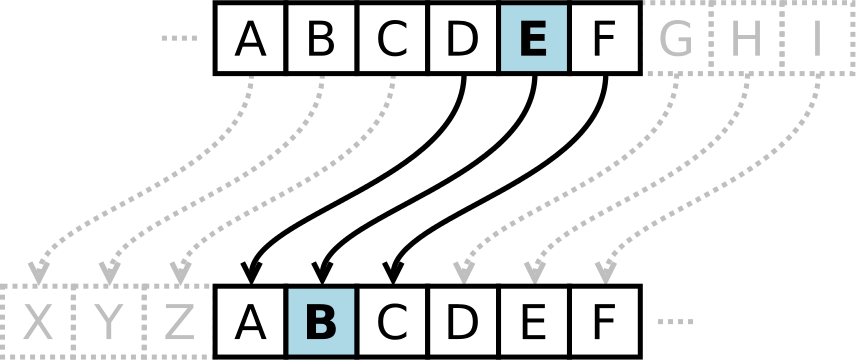
Ciphers and Codebreaking



My name is\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

My name in Caesar Rot7 is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Contents

[Section One: The Atbash 4](#_Toc452712681)

[Example 1: 5](#_Toc452712682)

[Section Two: Caesar’s Cipher 5](#_Toc452712683)

[A Special Case of the Caesar Cipher 10](#_Toc452712684)

[Some Examples for Decryption 10](#_Toc452712685)

[Example 2: Caesar, rot 5 10](#_Toc452712686)

[Example 3: Atbash 11](#_Toc452712687)

[Example 4: Caesar, rot unknown 11](#_Toc452712688)

[Example 5: Caesar, rot unknown 12](#_Toc452712689)

[Section Three: Beginning Decryption 12](#_Toc452712690)

[Small word! 12](#_Toc452712691)

[Brute Force 14](#_Toc452712692)

[Section Four: Keyword Ciphers 16](#_Toc452712693)

[Section Five: A Fine Cipher 18](#_Toc452712694)

[Encryption 18](#_Toc452712695)

[Decryption 21](#_Toc452712696)

[Example 6: Affine Cipher 21](#_Toc452712697)

[Section Six: Frequency Analysis 24](#_Toc452712698)

[Helpful Commonalities 26](#_Toc452712699)

[Example Seven: Unknown Substitution Cipher 27](#_Toc452712700)

[Section Seven: Home Made Recipes 28](#_Toc452712701)

[Pigpen 28](#_Toc452712702)

[Ulam and the Spiral 29](#_Toc452712703)

[Arielle’s 33](#_Toc452712704)

[Kerckhoffs's principles 34](#_Toc452712705)

[Section Eight: Vigenere’s Cipher 35](#_Toc452712706)

[Some Examples for Decryption 38](#_Toc452712707)

[Section Nine: Steganography 39](#_Toc452712708)

[Section Ten: P and NP 42](#_Toc452712709)

[Modern Crypto 42](#_Toc452712710)

[Section Eleven: Closing thoughts 44](#_Toc452712711)

[Navajo Code Talkers 44](#_Toc452712712)

[Defeating Frequency Analysis 46](#_Toc452712713)

[Un-cracked codes 48](#_Toc452712714)

[The Voynich Manuscript 48](#_Toc452712715)

[Taman Shud 51](#_Toc452712716)

# Section One: The Atbash

This is the earliest cipher we will discuss and was used by the ancient Hebrews around 500 BCE. In fact, many examples of text in the Biblical Old Testament are actually encrypted via the Atbash Cipher. As it turns out, it is fairly easy to send a message using this technique, which is equally difficult to break. However, lots of people around them were still using Hieroglyphs and cuneiform, which aren’t alphabets *per se* and so wouldn’t allow for this trick:

Plain: אבגדהוזחטיכלמנסעפצקרשת

Cipher: תשרקצפעסנמלכיטחזוהדגבא

All you do is reverse the alphabet’s order to get the cipher text:

Plain: abcdefghijklmnopqrstuvwxyz

Cipher: ZYXWVUTSRQPONMLKJIHGFEDCBA

Or, more simply, because the alphabet we use has an even number of letters:

First 13 letters: A|B|C|D|E|F|G|H|I|J|K|L|M

Last 13 letters: Z|Y|X|W|V|U|T|S|R|Q|P|O|N

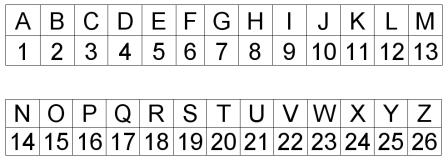
Like I said above, this is easy to decrypt because you do the same procedure to encrypt as you do to decrypt it. Try your hand out:

## Example 1:

GSV PRMT LU HSVHSZXS HSZOO WIRMP ZUGVI GSVN.

# Section Two: Caesar’s Cipher

Julius Caesar is reported to have used this next one. First, we start by giving each letter of the alphabet a number:

Next, we are going to take each letter and add three to its number. Contemporary accounts reported that he used +4, while his nephew Caesar Augustus was reported to use +1.

So now, ATTACK AT DAWN becomes

1 20 20 1 3 11 1 20 4 1 23 14

When you add 3 to each, you get

4 23 23 4 6 14 4 23 7 4 26 17, which then encrypts to:

DWWDFQ DW GDZT

To decipher this, your intended recipient must know that you shifted by +3, so they must shift by +3 also, right?

Wait, if I do that, I get: GZZGIN GZ JGCQ…wait, that’s not right at all. I guess we should subtract 3 to decrypt it back to ATTACK AT DAWN. Shifting it +3 a second time did bring up a good problem, though! What happens when we have Z/26 and must keep adding? Caesar Augustus just started over at AA and then kept going, but we are going to loop back around to the beginning of the alphabet and start over at A, which is how Z +3 became C.

Doing this kind of loop is a little bit like working with a clock: if you had 10 o’ clock and add four hours, you end up passing 12 and starting over, ending up at 2. Likewise, we will pass Z and start over at A, ending up wherever the number leads us. This kind of math is called “modular arithmetic” and is noted by having your process in (parenthesis) followed by mod \_\_\_, where the \_\_\_ has the number that is the highest before you start over.



A representation of an alphabet clock, with 26 possible positions

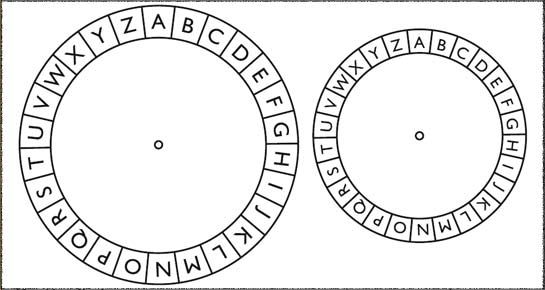
So for a clock, our example was (10+4)mod12, which was 2.

For any cryptography using the Latin alphabet with the English language, we will be working in (\_\_\_)mod26, because there are 26 letters before we start over again. In our example, we started with (n+3)mod 26, then did (n+3)mod26 again, so W became Z, which then became C.

