Cryptography: Hill Ciphers

Aadi Anand, Alex Lipson

June 2024

1 Introduction

As our world turns toward technological revolution after revolution, it is easy to see only the good in such changes. However, an unfortunate truth about progress is that it progresses everything, not all of which is positive. Thus, with technology grew the need for security, which incubated the increasingly intricate and beautiful field of cryptography. In this text, we begin with an analysis of the historical development of ciphers, in order ultimately to provide a detailed description of the mathematical intricacies contained in the Hill Cipher, the first major cryptographic method that employed linear algebra in its design.

2 A Brief History of Cryptography

Cryptography is an age old practice, dating back at least to the time of Julius Caesar (and likely millennia before him), after whom the "Caesar cipher" is named. The Caesar cipher, however, is incredibly elementary and as cryptographic necessity grew, such trivial methods fell apart and were replaced time and again, ultimately giving way to infamous ciphers like the Enigma, DES, and AES, the latter of which is now considered near perfect. Every cipher builds largely on the practices before it, and is almost an organic being of its own, whose design and development is reliant on resistance to the attacks that its ancestors have weathered. The internals of AES are beyond the scope of this paper, but broadly it relies on the very same building blocks as Hill Ciphers. Blocks such as those put forth by the Caesar cipher, and more abstract substitution ciphers.

2.1 Components of a Cipher

Before we get into a few elementary ciphers, it is important to recognize the components that all ciphers share, and what they aim to accomplish. Imagine two parties that want to communicate over an insecure channel, say Alice wants to send a message to Bob. Now, the channel is called "insecure," because anyone can access anything sent over it, so if Alice just sends the message directly to Bob, it can be intercepted easily by an eavesdropper, Eve. This is where the cipher comes in. Alice can encipher the message on her end, generating "ciphertext" from "plaintext," and send the ciphertext over the insecure channel instead. This way, even if Eve gets access to the transmitted ciphertext, there is no way to extract the original plaintext message from it. Now, in order for this to work, Bob has to be able to decipher the ciphertext, while Eve cannot. This difference in ability is due to the "key," some information that Alice and Bob have shared securely before, that Eve does not have access to. As an aside, this may seem silly, as if Alice and Bob had a way to share something securely, they could have just shared the message, but in practice secure channels are more costly or place constraints on communication - for example, communicating in person is typically secure, but that means you have to be close to one another, and cannot communicate over long distances. Most modern communication is over insecure channels, like the internet. Now, the sign of a good cipher is that it is very difficult to decipher the ciphertext without the key, and that it is deterministic, so that with the key, you can extract the plaintext perfectly. In every cipher we examine, the size and scope of the key is directly related to the security of the cipher.

2.2 Caesar Cipher

The Caesar ciphers, or shift cipher, is a subclass of substitution ciphers, where the plaintext is made up (usually) of the twenty-six characters in the English (or in Caesar's time, Latin) alphabet. The characters are placed in the usual order, and the enciphering step simply replaces every character in the plaintext with the character obtained by shifting a number of characters to the right (with wrap-around if you were to go past the end). The number by which to shift is the key. For example, with a key of 3, as Caesar typically used, the message "ETTUBRUTEXOXO" would be enciphered using the tabular mapping:

A	D
B	E
C	F
D	G
$\vdots$	$\vdots$
W	Z
X	A
Y	B
Z	C

Notice how at the end, the characters wrap around, with "X" going back to "A", etc. This is a feature of modular arithmetic, which is often used to ensure that our encoded values stay within the same domain of values we see in our plaintext. In a nutshell, modular arithmetic is the math we are used to doing with a clock - 6 hours after 20:00 isn't 26:00, it's 2:00 (modulo 24 hours). In general, addition, subtraction, and multiplication can be accomplished modulo m, by evaluating the remainder of any expression, after you divide by m. In the case of the Caesar cipher, we can assign to A through Z, the numbers 0 through 25, and consider our Caesar shifts modulo 26. This way, we see that with a key of 3, we have that "X", which is assigned the number 23 maps not to 26 (which has no analogous character in our mapping) but rather to the remainder obtained by dividing 26 by 26, which is of course 0, or in our mapping, "A". Modular arithmetic comes in handy frequently in cryptography.

