Exploration of the Mathematics of Historical and Modern Encryption Methods

Khalid N. Talakshi

Robert Bateman High School

Exploration of the Mathematics of Historical and Modern Encryption Methods

Since I was a kid, I always wanted to be a spy. Not like James Bond, but more like Q, who gave James Bond all the tech he needed to do his sleuthing and secret work. As I got older I began to develop an interest in programming and creating software. I also developed an interest in cryptography and secret codes. I wanted to learn more about how cryptography worked and its uses in software, since it is an essential part of data security. I learned that, over the course of my research, that cryptographic encoders and decoders can be built using simple mathematics, such as addition, modular arithmetic, and exponents, As such, in this mathematical exploration, I will be considering the world of cryptography and looking at different ways of encrypting and decrypting data.

My goal for this exploration is to explore the math of (different types of ciphers: Caesar Cipher and Multiplicative Cipher, which both share aspects with RSA encryption, a modern use of encryption. As I go through this exploration, I will be looking at not only the math behind these ciphers and encryption systems, but also the logic behind finding certain numbers used in encryption. I will also be explaining a lot of the math behind each of the ciphers as well. Finally, I will be looking at ways of breaking these ciphers using statistics and factorization.

The Basics of Encryption and Caesar Cipher

Encryption is just the conversion of data from one form to another. This allows the data to be kept secret during transmission and holding. Encryption is best explained in an example. Let’s take three people: Alice, Bob, and Eve. Alice wants to send a message to Bob without Eve knowing. Using encryption, Alice can send the message to Bob without Eve being able to understand it. Basic encryption consists of 3 parts: the plaintext, the cipher text, and the key. The plaintext is simply the message that Alice wants Bob to know. For example, the message could be:

the quick brown fox jumped over the lazy dog

It is common for the plaintext to be written in lowercase while the ciphertext to be written in Upper case to distinguish them. The key is the next part. The key is what converts the data from plaintext to ciphertext and back. In a simple shift cipher or Caesar Cipher, the key is simply the number of letters you want to shift the plaintext over by to get the cipher text. You do this by assigning each letter a numerical value. This is usually the order each letter appears in (i.e. A =1, B = 2, etc.). You then add the key value to the number and use the original placement to determine the ciphertext letter. In this example, Alice decides to shift the plaintext by three letters. We can write this out in the following way:

$$Key: X + 3\ (mod\ 26)$$

$$Key: X + K \ (mod\ m)$$

In the second example, it is a general expression of a key, where X represents the letter or plaintext number, K represents the shift and m represents the number of letters in the alphabet. The (mod 26) part at the end of the key stands for Modulo. This is an operator that acts as a wraparound for the key. If you were to shift the letter Z the same way as the other letters, you would get an out of range value, since there are only 26 letters in the Anglo-Saxon Alphabet. However, if you were to divide 29 by 26, you would get 1 with a remainder of 3. The modulo returns the remainder value, which will be used when determining the ciphertext. In this exploration, the symbol ≡ will be used to represent the remainder.

$$29 \div 26 = 1\ R\ 3$$
$$29 \ \div \ 26 \equiv 3$$

In this case, z would be enciphered to C, since C is the 3rd letter in the alphabet. The final part is the ciphertext. This is simply the text that has been changed to make the data secured. In the shift/Caesar cipher, this is simply the product of each conversion of the original message to the shifted message. In this case, the message would be

WKH TXLFN EURZQ IRA MXPSHG RYHU WKH ODCB GRJ

To the naked eye, this would seem indecipherable, incoherent, and incorrect, but, it is a simple message encrypted.

Breaking it Down

While encryption helps keep messages protected, it doesn't mean it always works. Keeping with the example above, let's say Eve somehow gets her hands on the message. While it is encrypted, there are a few ways that she can try and crack it. For a small message like this, the first way would be a brute force crack. This is when Eve try's every possible cipher to get a coherent message. If she understood that this was a shift/Caesar cipher, she would most likely be able to try all the possible ciphers, since there are only 25 possible options. Eventually she would be able to crack it. However, as the messages get bigger the brute force approach gets complicated and long. A longer message could have issues which may trick Eve, such as improper grammar or mathematical errors in the shift. It also becomes time consuming as more letters need to be checked for a correct shift. Another way would be to use a histogram and a statistical analysis of letter frequency, and compare it to theoretical probabilities of letter frequency. To show how this would work, we will use a large quote from Cory Doctorow's *Little Brother*:

> "Governments are instituted among men, deriving their just powers from the consent of the governed, that whenever any form of government becomes destructive of

these ends, it is the right of the people to alter or abolish it, and to institute new government, laying its foundation on such principles, and organizing its powers in such form, as to them shall seem most likely to effect their safety and happiness." I remembered it word for word.

