key cryptosystem, was created as an expert "tweaked" version of NTRU to add more protections to what the NTRU Prime team calls "NTRU Classic." The NTRU Prime team discussed (https://ntruprime.cr.yp.to/ntruprime-20170816.pdf) all the possible security issues with lattice-based cryptography and NTRU Classic, and then used different types of rings. The NTRU Prime team describes their cipher as "efficient implementation of high-security prime-degree large-Galois-group inert-modulus ideal-lattice-based cryptography" and which others describe as using "irreducible, non-cyclotomic polynomials." Either description is fairly alien to most non-math majors, and explaining requires far more math than is appropriate for this discussion. For the beginning of Round 2 for the NIST required security implementations, secret key sizes range from 1,518 to 1,999 bytes, public key sizes range from 994 to 1,322 bytes, and ciphertexts from 897 to 1,312 bytes.

NTRU Prime reduces some of the more obvious weaknesses of NTRU Classic, eliminates decryption failures, and does so in constant time to mitigate some timing side-channel attacks. Even though the NTRU Classic team submitted an "improved version" of NTRU Classic, the NTRU Prime team still warns of completely trusting any lattice-based cryptography, including itself. NTRU Prime was submitted to NIST by a multinational team, including Daniel Bernstein.

For more information on NTRU Prime, see https://ntruprime.cr.yp.to.

NTS-KEM

NTS-KEM is a code-based KEM variant of the McEliece and Niederreiter public key encryption schemes. Like many code-based ciphers, it requires large public key sizes, but unlike the earlier ones, it is able to achieve CCA-secure indistinguishability. NTS-KEM uses SHA3-256. For the beginning of Round 2 for the NIST required security implementations, secret keys range in size from 9,248 to 19,922 bytes, public keys range from 319,488 to 1,419,704 bytes, and ciphertexts range from 128 to 253 bytes. NTS-KEM has the largest public key and smallest ciphertext sizes of all the competitors. Only Classic McEliece is close.

The team is U.K.-based and had applied for patents in the U.K. and U.S. before abandoning them to participate in the contest. The current version is not constant-time. A team led by Daniel Bernstein has been arguing why Classic McEliece is better than NTS-KEM with a rebuttal from the NTS-KEM developers: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/official-comments/Classic-McEliece-official-comment.pdf.

For more information on NTS-KEM, see https://nts-kem.io.

ROLLO

ROLLO (Rank-Ouroboros, LAKE, and LOCKER) is a low-rank parity check (LRPC) code-based cipher group based on a merger of three other code-based ciphers from NIST Round 1: LAKE, LOCKER, and Ouroboros-R. LAKE (now named ROLLO-I) is a CPA-secure KEM, LOCKER (ROLLO-II) is a CCA-secure PKE (public key encryption), and Rank-Ouroboros (ROLLO-III) is a KEM.

LRPC is a relatively new type of coding based on rank metric, which offers fast performance and smaller key sizes. To learn more about LRPC, see `https://pdfs.semanticscholar.org/d791/016d78b1054ce6c756a55ac78909ede25fdb.pdf`. ROLLO ciphers are fast with smaller key sizes. For the beginning of Round 2 for the NIST required security implementations, secret keys are always 40 bytes, and public keys and ciphertext range from 465 to 947 bytes. ROLLO has some of the smallest key and ciphertext sizes of all competitors, along with SIKE. On the negative side, ciphers based on LRPC codes have not been well studied and are not as trusted as other mechanisms. ROLLO was submitted by a French team.

For more information on Rollo, see `https://pqc-rollo.org/`.

Round5

Round5 is a group of lattice-based ciphers relying on the general-learning-with-rounding (GLWR) problem to unify the well-studied learning-with-rounding (LWR) and ring-learning-with-rounding (RLWR) lattice problems for its protection. It is a merger of two separate NIST first-round candidates: Round2 and Hila5. R5_CPA_KEM is a CPA-secure KEM, and R5_CCA_PKE is a CCA-secure public key encryption cipher. The cipher and their indistinguishability claims were a little surprising to some reviewers because most submitted post-quantum KEMs are usually CCA-secure (i.e., chosen cipher attack resistant) and not CPA-secure (chosen plaintext attack resistant).

Round5 has good performance with low bandwidth as compared to other LWR- and RLWR-based ciphers. Both key sizes and ciphertexts are short. For the beginning of Round 2 for the NIST required security implementations, secret keys range from 16 to 32 bytes, public keys range from 634 to 1,117 bytes, and ciphertexts range from 682 to 1,274 bytes. Round5 has the fourth smallest combined size (after ROLLO, LAC, and SIKE). The Round5 team mostly hails from the Netherlands and the company Philips, with one U.K. and one U.S. member.

For more information on Round5, see `https://round5.org`.

RQC

RQC (Rank Quasi-Cycle) is a code-based post-quantum public key encryption cipher based on the hardness of solving quasi-cyclic rank syndrome decoding problems, which works with random rank codes. RQC uses Gabidulin codes, a generalization of well-known (within crypto and math circles) Reed-Solomon codes for decoding. Syndrome decoding is considered a well-understood, highly efficient way to decode errors found in a noise channel—or in layperson's terms, a good way to encode

and decode code-based cryptography. For the beginning of Round 2 for the NIST required security implementations, the private key is always 40 bytes, public key sizes range from 853 to 2,284 bytes, and ciphertexts range from 1,690 to 4,552 bytes. It has a low to zero failure rate.

The RQC team is French-based and the cipher was partially funded by the French DGA (the French Government Defense procurement office).

For more information on RQC, see `https://pqc-rqc.org`.

SABER

SABER is a lattice-based CPA-secure encryption and CCA-secure KEM suite whose protection relies on the hardness of solving module-learning-with-rounding (MLWR) problems. It offers three security levels:

- LightSABER: A post-quantum security level similar to AES-128
- SABER: A post-quantum security level similar to AES-192
- FireSABER: A post-quantum security level similar to AES-256

For the beginning of Round 2 for the NIST required security implementations, secret key sizes range from 832 to 1,664 bits, public key sizes range from 672 to 1,312 bytes, and ciphertexts range from 736 to 1,472 bytes. It has good performance, flexibility, and low bandwidth, with less randomness required to be secure. SABER's cipher cannot be used to do digital signing. SABER's submission team is based in Belgium.

For more information on SABER, see `www.esat.kuleuven.be/cosic/pqcrypto/saber/`.

SIKE

Supersingular Isogeny Key Encapsulation (SIKE) is the only isogeny-based cipher (suite) submitted to the NIST contest. SIKE.PKE is a CCA-secure, public key encryption scheme, and SIKE.KEM is a CPA-secure KEM. SIKE is based on an isogeny key-exchange construction known as *supersingular isogeny Diffie–Hellman* (SIDH). For the beginning of Round 2 for the NIST required security implementations, it has private key sizes ranging from 44 to 80 bytes, public keys ranging from 330 to 564 bytes, and ciphertext ranging from 346 to 596 bytes. SIKE has the smallest public key sizes and near smallest private key sizes. If you consider all three sizes combined together, SIKE has the smallest combined sizes of any competitor.

Isogeny ciphers are relatively new and less studied than other cipher types, although SIKE's creators are quick to point out that studies of isogeny computations between elliptic curves (over finite fields) have been studied since the 1990s. In general, isogeny ciphers have great potential with relatively small key sizes, but security and performance, especially, are still hotly debated. You can get a taste of the discussions around SIKE and other isogeny ciphers by reading the NIST SIKE reviewer comments (`https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/official-comments/SIKE-official-comment.pdf`). If supersingular isogeny

cryptography holds up to sustained attack methods over time, it has a great chance of being one of the more popular future quantum-resistant ciphers. SIKE has a multinational submission team and is part of the Open Quantum Safe project.

For more information on SIKE, see `https://sike.org`.

ThreeBears

ThreeBears is a lattice-based asymmetric public key exchange cipher built using a variant of MLWE. ThreeBears provides a mode that is secure against CPA only, and another that is secure against both CPA and CCA. The underlying mathematical ring uses what is known as a pseudo-Mersenne prime. A *Mersenne prime* is a prime number that is one less than a power of two (i.e., $x^2 - 1$), named after a French mathematician. They are often used in traditional elliptic curve cryptography as well. For more information on Mersenne primes, see `https://en.wikipedia.org/wiki/Mersenne_prime`. *Pseudo-Mersenne primes* are Mersenne numbers with an additional trait that the subtraction component can be any small number greater than 0 (and not just a 1). Per its creator, ThreeBears was named because its pseudo-Mersenne prime has the same mathematical structure as the one used by a previous cipher known as Goldilocks, and it has three different parameter sets named BabyBear, MamaBear, and PapaBear.

ThreeBears also relies on error correcting codes, is fairly fast, and has among the lowest number of expected key exchange failures as compared to its competitors. For the beginning of Round 2 for the NIST required security implementations, secret keys are always 40 bytes (tied for second smallest with three other ciphers) and are used as seed values. Public key sizes range from 804 to 1,584 bytes and ciphertexts from 917 to 1697 bytes.

The biggest unknown with ThreeBears is whether lattices based on pseudo-Mersenne primes are more or less resilient to cipher attacks than other lattice-based ciphers, a fact that its creator, Mike Hamburg of Stanford and Harvard universities, acknowledges. ThreeBears is supported by Rambus, Inc.

For more information on ThreeBears, see `https://shiftleft.org/papers/threebears/` or `www.shiftleft.org/papers/threebears/threebears-spec.pdf`.

One or more of these official second-round NIST candidates are likely to become the U.S. post-quantum asymmetric encryption standard(s) in the next few years. Table 6.3 summarizes the various post-quantum asymmetric public key exchange and KEMs and their key and ciphertext sizes for the most popular NIST security-level classification submissions (i.e., Levels 1, 3, and 5).

Reported values are selected samples taken from the cipher's NIST submission document and/or confirmed by a member of the submission team. They may reflect only one or more versions of the cipher even if more versions are reported. They may not reflect the largest or smallest values for a particular field for a particular version of the cipher but are selected to be at least fairly representative of the other possible values. Extreme values of some versions may not be adequately represented. The secret key sizes may include the public key sizes in some instances—in other words, they may not have been obviously broken out in the documentation.

Table 6.3: NIST Round 2 PKEs and KEMs and key and ciphertext sizes by NIST security classification levels 1, 3, and 5.

	Eq. AES-128 NIST 1			Eq. AES-192 NIST 3			Eq. AES-256 NIST 5		
Algorithm	**SK**	**PK**	**CT**	**SK**	**PK**	**CT**	**SK**	**PK**	**CT**
BIKE	1988	20,326	20,326	3090	39,706	39,706	4110	65,498	65,498
Classic McEliece	6452	261,120	128	13,568	524,160	188	13,892	1,044,992	240
CRYSTAL-Kyber	1632	800	736	2400	1184	1088	3168	1568	1568
FrodoKEM	19,888	9616	9720	31,296	15,632	15,744	43,088	21,520	21,632
HQC	40	3125	6234	40	5884	11,749	40	7989	16,984
LAC	512	544	712	1024	1056	1188	1024	1056	1424
LEDAcrypt	452	1872	1872	644	3216	3216	764	4616	4616
NewHope	1888	928	1120	-	-	-	3680	1824	2208
NTRU	935	699	699	1234	930	930	1590	1230	1230
NTRU Prime	-	-	-	1763	1158	1039	-	-	-
NTS-KEM	9248	319,488	128	17,556	929,760	162	19,992	1,419,704	253
ROLLO	40	465	465	40	590	590	40	947	947
Round5	16	634	682	24	909	981	32	1178	1274
RQC	40	853	1690	40	1391	2766	40	2284	4552
SABER	832	672	736	1248	992	1088	1664	1312	1472
SIKE	44	330	346	62	462	486	80	564	596
ThreeBears	-	-	-	-	-	-	40	1584	1697

Legend: SK = secret key size, PK = public key size, CT = ciphertext; all values are bytes.
Note: Figures represent particular cipher implementations in each cipher suite.

General Observations on PKE and KEM Key and Ciphertext Sizes

Here are some general comparative observations about various PKE and KEM key and ciphertext sizes as shown in Table 6.3.

- No particular algorithm cipher type (code, lattice, multivariate) as a class proved to consistently have the largest or smallest sizes. There were almost always representations of each class size in the smallest and largest sizes, with most in the middle. This goes against what many theoretical discussions predicted, with various types of ciphers having consistently larger or smaller key sizes. Definite patterns emerged that support the theoretical arguments, but with enough exceptions that it doesn't make the assumptions a practical rule.
- The smallest secret and public keys (and second smallest ciphertexts) were SIKE, followed by ROLLO.
- The largest secret keys are from Classic McEliece, FRODO-KEM, and NTS-KEM.
- Four ciphers (BIKE, Classic McEliece, FRODO-KEM, and NTS-KEM) consistently had the very largest public key sizes.
- Classic McEliece and NTS-KEM public key sizes were off the charts and in a class by themselves as compared to the other 15 ciphers, but they also had the smallest ciphertexts (followed closely by SIKE and ROLLO).
- The largest ciphertexts were from BIKE, FRODO-KEM, and HQC.
- The smallest combined key and ciphertext sizes were from SIKE, followed by ROLLO and then Round5.

A great online NIST post-quantum key comparison web page is `https://pqc-wiki.fau.edu/w/Special:DatabaseHome`.

Each algorithm is also being reviewed for many other characteristics beyond key and ciphertext sizes, including

- Performance (in both software and hardware implementations)
- Storage sizes (runtime and on media)
- Key generation
- Encryption speed
- Decryption speed
- Complexity
- Ease of implementation
- Failure rates
- Ability to provide security protection

All of these factors, and more, are being reviewed by stakeholders for each submitted cipher. The ciphers that best handle these factors, along with sustained security resilience, will progress to Round 3 and/or eventually be considered for the NIST post-quantum standard cipher selection.

Quantum-Resistant Digital Signatures

Quantum-resistant digital signature schemes are cryptographic digital signatures that are not overly susceptible to quantum computers running quantum-based algorithms. They do not use quantum-based properties to defeat attacks. There are over a dozen quantum-resistant digital signature schemes, although one or more of the nine Round 2 NIST asymmetric candidates are likely to be the eventual NIST federal standard. The NIST second-round digital candidates, in alphabetical order, are:

- CRYSTALS-Dilithium
- FALCON
- GeMSS
- LUOV
- MQDSS
- Picnic
- qTESLA
- Rainbow
- SPHINCS+

NOTE There are at least three other major quantum-resistant digital signature schemes that did not meet NIST submission criteria: Leighton-Micali Signatures (LMS), eXtended Merkle Signature Scheme-MT (XMSS), and Blockchained Post-Quantum Signatures (BPQS). At least the first two are stateful and can cause problems if not handled correctly during data restoration activities, and BPQS uses a relatively untested hybrid approach, which it calls a bridge between stateful and stateless. You can read more about XMSS and LMS at `https://eprint.iacr.org/2017/349.pdf` and about BPQS at `https://eprint.iacr.org/2018/658.pdf`.

CRYSTALS-Dilithium

CRYSTALS (Cryptographic Suite for Algebraic Lattices) encompasses two lattice-based cryptographic primitives: Kyber, a CCA-secure KEM, and Dilithium (covered earlier), an EUF-CMA-strongly-secure digital signature algorithm, CRYSTALS-Dilithium.

CRYSTALS-Dilithium is based on MLWE, which the authors state can be thought of lattices between unstructured LWE and structured RLWE. It also uses an interactive proof-of-knowledge idea known as *Fiat-Shamir with Aborts* (`https://www.iacr.org/archive/asiacrypt2009/59120596/59120596.pdf`), which is similar to but different from ZPK systems mentioned earlier. In an *interactive proof-of-knowledge* system, the prover doesn't prove to the verifier that it knows the value of *x*. A third process known as the *knowledge extractor* does the proving to the verifier. See `https://en.wikipedia.org/wiki/Proof_of_knowledge` for more details on interactive proof-of-knowledge theories and systems.

Dilithium creates relatively small digital signatures and public keys while providing AES-128-level or larger security. For the beginning of Round 2 for the NIST required security implementations, secret key sizes of 64 bytes for all implementations (NIST Security Levels 1, 2, and 3), public key sizes range from 1,184 to 1,760 bytes, and signature sizes range from 2,044 to 3,366 bytes. Like other quantum-resistant digital signatures, Dilithium starts with a small private key, 64 bits in this case, which is just a seed value that gets pseudo-randomized into another value, which the algorithm then uses to generate the public key and digital signature.

When looking at the size of quantum-resistant digital signatures and not any other characteristic, the sizes that mean the most for comparison purposes are the public key and the resulting digital signature (and their overall combined sizes). The "private key" can often be increased and decreased. Increasing the size of the private key, the original seed value version or the eventually computed larger value actually used to do the real work, would normally decrease performance at least slightly, and vice versa. Implementers can normally, if they choose, increase or decrease the key and signature values to make their own security versus performance trade-off.

NIST instructed submitters to choose particular values to meet the different NIST security level requirements. Teams sometimes slightly modified their algorithm or values to get improved size/performance characteristics. It's part of why NIST has a "contest." The competition improves many of the algorithms, or at least has them being as thoughtful as possible about their particular suggested implementations.

Interestingly, after CRYSTALS-Dilithium was submitted to NIST in Round 1, reviewers found a weakness, which turned out to be a simple two-line coding transposition error in the CRYSTALS-Dilithium RNG, a bug that the implementers acknowledged and fixed in a quick update. Giving the public and other teams a chance to review all algorithms, if they choose, can only help improve all algorithms. Good cryptographic algorithms sustain and improve under public review. Don't trust cryptographic creators who keep their algorithms private and don't allow the world to review and test. It's never a good sign of security. The Dilithium team is multinational and includes members from IBM and Google. It is part of the Open Quantum Safe project.

For more information on CRYSTALS, see `https://pq-crystals.org/` and `https://pq-crystals.org/dilithium/index.shtml`.

FALCON

FALCON (FAst fourier Lattice-based COmpact signatures over NTRU) is an NTRU lattice-based digital signature algorithm based on ring short integer solution (SIS) problems (as is qTESLA). SIS problems are very hard to solve, although most lattice-based cryptography uses what is known as shortest vector problems (SVPs). FALCON is also based on a 2008 work that led to a generic framework called the Gentry, Peikert, and Vaikuntanathan (GPV) framework for building secure hash-and-sign lattice-based signature schemes. It also uses floating-point arithmetic with 53 bits of precision.

Dilithium and qTESLA rely on the Fiat-Shamir paradigm, whereas FALCON uses a competing "hash-then-sign" paradigm. Algorithms based on the former rely on proof-of-knowledge systems and can have problems securely signing long messages. Hash-then-sign algorithms overcome the issue

by first hashing the message (and getting a much shorter hash result) and signing the hash result instead of the message.

NOTE CRYSTALS-Dilithium is based on Module-SIS, which is similar to but different from Ring-SIS. Essentially, they are all very hard math problems to solve, but some are harder than others. If you are interested in the mathematical and work effort differences, you can read https://eprint.iacr.org/2012/090.

FALCON was specifically designed to have good performance, especially in low memory environments. FALCON's creators intentionally positioned their algorithm to be a strong contender for smallest public key and signature size, but it does use floating point math (which decreases overall performance on platforms that don't inherently support that type of math).

For the beginning of Round 2 for the NIST required security implementations (FALCON submitted to NIST Levels 1 and 5 only), FALCON secret key sizes range from 1,280 to 2,304 bytes, public keys range from 897 to 1,793 bytes, and digital signatures range from 617 to 1,233 bytes. According to its creators, FALCON-512 with a public key size of 897 bytes and a digital signature of 617 bytes has security equivalent to RSA 2,048 bits (which has public keys and signatures of 256 bytes). FALCON's team is multinational, and Thomas Pornin is the primary creator.

For more information on FALCON, see https://falcon-sign.info.

GeMSS

GeMSS (Great Multivariate Signature Scheme) is an EUF-CMA-secure multivariate-based signature scheme producing fairly small signatures for post-quantum signing (258 to 576 bits, not bytes, long). For the beginning of Round 2 for the NIST required security implementations, it uses medium to large keys (public keys ranging in size from 352 to 3,041 kilobytes and secret keys ranging from 13 to 76 kilobytes). This gives GeMSS the smallest signatures and one of the two largest public key sizes of all competitors (Rainbow shares similar characteristics).

The signature creation process is fairly slow, but the verification of signatures is fast. GeMSS was built off an older multivariate signature scheme known as QUARTZ, and it uses vHidden Field Equations (-vHFE) by using "minus" and "vinegar" modifiers. You can read more about HFE and its modifiers here: https://en.wikipedia.org/wiki/Hidden_Field_Equations. GeMSS was created by a French-based submission team as part of a French national project.

For more information on GeMSS, see www-polsys.lip6.fr/Links/NIST/GeMSS.html.

LUOV

LUOV (Lifted Unbalanced Oil & Vinegar) is a multivariate public digital signature scheme that is based on Unbalanced Oil & Vinegar (UOV). UOV produces large key sizes and uses non-unique keys. This can be perplexing to anyone who has studied traditional cryptography. With traditional

asymmetric cryptography, every public-private key pair is unique to each other. With UOV, along with other multivariate algorithms, a single public key can end up with millions of different private keys. The relationship is not 1:1.

LUOV uses a modified version of UOV with an efficient secret key (32 bytes long) to allow smaller public keys and improved performance. LUOV's secret keys are on the smaller side. For the beginning of Round 2 for the NIST required security implementations, LUOV has public key sizes ranging from 12 to 76 kilobits and digital signatures ranging from 311 to 494 bytes, and it uses a pseudo-random number generator (PRNG). However, the team did not submit suggested implementations for NIST security level 1, 3, or 5, making it the only team to do so. The new, "lifted" method it is using for its protection has not being widely evaluated and security tested, although the underlying UOV has been widely reviewed (at least since 1996). LUOV is supported by a Belgian team.

For more information on LUOV, see `www.esat.kuleuven.be/cosic/pqcrypto/luov/`.

MQDSS

MQDSS (Multivariate Quadratic Digital Signature Scheme) is a multivariate digital signature scheme that uses a generalized Fiat-Shamir transform (as does CRYSTAL-Dilithium) and the 5-Pass Sakumoto, Shirai, and Hiwatari (SSH) identification scheme. It is the first multivariate digital signature that is *provably secure* relying only on the hardness of solving multivariate quadratic equations for its protection.

For the NIST Round 2 security submissions (Levels 1 to 4), it has extremely small public and private key sizes, ranging from 46 to 64 bytes and from 16 to 24 bytes, respectively (they did not submit a Level 5 instance). That's very small, tying for first or second smallest among all competitors. Unfortunately, it gives huge digital signatures, ranging from about 20 to 43 kilobytes. That's huge, tying for second largest of all competitors, and that's even after changes between Round 1 and Round 2 to double performance and halve signature sizes. It is inherently constant-time. MQDSS has promise but needs more research, testing, and optimization. MQDSS has a multinational team makeup.

For more information on MQDSS, see `http://mqdss.org`.

Picnic

The Picnic family of digital signature algorithms are the only NIST submission to use zero-knowledge proofs, which provides the proof with a single message. Per the Picnic team, it "does not rely on number-theoretic or algebraic hardness assumptions." Like MQDSS and CRYSTALS-Dilithium, Picnic relies on the Fiat-Shamir transform model for two of its variants (Picnic-FS and Picnic2) and the Unruh transform for a third (Picnic-UR). Picnic uses SHAKE. For the beginning of Round 2 for the NIST required security implementations, it creates small public and private key sizes (32 to 64 bytes and 16 to 32 bytes, respectively) but with much larger digital signatures (32 to 125 kilobytes) along with slower performance.

