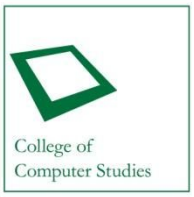


De La Salle University
College of Computer Studies
Software Technology Department

ADVDISC - Machine Project

Application of Matrices in Classical and Modern Cryptography

Section	S17
Group Members	Cardano, Marc Daniel T. Garcia, Markus Jeremi C. Lucas, John Martin G.

Table of Contents

Introduction

Cryptography is defined as the study of obscuring information (plaintext) into ciphertext (encryption) and vice versa (decryption). In cryptography, algorithms that encrypt and decrypt are called 'ciphers'. The cipher's algorithm determines how plaintext is encrypted and decrypted, and with some ciphers, a secret key, known by the communicants, is used in encryption and decryption.

Brief History on Cryptography

Caesar Cipher

The Caesar Cipher is one of the earliest known and simplest cipher. The cipher was named after the Roman General, Julius Caesar.The Cipher itself was mainly used as a tactic in his many military campaigns, to send messages throughout his army. No proof existed that the cipher itself was secure at Caesar's time but this cipher can be easily be cracked today and thus offer no security today.

ROT-13 Cipher

The Rotate by 13 (ROT-13) Cipher works like the aforementioned Caesar Cipher and just like the Caesar Cipher, it provides little or virtually no security whatsoever.The ROT-13 is mainly used in today's modern media to hide spoilers, puzzle answers, offensive materials and the like from a casual glance. It was dubbed as the "Usenet equivalent of a magazine printing the answer to a quiz upside down" due to its utter simplicity.

Hill Cipher

Hill Cipher was invented by Lester S. Hill on 1929. The cipher uses linear algebra and matrix operations in encryption and decryption. This cipher was considered as too complex for everyday use. The cipher was also considered to have low security since it only uses linear operation such as matrix multiplication, but its security can be improved combined along other non-linear operations.

Playfair Cipher

It is a cipher which was widely known because of the heavy promotion of Lord Playfair despite the fact that it was not of his own

invention. The Playfair cipher was an invention of Charles Wheatstone, who first described it in 1854. It was used for tactical purposes by British forces in the Second Boer War and in World War I and for the same purpose by the British and Australians during World War II. This was because the Playfair cipher was reasonably easy to do and does not need any special equipment to do, just pencil and paper. It was used for hiding non-critical information. So if the information on the encrypted message was to be broken, it would have already been useless for the enemy. The encryption is relatively difficult to break since the frequency analysis method does not work with it.

Block Cipher

The Block Cipher is one of the modern ciphers and is being used today in global communication (i.e. internet). There are many different implementations of block ciphers such as the Feistel Cipher by Horst Feistel. This cipher has been the best in maintaining security, authenticity, and privacy today.

Design

Cipher Alphabet

Before designing a cipher, an alphabet must be established. The alphabet contains the set of characters or symbols that the cipher can read. Every symbol will be assigned a numerical value as shown in the table below:

A	B	C	D	E	F	G	H	I	J
1	2	3	4	5	6	7	8	9	10

These numerical values are then what the ciphers use in their encryption or decryption process.

Caesar Cipher

For the Caesar Cipher, you only need to know what the order of the cipher alphabet is, then after that you are all set to encrypt and decrypt using the Caesar Cipher. To encode using the cipher, shift each character in the plaintext by ***n*** numbers to the right or left in the order of the defined Cipher

Alphabet, with ***n*** being the key for the cipher. To Decode, use the same action except that you do it with the opposite direction on how you encoded.

ROT-13 Cipher

The ROT-13 Cipher behaves similarly like the Caesar cipher except that the characters are shifted by a fixed number, 13. To decode a ciphertext encrypted by a ROT-13 cipher, simply shifting 13 more will return the plaintext.