The benefit of Caesar’s cipher is that you only need to learn one technique, but you get 25 different options on how to use it (why not 26?), unlike the Atbash cipher which only works in one way. On the other hand, it requires you and your intended recipient to confer ahead of time to decide what your constant of rotation will be (e.g.,+20).

The biggest benefit of the Caesar cipher is that it allows you to construct a wheel. Cut this one out and lock it in place with a brad. Start with A above A and B above B, etc. Then begin rotating letter by letter until you reach your desired constant of rotation. Once you have that in place, all that math is done for you!

Of course if your enemy has one, then your cipher isn’t very safe, is it?



## A Special Case of the Caesar Cipher

I said above that the process for encryption and decryption of the Caesar Cipher is different [(n+3)mod26 then (n-3)mod26, e.g.], but that’s not always true. If we rotate by 13, then we are half-way around our circle, and rotating by 13 again gets us back to where we started. That means that Rot13 is a special case of the Caesar Cipher where the process for encryption and decryption is exactly the same!

## Some Examples for Decryption

Using what you know about the Atbash and Caesar Cipher, can you encrypt or decrypt these texts, as needed?

### Example 2: Caesar, rot 5

Ktzw xhtwj fsi xjajs djfwx flt tzw kfymjwx gwtzlmy ktwym ts ymnx htsynsjsy, f sjb sfynts, htshjnaji ns Qngjwyd, fsi ijinhfyji yt ymj uwtutxnynts ymfy fqq rjs fwj hwjfyji jvzfq.

Stb bj fwj jslflji ns f lwjfy hnanq bfw, yjxynsl bmjymjw ymfy sfynts, tw fsd sfynts xt htshjnaji fsi xt ijinhfyji, hfs qtsl jsizwj. Bj fwj rjy ts f lwjfy gfyyqj-knjqi tk ymfy bfw. Bj mfaj htrj yt ijinhfyj f utwynts tk ymfy knjqi, fx f knsfq wjxynsl uqfhj ktw ymtxj bmt mjwj lfaj ymjnw qnajx ymfy ymfy sfynts rnlmy qnaj. Ny nx fqytljymjw knyynsl fsi uwtujw ymfy bj xmtzqi it ymnx.

### Example 3: Atbash

Dsvm rm gsv Xlfihv lu sfnzm vevmgh rg yvxlnvh mvxvhhzib uli lmv kvlkov gl wrhhloev gsv klorgrxzo yzmwh dsrxs szev xlmmvxgvw gsvn drgs zmlgsvi zmw gl zhhfnv znlmt gsv kldvih lu gsv vzigs, gsv hvkzizgv zmw vjfzo hgzgrlm gl dsrxs gsv Ozdh lu Mzgfiv zmw lu Mzgfiv'h Tlw vmgrgov gsvn, z wvxvmg ivhkvxg gl gsv lkrmrlmh lu nzmprmw ivjfrivh gszg gsvb hslfow wvxoziv gsv xzfhvh dsrxs rnkvo gsvn gl gsv hvkzizgrlm.

### Example 4: Caesar, rot unknown

Z yfgv pfl'iv veafpzex kyzj vovitzjv jf wri. Vetipgkzfe reu Uvtipgkzfe, reu kyv kyflxyk fw tfuvj reu crexlrxv xrdvj ze xvevirc, drbv wfi xffu dvekrc xpderjkztj. Kyvp jkivkty pfli sirze reu yvcg pfl xifn rj tizkztrc kyzebvij.

### Example 5: Caesar, rot unknown

Yjj Eysj gq bgtgbcb glrm rfpcc nyprq, mlc md ufgaf rfc Zcjeyc glfyzgr, rfc Yosgrylg ylmrfcp, rfmqc ufm gl rfcgp mul jylesyec ypc ayjjcb Acjrq, gl msp Eysjq, rfc rfgpb. Yjj rfcqc bgddcp dpmk cyaf mrfcp gl jylesyec, asqrmkq ylb jyuq.

# Section Three: Beginning Decryption

## Small word!

I hope you noticed something in Example 4 in the last section. It begins, “Z yfgv pfliv…” If ever you come across a cipher like this, your first instinct should be to gravitate toward the single letter words. In this case, we have a z by itself. There are only two real possibilities for what the constant of rotation could have been: Either the z represented the English word A or the English word I, and we can guess-and-check both of them.

If Z is A, then the text was rotated by 25 and we must simply rotate by negative 25 to decrypt it. Or, you know, we could just add 1 to everything. That would work too, because we are still using mod26.

Doing this, we get:

Z yfgv pfl'iv veafpzex kyzj vovitzjv jf wri.

A zghw qgm'jw wfbgqafy lzak wpwjuakw kg xsj.

Okay, we can probably rule that out.

If Z is I, what is the constant of rotation? What did you find?

Thism ethod ofana lysis means thatw henwe useou rownc odesw emust accou ntfor thisk indof trick Codeb reake rskno wthat singl eandd ouble lette rword saref ewinE nglis handt heyca nexpl oitth istoc racko urcar efull ycode dmess agesI norde rtoth wartt hiski ndofa nalys iswew illdo somet hingv erysi mplew ithou rchos entex twewi llarr anget hespa cesin words tobes uchth atyou canno ttell howlo ngeac hword isLoo kIved oneth atwit hthis entir epara graph

Of course, that paragraph is still readable in English, though sometimes it might take a second to parse. But when encrypted, it becomes impossible to count word lengths to look for clues. Here are the first two lines of the previous paragraph again, this time in rot 17.

kyzjd vkyfu fwrer cpjzj dvrej kyrkn yvenv ljvfl ifnet fuvjn vdljk rttfl ekwfi kyzjb zeufw kiztb tfuvs ivrbv ijbef nkyrk jzexc vreuu flscv cvkkv infiu jrivw vnzeV exczj

Not so easy now, huh?

Traditionally, cryptographers have used groups of 5 due to the constraints of telegrams, but these days it’s common for computers and advanced cryptographers to just eliminate the spaces altogether. It does make it tough for human eyes to parse, but computers have no trouble with that at all. What happens if the total number of letters in your text isn’t a multiple of 5? Just add a few random letters to the end. The person reading your message will realize they are dummy text, but the person trying to eavesdrop on your communique will be stuck.

## Brute Force

So if we can’t rely on word lengths, what can we do?