Picnic has been tested with TLS and x.509 digital certificates. It is part of the Open Quantum Safe project. The team modified OpenSSL (the world's most popular open-source cryptography program) to use Picnic, along with Picnic-based digital certificates (using Picnic-based keys and signatures). They used Picnic to establish TLS 1.2 connections to Apache web servers, perhaps the first publicly announced post-quantum algorithm to do so. OpenSSL had to be modified to accept and use the larger key sizes associated with Picnic. The team noticed that the TLS standard supports key sizes of only 65,535 bytes, so it will have to be updated to more easily support post-quantum algorithms. The team also tested Picnic working with secure hardware security module (HSM) devices connected to an example PKI application, and it was successful. This proved that post-quantum cryptography can be used, today, in real-world scenarios, with only minor modifications needed. Picnic was designed by a multinational team, including Microsoft researchers.

For more information about Picnic, see `https://microsoft.github.io/Picnic/`.

qTESLA

qTESLA is based on a number of previous schemes of the TESLA family, including BG-scheme, TESLA, ring-TESLA, and TESLA#. It is an EUF-CMA-secure, RLWE lattice-based digital signature scheme with two main primitive variants: *provably secure* (for high security needs) and *heuristics* (for better performance). Like other post-quantum cryptography, it relies on the Fiat-Shamir with Aborts transform (from 2012), and it is also based on a more efficient variant of the Bai-Galbraith signature scheme (originally announced in 2014). It is constant-time and has relatively average key sizes and signatures. For the NIST Round 2 required security implementations, public key sizes range from 1,504 to 6,432 bytes, secret keys range from 1,216 to 4,672 bytes, and digital signatures range from 1,376 to 5,920 bytes. qTESLA is part of the Open Quantum Safe project.

During the NIST review process, a weakness was revealed (see `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/official-comments/qTESLA-round2-official-comment.pdf`) and fixed in a subsequent version, although there is still an ongoing discussion about whether it is truly fixed. These types of tough, "forged through fire" debates are good for cryptography and for the eventual winner, whichever it may be. qTESLA has a multinational team, including Microsoft researchers (`www.microsoft.com/en-us/research/project/qtesla/`).

For more information on qTESLA, see `www.qtesla.org`.

Rainbow

Rainbow is an EUF-CMA-secure multivariate digital signature algorithm using a multilayered implementation of Unbalanced Oil & Vinegar. The proposed algorithm contains a few variants maximized for performance, size, or security. It uses the SHA-2 hashing algorithm from 256 to 512 bits depending on the security classification.

Signature generation is very fast and signature sizes are short. For the beginning of Round 2 for the NIST required security implementations, private keys range from 93 to 1,227 kilobytes, public keys range from 149 to 1,705 kilobytes, and signatures range from 512 to 1,632 bits (not bytes).

Like GeMSS, this gives among the smallest signatures of all competitors, but also among the largest secret and public keys. Key sizes can be made smaller but doing so would decrease overall performance.

Security-wise, Rainbow is among the more tested post-quantum algorithms. It was created in 2005, and the last successful attack that required a code change happened in 2008. That's more than 10 years without a better attack launched against it. With that said, researchers are continuing to mine the algorithm for weaknesses and attack surfaces, including this 2018 paper: `www.hindawi.com/journals/scn/2018/2369507/`. Rainbow was created and submitted by a multinational team.

For more information, download `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/submissions/Rainbow-Round2.zip`.

SPHINCS+

SPHINCS+ is a stateless hash-based digital signature scheme with three variants. It is based on an improved version of SPHINCS, introduced in 2015. The improvements concentrated on reducing resulting digital signature sizes. It's a flexible framework with over 36 combinations of variants including

- SPHINCS+-SHAKE256
- SPHINCS+-SHA-256
- SPHINCS+-Haraka

SPHINCS+ is based on long known and used hash-based digital signatures first created in the late 1970s (along with the first asymmetric ciphers). SPHINCS+ is a quantum-resistant improvement submitted in 2017 as the first quantum-resistant digital signature scheme. SPHINCS+ is stateless, which is important.

The creators of SPHINCS+ did this by having a top-level, unchanging XMSS-based public-private key pair that signs and verifies other, lower pseudo-random key pairs that do the signing. The creators call this a *hypertree* (as compared to a standard Merkle tree) based on the use of a few-time signature method at the bottom of the hypertree (which is what distinguishes it from a stateful hash-based signature). In the root node of the hypertree, the single, reused private key is used as sort of a seed value that is then used by a pseudo-random function to generate the lower node key pairs.

SPHINCS+ uses SHA256, SHAKE256, or Haraka, and it comes in small-signature and faster versions. The faster versions have larger signatures for the same key sizes. For the beginning of Round 2 for the NIST required security implementations, public keys range from 16 to 32 bytes, private keys are 64 bytes, and the digital signatures range from 8,080 to 49,216 bytes. This makes SPHINCS+ have some of the longest signatures of the competition.

SPHINCS+ was conservatively designed on purpose. Because it is hash-based, it is not susceptible to Shor's algorithm and mostly has to worry about Grover's algorithm advances. The only major concern is a cryptographical attack on the hash function itself (which is true for any signature method that uses an initial hash of the message). Unfortunately, the SPHINCS+ hash-based approach also means it is relatively slow compared to most of the competitors and generates longer digital signatures.

SPHINCS+ creation was funded by the European Commission through its Information and Communication Technologies (ICT) program and a U.S. National Science Foundation grant. SPHINCS+ was created by a multinational team, including lead researcher Andreas Hülsing and Daniel J. Bernstein.

For more information on SPHINCS+, see `https://sphincs.org`.

As you can see, there are a lot of quantum-resistant algorithms making it to Round 2 of the NIST evaluation process. Table 6.4 summarizes the various digital signature schemes along with their key and signature sizes.

Reported values are selected samples taken from the team's NIST submission document and may reflect one or more versions of the algorithm even if more versions are reported. They may not reflect the largest or smallest values for a particular scheme, but they were selected to be at least fairly representative of the other possible values. Extreme values of some versions may not be adequately represented.

General Observations on Signature Key and Sizes

Here are some general comparative observations about various digital signature key and signature sizes as shown in Table 6.4.

- There is no particular algorithm scheme type (hash, lattice, multivariate, or ZKP) as a class that proved to consistently have the largest or smallest sizes. There were almost always representations of each class size in the smallest and largest sizes, with most in the middle.
- The smallest secret keys are from LUOV and Picnic, closely followed by SPHINCS+. Picnic and SPHINCS+ also had the smallest public key sizes.
- The largest secret and public keys, by far, are from GeMSS and Rainbow, but they also produced the smallest signatures by far (the only ones measured in bits versus bytes).
- The largest signatures are from Picnic, followed by SPHINCS+ and MQDSS.

Each algorithm is also being reviewed for many other characteristics including

- Performance (in both software and hardware implementations)
- Storage sizes (runtime and on media)
- Key generation
- Encryption speed
- Decryption speed
- Complexity
- Ease of implementation
- Failure rates
- Ability to provide security protection

Table 6.4: NIST Round 2 digital signature algorithm key sizes by NIST security classification

	Eq. AES-128 NIST 1			Eq. AES-192 NIST 3			Eq. AES-256 NIST 5		
Algorithm	**SK**	**PK**	**Sig**	**SK**	**PK**	**Sig**	**SK**	**PK**	**Sig**
CRYSTALS-Dilithium	64	1184	2044	64	1760	3366	-	-	-
FALCON	1280	897	617	-	-	-	2304	1793	1233
GeMSS	13K	352K	258b	34K	1238K	411b	76K	3,041K	576b
LUOV	-	-	-	-	-	-	32	75K	494
MQDSS	16	46	20854	24	64	43728	-	-	-
Picnic	16	32	32,838	16	48	74,134	32	64	128,176
qTESLA (Heuristic)	1216	1504	1376	2368	3104	2848	4672	6432	5920
Rainbow (cyclical)	93K	149K	512b	511K	711K	1,248b	1,227K	1706K	1,632b
SPHINCS+ (small)	64	16	8080	64	24	17,064	64	32	29,792

Legend: SK = secret key size, PK = public key size, Sig = signature size.
Note 1: Sizes in bytes unless otherwise stated. K = Kilobytes, b = bits.
Note 2: Figures only accurate for particular algorithm implementations in each cryptographic suite.

All of these factors, and more, are being reviewed by stakeholders for each submitted algorithm. The cryptography that best handles these factors will progress to Round 3 and/or eventually be considered for the NIST post-quantum digital signature standard selection.

Caution Advised

Post-quantum cryptography is necessary in a world where much of our traditional cryptography can be broken. Our protection will be gained by increasing the key size of traditional ciphers and implementing both quantum-resistant and quantum-based cryptography. We will be using quantum-resistant cryptography first followed by quantum-based cryptography.

It's the nature of the security battle we are fighting. We will not have widespread, cheap, and available quantum cryptography until we have enough quantum computers and processing that vendors can build cheap quantum computers and customers can afford to buy them in mass. When that happens, we will likely all be using quantum-based cryptography.

Before that happens, though, there will be enough quantum computers and processors available to well-monied adversaries to attack our traditional quantum-susceptible cryptography. Using quantum-resistant cryptography is our bridge from now to the future. We are going to be forced to implement quantum-resistant cryptography as an intermediate defense.

However, there are a few significant reasons why people should not simply rush headlong into quantum-resistant (or quantum-based) cryptography prematurely. Three of the main issues are a lack of standards, performance concerns, and a lack of verified protection.

A Lack of Standards

This chapter has focused on NIST and the world's attempt to select post-quantum cryptographic standards. There are currently 26 cryptographic proposals under review for the U.S. standard, and only two (or a few more) will win. If you choose to implement a particular quantum-resistant algorithm now, there is a greater-than-normal chance you will *not* pick the one that becomes the new standard.

If you select incorrectly, you can always switch to the new standards once they become known or stay with your personally selected (nonstandard) implementation. History shows that the latter choice is very inefficient and significantly increases your security and/or operational risk. Choosing to forge ahead with a nonstandard algorithm can be risky, because there are often strong arguments about a particular algorithm (such as its purported security protection or performance issues) as to why it did not get selected as the eventual standard.

Moving from a prematurely selected post-quantum algorithm to the new standards is certainly more acceptable as a path forward but likely increases overall costs. You should start experimenting with implementing quantum-resistant cryptography, but be careful about widespread, full-production deployments. You don't want to spend too much money heading down a path that may not be the right one. A better choice is to begin to do limited experimentation and deployments with a few

trusted quantum-resistant algorithms, and make sure the products you are now buying are *crypto-agile*. Crypto-agile means that whatever existing cryptographic algorithm it uses now can be easily replaced with another as needed. More on this in Chapter 9, "Preparing Now."

Performance Concerns

Even if a quantum-resistant cryptographic standard has smaller key sizes, the workload efforts needed to create and verify keys is often much larger than with traditional cryptography. This is why the NIST contest requires lots of performance testing and submitters are trying their best to optimize the speed of their algorithm. NIST will probably be picking a post-quantum standard that has a good performance/security trade-off, but moving to a post-quantum algorithm is likely to decrease overall performance even on the best and fastest computers and devices. Moving wholesale to a quantum-resistant algorithm in a production environment or product must be done after careful consideration.

Lack of Verified Protection

Most importantly, most quantum-resistant cryptographic algorithms are relatively new and untested over long periods of time. There are a few exceptions, but most quantum-resistant algorithms are not even completely trusted as forever secure by their creators. Most involve new complex math that currently appears nontrivial to unbreakable. But all it takes is a new type of attack or a new algorithm to bring the security protection that quantum-resistant algorithms offer crashing down.

This is especially true if you look at today's modern cryptography and use its history as a guide. Symmetric key encryption using 128-bit keys was deemed very strong just a few years ago, but now quantum computers paired with Grover's algorithm are halving their protection. Shor's algorithm is getting ready to gut most of today's public key encryption. And there are a host of newer algorithms created since Shor's was published that claim to be better at prime cracking than Shor's. That's evolution. That's progress. The one truism about crypto attacks is that they only get better over time and the cryptography they attack only gets weaker.

We have no way of knowing if quantum-resistant cryptography is uncrackable for the foreseeable future. We won't know until decades have passed, in which each of the algorithms has undergone attack after attack and continues to survive. We have no way of knowing when the next Shor's or Grover's algorithmic breakthrough will be coming, but there are very likely to be further breaks. And then one day, our quantum-resistant algorithms will be weakened and fall—just like SHA1, MD5, DES, and a multitude of cryptography before them. Far more risks and vulnerabilities are likely to be discovered in the many real-world implementations of otherwise strong algorithms. It's the nature of humans that we rarely write bug-free code. When anyone tells you something is unhackable, they are always wrong.

With this said, we have no other better alternatives than to trust that quantum-resistant cryptography will give us more protection than traditional cryptography and will give us enough protection long enough until we make the full, long-term transition to quantum-based cryptography. Just be

aware that nothing, not even quantum-resistant cryptography, is unhackable or bug proof. It's a risk that we all have to accept until something else better comes along—just like we did with all the traditional, modern-day cryptography we rely on today.

For Additional Information

Here's a great 2009 paper by Daniel Bernstein on post-quantum cryptography: `https://pqcrypto.org/www.springer.com/cda/content/document/cda_downloaddocument/9783540887010-c1.pdf`.

For a great discussion on post-quantum cryptography and PKI, visit `www.primekey.com/wp-content/uploads/2017/08/post-quantum-algorithms-for-pki.pdf`.

Summary

This chapter covered quantum-resistant cryptography and summarized the 26 cryptographic algorithms that advanced to Round 2 in NIST's post-quantum cryptography standardization contest. It explored the different types of quantum-resistant algorithms, along with their strengths, weaknesses, and key sizes. At least two or more of these algorithms (one for public key encryption and one for digital signing) will likely become the new U.S. post-quantum cryptographic standard within the next few years. Chapter 7 will cover quantum-based cryptography.

7 Quantum Cryptography

Chapter 6 covered traditional, binary cryptography that is resistant to known quantum attacks. This chapter covers quantum-based cryptography, cryptography that exists and operates on quantum devices using quantum properties. Quantum-based cryptography is also inherently resistant to known quantum attacks, as well as attacks from traditional binary computers. Binary cryptography is an acceptable defense in a post-quantum world where not enough widely available, cheap quantum computing and networking devices exist. But quantum-based cryptography is theoretically much safer to all known attacks and will likely be the cryptographic choice for long-term security. In this chapter, we will cover the main types of quantum cryptography, including random number generators (RNGs), hashes, key distribution, and digital signatures. Chapter 8 will explore quantum networking.

Inherent in all of these quantum cryptography implementations are all the popular quantum mechanics properties. However, four particular quantum properties show up again and again as being central to how quantum cryptography provides its protective superpowers: superposition, entanglement, the observer effect, and the no-cloning theorem. Superposition gives more possible choices than what a binary digit can offer. Entanglement often provides the way for the involved cryptographic secrets to be transmitted between the authorized parties. The observer effect and the no-cloning theorem make it harder to accomplish undetectable eavesdropping. You'll learn more about these properties as we discuss the various quantum cryptographic implementations.

It is important to note that, because working quantum computers and devices are still relatively young, there are not hundreds of thousands of these systems. This is not to say that there aren't any or that some types do not exist in ever-growing quantities. In fact, there are a lot of existing quantum devices (i.e., quantum-based RNGs and key distribution systems likely number in the many thousands) that have been around for nearly two decades. But no one expects to see quantum-based cryptography at the same scale as its binary cousins for many, many years to come. Quantum-resistant crypto is sure to take hold first and be secure enough to bridge us to a time when we can be using only quantum-based cryptography. With that said, this chapter investigates the current state of

Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto, First Edition. Roger A. Grimes.

quantum-based crypto. Some types of quantum cryptography and devices are fairly mature and can be widely used today, whereas others are still so young and complex that it would take a rare, expensive use case for them to be considered today. We'll start by discussing RNGs, which are used in most cryptography regardless of type.

Quantum RNGs

As first covered in Chapter 5, "What Will a Post-Quantum World Look Like?," most cryptographic functions require strongly random numbers as a critical initial component of their algorithms. Without truly random numbers, any dependent algorithm or function is much less secure. In the traditional binary world, there is an inconvenient truth that most computer users do not know: a traditional binary computer cannot generate a truly random number. We can only get things that appear to be almost truly random.

Random Is Not Always Random

I learned this lesson—random is not always random—over two decades ago when I was the head of IT for a large conglomerate company that had a small business that drug-tested professional athletes in competitions. These athletes included professional tennis players and racecar drivers. Winners of events would always be tested, and everyone else would be tested on a random basis during the year. To randomly select which athletes would be chosen for a urine sample, the athletic association or competition would send the company a file each month with all the athletes' names and other identifying information, such as membership numbers. I would upload the file into the program and run the random sampling picker feature. The program would output which athletes had been randomly selected from that month's input file, and we would forward the results to the athletes' representative organization.

One day I was brought into an emergency meeting because a top driver had been "randomly" selected two months in a row. I was called in by senior management who was responding to the driver's complaint that he was being unfairly targeted. I looked at the program's coding (written in dBASE III+), saw no coding errors, and reported the same. Management did a big "dog and pony" show where the racecar organization's top officials and the driver's representatives heard from me, the resident computer expert, how the driver's duplicate selection was simply a factor of how real randomization works. With true randomness, every person is equally likely to get selected during each selection round, and the driver's sequential duplicate selection was just randomness being random. Everyone enjoyed learning how true randomness worked, the driver took his second drug test and passed, and everyone seemed glad to put the matter behind them. Then, the same driver got selected the following month.

The racecar organization and the driver never learned of this third pick. We realized we had a real problem. I reexamined the code and found no errors. But in desperation to find out what was going

on, I created a new program that contained just the identical randomization coding routine (as written in dBASE III+). Nothing more, nothing less. I used numbers 1 to 100 and told the program to run 10,000 times and to output how often any particular number was selected. In a truly random selection process, all numbers should have received around 1 percent of the attention, with some small deviations here and there. But when I got through running the program, a handful of numbers had been selected as often as 15 percent of the time, one that had over 20 percent popularity, and many dozens that had near zero percent. I was stunned. The random function of the software program was not even close to being truly random.

I decided to write a better program that did not rely on dBASE III+'s random function, which clearly had issues. This time around I used Microsoft Windows' RNG feature. I reran the same test, but this time with a thousand numbers and many tens of thousands of test rounds. Again, I was blown away; although Windows' RNG was better, patterns of over- and underselection still emerged. So, I wrote a program using assembly language (I had learned to disassemble computer viruses in the late 1980s) that used the computer's built-in RNG feature. I reran the test. And although it was better than Windows RNG for approximating randomness, it too wasn't completely random. I found definite patterns of favorites and avoidance, although they were far more subtle than the previous tests. That's when I learned that there is no such thing as true randomness in the computer world, but only pseudo-approximations of randomness. You'll see this fact pointed out by many cryptographers and cryptographic routines that use the term *pseudo-RNGs* (also known as *PRNGs*). They are acknowledging what we all now know is true: there is no true randomness on a binary computer.

Over the three decades since my own discovery, all the involved RNGs—a computer's built-in one, Microsoft Windows, and many other custom RNGs included with different programs—have been coded to be more approximate of true randomness. But if you were to do a test similar to the one I did in the early days, there would still be some infinitesimally small favorites and avoidance.

That's because traditional binary computers cannot do true randomness. All binary computers rely on one or more quartz crystals located on the motherboard and on other hardware for their operations and timing. Those quartz "clocks" vibrate (called *oscillation)* a certain and constant number of times per second (each individual time period is called a *clock tick* or *clock cycle*). Everything on a computer's motherboard runs off the timing signal sent by the motherboard's quartz clock oscillation cycle. Today's CPUs have their own internal clock oscillations, which control when a CPU can do something (e.g., move this bit into that CPU register, add this number and that number together, erase that value in that register). Each core in the CPU can do only one action at a time (unless you're doing parallel operations), and that operation can happen only on each tick of the CPU's clock. The CPU's timing clock is much faster than the motherboard's, but is usually an exact order of magnitude of the motherboard's clock cycle (e.g., for every motherboard cycle the CPU can do 100 times as many operations, each evenly dispersed over time). There can be other timing clocks as well, but the main motherboard and CPU clocks are the most important ones.

Everything the computer does originates from an evenly dispersed clock cycle, which happens a guaranteed number of evenly split times per second, which is the opposite of random. This lack of

randomness in the source of truth prevents anything relying on it from being truly random, no matter how much that upper dependent piece of hardware or software might try to approximate true randomness. The best hardware and software routines have what looks like very good approximations of true randomness, but they aren't truly random. The best we can do is get as close as possible to true randomness so that any resulting errors are minuscule to resulting dependent applications.

This is not to say that there aren't good and bad PRNGs, or that some aren't better than others. NIST created a series of tests that any—quantum or otherwise—RNG vendor or customer can run to see how good or bad their RNG is compared to a theoretically perfect random number generator. They documented the tests and requirements in NIST Special Publication 800-22 (`https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf`).

NOTE The official definition of an RNG is that its approximation of randomness must be *non-deterministic* (i.e., cannot be determined ahead of time) and a PRNG is *deterministic*. Interestingly, because PRNGs are deterministic they must begin with a seed value from an RNG, which we know can never be truly random on a binary computer. Ironically, a good PRNG can return a more random-looking number than the seed value given to it by an RNG on a binary computer.

Why Is True Randomness So Important?

Most cryptographic algorithms require a truly random number for the start of their algorithm (often known as the *initialization vector*, *seed value*, or *nonce*), which then uses hard math to produce a result that is truly hard to factor or guess. For example, RSA requires two large, randomly chosen primes to be selected as part of their algorithm (usually represented as p and q mathematically). Once the random prime numbers are chosen, they are then involved in math that produces a hard-to-factor back-out result.

But suppose the prime numbers, no matter how large, were not random at all. Suppose, because of some mistake, that the prime numbers selected each time were the same every time. This would be similar to having an algebraic formula like X + 23 = Z but you know that X is always 5. Having that knowledge, you would know whenever Z is 28 that X was 5 no matter how many times you ran the "algorithm." In this case, the result of RSA would always be the same if the primes were always the same. Any attacker learning this mistake, and seeing the same expected result, would immediately know what the prime numbers were that were used, and they would be able to immediately factor the result back to the original X and Z components. The cipher would have zero protection.

And if this scenario of zero randomness sounds a bit farfetched, it has happened many times in the computer world. People relying on what they were told were good, well-tested, solid ciphers to protect their confidential data later learned that the program implementing the cipher had a bug in its RNG, which completely invalidated the cipher's supposed protection. This has happened in widely used programs, protecting millions of websites and computers, more than once.

Debian OpenSSL RNG Bug

One of the most well-known examples of insecure RNG bugs is the Debian OpenSSL RNG debacle in 2006. Debian Linux is a very popular version of Linux, including a "distro" known as Ubuntu, which is often used by people trying to transition from Microsoft Windows to Linux for the first time. OpenSSL is the most popular open-source cryptography programming library and program used by open-source OS computers. When an updated version of OpenSSL was "forked" to Debian in 2006, one of the Debian OS developers incorrectly interpreted a compiler code warning message, removed a few lines of involved code, and accidentally removed almost all randomness.

The error was not caught until after many millions of nonrandom keys were issued and the bug was noticed in 2008. The removal of randomness meant that the possible keys went from many trillions of possible combinations to only 1 in 32,767 combinations at most. And in many instances, tens of thousands of implementations shared the exact same key pairs. Years later, tens of thousands of websites still contain the well-known and widely documented key pairs, and thousands still exist today (although most of the ones used on the Internet have now expired). Hackers even created tools that would check for and brute-force digital certificates and key pairs that relied on the bug; see `www.madirish.net/309`.

You can read more about the Debian RNG bug at these sites:

`www.schneier.com/blog/archives/2008/05/random_number_b.html`, `https://hdm.io/tools/debian-openssl/`, `https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0166`

`https://security.stackexchange.com/questions/143133/all-weak-debian-openssl-dsa-keys`

I don't want you to think that only one developer ever made such an error. There are many more examples of buggy RNGs throughout history here: `https://en.wikipedia.org/wiki/Random_number_generator_attack`. And that's not including the times governments and companies may have intentionally created buggy RNGs, as covered in Chapter 6, "Quantum-Resistant Cryptography." Confirming randomness is a desired need.

