Secret Codes, II: new developments

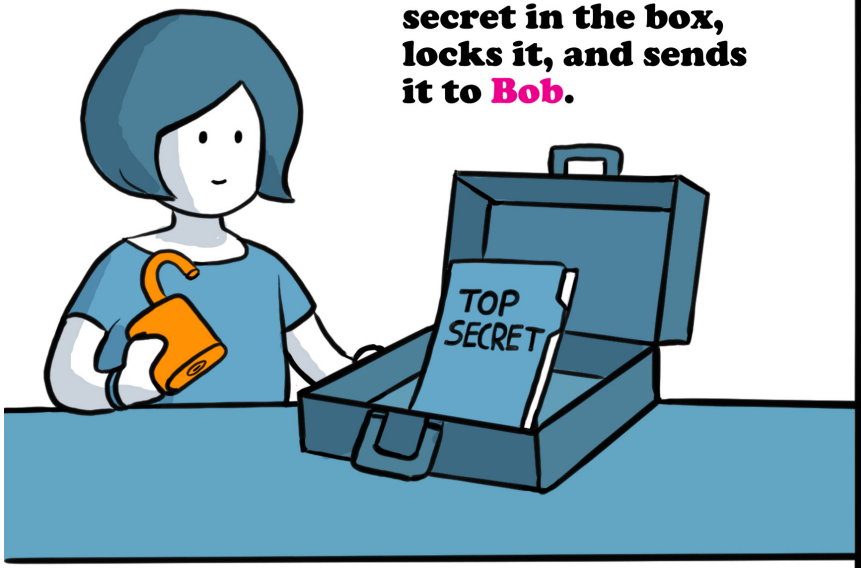
Peter J. Cameron

How the Light Gets In
Hay-on-Wye, June 2016

The problem of key exchange

A cryptosystem is no better than the security of the key. If increasingly large keys have to be shared between remote participants, key security becomes an increasing problem.
In the early 1970s, Diffie and Hellman came up with a really wonderful idea. Perhaps they thought like this. Alice has to send a valuable object (the key) to Bob. She doesn't trust the postman, Eve, who is likely to open the parcel and steal the key. So what does she do?
Alice puts the key in a chest, puts a padlock on the chest, and sends the chest to Bob. Bob cannot open Alice's padlock, but he puts on his own padlock and sends the doubly locked chest back to Alice. At this point, Alice can remove her padlock, and send the chest back, secured by Bob's padlock. When Bob receives it, he can unlock his padlock and open the chest.
The illustration following is by Neill Cameron (`www.neillcameron.com`).

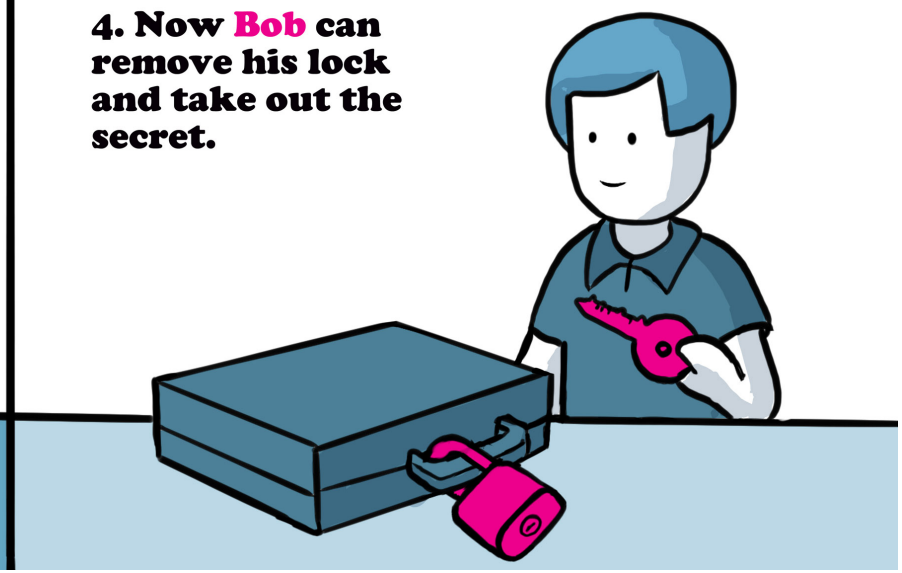
1. Alice puts the secret in the box, locks it, and sends it to Bob.



