

# Hill Ciphers

Todd Douglas and Dustin Helliwell  
Math 45 — College of the Redwoods

December 19, 1997

## 1 Introduction

Lester Hill introduced his coding/decoding process in 1929 in the journal of mathematics, the process became known Hill-ciphers. The process of encryption is called cryptography. In cryptography a message that has not yet been encrypted is called plaintext, after the encryption process the encrypted message is called ciphertext. The process of converting the plaintext to ciphertext is called enciphering and the reverse process where the ciphertext is converted to plaintext is called deciphering. Lester's method involves dividing the plaintext message into sets of  $n$  letters, each of which is replaced by  $n$  cipher letters, this is known as a polygraphic system. Hill-ciphers require a matrix based polygraphic system. For example  $\{abcdef \dots\} = ab\ bc\ de \dots$  or  $abc\ def \dots$  and so on.

For the purpose of our discussion we will show the classic plaintext and ciphertext conversion

A	B	C	D	E	F	G	H	I	J	K	L	M
1	2	3	4	5	6	7	8	9	10	11	12	13
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25	0

- Step 1. Choose an invertible modulo 26  $n \times n$  matrix with integer entries

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

to perform the encoding. (Certain conditions on  $a$  will be imposed later.)

- Step 2. Group successive plaintext letters into sets of  $n$  letters, adding an arbitrary "dummy" letter to fill out the last set if the plaintext does not have the same grouping number of letters, then replace each plaintext letter by its numerical value.

- Step 3. Successively convert each plaintext pair  $p_1, p_2, \dots, p_n$  into a column vector

$$\mathbf{p} = \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix}$$

and form the product  $Ap = c$ . We will call  $p$  a plaintext vector and  $Ap$  the corresponding ciphertext vector.

- Step 4. Convert each ciphertext vector into its alphabetic equivalent using the conversion table.

### 1.1 Example

We will encipher, "DAVE", for this example. The first step includes the choice of an invertible modulo 26  $n \times n$  matrix, for a hill 2-cipher we use the  $2 \times 2$  matrix

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}$$

Next group plaintext letters into pairs and replace with the corresponding numerical value from the classic conversion table.

$$\begin{bmatrix} D \\ A \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \end{bmatrix} \text{ and } \begin{bmatrix} V \\ E \end{bmatrix} = \begin{bmatrix} 22 \\ 5 \end{bmatrix}$$

use these to form the product

$$Ap = c$$

$$\begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} * \begin{bmatrix} 4 & 22 \\ 1 & 5 \end{bmatrix} = \begin{bmatrix} 6 & 32 \\ 3 & 15 \end{bmatrix}$$

Now

$$c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

so,

$$c_1 = \begin{bmatrix} 6 \\ 3 \end{bmatrix} \text{ and } c_2 = \begin{bmatrix} 32 \\ 15 \end{bmatrix}$$

This is where we run into trouble, 32 is larger than our 26 letter alphabet so, to account for this we replace integers that are larger than 25 by the remainder that results from dividing by 26.  $32/26 = 1$  remainder 6, so,

$$c_2 = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$$

$$c_1 = \begin{bmatrix} 6 \\ 3 \end{bmatrix} = [FC]' \text{ and } c_2 = \begin{bmatrix} 24 \\ 7 \end{bmatrix} = [FO]'$$

so, the encrypted message is "FCFO".

## 1.2 Modular Arithmetic

In the last example we discovered that integers greater than 25 are replaced by their remainders after dividing by 26. Working with remainders is at the core of a body of mathematics known as modular arithmetic. Because of its importance in cryptography we will digress for a moment and focus on some important points, if we were to explain modular arithmetic at any length the original topic, Hill Ciphers, would be taken to far afield.

- Definition: If  $M$  is a positive integer and  $a$  and  $b$  are any integers then we say that  $a$  is congruent to  $b$  modulo  $m$ , written

$$a \equiv b \pmod{m}$$

if  $a - b$  is an integer multiple of  $m$ , or

$$a \equiv b \pmod{m} \Leftrightarrow m|(b - a)$$

For example:

$$32 \equiv 2 \pmod{6} \text{ because } 6|(32 - 2)$$

$$48 \equiv 3 \pmod{5} \text{ because } 5|(48 - 3)$$

- Theorem 1: For any integer  $a$  and modulus  $m$ , let  $R$ =remainder of  $\frac{|a|}{m}$ . Then, the residue  $r$  of  $a$  modulo  $m$  is

$$\begin{cases} R & \text{if } a \geq 0 \\ m - r & \text{if } a < 0 \text{ and } R \neq 0 \\ 0 & \text{if } a < 0 \text{ and } R = 0 \end{cases}$$

In modular arithmetic we have the following corresponding concept:

- Definition: If  $a$  is a number in  $Z_m$  then a number  $a^{-1}$  in  $Z_m$  is called a reciprocal or multiplicative inverse of  $a$  modulo  $m$  if  $aa^{-1} = a^{-1}a = 1 \pmod{m}$

example:

$$3x = 1 \pmod{26}$$

$$3 * 9 = 27 = 1$$

$$3^{-1} = 9 \pmod{26}$$

- Theorem 2: A square matrix  $A$  with entries in  $Z_m$  is invertible modulo  $m$  if and only if the residue of  $\det(A)$  modulo  $m$  has a reciprocal modulo  $m$ .
- Corollary 1: A square matrix  $A$  with entries in  $Z_m$  is invertible modulo  $m$  if and only if  $m$  and the residue of  $\det(A)$  modulo  $m$  have no common prime factors.
- Theorem 3: let  $a$  and  $b$  be positive integers and let  $x$  and  $y$  be scalars; then the greatest common divisor of  $a$  and  $b$  is  $c$ .  $c$  can be written as a linear combination of  $a$  and  $b$

$$c = xa + yb$$

- Theorem 4: If the square matrix  $A$  satisfies corollary 1, then the greatest common divisor of  $\det(A)$  with  $m$  is 1. Now applying theorem 3, where  $a = \det(A)$ ,  $b = m$  and  $c = 1$ .

$$1 = x \det(A) + ym$$

then the modular inverse of  $\det(A)$  is  $x$ .

Note:  $x$  can be found using some techniques of number theory.

### 1.3 Deciphering

Deciphering the ciphertext of a hill cipher requires the inverse modulo  $m$  of the enciphering matrix. Suppose

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

is invertible modulo 26 and this matrix is used in hill 2-cipher if

$$p = \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix}$$

is a plaintext vector, then

$$c = Ap$$

is the corresponding ciphertext vector and

$$P = A^{-1}c$$

Thus, plaintext can be recovered from the corresponding ciphertext vector by multiplying it on the left by

$$A^{-1}(\text{mod } m)$$

### 1.3.1 Example:

This is the table of reciprocals modulo 26

$$\begin{array}{r} a \\ a^{-1} \end{array} = \begin{array}{cccccccccccc} 1 & 3 & 5 & 7 & 9 & 11 & 15 & 17 & 19 & 21 & 23 & 25 \\ 1 & 9 & 21 & 15 & 3 & 19 & 7 & 23 & 11 & 5 & 17 & 25 \end{array}$$

Starting with the original encoding matrix

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}$$

take the inverse

$$A^{-1} = 3^{-1} \begin{bmatrix} 3 & -2 \\ 0 & 1 \end{bmatrix}$$

from the table of reciprocals we find the inverse of 3 is 9, and apply modular arithmetic.

$$A^{-1} = 9 \begin{bmatrix} 3 & 24 \\ 0 & 1 \end{bmatrix} (\text{mod } 26)$$

multiply, and we get

$$A^{-1} = \begin{bmatrix} 27 & 216 \\ 0 & 9 \end{bmatrix}$$

once again apply modular arithmetic and

$$A^{-1} = \begin{bmatrix} 1 & 8 \\ 0 & 9 \end{bmatrix} \implies \text{This is our enciphering matrix}$$

so

$$\begin{bmatrix} 1 & 8 \\ 0 & 9 \end{bmatrix} \begin{bmatrix} 6 & 6 \\ 3 & 15 \end{bmatrix} = \begin{bmatrix} 30 & 126 \\ 27 & 135 \end{bmatrix}$$

apply modular arithmetic once again, and use the plaintext conversion table to decode the matrix.

$$\begin{bmatrix} 4 & 22 \\ 1 & 5 \end{bmatrix} = \text{DAVE}$$

The following code was used for a program that was part of a presentation for a linear algebra class, the program is used for encryption and decrypting in ASCII characters, where modulo 256 is used. these files can possibly be sent through E-mail and used by two parties to send coded messages if the need arises. These are the M-files for the program.

- Encrypt

```

function [ciphertext] = encrypt(plaintextfile,ciphertextfile,dim)
%ENCRYPT encrypts a text file.
% ENCRYPT(PLAINTEXTFILE,CIPHERTEXTFILE,DIM) opens a plaintext file with the specified
% name, creates a ciphertext file using the specified name and creates a square
% encoding matrix of dimension DIM were DIM is no greater than 6. If no dimension
% is specified the default encoding matrix [105 18;27 171] is used.
%
% A copy of the encoding matrix is saved in a file encodematrix.ypt this file will
% change every time ENCRPT is used unless the default matrix is used for encoding.
%
% See also DECRYPT, ..

% 1.Either creates a random encoding matrix or uses the default matrix.
if nargin<3 encodematrix=[105 18;27 171];dim=length(encodematrix);
else
    g=2*dim;
while g~=1
    encodematrix=mod(randint(dim,dim,255),256);
    [g c d]=gcd(mod(det(encodematrix),256),256);
end
end
encodematrix

%2.Creates a file with the encoding matrix.
z=fopen('encodematrix.ypt','w');
y=fwrite(z,encodematrix,'char');

%3.Opens the plaintext file.
a=fopen(plaintextfile,'r');
b=fread(a,inf,'char');

%4.Reshapes the plaintext file for left multiplication by the encoding matrix.
O=length(b);
if mod(length(b),length(encodematrix))==0
else
    for R=1:(length(b)-mod(length(b),length(encodematrix))+dim)-length(b)
        b(O+R,1)=32;
    end
end

%5.Encodes the plaintext file by left multiplying it with the encoding matrix.
plaintext=char(b)';
encodematrixmatrix=reshape(b,length(encodematrix),length(b)/length(encodematrix));

