Name(s)____ Period ______ Date ________________

Activity Guide - Public Key Cryptography

Introduction

This activity is similar to the “cups and beans” encryption we did in a previous lesson. However, instead of using cups and beans as analogy for math and one way functions, today we’ll use a version of the real thing.

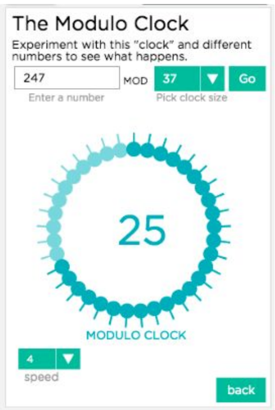
Clock Arithmetic (Modulo)

In class you should have reviewed modular arithmetic which we refer to as “clock arithmetic”. The thought is that if you have a clock with the numbers 1-12 on its face and you want to move the hour hand around the clock to count up to some number of hours the hour hand will always be pointing to a number 1-12, no matter how many hours have passed. We can count this way with any size “clock” and any number.

Do this: Experiment with the “Modulo Clock” in the Public Key Crypto Widget

First thing to do is to get familiar with the modulo operation by experimenting with the “Modulo Clock”. The widget helps you visualize the result of modular arithmetic and experiment with different clock sizes and numbers.

- Go to the Public Key Crypto Widget.
- Click on the “Modulo Clock” Button.
- Experiment with different numbers and clock sizes.
- What’s going on?



The Modulo Clock

Experiment with this "clock" and different numbers to see what happens.

247 MOD 37 Go

Enter a number — Pick clock size

The Modulo Clock widget showing the result of 247 MOD 37

What’s the point? Modulo as a One-Way Function

“Clock Arithmetic” (officially known as “modular arithmetic”) is a one-way function because you can’t tell what the number counted off was, just by looking at the clock face.

Terminology

The “clock” is a helpful visual and mental aid, but the real terms for what we’re doing are:

- **modulus** - the size, or number of ticks on the clock
- **modulo** or **mod** - the name of the mathematical operation for “wrapping” a number around a clock. For example, if we want to see the result of counting up to 247 on a clock of size 37, we would say that we are doing “247 MOD 37.”
- *Note:* The MOD operation is the same as finding the **remainder** when dividing the number by the clock size. So, 247 ÷ 37 = 6 with remainder 25. 247 MOD 37 = 25.

Welcome to the
Public Key Crypto Widget

First get to know the Modulo Clock

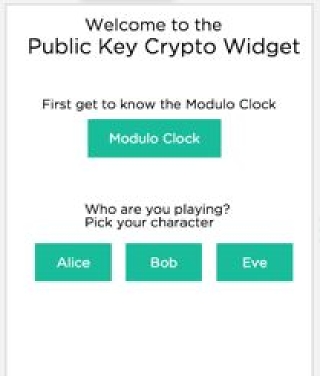
Using the modulo operation as a one-way function is one of the key ingredients behind the “real math” that makes public key cryptography possible. So now, instead of using cups and beans, let’s see how we can use modulo along with multiplication to make public and private keys that are difficult to crack, and have Bob and Alice send secret messages to each other over public channels.

Do this: Public Key Encryption Procedure

Setup

- Go back to the home screen of the Public Key Crypto Widget.
- Decide who is playing Alice, Bob and Eve, and click on the appropriate name to go to that character's screen. Each screen has the instructions for what that character should do.
- For the first round, Eve will read all the instructions out loud and Alice and Bob will follow along.
- Eve will intercept all messages going back and forth between Alice and Bob and try to crack them

Eve, read everything out loud, starting here...



Welcome to the
Public Key Crypto Widget

First get to know the Modulo Clock

Who are you playing?
Pick your character

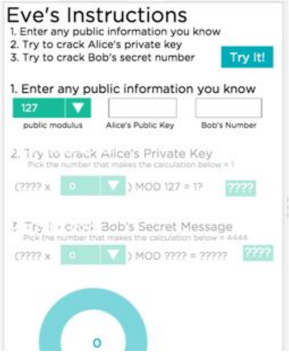
Home Screen

Eve...

In Public Key Cryptography some information is publicly known. One public piece of information is the size of the clock (the modulus) that will be used for encryption. Since it's publicly known, we'll even let Eve pick it. So...

1. Eve, first pick a clock size (modulus) that everyone else will use.
   - You can pick any number from the list provided. For the first round, we suggest choosing a number less than 200.

