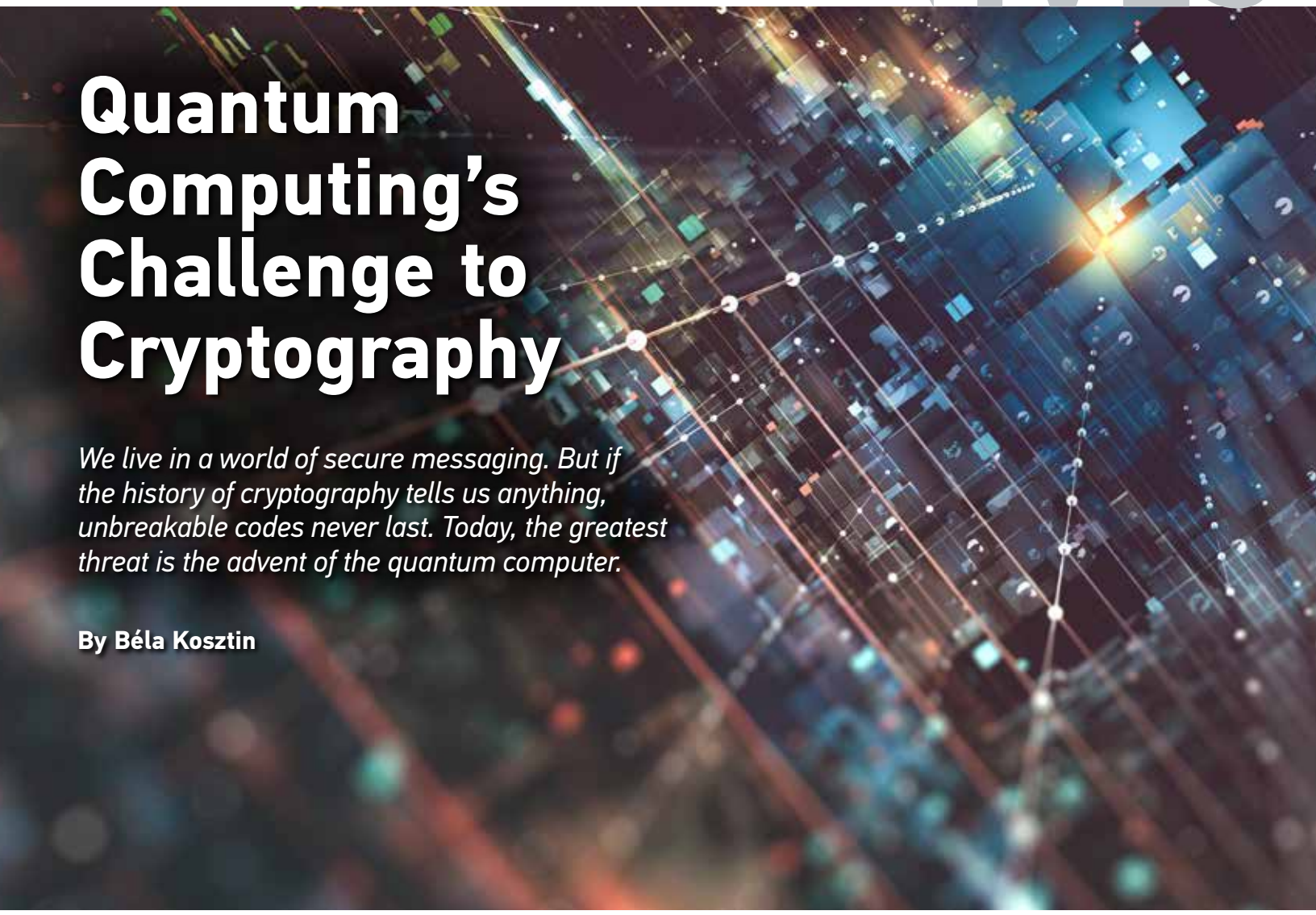
JULY 21, 2021

WORLDQUANT PERSPECTIVES



Quantum Computing's Challenge to Cryptography

We live in a world of secure messaging. But if the history of cryptography tells us anything, unbreakable codes never last. Today, the greatest threat is the advent of the quantum computer.

