Why We Need Quantum Cryptography

Word Count:
2948

Why We Need Quantum Cryptography

In 1994, at the Institute of Electrical and Electronics Engineers' 35th Annual Symposium, Peter Shor announced his creation of a startling piece of mathematics. Known as 'Shor's Algorithm', it outlined how a quantum computer could quickly factorise a number into its two prime factors. Suddenly, the effectively unbreakable methods of most encryption systems were at the mercy of the development of these quantum computers.[1]

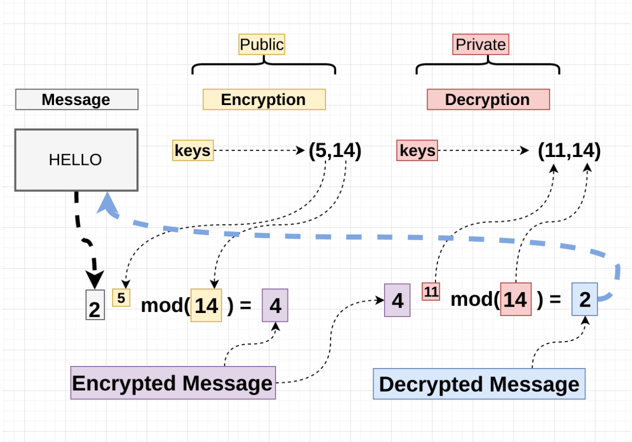
All encryption works on the basis of 'keys', which do something to systematically alter a message. For example, shifting the letters in a text by one letter in the alphabet would be a very basic example of an encryption key. In computing, all messages are numerical, so the keys take the form of mathematical functions. Since the late 1970s, almost all secure data transmissions via the Internet, such as instant messages, bank transactions or even login details, have been encrypted using a technique called RSA encryption.[2, 3]

RSA works by using two keys, a 'private' key and a 'public' key. If Alice wants to send a message to Bob, Bob's computer will generate the two keys and send his public key to Alice, which anyone can access. However, his computer will keep hold of the private key, which is what will decrypt Alice's message.

RSA is powerful because the public key encrypts the message using a one-way mathematical function. This works by using the modulo, or remainder function. For example, the remainder of 8 divided by 3 is 2, but even if you know that the result is 2 and the divisor is 3, it is impossible to recover the original number, 8. This means that Alice can encrypt her message using the public key, but if someone else also knows the public key and intercepts the encrypted message, then they can't decrypt it.

Both keys are made up of two numbers. The first number is an exponent, to which either the unencoded 'plaintext' message is raised in the case of the public key, or to which the encoded 'ciphertext' is raised by the private key. The larger number is what the exponents are divided by, to find the remainder, and is shared between the two keys.

For example, Bob could send Alice a public key consisting of the numbers (5, 14), and create a private key consisting of (11, 14). If Alice wanted to transmit the message 'Hello', represented by the number '2', then the public key would first raise 2^5, which gives 32. The remainder of 32 divided by 14 gives 4. Bob therefore receives the encrypted message of '4'. To decrypt it, he applies the same operations as Alice did to encrypt the message, just using the numbers in his key. 4^{11} gives 4194304, and the remainder of 4194304 divided by 14 gives 2 – the original message. Importantly, these steps could be attempted with *any* message, as long as it is an integer.

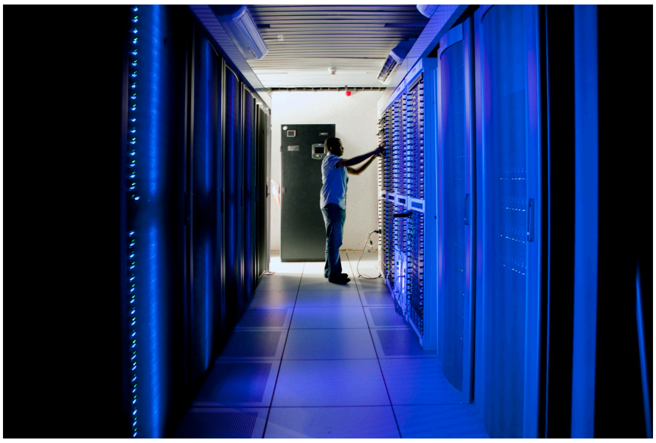


In this example of RSA, the message 'HELLO' is represented by the number 2. The message is encrypted using the public key but can only be decrypted using the private key.[4]

The larger number in the two keys is actually the multiple of two prime numbers, known as a semiprime. The two factors are what determine the first number in the public and private keys. If the factors are found, the first number in the private key can be found, and the message decrypted. In the example above, it is obvious what the two prime factors are, 2 and 7.