2. Bob adds his lock and returns it to Alice.

3. Alice removes her lock and returns it to Bob.

4. Now Bob can remove his lock and take out the secret.



Diffie–Hellman key exchange

This scheme would not work if Bob put Alice's locked chest inside another chest and locked that one. The crucial requirement is that the two operations "Alice locks chest" and "Bob locks chest" commute: they can be performed in either order, the result will be the same.
To take an example, putting on your shoes and your socks don't commute. So if you put on your socks and then your shoes, you cannot then take off your socks before your shoes!
So Diffie and Hellman solved the problem of key exchange by finding mathematical functions which are equivalent to the padlocks. The requirements are:

- each operation is easy to perform in both directions if the key is known, but very hard to undo if the key is not known (so that Eve, without the keys, cannot open the chest);
- the two operations commute with each other.

But commutativity is not enough.
In the first lecture we saw a one-time pad using binary addition. Let us suppose that Alice and Bob decide to use this method. Suppose the text to be exchanged is p, Alice's key is k_A, and Bob's key is k_B.
According to the protocol, Alice sends $p + k_A$ to Bob; Bob sends $p + k_A + k_B$ to Alice; and Alice sends
$(p + k_A + k_B) - k_A = p + k_B$ back to Bob. Then Bob subtracts his key and gets $(p + k_B) - k_B = p$. So far, so good.
If Eve intercepts just one of these messages, she can do nothing. But we must assume that she intercepts all three. Then she needs no cleverness at all to get the text p, just a little algebra. Add the first and third and subtract the second:

$$(p + k_A) + (p + k_B) - (p + k_A + k_B) = p.$$

Diffie–Hellman key exchange, 2

The operations Diffie and Hellman chose were, essentially, raising a number to a power. Easy to do, hard to undo (do you remember how to work out square roots by hand?)
The commuting requirement says that

$$(n^a)^b = (n^b)^a,$$

which is true.
(I have cheated slightly. Taking powers of numbers gives unreasonably large results, so they used a system called modular arithmetic to stop the numbers growing too large. But this is not a maths lecture . . .)

Trapdoor one-way functions

Mathematicians were used to asking whether a problem (like squaring the circle) could or could not be solved. They were much less familiar with the idea that the method to solve a problem might be well known but the labour required so large that, for practical purposes, the problem was unsolvable.
Just before the work of Diffie and Hellman, people like Jack Edmonds, Stephen Cook, and Richard Karp had introduced the ideas of the hardness of a mathematical problem, ready for the cryptographers to take up.
A trapdoor one-way function is a mathematical function (that is, a black box turning inputs into outputs) with the following properties:

- given an input, it is easy to calculate the output;
- given an output, it is hard to calculate the input, but there is an additional piece of information (the *key*) which makes this calculation easy.

Public-key cryptography

The next brilliant idea was that, given a rich supply of trapdoor one-way functions, it is possible to do more than just key exchange; one can create a full-blown cryptosystem in which key exchange is not necessary!

It works like this. Each user chooses a one-way function and a key to make inverting it easy. They publish their function (which is used for encryption) in a public directory.

So suppose that Alice wants to send a message to Bob. She looks up Bob's encryption function e_B in the directory, and uses this to encrypt her plaintext message p to Bob as $e_B(p)$, which she then sends to Bob.

Bob can decrypt the message since he possesses the key. But Eve doesn't have the key (Bob doesn't have to send it to anyone, and can keep it secure). So although in principle she can decrypt the message, in practice this takes so long (hundreds of years, say) that when she gets the answer, it is no longer of any use.

RSA

Merkle and Hellman suggested a possible class of one-way functions which could be used. But a much better class was proposed a couple of years later by Rivest, Shamir and Adleman. This quickly gained acceptance.

The RSA scheme went back to Diffie's and Hellman's idea of raising numbers to a power. The trick was that the modulus used for the modular arithmetic should be the product of two huge prime numbers. The key is the factorisation of this modulus: knowing this allows the inverse of taking powers to be done easily.

Thus, the security of RSA relies on the fact that, while it is fairly easy to find two large prime numbers, and very easy to multiply them, the reverse problem of breaking a composite number into its prime factors is very hard (as far as we know).

Adleman also took time out to "debunk" the Merkle–Hellman system. Whether or not he was right, this had the effect of killing it off as a commercial venture.

