Worksheet on Primitive Roots, Discrete Logs, Diffie-Hellman

April 3, 2019

1 Primitive Roots

Recall that the orders of elements of $\mathbb{Z}/n\mathbb{Z}$ must divide $\phi(n)$.

Definition 1. *An element $a \in \mathbb{Z}/n\mathbb{Z}$ is a* primitive root *if a has order $\phi(n)$.*

1. Fill in the table for $\mathbb{Z}/7\mathbb{Z}$ (easiest if you reference your modular power tables from a past worksheet or the website):

x	order of x
1	1
2	3
3	
4	
5	
6	2

2. List all the primitive roots modulo 7.

2 Discrete Logarithm

Definition 2. *Let a be a primitive root modulo n. If $a^x \equiv b \pmod{n}$, then we say that $\log_a(b) = x$ in $\mathbb{Z}/n\mathbb{Z}$. This is called the* discrete logarithm *of b to the base a. It is also sometimes called the* index *and written $\mathrm{ind}_a(b)$.*

1. Explain why the discrete logarithm is only unique up to multiples of $\phi(n)$. For example, $3^0 \equiv 3^6 \equiv 1 \pmod{7}$ so that $\log_3(1)$ could be 0 or 6. (*In essence, the discrete logarithm lives modulo $\phi(n)$.*)

2. Choose the primitive root 3 modulo 7. Complete the table of powers:

x	3^x
0	1
1	3
2	2
3	
4	
5	
6	

3. Draw the dynamics of multiplication by 3 modulo 7. At each position, write the number itself *and* as a power of 3, e.g. $2 = 3^2$.

4. Complete the table of discrete logarithms:

b	$\log_3(b)$
0	not defined
1	0 (mod 6)
2	2 (mod 6)
3	
4	
5	
6	

5. Why is the first row 'not defined'?

6. Is it true that $\log_a(b_1 b_2) = \log_a(b_1) + \log_a(b_2)$? Explain why or why not.

7. Is it true that $\log_a(b^c) = c \log_a(b)$? Explain why or why not.

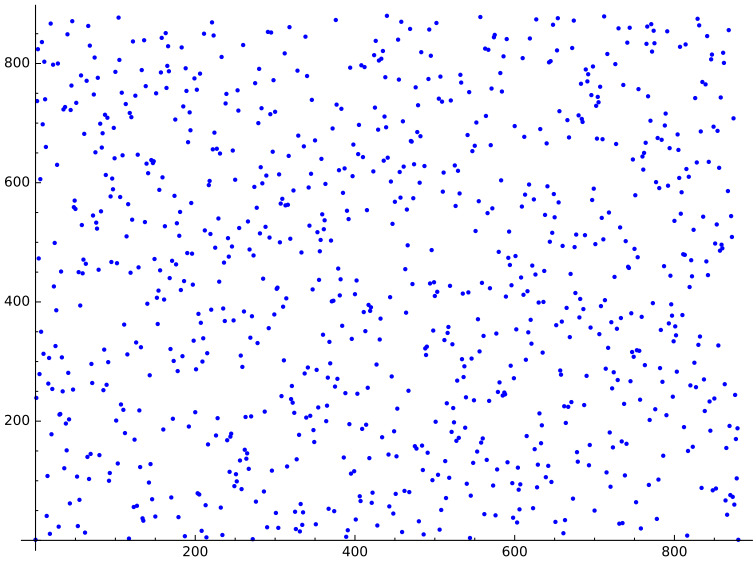
3 The Discrete Logarithm Problem

Definition 3. *The* discrete logarithm problem *is the problem of, given $g \in \mathbb{Z}/n\mathbb{Z}$, and some power $h = g^s \in \mathbb{Z}/n\mathbb{Z}$, determining s.*

This problem is believed to be extremely hard, when n is a large prime. In fact, it is believed to be humanly impossible when the size of prime we are talking about is around 2^{1024} (for comparison, the number of fundamental particles in the observable universe is estimated at 2^{280} or so). That's the size of prime used in cryptography on the internet today. Note that it takes only a thousand or so bits to write down the prime.

