Post Quantum Cryptography for Grandparents

Its actually not as complicated as it sounds. Let's get the maths over with first. Remember polynomials?

$$(x+1)(x+1)=x^2+2x+1$$

This would be an example of two first degree polynomials being multiplied together to create a second degree polynomial (or quadratic). In general two n-th degree polynomials when multiplied together create a $2n$-th degree polynomial result. Polynomials can also be added

$$(3x+5)+(5x+6) = 8x+11$$

Don't tell me that's hard! For the polynomial $8x+11$, the *coefficients* are 8 and 11.

Next consider polynomials where all of the coefficients are less than a fixed prime number q. If they ever get greater than q, they are reduced to their remainder when divided by q. So if $q=7$, then

$$(3x+5)+(5x+6)=x+4$$

because 8 leaves a remainder of 1 when divided by 7, and 11 leaves a remainder of 4 when divided by 7.

That's it for the maths. The next thing we will do is scale it up a little(!) Lets choose $q=12289$, and consider polynomials of degree 1024. Such a polynomial will look like

$$7862+8375x+2764x^2+\ldots\ 11765x^{1024}$$

We've shortened it a bit, but you get the idea. Again its easy to multiply two such polynomials, although obviously a computer is needed to do it. In fact due to the cunning choice for q and the degree as a power of 2 ($1024=2^{10}$), there is a particularly fast way to do the multiplication.

Now normally when we multiply two such polynomials, we get a 2048-th degree polynomial product. But here instead we chop this into two 1024-degree halves, and subtract the top half from the bottom half. That's our result, another 1024-degree polynomial.

So now we can quickly add, subtract and multiply 1024 degree polynomials to our hearts content in any order, generating 1024 degree polynomial results whose coefficients are all less than q.

We are now ready to do some crypto. First some notation. A polynomial as above with large coefficients we should denote as $F(x)$, but we will simply call it F. We will also make use of polynomials with small coefficients, like

$$3+6x+4x^2+..\ 7x^{1024}$$

We will denote these with lower case letters, like f. Note that they are small only in terms of their coefficients, they are still of high degree. We shall call a polynomial with large coefficients a "large polynomial", and a polynomial with small coefficients a "small polynomial".

Now consider this calculation with such polynomials

$$B=As+e$$

Given A, s and e, its easy to calculate B, its just a multiplication followed by an addition. However given B and A, its very hard to calculate s and e. Think about it for a while - or just take my word for it! In fact for the size of polynomials we are talking about here its impossible *even if you have a quantum computer!* We call the small polynomial e an error polynomial, and the small polynomial s is often a secret. The large polynomial A is often a globally known value.

OK let's do some crypto. Alice and Bob who have no prior arrangement or shared secrets (although they both know a public large polynomial A), nevertheless want to communicate in private. In fact Bob wants to transmit an encrypted message to Alice that only Alice can read.

First Bob encodes his message as a large polynomial M.

Alice then calculates $B=As+e$ where s and e are small secret and error polynomials of her own invention, and sends B to Bob. Bob calculates $U=At+f$ and $C=Bt+g+$M and sends U and C back to Alice, where t, f and g are small polynomials of his own invention.

Finally Alice calculates $C-Us$. Substituting for C and U, and then for B, this becomes $et+g+M-fs$ (there is some fortuitous cancellation). Check it for yourself.

At this stage the reader might well feel a little bewildered, and be wondering – so what? But this is the clever bit. Observe that in the expression $et+g+M-fs$, only M is a large polynomial. The other components are all small. So in effect M stands out from the "noise", and can thus be recovered by Alice. So Alice got the message, and anyone who eavesdropped their communication gets a problem they cannot possibly solve.

That's basically it. This is the NewHope post-quantum key exchange protocol as proposed by Alkim, Ducas, Poppelmann and Schwabe. The strength of the protocol depends on the difficulty of the so-called Ring Learning with Errors problem, which is a problem based on lattices, and for which there is no known effective quantum algorithm. There is of course a bit more to it (!) that given in this simple description, mainly concerning the ways in which the small polynomials are generated, and the statistical distribution of their small coefficients. This needs to be done carefully to avoid some attacks in some contexts. There is also the issue of effective encoding and decoding of the message M. But these are really just details.

At this stage I hope you are thinking – *that was surprisingly easy.* In fact post-quantum crypto, in my humble opinion, is often quite shallow mathematically. Its also blazingly fast. The downsize is that the amount of data that must be exchanged is relatively large – those large polynomials are seriously big chunks of data.

But it works!