But here's the bigger problem. Even if the RNGs don't contain a bad coding bug, they are still not truly random. Deep down in their core they all look like the dBASEIII+ RNG example. They may look random, but when analyzed and scrutinized by the right people with deep resources, pockets of nonrandomness are revealed. That's because the source of truth on a binary computer is a quartz crystal clock, and that clock cycle on which every process in the computer runs is not a random occurrence. That lack of true randomness gives cryptographic attackers a "crib" to easily break the cryptography and other mechanisms that require truly random numbers for their protection.

To make matters worse, most quantum-resistant ciphers rely on binary RNGs or PRNGs. The resiliency of their protection is still incredibly high, but deep down in their bowels, they are relying on an inherently weak RNG. This dilemma is often addressed by the various quantum-resistant cipher creators as a design risk that they cannot completely control. Because of these RNG issues, unintentional or otherwise, it is important that randomly generated numbers be both truly random

and provably random. Neither of those things is possible on traditional computers without the involvement of quantum devices.

Quantum-Based RNGs

Enter quantum-based RNGs (*QRNGs*). QRNGs are quantum-based devices that can generate truly provable random numbers. There have been multiple real-world QRNGs since at least 2001 and papers on how to create them before that. There are dozens of mass-produced examples. They are the most produced quantum devices available. QRNGs are created in various ways and use different quantum materials and mechanisms to generate truly random numbers.

Central to most QRNGs is the quantum properties of superposition, entanglement, and uncertainty. A quantum property can be all possible states until measured, and you cannot predict ahead of time which value a particular property will take. Together, you get the basic properties needed to generate true randomness. Most QRNG devices use the quantum properties of photons one way or another as their primary source.

Bell's Inequality Theorem

Since the discovery of quantum physics, physicists and cryptographers have worried about whether the quantum property they were watching was truly quantum behavior or something else (with a classical explanation that they had missed). In the 1930s, Einstein and other physicists theorized that it was possible that then currently unknown and unexplainable (classical) local hidden variables attached to an object were responsible for the supposed quantum behavior that all scientists were seeing. Essentially, Einstein was not convinced that what scientists were seeing and explaining as quantum behavior was really quantum behavior. He (and others) wondered if it couldn't be something else that fit within the standard classical model of physics. What they conjectured is known as the *local hidden variables* theory.

The *local* in local hidden variables refers to the fact that the variables or properties impacting the object and determining its behavior are on, in, near, or otherwise directly attached to the object. The *hidden* part means that the local variables are not currently seen or explained.

The local hidden variables argument can be explained using this silly allegory. Suppose a group of people in a very cold climate were consistently observed to have hands that were inexplicably always warmer than their surroundings and the rest of their body. No matter what the outside temperature was, their hands were exactly two degrees warmer.

Further suppose that a group of scientists postulated that the area of weather around the people's hands must always be warmer than the weather impacting the rest of the people's bodies. They could not explain why the weather was warmer only in the region of the people's hands—only that it was always this way in all observations and was a "reasonable" explanation for why the people's hands were always warmer. The scientists even gave it a name, "microweather," and it became widely accepted by a growing group of scientists for many years after decades of observation.

Later, when the imaginary scientists looked more closely they saw that all along the people had been pulling gloves out of their pockets to keep their hands warm. It wasn't something fantastically new and wonderful; it was something a lot less exciting and basic. This example illuminates physicists' fear of not correctly explaining what they were seeing in existing understood classical theory and possibly prematurely calling some otherwise unexplainable behavior *quantum behavior*. Quantum physics seemed to run counterintuitive to what every scientist had observed previously.

Einstein also proposed a conclusive way to prove or disprove the existence of local hidden variables. If local hidden variables could be disproven, that would provide more proof that quantum mechanics existed. The proof didn't come during Einstein's life (he died in 1955), but he gave the physics world an experimental way of ruling out one of the last contrary possibilities and figuring out if quantum physics was something new and wondrous or just a missed facet of classical physics.

In 1964, Irish physicist John Stewart Bell, in his seminal paper titled "On the Einstein-Podolsky-Rosen Paradox," proved mathematically that local hidden variables could not explain observed quantum behavior, and he suggested experiments that could be conducted to prove it. Without going into the theoretical explanation of what Bell proposed (it involves angles, spins, and their measured differences of quantum particles and their properties), it showed a slight difference in the expected measurements between an object's properties and what would be observed if only classical behavior (and local hidden variables) existed. Graphically, in a classical-only world the differences would be measured to follow straight lines, but in a quantum-is-real world the measurements would be seen more like a bell curve (see Figure 7.1 as an example). It turns out that the experimental observations were always shaped more like a bell curve, thus proving the existence of quantum properties. The difference between the expected classical measurements and the real-life quantum measurements is known as a *Bell's inequality violation*.

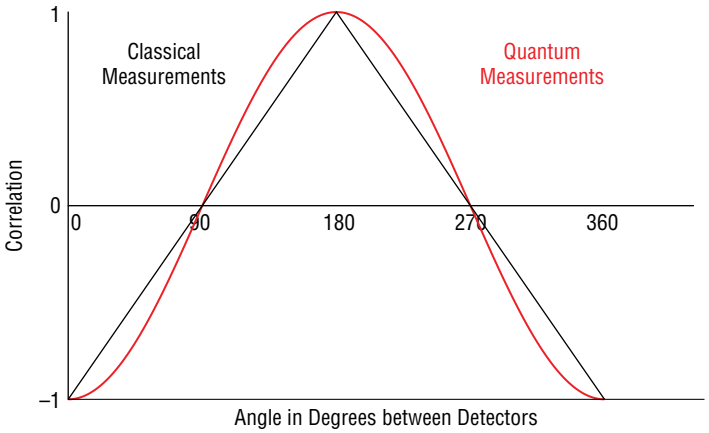

Figure 7.1: Bell's inequality expressed graphically

Simplistically paraphrasing, Bell created an experimental framework where the answer can only be false if classical physics is all we have, but we see that the answer is always true and so it cannot be only classical physics involved. In essence, Bell finally "proved" quantum physics by proving that local hidden variables could not be involved. Remember, physicists don't need to wait for pictures to believe something. The experimentation and supporting math is enough.

Starting in the 1970s, several physicists conducted experiments that gave the results expected by *Bell's inequality theorem*, but some viewers thought the experiments were poorly designed and that there could be some way sneaky local variables could still be at work (known as *Bell's loopholes*). But after hundreds of experiments never failing Bell's theorem, in 2015 a definitive, nonloophole experiment was conducted (`www.physicsforums.com/threads/another-loophole-free-test-of-bells-theorem.842620/`), finally proving Bell's theorem beyond a shadow of a doubt. A fairly good explanation of Bell's theorem is here: `https://en.wikipedia.org/wiki/Bell%27s_theorem`.

NOTE Although it might seem counterintuitive, "violating" *Bell's* inequality theorem is a good and expected outcome and means the experiment or device is correctly showing quantum properties. The "violation" is the measured results being different from the results you would get from only classical physics.

All quantum devices using entanglement, including QRNGs, should always be tested to make sure they pass Bell's tests. Developers want to make sure the results they are delivering are purely quantum and that there isn't some classical, nonquantum mistake being allowed into the results. This can be done during creation, testing, and certification. This is especially important to QRNGs whose true randomness must be assured to guarantee the strongest possible security during the rest of any dependent implementation.

There is also the theoretical fear that a QRNG using entanglement could be certified as passing Bell's tests, and then later be maliciously modified to output nonrandom numbers. For example, perhaps an enemy nation-state makes a QRNG component that a popular brand of QRNG relies on that appears safe; in reality, though, the long stream of "random-looking" numbers it outputs are secretly documented, meaning the nation-state knows what they are and will be. A QRNG should be testable by anyone to make sure it holds true to Bell's inequality theorem and is truly random. The best-designed QRNGs are designed so they self-test during operations to ensure they violate Bell's inequality. You'll find a good paper on this concept here: `www.nature.com/articles/npjqi201621`.

Additionally, QRNGs wishing to prove that they are consistent with being random will take the NIST 800-22 tests created to measure the randomness of any RNG (quantum or not) and post their results. Customers should be able to run the same tests and get similar results. You can find an example of one QRNG vendor's test results here:

`http://marketing.idquantique.com/acton/attachment/11868/f-004c/1/-/-/-/-/Randomness%20Test%20Report.pdf`

But the absolute best test for a QRNG (using entanglement) to prove that the numbers they are providing are truly random is to show proof that they were generated while violating Bell's inequality. The proof is provided at a quantum level. So far, the only working commercial QRNGs to do this are Cambridge Quantum Computing's QRNG called IronBridge (`https://cambridgequantum.com/cqc-unveils-the-worlds-first-commercially-ready-certifiable-quantum-cryptographic-device/`) and the private one NIST is currently working on (covered in a moment).

Working QRNGs

Early on, QRNGs filled up long laboratories and worked by shooting lasers between two devices separated by many football fields of laser optical cable. But QRNGs are increasingly being created in devices the size of a one-"pizza box" computer unit or even as small interface cards that can be plugged into a computer about the size of a stand-alone network interface card.

You can buy working QRNGs in all sorts of form factors from multiple vendors, ranging from a small original equipment manufacturer (OEM) chip to servers starting at prices under a thousand dollars. They are certified to work with many operating systems, including Windows, Linux, BSD, Solaris, and Apple. They have code libraries, APIs, and interfaces to several programming languages. Companies needing the services of QRNG have been buying and using them for a long time. QRNGs are in use in banking, science, lotteries, telecoms, finance, and the military.

Swiss company ID Quantique (`www.idquantique.com/random-number-generation/overview/`) was the first company to have a real-world working QRNG way back in 2001. They have released many ever-maturing products since. Figure 7.2 shows three different ID Quantique QRNG products built to be plugged into a computer's internal interface slot.

Other companies with QRNGs for sale include Australia's Quintessence Labs (`www.quintessencelabs.com/`), U.S.-based ComScire (`https://comscire.com`), and Canada's Quantum Numbers Corp (`www.quantumnumberscorp.com`). You'll find a good summary article on these companies and their products here: `www.nanalyze.com/2017/02/quantum-random-number-generator-qrng/`. You can even generate and use your own quantum-generated random number on several free places on the Internet, including `https://qrng.anu.edu.au`.

NIST QRNG Public Beacon

NIST created a QRNG in 2018 (`www.nist.gov/news-events/news/2018/04/nists-new-quantum-method-generates-really-random-numbers`) by shining a high-intensity laser into a crystal to create entangled photons. The randomness was proven to be to "within one trillionth of 1 percent," which is about as good as it gets. The goal of NIST's project is to create a public randomness beacon that anyone and any program can use (`https://csrc.nist.gov/projects/interoperable-randomness-beacons`). Initially NIST was going to develop a private service but decided that the world could benefit by having a trusted public QRNG source of truth.

Figure 7.2: Example ID Quantique QRNGs
Courtesy of ID Quantique

QRNG Disadvantages

Disadvantages of QRNGs include cost and interoperability. You can buy relatively cheap QRNGs (starting around several hundred dollars), but PRNGs are free or nearly free. Every computer already has one or more built in. To use a QRNG, you have to buy, install, interface, and use it. Currently, very few applications work with QRNGs, and no super popular off-the-shelf software works by default with QRNGs. The QRNG vendors have created software and drivers to allow existing applications to interface with their products, but most applications don't yet have the necessary "hooks." They can be fairly easily added by a developer, but they just don't exist for the vast majority of RNG-relying applications. Contrast that with traditional RNGs, which are already currently used by every existing device and relying application.

QRNGs are welcomed devices for even the pre-quantum world because they give us provably random numbers and can only improve all dependent cryptographic operations, classical and quantum.

Quantum Hashes and Signatures

This section discusses quantum-based hashes and digital signatures.

Quantum Hashes

As previously covered, hashes are one-way cryptographic functions that create/output a unique representative set of characters or bits (known as the hash, hash result, digital signature, or message digest) for examined unique content. Hashes are required and used in many other cryptographic processes such as encryption and digital signing. Traditional hashes are quantum-susceptible to pre-image attacks (although not collisions per `https://cr.yp.to/hash/collisioncost-20090517.pdf`). Accordingly, quantum-based hashes are needed.

Quantum-based hashes can take either traditional binary inputs or quantum inputs and return a hash based on quantum states. Quantum hashes that take binary content and return a quantum hash are known as *classical-quantum*. Like any other hash, traditional or quantum, it should be resistant to pre-image attacks, second pre-image attacks, and collisions. Quantum hashes naturally lead to quantum-based digital signatures.

A number of quantum-based hashes meet these conditions, although most have not been well tested over time. Although some quantum hashes have been implemented in real working devices, most are simply thought experiments to prove that quantum hashing is possible and can be implemented at scale when desired. Many scientific research papers are available regarding quantum hashes.

As an example, in 2013 Russians Farid Ablayev and Alexander Vasiliev proposed a theoretical classical-quantum hash (`https://arxiv.org/pdf/1310.4922v1.pdf`). Their quantum-hash algorithm (see Figure 7.3) is mathematically complex, containing enough advanced math that it is probably off-putting to most non–math majors. In a nutshell, using math proofs they both propose and prove all the necessary required hash properties as represented by quantum properties.

For a message $M \in \{0, 1\}^n$ we let

$$|h_K(M)\rangle = \frac{1}{\sqrt{d}} \sum_{i=1}^{d} |i\rangle \left(\cos \frac{2\pi k_i M}{N} |0\rangle + \sin \frac{2\pi k_i M}{N} |1\rangle \right)$$

Figure 7.3: Mathematical representation of Ablayev and Vasiliev's quantum-hash algorithm

Their crucial argument is that their algorithm allows qubits to accurately and uniquely represent any message *n* bits long with no more than *O(log n)* qubits. Without getting into the details, O(log n) essentially means less than *n* qubits required for each bit of the hashed message, which theoretically means the original message cannot ever be obtained from the smaller resulting hash.

NOTE If you are interested in learning more about what O(log n) mathematically represents, see `www.quora.com/How-can-we-check-for-the-complexity-log-n-and-n-log-n-for-an-algorithm`, and `www.quora.com/How-would-you-explain-O-log-n-in-algorithms-to-1st-year-undergrad-student-Can-any-one-explain-it-with-mathematical-proof-for-log-n-complexity-by-taking-a-simple-example-like-Binary-search-and-simple-to-understand`.

Holevo's Bound

Ablayev and Vasiliev's algorithm relies on another quantum theorem known as *Holevo's Bound* (`https://en.wikipedia.org/wiki/Holevo%27s_theorem`). This theorem says that a qubit can be one of two states, but when it is measured (decohered), it must break down into a measurement represented by only one of two states—one state of information is lost in each qubit measurement (i.e., a bit with only two states cannot accurately measure a property with three possible states). To accurately represent a single qubit (which can be three possible states) would require at least two binary bits (2 bits = 2^2 = 4, meaning they can represent 4 possible states).

They end their paper by creating a quantum-based digital fingerprint algorithm based on their hash. The paper and their research was funded by a Russian Foundation for Basic Research grant. Both hash creators went on to define even more quantum hashes and even discovered and proved even more complex math that could be used to base any quantum hash on.

For more information on quantum hashes, see `www.bjmc.lu.lv/fileadmin/user_upload/lu_portal/projekti/bjmc/Contents/4_4_17_Ablayev.pdf`.

Quantum Digital Signatures

The difference between a hash and a signature is one of identity authentication. A hash gives a unique fingerprint for unique content. A digital signature ties a hash to a subject's identity. For example, say you hash a file and the hash comes back as 1234. Then you use the private key of your asymmetric key pair to digitally sign the hash. The asymmetric key pair is tied to your identity. A digital signature is a subject locking in a particular hash at a particular point in time to a particular identity.

To get a digital signature, you need a hash followed by an asymmetric key pair and a digital signature algorithm. To verify any purported digital signature, a stakeholder would need the signer's verified, corresponding public key, which they would then use to "unlock" the hash, which is possible only if it was signed by the valid, corresponding private key. If the public key did not correctly reveal the valid, previously "encrypted" hash, which correctly represented the purported hashed contents, then the file's integrity or the digital signature would come under scrutiny. Either way, no stakeholder would trust the file or digital signature.

A traditional asymmetric key pair and digital signature could be transmitted using quantum communications, but in this chapter we are talking about a true quantum digital signature, one where the quantum digital signature is based on quantum properties. A quantum-based digital signature requires a quantum hash, a quantum-based asymmetric key pair, and a quantum-based digital signature algorithm, all represented by quantum properties. Quantum properties are currently not stable for long periods of time, and so they don't make fantastic asymmetric key pairs. There are many other additional scaling problems (discussed later) that do not make quantum-based digital signatures good for normal digital signing, especially as compared to the far less stringent (and still very secure) quantum-resistant digital signatures available. But for limited-use cases, quantum-based implementations would make ultra-secure digital signatures for short periods of time.

One of the reasons a quantum-based digital signature would be very secure is the use of a quantum-based hash. As discussed earlier, quantum-based hashes are very difficult to attack. They are based on the Holevo's Bound theorem and impossible (even using quantum computers) to compute back to the original message. The significant complicating problem is that because of the no-cloning theorem, a signer can't simply create a bunch of identical quantum public keys and send them to receivers. Instead, it gets a lot more complicated.

The First Quantum Digital Signature Algorithm

The first practical quantum digital signature algorithm was created in 2001 by Daniel Gottesman and Isaac Chuang (`https://arxiv.org/pdf/quant-ph/0105032.pdf`). It isn't pretty or efficient, but it works. First, the sender/signer, Alice, must separately sign every qubit/bit of the message. She can't just hash content and sign the hash in a single operation as is possible with classical signing algorithms. The hash (or the message) must be signed one qubit/bit at a time. Alice must create one or more private keys to be used if a bit of her message she is signing = 1 and a separate set of private keys if the bit of her message = 0. Then, using the asymmetric cipher algorithm, Alice generates a corresponding public key for each private key created and sends all of the public keys to all recipients, Bob and Charlie. The number of recipients cannot be more than a handful because each additional copied key pair starts to create an increased risk of key compromise.

Now for each 0 bit in the signed message, Alice sends Bob and Charlie all the 0 bit–related private keys, along with the 0 bit of the signed message. For each 1 bit in the signed message, Alice sends Bob and Charlie all the 1 bit–related private keys along with the 1 bit of the signed message. Bob and Charlie then use the previously sent public keys to validate the private keys of the signed bit. If Bob

and Charlie's comparison error rate is low, then the signed bit is validated. If Bob and Charlie's comparison error rate is not low, then it can be assumed there is a compromise somewhere in the system. The process must be repeated for each bit of the signed message/hash.

It's not a very efficient way to do digital signing even if you need ultra-high security, although protocols for increasing efficiency across all message bits have been proposed (https://www.researchgate.net/publication/312062995_The_postprocessing_of_quantum_digital_signatures). Still, it was the first quantum-based digital signature showing that it could be done, however inefficiently.

Phase-Encoded Digital Signatures

Then, in 2012 another, slightly more efficient but very similar quantum digital signature algorithm was published in *Nature* magazine that used "phase-encoded coherent states of light" (www.nature.com/articles/ncomms2172). With this method, Alice chooses a random set of quantum states (which equates to a private key) as can possibly be represented by different phases of light. Each message bit is still signed separately by Alice. For each 0 bit or 1 bit of the message, Alice generates a pair of phase-encoded states and sends a copy of the pair to Bob and a copy of the pair to Charlie. Bob and Charlie then decode the phases and verify the signed bit.

In 2013, quantum-based digital signatures were slightly improved on again (https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.112.040502), followed by many successful digital signature experiments including www.nature.com/articles/s41598-017-03401-9 and http://cnqo.phys.strath.ac.uk/research/quantum-theory-of-light/quantum-digital-signatures/. These latter experiments used the improved, phase-encoded signature variation.

Quantum-based digital signatures have progressed enough that cryptographers are now looking at the ways they can be successfully attacked, including the following two papers: https://link.springer.com/article/10.1007/s11128-019-2365-8 and www.sciencedirect.com/science/article/pii/S0030402617308069. Whenever a cryptographic algorithm or product undergoes attack review, it's a good sign and a maturing of the protocol. With that said, the need for quantum-based digital signatures, especially with the increased complexity and multitude of good quantum-resistant digital signatures, is likely to remain low for the foreseeable future.

For more information on quantum digital signatures, see https://arxiv.org/pdf/quant-ph/0105032.pdf, https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.113.040502, and https://en.wikipedia.org/wiki/Quantum_digital_signature.

Quantum Encryption Ciphers

Quantum encryption means protecting data at rest or during transport using quantum devices, software, and properties. Just like in the traditional binary world, quantum encryption ciphers can be symmetric or asymmetric. It is difficult to impossible to view, copy, or manipulate data protected by or in a quantum state. If someone unauthorized attempts to directly view the data or insert themselves

into a quantum-encoded data stream or storage area, the quantum state will be changed. This is guaranteed by the observer effect and the no-cloning theorem. Observers could manipulate the encoded data, but not in a way that would not be easily detected by the involved authorized parties. This is a desirable trait for cryptographers and users of cryptography. This section will cover asymmetric quantum ciphers in general but leave the larger discussion of quantum networking for Chapter 8.

NOTE Readers may be wondering why quantum-based symmetric ciphers are not being discussed. This is because the field is largely unstudied with very little literature and research available. Traditional binary symmetric encryption is currently considered resistant to known quantum attacks, and as such, there has not been much discussion about them. Perhaps the field of quantum-based symmetric ciphers will get more research and resources devoted to it in the future, but for now the field is largely unstudied.

Asymmetric cryptography has been around since the 1970s. We know that this type of encryption is normally used to securely transmit symmetric encryption keys between source and destination, and for digital signing and authentication. Traditional asymmetric cryptography has worked well for over a half century up until now.

As is the focus of this book, quantum computers are likely to soon break many forms of traditional quantum-susceptible public key cryptography, including RSA and Diffie–Hellman, the most popular implementations. As covered in Chapter 6, there are over two dozen quantum-resistant cryptographic algorithms competing to be the new NIST post-quantum standard. All of these quantum-resistant algorithms, covering both public key cryptography and key exchange, are based on binary computations with binary keys using binary devices.

Quantum-based asymmetric ciphers are based on quantum devices and properties. One type, currently the most popular, uses quantum properties to securely transmit a traditional secret symmetrically between authorized source and destination. This is known as *quantum key distribution* (*QKD*). Another method uses quantum properties to securely transmit the key, and the key itself is made up of quantum properties. Some researchers refer to them as *Quantum Public Key Cryptography* (*QPKC*) Class 1 and Class 2, respectively (including this paper: `https://arxiv.org/pdf/0810.2859.pdf`). QPKC Class 1 is QKD where the key is still composed of binary bits. QPKC Class 2 is QKD with a key made up of quantum qubits. QPKC Class 2 is harder to pull off because of the lack of quantum network devices and the inherent complexities needed to keep a quantum-based secret key stable for long periods of time. Thus, most QPKC algorithms and implementations are Class 1. Even those are proving challenging to implement in the real world right now, but we have plenty of successful implementations.

Quantum Key Distribution

There are many QPKC Class 1 key distribution algorithm and systems, and although they are not widespread, there have been many private and commercial networks using QKD since the early 2000s. QKD-based networks were first used in Europe in 2007 and have been used in the United

States since 2010. Today, many countries, especially China, use and improve them. There's more on these quantum-based networks in Chapter 8.

BB84

The first QKD algorithm was created by American Charles Bennett and Canadian Gilles Brassard in 1984 and is frequently discussed simply as *BB84*. Bennett and Brassard are considered two of the fathers of quantum cryptography. Bennett, an IBM Fellow, continued doing research into his 70s and was actively posting on a blog site called *The Quantum Pontiff* (`https://dabacon.org/pontiff/author/chb/`) as recently as 2016. Brassard also helped create the *Cascade error correction protocol*, which helps detect and correct "noise" caused by eavesdropping on quantum-protected cryptographic channels (Bennett did much research here as well) and did work in quantum teleportation and a new game theory known as *quantum pseudo-telepathy*.