2. Eve, announce this number and ask Alice and Bob to also set their clocks to the same number.



Eve's Instructions

1. Enter any public information you know
2. Try to crack Alice's private key
3. Try to crack Bob's secret number

1. Enter any public information you know

public modulus: 127
Alice's Public Key:
Bob's Number:

2. Try to crack Alice's Private Key
Pick the number that makes the calculation below = 1

(???? x 0) MOD 127 = 1?

3. Try to crack Bob's Secret Message
Pick the number that makes the calculation below = 4444

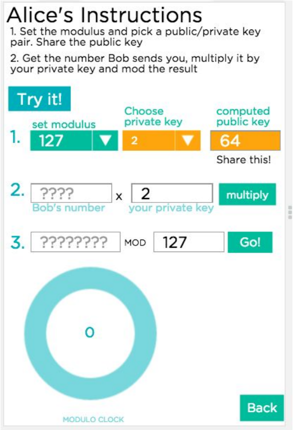
(???? x 0) MOD ???? = ?????

Eve's screen

Alice...

Alice, make sure Eve and Bob can't see your screen.

- After you set the modulus, you need to choose a private key. Pick one from the list provided and keep it private. Never tell anyone this number.
- After you choose your private key, you should see a public key that was generated.
- Please announce the public key out loud.
- Alice, Bob and Eve should all write this number down.
- *Write the Public Key here:*



Alice's Instructions

1. Set the modulus and pick a public/private key pair. Share the public key
2. Get the number Bob sends you, multiply it by your private key and mod the result

1. set modulus: 127 | Choose private key: 2 | computed public key: 64
Share this!

2. ???? (Bob's number) x 2 (your private key)

3. ???????? MOD 127

Alice's Screen

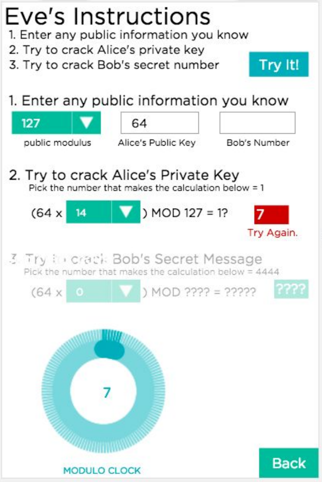
Eve…

Eve now has an opportunity to crack the Alice's private key. All Eve has to do is solve this problem: figure out what number to multiply the public key by, so when MODed by the clock size, it comes out to 1. Eve's screen in widget will help experiment. Here's the math.

- In the equation below, Eve knows 2 of the 3 numbers needed - the public key and clock size. The third number is the private key. Can you guess it? Good luck.

  (__________ x __________) MOD _________ = 1 ?
  public key   private key   clock size

- Eve's widget makes it a little easier for her to guess by doing the math for her.



Eve's Instructions
1. Enter any public information you know
2. Try to crack Alice's private key
3. Try to crack Bob's secret number

1. Enter any public information you know
127 (public modulus) 64 (Alice's Public Key) Bob's Number

2. Try to crack Alice's Private Key
Pick the number that makes the calculation below = 1
(64 x 14) MOD 127 = 1? 7
Try Again.

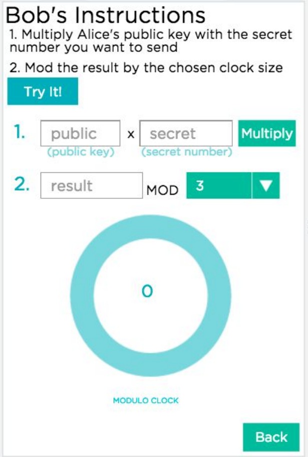
3. Try to crack Bob's Secret Message
Pick the number that makes the calculation below = 4444
(64 x 0) MOD ???? = ????? ????

Eve's Screen

Bob…

Bob, make sure that Eve and Alice can't see your screen while you do this:

- Enter the public key into the box provided on your screen.
- Enter a secret number to send to Alice. *Note:* It must be a number that is *less* than the clock size.
- Now multiply these two numbers together and mod the result by the clock size; the widget makes this easy.
- The number that comes up on the clock is the number you are sending to Alice in public.
- Announce the number you are sending to Alice. Alice and Eve can also write it down.
- *Write down Bob's number here:*



Bob's Instructions
1. Multiply Alice's public key with the secret number you want to send
2. Mod the result by the chosen clock size

