# Cryptography and Linear Algebra

Runtime Terror

September 2019

## 1 Introduction

Imagine A and B. A and B can be people, clients, servers, peer computers, network routers. A can send an encrypted* message to B, and B has the ability to decrypt the message sent from A. Then we have C and D, where C can only view the message sent by A but cannot modify it and on the other hand, D can view the message and modify it. But not only, assuming that D can delete that message and B will never receive it or D can hijack the network or spread malware or do a denial service while B is trying to access the message. But it will happen only if D knows the secret algorithm behind the encryption and decryption process of A and B. But most of the cryptosystems are not relying only on the algorithm being kept secret that is why there are low chances that D will never found out about the secret algorithm of A and B.

The process of encrypting and decrypting messages is called cryptography. Most of the time, cryptography is a method to secure communication and information through the use of codes so that only those for whom the information is intended can read and process it. The pre-fix "crypt" means "hidden" or "vault" and the suffix "graphy" stands for "writing". Nowadays in the modern cryptography methods, we use coded algorithms, hashes*, signatures and other functions. Cryptography relies extensively on mathematics. This includes basic function theory, permutations, probability, information theory, complexity theory, number theory, and more.

The history of cryptography is not recent. Its origin is usually dated from about 2000 B.C., with the Egyptian practice of hieroglyphics. These consisted of complex pictograms, the full meaning of which was only known to an elite few. The first known use of a modern cipher* was by Julius Caesar (100 B.C. to 44 B.C.), who did not trust his messengers when communicating with his governors and officers. For this reason, he created a system in which each character in his messages was replaced by a character three positions ahead of it in the Roman alphabet. Now in the 21st century, cryptography has turned into a battleground of some of the world's best mathematicians and computer scientists. The ability to securely store and transfer sensitive information has proved a critical factor in success in war and business.

## 2 Technical terms

Ensuring that the reader understood those: In our daily lives, we use encryption without noticing it or we may notice it but we do not know what is it exactly and how does it work. For example, using WhatsApp, connecting to the WIFI which the password is the key or login into your online bank account, you are using encryption. In order to understand the world of cryptography, these are the basic terminologies everyone should know:

- **Plaintext**: readable, plain message that anyone can read.
- **Ciphertext**: the result of the encryption process. The encrypted plaintext appears as apparently random strings of characters.
- **Cipher**: another way of referring to the encryption algorithm that transforms the plaintext. Here are some useful categories of ciphers:
  - **Classical**: A cipher easy enough to be performed by hand, usually character-based. Also called a manual.
  - **Modern**: Pretty much any cipher that isn't classical.
  - **Substitution**: Each character of the plaintext is replaced with one or more characters to make the ciphertext.
  - **Characters**: Characters in the plaintext are rearranged to form the ciphertext.
  - **Monoalphabetic**: A substitution cipher in which a character of the plaintext is always replaced by the same character.
  - **Polyalphabetic**: A substitution cipher that essentially uses multiple monoalphabetic substitution mappings.
  - **Homophonic**: A substitution in which one character can map to one of a set of characters.
  - **Polygraphic**: A substitution of blocks of characters for blocks of characters.
  - **Periodic**: A polyalphabetic cipher in which the replacement scheme repeats.
  - **Non-periodic**: Self-explanatory if you understand periodic.
  - **Block**: Encryption takes place not per character but per blocks of characters.
  - **Stream**: A cipher operating on a data stream of unknown length, usually incorporating feedback.
  - **Secret Key**: A cipher in which ke or ke and kd or kd are the same or trivially derivable from one another; requires the parties to meet in secret to exchange the keys they'll be using.

  - **Polygraphic Substitution**: Polygraphic substitution is a method whereby a plaintext is divided into groups of letters. Then, each group of letters is replaced by one of the predefined letters, numbers, graphic symbols, or by another group of characters.