However, if you were asked to determine the prime factors of 701,111, you would struggle even with a calculator. Of course, there are algorithms for computers to find prime factors, which could tell you that the factors are 907 and 773 in a trivial amount of time.

Because of this, RSA uses much larger numbers.[5] Modern RSA uses 2048 bit semiprimes, which correspond to 617 decimal digits. For comparison, in 2020, a 250 digit, 829 bit RSA semiprime was factorised by an international team, which took them approximately 2700 years of CPU time, running across tens of thousands of computers for a few months.[6, 7] It turns out that the time required to factorise semiprimes increases at an almost exponential rate to the number of bits.[8] Using a 2048 bit semiprime therefore appears to be absolutely secure.



A supercomputer at the CNRS institute in France, one of the institutes which participated in factorising the RSA-250 semiprime. [9]

Enter Shor's Algorithm. This algorithm used quantum computing to take advantage of a special property of the factorisation process, called 'period-finding'.

One of the most basic number sequences is the powers of 2:

2, 4, 8, 16, 32, 64, 128, 256, 512, …

If we divide each number in that series by a number composed of two prime factors, for example 15, and find the remainder, we get a new series:

2, 4, 8, 1, 2, 4, 8, 1, 2, 4, …

This sequence is periodic, with a period of 4. If we choose a different divisor, for example 14, we get a different series:

2, 4, 8, 2, 4, 8, 2, 4, 8, …

This sequence has a period of 3.

It turns out that we can actually predict the period of the sequence. According to Scott Aaronson, Professor of Computer Science at the University of Texas:

> 'There's a beautiful pattern discovered by Euler in the 1760s. Let N be a product of two prime numbers, p and q, and consider the sequence:
> $x \bmod N$, $x^2 \bmod N$, $x^3 \bmod N$, $x^4 \bmod N$, …
> Then provided x is not divisible by p or q, the above sequence will repeat with some period that evenly divides (p-1)(q-1).'

So when N is 15, (p-1)(q-1) is 8. The period of the sequence is 4, which is a divisor of 8. In the case where N is 14, (p-1)(q-1) is 6, and the period of the sequence is 3, which divides 6. If we were to try other values of x, we could find other divisors of (p-1)(q-1), and put them together to find (p-1)(q-1), and ultimately find p and q.[10]

Unfortunately, when p and q are large, the period of the sequence can be almost as large as N. In fact, using classical computers, finding the period of the sequence is as difficult as factoring N itself![10, 11]

This is where we can exploit the properties of quantum computing. Quantum computers use quantum bits, or qubits. These qubits, like a normal computer bit, represent a state '0' or a state '1'. However, unlike normal bits, it is also possible to form a linear combination of states $\psi = \alpha(0) + \beta(1)$, where ψ represents the state of the qubit. α and β are complex numbers, so the state of a qubit can be thought of as a vector in a two-dimensional *complex* vector space. The states 0 and 1 form an orthonormal basis for this vector space.

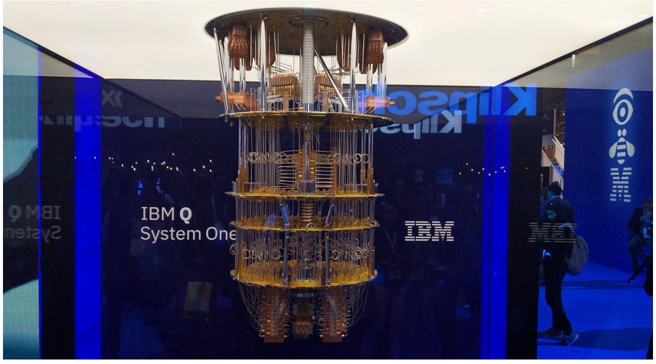
This linear combination of states is what is known as a *superposition* of states. However, a physical measurement cannot return a complex value, and when the state of a qubit is measured, the qubit can only return a state of either 0 or 1. The probability of measuring a state 0 is given by $|\alpha|^2$ and the probability of measuring a state 1 is $|\beta|^2$. As the probabilities must sum to one, $|\alpha|^2 + |\beta|^2 = 1$.[12]

However, qubit states can be manipulated and transformed so the measurements observed depend on the state of the qubit, and these manipulations and transformations lie at the heart of quantum computing.[12]