Now, lets encipher our message, "ETTUBRUTEXOXO", by performing our shifts. This of course results in "HWWXEUXHARAR", which is easily deciphered by Brutus (Bob), who knows that the key was 3, and so can perform the same shift, but backwards, yielding "ETTUBRUTEXOXO" back again. The ciphertext in this case looks pretty indecipherable without the key, but a major issue with the shift cipher with such a small alphabet is that there are really only 26 keys, corresponding to shifts by 0 to 25 characters. With such a small "key space", any attacker who intercepts the message can simply try all 26 keys and can pretty quickly narrow down which of the 26 messages makes the most sense, after which our code is totally shattered, because the key has been discovered. This is a small enough key space to where it's explorable even by hand, but with computers nowadays, it could be cracked in fractions of a second easily. The Caesar cipher is fundamentally insecure, as it is vulnerable to the "brute force attack," which simply tries all possible keys. For a small enough key space, the brute force attack is feasible, and so it is a necessary condition for any "good" cipher that the key space be sufficiently large. Nowadays this tends to mean of the order of magnitude of 2^{128} possible keys, but in this paper we'll settle for more than a few hundred thousand.

2.3 Substitution Cipher

Expanding on the relatively trivial Caesar cipher, it is natural to consider what might happen if we make things a little more sophisticated by taking an arbitrary, one-to-one reassignment of the characters. Note that the assignment has to be one-to-one, or we won't be able to decipher deterministically - all enciphering functions need to be invertible, and thus one-to-one. A key now becomes something that can capture this reassignment, for example any permutation of the alphabet where the ith character in the reassignment is the character to which the ith letter of the alphabet is reassigned. From such a key, we could construct a table like the one above, that we used for the Caesar cipher. And from this table, we could easily encrypt any plaintext of arbitrary length. Deciphering is accomplished by reversing the table, again pretty much what we did for the Caesar cipher.

The difference comes when we try to brute force this cipher, as the key space is not 26, but rather the number of possible permutations of 26 elements, 26 factorial (we have 26 choices of what character to map A to, then 25 remaining ones to which to map B, etc.). This is of the order of 2^{88}, a far cry from 26. Thus, brute force is no longer a viable option (well with a supercomputer these days it is, but we'll see why that doesn't matter). There is a bigger issue with this cipher, which is that it works on the level of characters, and thus preserves the patterns in characters. For example, in any sufficiently large English text, you will find that E is the most frequently occurring character, and accounts for about 13% of characters, so in any sufficiently large chunk of ciphertext, we can analyze the most frequently occurring characters, and map the top one to correspond to the plaintext "E", and by doing so for a few letters, we can pretty quickly reduce our key space to far more reasonable numbers than 26 factorial, from whereon we can brute force the key. Frequency analysis is another powerful attack against ciphers which preserve patterns in the plaintext, and thus are a good attack to be aware of when considering the design and structure of ciphers.

3 Modular Arithmetic

Before we move on to Hill Ciphers, there is some mathematical background to set up. Secure cryptography often relies on operations that are easy to perform, but difficult to invert. Modular arithmetic serves this purpose very well. In essence, it is just the usual arithmetic operations, except with the application of the "modulo" operator after the evaluation of any expression. We say for example that a "mod" $m = k$, where k is the remainder when you divide a by m. As an example, imagine that we are working with a system mod 5 - then, we have equations like:

$$2 \cdot 3 = 6 = 1 \text{ (mod 5)}.$$

Note that this means that the output of any operation, mod m, will be an integer from 0 to $(m-1)$, inclusive. As we saw with Hill Ciphers, we will find this particularly useful when we want to perform our cryptographic operations, but still end up with a string of integers that we can map back to text directly.