Hill Cipher

Hill ciphers require a key matrix A for encryption and decryption. A must be a square matrix. Given that:

$$A = \begin{bmatrix} 7 & -3 & -3 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Given a plaintext "CABBAGE", to encrypt the plaintext, the plaintext has to be broken up into chunks of 3 (because the key is a 3x3 matrix). After breaking up the plaintext, create vectors containing those chunks. The vectors would look like:

$$'CAB' = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}, 'BAG' = \begin{bmatrix} 2 \\ 1 \\ 7 \end{bmatrix}, 'E' = \begin{bmatrix} 5 \\ 0 \\ 0 \end{bmatrix}$$

The vectors contain the numerical values of each character defined by the cipher alphabet. Simply perform Matrix multiplication to the key and each of the vectors and the cipher will return the following result.

$$\begin{bmatrix} 7 & -3 & -3 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 12 \\ -2 \\ -1 \end{bmatrix} mod(26) = \begin{bmatrix} 12 \\ 2 \\ 1 \end{bmatrix} = 'LBA'$$

'CAB' will be encrypted into *'LBA'*. Repeat this process to the other vectors.

For decryption, the inverse of the key will be used rather than the key. In this case, the inverse of A is:

$$A^{-1} = \begin{bmatrix} 1 & 3 & 3 \\ 1 & 4 & 3 \\ 1 & 3 & 4 \end{bmatrix}$$

After finding the inverse of the key, simply repeat the process of encryption using the ciphertext and the inverted key.

Playfair Cipher

There are many variants of the Playfair cipher, so to make it simple, we modified the original Playfair and made it possible for the cipher to accept spaces, symbols and lowercase letters. As the original Playfair we implemented could only accept Uppercase letters. The Playfair cipher has three main steps: the first is to make the Key Matrix, chunk the message into pairs of two characters, and then encode/decode. For making the key matrix, one must create a matrix big enough to accommodate the whole defined cipher Alphabet and fill the whole matrix. Now to fill this matrix one must first input the keyword in the first free empty cells of the matrix. Now for simplicity's sake, an example of a 5x5 matrix with a cipher alphabet of: A-Z(all uppercase letters), and a keyword: "KEY". The resulting matrix would be like this:

$$\begin{bmatrix} K & E & Y & A & B \\ C & D & F & G & H \\ H & I & J & L & M \\ N & O & P & Q & R \\ S & T & U & V & W \end{bmatrix}$$

This matrix' sole purpose is to only show how the Key Matrix is formed. Each element in the matrix must have a unique value from the cipher alphabet so it means that the matrix must not contain any repeating characters. In the figure above, we added the Characters of the keyword "key" first then proceeded to add the other letters, skipping the letters in the keyword.

Now after creating the key matrix, the next step would be to chunk the plaintext string into pairs of two characters per pair. Essentially the chunking would need the plaintext string to have an even number of characters so there

are balanced pairs of two characters. So if the total count of the characters of the message is odd, we append a NULL character at the end, thus making the string's character count even. For an example we use SECRETSTUFF as the

SECRETSTUFF

SECRETSTUFF[null]

'SE' 'CR' 'ET' 'ST' 'UF' 'F[null]'

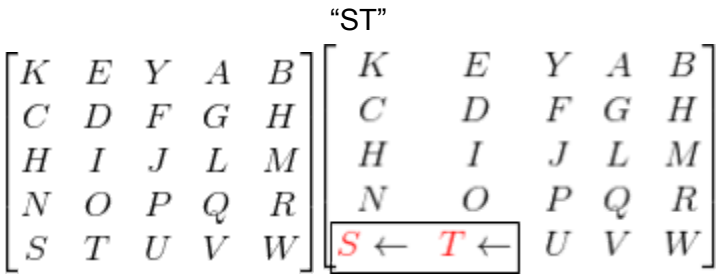
After the chunking of the plaintext is complete we go and start on the encoding process. The algorithm works on each character pair, after locating the letters in the pair in the key matrix we look on the three rules to follow in the Playfair cipher encoding process. (1)If the letters does not appear on the same row or column, then the two letters would have made a rectangular shape on the Key matrix. An example:

"SE"