- **Encryption**: Tthe process of applying a mathematical function to a file that renders its contents unreadable and inaccessible—unless you have the decryption key.
- **Decryption**: If encryption locks the file, then decryption reverses the process, turning ciphertext back to plaintext.
- **Keys**: The encryption process requires a key that tells the algorithm how to transform the plaintext into ciphertext. The key locks and unlocks the algorithm, allowing the encryption or decryption process to function.
- **Hash**: Different from encryption in that once the data is hashed, it cannot be unhashed. Or rather, it is extremely difficult. Hashing is really useful when you need to verify something's authenticity, but not have it read back. In this, password hashing offers some protection against brute-force attacks (where the attacker tries every possible password combination).
- **Salt**: Random piece of data used as an enhancement of a one-way function that hashes a passphrase. The purpose of using salts is to increase defense against a dictionary attack or safeguard passwords.
- **Symmetric Algorithm**: Use the same key for both encryption and decryption. Both parties must agree on the algorithm key before commencing communication.
- **Assymetric Algorithm**: Use two different keys: a public key and a private key. This enables secure encryption while communicating without previously establishing a mutual algorithm. This is also known as public-key cryptology (see the following section).
- **Public and Private Keys**: The public key can be sent to other people, while the private key is only known by the owner.
- **HTTPS**: HTTPS generates long-term private and public keys that in turn are used to create a short-term session key. The session key is a single-use symmetric key that the connection destroys once you leave the HTTPS site (closing the connection and ending its encryption).
- **End-to-End Encryption**: Social messaging platform service WhatsApp began offering its users end to end encryption (E2EE) in 2016, making sure their messages are private at all times. In the context of a messaging service, EE2E means that once you hit the send button, the encryption remains in place until the recipient receives the messages.

# 3 Application of Linear Algebra in Cryptography

In cryptography, there are lots of methods to encrypt and decrypt information. But what is actually happening behind the scenes? Well, mathematics is playing a huge role in the system itself. Amongst the various encryption techniques that use linear alegrba we discuss in depth here about a technique called Hill ciphere. Hill cipher is a polygraphic substitution cipher based on linear algebra. Each letter is represented by a number modulo 26 (we take the reminder of the divsion between the word to be encrypted with 26). Often the simple scheme A = 0, B = 1, ..., Z = 25 is used to represent letters, but this is not an essential feature of the cipher. To encrypt a message, each block of n letters (considered as an n-component vector) is multiplied by an invertible n  n matrix, against modulus 26. The matrix used for encryption is called the cipher key and to decrypt the message, each block is multiplied by the inverse of the matrix used for encryption.

**Example**:

- Say Mahaly wants to send Chris the message "Cryptography is cool!" The message that is to be sent is "Cryptography is cool!", removing the spaces and punctuation symbols, and convert the letters into all uppercase gives CRYPTOGRAPHYISCOOL.
- Conversion to numbers using A = 0, B = 1, C = 2, . . . , Z = 25, gives 2 17 24 15 19 14 6 17 0 15 7 24 8 18 2 14 14 11
- Alice chooses the block size n = 3 and chooses the encryption matrix E to be,

  $$\begin{pmatrix} 2 & 3 & 15 \\ 5 & 8 & 12 \\ 1 & 13 & 4 \end{pmatrix}$$

  Since det(E) (mod 26) = 11, and 11 is invertible modulo 26(reminder of integer division of 11/26 = 11), the matrix E is also invertible modulo 26.
- Since we are using a 3*3 matrix, we break the enumerated message above into a sequence of 3*1 matrix:

  $$\begin{pmatrix} 2 \\ 17 \\ 24 \end{pmatrix} \begin{pmatrix} 15 \\ 19 \\ 14 \end{pmatrix} \begin{pmatrix} 6 \\ 17 \\ 0 \end{pmatrix} \begin{pmatrix} 15 \\ 7 \\ 24 \end{pmatrix} \begin{pmatrix} 8 \\ 18 \\ 2 \end{pmatrix} \begin{pmatrix} 14 \\ 14 \\ 11 \end{pmatrix}$$

  Converting these blocks into a message a 3x6 Matrix and multiplying it with our encoding matrix gives

  $$\begin{pmatrix} 2 & 3 & 15 \\ 5 & 8 & 12 \\ 1 & 13 & 4 \end{pmatrix} \begin{pmatrix} 2 & 15 & 6 & 15 & 8 & 14 \\ 17 & 19 & 17 & 7 & 18 & 14 \\ 24 & 14 & 0 & 24 & 2 & 11 \end{pmatrix}$$