Factorisation

We have known since the time of Euclid that every whole number can be factorised into prime factors in a unique way (up to the order of the factors).
The problem of finding such a factorisation has fascinated mathematicians for centuries. Many methods have been proposed, including the general number field seive, Pollard's rho, and Fermat's method.
As the state of the art improves, users of RSA are forced to choose larger and larger moduli to keep the cipher secure.
Fortunately, large primes are easy to find: there are efficient tests for primality, so just pick large numbers at random until you discover a prime.

The alternative history

That is the story as it would have been told until the end of the 1990s.
In 1997, GCHQ put an announcement on their website saying that public-key cryptography and key exchange had been invented at GCHQ a few years earlier than the well-known developments in the USA.
In a nutshell, James Ellis had come up with the notion of public-key cryptography in 1970; Clifford Cocks had found a way to implement it; and Malcolm Williamson had invented Diffie-Hellman key exchange. The work was never made publicly available.
Ironically, Ellis died a month before the public announcement was made.

Diffie and Ellis

In a lecture on the history of cryptography, Whitfield Diffie told how he had heard rumours (possibly from NSA) of the independent British invention.
He went to Cheltenham, met up with James Ellis, took him to a pub, and bought him plenty of scrumpy cider. Their conversation ranged over many things, but according to Diffie, Ellis didn't breathe a word about his discovery many years earlier.
Credit doesn't always go to the right person, but in this case, the two American teams not only came up with the ideas but developed them into a working cryptosystem, while GCHQ did not. So Diffie, Hellman, Rivest, Shamir and Adleman probably deserve the credit.

Signing a message

The RSA formalism also allows messages to be signed; the signature gives a guarantee that the message really comes from the supposed sender.
Suppose that Alice wants to send a signed message to Bob. First she does a strange thing. She regards her plaintext as if it were a ciphertext addressed to her, and *decrypts* it: in other words, she applies her decryption function d_A to it. Then she encrypts it with Bob's public key, calculating $e_B(d_A(p))$, and sends this to Bob.
Bob decrypts with his secret key, obtaining $d_A(p)$. This is gibberish, but Alice has told him separately to expect a signed message from her. So he *encrypts* it with Alice's public key, giving $e_A(d_A(p))$.

Signing a message, 2

Remember the commuting property of the RSA system. According to this,

$$e_A(d_A(p)) = d_A(e_A(p)) = p,$$

since encrypting and then decrypting p gives back the original p.
So now Bob can read the message. But the fact that he can read it also tells him that it must have been encrypted with Alice's private key e_A. Since nobody but Alice has access to this key, Bob knows that the message really did come from Alice. Furthermore, Bob can keep a copy of $d_A(p)$. If at some later date Alice claims that she didn't send the message, Bob has the evidence to prove that she did!

An opportunity for the cryptanalysts

In some ways, public key cryptography makes Eve's life easier. In the old system, each message must be tackled independently. (If you know about Enigma, remember that the key changed each day, and until the day's key had been found, no messages could be read.)

But now, Eve can read Bob's public key; if she can "reverse engineer" his private key, she can read all messages addressed to Bob. This can be done independently of the messages being sent, and all the computing power that Eve has can be thrown at the problem. If solved, then old messages can also be read. (This is important in law enforcement, for example, as we have seen in recent cases involving iPhones.)

So the old discipline, such as changing keys frequently, is still as important as ever!

Not the answer to everything

Public-key cryptography offers enormous advantages: at a stroke it solves the problem of key distribution and allows messages to be signed.
But we noted that improvements in factoring algorithms keep pushing RSA users to larger and larger primes. The operation of RSA is computationally intensive, even though it can be done. Old-fashioned stream ciphers are computationally much simpler.
So typically, public-key cryptography is used by Alice to send Bob a secret key for a stream cipher, which they can then use in their communication.

Secret sharing

There are many other aspects of cryptography that impinge on us. For example, a bank president wants to ensure against a rogue director going to the vault and taking the money, and so wants a system where the presence of more than one director is required to open the vault. Moreover, even if some of the directors get together, they can deduce no information about the passwords held by the remaining directors. This is easy. Just give each director a random password and require that they are all entered.
But it may be impractical to get all the directors together. If there are n directors, and we require that any m can open the vault, but $m-1$ cannot get any information about the passwords they don't have, can it be done?
Yes, finite geometry provides us with a solution to this problem.