The encrypted text for a cipher shift of 3 would be:

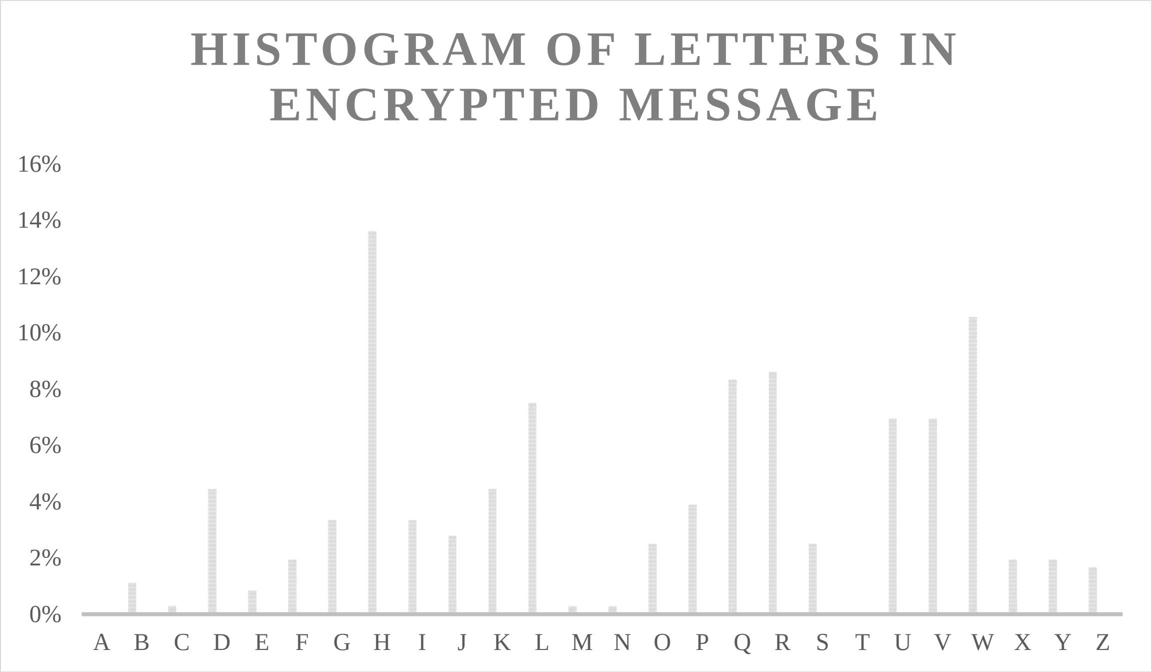
"JRYHUQPHQWV DUH LQVWLWXWHG DPRQJ PHQ, GHULYLQJ WKHLU MXVW SRZHUV IURP WKH FRQVHQW RI WKH JRYHUQHG, WKDW ZKHQHYHU DQB IRUP RI JRYHUQPHQW EHFRPHV GHVWUXFWLYH RI WKHVH HQGV, LW LV WKH ULJKW RI WKH SHRSOH WR DOWHU RU DEROLVK LW, DQG WR LQVWLWXWH QHZ JRYHUQPHQW, ODBLQJ LWV IRXQGDWLRQ RQ VXFK SULQFLSOHV, DQG RUJDQLCLQJ LWV SRZHUV LQ VXFK IRUP, DV WR WKHP VKDOO VHHP PRVW OLNHOB WR HIIHFW WKHLU VDIHWB DQG KDSSLQHVV." L UHPHPEHUHG LW ZRUG IRU ZRUG.

For this method, Eve would take a tally of the encrypted letters and how many times they appear. Rather than just writing down the frequency, it is better to write the relative frequency in order to compare theoretical data. Below, the table shows them sorted.

