**

**ENGINERING AND NATURAL SCIENCE FACULTY**

**Computer Engineering Graduation Project**

**Group 6 / Crypt Attack Application**

**Project Manager: Asst. Prof. Baha ŞEN**

**Group Members:**

*Ankara,2016*

CONTENTS

[*ABSTRACT 2*](#_Toc452918454)

[*1.INTRODUCTION 2*](#_Toc452918455)

[*2.LITERATURE 3*](#_Toc452918456)

[*2.1.* *Encryption Algorithms 3*](#_Toc452918457)

[*2.1.1 Symetric Key Algorithms 4*](#_Toc452918458)

[*2.1.1.1 Data Encryption Standart 4*](#_Toc452918458)

[*2.1.1.2 Monoalphabetic Substitution Cipher 4*](#_Toc452918458)

[*2.1.1.3 Hill Cipher 4*](#_Toc452918458)

[*2.1.2 Asymetric Key Algorithms 4*](#_Toc452918458)

[*2.1.2.1 RSA 4*](#_Toc452918458)

[*2.1.2.2 Hash Functions 4*](#_Toc452918458)

[*2.1.2.3 SHA Cipher Method 4*](#_Toc452918458)

[*2.2 Cryptoatack Algorithms 4*](#_Toc452918458)

[*2.2.1 Brut Force Attack 4*](#_Toc452918458)

[*2.2.3 Known Plaintext Algorithm 4*](#_Toc452918458)

[*3.METHODS 6*](#_Toc452918463)

[*3.1 An Expanded View on Hill Cypher 6*](#_Toc452918464)

[*3.1.1 Encryption of Hill Cypher Algorithm 6*](#_Toc452918464)

[*3.1.2 Decryption of Hill Cipher Algorithm 6*](#_Toc452918464)

[*3.1.2 Attact to be applied: Known-Plaintext Attact 6*](#_Toc452918464)

[*3.2 Ceasar Algorithm 6*](#_Toc452918465)

[*3.2.1 Encryption-Decryption of Ceasar Algorithm 6*](#_Toc452918464)

[*3.1.2 Attact to be applied: Brute-Force Attact 6*](#_Toc452918464)

[*4.CONCLUSION 8*](#_Toc452918468)

[*5.REFERENCES 8*](#_Toc452918469)

# **ABSTRACT**

***Project Definition:***

Implement application allowing to perform crypt attack:

2.1. Known plaintext-ciphertext attack for Hill cipher

2.2. Known ciphertext attack for mono alphabetic cipher

2.3. Known plaintext-ciphertext attack for Playfair cipher

2.4. Known ciphertext attack for poly alphabetic cipher

The application should ask from user all necessary for attack information, perform attack, and display results of attack.

# **1.INTRODUCTION**

Cryptography can be defined as science of keyword. The work is to encrypt messages that has different types according to predefined system, transmission of these messages in a reliable environment and to decrypt the transmitted message. Cryprology occurs from combination of cryptography and cryptanalysis. Cryptography means writing cryptogram. Cryptanalysis means decoding analysis the encryption system.

Crypattact project based on searching cryptography methods and understand its capabilities. But the main goal is to accomplish to write a crypto-attack program that working by estimating a key which we don't know previously, and get the plain text. Actually, the encryption steps are all to apply mathematical functions over different data types. Some functions have no inverse function but the encryption algorithms which we examined are the ones have inverse. When you guess the right key, you can get the plaintext by apply the function steps inversely.

The power of an encryption method which has known algorithm depends on the length of its key. Theorically, the open cryptographic algorithms which use one key can be decrypted by generate all possible keys and try to solve. To solve an open key algorithm, obtaining the true key from the ciphered key instead of etimating the plain key.

Cryptology has several methods to encrypt and decrypt a text. We can divide them in ain two categories as symetcial and asymetrical cipher methods.

# **2. LITERATURE**

2.1 Encryption Algorithms

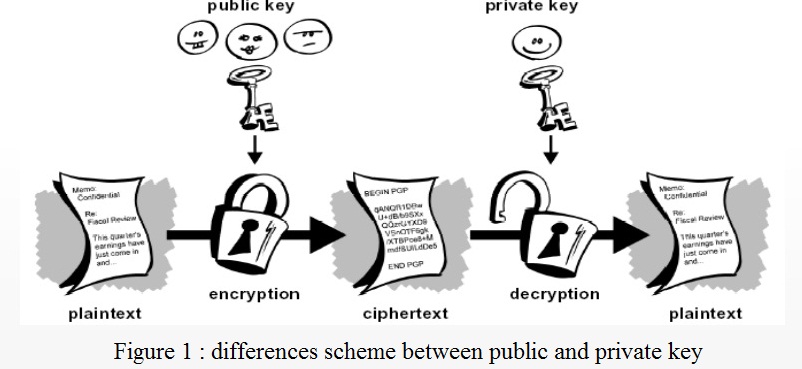
The primary purpose of encryption is to protect the confidentiality of digital data stored on computer systems or transmitted via the[Internet](http://searchwindevelopment.techtarget.com/definition/Internet) or other computer [networks](http://searchnetworking.techtarget.com/definition/network).