3.1 Systems of Linear Equations

Given that modular operations should obey scalar multiplication and addition (under the modular conditions), it is natural to consider how these operations might transfer to the solution of a system of linear equations, modulo a given m. Take the 2-variable, 2-equation case:

$$ax + by = f \text{ (mod } m\text{)}$$
$$cx + dy = g \text{ (mod } m\text{)}$$

where we hope to deterministically identify x and y that satisfy the pair of equations given all others as constants. Well, let us begin by tackling the problem as if the modulo doesn't even exist (for as long as we can) since that seemed to work well with simple multiplication. We may begin by rewriting the system as a matrix equation:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \text{ (mod } m\text{)}.$$

From here, there's nothing (so far) stopping us from simply inverting the given matrix, and multiplying through by it. After all, all we need from a matrix inversion is that multiplying it by the original matrix yields the identity, and the identity is still the identity modulo any m. So, we have:

$$\begin{bmatrix} x \\ y \end{bmatrix} = (ad - bc)^{-1} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \begin{bmatrix} f \\ g \end{bmatrix} \pmod{m}.$$

Here, we need to deal first with the negatives. This is easy enough, as we can more rigorously uniquely identify a mod m for arbitrary integer a as $k \in [0, m)$ such that $a = b \cdot m + k$, for integer b (note, not necessarily positive b). Thus, we can take negative values of a, for example:

$$-4 = -5 + 1 = 1 \pmod{5}.$$

The negative values therefore don't present an issue under modular conditions. However, there still exists the issue of the multiplicative inverse (division, effectively), to compute $(ad - bc)^{-1}$.

3.2 Introducing Modular Inversion

Now, observe that we were able to perform modular multiplication incredibly easily, as it is just the arithmetic multiplication that we are used to, followed by the modulo. However, if we try and invert the process we just did, in regular arithmetic we obviously end up with division, and this is no problem, however if we try that here, for example to find 3 given 1 and 2 in the above equation, so:

$$2 \cdot x = 1 \pmod{5},$$

we can try to just perform arithmetic division, which gets us:

$$x = \frac{1}{2} \pmod{5},$$

and we don't really have any way to progress from here - no way to get back to an integer on the interval $[0, 5)$.

In fact, this problem ends up being nowhere near as trivial as it might seem at first glance. Luckily, there are a few algorithms that we can use to simplify things. Begin by considering that we only need to be able to find what is called the "modular multiplicative inverse" of a number in order to divide by it. For example, if we know that:

$$2 \cdot 3 = 1 \pmod{5},$$

we say "3 is the multiplicative inverse of 2 modulo 5", and we can now solve the equation:

$$2 \cdot x = 3 \pmod{5},$$

by first multiplying both sides by the multiplicative inverse of 2 module 5, giving us:

$$2 \cdot 3 \cdot x = 9 = 4 \pmod{5},$$

because we know that under modular multiplication, we can simplify this to:

$$1 \cdot x = 4 \pmod 5,$$

and so:

$$x = 4.$$

This is of course trivial to verify, as:

$$2 \cdot 4 = 8 = 3 \pmod 5.$$

Thus our problem has been reduced to, given an integer m, and another integer $a \in [0, m)$, finding an integer $b \in [0, m)$ such that:

$$a \cdot b = 1 \pmod m.$$

3.3 Conditions for Invertibility

Note that the multiplicative modular inverse may not exist under all conditions. Specifically, a necessary condition of the invertibility of an integer a mod m is that $\gcd(a, m) = 1$. This is easy enough to show by contradiction.

Assume for a contradiction that $\gcd(a, m) = d \neq 1$, and we also have a multiplicative inverse for a mod m, call it k. Then we can by definition write $a = bd$ and $m = cd$, for some positive integral b, c. Note that d is also positive integral by the definition of the greatest common divisor. By definition:

$$ak = em + 1,$$

for some integer e. Then, substituting our expressions for a and m in terms of d, we have:

$$bdk = ecd + 1,$$

so

$$(bk - ec)d = 1,$$

where both terms are integers, which is not possible unless $d = 1$, which we have said it isn't, or $d = -1$, which again, since it has to be positive, it can't be. Thus we cannot have a multiplicative inverse for a if the greatest common divisor of a and m is not 1.

