# Making and breaking codes: Cryptology

**Introduction:** Students will learn how to write and read secret messages using modular arithmetic, or "clock math," explore decryption with frequency analysis, and get a brief introduction to public key encryption. They will practice sending each other messages with each method, as well as answer questions like "how can you read a message without knowing the code?" and "how can you share a code without other people being able to read the message?"

**Content Objectives:**

- Students will understand how math helps encrypt and decrypt messages. Specifically, the will look at modular arithmetic for both encryption and decryption, and frequency analysis to break codes.
- Students will know what are cryptography, cryptanalysis, and cryptology.
- Students will know what modular arithmetic is and how to us addition to encrypt messages.
- Students will know how to read basic Arduino code.
- Students will be able to write and decrypt messages using modular addition and frequency analysis.
- Students will be able to read basic Arduino code and use it to hack a locked Arduino.

**Acknowledgements:** These notes were written by Claire Merriman for use in the Summer 2016 Summer Illinois Math Camp Epsilon, along with Hannah Burson and Michelle Seat. Day 3 activities were developed in collaboration with the Champaign-Urbana Community FabLab, in particular, Colten Jackson, Tanya Josek, and Virginia McCreary.

Funding for the 2016 Summer Illinois Math Camp was provided by the Office of Public Engagement, University of Illinois for a Public Engagement Grant; Department of Mathematics, University of Illinois; Illinois Geometry Lab, University of Illinois; Association for Women in Mathematics, University of Illinois; Dolciani Mathematics Enrichment Grant, Mathematical Association of America; and the National Science Foundation, Grant Number DMS-1449269.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This material is based upon work supported by the National Science Foundation under Grant Number DMS-1449269.

## Contents

**1 Intro to Cryptology/Modular Arithmetic 2**
1.1 Lesson Notes . . . 3
1.2 Reflections: . . . 7

**2 Hack your Pokémon: locked boxes, brute force, counting and coding 8**
2.1 Lesson Notes . . . 9
2.2 Reflections . . . 13

**3 Codes and Counting 14**
3.1 Lesson Notes . . . 15
3.2 Reflections . . . 18

**4 Decryption practice and Skittle Key Cryptography 18**
4.1 Lesson Notes . . . 19
4.2 Reflections . . . 22

**5 Appendix 24**
5.1 Intro to Cryptology/Modular Arithmetic . . . 24
5.2 Hack your Pokémon: locked boxes, brute force, counting and coding . . . 34
5.3 Frequency analyisis . . . 35
5.4 Decryption practice and Skittle Key Cryptography . . . 39

# 1 Intro to Cryptology/Modular Arithmetic

**Central focus:** The goal of this lesson is to introduce and establish the history and modular arithmetic behind cryptology.

**Objectives:** Students will

- Be able to define cryptology in their own words, including its usefulness in the real world.
- Be able to do "clock math" in a chart and in congruency mod$n$ situations.
- Draw connection from the "clock math" model to the Caesar shifts model.

**Materials:**

- Tape measure
- Cipher wheels
- Modular arithmetic and encryption decryption practice worksheets
- Alphabet chart

## 1.1 Lesson Notes

### Agenda

| Time | Activity | Instructor | TA | Students |
|---|---|---|---|---|
| 10 min | Motivation/Hook, introduction | Ask questions | Record answers | Think of answers, take notes on definition |
| 15 min | Intro to clock math | Ask questions maintain focus | Lead explanation | Take notes, answer questions |
| 10 min | Theorem and proof | Explain proof | Maintain focus | Take notes, ask questions |
| 30 min | Break | Prep for after break | Supervise students | Relax, have fun |
| 10 min | Intro to mod chart | Maintain focus | Explain chart | Take notes |
| 30 min | Worksheet 1 | Guide students to answers | Guide students to answers | Work in groups |
| 30 min | Break | Prep for after break | Supervise students | Relax, have fun |
| 15 min | Intro to cryptology | Explain definitions, using wheels | Write definitions, demonstrate wheel | Take notes, ask questions |
| 20 min | Worksheet 2 | Guide students to answers | Guide students to answers | Work in groups |

### Lesson Plan

**Motivation/Hook:** Ask: What is Cryptology? “Does anyone have and guesses?”
**Definition.** *Cryptology is the study of sending an intelligible message converted into an unintelligible or secret form AND the means of analysis which allow you to get back the original message. i.e. making and breaking codes.*

Ask: Why is cryptology useful?

- Internet commerce (credit cards/banking information)
- Calling plays in sports games (wrist bands on each player in UI Softball)
- Ghostery application where it hinds your location while you are online.

Simply put it is useful in hiding information while sending information.

**Transition:** there is some useful math behind cryptology!

**Introduction to clock math:** You use this sort of math every day and don’t realize it.

**Example 1**

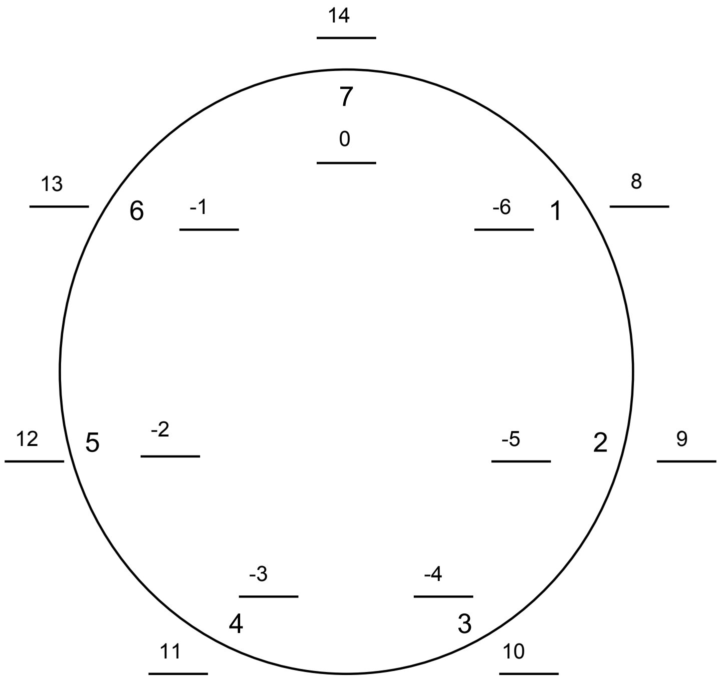
- If it is 1:30 now, what time is it in 2 hours?
- So 3:30, what time is it in 45 minutes?
- 4:15, what time is it in 9 hours? 1:15
- How long until its 2:30?

This is actually using modular arithmetic! We all did modular arithmetic just in our heads. We would call this math in “mod 12” for the hours and “mod 60” for the minutes.

For emphasis: Modular Arithmetic is as if you have taken the number line and wrapped it up, over itself. Like this tape measure. If I wrap this tape measure to be “mod 12” once you get to 13 you get back 1, once you get 14 you get 2, and so on.

—-*Draw the Modular Arithmetic Clock Demo”*—– (including negatives)

The modular clock demo shows that you can write numbers 1 to $n$ around a clock, then continue another circle $n+1$ to $2n$ outside and $-n-1$ to 0 on the inside. An example can be seen below. Can also write 0 to $n-1$ to be consistent with the standard representations instead of consistent with clocks.



This arithmetic can be represented in multiple ways. Algebraically we use congruence mod$n$.

**Definition.** *Two numbers are congruent modulo $n$ if $n \mid (x - y)$. Meaning $n$ divides $(x - y)$.*

**Example 2**

1. $5 \equiv 8 \mod 3$ Elaborate: $5 \equiv 11 \mod 3$ also $5 \equiv 14 \mod 3$ and so on.
   Show $3 \mid (8 - 5)$, that is, $8 - 5 = 3$, and $3/3 = 1$
2. $17 \equiv 5 \mod 3$ Elaborate if needed!! $3 \mid (17 - 5)$, $3 \mid 12$ check!
3. $9 \equiv 9 \mod 11$ This one seems a little tricky but is still true. $11 \mid (9 - 9)$, because you can divide 0 by any number.

This leads us to establish some important properties in modular arithmetic that can be expressed using this notation.

**Theorem and Proof**

**Theorem.** *Pick a whole number $n > 0$ and let $w, x, y, z$ be some integers. Then*

1. $x \equiv x \mod n$.
2. *If* $x \equiv y \mod n$, *then* $y \equiv x \mod n$.
3. *If* $x \equiv y \mod n$ *and* $y \equiv x \mod n$, *then* $x \equiv z \mod n$.
4. *If* $x \equiv y \mod n$, *then* $xz = yz \mod n$.
5. *If* $x \equiv y \mod n$, $w \equiv z \mod n$, *then* $x + w \equiv y + z \mod n$ *and* $xw \equiv yz \mod n$.