Well, for Caesar, “brute force” attacks work quite well. Use this chart below for the given text to see which rotation constant makes the text make sense. Use pencil and erase so that you can re-use this chart as needed.

jrccp yrjrs clvui vjjoo

|  |  |
| --- | --- |
| Original Text (rot 0) |  |
| Rot 1 |  |
| Rot 2 |  |
| Rot 3 |  |
| Rot 4 |  |
| Rot 5 |  |
| Rot 6 |  |
| Rot 7 |  |
| Rot 8 |  |
| Rot 9 |  |
| Rot 10 |  |
| Rot 11 |  |
| Rot 12 |  |
| Rot 13 |  |
| Rot 14 |  |
| Rot 15 |  |
| Rot 16 |  |
| Rot 17 |  |
| Rot 18 |  |
| Rot 19 |  |
| Rot 20 |  |
| Rot 21 |  |
| Rot 22 |  |
| Rot 23 |  |
| Rot 24 |  |
| Rot 25 |  |

Which rot made the text make any sense at all? If you got the right answer, your message was actually one of the 200 nonsense-sounding messages delivered over the radio in France to warn those in-the-know of the impending invasion on D-Day in 1944. Not bad!

# Section Four: Keyword Ciphers

Okay, so every Caesar Cipher can be solved by a few minutes spent on the brute force technique. If that’s the case, how were they ever successful for Caesar? Well, for one thing, the Burgundians weren’t literate, so most of the time when Caesar’s enemies intercepted his cipher, they couldn’t read it anyway. If they could, they might have thought Caesar was writing in a foreign language.

Are there ciphers that can’t be broken this way? I’m sure you’re already contemplating the creation of one, but before we get there, here’s an example that already exists!

Keyword ciphers involve the use of a code word to jumble up the alphabet. The better your code word the more jumbled it is! This means that there is no *constant* of rotation, because of the particular way each different codeword will jumble up the alphabet. An example, you ask? Why sure!

Let’s take as our codeword “DARK KNIGHT”

Now, for this to work, we have to assume each letter is used only once, so we will in fact be using the codeword: DARKNIGHT which still sounds cool.

First, we write the alphabet as usual. Then we write the alphabet again below it, but with all of the letters in the key word shifted to the beginning:

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

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

Note that although the beginning of the alphabet is jumbled, the closer you get to the end, the less “off” the letters are, with the result that after we moved the “T” all the remaining letters matched up. This can be infuriating for someone trying to analyze, as they try different mathematical rotations, each showing tantalizing glimpses, but never quite working. On the other hand, it means that when we choose a key word, it is important to choose one that incorporates letters near the end of the alphabet. On the other hand yet again, the last letters of the alphabet aren’t that common (see page 26), so maybe it doesn’t even matter.

Use the key word VIGENERE to encrypt the following message:

“The enemy is close and we are nearly surrounded. We need reinforcements as soon as possible. Send help, preferably with tanks.”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The problem with a keyword is, of course, that your recipient needs to know the keyword. If they forget, or are hit hard enough on the head, they may not be able to receive your message, which could be annoying. Likewise, if they are tempted to divulge the keyword, perhaps as a result of a bribe, then this cipher is easily cracked. So perhaps it’s best to use with someone who is exceptionally trustworthy and with a good memory.

# Section Five: A Fine Cipher

## Encryption

The Affine Cipher requires a key, just like above, but the key is two separate integers[[1]](#footnote-1). They are used in a binomial expression with the number, n, of the letter in question. That expression can be written like this:

(an+b)mod26

What that means is that for every letter of the alphabet, we can encrypt it if we first multiply its number by **a**, then add **b**. A person decrypting it will do this process in reverse.

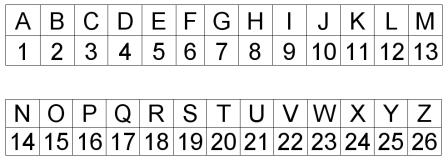
In the parlance of cryptographers, we would write

E(n) = (an+b)mod 26

Meaning, “the encryption with the number of our letter being ‘**n**’ is to multiply it by one number, **a**, then add another, **b**, and then subtract 26 as many times as necessary to get a number less than 26.”

There’s one catch, though: whatever number **a** is, it cannot have any common factors with 26[[2]](#footnote-2). That is to say, it must be *coprime* with 26, they only share the factor of 1. If you use a number that does share a factor with 26, for example 4, which shares the factor of 2, then when you encode into the Affine Cipher, you run the risk that two of your plaintext (original) letters might encrypt to the same ciphertext letter. The person you intend to read it will then not know: was this J supposed to be an A or an N? It could have been both! Very confusing. In fact, maybe it’s a good idea to use only prime numbers for this, although 26’s factors are 2 and 13 so as long as we avoid even numbers and multiples of 13 for **a**, we should be fine.

Having that number-letter correspondence chart will be helpful here:



Okay, let’s try an example with **a=5 and b=4.** Five is coprime with 26 because they have no common factors apart from 1.

Let’s start with A.

A is 1, so

(1 x 5) + 4 = 9

Nine is I, so in our example, A encodes to I.

Now fill out the rest of this chart below using the same mathematical process:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | b | c | d | e | f | g | h | I | j | k | l | m |
| I |  |  |  |  |  |  |  |  |  |  |  |  |
| n | o | p | q | r | s | t | u | v | w | x | y | z |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

What about E? E is 5, so we would do

(5 x 5) + 4 = 29

There is no 29! Oh, right, this is modular arithmetic still. We just subtract 26 and go around the alphabet again, to get 4

Okay, so E is actually D. Got it.

This looks pretty tough to decipher, doesn’t it? In addition to requiring some algebra skills, this cipher has the added benefit of requiring two separate integers to decode, meaning that if you have two different recipients but you don’t trust either of them to work alone, you can give each one of them one piece of the code! Not sure when that would be handy, but just in case you need to make sure none of your generals is going to go rogue and launch the … okay, I’m back. Sorry.

## Decryption

So how do we decode this, if we’re the recipient?

Suppose we got a message in the Affine Cipher that was **a=3** and **b=7.**

The message says,

### Example 6: Affine Cipher

I Z T V E J L Y J Q Q V W

Let’s start with I, which is 9.

To work backward from here, we have to undo the equation involved, which was encrypted like this:

(an+b)mod 26

As noted above, in the parlance of cryptographers, we would write

E(n) = (an+b)mod 26

Meaning, “the encryption with the number of our letter being ‘**n**’ is to multiply it by one number, **a**, then add another, **b**, and then subtract 26 as many times as necessary to get a number less than 26.”

The corresponding decryption would be

D(n)=a-1(n-b)mod26

Where **a-1** would be the “Modular multiplicative inverse” of **a**. Just like when doing regular algebra, we have to find the multiplicative inverse and apply it to negate the **a**, then we have to find the additive inverse of **b** (that is, **-b**) and apply it to negate **b**. Then all that remains is **n**, right? Adding and subtracting in **mod26** is the same as what you’re used to, but applying the multiplicative inverse is a bit different, which means we need a new concept.

Remember that the regular “multiplicative inverse” is any number that, when you multiply it by a given number, returns the “identity element.” In human words, that means that if I multiply 1/7 times 7, I get 1. Any number times 1 is that same number, so it’s the multiplicative identity element. That means that 1/7 is the multiplicative inverse of 7.