By Béla Kosztin

WorldQuant, LLC
1700 East Putnam Ave.
Third Floor
Old Greenwich, CT 06870
www.weareworldquant.com

AS LONG AS THERE'S BEEN WAR — OR PROXY WARS LIKE politics and finance — there's been cryptography. Soon after the first cryptography — the first attempt to communicate a secret message for whatever reason — came cryptanalysis, the attempt by outsiders to "read" that message. Code makers propagate code breakers, and vice versa. Technologies and codes give up their secrets. That dynamic mirrors, over many centuries of human conflict, the drive for dominance through ever more potent military technologies. Like armies, cryptography and its sibling cryptanalysis have engaged in a seemingly eternal and mutually reinforcing arms race that continues to this day.[1]

Warfare may be an incubator of cryptography, but today many industries depend on cybersecurity to function. The security of email, passwords, financial transactions, even electronic voting systems requires security objectives such as confidentiality and integrity. Banks send vast payment flows that must be secure against cyber criminals. Consumers use credit cards and payment apps to buy and sell on the internet. Governments not only need to communicate securely, but they are huge repositories of classified secrets. And, of course, militaries, which communicate digitally and increasingly depend on data gathering and computer power, view cryptography as both a powerful offensive tool and a potentially large national security vulnerability.

Today, cybersecurity, cryptography and cryptanalysis are colliding and threatening to disrupt the sprawling network of secure communications that has emerged from the development of asymmetric or public key cryptography. The most famous cryptographic technology, known as RSA, was developed secretly in Great Britain in 1973 and rediscovered in the U.S. in 1977 by Ronald Rivest, Adi Shamir and Leonard Adleman (the R, S and A of RSA).[2] RSA has proved to be highly secure, and for more than 40 years it has provided relative cryptographic stability, so much so that Philip Zimmermann, the inventor of Pretty Good Privacy (PGP), which provides strong, RSA-type cryptography for email, has described it as "the golden age of cryptography." (Zimmermann has less admiringly called our times "the golden age of surveillance."[3])

If the study of cryptography tells us anything, stability is not eternal, particularly in an age of such rapid technological advances. The greatest threat today emerged from a speculative lecture titled "Simulating Physics with Computers" that physicist Richard Feynman gave in 1981 at MIT.[4] Feynman imagined what today would be called a quantum computer, which uses the quantum behavior of elementary particles to calculate and compute with enormous speed. Quantum computers are still in their infancy, the challenges to their development remain immense, and there's little consensus about how long it will be before they can break RSA. However, they do pose a potential threat to the current regime by performing calculations that are beyond the range of the fastest supercomputers, thus opening the possibility for what one policymaker calls "a quantum cryptocalypse."[5]

The threat has stirred up the efforts of hundreds of the world's cryptographers to understand it and develop responses and new security standards for both classical and quantum computers. These efforts are complicated by the reality that a cryptographic method is normally effective only as long as it remains relatively secure itself. That's what remains special about RSA, whose specs have been available since the 1990s. Although the technologies are new and complex, the problem is ancient.

Quantum computing (QC) gives rise to many questions, such as: What threats does it pose? Can QC, like other new technologies, be used for both offense and defense — for cryptography and cryptanalysis? Can RSA adjust and evolve, or do we need entirely new methods? Is there, in fact, any truly unbreakable cryptographic method? In this article, we will survey the history of cryptography before turning to the latest disruption on the horizon.