1. public (public key) x secret (secret number)
2. result MOD 3

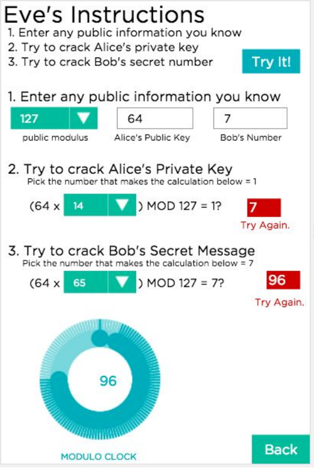
Bob's Screen

Eve…

You have another opportunity to crack the message here. All you have to do is solve a similar but slightly different problem: you have to figure out what number to multiply the public key by, so that when MODed by the clock size, it comes out to the number Bob just announced.

- This time you know 3 out of the 4 things you need - the public key, the clock size, and the number Bob just announced. Can you guess the secret number?

  (__________ x ______________) MOD _________ = ______________
  public key   secret number   clock size   Bob's number



Eve's Instructions
1. Enter any public information you know
2. Try to crack Alice's private key
3. Try to crack Bob's secret number

1. Enter any public information you know
127 (public modulus) 64 (Alice's Public Key) 7 (Bob's Number)

2. Try to crack Alice's Private Key
Pick the number that makes the calculation below = 1
(64 x 14) MOD 127 = 1? 7
Try Again.

3. Try to crack Bob's Secret Message
Pick the number that makes the calculation below = 7
(64 x 65) MOD 127 = 7? 96
Try Again.

Eve's Screen

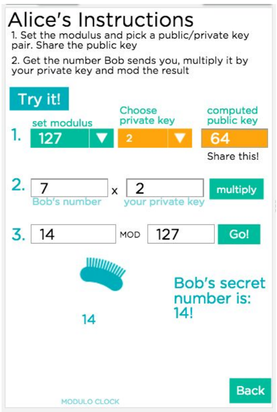
Alice…

You can now decrypt the message. Make sure Eve and Bob can't see your screen. Decrypting Bob's secret message is easy…

- Multiply your private key by the number Bob just announced; the widget helps you do this multiplication.

- Then MOD that result by the clock size.

- The number that results on the clock is the secret number Bob sent you!



Alice's Instructions

1. Set the modulus and pick a public/private key pair. Share the public key

2. Get the number Bob sends you, multiply it by your private key and mod the result

1. set modulus: 127 — Choose private key: 2 — computed public key: 64 Share this!

2. 7 (Bob's number) x 2 (your private key)

3. 14 MOD 127

14

Bob's secret number is: 14!

Recap:

- Alice and Bob did not have to agree on anything, or communicate ahead of time.
- Alice and Bob only exchanged information in public, right in front of Eve. Eve even chose one of the numbers that was used for calculations!
- Eve would have to guess either Alice's private key or the secret number Bob is trying to send.
- *Note:* The encrypted messages only go one way. If Alice wanted to send a message to Bob, Bob would have to generate his own public/private key pair.
- However, because the message can only go one way, the sender can feel safe that ONLY the intended recipient can decrypt the message.

Try it again?

- Change roles and try the procedure again to see how it works.
  - Alternatively, you can play all three roles, switching back and forth between the screens.
- See if you can do the exchange faster. It only takes a few seconds, once you know what to do.
  - Don't have Eve read the instructions. Just declare a clock size and Alice and Bob can start. Continue to "send" numbers by announcing them publicly.
- Try to make it harder for Eve:
  - Choose bigger numbers.
  - Eve should capture the numbers and start trying to crack them.

What do you actually need to know?

Public Key Cryptography can seem and feel complicated to think about. On a separate page, we explain the math behind the Public Key Crypto Widget and relate it to how a real asymmetric encryption scheme (called RSA) works. But if you don't care about the math, here are the key things you need to know about Public Key Cryptography:

- Cryptography has a mathematical foundation.
- It relies on asymmetric keys, which you can make using numbers and math.
- The modulo operation acts as a one-way function.
- When you multiply big numbers and mod them by other big numbers, it's really hard to figure out what the original numbers were; the technique is essentially reduced to random guessing.
- The security of publicly known encryption protocols is based on the fact that cracking a message by brute force would take an unreasonable amount of time.
- With a sufficiently large modulus (say, 256 bits, which would be roughly a 77-digit number), random guessing would take an unreasonable amount of time. Even if you had millions of computers working on it constantly, it would take trillions of years.
- Because the method of encryption is public, it actually increases the security, because good guys and bad guys know how hard it is to crack.