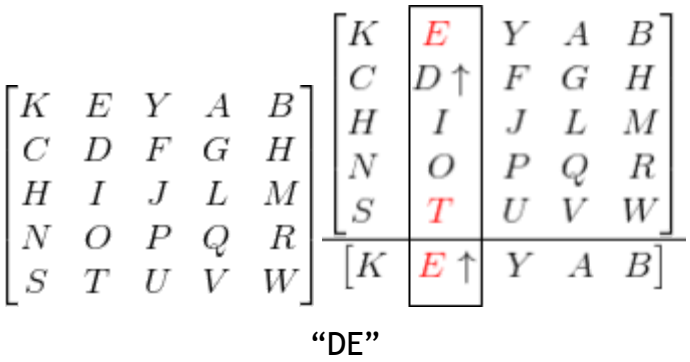
$$\begin{bmatrix} K & E & Y & A & B \\ C & D & F & G & H \\ H & I & J & L & M \\ N & O & P & Q & R \\ S & T & U & V & W \end{bmatrix} \quad \begin{bmatrix} K \leftarrow & E & Y & A & B \\ C & D & F & G & H \\ H & I & J & L & M \\ N & O & P & Q & R \\ S \rightarrow & T & U & V & W \end{bmatrix}$$

"TK"

If you find a pair following this rule you just swap the characters at the opposite end of the rectangle the original pair makes. The resulting encoded pair from the pair "SE" would then be "TK". (2)If the letters appear in the same row of the Key matrix then all you have to do is to replace them with the character to their immediate right. If the character is in the last element in the row then just simply wrap around the whole row. Example: "ST"



Now here, the letter pair ST when encoded will become "TU" since based form the figure above the character form the right replaced the original letter pair.(3) if the pair appears in the same column then just replace them with the letters below them, if the letter is at the most bottom of the column then we just wrap around the column. Example: "ET".



As you can see from the figure above, the column wrapped around the matrix to supply the replacement of the character 'T'. Thus, the encoded result of the letter pair "ET" is now "DE". To decode an encrypted text one would need the key matrix and then just simply do the encoding again except this time the actions on replacing would be the inverse.

Block Cipher

Similar to Playfair cipher and Hill cipher, Block cipher also requires a key for encryption and decryption. Block ciphers use fixed-length group of bits, called blocks, along with the given key to encrypt a given message. Unlike stream ciphers, where bits are encrypted one at a time, block ciphers encrypt and decrypt one block at a time. The encryption process uses bitwise operations in its algorithms. The more bitwise operations used, the more secure the encryption.

To decrypt a message encrypted using a Block cipher, use the same algorithm used to encrypt the message along with the same key. The resulting message should be th

Implementation

Programming Language

For our implementation, the Java Programming Language was used. All the members are very proficient in the language making it easier for the members to collaborate on the project. Also, MCO1's Matrix class was reused for this project because all of the operations we needed such as Gauss-Jordan Reduction and basic matrix operations, for this project was already coded in MCO1's Matrix class.

Cipher Alphabet

For our implementation, we have decided to only include 100 out of 256 characters in the ASCII table. We chose these characters based on how frequent they are used. The alphabet consists of alphanumeric characters, punctuation characters, mathematical symbols, and the null character. Below is the complete table of characters we used along with their corresponding numerical value. The alphabet is stored in a one-dimensional array and the indices serve as the numerical value of the characters.

NUL	A	B	C	D	E	F	G	H	I
0	1	2	3	4	5	6	7	8	9
J	K	L	M	N	O	P	Q	R	S
10	11	12	13	14	15	16	17	18	19
T	U	V	W	X	Y	Z	a	b	c
20	21	22	23	24	25	26	27	28	29
d	e	f	g	h	i	j	k	l	m
30	31	32	33	34	35	36	37	38	39
n	o	p	q	r	s	t	u	v	w

40	41	42	43	44	45	46	47	48	49
x	y	z	1	2	3	4	5	6	7
50	51	52	53	54	55	56	57	58	59
8	9	0	[	]	;	‘	,	.	/
60	61	62	63	64	65	66	67	68	69
-	=	~	!	@	#	$	%	^	&
70	71	72	73	74	75	76	77	78	79
*	(	)	_	+	{	}	|	:	“
80	81	82	83	84	85	86	87	88	89
<	>	?	\	SPACE	≈	π	±	÷	√
90	91	92	93	94	95	96	97	98	99

Caesar Cipher

For Caesar Cipher, we decided to apply matrix operations. The plaintext is first transformed into a row matrix, we shall use the plaintext “secret message” as an example. The resulting row matrix would look like

$$[45\ \ 31\ \ 29\ \ 44\ \ 31\ \ 46\ \ 94\ \ 39\ \ 31\ \ 45\ \ 45\ \ 27\ \ 33\ \ 31]$$

Once the plaintext is transformed into a matrix, a key matrix with the same dimensions as the resulting matrix is created containing the key value. In our implementation, the key value is 10. Add the two matrices and the sum matrix will be the ciphertext.