THE BEGINNINGS

In the introduction to his history of cryptography, *The Code Book,* Simon Singh describes what he calls the evolution of codes: "Evolution is a wholly appropriate term, because the development of codes can be viewed as an evolutionary struggle. A code is constantly under attack from codebreakers. When the codebreakers have developed a new weapon that reveals a code's weakness, then the code is no longer useful. It either becomes extinct or evolves into a new, stronger code."[6]

Initially, codes were literally hidden.[7] In ancient Greece, a messenger's head was shaved and a message written on his skull. As a result, the messenger would have to wait for his hair to grow back before departing. This form of coding is called steganography: The message is in a readable format, but its location is hidden. In China, messages were written on thin pieces of silk, covered in wax and swallowed by messengers. In Italy, they were written in organic ink that could not be seen until heat was applied. The downside was that once these techniques became common knowledge, they were easily decrypted and lost their value.

> *"Cybersecurity, cryptography and cryptanalysis are threatening to disrupt the sprawling network of secure communications that has emerged from the development of public key cryptography.".*

In time, cryptographers moved on, replacing the readable format with a seemingly random sequence of numbers and letters. The aim was less to hide the message than to disguise the information contained in it — to encrypt it. Julius Caesar, according to Roman historian Suetonius, used an encryption method for his private correspondence that involved shifting letters of the alphabet by a known amount to produce an unreadable string of seemingly random letters.

For centuries, this type of encryption was thought to be unbreakable. However, as Europe emerged from the Middle Ages, nations increasingly used code breakers to read the diplomatic correspondence of other states. For a time, France dominated code breaking through the efforts of François Viète, a privy councillor, lawyer and mathematician who found the key to the so-called Spanish ciphers, consisting of some 500 characters. King Philip II of Spain refused to believe his ciphers could be broken and demanded the pope execute Viète for witchcraft. But because the Vatican could also decrypt Spanish messages, Viète remained safe.

These types of ciphers had a name — "monoalphabetic," which also describes their flaw. Once the language of the message is known, code breakers can use a letter-frequency database to identify the most common letters and words. In English, the most common letter is "e." This method is called frequency analysis.

In the 16th century, a French diplomat named Blaise de Vigenère moved decisively past the flaws of monoalphabetic coding. Rather than use just one alphabetic cipher, the Vigenère method employed multiple ciphers — hence its name, polyalphabetic. It used a plaintext alphabet followed by 25 cipher alphabets, each successively shifted by one letter; this is known as a Vigenère square. To unscramble a message, the recipient only needed to know which row of the Vigenère square was used to encode each letter. This was achieved by using a keyphrase. The Vigenère cipher was invulnerable to frequency analysis; a letter that appeared multiple times in a ciphertext represented multiple plaintext letters and generated tremendous complexity for cryptanalysts. The cipher also offered a huge choice of keys. The sender and the receiver could agree on any word in the dictionary and any combination of words, even fabricated words. The cipher became common in the 17th and 18th centuries, and the development of the telegraph in the 19th century made it popular with business. The Vigenère cipher became known as *le chiffre indéchiffrable* — "the indecipherable cipher."

> *Turing believed that by studying old messages he could predict part of the contents of an encrypted message based on when it was sent and its source.*

By the 19th century, complexity and information had increased rapidly, and the first mechanical calculators had emerged. Charles Babbage, born in London in 1791, not only designed the first modern computer but, as he wrote in his autobiography, engaged in cryptography: "Deciphering is, in my opinion, one of the most fascinating of arts." As Singh notes, "Cracking a difficult cipher is like climbing a sheer cliff face: The cryptanalyst seeks any nook or cranny that could provide the slightest foothold." In a monoalphabetic cipher, the cryptanalyst uses letter frequency. In the polyalphabetic Vigenère cipher, the frequencies are more balanced, and at first sight the rock face seems perfectly smooth.

Babbage, however, found a foothold in repetitions. In English, the word "the" has a very high frequency. This provides a clue to the frequency of the repetition — that is, the length of the keyword. Once the length of the keyphrase is identified (*n*), we know that every *nth* letter is encrypted using the same row of the Vigenère square. Therefore, we can divide the ciphertext into n monoalphabetic problems, which can be cracked by frequency analysis.