Enter quantum theory

The biggest change on the horizon for cryptography is likely to be caused by quantum theory.
This theory, invented in the 1930s, is the most accurate physical theory ever derived, and makes predictions which have been tested to many places of decimals by careful experiment.
But it is also a puzzling theory, which denies that subatomic particles are "little hard balls", giving them a much more shadowy existence. The famous physicist Richard Feynmann said,

> *There was a time when the newspapers said that only twelve men understood the theory of relativity. I do not believe there ever was such a time . . . On the other hand, I think I can safely say that nobody understands quantum mechanics.*

Quantum basics

It is important to understand that quantum theory describes whole systems rather than individual particles. Also, it involves complex numbers, a number system including the square root of -1 which has many good properties lacked by the real numbers (e.g. any polynomial equation has a solution). The philosophical difficulties of quantum mechanics arise from the fact that we treat a system under observation in the laboratory as being separate from the system of the observer and the measuring instruments she uses to study it. Some sciientists speculate that, if we were to treat the entire universe as a single quantum system, these difficulties would disappear. But science proceeds by reduction ... so I will consider a system of n elementary particles in isolation in a magnetic field. The only property I will be interested in is their spin, which (according to quantum rules) is aligned with the magnetic field in one of two directions, "up" and "down". We can use these two directions to represent the bits (binary digits) 1 and 0.

Quantum basics, 2

But rather than the particle being definitely in one state or the other, the rules of quantum mechanics allow it to be in a "mixture" or "superposition" of the two states: so much "up" and so much "down", or mathematically $p \cdot \mathbf{Up} + q \cdot \mathbf{Down}$. Here p and q are complex numbers and satisfy the condition $|p|^2 + |q|^2 = 1$. The rules then specify that, if we measure the spin, then with probability $|p|^2$ we will find that it is up, and with probability $|q|^2$ we will find that it is down.

Moreover, after the measurement, if we actually found that the particle spin was up, then the particle will be in the state **Up**: that is, the description changes discontinuously when we make a measurement. This is called the collapse of the wave function, and is probably the biggest philosophical mystery in quantum mechanics.

Thus one particle is described by a 2-dimensional complex space. The rules also show that a system of n particles is described by a space of dimension 2^n, called a tensor product of the state spaces for the individual particles.

Quantum computation

How do we factorise a whole number N into prime factors? The simple answer is trial division: divide N by 2, 3, ..., until either we find a number which divides exactly (in which case we divide it out and have a smaller number to deal with) or we get up to $\sqrt{N}$ (when we can be sure that the number is prime). Trial division for a number of 100 digits can thus take up to 10^{50} steps, and would take far longer than the age of the universe even if each step could be carried out in a nanosecond. The sophisticated improvements we saw earlier don't affect this estimate much.

But a quantum system of n particles is represented by a state space of dimension 2^n. As long as $2^n > \sqrt{N}$, that is, n is bigger than about 170, we can assign a dimension to each trial divisor and arrange that only those corresponding to factors of N will survive to the output.

This is the basis of quantum computation.

Shor's theorem

The notion of building a quantum computer received a big push in 1994, when Peter Shor showed that indeed a quantum computer could factorise large numbers much more quickly than a classical computer (in time which is a polynomial in $\log N$ rather than of order $\sqrt{N}$).
In other words, if we could build a quantum computer, then factorising large numbers would become practical, the RSA cipher would no longer be secure, and the whole of internet commerce would be killed off overnight.
Clearly this hasn't happened yet! Could it happen in future?

Building a quantum computer

Since Shor's Theorem was proved (and similar theorems for other public key cryptosystems), many groups of scientists and engineers have tried to build a quantum computer.
There have been some small successes. The first quantum computer which could find the factors of 15 (this happened in 2001) raised some interest, but progress has been slow since then. However, work continues: documents revealed by Ed Snowden confirm that the NSA was attempting to develop a quantum computation facility. (Maybe it already exists?)
The problem is that the quantum system must be isolated from its environment while it does the computation, and then reconnected for the readout. Some scientists suspect that there is a relationship between this "time to decoherence" and the clock speed of the computer. This would put a physical limit on what can be achieved by a quantum computer. We don't know.

Building a quantum computer, 2

Other teams all over the world are rushing to build a practical quantum computer:

Quantum computers don't do everything

There are some problems, such as factorisation, and the discrete logarithm problem (used in the El-Gamal cipher and Diffie–Hellman key exchange) which quantum computers can solve very fast. But there are other problems where they offer no obvious speed-up.

