The Diffie–Hellman–Merkle key exchange (or an overcomplicated way to exchange secrets during class)

By Muhammad Siddiqui

Note: In order to understand this essay it would be beneficial to have at least a basic grasp of modular arithmetic. This video by TomRocksMath is a great place to start.

Say you're in the middle of French class. It's almost two, and you can't, for the life of you, bother to keep up with the barrage of french vocabulary your teacher's throwing at you. You pick up a few words, baguette this, croissant that, but you've had enough. As you look towards the window, gazing upon the birds freely dancing in the sky, you, in what is an uncharacteristically brave moment, decide to take a stand...to fight against this violent system of oppression. You pull out your pen and paper and decide you want to send a note, a secret, to your friend seated beside you. But as you look around, you realise you're not the only one with this bright idea. Right as you're about to send the note over to your friend, one of your classmates gets caught. He stares at the teacher like a deer caught in headlights, you watch as your teacher begins to assault him with a load of near incomprehensible French, nearly decimating your eardrums into oblivion. But this only strengthens your resolve more. If you're going to sneak your friend a message, you're going to have to be smart about it. You're going to have to outplay the system. You can't risk getting caught like your classmate, to be humiliated and embarrassed in front of the entirety of the class...to have all your secrets exposed! So before you write your note, you realise you're going to need a plan. You and your friend will have to outsmart the teacher. And so you realise that the only way to securely communicate with each other is through the beauty and elegance of mathematics, through the sheer power of the Diffie-Hellman-Merkle algorithm.

But before you and your friend can begin conversing, you must decide on a key. To do this, both you and your friend (who we'll call John) must mutually agree on a shared, public integer g, as well as a prime number p. In practice, this integer g must be a primitive root modulo of p, which means that for each number n, the result of $g^n \ mod \ p$ is unique and distinct. As such, the result of the shared secret key (which will be generated from g) will be any number between 1 to $p - 1$. This characteristic is unique only for integers that are the primitive root modulo of p, as non-primitive roots will have repeated values within this range. More on primitive roots will be discussed later.

Once the prime number p and primitive root modulo g have been decided (in this case 7 and 3), the fun finally begins. You and John now choose a random secret integer (which must be hidden from each other!), a *(3)* and b *(4)* respectively to generate your own public keys A and B using the formula below.

$$public\ key = 3^n \ mod\ 7$$

Where n is a or b for you and John respectively. Giving you:

$$\mathrm{A} = 3^3 \ mod\ 7 = 6$$
$$\mathrm{B} = 3^4 \ mod\ 7 = 4$$

Once done, both of you will send your public key to the other so you can each compute the following result below, where s is the shared secret/private key. Here is an example of what you would do:

$$s = B^a \ mod\ 7 = 1$$

While John will do the same, using $s = A^b \ mod\ 7$ instead, which you may be surprised to find out also equals 1. This is because in order for both of you to freely communicate and encrypt/decrypt each other's messages, you must have the same private/secret key. As such, the value of s for the both of you must be the same. The mathematics behind this is explained here:

Since A can be written as:

$$A = g^a$$

A^b can also be written as:

$$s = A^b = g^{ab} = s$$

While the same applies, to B^a :

$$B^a = g^{ba} = A^b = g^{ab} = s$$

Now, that we've established both of you have the same secret key s, you can now use them to secretly encrypt and decrypt each other's messages, freely able to secretly chat away about what I’m sure will be an incredibly insightful and eye-opening conversation...all without the risk of getting caught by your frankly *careless* French teacher. This is because while A and B are public keys, a and b are not. As such your teacher will never be able to compute $s = g^{ab}$, the secret key.

In practice, even if your teacher were to try, finding the secret key a and b would be incredibly difficult, even when using the most sophisticated of technology. The reasoning behind this is due to the nature of this problem and the *discrete logarithm.*

Take this as a very simple analogy, to begin with. After your French class, you and John head home and bake a cake. Now, while you and John, being the incompetent young men you are, find the arduous task of making a cake to be stupidly difficult, for the average person this isn't really the case. All you have to do is get some eggs, some flour, etc. put it all in a pan and then bake it in the oven. It's simple enough. However despite this, once you bake the cake, it still would be very difficult (if not impossible) to reverse engineer the cake back into its initial raw ingredients. This is the nature of a discrete logarithm, a function that is demonstrably easy to compute, but incredibly difficult to undo.

The mathematics behind this lies in the Diffie–Hellman–Merkle key exchange and why we used the primitive root modulo of *p* in the first place. The nature of the primitive root modulo is that it gives us distinct values for *p* no matter what the value of *n* for $g^n \; mod\, p$ is. However, this only works, provided *n* is less than or equal to *p-1.* As such, this means that they're an infinitely many solutions of *n* for $g^n \; mod\, p$, provided *n* is greater than *p-1.* Take this as an example:

$$3^{7-1} \;\; mod \; 7 = 1$$

Now, given that 1^n for any *n* is simply *1*, we can assume:

$$3^{6n} \;\; mod \; 7 = 1^n$$

Using this, we can now observe this for any function with a value of *n* less than *p-1.*

$$3^3 \;\; mod \; 7 = 6$$

So, since we also know

$$3^{6n} \;\; mod \; 7 = 1^n$$

Multiplying both sides, we now have:

$$3^{3\,+\,6n} \;\; mod \; 7 = 1^n \;\; * \; 6 \; or$$
$$3^{3\,+6n} \;\; mod \; 7 = 6$$

Thus, the equation now has infinite solutions for the value of 6 since the value of *n* can be any integer. This results in an equation that while simple to compute, has an infinite number of solutions, each of which is equally likely to be true.

In this case, however, given *x mod p* only has 7 solutions this would be very *easy* to brute force (provided your frankly very determined French teacher has the willpower and time to do so). To prevent this the value of *p* must be greater than *600* digits, which is a value that is too large for computers to brute force within a reasonable amount of time.

Thus is the beauty of the Diffie–Hellman–Merkle key exchange, it's an algorithm that is incredibly easy to compute (even inattentive students like you and John can do it!) but deceivingly difficult to reverse engineer. It's a keen reminder that despite the vast goliaths of raw computational power we presently have, it still serves to be no match for the ingenious minds of the mathematician.

Further Reading: *https://www.youtube.com/watch?v=UaanzpCkc8c*

Sources/ Bibliography:

https://en.wikipedia.org/wiki/Primitive_root_modulo_n
https://www.youtube.com/watch?v=pa4osob1XOk
https://zerofruit.medium.com/diffie-hellman-key-exchange-724871ce78d9
https://www.youtube.com/watch?v=SL7J8hPKEWY
https://www.doc.ic.ac.uk/~mrh/330tutor/ch06s02.html
https://www.britannica.com/science/Fermats-theorem
https://www.youtube.com/watch?v=5OjZWSdxlU0