In this project subareas of cryptography has been seached and reported. This part of the paper related to symetric and asymetric cryptology methods, hash functions and their derivations. And the aim is to execute an cryptoatack algorithm on a choosen encryption algorithm.

*2.1.1 Symetric Key Algorithms*

Encryption algorithm with symetric key uses one secret key to decrypt. The encryption key used is hidden from others and the key is predetermined by the people who are sender and reciever.

Hidden secret key is sent together with the text to be sent while the receiving and decoding process is performed. The difference between the public and private keys is shown in Figure 1.



One of the most important advantages of symmetric encryption is that it is quite fast. Compared with asymmetric encryption algorithms, symmetric is much more successful about speed.

However, due to the simple process that included in symmetric algorithm it is much easier to implement in electronic devices. Further, the length and thus the number of bits used symmetric key algorithms are much smaller .The advantages of symetric key algorithms are the speed of inner algorithms, easy of use with hardware and security in essence. Although this algorithm has some capacity problems. Security key distribution, authentication and integrity principles services is difficult to realize securely. Symmetric algorithms are divided into two as array cipher algorithms and block cipher encryption algorithms.

Block Encryption Algorithms operate over blocks. Block cipher algorithms have no internal memory, so it is called as no memory encryption. Generally, for applications requiring integrity checking block cipher algorithms are preferred.

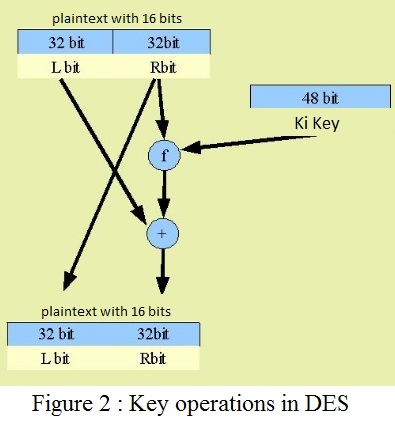
Sequence Data Encryption Algorithm operates the encryption process as a bit sequence.

An array called sliding key with the desired length is generated via a generator. The production of sliding key depends on time so these type of algorithms are named as encrypion with memory.

*2.1.1.1 Data Encryption Standart (DES)*

Data Encrption Standart algorithm is a symetric algorithm based on bit size. It basically means this algorithm divides the plaintext into blocks and encrypts independently and decrypt the cipher text among these blocks. The size of the each block is 64 bits. DES, is a standart wich is developed for encryption and decryption. For instance an 64 bit encryption steps are shown in Figure 2.

At the same time DES takes a key with 64 bit length. But the valid length of it is 56 bits because the 8 bits of the key is spent for parity. Work of the DES is given in Figure 2.



First of all the plaintext with 64 bits must be divided by 2 parts, the left-part and the right-side part are executed apart. This process is done by the 32 bit part and the key entering into function and replacing the parts. + sign means xor (exclusive) logical operator on Figure 2.

In the Figure 2 a transaction for DES is drawn, in DES encryption this process iterates 16 times. The f function in details is given in Figure 3.

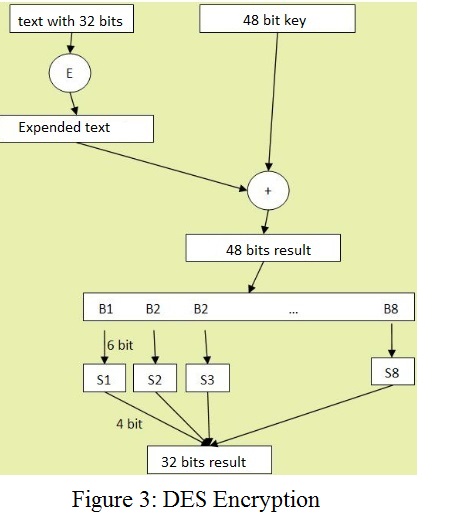
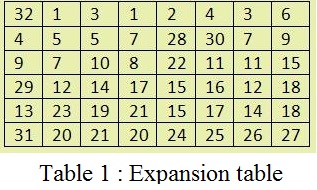


Figure 3 shows how f function works within a DES pass. This function produces 32 bit output from an 32 bit part with a 48 bit key.

