

Fun With Crypto – Keeping Secrets

[From Ancient Greek Warriors, Enemies of the Roman Empire,
Medieval English Kings, and Modern Superpowers]

Jonathan A. Poritz

jonathan@poritz.net or
jonathan.poritz@csupueblo.edu
www.poritz.net/jonathan

Center for Teaching and Learning *and*
Department of Mathematics and Physics
Colorado State University-Pueblo

21 July 2018



This work is released under a [Creative Commons Attribution 4.0 license](https://creativecommons.org/licenses/by/4.0/).

These slides can be found at <https://poritz.net/j/share/FwCKS>

Why **Crypto**?

Why is this session called “Fun With **Crypto**?” Well...

Crypto is short for **cryptography**, which comes from the Greek roots

- *kryptos*, *κρυπτος* secret, hidden
- *grapho*, *γράφω* draw, write, written

So **cryptography** means *[the study of] secret writing*.

What is a **secret**?

Why kinds of things do you keep secret?

Why? From whom?

Why would you write down a secret? Why would you send it to someone else?

Some Reasons for Secrecy

Here are some things which are thought to be important to keep secret:

- State secrets, particularly military ones
- Commercial secrets – plans, contracts, methods, ideas
- Financial secrets, both for individuals and companies
- Health information – as a matter of a law (e.g., one called **HIPAA**)
- Educational information – as a matter of law (e.g., **FERPA**)
- Many other kinds of personal information, with different legal protections in different countries
- Your own personal thoughts and ideas, which you might send from the past to the future by writing them down in a diary¹
- **Anything you send over the Internet**, because of the way it is built

¹ Leonardo Da Vinci kept a diary in which the writing was all mirror-reversed ... maybe to keep them secret?

Situations Requiring Secrecy: Ancient Greece

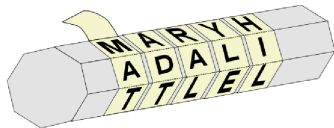
Lots of wars, lots of military secrets...



Wikimedia Commons: Theseven seas / *derivative work Perhelion (derived from: World map longlat.svg:) [CC BY-SA 3.0 <https://creativecommons.org/licenses/by-sa/3.0/>]

Wikimedia Commons: Map_Peloponnesian_War_431_BC-fr.svg: Marsyasderivative work: Aeonx [CC BY-SA 2.5, CC-BY-SA-3.0 <https://creativecommons.org/licenses/by-sa/2.5>, <http://creativecommons.org/licenses/by-sa/3.0/>]

The Scytale



What is the relationship between the letters of the original message and the letters as they appear after they've been written onto a strip using the scytale? *E.g.*, are there any new or missing letters?

Suppose a spy got a quick look at someone using a scytale – maybe from a distance, in dim light. Would that spy later always be able to figure out what a captured message said, or would they need an extra piece of information to do this?

Terminology

It helps to have special terms to talk about complicated things like crypto.

The **plaintext** is a message someone wants to keep secret, in the original form that makes sense to human readers.

The **ciphertext** is the scrambled version of the plaintext which has been transformed so that other people cannot read it.

Making the ciphertext out of a plaintext is called **encryption**.

Getting the plaintext back out of the ciphertext is called **decryption**.

An extra piece of information that the sender has to choose to encrypt the plaintext, which the receiver usually needs also to know in order to decrypt the ciphertext, is called a **key**.

What are *encryption*, *decryption*, and *keys* for the Scytale?

A Little Scytale

Make up your own secret message – your own plaintext – and use your scytale to encrypt it.

Find someone else *who has a different sized scytale from you* and trade messages with them.

Decrypt their message with your scytale.

Come write your decrypted message on the board.

If your decrypted message makes sense, you're done. If not, try again!

[Meaning: find someone else with a different sized scytale, trade messages, and decrypt.]

Situations Requiring Secrecy: Ancient Rome

Julius Cæsar wanted to conquer the whole “known” world. He made a good try. [He knew about India and didn't get it, or Egypt, or Scotland....]



All that conquering required lots of wars, and good secret [military] communications.

Wikimedia Commons: by Cristiano64 [GFDL or CC-BY-SA-3.0, <http://www.gnu.org/copyleft/fdl.html> or <http://creativecommons.org/licenses/by-sa/3.0/>]

The Cæsar Cipher

Cæsar decided to change the letters themselves, without moving them. *Encryption* of a plaintext went like this: take every letter in the plaintext and simply replace it with the letter three steps down the alphabet from the original.

Wait a second: What should we do with the letters **x**, **y**, or **z**?

What is the ciphertext for the plaintext “*Lazy Romans don’t win empires*”?

How should *decryption* work for the Cæsar cryptosystem? **Note:** a good answer will have to do be careful with ciphertext letters **a**, **b**, and **c**.

Decrypt the ciphertext “*ahuahvzdvnlgjrishuvld*”.