Babbage's successful breaking of the Vigenère cipher was probably achieved in 1854, but he never published it. His discovery occurred soon after the outbreak of the Crimean War and may have given Britain an advantage over Russia; it's possible the British military demanded Babbage keep his work secret. Babbage's finding came to light only in the 20th century, when scholars explored his notes. (Friedrich Wilhelm Kasiski, a retired Prussian army officer, came upon the same idea in 1863.)

A WIRELESS WORLD

In 1894, Guglielmo Marconi invented the radio. Communication through the air was a powerful technology but not secure; messages could be understood by anyone tuned into the correct frequency. In World War I, Germany suffered heavily from security breaches. A German telegram intercepted by the British on January 17, 1917, was immediately sent to Room 40, the British Admiralty's cipher bureau. Within a few days, the cryptanalysts there had discerned the outline of a new German naval offensive, which would include U-boat attacks on still-neutral American shipping. Their discovery helped draw the U.S. into the war.

Germany quickly caught up, though not quickly enough to win the war. In 1918, German inventor Arthur Scherbius helped found Scherbius & Ritter, an engineering firm. One of Scherbius' projects was to replace traditional codes and ciphers with encryption that exploited 20th-century technology. He developed cryptographic machinery he dubbed Enigma. To avoid repetition, Enigma used three scramblers that deepened the random appearance of

messages. For a full alphabet, these three scramblers would provide 26x26x26, or 17,576, distinct scrambler arrangements. To decipher a message, a receiver needed an Enigma machine and a copy of the codebook that contained the initial scrambler settings for that day.

Scherbius believed Enigma was unbreakable. So did the German military, which over two decades bought more than 30,000 Enigma machines. At the start of World War II, German military communications enjoyed an unparalleled level of encryption.

As the number of Enigma machines grew, Room 40's ability to decode shrank. The Americans and French also failed, but in Poland in 1932, mathematician Marian Rejewski led a team that, using French intelligence, began to reverse-engineer Enigma. The Poles' success proved that Enigma was not a perfect cipher — and proved the value of employing mathematicians as code breakers. Room 40 had been dominated by linguists and classicists, but now there was a concerted effort to bring on mathematicians and scientists. The recruits were sent to Bletchley Park, the home of Britain's new Government Code and Cypher School.

In the autumn of 1939, Bletchley Park worked to unravel the intricacies of the Enigma cipher and master the Polish techniques. Bletchley had more resources than Room 40 and thus was able to cope with the larger selection of scramblers. Thanks to Alan Turing, a brilliant young mathematician from Cambridge University, Bletchley began to crack the Enigma cipher. As the weeks passed, Turing realized that Bletchley had accumulated a massive library of decrypted messages, many of which conformed to a rigid structure. He believed that by studying old messages he could predict part of the contents of an encrypted message based on when it was sent and its source.

COMPUTER CRYPTOGRAPHY

With the advent of digital computers after World War II, governments retired electromechanical devices. In 2012, the U.S. National Security Agency declassified a 1955 handwritten letter by Princeton University mathematician John Nash about an offer he had made in 1950 to help design a new encryption machine.[8] In the letter, Nash made a distinction between polynomial time and exponential time, and conjectured that some problems could not be solved faster than in exponential time. He used this as the basis to pitch a secure system of his own design. He also anticipated that proving so-called lower bounds of complexity was a difficult mathematical problem. Nash's letter predates a 1956 letter from Kurt Gödel to his Princeton colleague John von Neumann that goes into much less detail about complexity but also anticipates complexity theory and the still-unsolved P versus NP problem.[9]

The NSA, however, pointed out to Nash that his machine "affords only limited security." When Nash wrote his letter, Colonel William Frederick Friedman, the leading U.S. cryptographer in World War II, responsible for breaking the Japanese Purple Cipher and developing the theory that allowed cryptanalysis of rotor-type machines, declared that no new encryption system was worth looking at unless it came from someone who had already broken a very hard one. He rejected Nash's plan.[10]