*Proof.*

1. Let $x$ be any integer. Then, $x - x = 0$ which is divisible by any non-zero integer. Thus, $x - x \mod n$.
2. Suppose $x \equiv y \mod n$, thus $n \mid (x - y)$. That is, there some number $k$ where $x - y = kn$. Since $y - x = -kn$, $n \mid (y - x)$. Therefore, $y \equiv x \mod n$.
3. Assume that $x \equiv y \mod n$ and $y \equiv z \mod n$. Then, there exist $k, m \in \mathbb{Z}$ with $x - y = kn$, $y - z = mn$. Combining these two equations, we get $x - y + y - z = kn + mn$, and thus $x - z = (k + m)n$, so $x \equiv z \mod n$.
4. Let $x \equiv y \mod n$. That is, there exists $k \in \mathbb{Z}$ such that $x - y = kn$, so $(kz)n = (x - y)z = xz - yz$. Thus, $xz \equiv yz \mod n$.
5. Let $x \equiv y \mod n$, $w \equiv z \mod n$, then there exist $k, m \in \mathbb{Z}$ such that $x - y = kn$, $w - z = mn$. Adding these equations we get $(x + w) - (y + z) = x - y + w - z = kn - mn = (k - m)n$. Applying (4)we get $xw - yw \mod n$ and $yw \equiv yz \mod n$, so by (3), $xw \equiv yw \mod n$.

□

This is largely to show that we can actually prove mathematical facts and use them later. This is where we inserted true false questions to check for understanding.

—-BREAK—-

**Using a chart:** Another way to display the same information is with a chart.

In the following chart, we will replace 5 with 0 to be closer to math conventions. This is because we can also think of mods as the remainder from division.

| $+_5$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 0 | 1 |
| 2 | 3 | 4 | 0 | 1 | 2 |
| 3 | 4 | 0 | 1 | 2 | 3 |
| 4 | 0 | 1 | 2 | 3 | 4 |
| 5 | 1 | 2 | 3 | 4 | 0 |

| $\times_5$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 | 4 |
| 2 | 0 | 2 | 4 | 1 | 3 |
| 3 | 0 | 3 | 1 | 4 | 2 |
| 4 | 0 | 4 | 3 | 2 | 1 |

This is a good place to pass out the modular arithmetic handout.

—-BREAK—-

**How “Clock Math” corresponds to Cryptology:** First let’s establish a few definitions.
**Definition.** *A code is a system that substitutes prescribed sets of characters for other sets of characters. To encode a message is to replace its elements by a different combination of characters or figures.*

**Example 3**

you may have agreed that each occurrence of the word “dog” will be replaced with “purple.”

**Definition.** *A cipher is a system where individual letters are replaced by other letters, either individually or in blocks.*

One famous example is the cipher Julius Caesar used to send letters to his army. He would take each letter and replace it with the one three places later in the alphabet. Here we say that the original message is the plaintext, the message that gets passed to the troops is the “ciphertext,” and the key is k=3, since you shift forward 3 letters in the alphabet. One way to do this is to use a cipher wheel, such as this one:

***Hand out cipher wheels****(and scissors and brads if necessary)

Do an example with an additive cipher along with the class. It is probably best to start with plaintext and help them convert to ciphertext.

**Ask:** Does this seem similar to anything we have already seen???

—-“The Modular Arithmetic Clock” or “Clock Math.” So clock math is like using an additive cipher. But we can use other ciphers with additive inverse, multiplicative and affine keys. (next available

day)
**Definition.** *additive inverse of a number x is a number y with x+y=0.*

In the integers, the additive inverse of x is -x. We can extend this idea to mod $n$ as well.

**Example 4**

$-2 \equiv 3 \mod 5$
$-1 \equiv 7 \mod 8$

Pass out encryption and decryption practice worksheet

**Closure:** Now we hope you have enjoyed learning a little bit about what it is to study cryptology. Tomorrow we will be working with computers to introduce the combinatorics, the counting behind cryptology. But for now, time for lunch!

### 1.2 Reflections:

Intro went well, got better definitions than expecting, great examples, pretty enthusiastic until proofs. They were able to give pretty good definitions for cryptography and cryptanalysis that we could put together for the definition of cryptology. This also gave a good way to introduce that we are doing both encryption and decryption.

The time activity worked well. Demonstrating with a tape measure seemed really cool, since I think of $\bmod 1$ as the $\mathbb{R}/\mathbb{Z}$ or $\mathbb{R}$ as the universal cover of $S^1$, but this did not seem to be super enlightening for the students. Drawing the clock definitely helped a lot of the students, so I introduced congruency classes.

Definitely need to introduce the fact that the representative of the congruency class of $x \mod n$ is the remainder of division of $x$ by $n$. None of them really seemed to use this definition, but it does justify why we use 0 instead of $n$. It would be beneficial to have the three methods/descriptions all in a row for the same example thus giving them multiple representations.

Did not do proofs the way they were written, since that is for an audience that is learning about proofs, and instead made sure they can follow the algebra, since not all have had algebra 1. I did explain factor/divides and that mods are the remainder from division before the proofs. . It is important to use the same wording used in the definition in the proof and decide that precise wording before the lesson. I also did not do number 5, but should probably add in $x + z$, $y + z$ before the multiplication. It might also help to introduce the idea of congruence classes with the clocks and reiterate from the beginning that there are infinitely numbers in a class, but it?s normally easiest to work with the smallest.

We added in some true/false examples after proofs to get them talking, working together and to check for understanding. We also drew mod 5 addition and multiplication charts and explained how to compute them and read them.

Before passing out second worksheet, do an example of how to encrypt with a given key and how to read the cipher wheels. It's probably good to walk through an example where you have a word, then numbers, then add the key, then back to letters, to emphasize the math. There was a lot of confusion about how to read the directions and wheels. Also, start with an example that has real words, since that confused a lot of them.

This day was also reflected by students as being very "worksheet heavy" and potentially tedious. There may be a way to make one or two of the worksheets into a group activity instead?not so that the work gets split up but that student start the week working together and don?t feel like they are doing just busy work.

## 2 Hack your Pokémon: locked boxes, brute force, counting and coding

**Central focus:**

**Objectives:** Students will

- Understand how the number of possible passwords grows with the number of allowed characters and password length
- See how brute force attacks can take a long time, even aided by computers
- Get a brief introduction to coding
- Understand how to count the number of possible passwords of a given length

**Materials:**

- Cryptography boxes from the FabLab
- Codes from the code folder/appendix
- 2 Arduinos per cryptography box
- 4 jumper cables per cryptography box
- Battery and connector for first Arduino, cable to connect second Arduino to computer
- Computer for each box
- Arduino program installed on each computer (https://www.arduino.cc/en/Main/Software)
- Google Presentation (tinyurl.com/hackpokemon)( pdf)
- Physical keys from FabLab or follow their guide to create them (goo.gl/fJk2Au)(pdf)
- Tokens/prizes

### 2.1 Lesson Notes

**Agenda**

| Time | Activity | Instructor | TA | Students |
|---|---|---|---|---|
| 5 min | Motivation/hook | Ask questions | Maintain focus | Answer questions |
| 5 min | Box activity 1 | Explain activity, help students think of way to systematically test combinations | Pass out boxes, help students think of way to systematically test combinations | Test possible combinations |
| 15 min | Intro to counting | Draw counting tree explain solutions | Maintain focus | Take notes, ask questions |
| 15 min | Passwords | Lead discussion | Contribute info | Contribute info |
| 20 min | Physical keys | Distribute keys explain activity | Help find correct key | Find correct key, unlock box, collect prize |
| 30 min | Break | Prep for after break | Supervise students | Relax, have fun |
| 10 min | Password breaking and ethics | Lead discussion | Contribute info | Contribute info |
| 35 min | Arduino Blink | Walk through loading blink, reading code | Maintain focus, answer questions, troubleshoot | Load blink, read and edit code |
| 20 min | Brute force password breaking | Intro to brute force, help load code | Answer questions, troubleshoot | Load code, ask questions, record times |
| 10 min | Wrap up | Summarize lesson | Maintain focus | Contribute to wrap up |

**Set Up**

Make sure all computers will work with Arduino.

Before the students get there, the boxes need to be set up.

1. Plug in Servo: red = 5V, brown = GND, Orange = A0 (analog zero), plug in Arduino to computer. In the arduino program, Select Arduino Uno/Genuino BOARD, and the right PORT
2. Open, then load "Servo Setup - sets to 90 degrees pin A0 1.0" code. (this moves the servo to 90 degrees)
3. When the servo stopped moving, attach the horn/arm/white thingy to the "open" position (along the blue servo box, parallel to the ground, not up or down)
   Now servo horn is in the right location!
4. Place token in box
5. Load crypto "Lock 3bit and 8bit" to lock box

A jumper cable from A5 to GND will put the box in 3 bit/toggle switch mode, removing this cable and resetting the Arduino with put it in “challenge mode” for the physical keys.

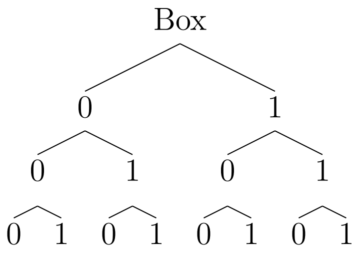
**Lesson Plan**

**Motivation/hook:** Why would you want a lock? What about a password? Can you think of locks that use passwords, mechanical or digital?

**Box activity 1:** Instructions to students:
We are going to give you a locked box. There is a little lever sticking out of the top that prevents it from sliding. Do not break anything. DO NOT TOUCH THE LEVER.
You will need to attach the battery to the box before it will work.
There are 3 switches on the side, you have to have each switch in the correct position to make the lever move and unlock the box. Once you get the right code, write it on the board and your name next to it. If your code is already on the board, just write your name next to it.

Have each student go to a computer with an Arduino, attach the battery, and try combinations. They might want to keep track of their trials on paper.

**Introduction to counting:** How many options were there for passwords? We can count using a tree.



By counting the number of options in the bottom row, we get 8 options.