- Then, it will give us the result:

$$\begin{pmatrix} 25 & 11 & 11 & 21 & 22 & 1 \\ 18 & 5 & 10 & 3 & 0 & 2 \\ 7 & 6 & 19 & 20 & 16 & 6 \end{pmatrix}$$

- Convert the encrypted matrix into the ciphertext.
  25 18 7 11 5 6 11 10 19 21 3 20 22 0 16 1 2 6
  ZSHLFGLKTVDUWAQBCG

- In order to decode the message, we use the inverse of the encoding matrix and multiply it by the encrypted matrix, D (mod 26).

$$D = E^{-1}$$

$$Inv \begin{pmatrix} 2 & 3 & 15 \\ 5 & 8 & 12 \\ 1 & 13 & 4 \end{pmatrix} \begin{pmatrix} -122 & -123 & -176 & -182 & -96 & -91 & -183 \\ 23 & 19 & 47 & 41 & 22 & 10 & 32 \\ 138 & 139 & 181 & 197 & 101 & 111 & 203 \end{pmatrix}$$

- Then, we get the encrypted matrix back again:

$$\begin{pmatrix} 2 & 15 & 6 & 15 & 8 & 14 \\ 17 & 19 & 17 & 7 & 18 & 14 \\ 24 & 14 & 0 & 24 & 2 & 11 \end{pmatrix}$$

- Convert the matrix E to the plaintext message gives back CRYPTOGRAPHYISCOOL So the receiver adds in a couple spaces to get CRYPTOGRAPHY IS COOL!

- Note that it is not necessary for the length of your message is a multiple of 3 because we can had a space character(with value= 27) at the end of the string to make it a mltiple of 3. However, the encryption matrix must be a sqaure matrix becuase it has to be invertible.

# 4 Hill Cypher with Python

Using python our team was able to create a simple encryption program that takes in a three character letter and encrypts it and outputs a cipher text.

```python
# Hill Cipher code for Runtime Terror group

#Generate vector for the Key matrix
keyMatrix = [[0] * 3 for i in range(3)]
#outputs [0,0,0
#         0,0,0]

# Generate vector for the message
messageVector = [[0] for i in range(3)]
# this vector takes the shape [0
#                             0
#                             0]

# Generate vector for the cipher
cipherMatrix = [[0] for i in range(3)]
# this vector takes the shape [0
#                             0
#                             0]


def HillCipher(message, key):  # defining the Hillcipher function with message and key

    # Get key matrix from the key string. The key value is passed to getKeymatrix function.
    getKeyMatrix(key)

    # Generate vector for the message
    for i in range(3):
        messageVector[i][0] = ord(message[i]) % 65
```

```python

# Following function encrypts the message
print(keyMatrix)

def encrypt(messageVector):  # definition of the messageVector function
    for i in range(3):  # loops through the rows of the key matrix
        for j in range(1):  # loops through the columns of the key
            cipherMatrix[i][j] = 0
            for x in range(3):
                cipherMatrix[i][j] += (keyMatrix[i][x] *      # vector multiplication of key matrix and messageVector
                                       messageVector[x][j])
            # we take modulus 26 of the matrix multiplication result
            cipherMatrix[i][j] = cipherMatrix[i][j] % 26


def main():

    # Get the message to be encrypted from the user
    message = input("Write your message to encrypted:")

    # Get the key
    key = "GYBNQKURP"
    #using letter representation A=0 B=1 ... this key corresponds to
    # the encryption matrix [6  24 1
    #                        13 16 10
    #                        20 17 15]
    # this matrix is an invertible modulo 25
    # Hillcipher function is passsed with message value and key value
    HillCipher(message, key)

```

```
    # Finally print the ciphertext
    print("Ciphertext: ", "".join(CipherText))


# Following function generates the key matrix for the key string
def getKeyMatrix(key):
    k = 0
    for i in range(3):  # loops through the rows of the key matrix
        for j in range(3): # loops through the columns of the key
            # the ord() function converts the letters in encryption key to their respective ascii representation
            # calculate the modulus of the key Matrix with 65 to convert the ascii values of the letters in the range [0,25]
            keyMatrix[i][j] = ord(key[k]) % 65

            k += 1


# Following function encrypts the message
print(keyMatrix)

def encrypt(messageVector):  # definition of the messageVector function
    for i in range(3):  # loops through the rows of the key matrix
        for j in range(1):  # loops through the columns of the key
            cipherMatrix[i][j] = 0
            for x in range(3):
                cipherMatrix[i][j] += (keyMatrix[i][x] *      # vector multiplication of key matrix and messageVector
                                       messageVector[x][j])
            # we take modulus 26 of the matrix multiplication result
            cipherMatrix[i][j] = cipherMatrix[i][j] % 26