For decryption, the same algorithm is used except the matrices are subtracted instead.

ROT-13 Cipher

For this cipher, the Caesar cipher was used and the key value is set to 13.

Hill Cipher

Before a plaintext was to be encrypted, it had to be separated into chunks. The size of these chunks are dependent on the ***n x n*** key matrix. Once the plaintext was separated, these values were put into a ***n x m*** matrix where m is the number of chunks.

The encryption of the Hill Cipher was implemented to return a matrix with the values of the ciphertext. The matrix will check against the the cipher alphabet to see what character each value corresponds to. For decryption, the same matrix will be passed to the cipher and will return another matrix containing values of the decrypted text. The key we decided to use for this cipher is a 3x3 matrix A, where:

$$A = \begin{bmatrix} 7 & -3 & -3 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

This matrix was used because its inverse are all integers with no floating point values, making it more accurate for encryption and decryption. The inverse of matrix A which was used for the decryption is:

$$A^{-1} = \begin{bmatrix} 1 & 3 & 3 \\ 1 & 4 & 3 \\ 1 & 3 & 4 \end{bmatrix}$$

Playfair Cipher

Like all the other Ciphers in our program the Playfair cipher uses the same Cipher Alphabet. That is why in our own implementation of the Playfair cipher, so to accommodate the whole entirety of our Cipher Alphabet a 10x10 matrix is needed. At the start of every Playfair Cipher, the plaintext would be processed. In the process of preparing the plaintext to be encrypted, we needed to first make sure that the plaintext has an even number of total characters, if not we simply append a '\0' or a NULL at the end of the plaintext. Afterwards, we then proceed to chunk the plaintext to its required chunk size of 2. Meaning that the message is to be split into pairs of two characters. That was why we needed to make sure that the total number of characters was even.

Now after chunking the Plaintext comes the creation of the Key matrix which will serve as main key to be used when encrypting. The key is created with the keyword "PlayFairrox123!@#". Then the next step is to put each

character of the keyword to the Key matrix, only when the whole keyword has been put to the matrix will the program put the rest of the Cipher Alphabet in the key Matrix. We also made sure that in the creation of the key matrix, it would only put a unique character in each cell of the matrix, ignoring already existing characters, thus eliminating repeating characters. Once the Key matrix was created the encrypting process may start.

In encrypting the plaintext not much was changed from the original Algorithm. The same main three rules which was explained in the Design section of this paper still applies. The algorithm we used is just a loop that goes through all the character pairs and stops once the last pair is reached. Within the loop we need to get the indexes of the first element in the pair and as well as the second. We did this with a function getIndex() that resides in the matrix class and not in the Playfair class. Basically what the function does is to loop through the whole Matrix until it finds the character we are looking for. After getting the indexes of each element in the current letter pair we are encoding, we then check if the indexes satisfies the three main rules of encoding: (1) Equal rows, (2) Equal columns, and (3) not equal rows and column. For the first two rules(equal rows or equal column) we just check if the matrix should wrap around during the character replacing process in case the index found was at the limit of the Matrix. Then lastly, we transform the encoded text into a matrix for uses of decoding.

For decoding, we need the encoded message to be in the form of a matrix. That was why we transformed the encrypted text back to a matrix. the input was needs to be a matrix for the sake of uniformity as all the other ciphers' decode function has a matrix to decode. The process of decoding is simple. The encoded matrix is put back to string with a simple toString() function and then processed again with the 3 same set of rules. Only this time, the operations are inverse.

Block Cipher

The key used for our block cipher implementation is "bloXLikeMinecraft098(#". The key string is transformed into a 1 x n matrix where n is the length of the key using the same cipher alphabet above. So the matrix formed by our key is:

[28 38 41 24 12 35 37 31 13 35 40 31 29 44 27 32 46 62 61 60 81 75]