Students seemed to have seen this before, so we did not spend much time on it, but in general, you multiply together the number of options you have for each spot. So if the first spot have 5 options and the second spot has 4 options, the two together have $5 * 4$ options.

For each position, there were 2 options, so there were $2 * 2 * 2 = 8$ total passwords.
What if we had 4 switches? 5 switches?
4 switches is 16 options, 5 switches is 32 options
If you use digits instead of numbers, each position has 10 options. An iPhone password is 4 digits, how many options are there?
10,000
What if we also allow letters? How many ways can you have a 3 letter or number password, no capitalization?
$36 * 36 * 36 = 46,656$

**Transition:**The number of options grows really quickly if you allow longer passwords, more characters. Numbers are big, computers are fast, but it is faster if you have a list of likely possibilities to check first such as common passwords and patterns.

**Passwords:** There are a lot of ways to reduce the number of passwords you check.
https://wpengine.com/unmasked/ has a lot of information about common passwords and password patterns. This is also included in the appendix, although new analysis comes out periodically and can be found on popular websites, warning people their passwords are not safe.

Common passwords:

1. qwerty
2. 123456
3. qwertyuiop
4. 123456789
5. password
6. 12345678
7. 12345
8. 111111
9. 1qaz2wsx
10. qwe123

**Physical Keys:** Have students remove the jumper cable that runs from A5 to GND and press reset. This should relock the box.

There are four different physical keys and the code randomizes which one unlocks the box. Students need to find which of the physical keys unlocks their box. Divide the students into groups and give each group a copy of each key. When the student gets the box unlocked, they can remove the token and claim a prize.

—-BREAK—-

**Password breaking and ethics:** What are some reasons that you might want to break a lock?

- Lose your key
- Forgot your password
- Found a phone and need to find its owner
- Trying to help someone who had some sort of accident and need to get contact information off their phone

**Arduino Blink:** Have students open Arduino on their computers and open the blink program. The presentation works through how the code works on slides 16-20 in the attached version, but most of the information is included here. Have the students plug in their Arduinos and upload blink, so that they can see it work, before reading through the code.

1. Header
    - Things that are always true (in the program)
    - Header can contain three types of statement.

    (a) Comments
    - These are in gray. They are just for the humans! They help the person reading the code understand it. Write lots of comments in your code, future you will thank past you! They can be a block of text between a /* and a */ or a line starting with a // Both are shown below.

    (b) Declarations
    - This is where we set up Variables we will want to use later in the program. A variable can be thought of as a tiny file that has a type, a name, and something stored in it. To create a new variable(or ?file?) you list the type, the name and store a value in it. For example: int led=13;
    - This statement creates a new integer Variable (or 'file') that contains the number 13. Now we can write 'led' somewhere in the code and know that when the arduino sees it it will go retrieve the value 13. And just like a normal file it will remain a 13 unless we overwrite it with another value.

    (c) Include Statements
    - We don't need one of these yet but it's a way to include code from someone else. And an include statement looks something like this: #include <FileToInclude.h>

2. Setup
    - The Setup just runs once at the beginning and gives us a chance to set things up before the never ending loop. It's really well named.
    - We only have one line of code inside our setup: pinMode(led, OUTPUT);
    - Lets figure out what it does:

        (a) pinMode is a function that tells the Arduino how we want to use one of its output pins.

        i. We programing folks will google functions we don't know. :) Try: http://arduino.cc/en/Refere and look up "pinMode()"

        ii. or just google: "pinMode arduino." The first result is usually at arduino.cc

        (b) So in this example we know that led currently has the value 13. So this code is going to set pin 13 to be an output. Making a pin an output means we can apply a voltage, or not, to anything outside hooked up to the pin.

   (c) Since the Arduino has an LED on the board that is wired up to pin 13 this means we will now be setup to be able to apply a voltage to that LED ,or not apply a voltage, and thus turn that led on or off.

3. The Loop

   (a) The loop runs forever! The Arduino will run each line of code starting from the top, one at a time. Whenever your Arduino reached the end of the loop code it will jump back to the start.

      i. The first thing we see is the digitalWrite function. Which we know nothing about... so lets look it up! Back to http://arduino.cc/en/Reference/

      ii. digitalWrite Writes a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode(), its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH, 0V (ground) for LOW.

      iii. Next let's look up delay.

      iv. delay Pauses the program for the amount of time (in milliseconds) specified as parameter. (There are 1000 milliseconds in a second.)

   (b) So remember we already know what the program does, because we ran it! It blinks an LED! So let's read through the loop code and see if we understand why?

Students can then hack the code, changing the delay time to change the how fast it blinks. This gives them a good idea of how to ready and edit code, instead of just uploading existing code.

**Brute Force password breaking:** When we guessed the password with the switches, we were using brute force. That is, we just guessed each password until it worked. For longer passwords, we can use a computer to do that.

The Brute Force 1.1 code creates random 8 bit, 16 bit, and 32 bit passwords, then guesses the code and prints out how long it took to guess. Have students load this code, then write on the board how long it took their computer to guess the 8 bit and 16 bit code. It is likely no one will have the 32 bit code done. Use this to reiterate the exponential growth.

### 2.2 Reflections

We largely followed the FabLab's lead on this project. The night before, we put a box at each computer, then had the campers sit at the tables for the intro section. This definitely helped with focus, and made the toggle switch activity really fast. Campers were able to very quickly come up with examples of physical and digital locks, as well as good examples of security and good reasons for breaking a codes or breaking past a security.

The tree was very useful for counting, definitely should have made them all draw it out for the examples. Most had seen it before, but it certainly helped those who hadn't.

The actual cracking of the codes for the boxes went pretty well, it would be good to add a slide with the directions, as well as a pre-drawn grid for the codes that work with their names listed, since that got pretty chaotic. It should have a bunch of rows so as to not give away the number of solutions.

The physical key activity was not written up. Each Arduino and box was set up to run off of a battery, with a jumper cable that could be removed to change the Arduino setting. The three cables that ran from the servo motor to the Arduino were one color (which changed from box to box) and the jumper cable was blue. When campers removed the jumper cable and pressed reset, the Arduino stopped running the code for the toggle switches, and started running the code that works with physical keys. There were 3 copies of each of the 4 physical keys. Students were confused about this, since there were only 2 key patterns. We should tie ribbons or something to make them distinctive, as well as create a slide explaining the activity. Students were to find the key that unlocked their box, then they could remove the token from their box and get a prize. May also be good to get token and then the prize at the end.

We moved the discussion of ethics and hacking to before the coding. The students really got into this part of the lesson.

It would be good to have an extra activity prepared for the students who already know some coding, since they got bored easily. Also, some kids got frustrated really quickly, so it is good to have a lot of people who can answer coding questions. If the activity is to be run outside of FabLab, make sure to check that the Arduino works with each of the computers, since a few had issues not supplying enough power to the USB port, and a few others would not run without admin privileges.

## 3 Codes and Counting

**Central focus:** The goal of this lesson is to give students practice encrypting and decrypting messages using various cryptanalysis tools.

**Objectives:** Students will

- Learn how to use frequency analysis to decrease the number of possible encryption schemes that they need to check
- Practice encrypting a message using modular arithmetic and the aid of a chart or wheel
- Practice decrypting a message where they do not know the encryption scheme

**Materials:**

- Letter and digraph frequency chart
- Mod 26 multiplication chart
- Alphabet chart from day 1
- Cipher wheels from day 1
- Frequency analysis practice worksheet

- Cryptanalysis presentation

## 3.1 Lesson Notes

### Agenda

| Time | Activity | Instructor | TA | Students |
|---|---|---|---|---|
| 5 min | Motivation/Hook | Introduce topic | Maintain focus | Listen |
| 25 min | Review of modular arithemtic | Brief review, give review problems | Maintain focus, answer questions | Take notes, answer questions |
| 15 min | Review of counting | Brief review, give review problems | Maintain focus, answer questions | Take notes, answer questions |
| 30 min | Break | Prep for after break | Supervise students | Relax, have fun |
| 10 min | Introduction to frequency analysis | Introduce topic | Maintain focus | Listen |
| 30 min | Frequency analysis practice worksheet | Guide students to answers | Guide students to answers | Work in groups |
| 30 min | Break | Prep for after break | Supervise students | Relax, have fun |
| 5 min | Encryption Review Brief | Review, give review problems | Maintain focus, answer questions | Take notes, answer questions |
| 30 min | Encryption practice | Introduce activity, check messages | Check messages, collect messages | Come up with and encrypt short message |

### Lesson Plan

**Motivation/Hook:** We have looked at the math behind encryption, now we will see the math behind decryption and get practice making and breaking codes.

### Lesson Procedure:

**Review of modular arithemtic:** Begin with a review of modular arithmetic.
**Definition.** *Two numbers $x$ and $y$ are congruent* mod$n$ *if you can divide $x - y$ by $n$ and get an integer.*

This is the same as numbers having the same remainder when divided by $n$. Students can check this with division or using the clock, which tells you if two number are the same congruence class.

Have students work the example problems on their whiteboards in groups. Make sure that they are working together and comfortable explaining the solution. This is a good place to emphasize there are multiple ways to solve the problems and multiple correct answers.

Have students write the true statements and correct the false statements on their whiteboards in groups.