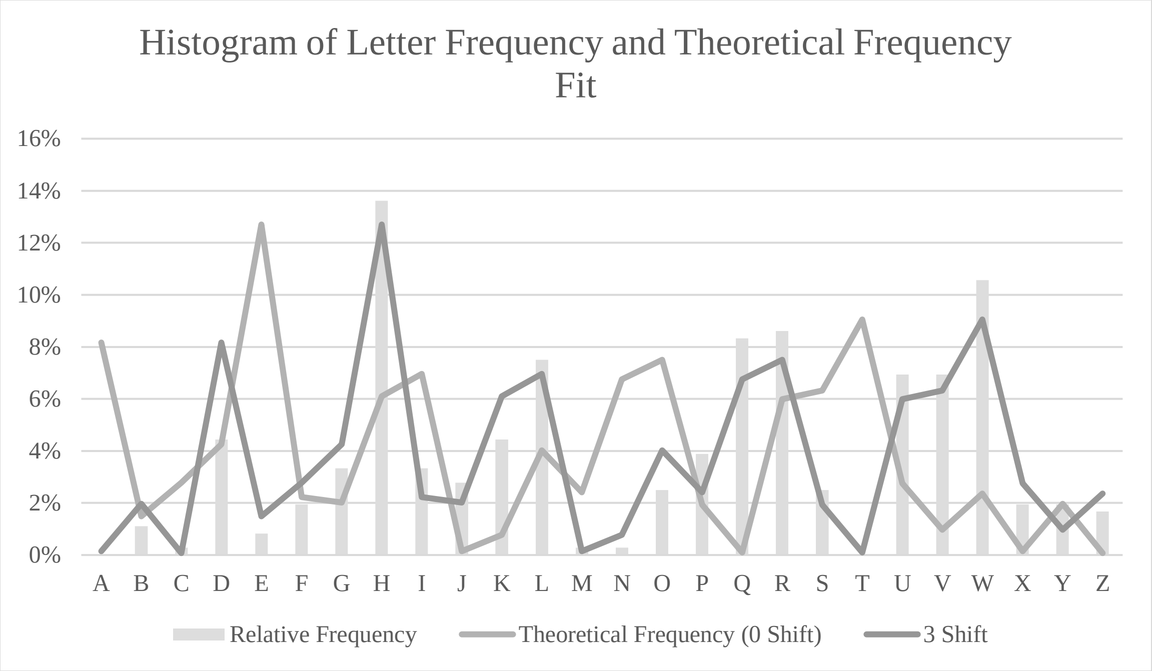
Table 1
Letter Histogram

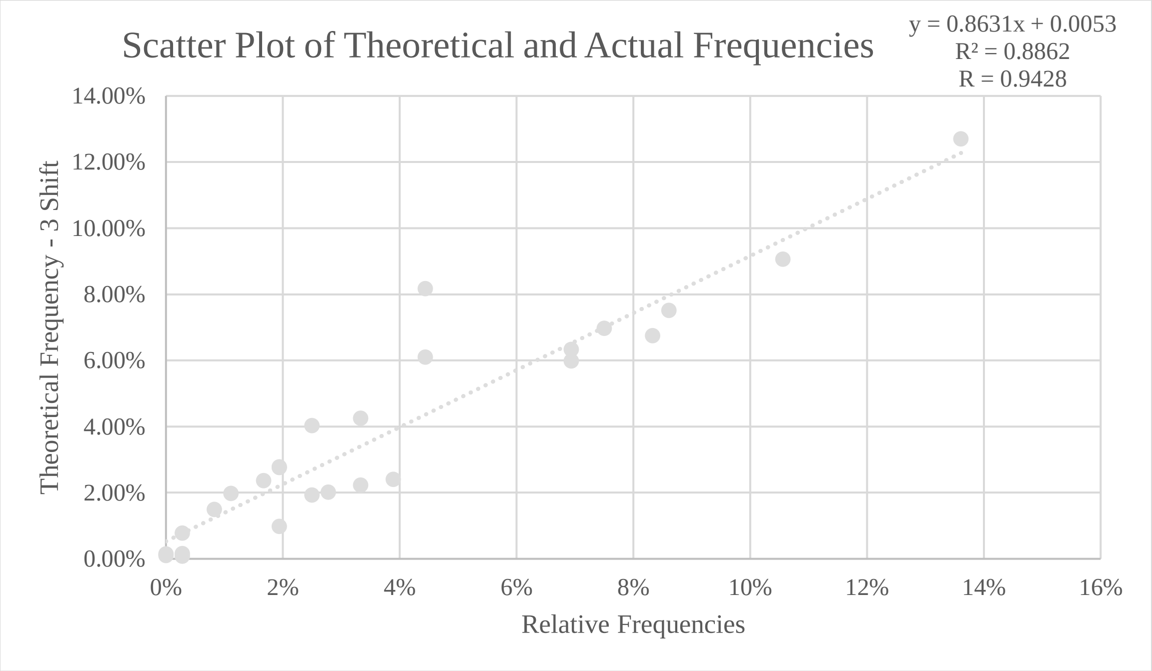
Letter (Ciphertext)	Occurrence	Relative Frequency (Occurrence/Character Count)
H	49	14%
W	38	11%
R	31	9%
Q	30	8%
L	27	8%
U	25	7%
V	25	7%
D	16	4%
K	16	4%
P	14	4%
G	12	3%
I	12	3%
J	10	3%
O	9	3%
S	8	3%
F	7	2%
X	7	2%

Letter (Ciphertext)	Occurrence	Relative Frequency (Occurrence/Character Count)
Y	7	2%
Z	6	2%
B	4	1%
E	3	1%
C	1	0%
M	1	0%
N	1	0%
A	0	0%
T	0	0%
Total	360	100 %



In order to determine the shift, we can apply theoretical data over top of the existing data. This should show us a similar shape, excepted shifted over some length. The length to which it needs to be shifted will give us the shift factor, and therefore the key to decrypt it. Below we can see the data above, as well as a few test fits to see which best matches the data.