1. Solve the discrete logarithm problem for $g = 3$, $h = 5$ in $\mathbb{Z}/7\mathbb{Z}$.

2. Here is a plot of the function $239^x \pmod{881}$ (note that 239 is a primitive root modulo the prime 881).



What do you observe about this and what does it tell you about the discrete logarithm problem?

4 Diffie-Hellman Key Exchange

Now, you will use the discrete logarithm problem to set up a shared secret across a public telephone line. Sound impossible? Here's how it works.

Shared Information:

$p =$
$g =$

Alice:

$a =$
$A = g^a =$
$B^a =$

Bob:

$b =$
$B = g^b =$
$A^b =$

1. Get into a group of two people. Designate one person *Alice* and the other *Bob*. Designate an inanimate object of your choosing *Eve*. Eve is the Eve-il Eve-sdropper. The rest of you are nice, decent people. Place Eve in a prominent position where she can see the goings-on.
2. In the table *Shared Information* above, choose a prime of approximately 3 digits, and put it in the table under p. To do this, you can go to the Single Cell Sage Server (`sagecell.sagemath.org`; works on a phone; app also exists) and type `next_prime(x)` (but replace x with some 3-digit number of your choosing).
3. Find a primitive root modulo p. To do this, you can type `primitive_root(p)` where you replace p with your prime. Write this primitive root in for g.
4. Alice chooses a random secret number a between 1 and $p-1$ and Bob chooses a random secret number b in the same range. Keep them secret! Alice should write a in the *Alice* table on her paper and Bob should write b in the *Bob* table on his paper, and then cover them up secretively. Make sure Eve isn't able to see.
5. Alice should compute $g^a \pmod p$, call it A and enter it into her table. Bob should compute $g^b \pmod p$, call it B and enter it into his table. You can use the Sage server for this, e.g. `Mod(g^a,p)`.
6. With Eve listening in, Alice should now announce A loudly to Bob. Bob should write A in the *Alice* table on his paper. With Eve listening in, Bob should now announce B loudly to Alice. Alice should write B in the *Bob* table on her paper.
7. Alice should compute $B^a \pmod p$ since she knows B and a. Bob should compute $A^b \pmod p$ since he knows A and b. Do this secretively.
8. Note that Eve knows A and B (they were announced loudly), but doesn't know a and b (they were kept secret).
9. Finally, the great reveal: Alice and Bob should compare A^b with B^a. What do you notice?

10. These two values should have been equal. Prove that these two values must be equal.

11. Explain how Eve could compute the shared secret if she could solve the discrete logarithm problem.

12. Can you think of a use for this shared secret value? It isn't an encrypted message, since neither Alice nor Bob can control what it is. It just a random shared secret.

5 Now you are Eve

1. You are playing the role of Eve now. Find another group, locate their inanimate object Eve and interrogate it to discover the A and B of that group. Write it here:

$$A = \qquad\qquad B =$$

2. Determine, with the help of the Sage server, what the shared secret is. You might find it useful to ask Sage to do discrete logarithms. If you type `log( Mod(5,11), Mod(7,11) )`, Sage will respond with `2`, reflecting the fact that $\log_7(5) = 2$ or $7^2 \equiv 5 \pmod{11}$. The first time you do this, type `Mod(7^2, 11)` (or your equivalent; the answer in this case should be 5) to double-check you are getting it working right.

3. Check with the other group to see if you are correct:

Yes! Yay! No! Boo! (analyse and try again)

6 More food for thought (discuss)

1. Why is it feasible to do cryptography with primes of 1024 bits? Can Alice even compute g^a?

2. If we replace the integers modulo p with the plain integers, is the discrete logarithm problem hard or easy? Could we built cryptography on that?

3. Does Eve have any other ways to determine the shared secret besides doing a discrete logarithm?

4. What is the most efficient method for solving a discrete logarithm by hand? Come up with the best algorithm you can and compare to your friends.

5. How can you create a shared secret amongst three people, over a public channel (i.e. with Eve listening in)? Describe the process step-by-step.