- If $x \equiv 7 \mod 14$, then $7 \equiv x \mod 14$
- If $4 \equiv 13 \mod 9$, then $8 \equiv 20 \mod 9$
- If $4 \equiv 16 \mod n$, then $4 \equiv n \mod 16$

The first is true. For the second, they can correct it several ways, including $8 \equiv 26 \mod 9$ and $8 \equiv 17 \mod 9$. There are many ways to fix the third, including changing $n$ to 12, changing the second statement to $16 \equiv 4 \mod n$, $4 \equiv 4 \mod n$, $16 \equiv 16 \mod n$, or changing the first statement to $n \equiv 4 \mod 9$, ect.

**Review of counting:** Have students answer the following problems on their whiteboards in groups. Make sure that they can explain their answer.

- Say you have three pairs of shorts- one yellow, one blue, and one white, and four shirts-one red, one green, one black, and one blue. How many outfits can you make?
- Each card in the Set deck has 4 attributes-color, shading, number and shape. There are 3 options for each attribute. How many cards are there?

This first is $3 * 4 = 12$ and the second is $3^4 = 81$. Make sure students can explain how they got the answer, some may have drawn out the chart or used another method.

—-BREAK—-

**Introduction to frequency analysis:** Since there are 26 ways to encrypt the letter A, 25 ways to encrypt the letter B (one has already been used), 24 ways to encrypt C,... there are 26! ways to scramble the English alphabet. This is more than 403 septillion, a number so big, I had to have Wolfram Alpha tell me what to call it! If you check 1,000,000,000 options a second, it would take 12.79 trillion years to check them all. We need a better method.

Some letters are used more often than others. In a standard block of English text, approximately 12% of the letters are E. We can count how many times each letter occurs, to help us guess the encryption. Analysis for "It does not do to dwell on dreams and forget to live."

| A | D | E | F | G | I | L | M | N | O | R | S | T | V | W |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 5 | 5 | 1 | 1 | 2 | 3 | 1 | 3 | 7 | 2 | 2 | 5 | 1 | 1 |

Pass out the frequency analysis practice worksheet. It is good to talk through how to divide up the work, and that the analysis does not necessarily line up with the expected value. A few students started to guess the passages, so make sure they can decrypt the entire thing and check it's the right length.

Before break, point out that they can use spaces to guess at an encryption scheme, for example, the only one letter words are "a" and "I." One way to get around this is to ignore where the spaces go and put them every 3 or 5 or n letters, or to use a different mod and encrypt the spaces.

—-BREAK—-

**Encryption Review:** Review using additive ciphers:

You can convert between letters and numbers, and use a key to get a secret message
I wanna be the very best
8 22-0-13-13-0 1-4 19-7-4 21-2-17-24 1-4-8-19
add 5 to each number, mod26
13 1-5-18-18-5 6-9 24-12-9 0-7-22-3 6-9-13-0
N bfssf gj ymj ajwd gjxy

Also review how to use the cipher wheels.

**Encryption practice** Students will be broken into 5 groups, each group comes up with a message to encrypt using an additive cipher. The plaintext message, key, and ciphertext are given to the instructors to type up for the next day. Alternatively, the encrypted message could be passed to another group, with instructions for running this activity multiple times in the same day at the end of the presentation.

Transition: Now we are going to do an encryption and decryption activity

Ask: What methods do you have to encrypt a message?

- Cipher wheel
- Mod 26 multiplication chart

Ask: What tools do you have to decrypt a message?

- Frequency analysis
- Knowing the encryption key
- As long as the encryption scheme does not change, each letter is always replaced with the same letter

Transition: Now let's practice using these tools

Activity: Students will be broken into 5 groups, each group comes up with a message to encrypt using a multiplicative or additive cipher. This encrypted message is given to another group to decrypt.

Rules for the message:

- You have 10 minutes to come up with AND WRITE a message
- Real, English words (also, must make sense)
- No cursing
- Can't be mean
- Must allow us to read it before encrypting
- No more than 3 sentences

Rules for encryption:

- You have 10 minutes
- Must use additive cipher
- Give the code with spaces every 3 letters

Closure: We have had some fun practice encrypting and decrypting codes. Tomorrow we will talk about ways to encrypt messages with people you don't know!

### 3.2 Reflections

We basically followed the Beamer as written, since we were editing it as we went.

The review of mods went well, students got the challenge problems pretty quickly. It was a good place to remind them that there are multiple ways of doing things, before moving into counting. Most of them did the counting problems by multiplication, but a few did use the tree, or a modified tree. The phrasing of the Set Deck question confused some, maybe include pictures.

Definitely explain frequency analysis before giving an example. It would probably be best to talk them through how to use the chart before any example, and maybe have a slide or instructions on the sheet. It's also good to emphasize that the expected value is not the same as the actual value, especially in short blocks of text or if the writer intentionally avoided common letters.

Students definitely focused a lot on counting the frequency. Some were frustrated that it took so long. This is a good place to talk about team work, probably before passing out the worksheet. Some students were starting to guess the encryption, which was great, but some were also just checking that their key made sense. We ended up asking each table if they had a representative who was willing to read each paragraph to the class, which helped them finish the activity.

We skipped the encryption review and saved it for the next day, since there was not time to talk about multiplication. We ended up having them write 2 sentences, have us approve them, and then turn in a piece of paper with the plaintext message, key, and encrypted message. Even with this check, it took some work to get them to write appropriate, coherent messages. This basically filled up the time, so the decryption was moved to the next day. This in the end made more work for the instructors however that helped to check the decryptions keys and check for typos so that may be the preferred way of running this activity in the future.

## 4 Decryption practice and Skittle Key Cryptography

**Central focus:** Students will apply frequency analysis to decrypt messages from the day before. The goal of the second part of the lesson is to introduce the Diffie-Hellman Key Exchange and challenge students to discover the result on as a team.

**Objectives:** Students will

- Work as a team to decrypt messages.

- Work as a team to solve a 'passing of a message' challenge.
- Learn the importance of safely sending encrypted messages.
- Debate and discuss methods of safely sending messages.

**Materials:**

- Decryption worksheet (Example)
- 12 cups with lids (2 per group)
- 12 bags of Skittles (2 per group)
- Hats or stickers with names Alice, Bob, and Eve (at least one per group-could have multiple Eves)
- Print out of script
- Print out of rules

### 4.1 Lesson Notes

**Agenda**

| Time | Activity | Instructor | TA | Students |
|---|---|---|---|---|
| 5 min | Review | Review how to use frequency chart | Maintain focus | Listen, ask questions |
| 45 min | Decryption activity | Explain worksheet, guide students to answers | Guide students to answers | Work in groups |
| 30 min | Break | Prep for after break | Supervise students | Relax, have fun |
| 15 min | Intro to Skittle key cryptography | Explain activity, skit | Skit | Pay attention |
| 30 min | Skittle key cryptography | Guide students through activity | Guide students through activity | Work in groups to find solution |
| 20 min | Diffie-Hellman Key Exchange | Brief review of modular arithmetic, explain the algorithm | Maintain focus | Listen, ask questions |

**Lesson Plan**

**Decryption Activity:** This lesson starts with decrypting the messages the day before. Before handing out the worksheet, review how to read the frequency analysis chart, as well as reminding students that the frequency is for really long passages. It is also possible some groups intentionally avoided common letters.

Remind students that they can divide up the work among the group. This is a good place to have a competition-we told the first table to get the entire worksheet correct could have candy at lunch.

**Skittle Key Cryptography:** This activity is based on Chocolate Key Cryptography by Dale J. Bachman, Ezra A. Brown, Chocolate Key and Anderson H. Norton [1], which is also available at https://www.math.vt.edu/people/brown/doc/ckc_math_teacher.pdf.

So we have learned a lot about cryptology, especially in the ways of encrypting and decrypting messages. But what we haven't discussed the way two people might develop a secret key that would be used to produce encryptions and decryptions. Now we are going to run through an activity that allows you to explore sending messages securely.

The scene: Alice= Claire, Bob= Hannah, Eve= Michelle. In this scenario, Alice and Bob want to communicate a message to one another however there is someone around eavesdropping (Eve).

Alice and Bob both start out with a "magical" machine (coffee cups w/lids) which allows them to encrypt and decrypt messages (skittles). The cups represent keys for encoding and decoding messages. We have to imagine that everyone in class has an encryption-decryption machine that used cups of skittles as keys. (It's like the cup is capable of inputting a message then encrypting it but also the other way around, getting a message and decrypting it.)

That is, if Alice wants to send a message to Bob, she picks a key (a cup of Skittles), puts the key in the slot, puts her message in the input chute, and cranks forward so that an encrypted message comes out. She then sends the message to Bob. Even when she sends the message through a public network (such as the Internet or in our case the multiple Eve's), the message will remain secure unless someone else has the same key for encrypting-decrypting.

When Bob receives the encrypted message, he will place his identical key (a cup with the same numbers and colors of Skittles as Alice has) in the slot of his encrypting-decrypting box. He will then put the encrypted message into the input chute and crank backward to get a decrypted message out.

The tricky part is for Alice and Bob to obtain identical cans to begin with, without anyone else obtaining copies of those cans or guessing their contents (knowledge of the contents would allow someone else to create an identical can thus defeating the purpose of encrypting the message.) In short, this is the main idea. We will emphasis what not to do with the skit.