Above are 2 graphs. The first shows the relative frequency of the ciphertext, the theoretical frequency of each letter, and 3 letters shifted, which I determined to be the best fit for the data based on the difference between the highest theoretical frequency and the largest relative frequency. To prove this is the best fit, I created a scatter plot with relative frequency on the xaxis and the theoretical, 3 Shift frequency on the y-axis. From this we found a general trendline and the Pearson coefficient value:

Table 2

Calculation Table of Pearson r coefficient

Data	Actual Value (X)	Theoretical Value (Y)	x (X – mean)	y (Y – mean)	x^2	y^2	xy
	0%	4.25%	-4%	-3.700%	0.15%	0.14%	0.14%
	1.11%	12.70%	-2.74%	-1.876%	0.08%	0.04%	0.05%
	0.28%	2.23%	-3.57%	-3.776%	0.13%	0.14%	0.13%
	4.44%	2.02%	0.59%	4.317%	0.00%	0.19%	0.03%
	0.83%	6.09%	-3.02%	-2.358%	0.09%	0.06%	0.07%
	1.94%	6.97%	-1.91%	-1.068%	0.04%	0.01%	0.02%
	3.33%	0.15%	-0.52%	0.403%	0.00%	0.00%	0.00%
	13.61%	0.77%	9.76%	8.852%	0.95%	0.78%	0.86%
	3.33%	4.03%	-0.52%	-1.622%	0.00%	0.03%	0.01%
	2.78%	2.41%	-1.07%	-1.835%	0.01%	0.03%	0.02%
	4.44%	6.75%	0.59%	2.244%	0.00%	0.05%	0.01%
	7.50%	7.51%	3.65%	3.116%	0.13%	0.10%	0.11%
	0.28%	1.93%	-3.57%	-3.697%	0.13%	0.14%	0.13%
	0.28%	0.10%	-3.57%	-3.078%	0.13%	0.09%	0.11%
	2.50%	5.99%	-1.35%	0.175%	0.02%	0.00%	0.00%
	3.89%	6.33%	0.04%	-1.444%	0.00%	0.02%	0.00%
	8.33%	9.06%	4.48%	2.899%	0.20%	0.08%	0.13%
	8.61%	2.76%	4.76%	3.657%	0.23%	0.13%	0.17%
	2.50%	0.98%	-1.35%	-1.921%	0.02%	0.04%	0.03%
	0%	2.36%	-4%	-3.755%	0.15%	0.14%	0.14%
	6.94%	4.25%	3.09%	2.137%	0.10%	0.05%	0.07%
	6.94%	12.70%	3.09%	2.477%	0.10%	0.06%	0.08%
	10.56%	2.23%	6.71%	5.206%	0.45%	0.27%	0.35%
	1.94%	2.02%	-1.91%	-1.092%	0.04%	0.01%	0.02%
	1.94%	6.09%	-1.91%	-2.872%	0.04%	0.08%	0.05%
	1.67%	6.97%	-2.18%	-1.490%	0.05%	0.02%	0.03%
Total			0%	0%	3.22%	2.70%	2.78%
Mean	3.85%	3.85%	0%	0%			

$$r = \frac{\sum xy}{\sqrt{\sum x^2 \sum y^2}}$$

$$= \frac{2.78\%}{\sqrt{3.22\% \times 2.70\%}}$$

$$
\begin{aligned}
&= \frac{2.78\%}{\sqrt{8.69\%}} \\
&= \frac{2.78\%}{2.95\%} \\
&= 0.9428
\end{aligned}
$$

Based on the Pearson coefficient value, we see that there is a strong correlation between our experimental relative frequency and our theoretical 3 shift frequency. As such, we can determine that the 3 shift cipher is most likely the cipher we can use to decrypt the message.

Another way is using a word histogram. Similarly, a word histogram would allow you to do the same thing as a letter histogram. The most common word in the English language is “the”. In the encrypted message, the letter combination “WKH” appears twice, more than any other letter combination. If we assume that this letter combination is “the”, then W would convert to T, K would convert to H, and H would convert to E. These are all 3-letter shifts, which indicates that not only is the word “the”, but the message is encrypted using a 3-letter shift. The best way around this would be to group the letters into smaller packs. This would make it hard to decrypt a message based on a word histogram. Unlike the brute force attack, Histograms work well when there is a large message, as this gives you more data to work with. While a Caesar cipher is a good way of protecting data, this is a weak cipher due to the limited number of possible combinations and the simple replacement of letters rather than rearrangement or replacement of alphanumeric symbols with other symbols. However, we can easily increase the security through the use of Multiplicative ciphers.

Increasing Security: Making and Breaking Multiplicative Ciphers

A simple shift/substitution cipher can easily be broken with simple probability or trial and error. However, what if we multiplied instead of added to the value representing the letter? This begins with multiplicative ciphers. Like Caesar ciphers, they are also based in the same key, where letters are assigned numbers, similarly to that based of their position.

