Cryptography: Hill Ciphers

Aadi Anand, Alex Lipson

May 2024

1 Introduction

The modern world is built on secrets.

2 Cryptography

3 Modular Arithmetic

Secure cryptography often relies on operations that are easy to perform, but difficult to invert. Modular arithmetic is a mathematical paradigm that implements exactly this principle. 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 m "mod" $n = k$, where k is the remainder when you divide m by n. As an example, imagine that we are working with a system mod 5 - then, we have equations like:

$$2 \cdot 3 = 6 = 1 \pmod 5.$$

Note that this means that the output of any operation, mod n, will be an integer from 0 to $(n-1)$, inclusive. We will find this extraordinarily 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.

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 n. Take the 2-variable, 2-equation case:

$$\begin{aligned} ax + by &= f \pmod n \\ cx + dy &= g \pmod n \end{aligned}$$

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:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \pmod{n}.$$

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 n. So, we have:

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

Here, we need to deal first with the negatives. This is easy enough, as we can more rigorously uniquely identify $a \bmod n$ for arbitrary integer a as $k \in [0, n)$ such that $a = m \cdot n + k$, for integer m (note, not necessarily positive integer m). 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 n, and another integer $a \in [0, n)$, finding an integer $b \in [0, n)$ such that:

$$a \cdot b = 1 \pmod{n}.$$

3.3 Conditions for Invertibility

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

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

$$a \cdot k = en + 1,$$

for some integer e. Then, substituting our expressions for a and n 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 n is not 1.

Now, in order to prove sufficiency, consider that if $\gcd(a, n) = 1$, then there exists integers b, m, such that $ab + mn = 1$, by using the Euclidean Algorithm to compute the gcd, but in reverse. Thus, there exist integers b, m such that:

$$ab = 1 - mn,$$

so

$$ab = 1 \pmod{n}.$$

3.4 Euler's Theorem

Assuming now that for a given a, n, a has an inverse modulo n, so $\gcd(a, n) = 1$, we may employ Euler's Theorem which states that:

$$a^{\phi(n)} = 1 \pmod{n},$$

where $\phi(n)$ is Euler's totient function, the number of positive integers, $k \leq n$ for which $\gcd(k, n) = 1$. This is incredibly useful, because by definition then we have:

$$a \cdot a^{\phi(n)-1} = 1 \pmod{n},$$

so we have our multiplicative inverse,

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

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 encrypt and decrypt 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, it is often convenient that the size of our character set be prime, though things sometimes work out regardless. The reason for this will be evident in our discussion of decryption.

4.1 Encryption

The process of encryption is simple, we begin by choosing the size of digraph we wish to use for our cipher. Calling this n, our resulting Hill Cipher becomes a Hill n-cipher. A digraph is the size of each "chunk" of our data, so to speak. Suppose we have the string "HELLOWORLD", if we choose $n = 2$, we would end up encrypting individually the 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.

Next, we need to map our characters to integers, from $[0, c)$, where c is the size of our character set. This is accomplished easily enough and for the standard character set of the uppercase English alphabet, we can just map "A" to 0, "B" to 1, and so on until "Z" to 25. Then, we pick our encryption key, an $n \times n$ matrix of integers $a_{ii} \in [0, 26)$. 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 c, so that we fit everything back into the same range.

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:

$$E\mathbf{x} = \mathbf{a} \pmod{c},$$

where E is our encryption matrix, $\mathbf{x}$ our original, integer version of our digraph, and $\mathbf{a}$, the encrypted output. So, in order to reverse this, we take what we did in the system of linear, modular equations, and multiply both sides by E^{-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 E 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 c.

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

Assuming that condition is met, the process of inversion is exactly that of solving the system of linear equations as described previously, which means that the Hill Cipher is not really that secure (because even if you don't have E, there's only so many options and once you have it, the code is easily breakable).

5 Frequency Analysis

Now, let us explore a common method of breaking ciphers, which is by frequency analysis. 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 "SONAFQCHMWPTVEVY," 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:

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

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:

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

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:

$$\left| \begin{bmatrix} 19 & 7 \\ 7 & 4 \end{bmatrix} \right| = 19 \cdot 4 - 7 \cdot 7 = 27,$$

which indeed doesn't share any factors with 26, 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:

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

Now, we can finally decrypt our ciphertext, 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, from integers, is "SENATORTOOKBRIBE" and the original message.

6 Increased Encryption

In order to increase the difficulty of breaking an encryption system, it is often sufficient to just apply it twice, for example by first encrypting it using a Hill 2-cipher by applying the matrix $\begin{bmatrix}3 & 11\\4 & 15\end{bmatrix}$ working modulo 26, and then applying the matrix $\begin{bmatrix}10 & 15\\5 & 9\end{bmatrix}$, working modulo 29, by padding our character set with a blank space, ?, and ! mapped to 26, 27, and 28 respectively.

6.1 Double Encryption

Using these schematics, we can encipher the message "SEND" as follows:

We represent the plaintext with integers and then successively apply the encryption matrices, careful to retain their respective moduli.

$$\left(\begin{bmatrix}10 & 15\\5 & 9\end{bmatrix}\left(\left(\begin{bmatrix}3 & 11\\4 & 15\end{bmatrix}\begin{bmatrix}18 & 13\\4 & 3\end{bmatrix}\right) \pmod{26}\right)\right) \pmod{29} = \begin{bmatrix}27 & 21\\2 & 10\end{bmatrix}.$$

This matrix gives the ciphertext, "?CVK."

6.2 Double Decryption

Similarly, using these schematics, we can decipher the message "ZMOY" as follows:

First, we find the modular inverses of the two encryption matrices,

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

and

$$\begin{bmatrix} 10 & 15 \\ 5 & 9 \end{bmatrix}^{-1} \pmod{29} = \begin{bmatrix} 18 & 28 \\ 19 & 20 \end{bmatrix}.$$

We apply these inverse matrices to decrypt the given cipher text, being careful to apply the modulus 29 matrix prior to the modulus 26 one in order to obtain the proper plaintext.

$$\left(\begin{bmatrix} 15 & 15 \\ 22 & 3 \end{bmatrix}\left(\left(\begin{bmatrix} 18 & 28 \\ 19 & 20 \end{bmatrix}\begin{bmatrix} 25 & 14 \\ 12 & 24 \end{bmatrix}\right) \pmod{29}\right)\right) \pmod{26} = \begin{bmatrix} 18 & 14 \\ 19 & 15 \end{bmatrix}.$$

This gives us the decrypted plaintext, "STOP."

7 Conclusion