FAST MODULAR EXPONENTIATION

CHEFOR FOZO[1] AND AMY FEAVER[2]

1. Introduction

Modular exponentiation is a *one-way function.* Though there is a more formal and mathematically rigorous definition, it is clearest to describe the concept of a one-way function informally. This is a function where computing the function itself is very straightforward and fast enough to be computable in a reasonable amount of time. However, the function's inverse is more difficult to compute (usually significantly more difficult). [**?**]

A reason that we concern ourselves with a one-way function, from the perspective of computer science, is often for the purposes of cryptography. We want cryptographic functions to be one-way, so that it is easy to encrypt our data before sending it across a network. However, we don't want just anybody to be able to grab our encrypted data and unencrypt it; this is why we use a one-way function, as unencrypting (computing the inverse of the function) is difficult computationally.

We assume that the reader knows the basics of modular arithmetic (if you do not, there are many resources online to look over before reading this!).

Modular exponentiation takes three integers as input: a base g, an exponent x and a modulus m. The output of this function is an integer C, $0 \leq C < m$ such that

$$C = g^x \pmod{m}.$$

There are a number of algorithms for computing C quickly, even when the input g, x, m consists of very large integers (integers that are hundreds of digits in length, even).

However, if you try to recover the value of g, even knowing C, x and m, it is very difficult.

2. Something

Definition. The *discrete log problem*, or *DLP* is the problem of finding the smallest nonnegative value of x if it exists in the equation $g^x \equiv h \pmod{m}$ where $g, h \in \mathbb{Z}$ and $m \in \mathbb{N}$.

This seems like a pretty straightforward problem, but it is of huge importance in modern cryptography. This is because it is a *one-way function*: if you know x, g and m, there are relatively fast algorithms for computing $g^x \pmod{m}$. However, if you know g, h and m, it can be computationally difficult to figure out what x was. Of course, for this one-way function to work, we need to *make good choices.* For example, choosing $g = 1$ would be pretty useless, since no matter how many times we exponentiate g, the smallest nonnegative value for the exponent x in the discrete log problem would just be 0, so it's not difficult to recover the value x.

Imagine someone chose $g = 9$, $m = 43$ and they chose a value of x but kept it a secret from you. Imagine all they told you was that $9^x \equiv 24 \pmod{4}3$. If you wanted to find x, you would have to brute force it. So you could go ahead and calculate it as follows:

$9^1 \equiv 9 \pmod{43}$
$9^2 = 81 \equiv 38 \pmod{43}$
$9^3 = 9 \cdot 9^2 \equiv 9 \cdot 38 \equiv 342 \equiv 41 \pmod{43}$ (Note that here I substituted in 38 for 9^2 since they are both equivalent mod 43. This keeps the calculations a little simpler.)
$9^4 = 9 \cdot 9^3 \equiv 9 \cdot 41 \equiv 369 \equiv 25 \pmod{43}$
$9^5 = 9 \cdot 9^4 \equiv 9 \cdot 25 \equiv 225 \equiv 10 \pmod{43}$
$9^6 = 9 \cdot 9^5 \equiv 9 \cdot 10 \equiv 90 \equiv 4 \pmod{43}$.
You would have to keep repeating this process until you figured out that $9^{20} \equiv 24 \pmod{43}$.

Of course, you could write a computer program to check a bunch of exponents until you found x, and that would work since our numbers are very, very small in the world of cryptography. But doing this out by hand you would see that using brute force to find x is quite a bit of work.

Many cryptosystems rely on the fact that x is very difficult to compute in order to secure information!!

However, if the person who chose x wishes to calculate $9^x \pmod{43}$, this is something they can do relatively quickly. There are computational techniques to speed up this modular exponentiation. Let's look at one called *repeated squaring.*

Using repeated squaring for modular exponentiation. This is best shown through example, because if we tried to write out general rules for doing this we would end up with a lot of notation and mumbo-jumbo. Let's look at 20 and break it up in a way so that the summands are the largest powers of two we can use. That is,

$$20 = 16 + 4.$$

We will keep this in mind; we especially want to note the largest power of 2 here, which is 16, because when we do repeated squaring we will not have to exceed the 16th power of g. Now, we will calculate 9^2, 9^4, 9^8 and 8^{16} all mod 43, using a bit of cleverness to keep the calculations easier. We already know that $9^2 = 81 \equiv 38 \pmod{43}$. But also notice that $38 \equiv -5 \pmod{43}$. This is nice because -5 is an easier number to work with than 38. Let's proceed from here:
$9^4 = (9^2)^2 \equiv (-5)^2 \equiv 25 \pmod{43}$
$9^8 = (9^4)^2 \equiv 25^2 \equiv 625 \equiv 23 \pmod{43}$
$9^{16} = (9^8)^2 \equiv 23^2 \equiv (-20)^2 \equiv 400 \equiv 13 \pmod{43}$.

Now, we use these powers of 2 that we just computed in order to compute $9^{20} \pmod{43}$:

$$9^{20} = 9^{16+4} = 9^{16}9^4 \equiv 13 \cdot 25 \equiv 24 \pmod{43}.$$

You will note that we had to do a lot fewer calculations and exponentiations to figure out $9^{20} \pmod{43}$ than we had to do in order to figure out that $x = 20$ when we only had the information that $9^x \equiv 24 \pmod{4}3$.

From this example, we can start to see that if we choose g and m wisely (and very very large), we can publish the final result of $g^x \pmod{m}$ publicly, but nobody (not even someone with full access to a supercomputer) could figure out what x is without their calculations taking a very very long time (that is, millions of years or more).

References

[1] Robshaw M.J.B. (2011) One-Way Function. In: van Tilborg H.C.A., Jajodia S. (eds) Encyclopedia of Cryptography and Security. Springer, Boston, MA. `https://doi.org/10.1007/978-1-4419-5906-5_467`

[1] Department of Computing Science, The King's University, 9125 50 St NW, Edmonton, AB, Canada

Email address: `jcheforfozo@yahoo.com`

[2] Nomadic Mathematician

Email address: `mathfeaver@gmail.com`