Cryptography II

Whitfield and Martin had a secret they wanted to share and a ham radio. What to do what to do...

Math Circle

January 14, 2018

Let's resume out discussion of cryptography. This time I want to talk more about symmetric schemes, and in particular the Diffie-Hellman Key exchange algorithm. Remember the following definitions:

- A **Scheme** is the general name for a plan or algorithm to encrypt & decode a message. If you are trying to communicate secretly, then you should assume that everyone and their mother knows what scheme you are using.
- A **Key** is a number which you use together with a scheme to encrypt & decode a message. You can think of the key as being like the settings of the scheme. You should assume that the key is not known (like the scheme). A given scheme can have more than one key.
- **ciphertext** is what you get when you apply a scheme with a key to a secret message.

A symmetric scheme is one for which the same key is used to encrypt and decrypt the message.

1. A one-time pad cipher is an encryption scheme that has a key which is as long as the encoded message. You add each element in the key to the letters in the message pairwise, and then compute the result mod 26.
   (a) Easy question to warm up. If you have a key, and the ciphertext, how do you compute the secret message?

(b) If I want to send a message of length N, how many possible keys are there? How many possible ciphertexts are there?

(c) How can one decode a one-time cipher?

(d) Here is some ciphertext, and a key. What was the original message?

`gldwzyhurkkgdxaohvtnrtxyobtvtdoblriiqkb`

`thisisthekeyitcanbeanythingitskindofcoo`

*Hint This is a problem that you should probably solve with you table. Have each person work on deciphering a few letters.

(e) If the key has to be as long as a the message that you want to send, why would anyone use a one-time cipher? Isn't it useless?

(f) It can be very impractical to carry around a one-time pad; it has to be quite long to be effective. Here is an alternative. Pick a number $a_1 \in 1, \ldots, 25$, and let $a_k = (a_{k-1}^2 \text{mod} 26)$. Shift the kth letter of the text by a_k. Now instead of having to keep a huge document, you can just remember a single value. What do you think about this scheme?

(g) **Prove that the one-time cipher is the only really totally secure encryption method. That is, that without the key it's impossible to do anything with ciphertext**

(h) Suppose that your computer could, given a key and ciphertext, encrypt or decrypt 10^7 letters per second. How long would it take the computer (approximately) to encode/decode the message in question 1.d?

(i) Suppose that you wanted to used that same computer to decode the secret message, by trying every possible one-time pad. How long would it take encode/decode every single possible one-time pad cipher for that same cipher text? What if the message was twice as long?

One of the problems with one-time pad ciphers (and all symmetric schemes) is that the symmetric key has to be agreed on ahead of time. But, what if you want to communicate with someone new? In other words, how can two people agree/share a symmetric key without having a secure channel to communicate over? After all, if it was possible to share a key securely, you wouldn't need cryptography anyways, right?

To answer this question, we're going to watch another YouTube video! In fact, we are going to watch two! The first one is (slightly) less mathematical then the ones that we have watched previously, but is still quite instructive. It is called "Public key cryptography - Diffie-Hellman Key Exchange (full version)" from the channel "Art of the Problem."

https://www.youtube.com/watch?v=YEBfamv-_do

And the second one is from computerphile. It's called "Diffie Hellman -the Mathematics bit- Computerphile" by the channel "Computerphile".

https://www.youtube.com/watch?v=Yjrfm_oRO0w

2. Alright! Now let's take a step back and look at the things said in the videos a little bit more closely.

   (a) One fact that was used a LOT in both videos, but was never called out, was the following fact. If $m \equiv n \bmod(p)$ where p, n and p, m are coprime, then $m^i \equiv n^i \bmod(p)$ for any natural number i. Prove this fact. *Hint, it certainly seems true, and might look obvious but you may find that proving it is harder then you think.

   (b) One of the claims in the video was that it's 'easy' to compute a modular exponent. By that they mean that computing something like $2017^{100} \bmod(100)$ is easier than you probably think. Please compute $2017^{100} \bmod(100)$. *Hint, make gratuitous use of the above fact that you just proved.

(c) Suppose that you have your trillion operation/second computer from before. Let's say that you have to compute $g^a \bmod(p)$ where g and p are relatively prime. If multiplying any two numbers takes 1 operation, and taking a modulo takes 1 operation, how many operations does it do your computation?

(d) Now suppose that you are given $g^a \bmod(p)$ and want to compute the key, a. If the only way that you can do this is by computing $g^1 \bmod(p)$, $g^2 \bmod(p)$, ... how long would it take your super computer to find a?

(e) One way to 'attack' the Diffie-Hellman protocol is to do a so called man-in-the-middle attack. Suppose that there was not just an eavesdropper listening in to Alice and Bob's communication, but there was someone Christy who had the power to stop messaged from being delivered completely, and instead send her own messages. In this case, Christy is the (wo)man-in-the-middle, because she is in the middle of Alice and Bob's communication. Describe how Christy could furtively intercept, and change the messages so that although Alice and Bob both think that there messages are getting through securely, they are in fact being intercepted and read by Christy.