What I need to do is find a number that, when I multiply it times our **a** (which was 3) I get 1(mod26). What numbers can be equal to 1(mod26)? Well, 1, obviously. But also 27, 53, 79, 105, and so on. Just keep adding 26. There should be only one number less than 26 that, when I multiply it by 3 (our **a** value) I get 27 or 53 or…

As it turns out, 9 x 3 = 27, so for this letter, **a-1** is 9.

So we would take our decryption formula,

D(n)=a-1(n-b)mod26

And plug in what we know:

9(n - 7) and we would plug in our starting value, which was also, by coincidence, 9[[3]](#footnote-3)

9(9 – 7) =9(2) = 18. So I maps to the 18th letter of the alphabet, R.

I go back to my original message, and anywhere I see an I, I eliminate it and put an R in its place. Do I have to repeat this process for every letter of the message? Yes. God, it almost seems like trying to break the code is easier than trying to decrypt it using the key, but trust me, it isn’t.

However, this doesn’t mean the Affine cipher is impossible to crack. Though I will make light of this in an upcoming paragraph, the truth is that there are only 312 possible Affine Ciphers. How is this possible, when there are an infinite number of **a** and **b** values? Well, there are only 12 numbers that are coprime with 26 that are less than 26. Anything bigger than 26 would be reduced in mod26 to something that’s less than 26, so we can rule all of them out. Having done that, there are only 26 possible b values at all (because, again, anything bigger than 26 would give the same result as something below 26). What that means is that there are only 12 x 26 = 312 possible Affine Ciphers that you can do that are unique. I guess if you had all day and *knew* that an Affine Cipher was used, you could brute force and try all of them. But I prefer…

# Section Six: Frequency Analysis

So how would we break a code like the Affine Cipher or the keyword cipher if we didn’t know the keyword or **a** and **b** numbers? I mean, like we said, the **a** numbers are restricted to anything not even or a multiple of 13, so that narrows it down to…infinity. Drat! On the other hand, assuming they use a real word for the keyword, we could just use the brute force method and try it with every word in the English language. How many could there possible be?[[4]](#footnote-4)

The answer is that we use frequency analysis, which is even more tedious than it sounds.

We take advantage of the fact that all languages follow statistical patterns over use. This is related to both the Law of Large Numbers and Zipf’s Law, and I recommend you take a break from reading this to go look those up if you don’t already know what they mean. Over a big enough body of text, the most frequent letter used in an English language text will be the letter e. The most common pair of consecutive letters will be ‘th.’ Unsurprisingly, the most common triad of consecutive letters is the combination of the above, ‘the.’ We can use this fact to match up the most common letter in our encoded text and see if it makes sense for it to be ‘e.’ We also do this with the most common digraph and call it ‘th’ and the most common trigraph and call it ‘the.’ Having done this, if we start to see patterns, follow them where they lead. If not, find the second most commonly used letter in your encoded text and repeat the process from there.

## Helpful Commonalities

The top most commonly used English words are:

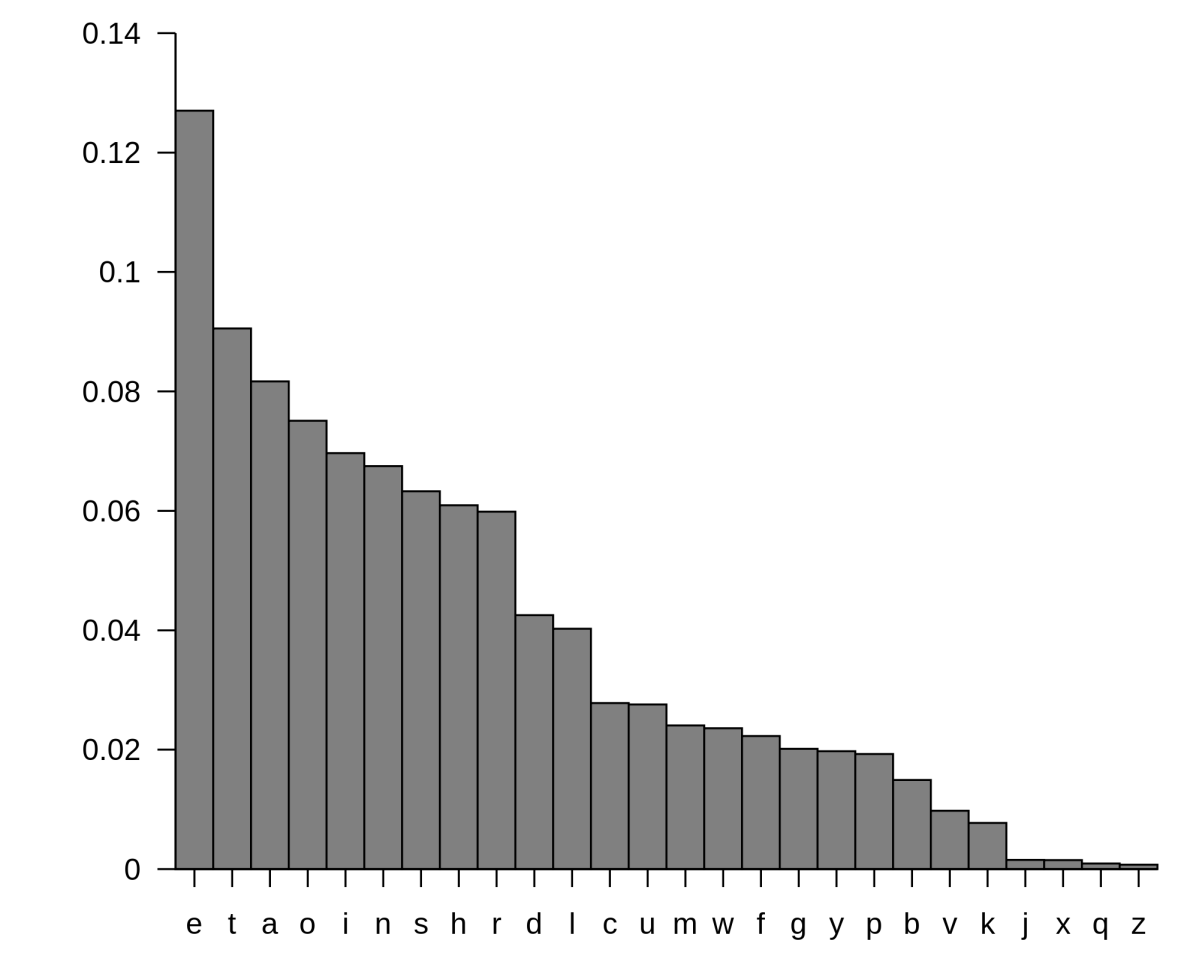
**the be to of and a in that have I it for not on with he as you do at**

The most common letters are listed below.

The most common letter pairs are

**TH HE AN RE ER IN ON AT ND ST ES EN OF TE ED OR TI HI AS TO**

The most common doubled letters as

**LL EE SS OO TT FF RR NN PP CC**

Also helpful is this chart: letter frequency within the English Language. Please note that the decimals on the side correspond with percentages of letters used, so “g” looks to be about 0.02 on the chart, or 2% of all the letters used in English. You thought it was more, didn’t you? You’re a good guy or girl, a great genius, but you gotta get going and stay on your guard.