```

# 5 Breaking Hill Cypher Encryption

The first thing to note is that when encoding in Hill Cipher each row of the key matrix encodes to 1 letter independently of the rest of the key matrix.

$$\begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = \begin{bmatrix} a \times L_1 + b \times L_2 + c \times L_3 \end{bmatrix} \bmod 26$$

Realising how the top row of the far left matrix is only involved in the top cell of the ciphertext matrix, the middle row is only involved in the middle cell etc can help us dramatically decrease the number of keys we have to test to break the Hill Cipher.

For square matrix of size N, there are 26NN unique keys (there will be less as not all matrices have an inverse). For N=3, there is 269  5.431012 keys, to test all of these is not feasible. However, if we test each row individually then there is only 26N keys we need to test, For N=3 there is 263 = 17,576 which is a very small number in comparison (Takes about 0.5 seconds) With this property of Hill Cipher we can go about cracking it.

First we will need to identify N (the size of the matrix) the size will be a multiple of the text length – this narrows it down a lot Now we will be to iterate over all the row vectors with a size of N and possible values of 0 (inclusive) to 26 (exclusive). For a 3 by 3 there are 17,576 combinations. They look will look something like this. On the left is the iteration number
1/17576 [ 0, 0, 0]
2/17576 [ 0, 0, 1]
3/17576 [ 0, 0, 2]

16249/17576 [24, 0, 24]
16250/17576 [24, 0, 25]
16251/17576 [24, 1, 0]
17576/17576 [25, 25, 25]

For each one of these possibilities we assume it is part of the key and multiply our ciphertext by it, we will multiply in blocks of N and get a single letter out for each block.

$$\begin{bmatrix} 21 & 18 & 12 \\ 9 & 0 & 23 \\ 8 & 3 & 2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 21a + 18b + 12c \\ 9a + 0b + 23c \\ 8a + 3b + 2c \end{bmatrix} \bmod 26$$

Once we have all the output letters for a particular possibility, we score the letters using the Chi-Squared Statistic. We then store the row vectors from smallest to largest Chi-Squared value.

Once we have checked all the possibilities. We take the best results from the list we have compiled and then go through all the permutations of creating an N by N matrix and checking it has an inverse in modular 26.

Example:

Let's say we know N=3 and the best row vectors found using this method were with a Chi-Squared value of...
[22, 6, 7] X2 = 71.721647 [23, 17, 18] X2 = 50.562860 [25, 0, 6] X2 = 81.987751

We then rearrange each row to every possible position (For R number of rows there is R!, R(R-1)(R-2)...1, permutations) The next (3! = 6) matrices are all the permutations of each row vector.

$$\begin{bmatrix} 25 & 0 & 6 \\ 22 & 6 & 7 \\ 23 & 17 & 18 \end{bmatrix} \begin{bmatrix} 25 & 0 & 6 \\ 23 & 17 & 18 \\ 22 & 6 & 7 \end{bmatrix} \begin{bmatrix} 23 & 17 & 18 \\ 25 & 0 & 6 \\ 22 & 6 & 7 \end{bmatrix}$$

Then we encrypt our ciphertext using these matrices (encrypting using the inverse key matrix is the same as decrypting using the key matrix). One of these results should be English – being our solution. If we wish to find the key matrix, we will need to inverse the inverse key matrix in mod 26.

# 6 Conclusion

The hill cypher encryption clearly demonstrates how linear algebra can be applied in cryptography. Even though, this encryption technique is not bullet proof on its own it serves as a basis layer for other encryption techniques.

