Project Documentation: Cryptography Program (Seven Level Cipher)

Submitted by:

Wayne Matthew A. Dayata

Submitted to:

Godwin S. Monserate

In partial fulfillment of the requirements of

CS 3106 - Information Assurance and Security

December 4, 2022

Program Overview

Cryptography is the practice and study of techniques for secure communication in the presence of adversarial behavior. More generally, cryptography is about constructing and analyzing protocols that prevent third parties or the public from reading private messages. In the simplest yet effective manner, we can apply various existing cipher algorithms or modify them in a way the messages can no longer be read nor easily guessed when transmitted in between senders and receivers. To be able to encrypt and decrypt them properly, we utilize keys which act as our formulas to translate, encode, or decode a given message to achieve the desired security levels.

Algorithms, Ciphers, and Languages Used

Cryptography ciphers:	1. RSA	2. Atbash Cipher
	3. Ceasar Cipher	4. Vigenere Cipher**
	5. Transpositional Cipher**	(**with modifications)
Programming Language:	**C**	**(C++03 standard)**

Instructions for Program Use

Open and run directly the provided exe file that has already been compiled beforehand.

Steps needed for the cipher machine:

1. Enter the source text file name (example: file1.txt)
2. Enter the mode (1 – encryption, 2 – decryption)
3. Enter an alphabetic string that will be used as the common key for encryption and decryption
4. Enter two numbers (prime numbers) for RSA encryption, or two numbers combining to be the RSA decryption key.
5. Wait for the program to process the text file. Then it will output the resulting file with the file name shown at the end of the program, stored at the same directory where the program and the source text file is.

Tip: The integrity of the decrypted file can be determined by comparing the values of the MD5 or any other hashing algorithm wherein the entire string must exactly match to verify that the program have successfully reverted the encoded message accurately back to its original form.

Cryptography Procedure

Part 1: Algorithm explanation

This program is a combination of implementing five cryptographic algorithms, two of which further modifications and variations provided. Below are the explanations to the processes of each algorithm to help understand the different ways the text has been manipulated to its multi-layer encoded state.

- **RSA Cryptography**

The RSA algorithm is the most widely used Asymmetric Encryption algorithm deployed to date. It takes prime numbers and makes use of various functions such as totient functions to arrive at public and private key sets to encrypt and decrypt the messages respectively.

1. Select two prime numbers P and Q — 7 and 19
2. Get the product -> N — 7*19 = 133
3. Get the totient (p-1)*(q-1) — (7-1)*(19-1) = 108
4. Select a public key E — 29
    - Prime, less than totient, not factor of the totient
5. Select a private key D — 41
    - The Product of the Public Key and the Private Key when divided by the Totient, must result in a remainder of 1.
6. Encryption: Cipher Text = M^E MOD N — 99 -> 99^29 MOD 133 = 92
7. Decryption: Original Message = C^D MOD N — 92 -> 92^41 MOD 133 = 99

- **Atbash cipher**

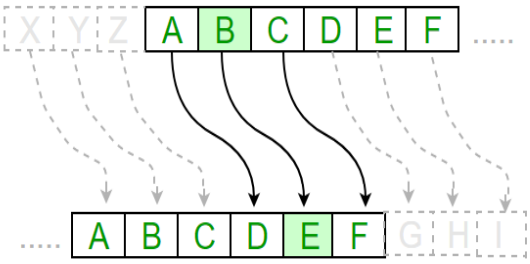
A cryptographic algorithm that maps the distance of each letter from the beginning to that from the end of the alphabet, i.e. a-z, b-y, c-x, etc.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A

The Atbash Cipher table

- **Ceasar/shift cipher**

A type of substitution cipher where each letter in the original message (which in cryptography is called the plaintext) is replaced with a letter corresponding to a certain number of letters shifted up or down in the alphabet. Note that the last few letters will automatically wrap back to the first letters.



Caesar cipher: Shift right by 3 characters

- **Vigenere (polyalphabetic) cipher**

A method of encrypting alphabetic text by using a series of interwoven Caesar ciphers, based on the letters of a keyword. It employs a form of polyalphabetic substitution.

The method still follows Caesar cipher but instead of consistently shifting by n characters throughout the string, it shifts for a varied number of letters dictated by repeated patterns of letters in a string as the key.

- **Transpositional cipher encryption**

A cryptographic algorithm where the order of alphabets in the plaintext is rearranged to form a cipher text. In this process, the actual plain text alphabets are not included. The characters are placed in a matrix with column size equal to the key string's length.

Each column of the matrix is rearranged in a way the characters being represented as the key strings will be sorted. The letters are then rewritten one column at a time, resulting in the cipher text.

```
A U T H O R
1 6 5 2 3 4
-----------
W E A R E D
I S C O V E
R E D S A V
E Y O U R S
E L F A B C
```

yields the cipher

W I R E E R O S U A E V A R B D E V S C A C D O F E S E Y L .