Instructors will:

- Open the lid (Eve eats them)
- Say the code out loud, in a whisper. (Eve can hear whispers, she employs bats)
- Bob tries to peak
- Remember! You don't want Eve to know the key.
- Remember! Cups duplicate when passed so the people can "hold their own"!!
- Can't combine two duplicates (instead they are holding multiple different cups with keys but can't peak into them to combine them)

**Instructor** : Alice wants to send Bob a message, but they have not agreed on a key, and they are too far away to talk about it. How do they figure out a key? I am going to be Alice, __ is going to be Bob, and __ is going to be Eve
**Eve:** Alice and Bob can?t talk to each other, even at a whisper, because I hear everything with my pet bats. They can?t signal each other, because I have spies everywhere!
Alice: Bob and I do have magical Skittle powered cryptology machine.If we can just somehow get the same number of each color skittle in our cups, we can encrypt and decrypt our messages.
**Bob:** What if I just take the lid off and look at what you put in the cup?
**Eve:** Then I run over and eat all of your skittles?
**Alice:** What if I just peak?
**Eve:**: I can still get your skittles!
**Bob:** But wait, Eve also has a magical skittle powered cryptology machine! We have to make sure we end up with the same skittles, but that they are different than what Eve touched.

Rules of the Game:

- Alice and Bob insert as many Skittles of whatever colors they like into their respective cups.
- No one can open either cup, not even Alice or Bob.
- Any student holding a cup is able to duplicate it magically, including its contents.
- This means that each student who has touched the can pretends that he or she still holds an identical copy, even after passing along the original can (but still one cannot open the copy to see what is inside).
- The goal is for Alice and Bob–and no one else–to end up with cans with identical contents, but all their messages to each other must pass through the classroom so that every student can touch and copy them.

**Ask:** How can Alice and Bob achieve this goal?

Put students into groups of 3. Let them work through the problem, but answer questions, especially about the rules. The solution is to have Alice and Bob each put some collection of Skittles in their cup, remember how many of each color they used, and then switch cups. They then put the same collection in the new cup.

**Diffie-Hellman Key Exchange:** The Skittle activity models Diffie-Hellman Key Exchange, which is a type of public key cryptography. It allows people who do not know each other, such as credit card companies and consumers, to exchange parts of a key without anyone knowing the whole key. This scheme was first published in 1976, but in 1997, the British government declassified documents showing they had developed it in 1974. This is not the most commonly used method of key exchange, but one that we can easily model.

Here, if Alice wants to send Bob a message, they first need to pick a prime number $p$, the bigger the better, which acts like your magical Skittle machine, and a publicly shared base $g$, with some restrictions, which acts like the cup. In fact, they need to choose $g$ where they can write $g^k \equiv a \mod p$ for any $a \neq 0$.

Alice and Bob then each choose numbers between 1 and $p-1$, call these numbers $a$ and $b$.

Alice computes $g^a \equiv A \mod p$, and Bob computes $g^b \equiv B \mod p$. Alice then sends $A$ to Bob, and Bob sends $B$ to Alice. Now, Alice computes $B^a \equiv (g^a)^b \equiv g^{ab} \mod p$, and Bob computes $A^b \equiv (g^b)^a \equiv g^{ba} \mod p$, so they have the same final key.

**Example 5**

Alice and Bob agree to use a modulus $p = 23$ and base $g = 5$.

Alice chooses a secret integer $a = 6$, then sends Bob $A = g^a \mod p$, where $A \equiv 5^6 \equiv 8 \mod 23$.

Bob chooses a secret integer $b = 15$, then sends Alice $B = g^b \mod p$, where $B \equiv 15^6 \equiv 19 \mod 23$.

Alice computes $s = B^a \mod p$, where $s \equiv 19^6 \equiv 2 \mod 23$.

Bob computes $s = A^b \mod p$, where $s \equiv 8^1 5 \equiv 2 \mod 23$.

Alice and Bob now share a secret (the number 2).

Note that only $a, b$, and ($g^{ab} \equiv g^{ba} \mod p$) are kept secret. All the other values - $p, g, g^a \mod p$, and $g^b \mod p$- are public. Once Alice and Bob compute the shared secret they can use it as an encryption key, known only to them, for sending messages across the same open communications channel.

## 4.2 Reflections

We started by handing out the messages from the day before. They had a list of everyone's encrypted message including their own groups, meaning groups had to stay the same from the day before as to not give any one group an unfair advantage. Students were very into the activity and were able to use both frequency analysis and intelligent guessing to figure out the messages. At the end of the activity, it would be good to reiterate shorter messages are harder to decrypt, as are those that have frequencies dramatically different that what is expected. Also a good place to reinforce dividing up the work.

At the end of the lesson, run through converting message to numbers, adding key, translating back to letters. This gives a good intro to multiplication. If time, they can play with a multiplication chart, but we did not have time.

Skittle key cryptography went pretty well. It's probably best to rehearse the skit ahead of time. It would be better to have a handout of the rules for each table, since there are a lot. The students were confused at the beginning about the rules. Along with having a handout of the rules, it is a good idea to go around to each table quickly and ask the students to explain what their goal is. That makes sure everyone is on the same page. The hint that helped most people was that you can switch cups. Each group was pretty excited when they figured it out.

I was prepared with a brief explanation of Diffie-Helman, but they wanted an example. It would also be good to have them run through mod 5 or 7 and find a primitive root. (For mod 5, 2 and 3 work, for mod 7, 3 and 5 work)

# 5 Appendix

## 5.1 Intro to Cryptology/Modular Arithmetic

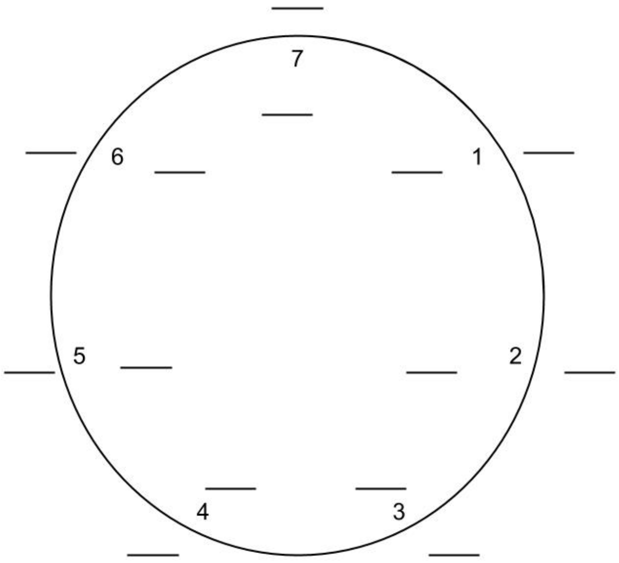
Day 1 materials.

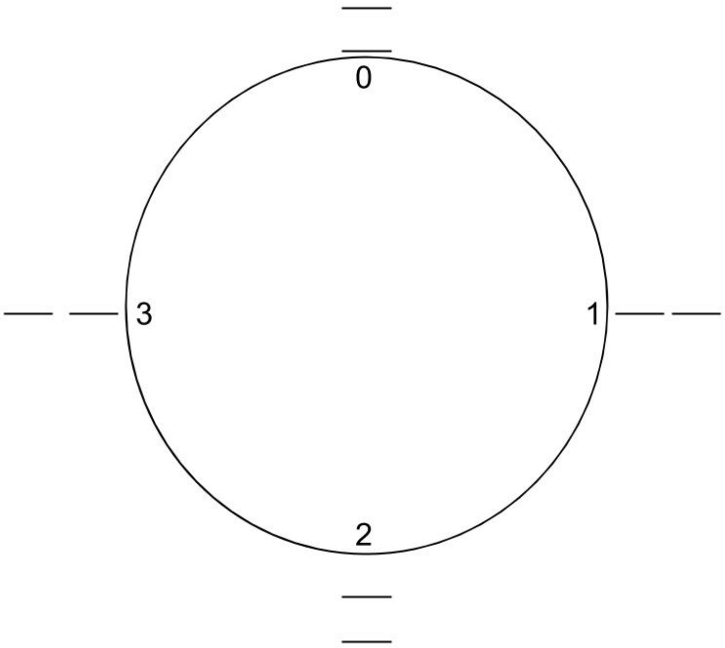
1. Modular arithmetic worksheet
2. Encryption and decryption practice worksheet
3. Decoder wheel with instructions, from http://dabblesandbabbles.com/printable-secret-decoder-wheel/
4. Black and white cipher wheel, larger
5. Alphabet chart

## SIM Camp Epsilon: Making and Breaking Codes

Modular Arithmetic

Complete these mod clocks, with numbers bigger than n outside the circle and negative numbers inside the circle.