A sample text to try this with is included below. Try your best to match things up and see where you get. Remember that when you hear hoof beats, you should think of horses (h=6.094%) not zebras (z=0.074%).

Also, it would probably be helpful to write your letters down as you go:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

## Example Seven: Unknown Substitution Cipher

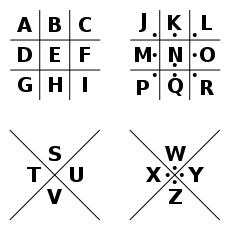
FUKAGPJ NXGJK RGPPF XJ RBZQ YPG YUGQPJK YAP KB YBZQ YPG PUGZH ZPUNJ U NZBMPG OAK BFZH JB UF YBAG KYPF ZPUN JAOJXQPJ KB ZPUN JB PQPF JUFT KB RGXPN JB QUMF RBPJ QBMF KB QUH FBKYXFR RBZQ VUF JKUH

Please note that this text is from a poem, so due to reasons related to rhyme and alliteration, it may not line up with natural speech or text. For that reason, I have left in the original word lengths to make it somewhat easier to decrypt. You’re welcome.

# Section Seven: Home Made Recipes

Perhaps the best way to fool your adversaries on a short term basis is to invent your own cipher that has never been used before. If it has some kind of uncommon quirk that is easy to remember, perhaps you and your recipient can have that quirk memorized and fool anyone who has never studied cryptanalysis before. Then again, maybe not?

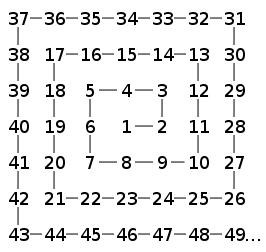
## Pigpen

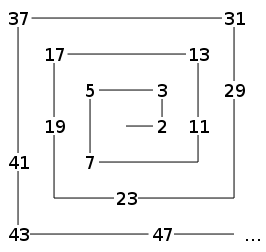
The pigpen cipher is any kind of easily-memorized system that replaces each individual Latin-alphabet letter with a symbol that has no correspondence that anyone on the outside would understand. For example, you could use the diagram on the right, and utilize any part of it that touches the letter area, like this:

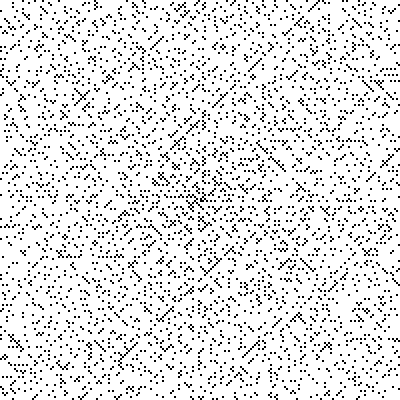
An example pigpen message

How hard would it be to break this code for someone who doesn’t know the system?  
Actually it turns out, quite simple, comparatively. Rather than being totally confused by the weird symbols, if someone knew that this message was encoded to English, all they would have to do is assign each symbol a random letter, then use frequency analysis to solve it. Bingo. Turns out that it is no more difficult to solve than example seven was above, except that for any good frequency analysis, the more letters you have in your example, the easier it is to solve (Law of Large Numbers).

## Ulam and the Spiral

Here’s one that Yours Truly developed many years ago, just for fun. Stanislaw Ulam was a Polish mathematician who was incredibly bored one day during a lecture on mathematics[[5]](#footnote-5). Since he was a number theorist, he was doodling numbers. His doodle was making a spiral out of every number, like so:



But Ulam is a number theorist, and since number theorists love primes, he started noticing something. All the primes started to line up in diagonal lines. Weird! I wonder why it does that? It turns out when you do an Ulam spiral for the first 10,000 numbers, it looks like this:

You can clearly see all of the prime lines going diagonally, and you can see the empty spaces where heavy factors line up, as well. Though primes are essential to understanding modern cryptography, that’s actually not what’s important here. I was inspired by his spiral, so I started writing my own messages in spiral format, as shown on the next page:

Take the message:

Many years ago I contracted an intimacy with a Mr. William Legrand. He was of an ancient Huguenot family, and had once been wealthy; but a series of misfortunes had reduced him to want. To avoid the mortification consequent upon his disasters, he left New Orleans, the city of his forefathers, and took up his residence at Sullivan’s Island, near Charleston, South Carolina.

Re-write it in all caps and with spaces removed:

MANYYEARSAGOICONTRACTEDANINTIMACYWITHAMRWILLIAMLEGRANDHEWASOFANANCIENTHUGUENOTFAMILYANDHADONCEBEENWEALTHYBUTASERIESOFMISFORTUNESHADREDUCEDHIMTOWANTTOAVOIDTHEMORTIFICATIONCONSEQUENTUPONHISDISASTERSHELEFTNEWORLEANSTHECITYOFHISFOREFATHERSANDTOOKUPHISRESIDENCEATSULLIVANSISLANDNEARCHARLESTONSOUTHCAROLINA

Oddly enough, that’s precisely 300 letters.

The closest odd “square” to 300 is 289, which is 17 squared, so I am going to make a big square out of this text on a piece of paper, but I am going to write it starting with the inside and spiraling out until it is finished, hopefully slightly bigger than 17 x 17, which I will then fil in with dummy text.

It might look like this after the first few words:

..... Y C A

T N O C I M  
R Y Y N O I  
A E M A G T  
C A R S A N  
T E D A N I

Which then I would write without spaces as if each line were its own word:

.....YCA TNOCIM RYYNOI AEMAGT CARSAN TEDANI

Pretty similar to gibberish, but easy enough to crack if the text is, say, 300 characters long. Try giving it to a friend and see how long it takes them to crack this without knowing the particular way the cipher works. For your advanced friend, try it in a Caesar cipher first, then apply the spiral cipher and see how unintelligible your response gets.

## Arielle’s

A similar cipher was used by a dear friend when she was in elementary school in order to pass notes to her friend. She would simply space the letters of a message out over an arbitrary number of “words.”

So “Andrew is the cutest boy in class” spaced out over six words would be

AIUOS NSTIS DTEN RHSC EETL WCBA

Pretty neat, and not that easy to crack unless you know what to look for.

This is actually a variant, of a kind, of something called the Rail Fence Cipher, which encodes in this way:

W . . . E . . . C . . . R . . . L . . . T . . . E

. E . R . D . S . O . E . E . F . E . A . O . C .

. . A . . . I . . . V . . . D . . . E . . . N . .

…to get “WECRLTE ERDSOEEFEAOC AIVDEN” which is still ridiculously simple to crack. Primarily, it is easy to crack because it violates rule number two on the next subsection.