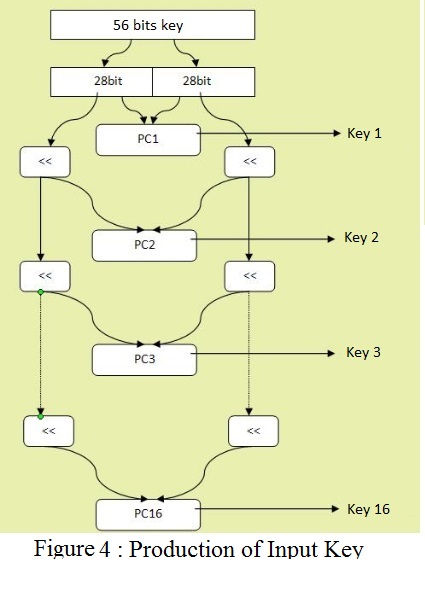
While this producing process, the most critical step is shown as E called expansion operation. Expansion operation basically foresees to can be generated more than one outputs from same bit.

Expansion process is done with a table like Table 1.



The logic behinds on this Table 1 is occurance of one value in more than one position. In this situation, for “1” either second and fourth places are responded.

DES works with 64 bit key and 8 bit parity check. Every key length which input into f function on each pass is given as 48 bits in Table 1. Ergo, a different key is generated for each pass indeed. These keys are produced from the essensial key with 56 bits. The production process of this key is shown in Figure 4.



The Figure 4 shows the production of the 56-bit input key is required for each key transition. This process is done in sixteen steps and the table is used for each step produced at that step. For encrypt a cipher text which is encrypted with DES, to give the cipher text with the right key would be enough.

*2.1.1.2 Monoalphabetic Substitution Cipher*

Monoalphabetic substitution cipher is a symetric encryption algorithm based on character.

In cryptography, a substitution cipher is a method of encoding by which units of plaintext are replaced with ciphertext, according to a fixed system; the "units" may be single letters (the most common), pairs of letters, triplets of letters, mixtures of the above, and so forth. The receiver deciphers the text by performing the inverse substitution. Substitution ciphers can be compared with transposition ciphers. In a transposition cipher, the units of the plaintext are rearranged in a different and usually quite complex order, but the units themselves are left unchanged. By contrast, in a substitution cipher, the units of the plaintext are retained in the same sequence in the ciphertext, but the units themselves are altered.

There are a number of different types of substitution cipher. If the cipher operates on single letters, it is termed a simple substitution cipher; a cipher that operates on larger groups of letters is termed polygraphic. A monoalphabetic cipher uses fixed substitution over the entire message, whereas a polyalphabetic cipher uses a number of substitutions at different positions in the message, where a unit from the plaintext is mapped to one of several possibilities in the ciphertext and vice versa.

*2.1.1.3 Hill Cipher*

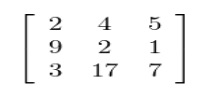
Invented by Lester S. Hill in 1929, the Hill cipher is a polygraphic substitution cipher based on linear algebra. Hill used matrices and matrix multiplication to mix up the plaintext.

To counter charges that his system was too complicated for day to day use, Hill constructed a cipher machine for his system using a series of geared wheels and chains. However, the machine never really sold.

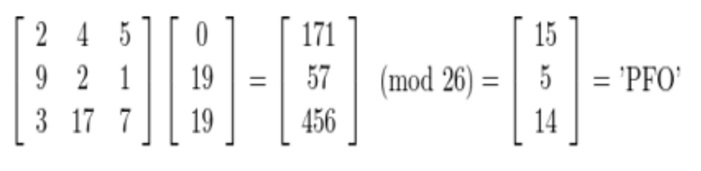
Hill's major contribution was the use of mathematics to design and analyse cryptosystems. It is important to note that the analysis of this algorithm requires a branch of mathematics known as number theory. Many elementary number theory text books deal with the theory behind the Hill cipher, with several talking about the cipher in detail (e.g. Elementary Number Theory and its applications, Rosen, 2000). It is advisable to get access to a book such as this, and to try to learn a bit if you want to understand this algorithm in depth.

***Hill Cipher Example:***

This example will rely on some linear algebra and some number theory. The key for a hill cipher is a matrix e.g.

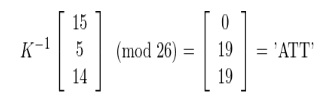


In the above case, we have taken the size to be 3×3, however it can be any size (as long as it is square). Assume we want to encipher the message ATTACK AT DAWN. To encipher this, we need to break the message into chunks of 3. We now take the first 3 characters from our plaintext, ATT and create a vector that corresponds to the letters (replace A with 0, B with 1 ... Z with 25 etc.) to get: [0 19 19] (this is ['A' 'T' 'T']). To get our ciphertext we perform a matrix multiplication (you may need to revise matrix multiplication if this doesn't make sense):



This process is performed for all 3 letter blocks in the plaintext. The plaintext may have to be padded with some extra letters to make sure that there is a whole number of blocks.