We can also do arithmetic with modular arithmetic. Complete the following addition charts.

| $+_{12}$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 0 | 1 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 0 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 0 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 0 | 1 | 2 | 3 | 4 |
| 5 | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | |

| $+_7$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 0 |
| 2 | 2 | 3 | 4 | 5 | 6 | 0 | 1 |
| 3 | | | | | | | |
| 4 | | | | | | | |
| 5 | | | | | | | |
| 6 | | | | | | | |

Find some example numbers that work in the following problems:

1. $18 \equiv m \mod 4$, $m =$ _____ _____ _____
2. $15 \equiv r \mod 14$, $r =$ _____ _____ _____
3. $3 \equiv l \mod 2$, $l =$ _____ _____ _____
4. $19 \equiv t \mod 11$, $t =$ _____ _____ _____
5. $11 \equiv z \mod 4$, $z =$ _____ _____ _____
6. $2 + 6 \equiv p \mod 19$, $p =$ _____ _____ _____
7. $13 - 6 \equiv a \mod 23$, $a =$ _____ _____ _____
8. $5 + 7 \equiv x \mod 14$, $x =$ _____ _____ _____
9. $9 + h \equiv 19 \mod 14$, $h =$ _____ _____ _____
10. $19 + b \equiv 17 \mod 5$, $b =$ _____ _____ _____
11. $15 + n \equiv 6 \mod 7$, $n =$ _____ _____ _____

Addition isn't the only way we do arithmetic, try these multiplication tables!

| $\times_3$ | 0 | 1 | 2 |
|---|---|---|---|
| 0 | | | |
| 1 | | | |
| 2 | | | |

| $\times_5$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | | | | | |
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |

| $\times_{12}$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 2 | 0 | 2 | 4 | 6 | 8 | 10 | 0 | 2 | 4 | 6 | 8 | 10 |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |  |  |  |  |  |

## SIM Camp Epsilon: Making and Breaking Codes

### Encryption and Decryption Practice

1. Find the additive inverse of $15 \mod 26$.
2. Find the additive inverse of $6 \mod 13$.
3. Solve $5 + x \equiv 12 \mod 27$.

Encrypt the following messages:

1. Meet me at the usual place at eight o'clock, key=3
2. So long and thanks for all the fish, key=15
3. "we dance round in a ring and suppose,
   but the Secrets sit in the middle, and knows", key=6

Decrypt the following messages. Remember to find the additive inverse of the encryption key!

1. X detc pi iwt radht, key=11
2. N bfssf gj ymj ajwd gjxy. Qnpj st tsj jajw bfx. Yt hfyhm ymjr nx rd wjfq yjxy. Yt ywfns ymjr nx rd hfzxj. N bnqq ywfajq fhwtxx ymj qfsi, xjfwhmnsl kfw fsi bnij. Yjfhm ymjr yt zsijwxyfsi ymj utbjw ymfy'x nsxnij. Ny?x dtz fsi rj N pstb ny?x rd ijxynsd, tm, dtz'wj rd gjxy kwnjsi ns f btwqi bj rzxy ijkjsi. F mjfwy xt ywzj. Tzw htzwflj bnqq uzqq zx ymwtzlm. Dtz yjfhm rj fsi N'qq yjfhm dtz. Ltyyf hfyhm 'jr fqq. Utpjrts!, key=5

# Secret Decoder

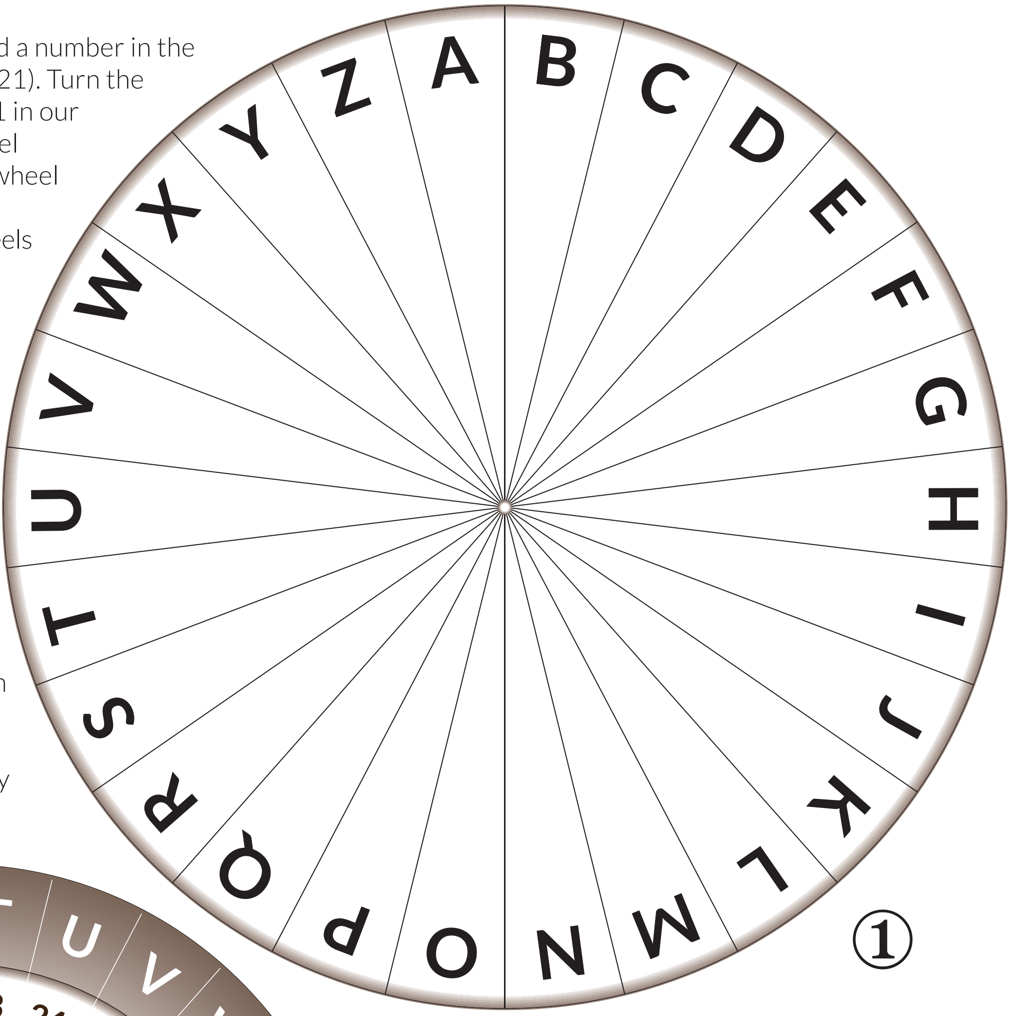
This is a super fun assignment for all the secret agents out there. Time to put your sleuthing skills to work.

**HOW TO MAKE THE DECODER WHEEL:**

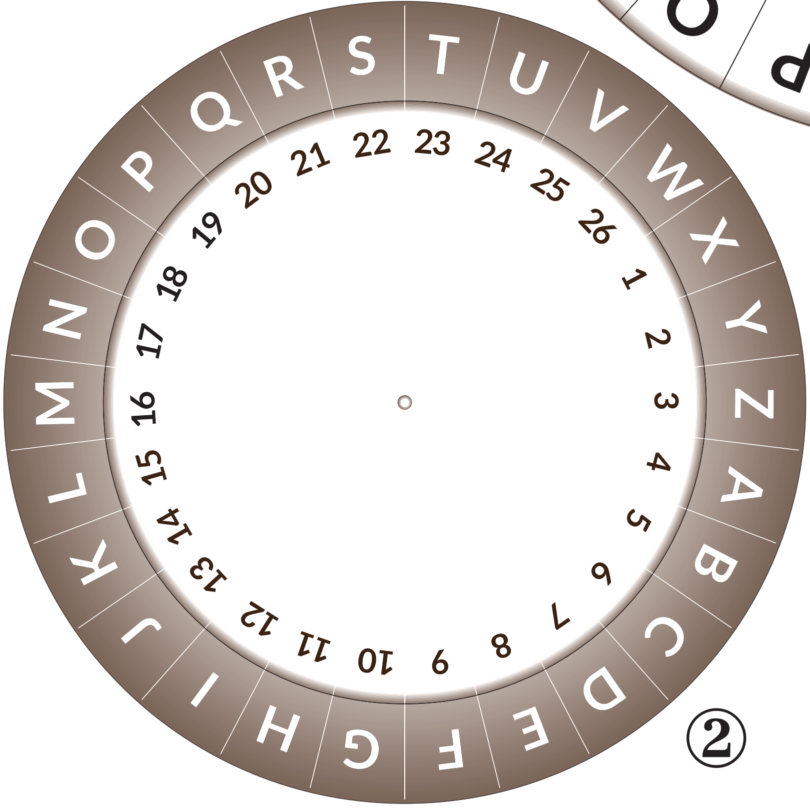
1. Cut out the circles below and stack in order from the number 1 wheel on the bottom, then number 2 wheel and finally number 3 wheel on the top.
2. Attach the 3 discs by carefully poking a round brad through the middle of all three wheels.

**HOW TO USE DECODER WHEEL:**

1. Pick a letter on the outer wheel and a number in the inner circle - this is your key (ex. M21). Turn the inner wheel so that the number (21 in our example) lines up with the out wheel letter (M in our example). On our wheel 21 also corresponds with R in the shaded section. Don't move the wheels now, keep them in place.
2. First, write down your message. No numbers (write them out), and no punctuations
3. For each letter of your message, find that character on the outer wheel, and write down the letter that is exactly beneath it on the inner wheel until your message is complete.
4. To read the encrypted message, get the key from the message sender and align the wheel. For each letter of your message, find that character on the inner wheel, and write down the letter that is exactly above it on the outer wheel.