## Kerckhoffs's principles

So in terms of code and cipher creation, we need some guiding principles to help us understand what a good code is and what not so great a code is. The good news is, Kerckhoff wrote some guidelines for us in 1883. His rules are as follows:

1. The system must be practically, if not mathematically, indecipherable;
2. It should not require secrecy, and it should not be a problem if it falls into enemy hands[[6]](#footnote-6);
3. It must be possible to communicate and remember the key without using written notes, and correspondents must be able to change or modify it at will;
4. It must be applicable to telegraph communications;
5. It must be portable, and should not require several persons to handle or operate;
6. Lastly, given the circumstances in which it is to be used, the system must be easy to use and should not be stressful to use or require its users to know and comply with a long list of rules.

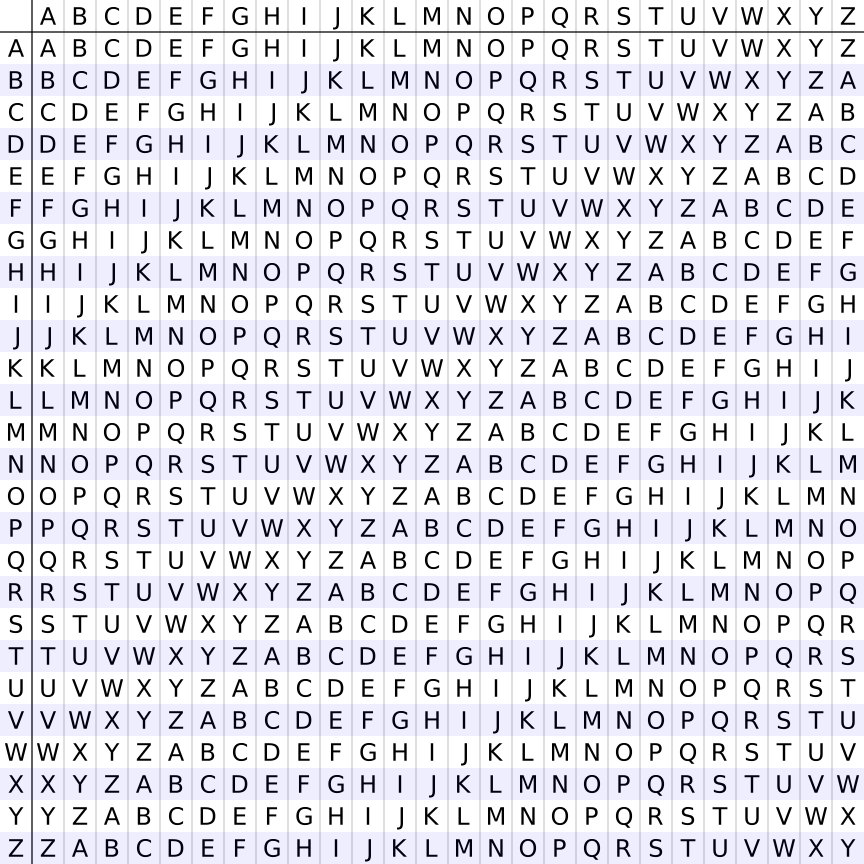
Some people might think that the strongest system is run through a series of different ciphers in order to produce the maximum jumbled-ness of a system. But this is a waste for two reasons. Firstly, any substitution cipher of that kind is going to fall victim to frequency analysis. Second, the more times you encrypt information, the more you are likely to make a transcription error. Likewise, the more complicated the decryption process is, the more likely your recipient is to make an error.

Kerckhoff tells us that the strongest system is one that is simple, but relies on a key that cannot be imitated to solve. Good luck!

# Section Eight: Vigenere’s Cipher

The next cipher resisted cracking for nearly 300 years and was known as the “unbreakable cipher” for much of that time. It is by far the most secure of any cipher in this booklet. It was used widely by the Confederacy in the American Civil War—Like some of the ciphers above, it has a keyword, and the Confederacy used “Complete Victory” as their keyword for much of the war.

The way it works is actually quite simple. You begin with a *tabula recta* of letter correspondences:



Then you write out your keyword as many times as you need to. Let’s use DARKKNIGHT again. This time, we are allowed to use double-letters.

Then we take our plaintext as well. Let’s encode the message:

With a clamor of bells that set the swallows soaring, the Festival of Summer came to the city Omelas, bright-towered by the sea.

Let’s line them up together:

D A R K K N I G H T D A R K K N I G H T D A R

W I T H A C L A M O R O F B E L L S T H A T S

Now, we line the top row with the rows in the table and the bottom row with the columns.

Lining D up with W gets Y, and so on.

D A R K K N I G H T D A R K K N I G H T D A R

W I T H A C L A M O R O F B E L L S T H A T S

Y \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

To decrypt it, you must know the code word, DARKKNIGHT. Then you can simply find the letter on the top row that corresponds with the letters in the middle generated by your ciphertext and the repeating code line.

Most of the cryptanalysis that is associated with this relies on finding the number of letters in the key word and breaking it down from there, but it’s beyond the scope of this booklet.

## Some Examples for Decryption

Example Seven: Affine Cipher, a=7 b=3

Dcc xm xcy. Qxgahqt fczf fufs. Fufs gshfy. Fufs mdhcfy. Qx jdggfs. Gsp dtdhq. Mdhc dtdhq. Mdhc kfggfs.

Example Eight: Unknown Cipher, Frequency Analysis

JBWP QUH X MXZZ RB KB UUGYAJ KB JPP YXJ EPUK-OGBMF YPUQ, KYP WXZQ EBQJ BN YXJ PHP-ZXQJ, YXJ EBXFKPQ JTXF VUE. XF KYP NZUK VBAFKGH FPUG OH WYPGP KYPH QAR YXW BAK, YXJ ZUJK RGAPZ BN MXFKPG JPPQJ VUTPQ XF YXJ JKBWUVY, NUTPQ PIVPEK NBG KYP VUE, FBBJP UFQ RXGQZP, X MXZZ JKUFQ U ZBFR KXWP.

Example Nine: Vigenere’s Cipher: Codeword CODEWORD

ACXXWYWWJSJSKRPRWHDEAHYHDOGCKIRGFHKIIIGDPRWLAFVBQIKERSKKG

# Section Nine: Steganography

Steganography refers to the practice of encoding your message in a way that makes it not obvious to the potential eavesdropper. All the codes that we have looked at so far are obviously codes. Steganography works differently: the principle here is “secrecy through obscurity” and the more obscured (hidden) your code is, the idea goes, the easier it is to get the message across without being discovered.

Many different methods for this exist, some of which are already familiar to you: using invisible ink (e.g., lemon juice or urine) to write a message on the back of a piece of paper that has an uninteresting message on it.

We are going to learn one more method of steganography, as well, attributed to the brilliant polymath and inventor of the scientific method: Francis Bacon[[7]](#footnote-7). In this cipher, the text itself is irrelevant (though for added complexity, you could apply this somehow to text that is already encrypted to create layers. Sneaky sneaky), but the way the text is presented is important.