Now for the tricky part, the decryption. We need to find an inverse matrix modulo 26 to use as our 'decryption key'. i.e. we want something that will take 'PFO' back to 'ATT'. If our 3 by 3 key matrix is called K, our decryption key will be the 3 by 3 matrix K-1, which is the inverse of K.



To find K-1 we have to use a bit of maths. It turns out that K-1 above can be calculated from our key. A lengthy discussion will not be included here, but we will give a short example. The important things to know are inverses (mod m), determinants of matrices, and matrix adjugates.

Let K be the key matrix. Let d be the determinant of K. We wish to find K-1 (the inverse of K), such that K × K-1 = I (mod 26), where I is the identity matrix. The following formula tells us how to find K-1 given K:



where d × d-1 = 1(mod 26), and adj(K) is the adjugate matrix of K.

d (the determinant) is calculated normally for K (for the example above, it is 489 = 21 (mod 26)). The inverse, d-1, is found by finding a number such that d × d-1 = 1 (mod 26) (this is 5 for the example above since 5\*21 = 105 = 1 (mod 26)). The simplest way of doing this is to loop through the numbers 1.25 and find the one such that the equation is satisfied. There is no solution (i.e. choose a different key) if gcd(d,26) ≠ 1 (this means d and 26 share factors, if this is the case K can not be inverted, this means the key you have chosen will not work, so choose another one).

That is it. Once K-1 is found, decryption can be performed.

*2.1.2 Asymetric Key Algorithms*

Biggest problem has been founded in symmetric key encryption algorithms is distribution .

Distribution of the same key in a multi-user system that uses symmetric algorithm can be problematic from a security perspective. To give different key to each user may be troublesome because there would be more than one key.

Asymmetric encryption algorithms have been developed to unravel these type problems. In asymmetric encryption algorithms and the key decryption key are different from each other. Public key is that used for encryption, the private key is used for decryption. Public key can be distributed to everyone, but should be ensure about which key belongs to who. The main advantage of asymetric encryption algorithms is considered as integrity, authentication and confidentiality services may be provided in a safe manner and this is an important case in cryptograph.

Moreover, the user can determne the key. On the other hand there is some disadvantages of these kind algorithms. For example, there may be occured latency because of the longer length of the key.

## *2.1.2.1 RSA Encryption*

RSA is one of the first practical public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and differs from the decryption key which is kept secret. In RSA, this asymmetry is based on the practical difficulty of factoring the product of two large prime numbers, the factoring problem. RSA is made of the initial letters of the surnames of Ron Rivest, Adi Shamir, and Leonard Adleman, who first publicly described the algorithm in 1977. Clifford Cocks, an English mathematician, had developed an equivalent system in 1973, but it was not declassified until 1997. A user of RSA creates and then publishes a public key based on two large prime numbers, along with an auxiliary value. The prime numbers must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large enough, only someone with knowledge of the prime numbers can feasibly decode the message. Breaking RSA encryption is known as the RSA problem; whether it is as hard as the factoring problem remains an open question.

RSA is a relatively slow algorithm, and because of this it is less commonly used to directly encrypt user data. More often, RSA passes encrypted shared keys for symmetric key cryptography which in turn can perform bulk encryption-decryption operations at much higher speed.

## *2.1.2.2 Hash Functions*

The mission of the hash functions is to produce a unique integer key from a given value. However, to find the appropriate function that will generate a unique number in application area is always difficult or even impossible in some cases. A hashing function must be provide that it have to take any value with any length. The key which has the predetermined length must be produced as output. The process should be one-sided. The value which is given to function should be produced from the function. Hashing function should not allow the collisions.

Collision means that to produce the same key from different values in hashing function.

As collision reslution, there are two main approaches can be applied.

First of them is open adressing. If hashing function produces an index value previously used in the table , another hashing function provides to produce an empty record with the next available index. This process lasts until to find an empty indice.

The second way to get rid of collisions is to use linking list. Records which have the same index can be related to each other by usig linking list.

Naturally, these methods have indivudual advantages and disadvantages.

In the usage of linked list, hashing process can be much faster as the multi-collisions can be unlimited.

*2.1.2.3 SHA Cipher Method*

Secure Hashing Algorithm is a family of cryptographic hash functions. It allows to database management based on hash functions. SHA1 algorithm does only encryption process, decryption can not be applied.