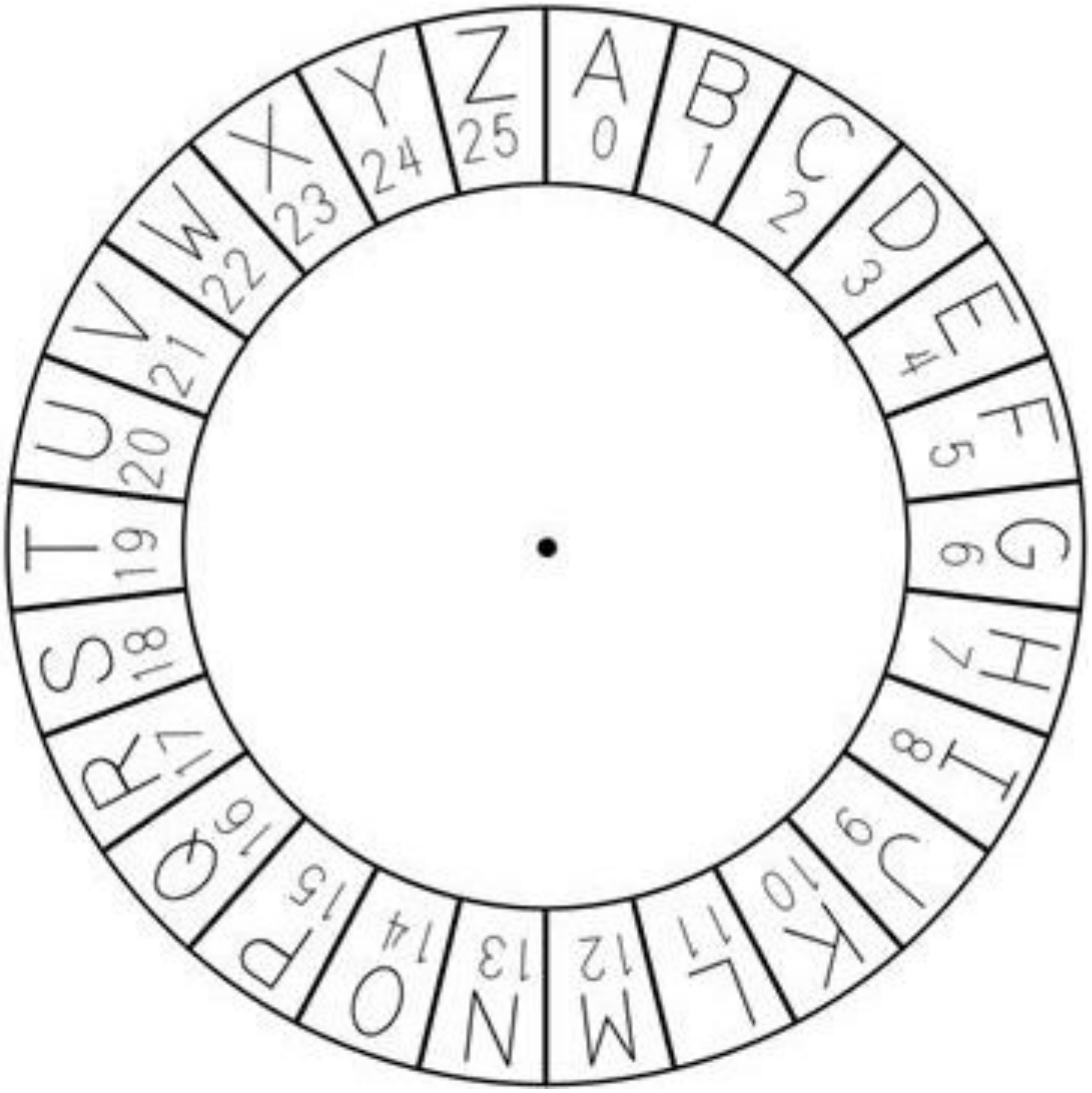
①

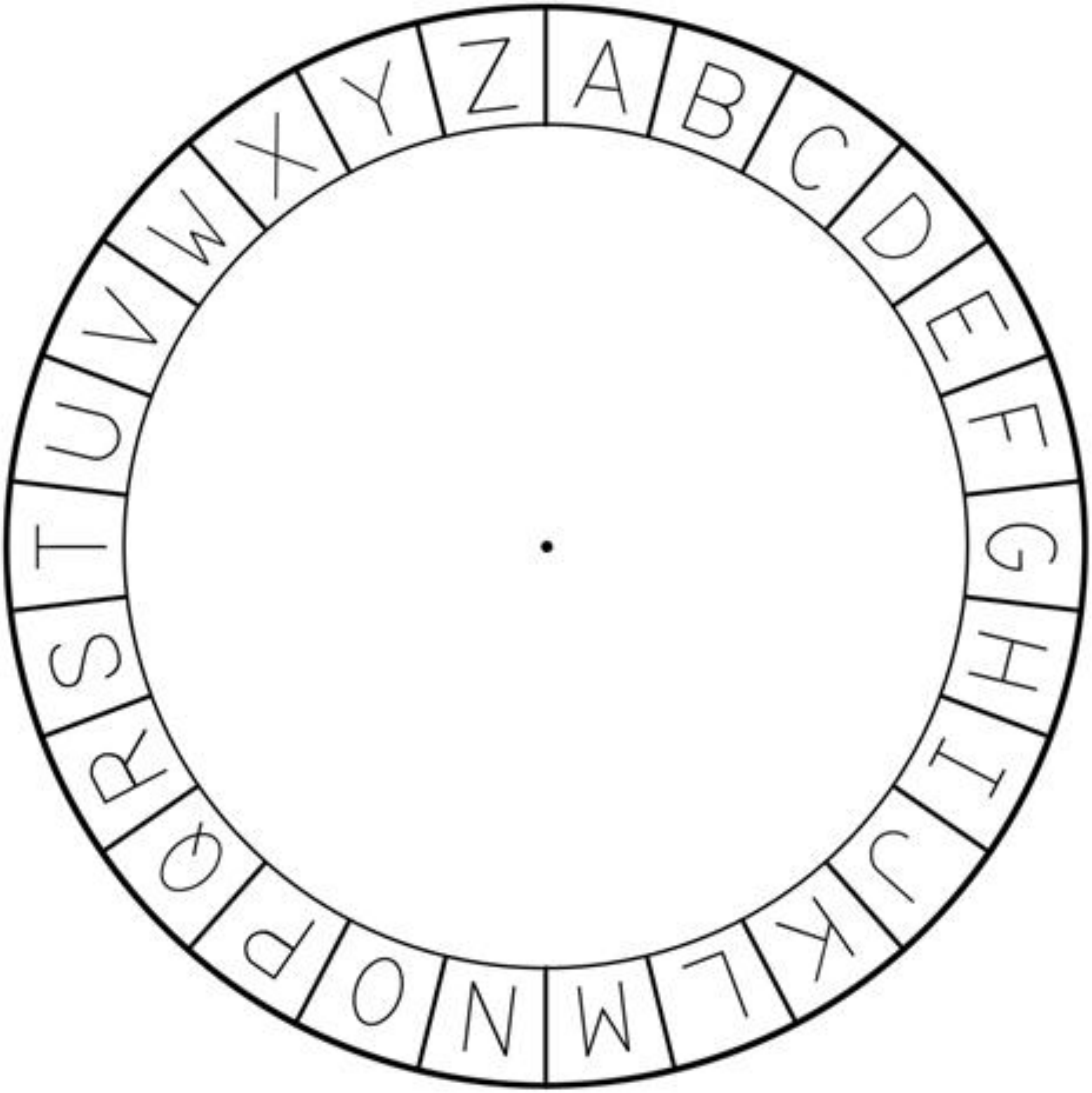


②

SECRET
DECODER

③


A
0
B
1
C
2
D
3
E
4
F
5
G
6
H
7
I
8
J
9
K
10
L
11
M
12
N
13
O
14
P
15
Q
16
R
17
S
18
T
19
U
20
V
21
W
22
X
23
Y
24
Z
25


Z
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y

| Letter | Number |
|---|---|
| a | 00 |
| b | 01 |
| c | 02 |
| d | 03 |
| e | 04 |
| f | 05 |
| g | 06 |
| h | 07 |
| i | 08 |
| j | 09 |
| k | 10 |
| l | 11 |
| m | 12 |
| n | 13 |
| o | 14 |
| p | 15 |
| q | 16 |
| r | 17 |
| s | 18 |
| t | 19 |
| u | 20 |
| v | 21 |
| w | 22 |
| x | 23 |
| y | 24 |
| z | 25 |

### 5.2 Hack your Pokémon: locked boxes, brute force, counting and coding

Links to material, available in presentations folder:

1. Set up codes, also available at https://drive.google.com/drive/folders/0B2ZfYsm_V__faHA2RThSek1WOUk
2. pdf of Google Presentation, also available at http://tinyurl.com/hackpokemon
3. pdf of instructions to make physical keys, also available at goo.gl/fJk2Au

### 5.3 Frequency analyisis

The materials in this section in order are:

1. Letter and digraph frequency chart
2. Mod 26 multiplication chart
3. Frequency analysis practice worksheet
4. Cryptanalysis presentation (link)

| Letter | Percent of Text |
|---|---|
| E | 12.51 |
| T | 9.25 |
| A | 8.04 |
| O | 7.6 |
| I | 7.26 |
| N | 7.09 |
| S | 6.54 |
| R | 6.12 |
| H | 5.49 |
| L | 4.14 |
| D | 3.99 |
| C | 3.06 |
| U | 2.71 |
| M | 2.53 |
| F | 2.3 |
| P | 2 |
| G | 1.96 |
| W | 1.92 |
| Y | 1.73 |
| B | 1.54 |
| V | 0.99 |
| K | 0.67 |
| X | 0.19 |
| J | 0.16 |
| Q | 0.11 |
| Z | 0.09 |