The pattern we will apply is this:

a AAAAA g AABBA n ABBAA t BAABA

b AAAAB h AABBB o ABBAB u-v BAABB

c AAABA i-j ABAAA p ABBBA w BABAA

d AAABB k ABAAB q ABBBB x BABAB

e AABAA l ABABA r BAAAA y BABBA

f AABAB m ABABB s BAAAB z BABBB

so if we wanted to encode the message “flee!” into a piece of text, we might assign a significance to each letter so that non-bolded letters are the “A” of the pattern and bolded letters are the “B” of the pattern.

“Flee” would be AABAB ABABA AABAA AABAA, or non-bold non bold bold non bold bold non bold bold non bold bold non bold non bold non bold bold non bold non bold… you get the idea.

So maybe our dummy text says, “Bacon has been called the father of empiricism. His works argued for the possibility of scientific knowledge based only upon inductive and careful observation of events in nature. Most importantly, he argued this could be achieved by use of a skeptical and methodical approach whereby scientists aim to avoid misleading themselves.”

We would encode “random” bolded letters in it that would secretly create our Bacon Cipher in the text:

Ba**c**o**n** h**a**s **b**een **c**alle**d** the father of empiricism. His works argued for the possibility of scientific knowledge based only upon inductive and careful observation of events in nature. Most importantly, he argued this could be achieved by use of a skeptical and methodical approach whereby scientists aim to avoid misleading themselves.

With just a few short, easy to miss bold letters, we have encrypted instructions to our partner within a body of text that is seemingly irrelevant to the message. Bold letters aren’t the only way to incorporate Bacon’s Cipher, just one easy to overlook example. If you can find a more subtle way of making this pattern appear in a pile of dummy text, then you are safer. Unless, of course, the person who intercepts it is looking for a Bacon cipher. This brings up Kerckhoff’s main point: once the Bacon Cipher is widely known, it is useless to us because any cryptanalyst worth his paycheck will already know about it and check to see if it is being used. Steganography has its uses, perhaps, but we must look for ciphers that are still effective, even when the enemy knows exactly what kind of cipher is being used and when.

# Section Ten: P and NP

## Modern Crypto

So far, we’ve only been learning Classical Cryptography, which is what flourished until the development of more complicated machinery and computers in the industrial revolution and information age, respectively. For much of that time, the Vigenere Cipher was the most effective cipher and Steganography was also used often, though Kerckhoff did demonstrate in 1883 that this was unwise.

In the last 50-100 years, cryptography has advanced immensely and relied on a mathematical hypothesis that is under debate within circles of computer scientists and logicians to this day. There are two kinds of problems that can be solved. Firstly, those that can be solved via an algorithm, or a rigid set of rules which, when applied in the correct order, always generate the correct solution. Other kinds of problems have a correct or best solution, but they require heuristics (problem solving principles without 100% accuracy, like, “the simplest explanation is to be preferred”) and creative thinking. The first kind is very easy to program a computer to do, and they are referred to as P. The second kind is currently impossible to program a computer to do, because it requires a kind of intelligence that is too similar to a human brain’s intelligence. We simply cannot program computers to think like us yet, though we can program them to be scarily accurate at games like chess, which require a combination of algorithmic and creative thinking.

In contemporary times, cryptographers have relied on computers to encode and decode their information, and computers, as I have said, can solve P problems easily. The trick, then, is to incorporate an NP style problem into your cipher.

The most popular of these is to incorporate incredibly long (100 digits!) prime numbers into the encryption. It is very easy for a computer to check if a number is prime (though it takes a while) but it is impossible to program a computer to find a 100 digit prime simply by following an algorithm. The best way is still “guess and check” though the guesses are more sophisticated than starting with 100….001 and going from there[[8]](#footnote-8). If you can find a previously unknown prime number with that many digits, you can actually garner a pretty hefty price for the knowledge of its existence, as these primes are essential for modern cryptology, which is used from everything to NSA secrets to your bank account pin.

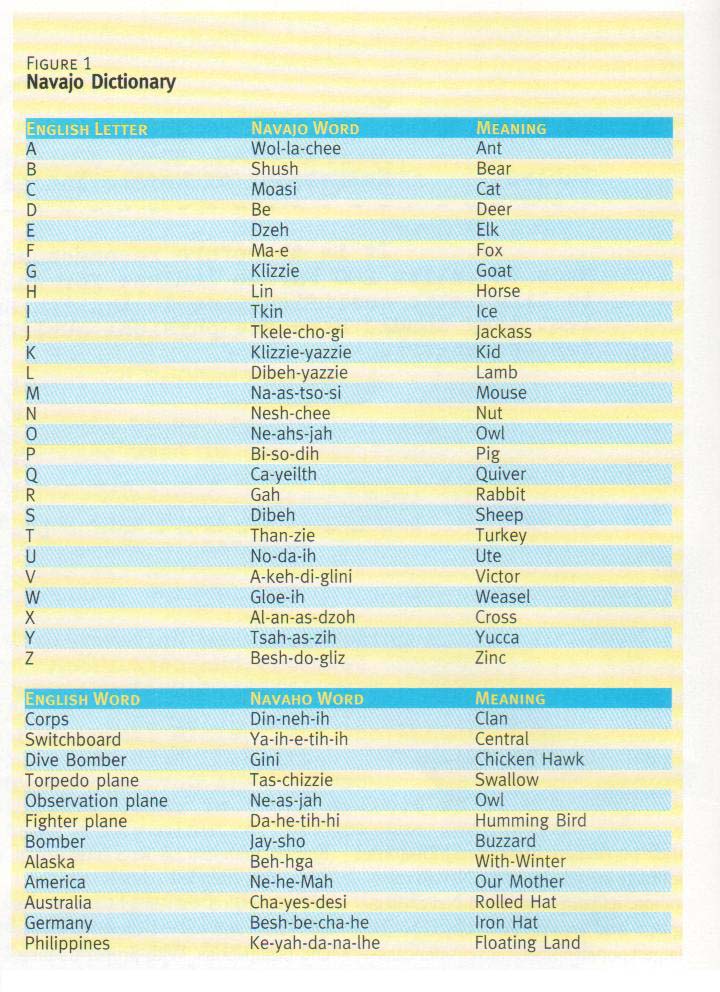
Notice that this process completely follows Kerckhoff’s rules: everyone knows how they work, but finding the one specific prime number that unlocks the entire code is basically impossible, and so our secrets remain safe. If we found a better way of finding prime numbers, one that really knocks the pants off of the Mersenne method, e.g., then our computer systems would become substantially less safe. Likewise, if we invent a computer system that is capable of creative thought to solve NP problems, which many computer scientists believe isn’t even possible[[9]](#footnote-9), then the same future awaits us.

# Section Eleven: Closing thoughts

## Navajo Code Talkers

During World War Two and other times during the 20th century, the United States military, specifically the Marine Corps, utilized several hundred bilingual soldiers from Native American Tribes to send messages, particularly in the Pacific Theatre of the war. The Japanese had a heck of a time trying to interpret the messages because no one outside of North America had ever learned those languages. Due to the existence of international linguistic scholarship and, you know, the internet, it is unlikely that this method would ever work today. However, the creation of a secret language for the transmission of certain ideas does hold some interest for some… mostly conlang weirdos, though.