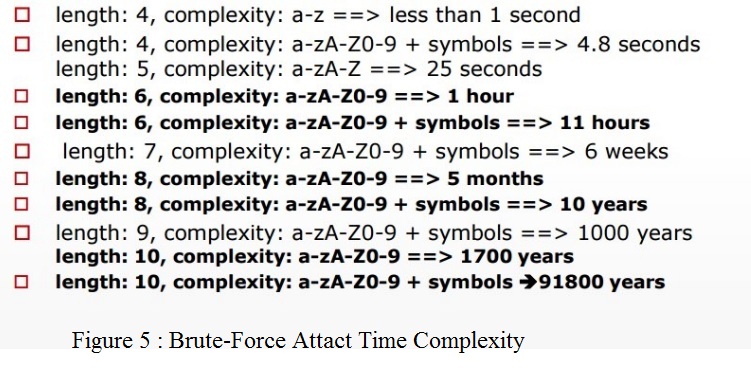
Collision problem is the common problem which is faced with in encryption algorithms.

In SHA algorithms, to barry such problems is tried by several times of giving different keys with different bit sizes.

2.2 Cryptoatack Algorithms

*2.2.1 Brut Force Atack*

In cryptography, a brute-force attack, or exhaustive key search, is a cryptanalytic attack that can be used against any encrypted data. In encryption algorithms, if the plain key can't be estimated, brutforce attack will be suitable to cryptoatack. Process time complexity depends on the counts of keys attempted. And the possible keys are increased according to lenghth and the complexity of keys. The decryption process in the brutforce attacks can lasts so long. You can trace the effect of compexity on time consumption in encryption programsin Figure 5.



When key guessing for modern cryptosystems, the key length used in the cipher determines the practical feasibility of performing a brute-force attack, with longer keys exponentially more difficult to crack than shorter ones. A cipher with a key length of N bits can be broken in a worst-case time proportional to 2n and an average time of half that.

*2.2.2 Known Plantext Attack*

The known-plaintext attack (KPA) is an attack model for cryptanalysis where the attacker has access to both the plaintext and its encrypted version. These can be used to reveal further secret information such as secret keys and code books.

First word of the encrypted text is known. And it would be the main tip to get plain key to decryption.

In cryptography, the known plaintext attack is an attack based on having samples of both the plaintext and corresponding encrypted or ciphertext for that information available. This information is used to conduct an analysis of the data in order to determine the secret key used to encrypt and decrypt the information.

# 3. METHODS

*3.1 An Expanded View on Hill Cypher Algorithm*

In a Hill cipher encryption, the plaintext message is broken up into blocks of length according

to the matrix chosen. Each block of plaintext letters is then converted into a vector of

numbers and is dotted with the matrix. The results are then converted back to letters and

the ciphertext message is produced. For decryption of the ciphertext message, the inverse of

the encryption matrix must be found. Once found, the decryption matrix is then dotted with

each -block of ciphertext, producing the plaintext message.

*3.1.1 Encryption of Hill Cypher Algorithm*

To encrypt a message using the Hill Cipher we must first turn our keyword into a key matrix (a 2 x 2 matrix for working with digraphs, a 3 x 3 matrix for working with trigraphs, etc). We also turn the plaintext into digraphs (or trigraphs) and each of these into a column vector. We then perform matrix multiplication modulo the length of the alphabet (i.e. 26) on each vector. These vectors are then converted back into letters to produce the ciphertext

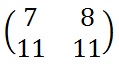
***2 x 2 Example***

We shall encrypt the plaintext message "short example" using the keyword hill and a 2 x 2 matrix. The first step is to turn the keyword into a matrix. If the keyword was longer than the 4 letters needed, we would only take the first 4 letters, and if it was shorter, we would fill it up with the alphabet in order (much like a Mixed Alphabet).

[](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/6927231_orig.jpg)

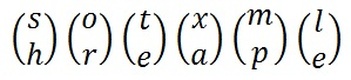
The keyword written as a matrix.

With the keyword in a matrix, we need to convert this into a key matrix. We do this by converting each letter into a number by its position in the alphabet (starting at 0). So, A = 1, B = 2, C= 3, D = 4, etc.

[](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/2127724_orig.jpg)

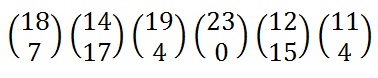
The key matrix

We now split the plaintext into digraphs, and write these as column vectors. That is, in the first column vector we write the first plaintext letter at the top, and the second letter at the bottom. Then we move to the next column vector, where the third plaintext letter goes at the top, and the fourth at the bottom. This continues for the whole plaintext.

[](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/3874131_orig.jpg?357)

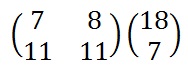
The plaintext “shortexample” split into column vectors.

Now we must convert the plaintext column vectors in the same way that we converted the keyword into the key matrix. Each letter is replaced by its appropriate number.