3.4 Computing the inverse

It so happens that the condition that we proved above was necessary is also sufficient, and there are two primary methods by which we can perform the inversion, assuming we know that $\gcd(a, m) = 1$. Namely, the Extended Euclidean Algorithm and Euler's Theorem, which employs Euler's totient function. The former is computationally lengthy, so we will prefer the latter.

We will now use m for the modulus, so it doesn't conflict with the n in Hill n-cipher below. Assuming now that for a given a, m, a has an inverse modulo m, so $\gcd(a, m) = 1$, we may employ Euler's Theorem which states that:

$$a^{\phi(m)} \equiv 1 \pmod{m},$$

where $\phi(m)$ is Euler's totient function, which is defined as the product of the unique prime factors of m.

In the case where m is prime, then the totient function merely returns the given input. This fact gives rise to Fermat's little theorem,

$$a^p \equiv a \pmod{p},$$

where p is prime.
So, in the case when m is prime, the inverse is given simply by,

$$a^{m-1} \pmod{m}.$$

Even when m is composite, this is still incredibly useful, because by definition we have:

$$a \cdot a^{\phi(m)-1} \equiv 1 \pmod{m},$$

so we have our multiplicative inverse,

$$a^{\phi(m)-1} \pmod{m}.$$

With this, and the system of linear equations we dealt with previously, we have everything we need to dive into Hill Ciphers.

4 Hill Ciphers

Hill Ciphers are a cryptographic method that uses linear algebra in order to encipher and decipher a message consisting (typically) of just a string of letters, though any finite character set can be encoded equivalently. Note that in order that the matrices with which we operate be invertible, we need the size of our character set to be relatively prime with the determinant of our matrix. Thus, it usually helps that the size of our character set is prime, so we can have more possible determinants for our matrix, and thus a larger key space.

4.1 Encryption

The process of encryption is simple, we begin by choosing the size of the "chunks" we want to use for our cipher. Calling this n, our resulting Hill Cipher becomes a Hill n-cipher. Suppose we have the string "HELLOWORLD", if we choose $n = 2$, we would end up encrypting individually the digraphs (two character chunks) "HE", "LL", "OW", "OR", and "LD". If necessary, we can just pad our string with some gibberish, say all "A"s, to make sure that the length of our string is divisible by n. Note that this entire process is something we did not have to consider with shift and substitution ciphers,

Next, we need to map our characters to integers, from $[0, m)$, where m is the size of our character set. This is accomplished easily enough and for the standard character set of the uppercase English alphabet, with $m = 26$ we can just map "A" to 0, "B" to 1, and so on until "Z" to 25, as we did for the Caesar cipher. Then, we pick our encryption key, an $n \times n$ matrix of integers, $K_{ii} \in [0, 26)$. From here on we will demonstrate with $n = 2$, for convenience.

Once we have chosen our encryption matrix, we can encrypt any string of arbitrary length of characters from our character set, by first padding it to be easily divisible into digraphs, then splitting it up into digraphs, converting each digraph to integers by our mapping, which will make them each an n dimensional vector, with each element corresponding to one character in the original digraph, and finally just multiplying that digraph vector by our matrix, yielding an encrypted vector that we can convert back to characters by the same process after taking each element modulo m, so that we fit everything back into the same range.

For example, suppose we have once again the plaintext "ETTUBRUTEXOXO", which we wish to encrypt using a Hill 2-cipher, with the key:

$$K = \begin{bmatrix} 7 & 12 \\ 23 & 1 \end{bmatrix}.$$

We can confirm the invertibility of this matrix for decryption by taking the determinant:

$$\det K = -269 = 17 \pmod{26},$$

and confirming that since $\gcd(17, 26) = 1$, our encryption matrix is invertible. Now, in order to operate on our plaintext, we have to first pad it with some character to make the size a multiple of $n = 2$, so that we can split it up evenly. Thus, we pad with "A"s, giving us "ETTUBRUTEXOXOA" as our plaintext. Now, we can convert every digraph to a 2-d vector of integers, by the usual mapping ("A" to 0 through "Z" to 25), and lay them all out in a single 2×7 matrix:

$$A = \begin{bmatrix} 4 & 19 & 1 & 20 & 4 & 14 & 14 \\ 19 & 20 & 17 & 19 & 23 & 23 & 0 \end{bmatrix},$$

note that this matrix must be read top to bottom, left to right to read out the plaintext characters in order, as each digraph is arranged vertically. Now, the encryption process is as trivial as the matrix multiplication:

$$\begin{aligned} KA &= \begin{bmatrix} 7 & 12 \\ 23 & 1 \end{bmatrix} \begin{bmatrix} 4 & 19 & 1 & 20 & 4 & 14 & 14 \\ 19 & 20 & 17 & 19 & 23 & 23 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 22 & 9 & 3 & 4 & 18 & 10 & 20 \\ 7 & 15 & 14 & 11 & 11 & 7 & 10 \end{bmatrix} \pmod{26}. \end{aligned}$$

Converting this matrix back to text, by the same process we used to initially encode A, we have our cipher text "WHJPDOELSLKHUK", encoded a lot more securely than your usual shift cipher, or even substitution cipher, as it disrupts single-character patterns (though it is still, as we will see, susceptible to frequency analysis attacks on digraphs, instead of characters).

4.2 Decryption

Decryption is accomplished by reversing exactly that process. Given the encryption key, we know that what we did in order to encrypt was:

$$KA = C \pmod{m},$$

where K is our encryption matrix, X our integer encoded plaintext matrix, and C, the encrypted ciphertext. So, in order to reverse this, we take what we did in the system of linear, modular equations, and multiply both sides by K^{-1}, which we can find by the methods described in the Modular Arithmetic section. The issue to be aware of here is that in order for the determinant of K to have a modular inverse, so that the matrix is actually invertible, it is not only necessary that it be non-zero, but also that it be relatively prime to m.

This is a direct consequence of the condition that $\gcd(a, m) = 1$ for a to have a multiplicative inverse mod m. Thus it is extremely convenient for m to be prime, so that we can have a near arbitrary E, but in the case that m is not prime, it is necessary that we pick E such that its determinant shares no common factors with m except 1.

Assuming that condition is met, the process of inversion is exactly that of solving the system of linear equations as described previously. Luckily, in our encryption example we showed that this is the case, and so we can reverse the process here, to get our plaintext back from the ciphertext.

The first step is again, to convert from our ciphertext "WHJPDOELSLKHUK" back to the integer matrix representation, as described above. This yields:

$$C = \begin{bmatrix} 22 & 9 & 3 & 4 & 18 & 10 & 20 \\ 7 & 15 & 14 & 11 & 11 & 7 & 10 \end{bmatrix}.$$

Now, what we want is to invert our key,

$$K = \begin{bmatrix} 7 & 12 \\ 23 & 1 \end{bmatrix},$$

modulo 26, and multiply this inverse by C, again modulo 26, thus returning our ciphertext. Thus, we have, by the Euler's theorem method of modular inversion:

$$K^{-1} = (17)^{-1} \begin{bmatrix} 1 & -12 \\ -23 & 7 \end{bmatrix} = 17^{12-1} \begin{bmatrix} 1 & 14 \\ 3 & 7 \end{bmatrix} = 23 \begin{bmatrix} 1 & 14 \\ 3 & 7 \end{bmatrix} = \begin{bmatrix} 23 & 10 \\ 17 & 5 \end{bmatrix} \pmod{26}.$$

Multiplying this inverse key by our matrix representation of our ciphertext, we have:

$$A = K^{-1}C = \begin{bmatrix} 23 & 10 \\ 17 & 5 \end{bmatrix} \begin{bmatrix} 22 & 9 & 3 & 4 & 18 & 10 & 20 \\ 7 & 15 & 14 & 11 & 11 & 7 & 10 \end{bmatrix} = \begin{bmatrix} 4 & 19 & 1 & 20 & 4 & 14 & 14 \\ 19 & 20 & 17 & 19 & 23 & 23 & 0 \end{bmatrix} \pmod{26}.$$

This of course lines up with our initial matrix representation of the plaintext, but just to be sure, we can convert it back to text, yielding "ETTUBRUTEX-OXOA", as we wanted. Caesar would be proud how far we've come!