> *In the early days of computers, encryption was restricted mostly to the government and the military.*

Using a computer to encipher a message — to convert it into a cipher — is similar to traditional forms of encryption, with a few differences. First, a mechanical device is limited by what can be built, whereas a computer can mimic a hypothetical cipher machine of immense complexity. Second, computers are much faster than mechanical devices. Third, and perhaps most significant, a computer scrambles numbers rather than letters. Computers deal in binary number sequences of ones and zeros; before encryption, any message must be converted into binary digits. This conversion can be performed according to various protocols, such as the American Standard Code for Information Interchange (ASCII).

In the early days of computers, encryption was restricted mostly to the government and the military. By the 1960s, businesses could afford computers and use them to encrypt important communications such as money transfers or trade negotiations. However, as commercial encryption spread, cryptographers were confronted with a major problem: key distribution.

ASYMMETRIC KEYS

Say a bank wants to send confidential data to a client via a telephone line but is worried that someone is tapping the wire. The bank picks a key and uses encryption software to scramble the message. To decrypt the message, the client needs to have a copy of the encryption software and know which key was used to encrypt the message. How does the bank inform a client of the key? It can't send the key via telephone lines, which are insecure. The only truly secure way is to hand it over in person. In the 1970s, banks attempted to distribute keys by employing dispatch riders who personally distributed keys to everyone who would receive messages over the next week. As business networks grew, more keys had to be delivered, creating a pricey, insecure logistical nightmare.

In 1958, the U.S. Department of Defense launched the Advanced Research Projects Agency (ARPA). One of ARPA's projects was to find a way to connect military computers across long distances; the agency then developed ARPANET, which evolved into the internet. The nascent internet required a new level of electronic security. Whitfield Diffie, an MIT-trained cryptographer and computer

security expert, wondered how two people meeting on the internet could send each other encrypted messages, such as an email containing encrypted credit card details.

Key distribution suffers from a classic catch-22. If two people want to exchange a secret message, the sender must encrypt it. That means using a key, which is itself a secret.

In 1976, Diffie and Martin Hellman at Stanford University published a paper describing a new type of cipher that incorporated a so-called asymmetric key.[11] All the encryption techniques were symmetric, which meant that unscrambling the code was simply the opposite of scrambling the message. In an asymmetric key system, encryption and decryption keys are not identical. In an asymmetric cipher, if senders know the encryption keys, they can encrypt messages but they cannot decrypt them. To decrypt, they need access to the decryption key. Computer encryption means the encryption key is a number and the decryption key a different number. The sender keeps the decryption key secret, so it is commonly referred to as a private key. However, the sender publishes the encryption key so that everybody has access to it; that's the public key.

Ron Rivest, an MIT cryptographer working with colleagues Adi Shamir and Leonard Adleman, focused in 1977 on building an asymmetric cipher that contained a one-way function used to encrypt a message — a system that remains ubiquitous in electronic communication. A number is plugged into a mathematical function, and the result is the ciphertext, another number. There is a particularly interesting aspect of the function, known as *N*, that makes it reversible under certain circumstances and is therefore ideal for use as an asymmetric cipher. This flexible aspect means that each person can choose a different value of *N* and personalize the one-way function.[12]

For example, to choose a personal value of *N*, the sender picks two prime numbers, *p* and *q*, and multiplies them. Say *p* equals 17,159 and *q* equals 10,247. Multiplying them together gives *N* = 175,828,273. (This example comes from Singh.) This becomes the public encryption key. If someone wants to encrypt a message to the sender, they look up the value of *N*, then insert it into the one-way function. The receiver now has a one-way function tailored with the sender's public key. But how can the initial sender decrypt the message? Rivest's one-way function is reversible only by someone *who knows the values* of p and *q* — that is, the initial sender. In fact, it turns out that if *N* is large enough, it is virtually impossible to deduce *p* and *q* from *N*; this is perhaps the most elegant aspect of the RSA asymmetric cipher.