[](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/9504506_orig.jpg)

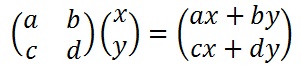
The plaintext converted into numeric column vectors.

Now we must perform some matrix multiplication. We multiply the key matrix by each column vector in turn. We shall go through the first of these in detail, then the rest shall be presented in less detail. We write the key matrix first, followed by the column vector.



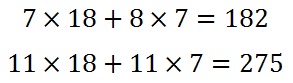
To perform matrix multiplication we "combine" the top row of the key matrix with the column vector to get the top element of the resulting column vector. We then "combine" the bottom row of the key matrix with the column vector to get the bottom element of the resulting column vector. The way we "combine" the four numbers to get a single number is that we multiply the first element of the key matrix row by the top element of the column vector, and multiply the second element of the key matrix row by the bottom element of the column vector. We then add together these two answers.

That is, we follow the rules given by the algebraic method shown below.

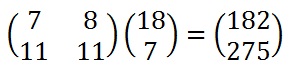
[](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/4654040_orig.jpg)

The algebraic rules of matrix multiplication.

In our case we perform the two calculations on the right. We then right these two answers out in a column vector as shown below.

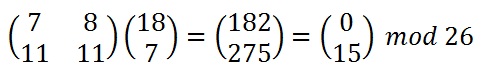
[](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/6927816_orig.jpg)

The calculations performed when doing a matrix multiplication.

[](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/9802105_orig.jpg)

The shorthand for the matrix multiplication.

Next we have to take each of these numbers, in our resultant column vector, modulo 26 (remember that means divide by 26 and take the remainder).

[](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/4006075_orig.jpg)

The whole calculation: converting to numbers; the matrix multiplication; reducing modulo 26; converting back to letters.

*3.1.2 Decryption of Hill Cypher Algorithm*

To decrypt a ciphertext encoded using the Hill Cipher, we must find the inverse matrix. Once we have the inverse matrix, the process is the same as encrypting. That is we multiply the inverse key matrix by the column vectors that the ciphertext is split into, take the results modulo the length of the alphabet, and finally convert the numbers back to letters.

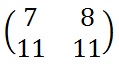
Since the majority of the process is the same as encryption, we are going ot focus on finding the inverse key matrix (not an easy task), and will then skim quickly through the other steps (for more information see Encryption above).

In general, to find the inverse of the key matrix, we perform the calculation below, where K is the key matrix, d is the determinant of the key matrix and adj(K) is the adjugate matrix of K.

[Picture](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/867635_orig.jpg)

General method to calculate the inverse key matrix.

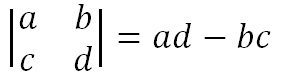
We shall decrypt the example above, so we are using the keyword hill and our ciphertext is "APADJ TFTWLFJ". We start by writing out the keyword as a matrix and converting this into a key matrix as for encryption. Now we must convert this to the inverse key matrix, for which there are several steps.

[](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/___6927231_orig.jpg)[](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/_2127724_orig.jpg)

The key matrix

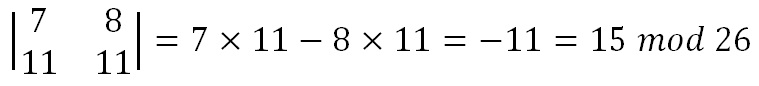
***Step 1 -*** Find the Multiplicative Inverse of the Determinant

The determinant is a number that relates directly to the entries of the matrix. It is found by multiplying the top left number by the bottom right number and subtracting from this the product of the top right number and the bottom left number. This is shown algebraically below. Note that the notation for determinant has straight lines instead of brackets around our matrix.

[](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/4533438_orig.jpg)

Algebraic method to calculate the determinant of a 2 x 2 matrix.

Once we have found this value, we need to take the number modulo 26. Below is the way to calculate the determinant for our example.

[](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/6449828_orig.jpg)Calculating the determinant of our 2 x 2 key matrix.

We now have to find the multiplicative inverse of the determinant working modulo 26. That is, the number between 1 and 25 that gives an answer of 1 when we multiply it by the determinant. So, in this case, we are looking for the number that we need to multiply 15 by to get an answer of 1 modulo 26. There are algorithms to calculate this, but it is often easiest to use trial and error to find the inverse.

[Picture](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/859481_orig.jpg)

If d is the determinant then we are looking for the inverse of d.

[Picture](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/5190131_orig.jpg)

The multiplicative inverse is the number we multiply 15 by to get 1 modulo 26.

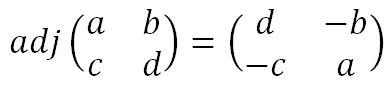
[Picture](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/2704112_orig.jpg)

This calculation gives us an answer of 1 modulo 26.