5 Frequency Analysis

Now, let us explore how frequency analysis could be used to attack a Hill 2-Cipher. Frequency analysis seeks to narrow down the search space of possible encryption keys by making the assumption that the most frequently occurring digraphs in a large sample of ciphertext (encrypted plaintext) map to the most frequently occuring digraphs in the original plaintext. This is a valid way to attack a Hill Cipher because it maintains linearity (an unfortunate consequence of the dependence on linear algebra).

As an example, assume that you intercept the message "SONAFQCHMW-PTVEVY," which you know was encrypted using a Hill 2-cipher. By prior analysis, we happen to know that the most frequently occurring digraphs were "KH" and "XW," so we assume those correspond to "TH" and "HE," respectively, as they are the most frequent digraphs in long plaintext (which we know from context). Our goal is to find the deciphering matrix, and decrypt the message.

Our first step in decryption is to convert the four digraphs with which we are working to integers, assuming the zero-based alphabetic encryption scheme discussed previously. Thus, we get "KH," "XW," "TH," and "HE" mapped to:

$$\begin{bmatrix} 10 \\ 7 \end{bmatrix}, \begin{bmatrix} 23 \\ 22 \end{bmatrix}, \begin{bmatrix} 19 \\ 7 \end{bmatrix}, \text{ and } \begin{bmatrix} 7 \\ 4 \end{bmatrix},$$

respectively. From here, we can set up two matrix equations, taking:

$$E = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

This gives us:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 19 \\ 7 \end{bmatrix} = \begin{bmatrix} 10 \\ 7 \end{bmatrix} \pmod{26},$$
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 7 \\ 4 \end{bmatrix} = \begin{bmatrix} 23 \\ 22 \end{bmatrix} \pmod{26}.$$

We can break these up into four linear, modular equations as:

$$\begin{aligned} 19a + 7b &= 10 \pmod{26}, \\ 19c + 7d &= 7 \pmod{26}, \\ 7a + 4b &= 23 \pmod{26}, \\ 7c + 4d &= 22 \pmod{26}. \end{aligned}$$

Pairing these off by the variables they address, we can recoalesce them into two matrix equations, one in terms of a and b, and one in terms of c and d. This gives us:

$$\begin{bmatrix} 19 & 7 \\ 7 & 4 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 10 \\ 23 \end{bmatrix} \pmod{26},$$
$$\begin{bmatrix} 19 & 7 \\ 7 & 4 \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} 7 \\ 22 \end{bmatrix} \pmod{26}.$$

In order to solve either of these equations, the first step is to invert the matrix:

$$\begin{bmatrix} 19 & 7 \\ 7 & 4 \end{bmatrix},$$

modulo 26.

First, we need to verify that this is even possible. So, we need to check that the determinant of the matrix is indeed coprime with 26, that is doesn't have $2, 13, 26$ as factors. The determinant is:

$$\left| \begin{bmatrix} 19 & 7 \\ 7 & 4 \end{bmatrix} \right| = 19 \cdot 4 - 7 \cdot 7 = 1 \pmod{26},$$

which indeed doesn't share any factors with 26 apart from 1, so we can take the inverse as:

$$\begin{bmatrix} 19 & 7 \\ 7 & 4 \end{bmatrix}^{-1} = (27 \pmod{26})^{-1} \begin{bmatrix} 4 & -7 \\ -7 & 19 \end{bmatrix} \pmod{26},$$

where $(27 \pmod{26})^{-1}$ is the inverse of 1, modulo 26, which luckily is just 1, trivially. Similarly, we can take the remaining matrix modulo 26 to get rid of the negatives, yielding:

$$\begin{bmatrix} 19 & 7 \\ 7 & 4 \end{bmatrix}^{-1} = \begin{bmatrix} 4 & 19 \\ 19 & 19 \end{bmatrix} \pmod{26}.$$