> *Singh argued that only a quantum computer could break the hold that RSA's private keys have over secure communications.*

The security of RSA depends on how difficult it is to factor N, because that is what you would have to do to find p and q. Two decades ago, Singh noted that for important banking transactions N tends to be at least 10308. With sufficiently large values of p and q, RSA appears to be impenetrable. The only caveat for public-key cryptography is that at some point someone may find a quick way to factor N and render RSA useless.

Which brings us back to quantum computing and its mate, quantum cryptography.

THE QUANTUM DIFFERENCE

Singh published his history of cryptography in 1999. His last chapter focused on quantum computing, describing quantum characteristics such as superposition, entanglement, teleportation and uncertainty. Singh argued that only a quantum computer could break the hold that RSA's private keys have over secure communications.[13] And he outlined the development of quantum cryptography and communications in the 1970s by Stephen Wiesner, then a Columbia University graduate student, who proposed the use of quantum uncertainty to detect intrusion in communications lines.[14] In the 1980s, IBM's Charles Bennett and Gilles Brassard of the University of Montreal designed what they envisioned as an unbreakable encryption key using polarized light, building on another Wiesner idea, known as quantum money, and his uncertainty error-detection concept.[15] Singh maintained that quantum cryptography would prove not only unbreakable but "absolutely unbreakable" as long as quantum physics remained viable as a theory.

This was a profound break in a long history. By 1999, cryptography had become all about the math. In the past 40 years, most high-level security schemes, such as RSA, have had at their heart excruciatingly difficult factoring problems. As Joël Alwen of cryptographer company Wickr wrote, each of the basic security schemes "comes equipped with a 'security proof,' which is a formal mathematical proof showing that breaking the scheme's security is at least as hard as solving one of the fundamental math problems like factoring . . . As the security of much of our digital infrastructure amongst other things rests on the watertight security of these building blocks, confidence in the proof is vital." But, added Alwen, "here's the problem: If someone ever builds a large scale quantum computer, this entire narrative collapses."[16]

Today, quantum computing is no longer just a concept. Private and public funding have poured into QC projects around the world. In late 2019, *Nature* published a study of quantum computer funding,

The deeper question is whether the advent of quantum computing marks the end of the historical pendulum swings between code makers and code breakers.

noting that private investors, mostly venture capital, had backed at least 52 QC companies since 2012.[17] That doesn't include the huge amounts of money invested by large companies including Google parent Alphabet, Hewlett-Packard, IBM and Chinese giants such as Alibaba, Tencent, Baidu and Huawei, as well as academic and governmental organizations. This spending is not just on the Holy Grail of general-purpose quantum computing; venture money has also gone into software companies developing QC algorithms or quantum computers for specialized purposes. That funding is producing results — and fears of quantum hype and a QC version of the artificial intelligence winter, a period when the hopes for AI seemed to wither.[18] In October 2019, for example, a Google team claimed it had achieved "quantum supremacy" over classical computing, building a machine with 54 qubits — quantum versions of individual bits that can simultaneously be 0 or 1 — completing calculations that would have taken the best supercomputer 10,000 years to complete. As impressive as that was, experts calculate that it will require a minimum of 1 million qubits to break RSA.[19]

There's already an algorithm that could perform those calculations. (Singh notes that an *algorithm* embodies a general encryption method, while the *key* specifies the exact details.) In 1995, MIT applied mathematician Peter Shor published an algorithm that showed how a hypothetical quantum machine could quickly solve prime factorization problems.[20] A year later, he published a paper on how to process information as qubits through an error-correction technique that could adjust for quantum uncertainty — an extension of Wiesner's work. In 2020, Shor noted that progress had been made, but he offered several nuances. First, if anyone breaks RSA first, he said, it will probably be the National Security Agency or "some other large organization." Second, there are easier and quicker ways to break internet security than building quantum computers. Although Shor believes quantum computing will succeed and that post-quantum crypto systems will eventually replace RSA, he warns that the effort will be enormous, and "if we wait too long, it will be too late."[21]