Here is the general dictionary for the codetalkers from that era:



## Defeating Frequency Analysis

Rory comes out of her tent and sees everyone dressed in turn of the century clothes. She catches up to a couple.)

RORY: Hey. Rory Gilmore. Um, this is quite a soiree. Are all the Life and Death Brigade gatherings this elaborate? (They ignore her. She wanders over to a group of guys.)

GUY #1: How about (?) social stands?

GUY #3: Ridiculous. Total stand-still for all in his vicinity. What do you say?

GUY #4: I concur totally.

GUY #1: Crazy construct if you think for a bit.

GUY #2: Dubious logic if you ask this thoughtful guy.

RORY: Hello, everybody.

GUY #3: My God.

GUY #1: Shocking.

GUY #3: Silly girl. Not adjusting to this proud point of ours.

GUY #4: Sad, this diminishing vision.

RORY: Excuse me?

GUY #4: Full count is six, I say?

GUY #3: Six, no doubt. Ay, again I concur.

GUY #4: Point in fact, daft lady, to catch on would prompt our congratulations.

RORY: It’s a game?

GUY #3: At which you totally fail.

GUY #4: You want for instruction?

RORY: Apparently.

GUY #4: Said gap ‘twixt ‘d’ and ‘f’ shall not slip from lips in any word this group allows.

RORY: Said gap ‘twixt ‘d’ and ‘f’ … you’re not using the letter ‘e’?

GUY #4: Said this thing our group did banish.

GUY #1: Loud, for all to drink in!

GUY #3: Daft girl.

RORY: So, no one is supposed to say the letter ‘e’.

GUY #4: My God, this woman hounds us with this thing I banish.

GUY #3: Dumbfound.

RORY: Um, I’ll catch up with you guys later. Have fun. If that’s what you’re doing. (She walks away.)

GUY #4: Bloody horror, that woman.

GUY #1: Ostracism should occur, I think.

(Rory wanders through the camp.)

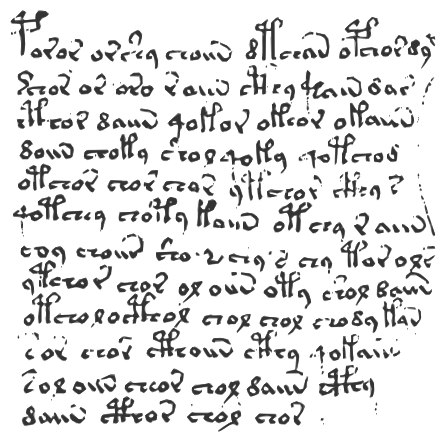
If said gap twixt ‘d’ and ‘f’ is in fact a most common part of our talking to an individual, it might show wisdom to avoid it at all costs. This would confound a man who wants to find our as now unknown information, would it not?

## Un-cracked codes

Just for fun, I’d like to describe two uncracked codes or potential nonsense—no one is really sure—that have interesting backgrounds.

### The Voynich Manuscript

The Voynich Manuscript is a rare book—there’s only one copy!—currently housed at Yale but with a long history of owners that might include King Rudolf II, no one is really sure. It was “discovered” by a fellow named Voynich, though, so it bears his name. The book is written in a language that has yet to be deciphered, though it does seem to match with statistical properties of real languages.



The book seems to be an ancient encyclopedia, based on the pictures which include topics of astronomy, botany, and mythology. No one knows for sure, though!

Recently, a team believes they have been able to decode one or two words that are proper nouns[[10]](#footnote-10), possibly in an eastern or near eastern language. I’ve included a few pictures of the text, for your general amusement.



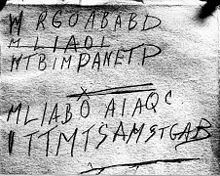
### Taman Shud

The Taman Shud case, which really ought to be the Tamam Shud case[[11]](#footnote-11), refers to a man who washed up on a beach in Australia, dead of unknown causes. His identity has never been determined. He was carrying a slip of paper with the words ‘tamam shud’ from a *Rubaiyat* poem, written by Omar Khayyam, which translates to “it is finished.” Because of the mysterious death and lack of identification and the involvement of a code (see below), and because this occurred right as the cold war was kicking off, many believe that the individual was a spy who was killed in some kind of covert spy-v-spy action.

Some investigators believe that he had a romantic connection with a local woman who was connected to the case and they point to genetic similarities between him and her children as evidence. The case started in 1948, however, so many of those who were involved at that time are deceased and any secrets they had went with them to their graves.

Because he had a page torn from a book of which there were relatively few copies, police requested a nationwide search, which turned up a copy that had a torn page wherein dead man’s torn page fit. It is unclear from contemporary reports when or where the book was discovered, but it was turned over to police sometime around seven months after the discovery of his body. Inside this copy of the book was some handwriting, with the following code.

No one has been able to decode it so far.

The person who does will solve a case that is at least 68 years old and may involve international spy intrigues or just one failing marriage. Maybe that person could be you?

1. Within the context of codes and ciphers, we’ll mostly be dealing with integers, rather than real numbers or rationals—this branch of math is called finite math or discrete math for that reason. If every letter of the alphabet is a number from 1-26, then 5.5 doesn’t really make sense, does it? [↑](#footnote-ref-1)
2. …in our alphabet, that is. Substitute your own number of letters if you’re doing it in another [↑](#footnote-ref-2)
3. That was because our starting letter, way back on the last page, was I, the ninth letter of the alphabet. Confused yet? [↑](#footnote-ref-3)
4. About 600,000. Seriously. Don’t try that. Although I suppose you could write a computer program that would try this and then only notify the user when results include an arbitrary number of ‘real’ words in the decrypted text, even coding that program is still a waste of your time as compared to what you’re about to learn. [↑](#footnote-ref-4)
5. Perhaps you can relate? [↑](#footnote-ref-5)
6. This is sometimes restated as “the enemy knows the system,” meaning that you should always assume that someone else will crack your *system* easily, even if they don’t know the right key to completely solve it. The opposite philosophy, ‘security through obscurity,’ will be discussed in chapter nine. [↑](#footnote-ref-6)
7. Knowledge is Power. France is Bacon. [↑](#footnote-ref-7)
8. For example, we tend to check for Mersenne primes, which follow a known pattern: 2p – 1 where p represents a known prime number. So 3 is a prime number and then 23-1=7, which is also prime. This doesn’t work for every prime number; however, it does substantially narrow down our list, which is why it’s used. [↑](#footnote-ref-8)
9. Specifically, 77% of computer science professors believe this will be impossible. See here: https://www.cs.umd.edu/~gasarch/papers/poll.pdf [↑](#footnote-ref-9)
10. The constellations Taurus and Pleiades, as well as the plant, centaurus. [↑](#footnote-ref-10)
11. A newspaper made a typo and now we’re stuck with it. [↑](#footnote-ref-11)