Thus, we have from our original equations:

$$\begin{bmatrix} 4 & 19 \\ 19 & 19 \end{bmatrix} \begin{bmatrix} 19 & 7 \\ 7 & 4 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 4 & 19 \\ 19 & 19 \end{bmatrix} \begin{bmatrix} 10 \\ 23 \end{bmatrix} \pmod{26},$$
$$\begin{bmatrix} 4 & 19 \\ 19 & 19 \end{bmatrix} \begin{bmatrix} 19 & 7 \\ 7 & 4 \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} 4 & 19 \\ 19 & 19 \end{bmatrix} \begin{bmatrix} 7 \\ 22 \end{bmatrix} \pmod{26}.$$

So, cancelling out

$$\begin{bmatrix}4 & 19\\19 & 19\end{bmatrix}\begin{bmatrix}19 & 7\\7 & 4\end{bmatrix} = \begin{bmatrix}209 & 104\\494 & 209\end{bmatrix} = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \pmod{26},$$

we have:

$$\begin{bmatrix}a\\b\end{bmatrix} = \begin{bmatrix}4 & 19\\19 & 19\end{bmatrix}\begin{bmatrix}10\\23\end{bmatrix} = \begin{bmatrix}477\\627\end{bmatrix} = \begin{bmatrix}9\\3\end{bmatrix} \pmod{26},$$
$$\begin{bmatrix}c\\d\end{bmatrix} = \begin{bmatrix}4 & 19\\19 & 19\end{bmatrix}\begin{bmatrix}7\\22\end{bmatrix} = \begin{bmatrix}446\\551\end{bmatrix} = \begin{bmatrix}4\\5\end{bmatrix} \pmod{26}.$$

So, the original encryption matrix must be:

$$\begin{bmatrix}9 & 3\\4 & 5\end{bmatrix}.$$

Now, we need to invert this encryption matrix, modulo 26, so that we can decrypt the original matrix:

$$\begin{aligned}\begin{bmatrix}9 & 3\\4 & 5\end{bmatrix}^{-1} &= 7^{-1}\begin{bmatrix}5 & -3\\-4 & 9\end{bmatrix}\\ &= 7^{\phi(26)-1}\begin{bmatrix}5 & 23\\22 & 9\end{bmatrix}\\ &= 15\begin{bmatrix}5 & 23\\22 & 9\end{bmatrix}\\ &= \begin{bmatrix}23 & 7\\18 & 5\end{bmatrix} \pmod{26}.\end{aligned}$$

Now, we can finally decipher our ciphertext direction, after we convert the digraphs to integers. This gives us that "SONAFQCHMWPTVEVY" came from:

$$\begin{bmatrix}23 & 7\\18 & 5\end{bmatrix}\begin{bmatrix}18 & 13 & 5 & 2 & 12 & 15 & 21 & 21\\14 & 0 & 16 & 7 & 22 & 19 & 4 & 24\end{bmatrix} = \begin{bmatrix}18 & 13 & 19 & 17 & 14 & 10 & 17 & 1\\4 & 0 & 14 & 19 & 14 & 1 & 8 & 4\end{bmatrix} \pmod{26},$$

which, converting back to letters (remember, reading top to bottom then left to right, because each column is a separate digraph) corresponds to the plaintext message "SENATORTOOKBRIBE", we've successfully cracked the code!

6 Increased Encryption

It was pretty easy for us to attack that by frequency analysis. If we want to be more secure without upping the size of our cipher, it is often sufficient to just apply it twice, with two keys, for example by first encrypting it using a Hill 2-cipher by applying the matrix $K_{26} = \begin{bmatrix}3 & 11\\4 & 15\end{bmatrix}$ working modulo 26, and then applying the matrix $K_{29} = \begin{bmatrix} 10 & 15 \\ 5 & 9 \end{bmatrix}$, working modulo 29, by padding our character set with " ", "?", and "!" mapped to 26, 27, and 28 respectively. Since the matrices are defined modulo different numbers, there is not a trivial way to compose the two transformations, and so this is in fact a more intricate cipher than a single Hill 2-cipher.