This is what Jon Lindsay, a University of Toronto professor of political science, calls "the window of vulnerability."[22] Lindsay describes two approaches that the U.S. National Institute of Standards and Technology (NIST) has taken to close that window. The first is what's known as classical post-quantum cryptography (PQC), which uses mathematical problems that are not vulnerable to Shor's algorithm and thus not threatened by either classical or quantum computers. The transition to PQC is already underway in the U.S. and will take a decade or more. The second is quantum key distribution (QKD) for communications networks; this approach uses the work that came from Wiesner's invention of what he called quantum money (which used quantum cryptography to create and validate banknotes) and a technique for detecting errors along communications lines that suggest the presence of intruders. QKD could allow the long-range distribution of private keys, but it's not quite as far along for practical use as PQC.

Given all this, Lindsay is mildly optimistic that a cryptocalypse can be avoided. The response by NIST, as well as increased funding by the U.S. government, suggest that there's a growing awareness of the perils.

UNBREAKABLE?

The deeper question is whether the advent of quantum computing and quantum cryptography marks the end of the historical pendulum swings between code makers and code breakers. Singh was definitive: Quantum computing will, in the end, provide victory for the code makers — and security. "Quantum cryptography is an unbreakable system of encryption," he wrote. He admitted that, historically, code makers right up to Rivest and his colleagues expressed confidence that their schemes were unbreakable, but that proved not to be. Singh's reasoning is based on the belief that quantum theory "is the most successful theory in the history of physics." In particular — echoes of QKD — any attempt at deciphering would be detected. And if it wasn't, wrote Singh, "it would mean quantum theory is flawed, which would have devastating implications for physicists; they would be forced to reconsider their understanding of how the universe operates at the most fundamental level."[23]

Lindsay, for his part, is more cautious. "QKD is hardly a silver bullet," he wrote.[24] The same mechanism that prevents copying would enable adversaries to "impose service denial attack on a quantum channel." He then ticked off a number of other possibilities, including that eternal reality, human gullibility. Real quantum computing could still be decades away, he noted: "No technical advantage can be sustained forever, if indeed it can be realized in the first place." The secret will get out. And a secret loose in the world will eventually be blunted, commoditized or rendered inoperative.

There's also another possibility, that the ability to keep matters secret may not, in the long run, be a question of quantum computing's ability to bust a mathematical lock through brute-force calculations. It's called homomorphic encryption. Stanford University's Craig Gentry first laid out the construction of the method in his PhD dissertation, "A Fully Homomorphic Encryption

Scheme," in 2009.[25] Gentry was responding to an idea first envisioned by Rivest, Adleman and Michael Dertouzos in 1978, soon after the introduction of RSA.[26] Gentry's scheme allows the customers of, say, cloud computing providers to access their data in a way that the data and computational results can only be decrypted by a secret key held by the customer; the processor of the data sees only encrypted data. Gentry's conceptual breakthrough was particularly suited to the accelerating shift in data and data processing to cloud providers like Amazon Web Services, Microsoft Azure, Google Cloud, IBM and others. The term "homomorphic" refers to a similarity in form, without an actual relationship in terms of structure or function.

Homomorphic encryption is hardly a universal answer to cybersecurity in the age of quantum computing, but it may well be one of the answers, particularly in terms of protecting data.[27] At the heart of homomorphic encryption is more math: Gentry and others use so-called lattice problems in crypto algorithms, which have long proved difficult to solve, thus providing a replacement for the public-key factorization problems that quantum computing makes so vulnerable. Gentry's initial scheme was focused on classical computing. But homomorphic encryption approaches were quickly developed for quantum computing — notably, so-called blind quantum computing, in which, as one early paper said, both "the input, the computation, or the output are unreadable by the computer" — that is, the QC.[28]

This is quantum computing on defense, not offense. And so the pendulum swings. ■

Béla Kosztin is a Vice President, Research, at WorldQuant and has a PhD in applied mathematics from Keele University in Staffordshire, England.