Letter	**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**
Number	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	26

Rather than simply adding a key value to their number property, we multiply it by a number. However, there is an issue with the number of keys we can have. To start, we are limited to 26, since there are only 26 letters, so key 27 is the same as key 1. Another thing we must consider is that if we were to multiply by 2, we would get letters being represented by the same number. For example, both ‘B’ and ‘O’ would be represented by the number 4 if we had a key of 2. This eliminates half the keys that are a multiple of 2. If we were to multiply by 26, we would get a bad key as it would be the same as multiplying by 0, which would represent every letter the same. Finally, 13 is bad since each letter would be represented by a possible 1 of 2 letters, as 13 wraps around 26 in 2 possible ways. So, we are limited to 12 keys when using a multiplicative cipher. Now let's see this in action. For this sample, we will use the same excerpt from the Caesar cipher:

> "Governments are instituted among men, deriving their just powers from the consent of the governed, that whenever any form of government becomes destructive of these ends, it is the right of the people to alter or abolish it, and to institute new government, laying its foundation on such principles, and organizing its powers in such form, as to them shall seem most likely to effect their safety and happiness." I remembered it word for word.

It is also important to note the use of prime numbers in multiplicative ciphers. The use of prime numbers helps increase security for breaking into these ciphers. For this example, we will use 7 as our key, and encrypt our alphabet as such:

Plain Text	**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**
Number	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	26
Key 7 (mod 26)	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	26
Cipher Text	G	N	U	B	I	P	W	D	K	R	Y	F	M	T	A	H	O	V	C	J	Q	X	E	L	S	Z

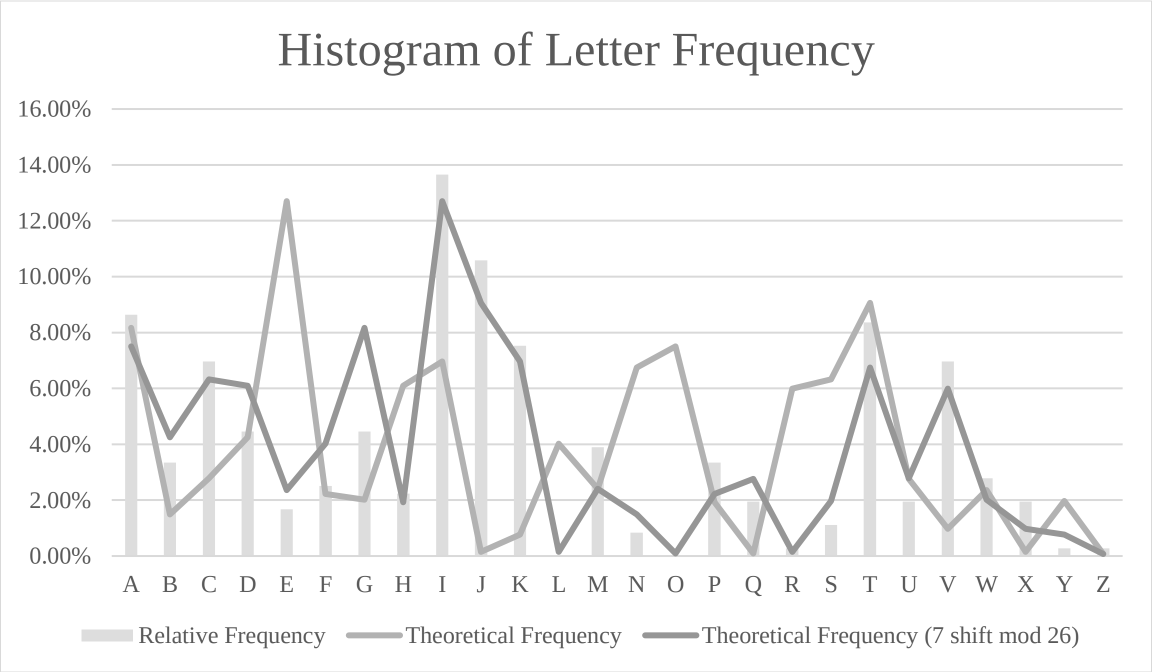
This means that our encrypted message will appear as such (note that the punctuation is missing):