6.1 Double Encryption

Using these schematics, we can encipher the message "SEND" as follows. First, as usual, we split into digraphs and convert them to 2-d integer vectors, by the usual mapping. This leaves us with:

$$A = \begin{bmatrix} 18 & 13 \\ 4 & 3 \end{bmatrix}.$$

Now, we apply the first Hill cipher,

$$K_{26}A = \begin{bmatrix} 3 & 11 \\ 4 & 15 \end{bmatrix} \begin{bmatrix} 18 & 13 \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 20 & 20 \\ 2 & 19 \end{bmatrix} \pmod{26}.$$

And subsequently apply the second Hill cipher:

$$K_{29}(K_2 6A \pmod{26}) = \begin{bmatrix} 10 & 15 \\ 5 & 9 \end{bmatrix} \begin{bmatrix} 20 & 20 \\ 2 & 19 \end{bmatrix} = \begin{bmatrix} 27 & 21 \\ 2 & 10 \end{bmatrix} \pmod{29}.$$

Finally, converting this back to text by the integer-text encoding we employed, we have our ciphertext "?CVK", relatively immune to frequency analysis attacks.

6.2 Double Decryption

Similarly, using these schematics, we can decipher the message "ZMOY" as follows. Now, we have two inverses to calculate and we have to check their validity independently. First, we check:

$$\det K_{29} = 15 \pmod{29},$$

so we're good on that front. Second, we check:

$$\det K_{26} = 1 \pmod{26},$$

so both matrices are invertible, and we can proceed with actually computing those inverses. First:

$$K_{29}^{-1} = (15)^{\phi(29)-1} \begin{bmatrix} 9 & -15 \\ -5 & 10 \end{bmatrix} = 15^{27} \begin{bmatrix} 9 & 14 \\ 24 & 10 \end{bmatrix} = 2 \begin{bmatrix} 9 & 14 \\ 24 & 10 \end{bmatrix} = \begin{bmatrix} 18 & 28 \\ 19 & 20 \end{bmatrix} \pmod{29}.$$

Note that $\phi(29) = 28$, because since 29 is prime, every positive integer less than 29 is relatively prime to it. Second:

$$K_{26}^{-1} = 1^{11} \begin{bmatrix} 15 & -11 \\ -4 & 3 \end{bmatrix} = \begin{bmatrix} 15 & 15 \\ 22 & 3 \end{bmatrix} \pmod{26}.$$

Finally, we can convert our ciphertext into an integer matrix representation, as we usually do, remembering to use the extended version (as our ciphertext could contain the three new characters we defined, though of course in this case it doesn't) which gives us:

$$C = \begin{bmatrix} 25 & 14 \\ 12 & 24 \end{bmatrix}.$$

Putting it all together to invert our ciphertext, we first apply K_{29}^{-1} (remember, the usual order of matrix inversions applies, we want to go from the outside in):

$$K_{29}^{-1}C = \begin{bmatrix} 18 & 28 \\ 19 & 20 \end{bmatrix} \begin{bmatrix} 25 & 14 \\ 12 & 24 \end{bmatrix} = \begin{bmatrix} 3 & 25 \\ 19 & 21 \end{bmatrix} \pmod{29}.$$

We follow this up by applying K_{26}^{-1}:

$$K_{26}^{-1}(K_{29}^{-1}C \pmod{29}) = \begin{bmatrix} 15 & 15 \\ 22 & 3 \end{bmatrix} \begin{bmatrix} 3 & 25 \\ 19 & 21 \end{bmatrix} = \begin{bmatrix} 18 & 14 \\ 19 & 15 \end{bmatrix}.$$

We can convert this back to text as usual, yielding the plaintext message "STOP". Indeed, this seems like a good place to stop.

7 Conclusion

TBD