BB84 was not only the first QKD scheme, but by definition the first algorithm to mathematically show that using quantum states was provably secure from eavesdropping. Although I'm skipping the mathematical detail, here are the basic ideas involved with BB84. Alice wants to send a message to Bob across an untrusted channel. Alice needs to get a shared symmetric key to Bob so they can start encrypting secret messages to each other. Here are the basic BB84 steps:

1. Alice creates two random binary strings, say *a* and *b*, and then encodes them using qubits and the BB84 mathematical algorithm into a single result, say *n*. *a* and *b* are mathematically linked to each other, but no one can know what *b* is without first knowing what *a* is.
2. Alice sends *n* across a quantum channel as qubits to Bob, and it's the last time the quantum channel is used. The remaining communications happen on a public classical channel.
3. Bob measures all the received *n* qubits, which decoheres the qubits into bits. Bob measures half the qubits one way (say Method 1) and half the qubits another way (say Method 2). Only one of the methods is the right method for what Alice sent. Method 1 and Method 2 give different answers during measurement depending on whether the qubit represented a 0 or a 1. The right method that aligns with what Alice sent will correctly measure all 100% of its half of the qubits (i.e., 50% of the total) and the wrong method will end up only correctly measuring 50% of its half of the qubits (i.e., another 25% of the total). If Bob received and measured all qubits correctly from Alice, due to the way they are measured, using both the right and wrong methods, only 75 percent of the bits will accurately represent the qubits sent by Alice. This is expected.
4. Bob communicates to Alice which method, 1 or 2, he used to measure each qubit.
5. Alice, now knowing which method Bob used for each qubit and what the outcome would have been, tells Bob which qubits he measured were measured correctly and which were measured incorrectly.
6. Both Alice and Bob will discard the incorrectly converted bits, and the 75 percent remainder becomes their shared secret key.

7. Just prior to the using the newly shared secret key for future trusted communications, as a test Alice and Bob will share a short series of the key's bits between each other over the untrusted channel. If the test comparison matches 100 percent, they will begin securely communicating by using the rest of the shared key. If not, they will assume noise or eavesdropping and not trust the current shared key.

The BB84 protocol has more steps and is more complicated than this, but this series of steps contains the overall gist of the protocol. A good video representation of BB84 is here: `www.youtube.com/watch?v=UVzRbU6y7K`. Almost all QKD schemes are improved versions of BB84, or at least provide even better protection than BB84.

If an eavesdropper, Eve, was able to intercept the original qubits between Alice and Bob, Eve's measurement of the qubits would decohere into only 75 percent of the correct bits (as would happen to any original, legitimate measurer). But Alice doesn't know which bits were measured correctly and which were not, and so she would have to send what she measured (including the 25 percent of the wrong bits) to Bob. Resending those measured bits as qubits along to Bob would result in Bob getting only 75 percent of the original correct qubits intended to be sent to him. When he decoheres them by feeding them into his two methods, he gets a percentage less than 75 percent right instead of exactly 75 percent. When Bob and Alice communicate about what the received bits were and what methods Bob used to read them, if Bob gets anything less than 75 percent accuracy, then Alice and Bob know that the channel had noise or was eavesdropped on.

QKD methods like BB84, where the quantum state of one photon is sent between sender and detectors, are known generally as *discrete-variable QKD*. Many other improved QKD systems based on BB84 were developed, including B92 (`www.semanticscholar.org/paper/Quantum-cryptography-using-any-two-nonorthogonal-Bennett/e99dc04d91409a4668ad0368ef7017e27a034008`), created by BB84 coauthor Bennett; SARG04 (`https://en.wikipedia.org/wiki/SARG04`); and the Six-State Protocol (`https://en.wikipedia.org/wiki/Six-State_Protocol`).

Entangled QKD

In 1991, British-Polish professor Artur Eker introduced a fundamentally different QKD approach using entanglement in his paper, "Quantum Cryptography Based on Bell's Theorem" (`http://cqi.inf.usi.ch/qic/91_Ekert.pdf`). Eker's method looks somewhat like this:

1. Alice creates a secret key using a split entangled qubit for each bit of the key.
2. Alice keeps one side of the entangled pair and sends the other side to Bob across a quantum channel.
3. Both Alice and Bob measure their qubits (decohering them into bits), using their own detectors with different orientation combinations for each qubit.
4. After measuring all qubits, Alice and Bob announce the orientation of their individual detectors for each measured qubit.

5. The qubits that were measured by the same detector orientation are discarded.
6. Both Bob and Alice, now knowing what each other's detector orientations were, can convert the remaining bits into their resulting binary representations.
7. Lastly, both Bob and Alice use a Bell inequality test to perform an entanglement check. If entanglement was broken, then it can be assumed an eavesdropper or bad noise was involved. Either way, the resulting secret key would not be trusted.

Eker's QKD approach (also known as *E91*) led to a bunch of new QKD algorithms and other related schemes.

Photon Number Splitting Attack

In theory, QKD systems are resistant to eavesdropping because of the quantum physics observer effect and no-cloning theorems. As long as a single qubit is used to represent a single bit during transmission, it is difficult for Eve to eavesdrop without detection. But in practice, most QKD systems cannot send just a single photon for each transmitted bit. The photon quickly loses its strength as it travels along the fiber-optic cable, and the read detectors have a hard time detecting a single photon accurately. Because of this, most QKD systems send multiple photons for each transmitted qubit/bit. In multiphoton systems, it is possible for Eve to siphon off one or more of the duplicate photons while letting the rest continue on between Alice and Bob. QKD protocols (like SARG04) have been created, and most real-world QKD systems include additional protections and error correction mechanisms to decrease the risk of photon number splitting attacks.

Continuous-Variable QKD

A second major, newer type of QKD is known as *continuous-variable QKD* (*CV-QKD*). In CV-QKD, quantum properties are encoded in the modulation of amplitudes and phases of a laser beam stream, which can then be decoded by a detector known as a *homodyne detector*. These methods are more resilient against photon number splitting attacks. CV-QKD systems can send more keys per time period (as compared to discrete-variable QKD systems) and are cheaper to implement, but they cannot function over the multikilometer distances that discrete-variable systems can. Examples of CV-QKD schemes can be found here: `https://arxiv.org/ftp/arxiv/papers/0705/0705.0515.pdf` and `https://arxiv.org/pdf/1703.09278.pdf`. You will frequently read about quantum network devices supporting discrete-variable or continuous-variable algorithms.

Repeater Issues

QKD systems, because of the no-cloning theorem, are resistant to undetectable eavesdropping. In cryptography circles, this is a very good thing. In practice, trying to use QKD across large networks is a huge problem. QKD systems are essentially created to be point-to-point. You can't just stick a traditional type of repeater or router in between two QKD endpoints to repeat the message along the way. The repeating device would be treated like an eavesdropper. Even in long, point-to-point

connections, there are only so many miles you can send a quantum light signal before it loses its strength and has to be repeated.

NOTE Research is being done to increase the distance a quantum signal can be sent before needing repeating, including this: `www.nature.com/articles/s41586-018-0066-6`.

So, QKD networks have to be one contiguous, point-to-point system or the qubits have to be measured, decohered to binary, and then retransmitted using quantum repeaters. Each of these binary locations is a weak link in the chain where an eavesdropper could insert themselves and learn the encrypted information with impunity.

Think about how many repeater points you have just in your simple home network. Your mobile devices probably connect to a Wi-Fi router, which connects to your cable modem, which connects to a device outside your house, which connects to a neighborhood aggregation point, which then connects to dozens to hundreds of other repeaters on its way to your Internet service provider (ISP). Then your ISP connects to one to three dozen nodes to get around the Internet (each of which could have any number of repeaters to and from them), and finally goes into the eventual destination computer or device—and that path is reversed for every network packet needing to be sent back. The typical Internet network packet easily travels across dozens of repeaters. This creates a huge challenge if we ever want to have a quantum-based Internet one day. This challenge and the solutions will covered in more detail in Chapter 9.

Other QKD Concerns

QKD systems have lots of critics, including Bruce Schneier (`www.schneier.com/blog/archives/2018/08/gchq_on_quantum.html`) and the U.K. National Cyber Security Centre (`www.ncsc.gov.uk/whitepaper/quantum-key-distribution`). Many people wonder if the cost of developing a pure quantum network is worth the benefits. What we know we can make today, especially when you add the quantum-resistant crypto, would probably handle all the highest security needs for a long, long time.

Another practical concern is the lack of current interoperability between existing QKD implementations, even when they are supposedly using the same algorithms, but some projects are starting to prove the ability to interoperate. More generally, many critics find the problems that they solve to be expensive edge cases and not something that can be practically implemented on a large scale any time soon. The U.K. whitepaper contends that it will be impractical for a long time to think that QKD ciphers can be used on a mass scale, such as in the billions of Internet of Things (IoT) devices. Instead, critics recommend sticking with slightly more proven quantum-resistant ciphers.

Another serious concern is the overall security of real-world QKD systems. They haven't been around long enough to learn what the big security issues are. Beyond the weak links created for each repeater node, QKD systems themselves are open to a myriad of hacker attacks, but at the binary and quantum levels. Cryptographers don't like to use systems that have not withstood the test of time by lots of researchers and attackers.

These are correct assessments for the current time period, but similar arguments have been made for why cars could never replace horses, how it's impractical to think that electric cars could ever replace gas cars, and I'm sure our IT ancestors would have a hard time imagining the computing power we have in our tiny, handheld mobile phones. Obstacles of cost, security, and distribution are usually overcome, especially in the computer world.

Kak Protocol

QKD inherently uses a bunch of classical systems and channels. If quantum keys are going to be used, QKD is preferred, concerns and all, because there aren't that many pure quantum cryptographic systems. In the future, completely quantum QPKC Class 2 systems will probably be used. If so, they will likely be based on something called *Kak's three-stage protocol*.

Before the advent of asymmetric ciphers, one of the common proposed solutions was to have a multistage, multikey (three-stage) key exchange. In the traditional example where Alice and Bob are trying to securely communicate across an untrusted channel using a three-stage symmetric key exchange, Alice would encrypt the intended secret with a private key that only she knew and then send it to Bob. Bob would encrypt what was sent to him with his own private key that only he knew, and then send it back to Alice. Alice would then remove her encryption, but it would remain encrypted with Bob's secret key, and then send it back to Bob. Bob would then remove his encryption and read Alice's plaintext message. Essential to this type of three-stage encryption is the ability of both parties to be able to remove their own encryption, especially the ability of Alice to accurately remove her encryption after Bob's encryption has been applied (see Figure 7.4).

In 2005, Louisiana State engineer Subhash Kak published a paper (`https://arxiv.org/pdf/quant-ph/0503027.pdf`) showing how to do the symmetric three-stage key exchange using quantum properties. It is now known as the *three-stage quantum protocol,* or *Kak's three-stage protocol*. It has been implemented in real-world quantum devices using single- and multiphoton approaches. Single-photon approaches are difficult to impossible to eavesdrop on without the authorized parties being aware. A man-in-the-middle attacker could disrupt the process, but the disruption could be offset by error correction mechanisms.

Since Kak's paper was published, additional versions using multiple photons and error correction improvements have been developed. Multiphoton implementations allow more potential applications but increase the risk of successful eavesdropping. Developers of the multiphoton implementations include other protections to offset the increased risk. Kak's quantum protocol was widely welcomed because it is purely quantum and does not use classical components, as other quantum key exchange protocols do.

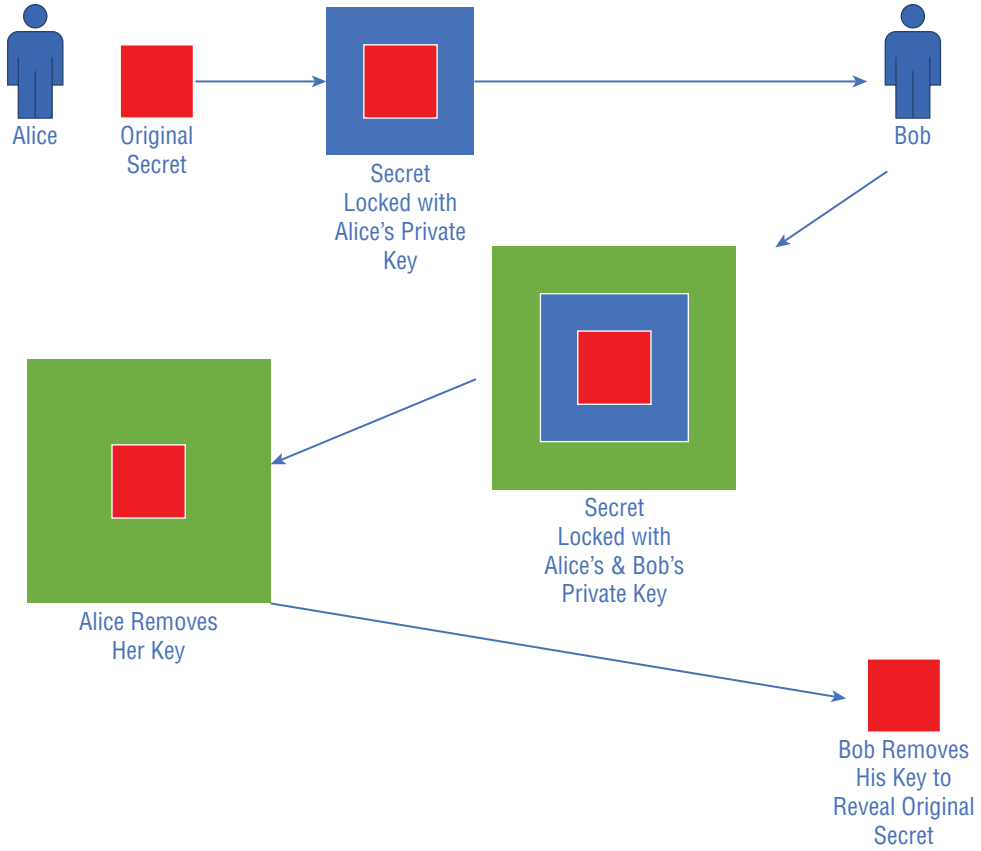

Figure 7.4: Three-stage symmetric encryption

QKD Companies

There are many companies that make QKD systems, including

- ID Quantique (`www.idquantique.com`)
- MagiQ Technologies (`www.magiqtech.com`)
- Quintessence Labs (`www.quintessencelabs.com`)

ID Quantique makes three QKD systems (`www.idquantique.com/quantum-safe-security/products/#quantum_key_distribution`), including a larger server that contains optical blades. Among other things, each blade can represent Alice or Bob in the QKD communications stream. Their QKD

systems can go up to 100 kilometers/62 miles before needing a repeater and will automatically send an alert and sound an alarm if an eavesdropper is detected. MagiQ Technologies makes a QKD device known as QPN, which works using the BB84 method (www.magiqtech.com/solutions/network-security/).

For more information on QKD, see: https://arxiv.org/pdf/1504.05471.pdf and https://ieeexplore.ieee.org/abstract/document/6459842.

Summary

In this chapter, we explored quantum-based cryptography. Quantum cryptography includes random number generators, hashes, digital signatures, and key distribution. There have been quantum-based random number generators and key distribution schemes for nearly two decades. Many companies offer several mature models of both types. Sizes range from large, football field–sized rooms to small computer interface cards and chips. Prices range from only a few hundred dollars to many tens of thousands. These types of quantum cryptography devices continue to be improved. Their complexity and price are lowered, their useful life and useful distances are improved, and their ability to interface with existing systems gets better each day. New protocols and attacks are being developed and tested. Unfortunately, there is not a whole lot of need for quantum-based asymmetric ciphers, digital signatures, or symmetric ciphers at the current moment and possibly not for many years. The multitude of quantum-resistant ciphers will likely decrease the demand for all quantum-based cryptography, at least until enough cheap, quantum-based devices are available to be competitive.

The next chapter will cover how that cryptography is used and being improved to create quantum-based networks.

8 Quantum Networking

This chapter covers networking as specifically done by quantum devices using quantum properties. The quantum-resistant cryptography covered in Chapter 6 and the quantum-based cryptography covered in Chapter 7 can certainly be and often is used but is not a requirement for quantum-based networking. Quantum networking has its own inherent challenges and has been pursued by various companies and nations for well over a decade. Now that quantum supremacy is close to becoming a reality, the rush to find a sustainable quantum networking model is gathering steam. Specifically, this chapter will cover quantum networking components, challenges, and likely applications.

Networking is using an agreed-upon communication protocol to move information and content from source to destination over some sort of transmission medium, be it wireless, wired, or otherwise (e.g., human-based networks). Quantum networking uses quantum devices, properties, algorithms, and protocols to transmit (quantum) information across a network. As with every other quantum technology, quantum networking uses the full gambit of quantum properties, although you will hear the most discussion around superposition, entanglement, and the no-cloning theorem. Quantum networking can be done to link more quantum devices, physically spread apart, into a more powerful collective and to facilitate the transmission of information and content (including quantum teleportation). If done correctly, quantum networking promises a far more secure network more impervious to unauthorized eavesdropping than today's nonquantum methods.

Quantum Network Components

Like any network, quantum networking consists of transmission media, protocols, and networking devices. Quantum networks may be made up of classical and quantum components or be purely quantum-based.

Transmission Media

Transmission media includes both physical cables and free-space media.

Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto, First Edition. Roger A. Grimes.

Fiber-Optic Cables

Quantum physical networking mostly uses photons and the light is usually transmitted through fiber-optic cable. In some implementations, normal (high-quality) fiber-optic cable already used for light-based classical networking can be used for quantum networking as well. All implementers have to do is change out the instrumentation used to send and receive the signals. Instead of sending long waves of light, which are encoded and decoded into binary representations, individual photons are used, and the encoding is done across the individual quantum properties of each involved photon. Other times, specially built, ultra-high-quality fiber-optic cables are created and deployed. Quantum networking fiber-optic cables are more protected against external influences and internally more sensitive to changes, which quantum networking requires. Cable quantum transmissions are usually less than 100 kilometers long, although longer networks have been created.

Theoretically, only a single photon of light is used to send each qubit, and this method provides the easiest inherent security. But single-photon qubits get more easily blocked, lost, and decohered the farther they travel across a network in the real world. In many quantum networks, multiple photons are created, encoded, and transmitted representing the same qubit of information to increase the chances of the represented single qubit to successfully make it from source to destination. This, however, causes security issues. More on this in the Entanglement Purification section below.

Different wavelengths of light are used depending on the type of quantum network devices involved. Oftentimes diamonds, crystals, and other gems and materials are used to generate different wavelengths and colors. They may be intentionally doped with defects or selected for particular natural defects, to create a desired trait that can be used in quantum networking.

NOTE One of the most common defects found in diamonds, not just for quantum networking, is called the *nitrogen-vacancy center*. Diamonds are normally made up of all carbon atoms, but sometimes a nitrogen atom gets thrown into the center of the diamond's carbon lattice. Carbon and nitrogen are next to each other in the periodic table of elements, and a nitrogen atom has one more electron than a carbon atom. This creates an "extra," very usable, free-floating electron that can be used in quantum computing and communications (among many uses). While too many carbon defects are bad for jewelry lovers, they're great for creating additional usable colors and wavelengths and manipulating subatomic quantum properties. For more information, see `https://en.wikipedia.org/wiki/Nitrogen-vacancy_center`.

Free-Space Media

Free-space media includes any transmission media not bound to a physical object. The most common quantum free-space transmission media is electromagnetic waves of some type, including microwaves, laser beams, and sound waves. Quantum transmissions can be sent point-to-point from ground-based sending and receiving stations, from ground station to satellite and back, or in some other hybrid arrangement. Free-space media transmissions are much more likely to be impacted by external influences.

A quantum satellite experiment has demonstrated sending entangled photons over 1,200 kilometers and single photons have been sent from a satellite from over 20,000 kilometers away. Like with today's traditional networks, quantum satellite networks are most often seen as a way to connect distributed ground-based network and mobile systems, although quantum networking has particular applications to some sorts of satellites all on their own (covered in the Quantum Network Applications section below).

It is difficult and expensive to beam a quantum photon to a high-orbit satellite 20,000 kilometers away. If you think trying to transmit a single qubit across a protected fiber-optic cable is difficult, imagine trying to successfully transmit that same qubit through unprotected space with all the trillions of other quantum particles it might encounter every second along the way. It has been done, but it was not easy.

The Chinese have successfully experimented with using relatively inexpensive (and lower) flying drones to send and receive quantum information between two ground-based nodes. This method might be used to connect ground stations several hundred kilometers away. Once the relevant issues are worked out, free-space media will likely be the way wide-area physical quantum networks work and get extended.

Distance vs. Speed

All networks and their materials and devices are subject to physical laws of nature that govern how far and fast they can transmit information, even when using the best and most efficient equipment. Quantum networking is no different, although the involved physics laws may seem even stranger. Those laws govern how far and fast a quantum network can be, at least without a repeater device (unless there is some new, currently unknown, epic development to the contrary). In general, the longer a quantum network is the slower the transmission of information; and vice versa. There are theoretical maximums for both, depending on the transmission media used.

PLOB Bound

A quantum physics law known as the Pirandola-Laurenza-Ottaviani-Banchi (PLOB) bound (`https://arxiv.org/pdf/1510.08863.pdf`) says the maximum rate at which qubits can be transmitted over any known point-to-point, two-way, single-network segment without additional repeaters is equal to $-\log_2(1 - x)$, where x is the network transmission's channel's transmissivity. *Transmissivity* is the (maximum) degree/speed/length to which any medium will allow something to pass through it. Each transmission medium can only physically allow something (i.e., electricity, light, electromagnetic radiation, etc.) to be transmitted through it at a certain speed for a particular distance before the transmitted substance starts to degrade or be stopped completely.

The PLOB bound sets a maximum theoretical rate on the speed and length of a single point-to-point quantum network segment without a repeater involved, and hence the maximum distance a single quantum repeater or network segment can transmit information depending on the transmission medium. It sets an upper boundary for how long and/or fast any quantum network can be (without a repeater) for quantum transmission including for fiber-optic cables and satellite links. It even

considers quantum entanglement and discusses how natural entropy limits entanglement. Real networks are unlikely to ever be able to obtain the maximum speeds and distances as defined by the theoretical PLOB bound because of environmental and physical resistant issues.

But quantum networking vendors are working to increase the maximum distance and rate their equipment can work across a single segment. Most quantum networking device vendors test their equipment across a range of distances and speeds, and many publish those figures so customers can see what they are considering buying performance-wise and for comparison shopping purposes (either for different equipment models within their own line or with another competitor's products). You will usually see a commercial quantum network-related equipment vendor share the following figures at a bare minimum: maximum transmission loss acceptable (in dB), maximum transmission channel length (in kilometers), along with various speed ratings, such as 1.4 kilobits of key generation at 50 kilometers. Speed ratings will always go down as the distance increases. A good white paper on quantum networking distance versus speed can be found here: `https://www.nature.com/articles/s42005-019-0147-3.pdf`.

In every quantum network, additional devices and mechanisms are needed to convert the network-transmitted qubits onto the eventual destination devices, whatever they may be. In the classical world, these devices are often known as network interface cards. In the quantum world they are often known as optical switches, beam splitters, detectors, or repeaters (covered in more detail below).

Point-to-Point

Almost all current-day quantum network transmission types are point-to-point, meaning from one location directly to another, source to destination. The network transmission media is not shared or distributed. Essentially just picture a string that runs from the origination node to the destination node. To extend the quantum network to more nodes, additional point-to-point connections are added.

This is due to several reasons, not the least of which is that qubits don't want to be shared (e.g., the no-cloning theorem), the difficulty in keeping quantum signals isolated from the external world, the cost (point-to-point links are cheaper), and the sheer complexity needed to pull off a non-point-to-point network connection. A shared, cloud-based network arrangement is a much sought-after Holy Grail in quantum networking. Surely one day they will be the norm, but right now single point-to-point connections compromise most quantum networks. A smaller, but growing number of quantum networks are increasing the number of point-to-point segments that end up creating a larger, sometimes metropolitan-wide area network.