```

```

codedmatrix=encodematrix*encodematrixmatrix;
c=mod(round(reshape(codedmatrix,dim*size(codedmatrix,2),1)),256);
ciphertext=char(c)';
%6.Creates a file containing the ciphertext.
f=fopen(ciphertextfile,'w');
g=fwrite(f,ciphertext,'char');
h=fopen(ciphertextfile,'r');
i=fread(h,inf,'char');
fclose('all');

```

## • Decrypt

```

function [plaintext] = decrypt(ciphertextfile,plaintextfile)
%DECRYPT Decrypts a textfile.
%   DECRYPT(CIPHERTEXTFILE,PLAINTEXTFILE) opens the specified ciphertext
%   file then decrypts the ciphertext and creates a plaintextfile with the specified name.
%
%   The file 'encodematrix.ypt' created with the original ciphertext file
%   must be included.
%
%   See also ENCRYPT, :.

%1.Opens the file 'encodematrix.ypt' which is the encoding matrix.
z=fopen('encodematrix.ypt','r');
t=fread(z,inf,'char');
encode=reshape(t,sqrt(length(t)),sqrt(length(t)));

%2.Opens ciphertext file to be decrypted.
h=fopen(ciphertextfile,'r');
i=fread(h,inf,'char');

%3.Reshapes the ciphertextfile (still in ASCII code) for left multiplication by
%the decoding matrix.
decodematrix=reshape(i,length(encode),length(i)/length(encode));

%4.Creates the decoding matrix by taking the modular inverse of the encoding matrix.
[g c d]=gcd(mod(det(encode),256),256);
if g~=1,disp('invalid encoding matrix'),return
end
encode;
cofactor(encode);
decode=mod(c*ans',256)

%5.Left multiplies the-now reshaped-chiphertextfile by the decoding matrix.
decoded=mod(decode*decodematrix,256);

```

```

%6.Reshapes the variable, 'decoded' into a vector and converts it from ASCII code to
%ASCII characters.
e=reshape(decoded,length(decode)*length(decoded),1);
plaintext=char(e)';

%7.Creates a file containing the plaintext.
f=fopen(plaintextfile,'w');
fwrite(f,plaintext,'char');
fclose('all');

```

- Cofactor

The following program Created by Gilbert Strang of MIT is used by the two programs listed above.

```

function C = cofactor(A,i,j)
%COFACTOR Cofactors and the cofactor matrix.
%      COFACTOR(A,i,j) returns the cofactor of row i, column j.
%      COFACTOR(A) returns the matrix C of cofactors.

if nargin == 3
    % Remove row i and column j to produce the minor.
    M = A;
    M(i,:) = [];
    M(:,j) = [];
    C = (-1)^(i+j)*det(M);
else
    [n,n] = size(A);
    for i = 1:n
        for j = 1:n
            C(i,j) = mod(cofactor(A,i,j),26);
        end
    end
end
end

```

## 1.4 References

1. Abraham Sinkov, Elementary Cryptanalysis, a Mathematical Approach (Mathematical Association of America, Mathematical Library, 1966).
2. Alan G. Konheim, Cryptography, a primer (New York: Wiley-Interscience, 1981)
3. Gilbert Strang; M-file: MIT
4. David Arnold, Modular Arithmetic, Inverse Multiplication, Hill ciphers: Collage of the Redwoods Mathematics Department.

5. Howard Anton and Chris Rorres, Elements of Linear Algebra, (Drexel University, New York John Wiley & sons, Inc1973).

6 David C. Lay, Linear Algebra and its Applications; second addition (university of Maryland, Addison- Wesley Longman, Inc.,1997)

**Further Readings** 1. Lester S. Hill, Cryptography in an Algebraic Alphabet, (The American Mathematical Monthly, 36), June-July 1929, pp. 306-312.

2. Lester S. Hill, Concerning Certain Linear Transformation Apparatus of Cryptography, (The American Mathematical Monthly, 38), March 1931, pp.135-154).