Next is to transform the message to be encrypted into a matrix. The matrix will be separated into chunks which are based on the length of our key. Because our key is of length 22, our chunk size which we will use to create the matrix for the message to be encrypted is equal to 22 as well. If there are not enough characters to fill out a chunk, the remaining space in the chunk will be filled out with space characters. For this segment, we will use the message "I went to eat the the new Applebees that opened in Fort Bonifacio" as a basis for showing how our implementation of Block cipher works. The message will be transformed into the matrix:

$$\begin{bmatrix} [9 & 94 & 49 & 31 & 40 & 46 & 94 & 46 & 41 & 94 \\ 31 & 27 & 46 & 94 & 46 & 34 & 31 & 94 & 46 & 34 \\ 31 & 94] & & & & & & & & \\ [40 & 31 & 49 & 94 & 1 & 42 & 42 & 38 & 31 & 28 \\ 31 & 31 & 45 & 94 & 46 & 34 & 27 & 46 & 94 & 41 \\ 42 & 31] & & & & & & & & \\ [40 & 31 & 30 & 94 & 35 & 40 & 94 & 6 & 41 & 44 \\ 46 & 94 & 2 & 41 & 40 & 35 & 32 & 27 & 29 & 35 \\ 31 & 94] & & & & & & & & \end{bmatrix}$$

The result matrix with our encrypted message will have the same dimensions as the message matrix, which in this case is (3, 22). Each row then undergoes a bitwise operation using the key. For our bitwise operation, we use the XOR function to compare one matrix to the other. Java works well with this cipher as it has a built in XOR function. So using the first row of the message matrix and the key matrix, the results from the bitwise operation will be:

Key - First Column - XOR result

28 - 9 - 21

38 - 94 - 120

41 - 49 - 24

24 - 31 - 7

12 - 40 - 36

35 - 46 - 13

37 - 94 - 123

31 - 46 - 49

13 - 41 - 36

35 - 94 - 125

40 - 31 - 55
31 - 27 - 4
29 - 46 - 51
44 - 94 - 114
27 - 46 - 53
32 - 34 - 2
46 - 31 - 49
62 - 94 - 96
61 - 46 - 19
60 - 34 - 30
81 - 31 - 78
75 - 94 - 21

Getting the XOR results and getting their corresponding letters would result in the encrypted message. For the message “I went to eat the the new Applebees that opened in Fort Bonifacio” the encrypted message will be “UPXGjMSwjU3DyJ1Bw委Sd^Uz5X-MIO5R[3”.

Decryting the encrypted message uses the same process and key as the encryption process. Like all ciphers, the decryption process is not 100% accurate. So if you were to decrypt “UPXGjMSwjU3DyJ1Bw委Sd^Uz5X-MIO5R[3” the decrypted message would be “I2went2to2eatlthe the new Applebeeslthat o'ened in2Fort Bonifaci; “

Conclusion

Matrix operations are a good way to implement ciphers for cryptography. It is possible to combine ciphers to improve the security of the encryption. A good example of this would be the Hill Cipher. There are many more possible ciphers out there that can be done via matrices.

Appendix

Contribution of Members

Cardano, Marc Daniel T.	Implementation of Play-Fair Cipher Implementation of User Interface Merged the Interface with Ciphers Documentation for Ciphers
Garcia, Markus Jeremi C.	Implementation of Matrix Classes and Operations Implementation of Cipher Alphabet Implementation of Hill Cipher Implementation of Caesar Cipher Implementation of ROT-13 Cipher
Lucas, John Martin G.	Implementation of Block Cipher Implementation of Play-Fair Cipher Documentation for Ciphers

References

Lyons, J. (2012). Ciphers. Retrieved December 5, 2015, from http://practicalcryptography.com/ciphers/

Lyons, J. (2012). Caesar Cipher. Retrieved December 5, 2015, from http://practicalcryptography.com/ciphers/classical-era/caesar/

Learn Cryptography — Block Cipher. (2014). Retrieved December 5, 2015, from http://learncryptography.com/block-cipher/

Lyons, J. (2012). ROT13 Cipher. Retrieved December 5, 2015, from http://practicalcryptography.com/ciphers/classical-era/rot13/

Lyons, J. (2012). Playfair Cipher. Retrieved December 5, 2015, from http://practicalcryptography.com/ciphers/classical-era/playfair/

Lyons, J. (2012). Hill Cipher. Retrieved December 5, 2015, from http://practicalcryptography.com/ciphers/classical-era/hill/

Dhenakaran, S.S, & Ilayaraja M. Extension of Playfair Cipher using 16X16 Matrix. Retrieved from

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.258.7527&rep=rep1&type=pdf