This "walk before you can run" network distribution model can be likened to the early days of classical networking. The early first networks were point-to-point, dial-up analog phone connections. Eventually, traditional networking matured to a point where either a shared "shuttle token" picked up and dropped off bits, (e.g., Token Ring, etc.) or the media could be shared by multiple nodes with frequent retransmissions (e.g., Ethernet). Internet connections used to be point-to-point and required external phone dialing (e.g., RJ-11 analog, ISDN, Frame Relay, etc.), but eventually matured to the

model we have today where all a house or node has to do is connect to one of the many neighborhood aggregation points (based on the Internet service provider) or satellite uplinks. This is often known as the "last mile." The neighborhood connections hook to larger wide area networks, which then hook into the global Internet. Now, almost anyone can send a network data packet around the world in a few seconds. There is no doubt that one day quantum networking will be extended to the house/node model we are used to today. But we are in the very early days of quantum networking. Right now, most quantum networks are private point-to-point experiments, and often contain repeaters to extend their maximum transmission lengths. There are two basic types of quantum repeaters.

Trusted Repeaters

In the classical networking world, all a repeater has to do is capture, re-amplify, and re-transmit all captured bits. It can do this at the physical level, simply essentially detecting and re-transmitting electricity at nearly the speed of light. However, the no-cloning theorem means that signal duplication and application cannot as easily be done on a quantum network. The quantum information can always be read (which decoheres it to classical binary) and then be re-encoded onto the next quantum network segment and sent along as new qubits. And this is the most common quantum repeater method used today.

The key concern to using this method is how anyone can be completely assured that the repeater reading and recoding the information is doing the job accurately and without unintentional or intentional malicious malformation? In the pure quantum world, quantum mechanics provides the trust. You don't need to trust any of the intermediary devices, because if anyone tampers with them or eavesdrops, that interference can be immediately detected. But when not all the repeating devices are completely quantum, as is often the case in today's quantum networks, secure transfer of information has to be ensured another way.

The answer is to create what is called a *trusted repeater*, where the information is protected using quantum encoding sent along by one or more secure repeaters that everyone involved trusts along with multiple sets of quantum-based encryption keys. For example, suppose we have a quantum network trying to move quantum information from origination point A to destination point Z, with trusted repeater R in between (as shown in Figure 8.1). Both sides have to trust R to be secure.

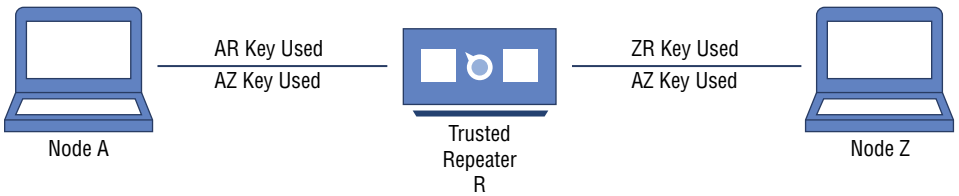

Figure 8.1: Quantum network using a trusted repeater

Both A and Z would each separately create quantum keys (using quantum key distribution as covered in Chapter 7) to be used to encrypt and securely transport other key(s) that would do the actual data encrypting. Let's call them the AR and ZR keys. They would share AR and ZR only with R. Node A would then create a data encryption key to be used between itself and Z (let's call it the AZ key). Node A would encrypt the AZ key with the AR key and send to Trusted Repeater R. R would decrypt the key sent by A, re-encrypt it with the ZR key, and then send onto the Z node. The Z node would decrypt, now securely getting the AZ key created by A. When A and Z wanted to send data to each other, they would encrypt that data with the AZ key and then send to R. R would read/decohere the encrypted quantum data (leaving it in its encrypted state), re-encode, and resend onto the other side. Clearly R has to be secure and trusted during the initial AZ key exchange in order for everything to remain secure. A Kak three-stage key exchange (as covered in Chapter 7) would also work without having to explicitly trust the repeater, but the trusted repeater model is the main quantum repeating model used in most extended quantum networks today.

True Quantum Repeaters

An even better idea to maintain pure quantum-ness throughout the networking transmission is to use quantum entanglement and teleportation (as covered in Chapter 5). Why worry about reading/decohering quantum data and re-encoding it again if the quantum state can just be transmitted from source to destination in its original quantum state? We still need repeaters, because there is still a maximum distance that a single point-to-point network segment can be, but at least when the repeater is used, it can keep the data in its original quantum state without decohering the data. Using a true quantum repeater also means allowing the inherent quantum properties to provide protection against unwanted eavesdropping. With a true quantum repeater, you get both accuracy and security beyond what a trusted repeater can provide on a quantum-based network.

True quantum repeaters use quantum teleportation to convey quantum information between segments. As covered in Chapter 5, quantum teleportation is an indirect way of using one or more entangled qubits to transmit quantum states. To recap quantum teleportation, first, the entangled quantum particles must be created and taken to their source and destination areas. Then an additional quantum particle(s) is added to the source quantum particles, and measurements are taken to record the differences. These differences are then transmitted (using any of many different classical methods) to the destination area and are used to reconstruct the wanted qubits using the destination entangled particles. Once the destination qubits are measured, the entanglement is broken.

NOTE As covered in Chapter 5, teleportation (at least currently, if not forever) does not allow faster-than-light transmissions because nothing can travel faster than the speed of light, not even quantum teleportation. Additionally, quantum teleportation always requires a classical transmission method to transmit the changes on the source side to the destination side, which means quantum teleportation will never be faster than classical methods by definition.

Quantum repeaters using entanglement have been successfully accomplished and are likely to be the top choice for future mature quantum networks. Of course, there are plenty of issues, not the least of which is that human-made entangled photons are exceeding easy to break. Early on we could not get entangled qubits to be more than a few millimeters away from each other, even though in the real world we are fairly sure entangled photons are light-years away from each other. Today, we have successfully demonstrated using quantum entanglement and repeaters across quantum networks up to at least 50 kilometers long.

It isn't easy. The types of entangled photons currently created in labs and on computers are often lost to decoherence, especially the longer they are sent along networks. They are more likely to disappear among the noise of the environment as the network lengthens. There is a whole subsection of quantum information sciences that studies *entanglement fidelity* and *entanglement optimization*. One solution takes an entangled photon coming from a trapped ion quantum computer, which would normally not last long when sent along a fiber-optic cable, and sends it through a specially designed crystal that converts the entangled photon's wavelength to something that is far more likely to be successfully transmitted. It turns out that the qubits created inside quantum computers aren't the best for transmitting across a network. This method converts the qubits into another state that is more agreeable for transmitting across a network without having to decohere the qubit out of its quantum state first.

Entanglement Swapping

Another commonly investigated quantum repeater technology is known as *entanglement swapping*. Essentially, it's an "If A trust B and B trusts C, then A trusts C" solution. Let's assume we have Nodes A and Z, representing source and destination, as shown in Figure 8.2. Node A has a quantum entanglement with the quantum repeater, which we will call R1. Node Z also has another quantum entanglement with another photon at the quantum repeater, which we will call R2. Node A teleports quantum information to the quantum repeater R using entanglement R1. The quantum repeater takes the same differencing information needed to conduct teleportation between itself and Node A, and re-transmits it between itself and Node Z. Node Z, using entanglement R2 and the transmitted differencing information, can reconstruct the information being transmitted by Node A.

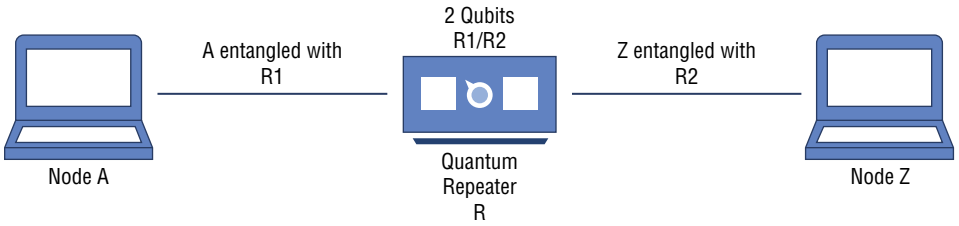

Figure 8.2: Representative entanglement swapping

There have been successful entanglement swapping experiments and it is likely to be the quantum repeater method of the future. Currently, useful entanglement swapping has been demonstrated up to 1.3 kilometers between nodes. With enough quantum repeaters and quantum swapping we could create something that very much looks like our Internet, but in a purely quantum state.

Quantum Network Protocols

Every network needs protocols. Network protocols are pre-agreed-upon methods and formats for transmitting data between two or more participating nodes. No one knows what the final quantum network protocols will look like yet, although there are some standards starting to be proposed and tested. A group of quantum researchers have created an official Internet draft (`https://tools.ietf.org/pdf/draft-dahlberg-ll-quantum-02.pdf`) defining the link networking layer of what might become the quantum Internet.

NOTE The (data) link-layer is a lower-level layer of a common abstract network protocol layers model. In the most common model, the Open Systems Interconnect (OSI) model, all networking communications can be defined as existing on one of seven stacked, inter-reliant, layers: physical, data-link, network, transport, session, presentation, and application. The link-layer helps get data (in the form of datagrams) to and from two directly communicating nodes. It handles error correction (that are due to the physical layer), and communication channel setup and tear down. In traditional networks, Ethernet bridges and switches function at this layer.

In quantum networks, the link protocols will be involved in helping two quantum nodes communicate, including using entanglement, and specifically entanglement swap (covered earlier). The protocol designers want to make it easier for any three participating nodes to initiate entanglement swap with the goal of more easily allowing a myriad of long-distance networks full of entanglement swaps, creating fully meshed, distributed, quantum networks similar to what we have today, although with more inherent defenses against eavesdropping.

One of the main tasks of the quantum link-layer is to help nodes transmit the intended entangled qubits to allow communication. As you remember from Chapter 1, creating/measuring quantum properties is probabilistic, meaning you can never completely predict the outcome of any single quantum operation. You can only predict the percentage likelihood of particular outcomes along a series of tries. So, say for example you need to send a "1" as quantum information along a network. You can't just deterministically create a qubit representing a "1" (as defined by the protocol) on the first try as you can easily do in a classical network. You can't say "I need a qubit representing a '1'" now and magically have the first qubit you create guaranteed to be a "1." You can, however, create one or more qubits until you get a qubit that represents a "1." Remember, this is made far more difficult by the fact that if you directly measure a qubit you decohere it into a nonquantum state. Similarly, you can't always be guaranteed to immediately create entangled qubits and the right entangled qubits on your first try. Quantum is probabilistic, not deterministic.

The quantum link-layer protocol was designed specifically to help tackle this problem. The physical layer will still have to originally create the right entangled qubits (it might take it one or more tries). The link-layer will guarantee that the qubits are the "right" qubits and that they are entangled. The link-layer will allow nodes to discard the "wrong" entangled qubits, re-create and re-transmit the "right" entangled qubits, and tell the nodes to distinguish between the "right" qubits and noise.

To do this, the link-layer does a few things, including creating and using a *heralding signal*, which indicates that an entangled qubit pair has been created and assigns a logical "entanglement identifier" to each entangled set of qubits. This allows the various nodes to keep track of the "right" entangled qubit pair as it is swapped between nodes. Higher quantum network layers can request entanglement pairs by sending a *CREATE* message; the link-layer then works with the physical layer to create the designed entangled pairs. The link-layer responds to the upper layers with an *ACK* (i.e., acknowledgment) or an *OK*. The ACK response tells the higher layer that the link-layer has accepted its request and has scheduled an entanglement pair for generation. It also includes a *CREATE ID* so everyone can keep track of the exact requested entanglement pair. An OK means the requested entangled qubits have been created.

There are even *goodness* and *time of goodness* values, labels I haven't seen in other network protocol standards (although to be fair I don't spend all my free time reading network protocols). The goodness value indicates how strong the entangled pair is (a value that the protocol and quantum entangle theory also calls *fidelity*). The time of goodness is an estimate of how long the entanglement should hold before decohering. This is important so the involved nodes understand how long they have to transmit/swap the entanglement to the other node before the entanglement bonds become unreliable. There are a lot of other fields and types of information sent back and forth between the link-layer and the upper layers. Figure 8.3 shows the structure and makeup of a Type K link-layer OK message as currently defined in the draft proposal. The quantum link-layer protocol was funded by *EU Flagship on Quantum Technologies*, part of the *Quantum Internet Alliance* (covered in the Quantum Internet section below).

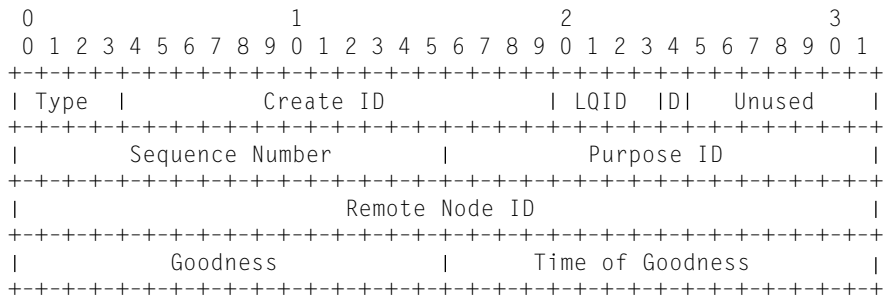

Figure 8.3: Structure and makeup of a quantum link-layer Type K OK message

The link-layer handles making quantum entanglement and swapping easy and reliable. The network-layer ensures that the transmitted entangled qubits are reliably transmitted from source to destination across the entire network. The transport-layer handles the qubits transported across the network, detached from the entanglement or network-layer, and hands the results off to the application layer. The proposed lower layers of the quantum network stack look something like Figure 8.4.

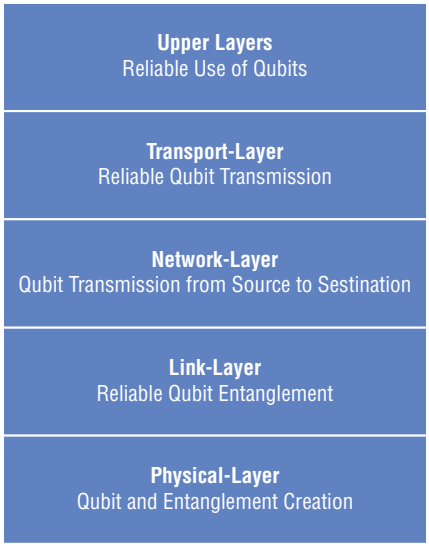

Figure 8.4: Proposed lower layers quantum network stack representation

The upper layers of the model are essentially application layers, and if the lower layers have done their job, the upper layers do not have to worry about the reliability of qubit transmission. They are needed and they appear. The destination devices and applications just grab them. Easy-peasy.

Of course, none of the quantum network stack is beyond proposed theory and individual experimentation yet. But there have been demonstrated network communications using proposed protocol messages exchanging information between different types of quantum devices. Overall, the idea that many people are already trying to figure out how to make quantum network communications, including entanglement and swapping, routine and reliable should give everyone great comfort. It not only shows you that quantum information sciences in general are maturing, but that as the technology allows, it will be reliable.

For more information on quantum network protocols see: `https://arxiv.org/pdf/1903.09778.pdf%3E`, `https://arxiv.org/pdf/1904.08605.pdf` and `https://tools.ietf.org/pdf/draft-van-meter-qirg-quantum-connection-setup-00.pdf`.

Entanglement Purification

Currently, quantum entanglement and swapping isn't an easy thing to accomplish. You start with quantum being probabilistic in the first place, and then you try to create entangled pairs that can then be sent over a network using QKD, trusted repeaters, or quantum repeaters. Nearly everything along the way (i.e., the external environment) is trying to defeat the transfer of quantum information in a useful way. Any errors that happen along the way and aren't caught at the physical or link-layers will be propagated along the rest of the network and network stack. One way to decrease errors is to use a link-layer, which handles error detection from the physical layer and uses re-transmission to help create reliable quantum entanglements and swaps along the entire network chain.

Another method is to create multiple, identical quantum entanglement pairs at the origination, with the idea that by creating more, at least one is more likely to end up at the destination. For example, perhaps the origination node creates 20 identical entanglement pairs, each representing the same qubit of information. Let's say 10 of the original 20 pairs end up at the destination, but 3 of the "identical" pairs are different. Most logical observers would say that the qubit value represented by the 7 identical pairs is more likely to correctly represent the original qubit property, when compared to the 3 pairs in the minority. The process of using multiple, identical qubit entangled pairs to create a more accurate network transmission is known as *entanglement purification*. The long-term goal is to "reduce" the number of needed duplicate copies into fewer copies with better *fidelity* (i.e., accuracy). There are many scientists working on entanglement purification and fidelity, much like quantum scientists are trying to reduce quantum computing qubit errors using quantum error correction—the end goal of which is to increase the length and reliability of quantum networks.

Quantum Network Applications

So why do we need quantum networks? What are the applications?

NOTE This section focuses on the strengths that come from quantum networking, not from individual quantum computers and the applications that run on them (which was covered in Chapter 5).

More Secure Networks

First and foremost, quantum networks, because they can't as easily be eavesdropped on, have built-in, inherent protections that classical networking simply can never have. This is not to say that quantum networks can't be eavesdropped on (nothing is unhackable), but that by default they are much harder to eavesdrop on as compared to classical networks. No doubt that humans will mess up implementing

quantum networks and leave lots of big vulnerabilities that hackers will find and exploit. But when those holes are closed, the default state of quantum networks will be harder to hack, and that is a huge bonus.

NOTE All of the world's governments and law enforcement agencies are fighting any default encryption channels that might make it harder to (legally) collect evidence of wrongdoing when conducting investigations. In much of the world, network providers *must* provide methods for law enforcement to eavesdrop on any network communication stream under the provider's control. It will be interesting to see how these eavesdropping requirements are handled when the underlying technology prevents eavesdropping by default. There is a chance that it could be impossible for a network provider or law enforcement agency to spy on someone, and there's potentially nothing they can do technically about it without getting rid of the quantumness at some point along the network. It will be interesting times and challenges for how societies balance quantum security and legitimate law enforcement needs.

Quantum Computing Cloud

Quantum computing promises to solve problems and issues we have sought to solve for millennia. It will give us the ability to better understand the nature of our universe, to predict the future, and to create products and services we cannot currently imagine. Now imagine how great it would be to collect as many quantum computers as possible and have them work on the same problem(s) all at once. It would be like having many "Albert Einsteins" working together instead of just one. A quantum computing cloud is a collection of quantum supercomputers networked to synergistically work together to solve more problems faster. Quantum computing cloud is about faster collaboration.

Better Time Syncing

There are many applications that require very accurate time. We already have very, very accurate time clocks. They are known as "atomic clocks" and already they are so accurate that they lose less than a second every billion years (`https://en.wikipedia.org/wiki/Atomic_clock`). One of the remaining challenges is in getting the most accurate time to a reliant device very far away or in ensuring that every reliant device in a particular service or network has the exact same time. It takes time to transmit time.

Quantum networks don't offer faster time transmittal (remember, classical things are already going at nearly the speed of light and quantum mechanics will likely not beat the speed of light unless our fundamental understanding changes). But what quantum networks can do is offer better time synchronization between reliant devices. Because quantum devices and networks can consider more time synchronization factors when doing time error correction/synchronization, the synchronize time drift will be less across any big network.

For example, our Global Positioning System (GPS) works using a series of orbiting satellites that can be used by GPS receivers to determine their geographic position. In order to correctly determine where a GPS receiver is, both the GPS satellites and receivers must have synchronized time. The more accurate the time keeping is on all devices and the more accurately it is synchronized with all involved parties, the more accurate a GPS receiver can be in detecting its position.

GPS satellites have atomic clocks onboard, and as accurate as they are, without time synchronization they could lose up to 10 nanoseconds every 24 hours (`https://www.insidescience.org/news/quantum-way-synchronize-atomic-clocks`), and a nanosecond equates to about a foot of difference in distance. Because of this, GPS clocks' time is updated so that it does not lose even a single nanosecond in a year. The GPS time synchronization service is constantly matured and updated to keep the satellites as accurate as possible. Back in 2000, a GPS receiver could only be accurate to within 5 meters/16 feet. Today, because of improvements in GPS technology, including time synchronization, GPS receivers can track their position to within 30 centimeters/11.5 inches. Quantum time synchronization will only make the GPS system even more accurate.

Although all atomic clocks ultimately work on quantum mechanics, the time synchronization network that GPS satellites use is a classical one. This means the tiny quantum properties that are actually doing the real work have to be decohered into our classical world and then transmitted classically. But quantum time synchronization skips the conversion step, stays quantum, and allows the time synchronization to be even more accurate.

This means that the GPS readings of the future may be able to be measured in millimeters instead of centimeters, or at least only a few centimeters instead of 30. All-in-all, this means everything that relies on GPS can more accurately know where it is more often. It not only means you won't miss your left-hand turn while driving, but quantum systems will be more accurately able to keep track of the hundreds of millions of self-driving cars we all will be in within a decade or two. It means people walking in cities with tall buildings will be more accurately able to determine which door they are standing in front of that they should go in. Engineers will be able to more accurately make measurements, and so on. Quantum-based time synchronization will make every reliant time service more accurate and successful.

NOTE Interestingly, for many years, quantum clocks were considered the most accurate atomic clocks possible, but optical lattice clocks have now supplanted them for accuracy records. But even timing devices based on optical lattice clocks can benefit by quantum time synchronization.

Prevent Jamming

Even before the 1940s when Austrian/American actress Hedy Lamarr and her co-inventor George Antheil invented a new way to prevent military jamming of wireless torpedoes (they invented *frequency hopping spread spectrum* as a jamming defense), the world's militaries have tried both to

jam an enemy's wireless signals and to make their own communication method more impervious to jamming. In general, anti-jamming defenses incorporate control instructions into a series of modulated frequency/signal changes, which the enemy cannot figure out how to simulate or block.

Jammers try their best to figure out how a signal is being communicated to a controlled device. If they can figure out all the rapid frequency changes, they can reconstruct the control instructions, learn what they are, and even possibly reprogram a device to use their own. In war, an opponent might be able to send an enemy's missile or torpedo back toward the sender. This isn't conjecture. In World War II, it happened all the time.

A jammer doesn't even have to figure out how the enemy's control signal works. All they need to do is block all possible involved frequencies, and it would prevent a wireless device from getting new or updated instructions. In most cases, a jammed device, such as a missile or flying drone, will go into some default mode, such as keep flying on the last course position or fly back to home. Today's military systems are full of jamming and anti-jamming technology.

Weapons developers keep trying to make their weapons more resistant to jamming, and jammers keep coming up with new ways to jam their control systems. It's slightly akin to how a cryptographic attacker might attempt to figure out a secret code and a cryptographer attempts to improve their cipher to prevent successful attacks.

Quantum networks, because they have the inherent properties of superposition, entanglement, and no-cloning, are perfect for creating jam-proof signals. A quantum-enabled device using a quantum-based network will be far more likely to be able to create more sophisticated, jam-resistant control signals and ignore an enemy's background jamming signal. So even if the enemy sends a million erroneous frequency changes across the thousands of possible frequencies, over and over in a jamming attempt, a quantum-based device is more likely to be able to figure out which commands are the legitimate ones and which to simply ignore as background noise.

Preventing jamming isn't just for military uses. It also prevents attackers from jamming cell phone conversations and, Wi-Fi network traffic, and can even stop attackers from trying to send erroneous instructions to self-driving cars. Preventing jamming makes all of our lives easier, and quantum networking can help do that.

Quantum Internet

The Holy Grail of quantum networking is a one-for-one replacement for the existing classical Internet. And like the classical Internet, a quantum-only Internet would likely start out as a hodge-podge of separate networks, beginning from well-funded military, government, and university networks, and then spread out to help connect everyone else. The millions of Internet-connected quantum devices would eventually form a huge, global computing cloud. It would be harder to compromise, harder to jam, and would allow our world to enjoy all the quantum improvements we are destined to see together, just as today's Internet has already done for us using classical networking.

NOTE A program called SimulQron (http://www.simulaqron.org/) is a simulator for helping to develop quantum Internet software (https://arxiv.org/pdf/1712.08032.pdf).