So the multiplicative inverse of the determinant modulo 26 is 7. We shall need this number later.

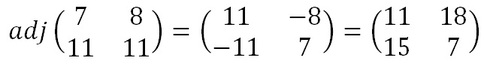
***Step 2 -*** Find the Adjugate Matrix

The adjugate matrix is a matrix of the same size as the original. For a 2 x 2 matrix, this is fairly straightforward as it is just moving the elements to different positions and changing a couple of signs. That is, we swap the top left and bottom right numbers in the key matrix, and change the sign of the the top right and bottom left numbers. Algebraically this is given below.

[](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/2475976_orig.jpg)

The adjugate matrix of a 2 x 2 matrix.

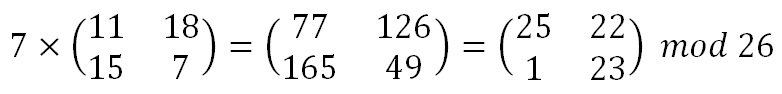
Again, once we have these values we will need to take each of them modulo 26 (in particular, we need to add 26 to the negative values to get a number between 0 and 25. For our example we get the matrix below.

[](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/8133953_orig.jpg?493)

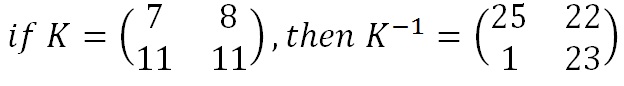
The adjugate matrix of the key matrix.

***Step 3 -*** Multiply the Multiplicative Inverse of the Determinant by the Adjugate Matrix

To get the inverse key matrix, we now multiply the inverse determinant (that was 7 in our case) from step 1 by each of the elements of the adjugate matrix from step 2. Then we take each of these answers modulo 26.

[](http://crypto.interactive-maths.com/uploads/1/1/3/4/11345755/3227655_orig.jpg)

That is:



Now we have the inverse key matrix, we have to convert the ciphertext into column vectors and multiply the inverse matrix by each column vector in turn, take the results modulo 26 and convert these back into letters to get the plaintext.

3.1.3 Attact to be applied: Known-Plaintext Attact

We are assuming that this message was encrypted using a 2 2 × Hill cipher and that we have a crib. We believe that the message begins “hill.”

H i| l l

[8,9] | [12,12]

g q| f x

[7,17] | [6,24]

We could either solve for the key or the key inverse. To solve for the key, we would solve

=

And

=

To solve for the key inverse, we would solve

=

And

=

We will solve for the key.

= represents two linear equations: 8a + 9b =7

8c + 9d = 17

Now we solve the following linear congruences mod 26

8a + 9b =7 8c + 9d = 17

12a+12b =6 and 12c + 12d= 24

We will solve the pair of congruences 8a + 9b =7

12a+12b =6 first.

To eliminate an unknown, multiply congruence1 by 3,2 by 2.

24a + 27b = 21

24a + 24b = 12

and subtract congruence 1 from congruence 2.

3b =9 b=3

8a + ( 9\*3) =7

8a = - 20 = 6mod26

a= 5\*8a = 5 \*6 = 30 =4mod26

the key is looks like

Now solve the system 8c + 9d =17

12c + 12d =24

multiply congruence 1 by 3,2 by 2; 24c + 27d =51

24c + 24d =48

and subtract congruence 1 from congruence 2.

3d =3 d =1

From congruence 2, 24c +24 =48 c =1

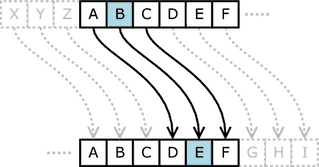
The key is

***Discussion***

The most important item that must be discussed regarding the use of the Hill Cipher is that not every possible matrix is a possible key matrix. This is because, in order to decrypt, we need to have an inverse key matrix, and not every matrix is invertible. Fortunately, we do not have to work out the entire inverse to find it is not possible, but simply consider the determinant. If the determinant is 0 or shares a factor, other than 1, with the length of the alphabet being used, then the matrix will not have an inverse. If this is the case, a different key must be chosen, since otherwise the ciphertext will not be able to be decrypted. In order to be a usable key, the matrix must have a non-zero determinant which is coprime to the length of the alphabet.

*3.2 Ceasar Cipher*

Caesar cipher is categorized as a substitution cipher in which the alphabet in the plain text is shifted by a fixed number down the alphabet.

****

*3.2.1 Encryption-Decryption of Ceasar Algorithm*

Encryption of a letter x by a shift n can be described mathematically as

En(x)=(x+n)\ mod {26}.