## 6.1 Characteristics

These are the characteristics of cryptography:

- **Confidentiality**: The information cannot be understood by anyone for whom it was unintended.
- **Integrity**: The information cannot be altered in storage or transit between the sender and intended receiver without the alteration being detected.
- **Non-Repudiation**: The creator/sender of the information cannot deny at a later stage his or her intentions in the creation or transmission of the information.
- **Authentication**: The sender and receiver can confirm each other's identity and the origin/destination of the information.

### 6.2 Importance and Relevance

Lots of problems can be solved using cryptography. In general, the hill cipher will not be used on its own, since it is not all that secure. It is, however, still a useful step when combined with other non-linear operations, such as S-boxes (in modern ciphers). It is generally used because matrix multiplication provides good diffusion (mix and transformation of characters). Some modern ciphers use a matrix multiplication step to provide diffusion e.g. AES and Twofish use matrix multiplication as a part of their algorithms.

### 6.3 Real life demonstration of Hill Cypher Crypotgraphy

Using any programming language, it is possible to create an application that takes a message input from the user along with an encryption key and gives back the encrypted text with a decryption key. The application below is an open source project on Github and really demonstrates the power and relevance of using hill cypher encryption to transmit secret messages.
The following link directs to the github repository to access the open source code. This application is written in C++ and has advanced features of uploading a text file for encryption and outputing a decrypted text file.

Click the link below to find the file
Advanced version of hill cypher by Jhon Almardeny

A very short slide about hill cipher encryption is illustrated in this presentation slides file.click this link

## 7 Group Roles

- Natnael: Research on softwares that use linear algebra for cryptography and editing document on Latex
- Mahalinoro: Research on application of Linear Algebra, Hill cypher and editing document on Latex

- Chris: Research history of cryptography and slide design
- Zubery: Research on softwares that use linear algebra for cryptography and a

# 8 Bibliography


- Phys.org. (2019). World's first encryption technology able to match multi-source data encrypted with different keys. [online] Available at: https://phys.org/news/2016-02-world-encryption-technology-multi-source-encrypted.html [Accessed 25 Sep. 2019].
- Practicalcryptography.com. (2019). Practical Cryptography. [online] Available at: http://practicalcryptography.com/ciphers/hill-cipher/ [Accessed 25 Sep. 2019].
- Docs.aws.amazon.com. (2019). What is Cryptography? - AWS Cryptography Services. [online] Available at: https://docs.aws.amazon.com/crypto/latest/userguide/awscryp-whatis-toplevel.html [Accessed 29 Sep. 2019].
- Cs.lmu.edu. (2019). cryptology. [online] Available at: https://cs.lmu.edu/ ray /notes/cryptology/ [Accessed 29 Sep. 2019].
- SearchSecurity. (2019). What is cryptography? - Definition from WhatIs.com. [online] Available at: https://searchsecurity.techtarget.com/definition/cryptography [Accessed 29 Sep. 2019].
- The pCloud Blog. (2019). 10 common cryptography terms and their meanings — The pCloud Blog. [online] Available at: https://blog.pcloud.com/10-cryptography-terms/ [Accessed 29 Sep. 2019].
- MakeUseOf. (2019). 10 Basic Encryption Terms Everyone Should Know and Understand. [online] Available at: https://www.makeuseof.com/tag/encryption-terms/ [Accessed 29 Sep. 2019].
- Khan Academy. (2019). What is cryptography?. [online] Available at: https://www.khanacademy.org/computing/computer-science/cryptography/crypt/v/intro-to-cryptography [Accessed 29 Sep. 2019].
- GeeksforGeeks. (2019). Hill Cipher - GeeksforGeeks. [online] Available at: https://www.geeksforgeeks.org/hill-cipher/ [Accessed 29 Sep. 2019].
- Aix1.uottawa.ca. (2019). Matrices-Application to Cryptography. [online] Available at: http://aix1.uottawa.ca/ jkhoury/cryptography.htm [Accessed 29 Sep. 2019].
- Almardeny, J. (2019). John-Almardeny/Hilly. [online] GitHub. Available at: https://github.com/John-Almardeny/Hilly [Accessed 1 Oct. 2019].

- Brater, A. (2019). Cryptanalysis of Hill Cipher. [online] Cryptography. Available at: http://alexbarter.com/cryptanalysis/breaking-hill-cipher/ [Accessed 4 Oct. 2019].