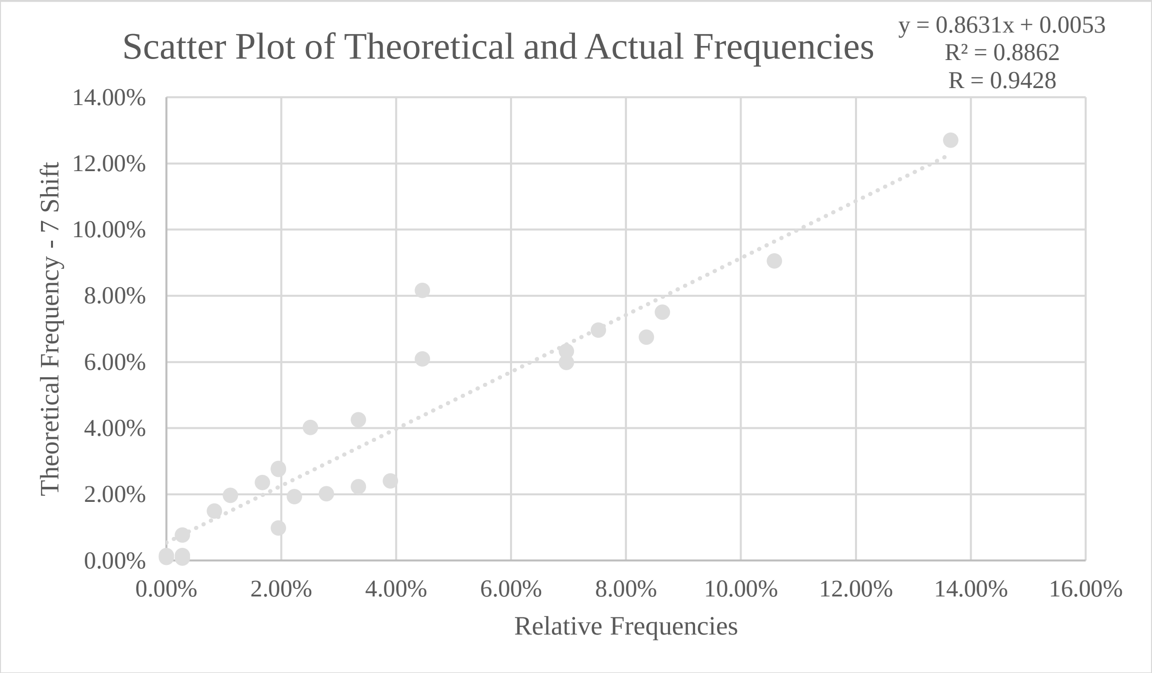
> WAXIVTMITJC GVI KTCJKJQJIB GMATW MIT BIVKXKTW JDIKV RQCJ HAEIVC PVAM JDI UATCITJ AP JDI WAXIVTIB JDGJ EDITIXIV GTS PAVM AP WAXIVTMITJ NIUAMIC BICJVQUJKXI AP JDICI ITBC KJ KC JDI VKWDJ AP JDI HIAHFI JA GFJIV AV GNAFKCD KJ GTB JA KTCJKJQJI TIE WAXIVTMITJ FGSKTW KJC PAQTBGJKAT AT CQUD HVKTUKHFIC GTB AVWGTKZKTW KJC HAEIVC KT CQUD PAVM GC JA JDIM CDGFF CIIM MACJ FKYIFS JA IPPIUJ JDIKV CGPIJS GTB DGHHKTICC K VIMIMNIVIB KJ EAVB PAV EAVB

While we may be losing keys, going from 25 possible in the Caesar cipher to just 12 in a multiplicative cipher, we gain security as the numbers are not encrypted in sequential order. Let’s look at a histogram of the message above:

Table 3
Letter Histogram

Letter (Ciphertext)	Occurrence	Relative Frequency
A	31	8.64%
B	12	3.34%
C	25	6.96%
D	16	4.46%
E	6	1.67%
F	9	2.51%
G	16	4.46%
H	8	2.23%
I	49	13.65%
J	38	10.58%
K	27	7.52%
L	0	0.00%
M	14	3.90%
N	3	0.84%
O	0	0.00%
P	12	3.34%
Q	7	1.95%
R	1	0.28%
S	4	1.11%
T	30	8.36%
U	7	1.95%
V	25	6.96%
W	10	2.79%
X	7	1.95%
Y	1	0.28%
Z	1	0.28%
Total	359	100%





Outlined above are 2 graphs. The first shows the frequency relative to the total character count. It also shows the theoretical frequency in their plaintext, as well as a 7 mod 26 shift. Based on relative frequency, I found this by matching the most frequent with the theoretical most frequent, the second most together and so on. Eventually, I formed a pattern of a multiplication factor of 7, thus giving me the key. The second graph shows the correlation between the actual and the theoretical and actual frequencies. As you can see, the graph is identical to the Caesar cipher graph. This is to be expected since the phrase used was the same, and the shift resulted in the right key. This also means there is a strong correlation between the frequencies, giving me a clear indication as to what the key is. There is another way to find the key without going through the long process of determining correlation. From this histogram, if we assume ‘I’, the letter with the highest frequency, is the cipher text of the letter ‘e’, and ‘G’, the letter with the second highest frequency, is the cipher text of the letter ‘a’, then we can try and find the key using the greatest common divisor. Let’s assign I the number 9, which is its position in the alphabet and give G the number 7 based on its position. Based on these numbers, we figure can figure out the GCD using Euler’s algorithm. Euler’s algorithm states that the GDC of two numbers is equal to the GCD of the lesser of the two numbers and the remainder between the original two numbers. To find the GCD, you first start by dividing the larger number by the smaller number. If it returns a remainder, you then divide the smaller of the original numbers by the remainder. You keep doing this until you get no remainder. For this example, you can see this algorithm in work down below:

$$9 = 7 * 1 + 2$$
$$2 = 2 * 1 + 0$$

This shows that the greatest common divisor between 9 and 7 is 1. However, if the key was 1, the message would be the same in plaintext and ciphertext. As such we need to look at other options. Since encryption in our alphabet has a wraparound at 26, we can try and find a common divisor with 9 + 26, which is 35. If we try and find the greatest common divisor now, we will get a new number:

$$35 = 7 * 5 + 0$$

Since 7 is greater then 5, we can determine the greatest common divisor between 7 and 35 is 7, and since 35 mod 26 is 9, which represents ‘I’. This means our key is 7, and we can decrypt each letter by adding 26 to its positional number until it reaches a multiple of 7, and then divide that number by 7 to get the plaintext positional number, which will in turn give us a letter in plaintext. The brute force attack works just as well for Caesar Cipher. Since letters aren’t encrypted in sequential order, it is harder to find the key. However, since there are less possible keys, it evens out. Finally, you can also use a word histogram like the Caesar cipher, but this can also be easily reduced by simply grouping the letters together.