Decryption is performed similarly,

Dn(x)= (x-n) \ mod {26}

With Key=13,

Plaintext: ybu bilgisayar mühendisliği

Ciphertext:loh ovytvfnlne züuraqvfyvtv Or

With key =13,

Ciphertext:loh ovytvfnlne züuraqvfyvtv

Plaintext: ybu bilgisayar mühendisliği

*3.2.2 The attact to be applied :The Brute-Force Attack*

Nothing stops a cryptanalyst from guessing one key decrypting the ciphertext with that key, looking at the output,ifit is not in dictionary ,the key is not the correct key then program move to the next key. The technique of trying every possible decryption key is called a brute-force attack. It isn’t a very sophisticated , but through sheer effort (which the computer will do for us) the Caesar cipher can be broken.

# 4. CONCLUSION

As a result of this research, the principle of cryptography has been learnt. Cryptology techniques are experienced in detail. The encryption and decryption algorithms of Hill cipher, monoalphabetic and polyalphabetic cipher and playfair cipher have been executed and traced the processes.

The key principle of related-key attacks are based on estimating the right key and until find it produces all possible keys. The codes which are writtten for this project attempts to generate right key to decrypt the encrypted cypher text into plain text.

The last step of this research was to apply Brute-force and Known Plain-Text Attack crypt attack programs on two encryption code with Hill Cipher and Ceasar algorithms. The results of these attacks and the discussion is present in method part of this paper. This work was instructive in the security aspect.

# 5. REFERENCES

1. Canetti R, Halevi S, & Katz J. A Forward-Secure Public-Key Encryption Scheme. Retrieved June 6, 2016, from [http://download.springer.com/static/pdf/601/chp%253A10.1007%252F3-540-39200-9\_16.pdf?originUrl=http%3A%2F%2Flink.springer.com%2Fchapter%2F10.1007%2F3-540-39200-9\_16&token2=exp=1465182484~acl=%2Fstatic%2Fpdf%2F601%2Fchp%25253A10.1007%25252F3-540-39200-9\_16.pdf%3ForiginUrl%3Dhttp%253A%252F%252Flink.springer.com%252Fchapter%252F10.1007%252F3-540-39200-9\_16\*~hmac=0aabca281d8c8773677ac797e77e2830890effa7b2bba40d3df9525f655d4905](http://download.springer.com/static/pdf/601/chp%253A10.1007%252F3-540-39200-9_16.pdf?originUrl=http%3A%2F%2Flink.springer.com%2Fchapter%2F10.1007%2F3-540-39200-9_16&token2=exp=1465182484~acl=%2Fstatic%2Fpdf%2F601%2Fchp%25253A10.1007%25252F3-540-39200-9_16.pdf%3ForiginUrl%3Dhttp%253A%252F%252Flink.springer.com%252Fchapter%252F10.1007%252F3-540-39200-9_16*~hmac=0aabca281d8c8773677ac797e77e2830890effa7b2bba40d3df9525f655d4905)
2. Known Plain Tect Attact. Retrieved June 06, 2016, from <https://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/1_Monoalph/knownplain.html>
3. Eşberk, Ş.. 6 Haziran 1993, <http://safakesberk.com/2014/09/10/turk-alfabesi-ile-sezar-sifreleme-brute-force-ile-sifre-cozme/>
4. Ceasar Cypher Cryptography. Retrieved June 06, 2016, from <http://crypto.stackexchange.com/questions/30741/breaking-a-modified-caesar-cipher>
5. Known Plaintext Attack Example. Retrieved June 06, 2016, from <http://www.nku.edu/~christensen/092mat483%20known%20plaintext%20attack%20of%20Hill%20cipher.pdf>
6. Crypto Corner. Retrieved June 06, 2016, from <http://crypto.interactive-maths.com/hill-cipher.html>
7. Stinson, D.R., 2002. Cryptography, Chapman & Hall/CRC Press Company, United States of America.
8. Hsu, Y.C., Tsai, K.F., Liu J.T. and Lin, E.S., 1995. VHDL Modeling For Digital Synthesis, Kluwer Academic Publishers, Riverside.
9. Savaş, E., 1994. Dizi Şifreleme Sistemleri ve Doğrusal Karmaşıklık, Yüksek Lisans Tezi, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
10. Menezes, A., Van Oorschot, P.,Vanstone, S., 1996. Handbook of Applied Cryptography, CRC Press Company.
11. Apohan, A.M., 1993. Kriptoloji, Yüksek Lisans Tezi, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
12. Braeken, A., Lano, J., Mentens, N., Prenel, B., Verbauwhede, I., 2005. SFINKS :A synchronous Stream Cipher for Restricted Hardware Environments , Belgium.
13. Brunner H., Curiger A., Hofstetter M. , 1993. On Computing Multiplicative İnverses in GF(2m ), IEEE Transactions on Computers, 42, 1010-1015.