Several teams and consortiums are working toward a quantum Internet, including the European Union's Quantum Internet Alliance (https://twitter.com/eu_qia) project. The Quantum Internet Alliance is starting with a four-city demonstration project in the Netherlands, including the cities of Amsterdam, Leiden, The Hague, and Delft. It includes all the components needed to create a large quantum-based wide area network and is slated to be operational by 2020. They then want to expand the project to all of the EU.

Other Quantum Networks

Currently, all existing quantum networks are fairly limited and experimental, even as more gains are being made each day. In 2018, fiber-optic-based QKD was successfully demonstrated at 421 kilometers/261 miles. Most existing QKD systems for commercial sale have maximum transmission distances of 100 kilometers/62 miles, so this experimental network is over four times that. Expect quantum QKD systems to keep expanding the distance they can transmit.

China has been especially focused on the networking aspects of quantum information science. In 2016, China launched Micius (https://www.bbc.com/news/world-asia-china-37091833), the first quantum-based communications satellite. As the satellite flew overhead, it used QKD to securely send individual secret keys to ground stations in Beijing and Vienna, 7,500 kilometers/4,700 miles apart. Then it created a third key to be shared among all parties. It encrypted the third key with each of the ground station's previously transmitted individual secret keys. Each ground station could then decrypt the shared secret, third key, and begin encrypting messages to and from each other. It was a landmark moment.

Micius can only operate with line-of-sight and cannot operate in sunlight; still, it was used to successfully demonstrate a 75-minute video conferencing call, which is more than an adequate demonstration for a first generation, first-of-its-kind quantum satellite. You can read more about Micius here: https://www.wired.com/story/why-this-intercontinental-quantum-encrypted-video-hangout-is-a-big-deal/, https://cosmosmagazine.com/technology/the-quantum-internet-is-already-being-built and https://en.wikipedia.org/wiki/Quantum_Experiments_at_Space_Scale. China is also creating a 2,000-kilometer/1,200-mile quantum-based communication backbone between Shanghai and Beijing. It connects 4 cities and has 32 nodes.

NOTE Some quantum information scientists are skeptical of China's quantum networking claims, or doubt that it was 100% quantum-based.

Japan has demonstrated a quantum network using quantum repeaters (https://qiqb.otri.osaka-u.ac.jp/wp-content/uploads/2019/01/AllPhotonicQR-QIQBen.pdf). It used optical devices as quantum repeaters (known as *photonics*), which allowed it to send quantum information without needing classical components and also without using quantum memory.

A United States team is working on building a 48-kilometer/30-mile quantum network using quantum entanglement and teleportation (https://spectrum.ieee.org/tech-talk/telecom/security/us-national-labs-join-forces-on-a-quantum-network). Most of the other long-distance quantum networks used QKD, but this U.S. national lab funded project, spearheaded near Chicago, will be one of the first to use teleportation. The U.S. Department of Energy provided several millions of dollars. The national labs (Argonne National Laboratory and Fermi National Accelerator Laboratory) along with University of Chicago formed an official entity known as the Chicago Quantum Exchange (https://quantum.uchicago.edu/). It involves over 100 quantum information science researchers.

Quantum-based networking is already up and working in the experimental stages, with several real-world projects and devices coming online in the next few years.

For More Information

Quantum Network, Rodney Van Meter, Wiley, 2017, https://www.worldcat.org/title/quantum-networking/oclc/879947342

The Quantum Internet Is Emerging, One Experiment at a Time, Anil Ananthaswamy, Scientific American magazine, June 19, 2019, https://www.scientificamerican.com/article/the-quantum-internet-is-emerging-one-experiment-at-a-time/

The Quantum Internet, H. J. Kimble, June 25, 2008, white paper, https://arxiv.org/pdf/0806.4195.pdf

A Poisson Model for Entanglement Optimization in the Quantum Internet, Laszlo Gyongyosi and Sandor Imre, Quantum Information Processing, June 5, 2019, https://link.springer.com/content/pdf/10.1007%2Fs11128-019-2335-1.pdf

The quantum internet has arrived (and it hasn't), Davide Castelvecchi, Nature Magazine, February 14, 2019, https://www.nature.com/articles/d41586-018-01835-3

Internet Research Task Force (IRTF) research group on quantum Internet called QIRG. Qirg mailing list submissions to qirg@irtf.org, To subscribe or unsubscribe via the World Wide Web, visit https://www.irtf.org/mailman/listinfo/qirg.

Summary

Quantum networking will progress and mature much like the rest of quantum information science, in fits and spurts as underlying technology advances. Quantum networks will start with more classical pieces (like trusted repeaters) and use QKD to exchange secure keys and then move to a complete

quantum-based network stack using true quantum repeaters, quantum entanglement, teleportation, and quantum swapping. The limited, experimental networks will soon be replaced by real-world, working, quantum networks. Eventually, much, if not all, of our classical Internet will be replaced by quantum networks for a myriad of reasons, not the least of which is the stronger inherent security that quantum mechanics provides.

All the previous chapters covered quantum mechanics, computing, cryptography, and why quantum computers are likely to soon break today's public key cryptography. Part II began by covering quantum-resistant and quantum-based cryptography. This chapter covered quantum networking components, challenges, and likely applications. Chapter 9 will cover how all stakeholders should prepare for the coming quantum break and revolution.

9 Preparing Now

The coming quantum cryptographic break will happen; it's only a matter of when. When it does happen, it will invalidate much of the world's traditional public key cryptography and weaken other existing cryptography by at least 50%. And this is a best-case scenario that does not include any other quantum advances that could make solving symmetric ciphers and hashes even easier.

In the previous chapters, we discussed quantum mechanics, quantum computers, networks, and the coming changes, including the likely cryptographic breaks. This chapter discusses how you and your organization can start preparing today, before the break has happened. This chapter is likely the reason many of you bought this book. First, we will cover the four major stages of any post-quantum mitigation project, and then we will focus on the project steps.

Four Major Post-Quantum Mitigation Phases

Most organizations' post-quantum mitigation projects will include these four major stages:

- Stage 1: Strengthen current solutions.
- Stage 2: Move to quantum-resistant solutions.
- Stage 3: Implement quantum-hybrid solutions.
- Stage 4: Implement fully quantum solutions.

Figure 9.1 shows each stage represented graphically. Each project stage will be discussed in more detail in the following sections.

Stage 1: Strengthen Current Solutions

Every organization should, as soon as possible, update any weakly quantum-resistant cryptography and use existing quantum-resistant cryptography and key sizes where feasible. Quantum computing using Grover's algorithm halves the protective power of existing symmetric ciphers and hashes, so doubling their key and hash output sizes, especially where it is easy to do so, makes sense. For

Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto, First Edition. Roger A. Grimes.

instance, any systems using AES-128 should be moved to AES-256 or greater. If you've got critical data that needs to be protected for 10 years, even if it is using AES-256, maybe you should move it to AES-512, and so on.

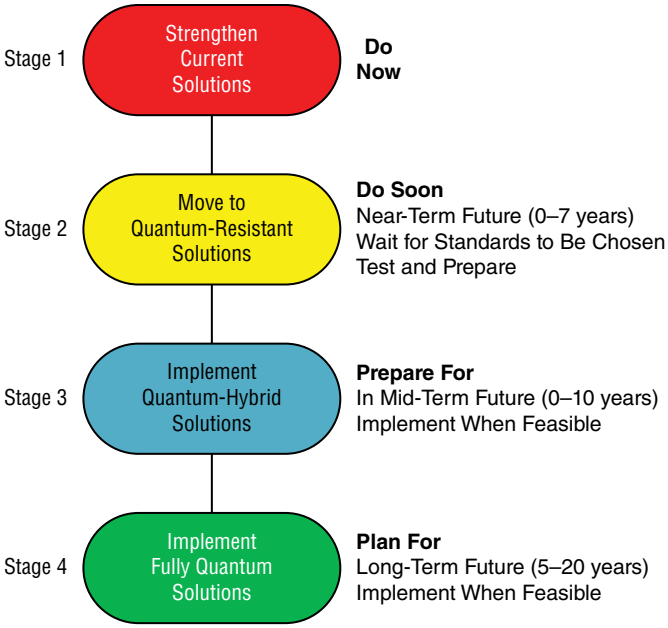

Figure 9.1: Four major post-quantum mitigation project stages

The key sizes of asymmetric ciphers should be updated to at least 4,096 bits. Most public key ciphers are 2,048 bit, with plenty of 1,024-bit ones still around. As quantum computers gain qubits and capability, they will be capable of cracking the smaller key sizes first. By moving to the larger key sizes of your existing cryptography, you reduce risk, although not nearly as much as if you were able to immediately transition to true quantum-resistant cryptography. Change your organization's policy to dictate minimum acceptable key sizes.

With this said, it will not always be possible to go to larger key sizes for existing cryptography. Many applications are hard-coded and many others can accept only particular key sizes (and not all key sizes as published in the standard). For example, SHA2 comes in 224-, 256-, 384-, and 512-bit sizes. For many years Microsoft Windows could flawlessly use SHA2-256 (the default SHA2 key size) but would have operational issues on TLS-protected websites if you went to SHA2-512 (this "bug" was fixed years ago). Windows still does not offer SHA2-224 by default (even though it's part of the official standard) when you use its built-in software. Many applications accept only one key size or maybe two. And some will accept the larger key sizes but have unexpected operational issues. So, increasing key sizes should always be done after thorough testing to make sure doing so does not result in operational issues.

You also have to be aware that moving to the largest possible key sizes can cause performance issues. Every bit you add to a cryptographic key increases the computations needed to use the cryptography. In many cases, such as moving from RSA 2,048-bit to 4,096-bit, the performance hit is there but almost unnoticeable to most users and in most scenarios. But in some high-transaction scenarios, such as popular websites or high-transaction databases, the change could become noticeable and have an adverse effect as traffic increases.

In some scenarios, the performance hit could be unacceptable for even a single user and low traffic. For example, years ago one of my customers moved his public key infrastructure (PKI) certification authority server's digital certificate from 2,048 to 16,384 bits, for no reason other than he wanted to be as secure as possible. The bit increase was so significant that it took most of his computers over a minute to open any encrypted message, which ultimately chained up to the 16 KB root certification authority digital certificate. Compare that to the 1–1.5 seconds to open the same message using a 2,048-bit key. The larger digital certificate was secure, but likely excessively secure and definitely too slow for the hardware and applications being used—especially when using 4,096-bit keys would have been both secure enough and fast.

Should Asymmetric Key Sizes Be Increased?

Some readers may be wondering if it makes any practical sense to upgrade asymmetric key sizes to try to stay ahead of expected continuing increases in quantum computer capability. The answer isn't an easy yes or no, although it cannot hurt to increase asymmetric key sizes to at least 4,096-bit, especially if you can do it naturally and easily as older keys expire.

But there are good arguments both for and against increasing asymmetric key sizes to fight the coming quantum crypto break. For example, using Shor's algorithm, a quantum computer needs at least $(2 \times n) + 3$ stable qubits, where n is the number of key bits to crack. So, to crack a 2,048-bit RSA key, a quantum computer needs 4,099 stable qubits, and to crack a 4,096-bit RSA key, it needs 8,195 stable qubits. Thus, if you move your asymmetric ciphers from 2,048 to 4,096 bits, that strategy would protect you until quantum computers got to at least 8,195 stable qubits (assuming using fewer qubits doesn't allow lesser quantum computers to still get there relatively fast).

At the present time, we have no idea how long it will be before quantum computers get to 4,099 stable qubits and, once there, how long it will take to get to 8,195. But it probably won't take as long as it took us to get to the first 100 qubits, or from 100 to 4,099. Once society learns how to massively scale qubits, the increases should come quite fast.

To add considerations and complexity, if some error correction estimates are to be believed, each stable qubit currently takes hundreds to over a million error-correcting qubits, so we could be talking about a qubit difference in the many billions needed to crack a 2,048-bit key versus a 4,096-bit key. If the error correction estimations are true, the number of ancillary qubits needed is likely to provide a longer period of protection and is probably worth the move.

At the same time, many developers of newer quantum factoring algorithms are claiming that they can factor prime number equations with far fewer qubits than what Shor's requires. And if individual

qubits get more stable, or far less error-correction qubits are needed than the higher echelon estimations are calculating, it argues back in the other direction.

I don't share these arguments to confuse you, but only to say that increasing the number of bits of an asymmetric key isn't as straightforward as increasing symmetric key and hash sizes. But in general, if you want to reduce the risk of your traditional quantum-susceptible asymmetric cryptography until you are able to migrate to quantum-resistant cryptography, it can't hurt to increase asymmetric key sizes, especially if it is easy to do so.

Crypto-agility

Crypto-agility is the ability of a device, software, or system to have its cryptography changed out with another cipher, scheme, or key size without an undue burden. Ultimately, all involved systems should be designed so that you can switch out the involved cryptography with as little effort as possible. Unfortunately, this is something that must be done by the developer of those systems. It is difficult for customers and end users to accomplish without the developer first creating the underlying structure to allow it to happen.

Some popular developers have already done this for you. For example, Microsoft Windows and most Microsoft products separate the cryptography from the software and hardware systems that use it. Microsoft does this by recommending that cryptographic ciphers and schemes be represented in discrete, individual key storage provider (KSP) modules (they used to be called cryptographic service providers [CSPs] in earlier Windows versions) that can be installed and uninstalled separately from the applications that use them.

Windows comes with many built-in KSPs, which contain all the popular cipher and scheme standards. Third parties and customers can create their own, and swapping one KSP for another is as easy as installing the new one (which is usually fairly small and quick to install) and selecting it from a KSP pull-down menu. Microsoft's flagship certificate authority product, Active Directory Certificate Services, allows different KSPs to be installed to support different types of cryptography.

In the open-source Linux world, many of the most popular applications and utilities, like OpenSSL and SSH, allow different types of cryptography to be interchanged. Part of the quantum-resistant Picnic signature scheme team used Picnic, LWE-FRODO, and SIDH with OpenSSL and Apache Web Server to create valid TLS 1.2 HTTPS connections (see section 8.2 of the Picnic design document: `https://github.com/Microsoft/Picnic/blob/master/spec/design-v1.0.pdf`). OpenSSL and Apache did not require significant modifications, although OpenSSL did require a minor modification to allow TLS to use the larger key sizes generated by Picnic.

Compare that easy versatility (i.e., crypto-agility) with most applications' hard-coded cryptography, which cannot be replaced without updating the cipher coding in program, recompiling, and reinstalling the whole program.

Strive to get all your vendors (and your own developed software) to be crypto-agile. That way, when the next forced crypto migration event happens, it will be easier to make the transition. Begin this stage of your quantum crypto migration project now. Make crypto-agility a word that everyone on the IT team knows, understands, and requests.

Stage 2: Move to Quantum-Resistant Solutions

As covered earlier, most organizations cannot move to the new quantum-resistant cryptography until their national standards body (such as the National Institute of Standards and Technology [NIST] or the European Union Agency for Network and Information Security) declares the official post-quantum cryptographic standards. But it makes sense for many organizations, especially those with developers and their own internal applications that use cryptography, to begin experimenting.

Many quantum coding libraries, APIs, simulators, and software development kits (SDKs) are available that can assist with an organization's move to quantum-resistant cryptography. Many organizations have software, tools, resources, and knowledgeable people to help you with your quantum-resistant cryptographic transition. They may have everything your organization needs to begin its quantum cryptographic migration, including experienced quantum developers and testers. Check these out:

- Open Quantum Safe Project (`https://openquantumsafe.org/`)
- Open Source Quantum Software Projects on GitHub (`https://github.com/qosf/os_quantum_software`)
- Open source and commercial quantum software projects and online quantum portals (`https://github.com/qosf/os_quantum_software`)

And, of course, quantum vendors such as Microsoft, IBM, and Cambridge Quantum Computing have many resources and tools. Work with your existing vendors. Even if all you do is conduct one small demonstration project, this gives you a leg up on the competition. If the quantum crypto break happens sooner than your plan estimates, the time, effort, and resources put into any test projects will be worth their weight in gold. Test projects are about getting the bus and the people on the bus going in the right direction.

When nation-state bodies approve the official quantum-resistant standards, you should start to move toward quantum-resistant cryptography as quickly as you can. The same operational and performance caveats apply as updating the key sizes of your traditional cryptography (performance and operational issues), but any selected standard is likely to be a good combination of performance and usability. Usually by the time the standard is approved, lots of real-world resources and software libraries are waiting to help developers and implementers. Once the standards are announced, the whole nation (or world) will begin to move in the same direction. Make sure you are part of that movement. You don't need to be on the bleeding edge, but near the edge is a great place to be.

This stage of your migration project will probably last at least two years from the time the national quantum-resistant standards are chosen. NIST claims the U.S. national standard will be chosen sometime between 2022 and 2024, which means this stage of most U.S. projects will probably take 3–7 years before completion. Of course, this project stage could be moved up if someone suddenly announces the quantum crypto break ahead of current time expectations.

NOTE Completing stage 2, moving to quantum-resistant solutions, will result in a major decrease in risk, at least until another technological breakthrough happens to mitigate the provided strength of quantum-resistant cryptography. The next two stages will continue to reduce risk, but the reduction will likely not be as drastic. Completing stage 2 will likely be the top project milestone.

Securing PKI

PKI makes most of the Internet and businesses run. The coming quantum crypto break specifically breaks every existing PKI, because they run on quantum-susceptible crypto (Rivest–Shamir–Adleman, Diffie–Hellman, Digital Signature Algorithm, Elliptic Curve Digital Signature Algorithm, etc.). As covered previously, Microsoft and other researchers have been able to rudimentarily show that at least some existing PKIs, digital signatures, and hardware security modules (HSMs) can run on quantum-resistant ciphers, and with a little tweaking those digital certificates can be used on the Internet with TLS. There are many other PKI software programs and vendors, but when the time comes for them to move to quantum-resistant ciphers, the majority will likely be ported to do so.

Any of the NIST Round 2 digital signature scheme candidates covered in Chapter 6 (CRYSTALS-Dilithium, FALCON, GeMSS, LUOV, MQDSS, Picnic, qTESLA, Rainbow, and SPHINCS+) are in strong consideration for the post-quantum digital signature PKI replacements. You'll find a good white paper on PKI crypto-agility here: `www.isara.com/wp-content/uploads/2018/05/ISARA_Corp_PKI_Migration_WhitePaper_FINAL.pdf`.

Leighton-Micali Signatures (LMS) and XMSS (eXtended Merkle Signature Scheme) variants are being discussed as well, but because they are stateful, they were dropped as official NIST candidates. They are both purportedly still being considered by NIST in a subproject. A multitree variant of XMSS is known as XMSS-MT or XMSS-MTS (Merkle Tree Signature), and a multitree variant of LMS is known as HSS. Both have been submitted as requests for comments (RFCs), which is the last step in the approval process, to the Internet Engineering Task Force (IETF). You can read more about XMSS and LMS at `https://eprint.iacr.org/2017/349.pdf` if you are interested. A good place to stay up to date on XMSS and LMS progress is `https://www.isara.com/standards/`, and you'll find a good white paper on XMSS here: `https://tools.ietf.org/html/draft-irtf-cfrg-xmss-hash-based-signatures-11`.

Give your strongest post-quantum PKI attention to the announcements from the CA/Browser Forum (`https://cabforum.org/`) and in particular their PKI standards known as Baseline Requirements (`https://cabforum.org/baseline-requirements/`). This group is made up of some of the biggest PKI vendors and implementers. What they require usually controls what all "public" CA vendors follow and indirectly what most private companies end up following. They control what cryptography and best practices are used with public CAs, along with most factors related to PKI operations. They were the group that successfully forced most CAs to migrate from SHA1 to SHA2; that move was heeded and has led other cryptographic migrations in a thoughtful, successful way.

CA/Browser Forum discussions and awareness around the coming quantum changes took place in March 2019 (https://cabforum.org/2019/05/03/minutes-for-ca-browser-forum-f2f-meeting-46-cupertino-12-14-march-2019/#Quantum-Cryptography-problem-need-solutions-and-timeframe--assign-ForumSCWG-liaisons). If you have a PKI, follow the updates and requirements of the CA/Browser Forum.

For more information on PKI and X.509 digital certificates in the post-quantum world, check out https://eprint.iacr.org/2018/063.pdf and https://eprint.iacr.org/2017/349.pdf.

Get Other Quantum Devices and Services

This is also the time to start considering getting quantum-based (certifiable) random number generators (RNGs) and quantum key distribution devices (as covered in Chapter 7). Both of these devices are relatively inexpensive, have been in use for almost two decades, and can be used to improve your cryptography with or without other quantum cryptography or devices involved.

NOTE Quantum-based random number generators have been around for a long time, but "certifiable" RNGs are just starting to show up in 2019. Here's the first commercial version available: https://cambridgequantum.com/cqc-unveils-the-worlds-first-commercially-ready-certifiable-quantum-cryptographic-device/.

Now is the time for IT security to decide when they need to get these types of systems, but they will be required around the time you are implementing quantum-resistant cryptography. If you don't need or want to own this type of system, it makes sense to at least avail yourself of the services of one. As quantum computers start to gain access, consider buying quantum computers or buying quantum-based services. Don't get caught remaining in a pure binary world as the world starts to go quantum.

Stage 3: Implement Quantum-Hybrid Solutions

Most early quantum-involved networks and systems will use a combination of quantum and classical, binary computing devices to perform their work. For example, quantum key distribution (QKD) devices usually transfer quantum-derived keys across a classical, binary network channel. Most quantum-based RNGs create incredibly random numbers that are then used in classical binary-based systems. Trusted repeaters, which securely share quantum-based cipher keys, do so across classical networks. All existing quantum-resistant ciphers work in the binary world and use classical, binary-based hashes. And so on. Many of the early quantum-involved systems and devices you will use will be a mix of quantum and classical traits. This should be expected, especially before we are able to go to fully quantum-based networks and devices in the long run.

One important point is to figure out when it makes security and fiscal sense to move from the classical binary world to the quantum-hybrid model. Moving to a quantum-hybrid model, when

available, will not always be a cost-effective solution. The whole reason for moving to a quantum-based solution is to get the built-in protection from quantum mechanics. When it's mixed up with the classical world, as it is in hybrid models, then the maximum security protection is the easiest-to-hack technology (i.e., classical). The classical reliance could substantially negate the security reasons for going to a quantum-hybrid model and usually at a far greater cost.

There will be times when a purely classical solution will provide all the necessary security for the needed time period. It is up to each project team to decide on the right time to move and to which type of technology. Moving to quantum-hybrid systems is expected to take longer than moving to all quantum-resistant cryptography, perhaps lasting as long as 10 years from your project start.

Stage 4: Implement Fully Quantum Solutions

Lastly, the ultimate objective, many years down the road (hopefully) is to go fully quantum on both cryptography and network devices. Or perhaps your organization's security requirements need fully quantum-based systems as soon as possible. Quantum mechanics makes anything using quantum cryptography and mechanics inherently more secure. Quantum-resistant cryptography will eliminate most of the risk from quantum cryptographic attacks, but quantum-based cryptography and devices are the ultimate protection.

For example, regarding your network devices, start today to ensure that all your network devices are using quantum-resistant key sizes of existing traditional cryptography. Move them to true quantum-resistant cryptography as the national standards are approved. Then move to classical/quantum solutions, like trusted repeaters, and then to fully quantum solutions, like quantum repeaters. And do this for all your data protection systems. Stage 4 might last up to 20 years from the start of your post-quantum migration project.

These four stages do not have to happen sequentially. There are likely to be several parallel tracks going on at different times involving different systems. For example, you may want to stay at stage 1, with larger implementations of existing cryptographic standards for many systems for the foreseeable future, while at the same time exploring other stages for other systems. You may decide to buy a quantum RNG for some projects and use it in a hybrid solution. You may still be involved with transitioning a bunch of projects to quantum-resistant crypto when some vendor comes up with a fully quantum-based solution at a reasonable cost. Except for the first stage, these stages are likely to occur sequentially in some cases and parallel in others depending on the scenario. It's to be expected.