ENDNOTES

1. Simon Singh. *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography* (New York: Anchor Books, 1999).
2. Ronald L. Rivest, Adi Shamir and Len Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM* 21, no. 2 (February 1978): 120–126.
3. Juliette Garside. "Philip Zimmermann: King of Encryption Reveals His Fears of Privacy." *Guardian* (May 25, 2015).
4. Richard P. Feynman. "Stimulating Physics with Computers." *International Journal of Theoretical Physics* 21, nos. 6–7 (1982).
5. Jon R. Lindsay. "Surviving the Quantum Cryptocalypse." *Strategic Studies Quarterly* 14, no. 2 (Summer 2020): 49–73.
6. Singh, *The Code Book*, xiii.
7. Paul Lunde. *The Secrets of Codes: Understanding the World of Hidden Messages* (Richmond, CA: Weldon Owen, 2012).
8. Noam Nisan. "John Nash's Letter to the NSA." *Turing's Invisible Hand: Computation, Economics, and Game Theory* blog (February 17, 2012).
9. Stephen Budiansky. *Journey to the Edge of Reason: The Life of Kurt Gödel* (New York, NY: Norton, 2021): 258.
10. Singh, *The Code Book*, 92–94.
11. Whitfield Diffie and Martin E. Hellman. "New Directions in Cryptography." *IEEE Transactions on Information Theory* 22, no. 6 (November 1976): 644–654.
12. Rivest, Shamir, Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems."
13. Singh, *The Code Book*, 332–339.
14. Stephen Wiesner. "Conjugate Coding." *SIGACT News* 15, no. 1 (Winter-Spring 1983).
15. Charles H. Bennett and Gilles Brassard. "Quantum Cryptography: Public Key Distribution and Coin Tossing." *International Conference on Computers, Systems and Signal Processing* (December 1984): 174–179.
16. Joël Alwen. "What Is Lattice-Based Cryptography & Why Should You Care." Wyckr Crypto+Privacy blog, (June 15, 2018).
17. Elizabeth Gibney. "Quantum Gold Rush: The Private Funding Pouring into Quantum Start-ups." *Nature* (October 2, 2019).
18. Davide Castelvecchi. "The Quantum Internet Has Arrived (and It Hasn't)." *Nature* (February 14, 2018).
19. Elizabeth Gibney. "Hello Quantum World! Google Publishes Landmark Quantum Supremacy Claim." *Nature* (October 23, 2019).
20. Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer." *SIAM Journal on Scientific Computing* 26, no. 5 (October 1997): 1448.
21. Davide Castelvecchi. "Quantum-Computing Pioneer Warns of Complacency over Internet Security." *Nature* (October 30, 2020).
22. Lindsay, "Surviving the Quantum Cryptocalypse," 50.
23. Singh, *The Code Book*, 349.
24. Lindsay, "Surviving the Quantum Cryptocalypse," 60.
25. Craig Gentry. "A Fully Homomorphic Encryption Scheme." PhD dissertation for Stanford University Computer Science Department (September 2009).
26. Ronald L. Rivest, Len Adleman and Michael L. Dertouzos. "On Data Banks and Privacy Homomorphisms." Academic Press (1978).
27. Jonas Zeuner, Ioannis Pitsios, Si-Hui Tan, Aditya N. Sharma, Joseph F. Fitzsimons, Roberto Osellame and Philip Walther. "Experimental Quantum Homomorphic Encryption." *Quantum Information* 7 (2021).
28. Stefanie Barz, Elham Kashefi, Anne Broadbent, Joseph F. Fitzsimons, Anton Zeilinger and Philip Walther. "Demonstration of Blind Quantum Computing." *Science* 335, no. 6066 (January 2012): 303–308.