To complete the factorisation of N, a superposition of every single x^n mod N is created. One of the misconceptions about quantum computing is that once this superposition is complete, the computer can try and divide N by all of the different numbers at once, therefore speeding up the process. In fact, the quantum computer speeds up the calculation by analysing the properties of the superposition, and therefore the entire system, as a whole. It does this by performing a Quantum Fourier Transform (QFT) on the superposition. Much like how a Fourier Transform of an audio signal can convert a waveform with a certain period into a frequency spectrum of that waveform (and vice-versa), a QFT converts a 'waveform' (the superimposed qubit), which has a certain period of repetition into a signal showing the 'frequencies' making up that qubit. The peak 'frequencies' correspond to the period of the sequence, which can then be simply found and manipulated to find the factors of N.[10, 11, 12]

It turns out that carrying out this superposition and QFT actually factorises semiprimes exponentially faster than any classical computer can, as it analyses the entire system at once rather than every number individually. A quantum computer could therefore crack RSA codes in a very manageable time, rendering most of the world's cybersecurity utterly useless.[10, 12]

Quantum computers have already used Shor's algorithm to factorise semiprimes successfully; in 2019, a team in the US used a seven qubit quantum computer built by IBM to factor 35, the largest semiprime number to be factored using Shor's algorithm.[13] While 35 is far too small to be used in an RSA key, this demonstrates that the algorithm is indeed something that can be put into practice, rather than purely a theoretical concept. The security of RSA therefore depends only on the progress of quantum computation.



The IBM Q System One Quantum Computer, the 'first fully-integrated commercial quantum computer', used to factorise 35 using Shor's Algorithm.[14]

Back in the 1970s, even before RSA was invented, scientists were investigating how to use quantum physics to encrypt information.[15, 16] By 1991, three years before Shor published his algorithm, a protocol for a quantum encryption technique was released. Known as quantum key distribution, or QKD, it harnesses a strange property of quantum physics – quantum entanglement.[17]

In 1927, Werner Heisenberg discovered a fundamental result of quantum mechanics. This, known as Heisenberg's Uncertainty Principle, states that it is impossible for the position and momentum of any system to be measured with complete precision. The more precisely the position of a system is determined, the less precisely can its momentum be predicted.[18] In fact, quantum systems can be described mathematically using a 'wavefunction'. This wavefunction represents the 'probability amplitudes' of the state of a system. Most commonly, this wavefunction uses a position basis, so the wavefunction predicts the probability of where the particle is in space, although this can be changed to other bases, such as the momentum basis. Finally, another cornerstone of quantum mechanics is that measuring the property of a particle fundamentally changes the state of that particle.

For example, say the wavefunction, or probability amplitude, of the position of an electron was a Gaussian centred around x = 0, with a standard deviation of 1. If the particle is measured to be at x = 2 (which will have a low probability), then the state of the particle will change, to a Gaussian centred around 2. This makes sense: it has just been measured to be there, so of course it's now most likely to be found at x = 2! But according to the Uncertainty Principle, there will be an uncertainty in the measurement, which determines the width of the new Gaussian. After measuring the particle, there will be no way to tell what the previous wavefunction was, so some information will be lost. This process is known as 'wavefunction collapse'. So to sum up: quantum mechanical processes are inherently probabilistic, and if a system is measured then its state will change.

One of the consequences of the probabilistic nature of quantum mechanics is that most probabilities are not independent, but dependent. If two particles interact with each other, their wavefunctions will change, and they will change based on the other's wavefunction. This dependance is known as 'entanglement'.[12, 19]

The most drastic form of entanglement is the forming of what is known as an Einstein-Podolsky-Rosen, or EPR pair.[20] EPR pairs are special, because the state (in the case of a qubit, whether it is a 0 or a 1) is copied to both, so that both are, and remain, in the same state. Importantly, the state which they are in cannot be known before measuring them, but once one is measured, the state of the other can be determined with complete certainty to be the same state as the first one. This is true no matter how far apart they travel, as long as they do not interact with another particle before the measurements occur.[12]

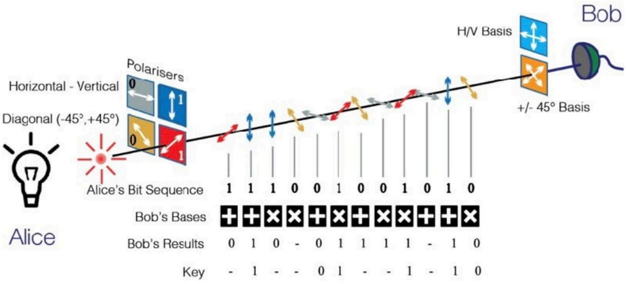
Now, if Alice wants to send a message to Bob, she could create a large set of entangled qubit pairs. If she sends one of each pair in order to Bob and keeps the other of the pairs in the same order, Alice and Bob will have the same qubit string. Alice and Bob can then both measure those qubits, which by definition will be the same, as they were entangled. Alice can then encrypt her message using that key, and Bob can decrypt it using the same key.[12] What makes it so useful, and so much better than RSA, is that it is completely secure from outside observation, providing the laws of physics as they are currently known are true.[16]