Importance of Basic Ciphers

The idea behind these ciphers is not for practical use. This is most likely attributed to the lack of advancements in technology. At the time of the creation of these ciphers, we see that the technology available at the time was not capable of brute forcing, or using histograms to determine letter frequency. However, as technology advances, the ability to decrypt ciphers has become more efficient. As such, in todays time these ciphers can be easily decrypted through methods such as brute force and histograms. The importance of these ciphers not only show the potential for other ciphers, but are the building blocks of bigger, more secure, and more practical ciphers such as enigma, one-time pads, and RSA encryption, which have been used in communications, security, and privacy.

Modern Applications of Encryption: RSA Encryption and the Public Key System

RSA is an example of a public key cipher, which uses a public key and a private key to encrypt and decrypt your data. It is also a type of exponential cipher, where the plaintext number is raised to some power, and is wrapped around some integer, usually larger than the number of letters in the alphabet for which the cipher is being used. RSA uses a compound modulo, which is found based on a set of numbers. It all starts with two prime numbers, p and q. These numbers are large, commonly 300 or more digits long. For this example, we will use 2 small, 2-digit prime numbers:

$$p = 37$$
$$q = 17$$

We then use these numbers to make *n*, the modulo or wraparound for this encryption.

$$\begin{aligned} n &= p \times q \\ &= 37 \ \times 17 \\ &= 629 \end{aligned}$$

We also use *p* and *q* to make *Φ(n),* which represents Euler's totient function. This function finds the number of integers for which they are relatively prime to *n,* or in other words share a GCD of 1. Since we know *n* is a prime number and is the product of *p* and *q*, we can use the following function to find *Φ(n).*

$$\Phi(n) = (p-1)(q-1)$$

Inputting *p* and *q* give us the following:

$$\begin{aligned} \Phi &= (p-1)(q-1) \\ &= (37-1)(17-1) \\ &= (36)(16) \\ &= 576 \end{aligned}$$

Φ(n) is used to find the public key (represented by *d*) and private key (represented by *e*). We now have *n* and *Φ(n)*. With this we can now create the private key. The private key is just a number which needs to be prime relative to *Φ,* or otherwise have a GCD of 1 with *Φ(n).* We can use prime factorization to find a prime that meets this criterion:

$$\mathbf{576 = 2^6 \times 3^2}$$

Since 576 when prime factorized has prime factors of 2 and 3, we can multiply any number of prime numbers other than 2 and 3 to find a private key. For this example, we will use 5 as our private key. Now it is time to find a public key. This key must be found such that the public key is the inverse multiple of the private key mod *Φ*.

$$d = \bar{e} \left(mod \ \Phi(\mathrm{n})\right)$$
$$e \ \times d \ \equiv 1 \left(mod \ \Phi(\mathrm{n})\right)$$

To find this, we simply multiply *d* and *e* and divide by *Φ(n)* such that it has a remainder of 1. This is where it becomes difficult, as we need some logic operators from predicate calculus. We need to create a formula that checks all the values between 0 and *Φ(n) -1* to see if any share a GCD of 1 with e. We will set K_x to represent "x is a key", and x is a placeholder in this. As such, we can create the following formula:

$$\exists K\ (\forall(0, \Phi(n) - 1)(e * d \equiv 1\ mod\ \Phi(n) \supset K_d))$$

The formula above translates to "for all values in range of 0 and *Φ(n) -1*, if *e* times *d* mod *Φ(n)* is 1, then *d* is a possible key." In this example, the equation would return the value 461. Now we have *n, Φ(n), e,* and *d*, we can begin our encryption. To begin you need to publish the public key (e, n). You can now create your private key (d, n). To send a message, you first need to assign a value to each letter. Let us use the phrase "Heroes Never Die". You then have to raise the number to the power of your public key, mod n. This will return another value, which is the cipher text number.

Letters	H	E	R	O	S	N	E	V	E	R	D	I	E
Plaintext Number	8	5	18	15	19	14	5	22	5	18	4	9	5
Ciphertext Number (raised to the power of *e)*	60	609	52	172	355	29	609	235	609	52	395	552	609

So, the final message is:

60 609 52 172 355 29 609 235 609 52 395 552 609

To decrypt it, you use your private key (d, n). You must raise your ciphertext numbers to the power of d mod n.