The Six Major Post-Quantum Mitigation Project Steps

With the coming quantum cryptographic break, every organization wanting to protect digital secrets needs to have a plan. This section of the book describes that plan. The major general project phases are as follows:

1. Educate.
2. Create a plan.

3. Collect data.
4. Analyze.
5. Take action/remediate.
6. Review and improve as needed.

Figure 9.2 summarizes the overall post-quantum migration project life-cycle phases.

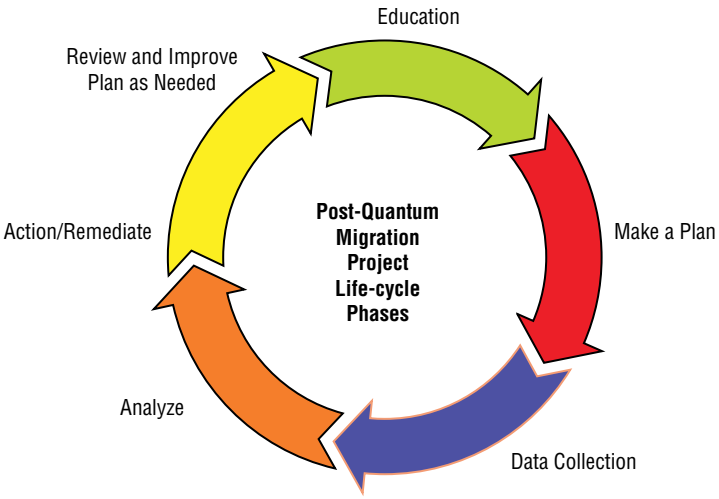

Figure 9.2: Post-quantum migration project life-cycle phases

NOTE The plan presented in this chapter has been tested and used many times before, although against a different type of cryptographic migration. I have helped many dozens, if not over a hundred, different companies perform SHA1 to SHA2 migrations during 2014 to 2017. Although the goals are different, the plan and steps are very similar.

The rest of the chapter is dedicated to discussing each step of the plan in more detail.

Step 1: Educate

Because you are reading this book, you are already on the first step of the plan. You need to educate yourself, your team, management, and every other end user about the coming quantum cryptographic break and what your organization plans to do about it. You especially want to educate developers and stakeholders who are involved in the decisions to make and buy software and hardware.

For yourself, use this book and all the additional references it recommends to continue your quantum computing education journey. A key challenge for you is to stay up on the latest changes in quantum computing advances and how they will impact your quantum computing preparation plan. Simply paying more attention when you see the word *quantum* in general media news articles

will help you, but subscribing to specific quantum computing mailing lists and blogs is also a great way to stay up to date.

Quantum Computing Mailing Lists and Blog Sites

Here are some possible quantum computing mailing lists and blogs you can join or follow:

- https://quantiki.org
- www.scottaaronson.com/blog/
- www.quantiki.org/wiki/mailing-lists
- https://golem.ph.utexas.edu/category/2010/12/quantum_foundations_mailing_li.html
- https://accounts.eclipse.org/mailing-list/quantum-computing-wg
- https://hepsoftwarefoundation.org/workinggroups/quantumcomputing.html
- https://dabacon.org/pontiff/
- https://quantumcomputingreport.com/news/
- https://quantumcomputing.stackexchange.com (great for technical questions)
- https://geekforge.io
- It can't hurt to follow my articles (www.csoonline.com/author/Roger-A.-Grimes/), Twitter (@rogeragrimes), or LinkedIn (www.linkedin.com/in/rogeragrimes/) postings, although they deal with a broad range of computer security topics and not just quantum.

The appendix following this chapter will list even more resources. I apologize if I missed your favorite quantum mailing list, blog, or website.

Attempt to understand quantum mechanics and quantum computing as well as you can. By being an advocate for preparing for the coming quantum crypto break, you might be expected to understand quantum fairly well. Hopefully, this book gave you a good summary understanding on all things quantum and quantum computing, but I understand if you would like to read additional resources. It took this author over two decades of experience and reading hundreds of very different articles on quantum mechanics and computers to understand it as well as I do today.

Here's a list of online courses dealing with quantum theory and quantum computing: https://quantumcomputingreport.com/resources/education/ along with another similar site: https://hackernoon.com/16-best-resources-to-learn-quantum-computing-in-2019-e5d8b797aeb6.

Slide Presentation

All IT members should be familiar with the basic technological issues and challenges. Involve all end users, to a lesser degree, if the coming changes will impact their lives (such as software updates and data protection standard changes).

A good way to introduce others to quantum mechanics, computing, and the coming cryptographic break is to cover the topic with a slide presentation. I have been giving a one-hour slide presentation for years with much success, and I welcome readers to download and reuse my own introductory slide deck. Figure 9.3 shows an example slide. The slide presentation covers quantum mechanics,

quantum computers, cryptography, quantum supremacy, the quantum break, and how to prepare. Viewers will need at least a little general cryptographic knowledge to get the maximum benefit from the presentation.

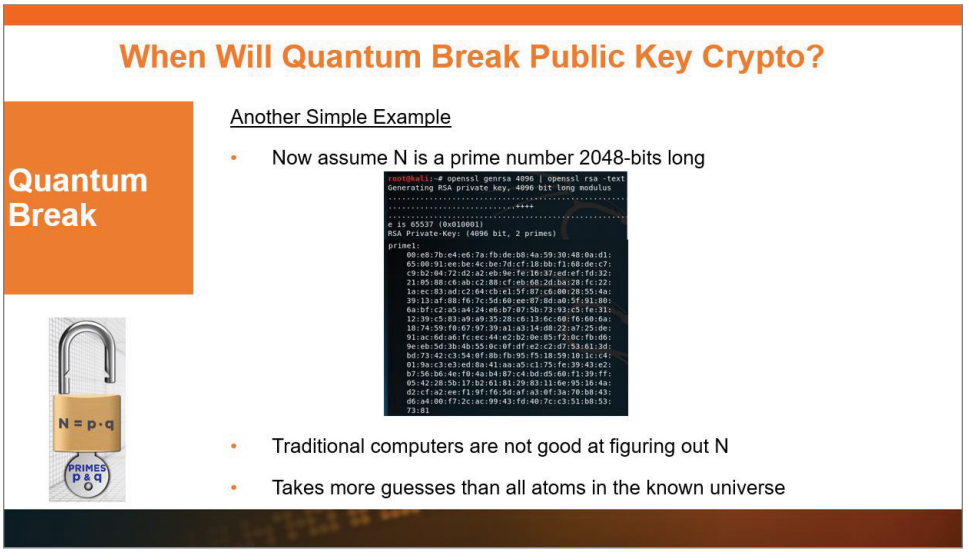

Figure 9.3: Example slide from presentation

The general presentation, in both Microsoft PowerPoint and Adobe PDF formats, is available at `www.wiley.com/go/cryptographyapocalypse`. Feel free to borrow anything you like from the presentation and customize it for your audiences.

Management Two-Page Briefer

Senior management is busy. There is a good chance that they have not heard, or not heard a lot, about the coming quantum crypto challenges. You should make senior management aware of the issue and its importance. It may make sense to create a short, one- to two-page briefing document that you can give to senior management along with a short slideshow (maybe 5 to 10 slides) that covers the basics. The one- to two-page briefing document is typically a short introduction memo followed by some frequently asked questions (FAQs) and their answers. You can download an example at `www.wiley.com/go/cryptographyapocalypse` and/or read it here:

To: Whom It May Concern

From: <you>

Date: <date>

Regarding: Preparing for Coming Quantum Cryptographic Break

This document was created to help introduce management to a new, evolving project titled the Quantum Resistant Data Protection project, and to share the general details and projected timeline.

Computers based on quantum mechanics are maturing to a point where they seriously threaten to compromise much of today's existing traditional cryptography, including HTTPS, Wi-Fi networks, logon authentication, smartcards, multifactor authentication, and public key infrastructure (PKI). No one knows exactly when quantum computers will mature to the point of being a real threat to most organizations, but estimates range from a few years to less than 10 years. In 2016, the National Institute of Standards and Technology (NIST) and the National Security Agency (NSA) recommended that all organizations start to prepare for the coming cryptographic break. We are [X] years late in beginning our preparation.

As part of following those recommendations and due to ever increasing improvements in quantum computing, we are preparing this organization by creating a special project team to address the issue. We will be doing a data protection inventory (to determine which of our critical digital assets need long-term protection against unauthorized access) and creating a plan to ensure that critical date is protected by the appropriate quantum-resistant cryptography and other mitigations.

The first major phase of the project, along with the work of our newly formed project team, will begin in the next few weeks and is expected to last multiple years until the threat is fully remediated. Our goal is to upgrade our quantum-susceptible cryptography to quantum-resistant forms before the quantum cryptography break occurs.

It is likely to impact many of our existing data protection implementations, and our reason for beginning this project now is to minimize future business disruption and costs. Overall project costs, resources, and timelines cannot be adequately estimated until after the Data Protection Inventory and Analysis tasks, which are expected to be accomplished in the next 12–24 months. We will be following and using industry guidelines and methodologies wherever possible.

I will be glad to answer any of your questions and/or provide you with more details and education.

Sincerely,

<Your name and title>

Page 2 – Frequently Asked Questions (FAQ)

What is quantum mechanics?

Quantum mechanics/physics is a long-proven physical science that describes actions and properties of very small particles. Everything in the universe works and depends on quantum mechanics. It's how the world works. Computers and software are being created that function using quantum particles and properties. Within a few years, if not already, we will have quantum computers capable of doing things nonquantum computers cannot, including breaking many forms of traditional cryptography and creating new, unbreakable forms of cryptography.

How long have quantum computers been around?

The first working quantum computer was created in 1998. Today there are well over a hundred quantum computers and dozens of different types of quantum devices. All known quantum computers are still relatively weak, in the laboratory and experimental stages, but are predicted to become stronger than traditional computers by the end of 2019 or soon thereafter. The world's governments and corporations are spending tens of billions of dollars a year in the pursuit to build quantum supercomputers and networks. Quantum computer vendors include the world's largest companies, such as Google, IBM, Intel, Microsoft, and Alibaba.

How is quantum computing able to threaten traditional cryptography?

Particular types of quantum computers, armed with a mathematical algorithm known as Shor's algorithm, can quickly factor math equations that involve large prime numbers. Equations involving large prime numbers are what gives most traditional public key cryptography its protective capabilities. Traditional binary-based computers cannot easily factor large prime number equations. Quantum computers with enough "qubits" can factor large prime number equations in a very short amount of time, measured in minutes to hours.

When will quantum computers break traditional public key cryptography?

No one knows for sure, although as soon as quantum computers get four thousand or so "stable" qubits, it is believed that traditional public keys 2,048 bits long or shorter will be quickly crackable. Most of the world's existing public cryptography relies on such keys. Quantum computers are capable of removing half the protective power of the other types of cryptography. General estimates of time until quantum computers are capable of breaking traditional public crypto range from a few years to less than 10 years. Either way, most experts say now is the time to start preparing. If the break happens sooner than people are expecting, then we are better prepared to respond appropriately.

What are we doing?

We are forming a new project team, called the Quantum Resistant Data Protection project group, to look at all the places where our critical data protection could be impacted and the risk may need to be mitigated. Near-term mitigations are likely to include increasing existing cryptographic key sizes, isolating critical data, and moving to quantum-resistant cryptography. Long-term mitigations, many years out, include migrating to quantum-based ciphers and devices.

How can you help?

We need one or more senior management stakeholders to approve this project and give it their backing. That stakeholder needs to attend the first project meeting, perhaps attend further meetings, and answer questions from other senior managers.

Feel free to use and modify this document in any way you like. The online version of this document (`www.wiley.com/go/cryptographyapocalypse`) may be updated and improved in the future.

Step 2: Create a Plan

Creating a project plan involves many subcomponents, including creating a project team, a project plan, and a timeline.

Create a Project Team

Mitigating the threat of quantum computers will likely take many people years to accomplish. If there ever was an IT project that required a project team, this is it. If you do not have solid project management skills, get a great project manager assigned to your team or gain the necessary skills yourself. Other team members should include the following:

- Senior management sponsor
- Project leader, knowledgeable with quantum computing and other related topics (this should probably be you)
- IT security manager
- Other IT employees as required
- Cryptography subject-matter expert
- End-user representative
- Communications specialist
- Accounting/budgeting/purchaser
- Inventory manager/specialist

In small companies, many of these roles may be represented by a single person. In a really small company, all of these roles be represented by one person.

Although not needed early on, working with vendors of the impacted systems during later stages will be crucial. They need to be clued into your concerns and be able to communicate back to you what their company is doing to address your quantum break concerns. Ideally, you want them to assist with the mitigation solutions. You might need to get the vendor involved during or right after the data collection phase of the project.

A communications specialist is essential. If all goes according to plan, and all the mitigations are put into place before the quantum crypto break happens, then the communications specialist can help communicate the project to the organization as it is smoothly accomplished. If, however, the quantum crypto break happens before all issues are mitigated, an emergency accelerated plan and timeline will have to be enacted. Critical impacted assets may need to be taken offline. There could be business interruption issues. You will need an incident response plan. Let management and the communications specialist know that the accelerated timeline scenario has a lower likelihood of happening but that you want to be prepared in case it is required.

It also might be wise to initiate discussions with others in your industry, trade organizations, and even competitors. Every organization is going to be tackling a quantum resistance project of some type, although with varying timelines and objectives. Still, everyone will be having similar overall goals (i.e., migrating to quantum-resistant cryptography) and can share what actions did and didn't work for them. Make phone calls, have meetings, and bring up the topic at industry meetings and conferences. This is a major project of the biggest proportions. We are all in this together. We need each other. This is not an issue of creating a competitive advantage, but one of survival.

Create a Project Plan

Every project leader needs to create a detailed project plan, likely using some project software like Microsoft Project (`https://products.office.com/en-us/project/project-and-portfolio-management-software`) or any one of the other competing good products (for suggestions, see `www.pcmag.com/roundup/260751/the-best-project-management-software`). It's critical to figure out and document the key tasks and critical paths. The more detail and estimated timelines, the better. Overall, any project management plan should encompass the four major post-quantum mitigation project phases listed earlier and the six project steps outlined in this section.

Create a Timeline

The ultimate objective is to move all quantum-susceptible cryptography and systems protecting sensitive critical data to quantum-resistant crypto before the quantum break is a realistic threat to your organization. Not all organizations and industries will immediately be targeted when the quantum crypto break happens. Early on, most targeted attacks will probably be conducted by nation-states against military and government targets, followed by very large organizations. Any organizations in the supply chain of those entities would also be first-order targets. But once the break is made, you cannot predict when quantum computer resources will be used against your organization.

To prepare, you need to take your best guess at how long it will be until the quantum crypto break happens (use Chapter 4, "When Will the Quantum Crypto Break Happen?," to help with your guess) and prepare a timeline for how long it will take your organization to protect all critical sensitive data with quantum-resistant cryptography.

Let Mosca Inequality Be Your Guide Let the Mosca Inequality be your initial guide for timeline planning purposes. First covered in Chapter 4, it states that we need to start worrying about the impact of quantum computers when the amount of time that we wish our data to be secure for added to the time it will take for our computer systems to transition from classical to post-quantum is greater than the time it will take for quantum computers to start breaking existing quantum-susceptible encryption protocols (see Figure 9.4). When this occurs, you will not have enough time to adequately protect your data before quantum computers break the current quantum-susceptible data protection. For example, if you need your critical data to be secure for 10 future years and it will take you 5 years to transition, then you need to start moving to post-quantum systems before 15 years out from the post-quantum world.

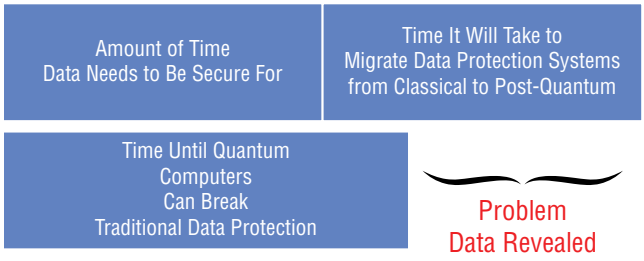

Figure 9.4: Mosca inequality

The hardest part of the equation is that no one knows exactly when the quantum crypto break will happen. Most knowledgeable experts think it is less than 10 years away, and a small percentage, like this author, think it might be as early as 2–3 years away. A relatively safe bet would be to assume it's 5 years away. Most people want to protect their most critical sensitive data from eavesdropping for 5 to 10 years, so take 7 years as a safe middle value.

The Mosca Inequality theorem says you should start working on your post-quantum migration 12 years ahead of the quantum crypto break if you want to keep your data securely protected for the entire 7 years, with 5 years of migration included. Essentially, if you haven't started your preparation with enough lead time, there is a good chance that you are already behind on getting started on your quantum resistance project.

Key Off Post-Quantum Cryptography Standard Selections For most organizations, moving to quantum-resistant cryptography will ultimately need to key off their respective nation-state's official selections of post-quantum cryptography. Few organizations will benefit by picking a

nonstandard cryptographic standard to implement across their production environment ahead of an official national standard. An organization can pick a nonstandard cipher or scheme to work with in a production environment, but doing so usually requires a large amount of independent justification. Nonstandard ciphers and schemes are usually less tested and trusted, even if they seem or are actually stronger and more secure. For picking cryptography to standardize on, there is safety in numbers and in waiting for a standard to be picked.

NOTE For example, for a few years early on, elliptic curve isogeny cryptography was thought to be quantum-resistant. But then someone discovered a way to quickly factor the involved isogenic curves. A "fix," one involving using only supersingular elliptic curves, defeated that particular attack, and now elliptic curve isogeny cryptography is known as supersingular elliptic curve isogeny cryptography. But someone could come up with a new attack that breaks even supersingular elliptic curves isogeny math. Most quantum-resistant cryptography is very new. Ciphers and schemes are still being attacked and tested. Pick a solution too soon and you increase the risk of picking a faulty cryptographic solution and having to do rework.

In the United States and much of the world, this means needing to wait for NIST to announce their post-quantum cryptographic standards, which they currently say should be between 2022 and 2024. NIST has also stated that they may make selections earlier if new information forces an accelerated timeline. This means that most organizations will have to wait until 2022 to 2024 before beginning large-scale migrations to quantum-resistant cryptography. And a complete move will likely take one to two years (at best) after the post-quantum cryptographic standards are selected as all impacted vendors begin to migrate their hardware and software offerings. There will be a natural delay between the time the cryptographic standards are selected and when they become widely available.

You can—and should—begin to experiment, test, and develop around any likely post-quantum cryptographic standards. That can only benefit your organization for when the appropriate time for full production deployment occurs. But I caution any normal organization against going to full production deployment with any nonstandard post-quantum cryptography ahead of the official selection. In the same vein, doing nothing until the official selections are made isn't optimal either. Each company needs to start investigating, planning, and protecting critical data now, and some organizations should also begin test developing in order to appropriately prepare.

Create an Emergency Accelerated Timeline Scenario It is essential that you create a backup, emergency accelerated timeline for a scenario where the quantum break suddenly happens before you've fully moved your organization's assets to quantum-resistant cryptography. Imagine, tomorrow you wake up and the NSA announces that not only has a foreign government obtained the ability to break traditional public key crypto, but the NSA has also been detecting such attacks against companies. Instead of years to get moved to quantum-resistant crypto, you need to move now. How does that change your plan? What do you drop? What counts as a critical asset to move

now and what doesn't? Can you get more resources to accomplish the project phases faster? Do you have to alert customers and the board of directors to the new imminent threat? Where is the funding going to come from? Create two timelines: one for what you think is the hoped-for, expected, normal timeline of events and one for an emergency accelerated timeline. Plan for both outcomes.

Create Project Phase Timeline Estimates How long will it take you to complete the entire quantum-resistance migration project? Estimate how long it will take your organization to move all involved data protection systems to post-quantum implementations once you start your quantum-resistant project so that senior management and project members can see likely expectations. The answer is unique for every organization and depends on what you have to move, how, and when you are able to move them. Every organization will be different, but project leaders should start with some basic guesses for each phase of the project. For example, see Table 9.1.

Table 9.1: Example quantum-resistant project tasks and timelines

Project steps	Estimated time to complete
Educate.	1 month
Form a project team and plan.	1 month
Create a timeline.	1 month
Take a data protection inventory.	3 months
Analyze the quantum plan and make recommendations.	6 months
Prevent future data leaks.	3 months
Strengthen existing traditional cryptography.	12 months
Move to quantum-resistant cryptography.	60 months
Move to quantum-hybrid technologies.	60 months
Go fully quantum.	Undetermined at this time

By assigning some general numbers to these tasks, you can come up with a rough estimate of time needed for the post-quantum migration project and inform all stakeholders of the tasks and timings. Some tasks, such as education, forming a project team and planning, and creating a timeline can be accomplished simultaneously. With the example numbers listed in Table 9.1, the actual project work would take six to seven years, with most of that belonging to the "Move to quantum-resistant cryptography" phase. Of course, the move to quantum-resistant cryptography and ultimately quantum-based cryptography depends on factors beyond your control.

By appropriately planning for all phases, including the post-quantum cryptographic migration, even if a faster, accelerated move to quantum-resistant cryptography is needed than you originally planned, your organization can be better prepared for the move. Proper planning saves time. Even quantum mechanics doesn't change that.

Step 3: Collect Data

Critical to any quantum-resistant cryptographic migration project is to do a complete data protection inventory. It should include five components:

- Locating all sensitive data and devices
- Identifying data and device stakeholders
- Performing confidentiality ranking
- Determining how long the data or devices need to be protected from unauthorized eavesdropping
- Identifying current data protection systems involved with protecting devices that use cryptography, particularly identifying involved cryptography and key sizes

The inventory should include a thorough accounting of all sensitive data and devices that you need to prevent from unauthorized disclosure. There is a saying in computer security: "If you think you know where all your data is, you're mistaken or lying to yourself." With that said, you need to do the best accounting of where all sensitive data and devices are located, along with the stakeholders who own, safeguard, and rely on the information and devices. Stakeholders are needed to ascertain all the other needed information.

All devices holding or involved with holding sensitive information must be inventoried and investigated. This includes computers, laptops, pad devices, smartphones, network equipment, authentication devices, physical security devices, and more. The inventory should include critical data and Internet of Things (IoT) devices such as security cameras, badging systems, and building access control systems. If IoT devices are recording sensitive information or in sensitive areas, they need to be on the inventory list. If a device uses cryptography and protects something nonpublic, it needs to be on the list.

All data and devices should be ranked according to data sensitivity and the need to keep it protected. Some organizations use the traditional military rankings: top secret, secret, confidential, and public. Others use the more corporate equivalents: high business value, medium business value, low business value. Some organizations simply go for confidential and public. Whatever data classification system you use, all data and devices should be ranked by criticality and value to the organization.

Next, stakeholders need to determine how long identified data and devices needs to be securely retained, stated in years or critical bands (e.g., long-term, mid-term, short-term, or 10 years or more, 5–10 years, less than 5 years).

Then, identify the data protection systems involved in protecting that data and those devices if they use cryptography. Not all data protection systems directly use cryptography, but most do at least indirectly. For example, many access control systems use only operating system permissions, which don't directly involve cryptography, but the overall success of the access control permission depends on the security of the operating system, and all operating systems use cryptography (usually

multiple ciphers, schemes, and systems). You want to identify all cryptography involved in securely storing and transmitting identified data and devices. Critical data systems will also include all your vital IT infrastructure, such as authentication systems and infrastructure services. For each data protection system, identify the cryptography used (symmetric, asymmetric, hashes, digital signatures, etc.) and key sizes.

For many systems, you may be unable to determine what the ciphers, schemes, and key sizes are. The vendors or developers may be gone, with little to no available documentation regarding the implemented cryptography. The project team will need to review each found "unknown" and determine whether the risk is so minor that it can be ignored for upgrading purposes or it needs to be considered a top priority (just to decrease potential risk). Plan for your unknowns.