## The most common first letter in a word in order of frequency

T, O, A, W, B, C, D, S, F, M, R, H, I, Y, E, G, L, N, O, U, J, K

## The most common digraphs on order of frequency

TH, HE, AN, IN, ER, ON, RE, ED, ND, HA, AT, EN, ES, OF, NT, EA, TI, TO, IO, LE, IS, OU, AR, AS, DE, RT, VE

## The most common two-letter words in order of frequency

of, to, in, it, is, be, as, at, so, we, he, by, or, on, do, if, me, my, up, an, go, no, us, am

| $x_{26}$ | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 2 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 | 22 | 24 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 | 22 | 24 |
| 3 | 0 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 1 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 |
| 4 | 0 | 4 | 8 | 12 | 16 | 20 | 24 | 2 | 6 | 10 | 14 | 18 | 22 | 0 | 4 | 8 | 12 | 16 | 20 | 24 | 2 | 6 | 10 | 14 | 18 | 22 |
| 5 | 0 | 5 | 10 | 15 | 20 | 25 | 4 | 9 | 14 | 19 | 24 | 3 | 8 | 13 | 18 | 23 | 2 | 7 | 12 | 17 | 22 | 1 | 6 | 11 | 16 | 21 |
| 6 | 0 | 6 | 12 | 18 | 24 | 4 | 10 | 16 | 22 | 2 | 8 | 14 | 20 | 0 | 6 | 12 | 18 | 24 | 4 | 10 | 16 | 22 | 2 | 8 | 14 | 20 |
| 7 | 0 | 7 | 14 | 21 | 2 | 9 | 16 | 23 | 4 | 11 | 18 | 25 | 6 | 13 | 20 | 1 | 8 | 15 | 22 | 3 | 10 | 17 | 24 | 5 | 12 | 19 |
| 8 | 0 | 8 | 16 | 24 | 6 | 14 | 22 | 4 | 12 | 20 | 2 | 10 | 18 | 0 | 8 | 16 | 24 | 6 | 14 | 22 | 4 | 12 | 20 | 2 | 10 | 18 |
| 9 | 0 | 9 | 18 | 1 | 10 | 19 | 2 | 11 | 20 | 3 | 12 | 21 | 4 | 13 | 22 | 5 | 14 | 23 | 6 | 15 | 24 | 7 | 16 | 25 | 8 | 17 |
| 10 | 0 | 10 | 20 | 4 | 14 | 24 | 8 | 18 | 2 | 12 | 22 | 6 | 16 | 0 | 10 | 20 | 4 | 14 | 24 | 8 | 18 | 2 | 12 | 22 | 6 | 16 |
| 11 | 0 | 11 | 22 | 7 | 18 | 3 | 14 | 25 | 10 | 21 | 6 | 17 | 2 | 13 | 24 | 9 | 20 | 5 | 16 | 1 | 12 | 23 | 8 | 19 | 4 | 15 |
| 12 | 0 | 12 | 24 | 10 | 22 | 8 | 20 | 6 | 18 | 4 | 16 | 2 | 14 | 0 | 12 | 24 | 10 | 22 | 8 | 20 | 6 | 18 | 4 | 16 | 2 | 14 |
| 13 | 0 | 13 | 0 | 13 | 0 | 13 | 0 | 13 | 0 | 13 | 0 | 13 | 0 | 13 | 0 | 13 | 0 | 13 | 0 | 13 | 0 | 13 | 0 | 13 | 0 | 13 |
| 14 | 0 | 14 | 2 | 16 | 4 | 18 | 6 | 20 | 8 | 22 | 10 | 24 | 12 | 0 | 14 | 2 | 16 | 4 | 18 | 6 | 20 | 8 | 22 | 10 | 24 | 12 |
| 15 | 0 | 15 | 4 | 19 | 8 | 23 | 12 | 1 | 16 | 5 | 20 | 9 | 24 | 13 | 2 | 17 | 6 | 21 | 10 | 25 | 14 | 3 | 18 | 7 | 22 | 11 |
| 16 | 0 | 16 | 6 | 22 | 12 | 2 | 18 | 8 | 24 | 14 | 4 | 20 | 10 | 0 | 16 | 6 | 22 | 12 | 2 | 18 | 8 | 24 | 14 | 4 | 20 | 10 |
| 17 | 0 | 17 | 8 | 25 | 16 | 7 | 24 | 15 | 6 | 23 | 14 | 5 | 22 | 13 | 4 | 21 | 12 | 3 | 20 | 11 | 2 | 19 | 10 | 1 | 18 | 9 |
| 18 | 0 | 18 | 10 | 2 | 20 | 12 | 4 | 22 | 14 | 6 | 24 | 16 | 8 | 0 | 18 | 10 | 2 | 20 | 12 | 4 | 22 | 14 | 6 | 24 | 16 | 8 |
| 19 | 0 | 19 | 12 | 5 | 24 | 17 | 10 | 3 | 22 | 15 | 8 | 1 | 20 | 13 | 6 | 25 | 18 | 11 | 4 | 23 | 16 | 9 | 2 | 21 | 14 | 7 |
| 20 | 0 | 20 | 14 | 8 | 2 | 22 | 16 | 10 | 4 | 24 | 18 | 12 | 6 | 0 | 20 | 14 | 8 | 2 | 22 | 16 | 10 | 4 | 24 | 18 | 12 | 6 |
| 21 | 0 | 21 | 16 | 11 | 6 | 1 | 22 | 17 | 12 | 7 | 2 | 23 | 18 | 13 | 8 | 3 | 24 | 19 | 14 | 9 | 4 | 25 | 20 | 15 | 10 | 5 |
| 22 | 0 | 22 | 18 | 14 | 10 | 6 | 2 | 24 | 20 | 16 | 12 | 8 | 4 | 0 | 22 | 18 | 14 | 10 | 6 | 2 | 24 | 20 | 16 | 12 | 8 | 4 |
| 23 | 0 | 23 | 20 | 17 | 14 | 11 | 8 | 5 | 2 | 25 | 22 | 19 | 16 | 13 | 10 | 7 | 4 | 1 | 24 | 21 | 18 | 15 | 12 | 9 | 6 | 3 |
| 24 | 0 | 24 | 22 | 20 | 18 | 16 | 14 | 12 | 10 | 8 | 6 | 4 | 2 | 0 | 24 | 22 | 20 | 18 | 16 | 14 | 12 | 10 | 8 | 6 | 4 | 2 |
| 25 | 0 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

# SIM Camp Epsilon: Making and Breaking Codes

Count the occurrence of each letter in the following passages:

1. Mr. and Mrs. Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M |
| | | | | | | | | | | | | |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| | | | | | | | | | | | | |

2. Ty. Kbyzslf dhz aol kpyljavy vm h mpyt jhsslk Nybuupunz, dopjo thkl kypssz. Ol dhz h ipn, illmf thu dpao ohyksf huf uljr, hsaovbno ol kpk ohcl h clyf shynl tbzahjol. Tyz. Kbyzslf dhz aopu huk isvukl huk ohk ulhysf adpjl aol bzbhs htvbua vm uljr, dopjo jhtl pu clyf bzlmbs hz zol zwlua zv tbjo vm oly aptl jyhupun vcly nhyklu mlujlz, zwfpun vu aol ulpnoivyz. Aol Kbyzslfz ohk h zthss zvu jhsslk Kbkslf huk pu aolpy vwpupvu aolyl dhz uv mpuly ivf hufdolyl.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M |
| | | | | | | | | | | | | |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| | | | | | | | | | | | | |

3. Sbhe fpber naq frira lrnef ntb bhe snguref oebhtug sbegu ba guvf pbagvarag, n arj angvba, pbaprvirq va Yvoregl, naq qrqvpngrq gb gur cebcbfvgvba gung nyy zra ner perngrq rdhny.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M |
| | | | | | | | | | | | | |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| | | | | | | | | | | | | |

## 5.4 Decryption practice and Skittle Key Cryptography

The materials in this section in order are:

1. Example sentences worksheet, with frequency analysis and decrypted messages

Solve the six messages. Each message has a DIFFERENT KEY!!!

1. Ocz mzd nji zcp iym zyv iya jpm yvt nod gnp hhz mqv xvo dji odg nxc jjg xjh znv gji bep noo jzi ydo.

Key:____________

Original Message:

2. Jrq bab gjn agg bjb ex. Qha pna qvq rir elg uva t.

Key:____________

Original Message:

3. Sio wuh qyu lgs mqy unm bcl n. –Duw uvM oln iLc iom

Key:____________

Original Message:

Solve the six messages. Each message has a DIFFERENT KEY!!!

4. Jdr jyz jar qq. Fky viw zxy kze xxr dvj riv tcr jjz trc dlj zt. –Gif x.

Key:____________

Original Message:

5. Oek qhu qrb ere vsu bbi. Xux qju ije huq tqd txu bya uiu qjy dwv eet.

Key:____________

Original Message:

6. Bpt dxn ibg max fhk gbg zma xgb wtu.

Key:____________

Original Message:

## References

[1] Dale J Bachman, Ezra A Brown, and Anderson H Norton. Chocolate key cryptography. *Mathematics Teacher*, 104(2):100–104, 2010.

[2] Diffie-Hellman key exchange. https://en.wikipedia.org/wiki/diffie?hellman_key_exchange.

[3] Robert Lewand. *Cryptological mathematics*. MAA, 2000.

[4] Unmasked: An Analysis of 10 Million Passwords. https://wpengine.com/unmasked/.