Breaking RSA encryption is harder than Multiplicative ciphers and Caesar ciphers. To break this we need to use Shor's algorithm, which involves some quantum computing. To do this you need to find the period of the function f(x) = m^x mod N, where m is a random integer less then N, N is the number you are trying to factor (in this case *n* from above). You then use the quantum computer to create a quantum superposition and perform the quantum Fourier transformation. As such, people are limited to breaking it without access to a quantum computer. In this IA we have talked about using histograms to help decrypt the messages. However, RSA renders this tactic useless with the private key. Since we know the public key, and only the private key can decrypt it, a histogram would give us no new or useful information, thus not working. While RSA seems strong, there is one issue: it is a one-way system. Let's say Alice wanted to send a message to Bob, she would encrypt the message with Bob's public key and send it to him for him to decrypt it using his private key. However, if he wanted to respond to that message, he can't just encrypt using his private key because anyone can just decrypt it using his public key. The simple solution is for Alice to create a key of her own and publish it. While not inconvenient, it just means that to encrypt they would have to have 2 sets of keys. It is also common to mix AES encryption with RSA which reduces the system to 1 key. In this hybrid system, Alice would create an AES key, and encrypt it using Bob's public key. She would then send this key to Bob, who would decrypt the message to get Alice's AES key. Finally, they would encrypt and decrypt their messages using the AES key. While RSA is secure and more complex than other systems, it is important to note that any encryption becomes weak over time. With many threats to privacy and security at home and abroad, it's important to note that government agencies, like the NSA, have tried to include weaknesses in mainstream encryption systems for them to access secure data. As such we dive into the realm of ethics and morality when debating the major issue of cryptography today.

Conclusion and Reflection

With our lives being heavily oriented in the digital realm, it is important that we understand what encryption is and how it works. It protects all our data and helps keep us safe from identity theft and espionage. More importantly, it gives us privacy, allowing us to not only be expressive with less repercussions, but help us protect our rights as citizens in a time where these rights are under attack from various angles. We see this in the FBI vs Apple issue, where the FBI has asked Apple to make a code to unlock the iPhone of a terrorist. Apple declined as this code could then be used to open other iPhones, which will then weaken the security for which Apple prides themselves on. If it was not for encryption, these issues would be less common, as many authority figures could easily unlock them.

As I was writing my IA, I got to see a whole new side of encryption. I always loved the secrets and the mysteries behind it. I also loved the software and computer side of it. However, I never had the opportunity to really delve in to the math part of encryption. I always knew there was some aspect of math behind it, but I always believed it was only possible through calculus and university level mathematics. Yet as I explored these encryption systems, it amazed me how I was exploring these concepts that were vaguely familiar to me were so simple in mathematics. I also got to explore new areas of mathematics, such as statistical analysis and predicate calculus. The idea of using a correlation analysis to determine how likely that this shift is the cipher made me feel as though I was a spy out of James Bond.

The question now becomes "Where do I go from here?" As a software designer, data security is a big part of my job, but also is a responsibility as someone who is developing tools for the world. Being able to understand how encryption not only works on the software side, but also the mathematics side gives me a true edge when trying to secure data. I also learned throughout my IA that there are flaws in every encryption method, whether they be a limited amount of keys, sequential encryption or even government intervention. However, we can limit the influence of these issues in security by looking at new mathematical ways of encrypting data. Going forward, I will continue to learn more about encryption, and would like to specialize in encryption software as well as encryption in software. While I may not be the spy I always dreamed of, I still feel that I truly learned more about mathematical influence in our lives.

References

Carvell, T., Gurewitch, D., & Oliver, J. (2016, March 13). Encryption. *Last Week Tonight*. USA.

Holden, J. (2017). *The Mathematics of Secrets.* Princeton: Princeton University Press.

Khan Academy. (n.d.). *The Euclidian Algorithm*. Retrieved from Khan Academy: https://www.khanacademy.org/computing/computer-science/cryptography/modarithmetic/a/the-euclidean-algorithm

Kharpal, A. (2016, March 29). *Apple vs FBI: All you need to know*. Retrieved from CNBC: https://www.cnbc.com/2016/03/29/apple-vs-fbi-all-you-need-to-know.html

Marchenkova, A. (2015, August 13). *Break RSA encryption with this one weird trick*. Retrieved from Medium: https://medium.com/quantum-bits/break-rsa-encryption-with-this-one-weird-trick-d955e3394870

Simpson, S. G. (1999, April 30). *Logic and Mathematics*. Retrieved from Pennsylvania State University: http://www.personal.psu.edu/t20/papers/philmath/philmath.html#SECTION00022000000000000000

Weisstein, E. W. (n.d.). *RSA Encryption*. Retrieved from Mathworld--A Wolfram Web Resource: http://mathworld.wolfram.com/RSAEncryption.html

Weisstein, E. W. (n.d.). *Totient Function*. Retrieved from MathWorld--A Wolfram Web Resource.: http://mathworld.wolfram.com/TotientFunction.html