Step 4: Analyze

Computer security is mostly about risk assessment. In this most important phase of the project, you will identify the most at-risk data and systems and determine which steps must be taken to mitigate the risk of the coming quantum crypto break. Analysis and recommendations will include the following tasks:

- Identifying quantum-susceptible data protection systems protecting critical sensitive data and devices
- Ranking the risk factor for threatened systems, including which systems need priority remediation
- Determining remediations
- Determining related resources, costs, and timelines

Once you have the results of the data protection inventory, identify the quantum-susceptible cryptography used and, in particular, that protecting critical sensitive data and devices. Compare what you find to the list of cryptography and key sizes that are known to be quantum-susceptible, as shown in Chapter 5, Table 5.1. Chapter 5 also covered the quantum-susceptible asymmetric ciphers (e.g., RSA, DH, ECC, ElGamal, etc.).

Any data protection system protecting critical data or devices and using susceptible cryptography should be highlighted, with special attention paid to the highest critical systems. This will give you a bird's-eye view of how big the challenge will be to move to a post-quantum world. It is likely to be a big wake-up call to the size and extent of the problem, although perhaps you will be surprised to learn that a lot of your systems are using less susceptible cryptographic forms such as AES-256 or AES-512. A thorough risk assessment may lead you to conclude that the problem isn't as bad as you feared. Unfortunately, this is not likely to be the case for most organizations, particularly where asymmetric ciphers are used.

Remediation should include all the possible solutions required to fix a quantum-susceptible system, starting with education for all system stakeholders. Remediation should also identify which solutions

are simple key size upgrades (e.g., from AES-128 to AES-256) and which solutions are entire cipher replacements. Which can be upgraded and which must be replaced? How hard is the effort expected to be for each system? What is the expected cost and timeline for each remediation solution? You'll need to talk to both stakeholders and vendors.

Many systems may end up taking hybrid approaches. For example, perhaps the root certification authority (CA) digital keys, which can be expected to have long lives, get renewed using a quantum-resistant cryptography, but the individual end-user keys with much shorter lives get updated to use traditional cryptography with bigger key sizes.

It's important to involve vendors as soon as possible so that they understand your concern and can let you know how their company is planning to help. In many cases, the vendors may be completely unaware of the issue, or they are aware of the pending quantum break but have not practically addressed it because they think it's 10–20 years off. Discussing your concerns with them may be the start of serious discussion within their company. You may even learn that they don't plan on moving to a post-quantum implementation or that it requires upgrading to the latest version of their software to fix. Many vendors used the SHA2 migration move to force customers to upgrade to the latest software.

Many companies may have internally developed applications and will need to find out who is in charge of them and how they can be upgraded. It is common to find internally developed applications for which no information can be found and for which no one can be hired to analyze and update. Your project team should have already decided how to handle such applications, or perhaps they are reviewed on a case-by-case basis.

Even systems that are not currently considered quantum-susceptible must be updated. For example, you may have highly critical data that is currently protected by AES-256, which is not considered quantum-susceptible. But if you plan to keep the data for 10 years or longer, you may want to move it to AES-512. This move from a slightly quantum-susceptible cryptography to a far less quantum-susceptible cryptography may not be a top priority but should still be done.

Then, after a thorough examination of all the involved systems, criticalities, timelines, and costs, stakeholders should decide what remediation to deploy and when. Document the decisions and present them to senior management for approval and budgeting. If you've done the analysis and review correctly and thoroughly, the possible solutions make the ultimate decision easy for most of the decisions.

Protect Some or All Data

Up front you may want to decide whether it makes sense to protect just the highest critical data or to protect everything. Sometimes it saves money and time to protect everything. For example, this is the answer for many companies doing business in the European Union (EU) or with EU customers when trying to comply with the EU's General Data Protection Regulation (GDPR) requirements. Many multinational companies have decided it was cheaper to apply GDPR requirements to all data and customers located anywhere, even outside the EU, than to segregate and apply those requirements

to EU-specific data alone. They also worried about the legal and financial costs if some data that should have had GDPR applied to it slipped into their lesser protected data repositories. Rather than face the risk, those companies just applied the stronger controls to all data.

Conversely, some companies have determined it was much cheaper to put sensitive data into an isolated data realm, which was then protected by the more expensive controls. Many U.S. firms made this decision for PCI- and HIPAA-regulated data. This strategy can make sense when the amount of data is a small percentage of what the organization deals with and if the covered data can be trusted to remain in the isolated data domain. Each organization must decide if they will protect all or just some of the data. The middle-of-the-road decision is to protect only the most critical data but to do it across the enterprise by focusing on the systems that protect it.

Organizations must decide which data needs to be protected against quantum attacks and which cryptography need to be updated and when.

Step 5: Take Action/Remediate

Now is the time to take action. Update ciphers, schemes, and key sizes. Migrate to quantum-resistant cryptography and perform the other post-quantum actions. Each step should be thoroughly tested in limited pilot projects before moving to full-scale production deployments. Every action plan should include an emergency "backout" plan in case the cryptographic migration step breaks something. The backout steps should be documented and thoroughly tested before the protection steps are taken.

NOTE I used to be proud to say that in my 30 years of deploying cryptographic projects and updates, I never caused a single major unintended operational interruption event—that is, until a few years ago. And it was a doozy. The customer team and I were updating digital certificates on mission-critical networked medical devices as part of a much larger global update project. A backout procedure was created, which allowed any of the newly issued certificates to be remotely replaced with the original, older working certificate if needed by running a single, well-tested command. If anything went wrong with the production update, we could run one command and sit back as we quickly recovered all impacted devices.

A few tests of the certificate update process went so well that I recommended we deploy all remaining tens of thousands of certificates immediately, globally. I had performed the same global update dozens of times before and believed there was no way the certificate update could cause a problem. No way. Oops. Unbeknownst to me, the customer had made a custom change to half their devices that caused these newly updated devices (critical medical equipment) to disconnect from the network when the new certificates were installed. Every device malfunctioned, and because they also disconnected from the network, our "failsafe" backout procedures using a single command could not be run. The hospital's global network operations were down for several days, and it required teams of non-IT-trained healthcare workers to perform dozens of hands-on steps to get the devices

back online and functioning. It was the biggest (and gratefully, only) customer disaster of my professional career. Lesson learned. Test and test again. Don't rush to production deployments without enough testing. And test your backout procedures and make sure they do not rely on network connections to work.

Update Existing Policies and Standards

It goes without saying that right now you should make sure all organizational policies and standards require strongly quantum-resistant cryptography. You want to stop the bleeding. You don't want new systems coming into your organization that contain weakly quantum-resistant cryptography and will just add to your problems down the road. Update your policies and standards to require automatic rejection of weak cryptographic systems. While you're at it, update policies to state that all systems must use widely accepted cryptographic standards if you don't already have that language. And if you want to reach for the stars, require that all newly bought cryptography-using systems be crypto-agile.

Prevent Current-Day Future Data Leaks

There is the risk that outside entities with current or near future access to quantum cracking technologies could be eavesdropping on your "currently protected" data today, saving it, and waiting to crack into that data when they have the appropriate capability. We should assume this is happening by all nation-states against other nation-states today. They would be stupid and neglectful not to do this. Large, morally corrupt corporations that participate in corporate espionage may be doing this as well against competitors.

For example, perhaps your Wi-Fi network routers use AES-128. Anything protected by AES-128 is soon to be quantum-susceptible. Your enemies could be sniffing your Wi-Fi data stream and saving it for future cracking. If this is a possibility against your organization, consider all the methods you need to use to prevent future eavesdropping against your currently secured data. These methods include the following:

- Removing critical data from any online storage or network transmission (so it can't be sniffed or stolen in the first place)
- Using quantum-resistant crypto today
- Isolating physical data domain (where it cannot be sniffed or stolen)
- Using network isolation equipment that is not susceptible to quantum cracking (not the same as using quantum-resistant crypto)

Regarding the last item, there are vendors who make military-grade, highly secure network cards and equipment, which cannot be easily eavesdropped on. They do not use traditional cryptography but instead use shielding and signaling methods that cannot be eavesdropped on to even allow an attacker to read the encrypted or protected information in the first place.

Step 6: Review and Improve

Any project plan should end with a review-and-improve stage. There will always be lessons learned, both good and bad, from any complex project along the way. Every project plan should have multiple points where everyone assesses how the plan and remediation actions are going and makes recommendations to improve where needed.

Summary

This chapter discussed how you and your organization can start preparing today for the coming quantum break before it happens. This strategy includes getting senior management involved, forming a long-term project team, providing education, and moving your organization's systems away from traditional classical cryptographic protection systems. To do this in a methodical way, you need to do a complete and thorough data protection inventory to identify the systems to be migrated. Then, decide on the correct mitigations, which include increasing key and hash sizes of existing, currently used traditional cryptography, followed by moving to quantum-resistant cryptography, and finally moving to fully quantum cryptography and devices. Planning and being thoughtful about the process will save any organization time and money and, more importantly, efficiently reduce computer security risk.

I want to thank readers for allowing me to take them on the journey from learning what quantum mechanics is and how it will allow many forms of traditional cryptography to be cracked. This book covered quantum-resistant and quantum-based cryptography and devices as well as any resource in the market today and finished with a solid plan summary of how your organization can start preparing for the post-quantum world today. If you have any questions, suggestions, or corrections, feel free to email me at `roger@banneretcs.com`. I will attempt to answer questions within 24 hours.

The appendix closes the book by listing many quantum and quantum computing online resources, and it can be used as a research reference for education and news.

Appendix: Additional Quantum Resources

This appendix lists dozens of resources you can use to improve and extend your understanding of quantum mechanics, quantum computers, and cryptography.

Fully understanding quantum mechanics is something that all the experts in the field of quantum mechanics struggle with. Not only is it a complex subject that challenges our traditional understanding of natural science, but it is still in the early days of understanding with lots of missing pieces. So, you can be forgiven if you don't fully understand quantum mechanics, computers, or cryptography. Every quantum-related article and resource you consume will improve your understanding. With that in mind, this appendix gives you a list of resources that includes citations from the other chapters as well as new resources. It is broken out by various categories, including books, videos, online courses, websites, blogs, government-sponsored programs, vendor websites, and so forth. For many of the resources, I made comments at the end of the citation in parentheses, where I thought I could add any value in helping you to understand whether the resource is right for you.

Books

Aaronson, Scott (2013). *Quantum Computing Since Democritus*. Cambridge: Cambridge University Press. (Great read by a very active quantum mechanics and quantum computer researcher, with a heavy focus on computer logic. Explains entanglement and quantum teleportation particularly well.)

Bell, Philip (2018). *Beyond Weird: Why Everything You Knew About Quantum Physics Is Different*. Chicago: University of Chicago Press. (Great read; heavy on debating different quantum interpretations; explains some quantum mechanic concepts in extraordinarily different, great ways (Bell's explanation of decoherence and entanglement is the best I've ever read). Should not be your first or only book on quantum physics but should be required reading for anyone with an interest in understanding quantum mechanics.)

Bernhardt, Chris (2019). *Quantum Computing for Everyone*. Cambridge, MA: MIT Press.

Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto, First Edition. Roger A. Grimes.

Carroll, Sean (2019). Something Deeply Hidden. United States: Dutton. (Passionate, logical defense of the Many Worlds interpretation.)

Johnston, Eric R., Nic Harrigan, and Mercedes Gimeno-Segovia (2019). *Programming Quantum Computers: Essential Algorithms and Code Elements*. Sebastopol, CA: O'Reilly. (Newest book on quantum I'm aware of besides this book.)

Kumar, Manjit (2009). *Quantum*. New Delhi: Hachette India. (Recommended by Philip Bell as a great introduction to quantum mechanics.)

Kumar, Manjit (2011). *Quantum: Einstein, Bohr, and the Great Debate about the Nature of Reality*. New Delhi: Hachette India.

Orzel, Chad (2009). *How to Teach [Quantum] Physics to Your Dog*. New York: Scribner. (Great read; perfect for first-timers to learn quantum physics.)

Orzel, Chad (2018). *Breakfast with Einstein: The Exotic Physics of Everyday Objects*. Dallas, TX: BenBella Books, Inc. (Great read; perfect for the first-timer to learn quantum physics.)

Rhodes, Richard (2012). *Hedy's Folly: The Life and Breakthrough Inventions of Hedy Lamarr, the Most Beautiful Woman in the World*. New York: Doubleday. (Great read; learn how Hedy Lamarr co-created encryption that is the base of what protects most wireless communications even today.)

Videos

Anyons and quantum topological computers:

`www.youtube.com/watch?v=igPXzKjqrNg`

`www.youtube.com/watch?v=RW44rIrAZHY`

`www.youtube.com/watch?v=qj-w6ISQL5Y`

`www.youtube.com/watch?v=Xyfsr-coriQ`

BB84: `www.youtube.com/watch?v=UVzRbU6y7Ks`

D-Wave/annealing process: `www.youtube.com/watch?v=UV_RlCAc5Zs`

`www.youtube.com/watch?v=kq9VqROZGNc`

`www.youtube.com/watch?v=Yy93LMGQbpo`

Double Slit Wave Experiment animation: `www.youtube.com/watch?v=fwXQjRBLwsQ`

Ion-trap quantum computers:

`www.youtube.com/watch?v=9aOLwjUZLm0`

`www.youtube.com/watch?v=WOQ_jWe62EA`

Quanta Magazine YouTube channel: `www.youtube.com/c/QuantamagazineOrgNews`

Quantum Physics for 7 Year Olds: `www.youtube.com/watch?v=ARWBdfWpDyc`

Quantum teleportation:

www.youtube.com/watch?v=Czi5elPLfvA

www.youtube.com/watch?v=hTe2PYwnEpc

Quantum Theory—Full Documentary HD: www.youtube.com/watch?v=CBrsWPCp_rs

Neil deGrasse Tyson explains quantum entanglement: www.youtube.com/watch?v=q8CQAOwi2RI

Online Courses

Scott Aaronson (2006). "Quantum Computing Since Democritus": www.scottaaronson.com/democritus/

Quantum Computing Report's list of online educational resources: https://quantumcomputingreport.com/resources/education/

Kirill Shilov, "16 Best Resources to Learn Quantum Computing in 2019": https://hackernoon.com/16-best-resources-to-learn-quantum-computing-in-2019-e5d8b797aeb6

Leonard Susskind (2011), professor of physics at Stanford University. Quantum Mechanics: "The Theoretical Minimum": http://theoreticalminimum.com/courses/quantum-mechanics/2012/winter

Websites

Daniel J. Bernstein's personal website: https://cr.yp.to

Caltech's Institute for Quantum Information and Matter: https://quantumfrontiers.com/

GeekForge: https://geekforge.io/

High Energy Physics Foundation Quantum Computing working group: https://hepsoftwarefoundation.org/workinggroups/quantumcomputing.html

IFLScience!: www.iflscience.com/physics/

Inside Science quantum-related articles: www.insidescience.org/search/node/quantum

ISARA: www.isara.com/standards/ (list of ongoing postquantum cryptographer efforts)

Quantiki: www.quantiki.org

Quantum Algorithm Zoo: https://quantumalgorithmzoo.org/

Quantum Computing Reporting: https://quantumcomputingreport.com

Quantum Computing Stackexchange: https://quantumcomputing.stackexchange.com/ (great for technical questions)

Quantum for Quants: www.quantumforquants.org/ (financial-related site)

Phys.org—Quantum Physics news: https://phys.org/physics-news/quantum-physics/

Physics Forums, Quantum Physics Forum: www.physicsforums.com/forums/quantum-physics.62/

Post-Quantum Cryptography wiki: https://pqc-wiki.fau.edu/w/Special:DatabaseHome

PQCrypto: http://pqcrypto.eu/

Sam Mugel's website for quantum technology in simple words: www.qwise.org

Science Daily: www.sciencedaily.com/news/matter_energy/quantum_physics/

ScienceDirect's quantum-related articles: www.sciencedirect.com/search/advanced?qs=quantum

The Quantum Pontiff: https://dabacon.org/pontiff/

UK's National Quantum Technologies Hub for Networked Quantum Information Technology: www.nqit.ox.ac.uk

Blogs

Scott Aaronson's blog: www.scottaaronson.com

Quantum Physics blog: www.techbubble.info/blog/quantum-physics

Top 25 Quantum Computing Blogs: https://blog.feedspot.com/quantum_computing_blogs/

Podcast

Stupid Qubit: https://stupidqubit.com/ (funny, edgy podcast on quantum)

Quantum Magazines/Newsletters

Quanta Magazine, Facebook: www.facebook.com/QuantaNews

Quanta Magazine newsletters: https://us1.campaign-archive.com/home/?u=0d6ddf7dc1a0b7297c8e06618&id=f0cb61321c

Nature Magazine, quantum-related articles: www.nature.com/search?q=quantum

Wired Magazine's quantum-related articles: www.wired.com/search/?q=quantum

Quantum Mailing Lists

Quantiki list of mailing groups: www.quantiki.org/wiki/mailing-lists

Eclipse Foundation's Quantum Computing Working Group: https://accounts.eclipse.org/mailing-list/quantum-computing-wg

Quantum Foundations mailing list: https://golem.ph.utexas.edu/category/2010/12/quantum_foundations_mailing_li.html

Quantum Internet: www.irtf.org/mailman/listinfo/qirg

Miscellaneous Quantum Articles

Philip Ball, Quanta Magazine articles: www.quantamagazine.org/authors/philip-ball/

Mark G. Jackson's articles for popular audiences: http://physicsjackson.com/articles/

Quantum Vendors

Accenture: www.accenture.com/us-en/insight-quantum-computing

Alibaba: https://us.alibabacloud.com/

Atos: https://atos.net/en/insights-and-innovation/quantum-computing/atos-quantum

Baidu: http://research.baidu.com/Research_Areas/index-view?id=75

Cambridge Quantum Computing: https://cambridgequantum.com/

ComScire: https://comscire.com

D-Wave: www.dwavesys.com

Google: https://ai.google/research/teams/applied-science/quantum-ai/

Honeywell: www.honeywell.com/en-us/company/quantum

Huawei: www.huaweicloud.com/en-us

IBM: www.research.ibm.com/ibm-q/

ID Quantique: www.idquantique.com

Intel: https://newsroom.intel.com/press-kits/quantum-computing/#quantum-computing-news

IonQ: https://ionq.co/

MagiQ Technologies: www.magiqtech.com

Microsoft: www.microsoft.com/en-us/research/research-area/quantum/

Quantum Computing, Inc.: https://quantumcomputinginc.com/

Quantum Numbers Corp.: www.quantumnumberscorp.com

Quintessence Labs: www.quintessencelabs.com

Raytheon: www.raytheon.com/capabilities/products/quantum

Rigetti: www.rigetti.com/qcs

Toshiba: www.toshiba.eu/eu/Cambridge-Research-Laboratory/Quantum-Information/

Twitter

Caltech: https://twitter.com/IQIM_Caltech

European Union's Quantum Internet Alliance: https://twitter.com/eu_qia

Qiskit: https://twitter.com/qiskit

Quanta Magazine: https://twitter.com/QuantaMagazine

Quantiki: https://twitter.com/quantiki

UK's National Quantum Technologies Hub for Networked Quantum Information Technology: https://twitter.com/NQIT_QTHub

Software-Related

List of quantum algorithms: http://quantumalgorithmzoo.org/

List of quantum computing simulators: www.quantiki.org/wiki/list-qc-simulators

List of quantum open-source projects: https://arxiv.org/pdf/1812.09167.pdf

List of quantum software: https://github.com/qosf/os_quantum_software

IBM Quantum Q Experience: https://quantumexperience.ng.bluemix.net/qx/editor

Microsoft Quantum Software Development Kit: https://marketplace.visualstudio.com/items?itemName=quantum.DevKit

Open Quantum Safe Project: https://openquantumsafe.org/

Open-source and commercial quantum software projects and online quantum portals: https://github.com/qosf/os_quantum_software

Python quantum open-source library: https://github.com/rigetti/pyquil

Quantum Open Source Foundation: https://qosf.org/

Quirk, drag-and-drop quantum simulator: https://algassert.com/quirk

Miscellaneous Quantum Consortiums

Alliance for Quantum Technologies: http://inqnet.caltech.edu/index.html

Quantum Worldwide Association: http://quantumwa.org/

Government-Sponsored Programs and Nonprofits

Australia, Australian Research Council's Centre of Excellence for Engineered Quantum Systems: https://equs.org/

Australia, Center for Quantum Computation & Communication Technology: www.cqc2t.org

Barcelona, Catalonia, Spain, Institute of Photonic Sciences: http://quantumtech.icfo.eu/

Barcelonaqbit: www.barcelonaqbit.com/

Beijing Academy of Quantum Information Science: www.baqis.ac.cn/en/

Berkeley Quantum: https://berkeleyquantum.org/

Brookhaven National Laboratories: www.bnl.gov/compsci/quantum/

China, CAS Key Laboratory of Quantum Information: http://lqcc.ustc.edu.cn/

Entanglement Institute, Newport, Rhode Island: www.entanglement.institute/

Fermilab Quantum Information Science Program: https://qis.fnal.gov/

France, Grenoble Quantum Silicon: www.quantumsilicon-grenoble.eu/

German Research Foundation's Matter and Light for Quantum Computing: https://ml4q.de/

IARPA's Coherent Superconducting Qubits: www.iarpa.gov/index.php/research-programs/csq

IARPA's Logical Qubits: www.iarpa.gov/index.php/research-programs/logiq

IARPA's Multi-Qubit Coherent Operations: www.iarpa.gov/index.php/research-programs/mqco

IARPA's Quantum Enhancement Optimization: www.iarpa.gov/index.php/research-programs/qeo

India, Light and Matter Physics: www.rri.res.in/light-matter-physics.html

Korea, Center for Quantum Information: http://quantum.kist.re.kr/

Leti, France: www.leti-cea.com/cea-tech/leti/english/Pages/Applied-Research/Strategic-Axes/Quantum-leti-initiative.aspx

Los Alamos Quantum Institute: https://quantum.lanl.gov/about.shtml

NASA Quantum Artificial Intelligence Laboratory: https://ti.arc.nasa.gov/tech/dash/groups/quail/

National Science Foundation's Enabling Practical-Scale Quantum Computing: www.epiqc.cs.uchicago.edu/

National Science Foundation's Quantum Information Science: www.nsf.gov/funding/pgm_summ.jsp?pims_id=505207

Netherlands, QuSoft Research Center for Quantum Software: www.qusoft.org

Netherlands, QuTech Academy: http://qutech.nl/

NIST Joint Center for Quantum Information and Computer Science: http://quics.umd.edu/

NIST Joint Quantum Institute: https://jqi.umd.edu/

NIST Post-Quantum Cryptography contest: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography

NIST Quantum Information Science: www.nist.gov/topics/quantum-information-science

Oak Ridge National Laboratory Quantum Computing Institute: https://quantum.ornl.gov/

Oak Ridge National Laboratory Quantum Information Science Group: https://web.ornl.gov/sci/qis/index.shtml

Paris Centre for Quantum Computing: www.pcqc.fr

Perimeter Institute for Theoretical Physics Quantum Information Research Group: http://perimeterinstitute.ca/research/research-areas/quantum-information

Russian Quantum Center: https://rqc.ru/

Singapore, Centre for Quantum Technologies: www.quantumlah.org

Singapore, Quantum Technologies for Engineering Programme: www.a-star.edu.sg/imre/Research/Programmes-Centres/Quantum-Technologies-for-Engineering-Programme

Spanish National Research Council: https://qst.csic.es/

Swiss National Science Foundation's Quantum Science and Technology: https://nccr-qsit.ethz.ch/

United Kingdom's National Quantum Technology Programme: www.nqit.ox.ac.uk/

Universities Space Research Association: www.usra.edu/quantum-computing

U.S. National Science & Technology Council, National Strategic Overview for Quantum Information Science: www.whitehouse.gov/wp-content/uploads/2018/09/National-Strategic-Overview-for-Quantum-Information-Science.pdf

NOTE Much of the information in this last section was provided by https://quantumcomputingreport.com/players/governmentnon-profit/.

Index

Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto, First Edition. Roger A. Grimes.

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z