What is the *key* we are using ... without really thinking about it? [*Hint: when Cæsar sat down for the first time to encrypt a plaintext, what choice did he make that he could have easily made in a different way?*]

The Tasty Math of Modular Arithmetic

That thing we did with *wrapping the letters around the end*, where three steps after **y** was **b**, etc., is actually something we are all used to doing with numbers on a clock face.

Imagine there might be n numbers on a clock face – so n would be 12 for a regular clock and would be 26 for the letters used in a Cæsar cipher. Then arithmetic on this weird clock is called **arithmetic mod n** .

Arithmetic mod n is some very pretty, interesting, and useful mathematics. It was invented by **Carl Friedrich Gauss** around 1800 and has since become a basic but important tool in the toolbox of modern mathematics.



The fact that we are comfortable doing modular arithmetic and related mathematical topics is part of why mathematicians are so useful in computer security and cryptography.

Back To The Cæsar Cipher: Different Keys

So Julius Cæsar used the key $k = 3$. His nephew Octavian, who later took the name Augustus Cæsar, liked to use the key $k = -1$.

How many possible keys are there?

Suppose you intercept a ciphertext made with the Cæsar cipher, but you don't know the key. What would you do to figure out the plaintext?

What plaintext produced this ciphertext, and what key was used?

nvyfcukyvjvkiikyjkfsvjvcwvmzuvekkyrkrccdverivtivrkvu
vhlrckyrkkyvprivveufnvuspsyvzitivrkrfinzkytvikrzelerc
zverscvizxykjkyrkrdfexkyvjvrivczwvczsvikpreukyvglijl
zkfwyrggzevj

Cryptanalysis:

Security Through Obscurity vs Kerckhoff's Principle

When you know what the method of encryption was to make a ciphertext, but not the key, trying to get the plaintext back by some other method is called **cryptanalysis**. Colloquially, this is *breaking* or *cracking the code*.

You may say: “Why not keep the method of encryption a secret, too?” This approach is called **security through obscurity** and is **a very bad idea** unless you (a) really know what you're doing and (b) will only be using this secret communications method once.²

The idea that we should make the encryption and decryption methods public and only keep the key secret is called **Kerckhoff's Principle** and is named after a French military cryptographer from the 1880s.

²See, e.g., the *Voynich Manuscript*.

“Brute-Force Attack”

When we try cryptanalyzing a mysterious ciphertext by trying all possible keys, it's called using a **brute-force attack**. This only works if there is a fairly small number of keys, so we can do all this work and not die of boredom (or old age).

Let's try decrypting just the first two letters of that long mystery message with all possible keys (Cæsar's soldiers could have done that much work).

1: mu	5: iq	9: em	13: ai	17: we	21: sa	25: ow
2: lt	6: hp	10: dl	14: zh	18: vd	22: rz	26: nv
3: ks	7: go	11: ck	15: yg	19: uc	23: qy	
4: jr	8: fn	12: bj	16: xf	20: tb	24: px	

We could keep going on the ones which might be real words... but it gets boring really fast. Which is why Cæsar would have liked a computer.

Teaching A Computer To Do The Boring Stuff, 1

Computers are very fast – they can do *millions* of calculations every second – but extremely stupid: you have to tell them exactly what to do, in very small, simple words [called a programming language].

Start your computers, and get **Firefox** and **Octave** running.

Octave is a programming environment, we are going to teach it to do Caesar en/decryption. **Octave** is free: gnu.org/software/octave/.

To put some information in a container – called a **variable** – you just type

```
x = <the information> [and hit the ENTER key]
```

If that information is a bunch of letters (some text) – called a **string** in the programming world, just put it inside quotation marks (single or double, either work), like

```
x = 'this is a happy little string'
```

Or the information could be a number, like

```
x = 17
```

Teaching A Computer To Do The Boring Stuff, 2

If the variable x has a string which we want to do Cæsar encryption to, using the key k , we might try doing

$$x + k$$

Octave would do the right thing and add the number k to every letter of the string x . But it wouldn't do that cool modular arithmetic stuff!

We have to move all those letters to numbers, starting with "a" as 0, by

$$x - 'a'$$

and then do the Cæsar shifting the letters by k with

$$x - 'a' + k$$

and finally do the "wrapping around the end of the alphabet" thing we talked about before, using

$$\text{mod}(x - 'a' + k, 26)$$

Unfortunately, this will produce an result which is just a bunch of numbers, so we want to turn them back into a string by

$$\text{char}(\text{mod}(x - 'a' + k, 26) + 'a')$$

Cæsar's Computer, 1

Using the **Firefox** browser, please just cut and paste the four lines from poritz.net/j/share/FwCKS/caesar.html into your running **Octave** environment.

[This does exactly what we did above, but first it cleans up the plaintext by taking out all non-letter characters and making it all lower case.]