So some cryptographers are looking back at some old ciphers, such as the Merkle–Hellman knapsack cipher, and the McEleice cipher based on error correction, which were abandoned in favour of RSA, to see whether they might be the basis of a public key system not vulnerable to a quantum computer.

Both these are based on trapdoor one-way functions. The knapsack problem asks, given a knapsack of capacity b and a number of items with sizes $a_1, \ldots, a_n$, we can choose a set of a_is which together exactly fill the knapsack – hard in general, but easy in some special cases. The secret key hides the structure of these special cases.

Quantum cryptography

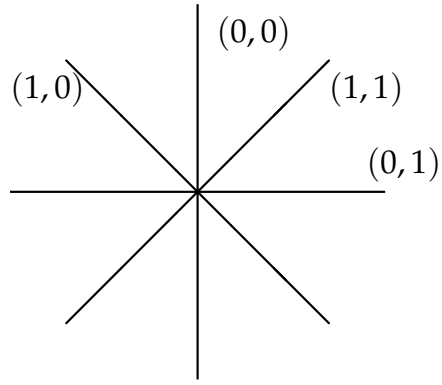
However, while quantum theory raises the spectre of breaking RSA and similar ciphers, it also brings the possibility of an unbreakable cipher whose key distribution is based on quantum theory.
The essential principle is that, when we observe a quantum system, we change its state in an unpredictable way. So Eve cannot intercept Alice's transmission to Bob without leaving her fingerprints on it: Alice and Bob can check whether Eve has intercepted the key transmission, and discard it if so.
I will describe the process in a bit more detail. As just suggested, Alice uses the quantum channel to send a "session key" to Bob; if this key is a random string, then Alice and Bob can communicate securely over a conventional channel using it as a one-time pad.

Quantum cryptography 2

We use photons ("particles of light"), with the advantage that they can be transmitted over optical fibre.
To simplify slightly, a photon has a direction of polarisation, perpendicular to its direction of travel. We can prepare a photon with a given direction of polarisation; we can then measure polarisation in any two perpendicular directions. If one of these directions corresponds to the prepared direction, the answer is "yes" for that direction and "no" for the other. But if it makes an angle of 45° with the prepared direction, the answer is "yes" to one and "no" to the other, randomly with probability 1/2 for each outcome.

Quantum cryptography, 3

Alice prepares two random binary sequences of length N, say $a_1 \ldots a_N$ and $b_1 \ldots b_N$. She sends N photons to Bob, polarised according to the values of (a_i, b_i) in the following scheme:



The first number chooses orthogonal or diagonal axes, the second specifies one of the two axes.
Bob chooses a random sequence $c_1 \ldots c_N$. If $c_i = 0$, he measures the polarisation in the vertical and horizontal directions, and writes $d_i = 0$ if he finds horizontal, $d_i = 1$ for vertical. If $c_i = 1$, he measures in the two diaagonal directions, and puts $d_i = 0$ for NW-SE and $d_i = 1$ for NE-SW.

Quantum cryptography, 4

Now Alice and Bob communicate over an ordinary (maybe insecure) line. Alice reads out her a sequence and Bob his c sequence. They will agree about half the time, since they are random; positions where they disagree are discarded. It follows that $b_i = d_i$ on the remaining positions, so Alice and Bob have shared a key of length about $N/2$.

To check for tampering by Eve, Alice and Bob sacrifice a certain number of bits. Whatever Eve does, about half the time her interference will change the polarisation of the photon, and half of the time this happens Bob will detect it. So by comparing a randomly chosen set of bits, say $2n$ (where n is much smaller than N) and revealing these bits to each other, Eve's effect will be visible except for the $(3/4)^n$ chance that "the coin came down heads every time". Taking $n = 241$, the chance of this happening is smaller than 1 in 2^{30}, much less than experimental error.

Not there yet . . .

This is not yet the answer to a cryptographer's dreams. Apart from the trouble and expense required to communicate in this way, our technology is not yet good enough to reliably send just one photon at each step of the process. If a small group of photons are sent, Eve might trap one of them to observe and leave the others to continue to Bob.

Despite the problems, this appears to be a practical solution, and the implementation is further advanced than quantum computation.

Perhaps in the future we will all have small quantum devices on our desks, and sending secure messages to each other. But the lesson of history is that the cryptanalysts have always caught up with the cryptographers sooner or later . . .