The only way an attacker could determine the key is by measuring the qubits. But in doing so, the state of the qubit will be altered! By the time it gets to Bob, the qubit will be in a different state to the state that Alice's is in, so his key won't work. In fact, Alice could add random digits at random intervals into the key she is transmitting. Once Bob has received the key, Alice could announce the location and value of those random numbers; if Bob has a different number at that point, then the key has been breached and someone is intercepting the transmission. Alice and Bob would have to create another key, but the breach would be detected before any message is sent.[12, 15]

In reality, the qubits are usually encoded in the form of polarised photons. For instance, a vertically polarised photon could represent a '1', and a horizontally polarised photon could represent a '0'. However, if an attacker placed a polariser angled exactly the right way, so that a vertically polarised photon was guaranteed to pass through and a horizontally polarised one was guaranteed not to, then the attacker could still gain information about the key, as he hasn't changed the number of vertical photons arriving at Bob's polariser.

To counter this, both Alice and Bob use a second basis of polarity, where '1' is represented by a 45º angle and '0' is represented by a -45º angle. The attacker is limited to one polariser, as once light has gone through a polariser it has no memory of its previous polarisation; if a vertically polarised beam of light falls upon a polariser at 45º to it, the light that passes through will be polarised at 45º.

Therefore, Alice randomly changes her bases and Bob randomly changes his bases during the transmission, and at the end of the transmission Alice publishes the bases she used.



Alice's Bit Sequence	1	1	1	0	0	1	0	0	1	0	1	0
Bob's Bases	+	+	×	×	+	×	+	×	×	+	+	×
Bob's Results	0	1	0	-	0	1	1	1	1	-	1	0
Key	-	1	-	-	0	1	-	-	1	-	1	0

An example of how Alice and Bob could change bases to create a key using quantum key distribution.[14]

For the sake of argument, consider that the attacker is using a vertical basis for his polariser. In this case, there are three possibilities. If Alice is using the vertical basis and Bob is using the vertical basis, then the attacker can determine the polarity of the photon without either Alice or Bob knowing. If Alice and Bob use different bases, then that photon will be discarded. They know they used different bases so Bob's measurement will be random and useless. If Alice uses the angled basis and Bob uses the angled basis, then the vertical polariser will allow 50% of both the '0' state and the '1' state photons to pass through. If Alice and Bob then compare the random numbers Alice added to the key, some of these will differ, and they will know that they have been attacked. Alice then won't transmit the secret message at all, as they have discovered that the key has been breached. There is therefore no way an attacker could decrypt a secret message encoded by a quantum key. [12, 15]

Like Shor's Algorithm, QKD proof-of-concepts have been succesfully implemented. In 2004, a Chinese team achieved a QKD transmission through a 404 km long coiled fiberoptical wire, and in 2019, another team managed to transmit a QKD signal between ground stations 1120 km apart.[21, 22] However, this required a specially built quantum satellite, and QKD transmission using more conventional, and significantly cheaper fiberoptic cable between two different ground stations has only been achieved across a distance of 144 km.[23]

Quantum cryptography is becoming reality. The complete security of QKD will become necessary once quantum computers have developed to a point where they can crack RSA encryption. The technology is being constantly developed; the European Commission has created its Quantum Flagship program which launched a QKD testbed in 2019, bringing it closer to becoming a commercial success.[15]

Quantum computing is also expected to develop rapidly. According to John Preskill, Professor of Theoretical Physics and Director of the Institute for Quantum Information and Matter at Caltech, we are nearing the creation of 50-100 qubit quantum computers. While he warns us that these 'will not change the world right away', he suggests that they may 'surpass the capabilities of today's classical digital computers…we should regard it as a significant step toward the more powerful quantum technologies of the future'.[24]

References:

1. P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*. [Online] 1994. p. 124–134. Available from: doi:10.1109/SFCS.1994.365700
2. Simmons G. *RSA Encryption*. [Online] Encyclopedia Britannica. Available from: https://www.britannica.com/topic/RSA-encryption [Accessed: 6th January 2021]
3. Rivest RL, Shamir A, Adleman L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. 1978; Available from: doi:10.21236/ada606588 [Accessed: 6th January 2021]
4. Garcia J. *How Does RSA Work? | Hacker Noon*. [Online] Hackernoon. Available from: https://hackernoon.com/how-does-rsa-work-f44918df914b [Accessed: 7th January 2021]
5. Lake J. *What Is RSA Encryption And How Does It Work? | Comparitech*. [Online] Comparitech. Available from: https://www.comparitech.com/blog/information-security/rsa-encryption/ [Accessed: 7th January 2021]
6. Zimmerman P. *[Cado-nfs-discuss] Factorization Of RSA-250*. [Online] Inria. Available from: https://lists.gforge.inria.fr/pipermail/cado-nfs-discuss/2020-February/001166.html [Accessed: 7th January 2021]
7. Patringenaru I. *New Record Set For Cryptographic Challenge*. [Online] phys.org. Available from: https://phys.org/news/2020-03-cryptographic.html [Accessed: 7th January 2021]
8. Thomé E. *[Cado-nfs-discuss] 795-bit Factoring And Discrete Logarithms*. [Online] Inira. Available from: https://lists.gforge.inria.fr/pipermail/cado-nfs-discuss/2019-December/001139.html [Accessed: 7th January 2021]
9. Fresillon C. The new Jean-Zay supercomputer, acquired in January 2019, is located at the CNRS Institute for Development and Resources in Computer Science (Idris). Available from: https://news.cnrs.fr/sites/default/files/styles/lightbox-hd/public/assets/images/math_industrie_calculhp_2012n00852.jpg

10. Aaronson S. *Shtetl-Optimized » Blog Archive » Shor, I'll Do It*. [Online] scottaaronson.com. Available from: https://www.scottaaronson.com/blog/?p=208 [Accessed: 8th January 2021]
11. Liu R. *Shor's Algorithm*. [Online] North Carolina State University. Available from: https://riliu.math.ncsu.edu/437/notes3se4.html [Accessed: 8th January 2021]
12. Nielsen MA, Chuang IL. *Quantum Computation and Quantum Information*. Cambridge University Press; 2010. p. 13, p. 216–238.
13. Amico M, Saleem ZH, Kumph M. Experimental study of Shor's factoring algorithm using the IBM Q Experience. *Physical review. A*. 2019;100(1).
14. Cardinal D. *IBM Unveils Q System One Quantum Computer - ExtremeTech*. [Online] ExtremeTech. Available from: https://www.extremetech.com/extreme/283427-quantum-computing-goes-commercial-with-ibms-q-system-one [Accessed: 11th January 2021]
15. Quantum Flagship. *Quantum Key Distribution (QKD) - Quantum Technology*. [Online] European Commission. Available from: https://qt.eu/discover-quantum/underlying-principles/quantum-key-distribution-qkd/ [Accessed: 10th January 2021]
16. Bennett CH, Brassard G. Quantum cryptography: Public key distribution and coin tossing. Bangalore: International Conference on Computers, Systems and Signal Processing; 1984. p. 175–179.
17. Ekert AK. Quantum cryptography based on Bell's theorem. *Physical Review Letters* . Ridge, NY: American Physical Society; 1991;67(6): 661–663.
18. Heisenberg W. Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. *Z. Physik*. [Online] 1927;43(3–4): 172–198. Available from: doi:10.1007/BF01397280 [Accessed: 10th January 2021]
19. Wilczek F. *Your Simple (Yes, Simple) Guide To Quantum Entanglement*. [Online] Wired. Available from: https://www.wired.com/2016/05/simple-yes-simple-guide-quantum-entanglement/ [Accessed: 10th January 2021]
20. Einstein A, Podolsky B, Rosen N. Can Quantum-Mechanical Description of Physical Reality Be Considered Complete? *Physical Review* . 1935;47(10): 777–780.
21. Yin H-L, Chen T-Y, Yu Z-W, Liu H, You L-X, Zhou Y-H, et al. Measurement-Device-Independent Quantum Key Distribution Over a 404 km Optical Fiber. *Physical Review Letters* . United States; 2016;117(19): 190501–190501.
22. Yin J, Li Y-H, Liao S-K, Yang M, Cao Y, Zhang L, et al. Entanglement-based secure quantum cryptography over 1,120 kilometres. *Nature*. [Online] 2020;582(7813): 501–505. Available from: doi:10.1038/s41586-020-2401-y [Accessed: 11th January 2021]
23. Schmitt-Manderbach T, Weier H, Fürst M, Ursin R, Tiefenbacher F, Scheidl T, et al. Experimental demonstration of free-space decoy-state quantum key distribution over 144 km. *Physical Review Letters* . United States; 2007;98(1): 010504–010504.
24. Preskill J. Quantum Computing in the NISQ era and beyond. *Quantum* . [Online] Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften; 2018;2: 79. Available from: doi:10.22331/q-2018-08-06-79 [Accessed: 11th January 2021]