Now your **Octave** knows how to do Cæsar encryption of a string in the variable **x** with key **k** by just typing

```
caesar(x, k)
```

Or you can input the data directly in either place, or both, with

```
caesar('this is a test', k)      or
```

```
caesar(x, 3)      or
```

```
caesar('this is a test', 3)
```

You can also store the output ciphertext in a variable **y** by doing

```
y = caesar(x, k)
```

Cæsar's Computer, 2

Use the program **caesar** in your **Octave** to do Cæsar encryption of a word or phrase you like (the name of the street where you live, the name of your favorite food, whatever) using as key whatever number you like. Write the resulting ciphertext on the board.

Wait! How are we going to do Cæsar *decryption* in **Octave**? *[Hint: you can do Cæsar decryption with key **k** using our **encryption** program **caesar** but giving it a different key!]*

Let's decrypt one of the messages on the board, someone tell the key they used.

Now let's try to cryptanalyze the next message on the board, using the brute force approach. We could start typing in all the **Octave** commands

Cæsar's Computer, 3

starting with

```
caesar(x, -1)
```

```
caesar(x, -2)
```

```
caesar(x, -3)
```

```
caesar(x, -4)
```

```
...
```

But that gets boring really fast – and computers can do boring things really accurately and quickly!

So we can make the computer try all possible keys for decryption by

```
for i=1:26
```

```
    caesar(x, -i)
```

```
endfor
```

and then we just pick the one that looks like a plaintext.

Another Approach To Cryptanalysis: Counting Things

Another way to do cryptanalysis is to look at that mystery ciphertext
nvyfcukyvjvkiikyjkfsvjvcwvmzuvekkrykrccdverivtivrkvu
vhlrckyrkkyvprivveufnvuspsyvzitivrkfinkzytvikrzelerc
zverscvizxykjkyrkrdfexkyvjvrviczwvczsvikpreukyvglijl
zkfwyrggzvjj

and to notice that there are a lot of **v**'s, **k**'s, and **j**'s. Hmm....

But what are the three most common letters in the English language?

Get the program **histo** from the site poritz.net/j/share/FwCKS/histo.html (the same way we got **caesar**). Then you can see which letters are most frequent in a string contained in the variable **x** by typing

```
histo(x)
```

and looking at the pretty picture.

If you want to cut-and-paste the above mystery ciphertext, it is in poritz.net/j/share/FwCKS/ciphertext.html

Cryptanalysis using letter frequencies is called **frequency analysis**.

More Advanced Encryption: Vigenère and One-Time Pads

If you want to prevent a brute-force attack, you have to make Cæsar stronger, somehow. One way, called **Vigenère encryption**, is to use different Cæsar keys for the first, second, and third letters of the message, and then repeat those three key values for the next letters, *etc.* Or even more than three different Cæsar keys!

Vigenère encryption was unbreakable for hundreds of years, and was used by European royalty to protect their secrets. But it was eventually broken, too – I teach my students in a class here at CSU-Pueblo how.

Except if there are as many Cæsar keys as letters in the plaintext, then the whole pile of Cæsar keys is called a **one-time pad**, and there is a mathematical proof that

Encryption with a one-time pad is, and will always be, unbreakable.

[Yes, unbreakable by everybody,³ including the US Government, because mathematics is more powerful than the world's greatest superpower.]

³Take that, Hollywood movies where someone says “There’s no such thing as an unbreakable code!”

Homework Assignment

[Not really homework.]

If you want to look over these slides in the future or share them with someone, I'll put them on my website at

poritz.net/j/share

(I put everything I write or use in public up there, so you might have to look through the list to find this particular presentation.)

If you are interested in crypto, there is a ton of interesting books, free software, videos, *etc.*, that you can learn from. Send me an email (my contact info is at poritz.net/j) and I can make some suggestions.

I also love to talk to students (and teachers) about all aspects of crypto, computer security, modular arithmetic, and many other areas of math and science. So don't hesitate to contact me for anything at all related to math/science/computers!

XYZABCDEFGHJKLMNOPQRSTUVWXYZABC

Encrypt, for Julius Cæsar, the plaintext

Lazy Romans don't win empires

Julius's spy sent him the ciphertext

ahuahvzdvnjqjrishuvld

What was the original plaintext the spy meant to send?

A Challenge

Figure out what plaintext and what key produced this ciphertext

```
rtgpxpjzfcetcpojzfcazzcjzfcsoowpoxlddpdjplcytyrezmcpl  
espqcppesphcpenspocpqfdpzqjzfcpeppxyrdszcpdpyoespdpesp  
szxpwpddep xapdeezdeezxptwtqexjwlxampdtopesprzwopyozzc
```

[This can be cut-and-pasted from
poritz.net/j/share/FwCKS/challenge.html .]

Again, plaintext and key for this ciphertext – this one is harder.

```
kllkofkdplrqxtxpmjxhfkdxkljfkrlrpplrkaxplrka xhfkqlxhix  
ulkloxqlzpfkcifqpxylrqxrdrpqrptelexpexaxyakfdeqpfqpr  
myifkhfkdxkamroyifka
```

[This can be cut-and-pasted from
poritz.net/j/share/FwCKS/harder.html .]