Transposition cipher encryption

Here, AUTHOR is the keyword. The letters in AUTHOR are arranged in alphabetical order, hence the numbers 1 6 5 2 3 4. They represent the order of the columns to be written in as the cipher text.

Part 2: Execution of algorithms in the program

As an additional security layer or feature, the presented cipher algorithms are NOT to be implemented sequentially. But instead, it is to be determined by the key inputted by the user in the program.

To identify the sequence, we first compute and identify the sum of the characters present in the key string according to its ASCII values. This means that the cases of the keys will affect the result (since this can alter the order, even if the letters themselves may not be during the cipher operations). We then take the **sum modulo 7** and adding 1 to the obtained value to get the starting operation. For instance:

E x a m p l e

69 120 97 109 112 108 101

This gives a sum of 716. The remainder when divided by 7 is 2. So this means we begin from algorithm #3.

Next is to determine the interval of the succeeding operations. We take the **sum of the digits of the previous result modulo 6 and adding 1** to the obtained value. Therefore, 7+1+6 = 14 mod 6 = 2. So this mean we have an interval of 2. The resulting sequence becomes

3 -> 5 -> 7 -> 2 -> 4 -> 6 -> 1

And for decryption, we simply apply the reverse which will yield

1 -> 6 -> 4 -> 2 -> 7 -> 5 -> 3

Where the most recent operation in encryption shall be the first decryption operation.

The numbers in the sequence represent the following cipher algorithms:

1. RSA Cryptography	2. Atbash Cipher
3. Caesar (shift) cipher	4. Vigenere cipher – original key
5. Vigenere cipher – reverse key with double value	
6. Transpositional cipher – original key	7. Transpositional cipher – reversed key

With this being implemented, there is a greater increase in the overall complexity of this custom cryptography algorithm and program, making it harder for the message to be decoded.

Example (Encryption)

Message	**Information Assurance and Security Project**		
Keyword	**CIpheRS**	RSA Keys	**2, 13 -> e(5, 26)**

Sum of character ASCII values: 67 + 73 + 112 + 104 + 101 + 82 + 83 = 622

Starting algorithm: 622 mod 7 = 6 (Algorithm 7)

Interval: (6 + 2 + 2) mod 6 = 2

Sequence: 7 -> 2 -> 4 -> 6 -> 1 -> 3 -> 5

7. Transpositional cipher – reversed key (ciphers -> srehpic)

S	R	E	H	P	I	C
I	n	f	o	r	m	a
t	i	o	n		A	s
a	u	r	a	n	c	e
	a	n	d		S	e
c	u	r	i	t	y	
P	r	o	j	e	c	t

Round 1 result: `asee tfornroonadijmAcSycr n teniuaurIta cp`

2. Atbash cipher

Round 2 result: `zhvv gulimillmzwrqnZxHbxi m gvmrfzfiRgz xk`

4. Vigenere cipher – original key (ciphers)

`zhvv gulimillmzwrqnZxHbxi m gvmrfzfiRgz xk`

`ciph erscipherscipherscip h ersciphersc ip`

Round 3 result: `bpkc kldkuxspdryzfuDoZdfx t kmetnommIyb fz`

6. Transpositional cipher – original key (ciphers)

C	I	P	H	E	R	S
b	p	k	c		k	l
d	k	u	x	s	p	d
r	y	z	f	u	D	O
Z	d	f	x		t	
k	m	e	t	n	o	m
m	I	y	b		f	z

Round 4 result: `bdrZkm su n cxfxtbpkydmIkuzfeykpDtofldO mz`

1. RSA Encryption

p=2, q=13, N=26, totient=12

Choose e=5 -> e(5, 26)

$C = M^5 \bmod 26$

```
Original:   2  4 18 26 11 13 19 21 14  3 24  6 24 20  2 16 11 25  4 13  9 11 21 26  6  5 25 11 16  4 20 15  6 12  4 15 13 26
Cipher:     6 10 18  0  7 13 15 21 14  9 20  2 20 24  6 22  7 25 10 13  3  7 21  0  2  5 25  7 22 10 24 19  2 12 10 19 13  0
```

Round 5 result: `fjrZgm ou n itbtxfvgyjmCguzbeygvJxsbljS mz`

3. Caesar Cipher (distance to shift = length(key) = 7)

Round 6 result: `imuCjp rx q lwewaiyjbmpFjxcehbjyMaveomV pc`

5. Vigenere cipher – reverse key with 2x value per character (srehpic -> 38 36 10 16 32 18 6)

L J J P F R F

`imuCjp rx q lwewaiyjbmpFjxcehbjyMaveomV pc`

`ljjpfr fl j jpfrfljjpfrfljjpfrfljjpfrfl jj`

FINAL RESULT (ENCRYPTED): tvdRog wi z uljnfthsqrgKugltmsojVjkjfrG yl

Example (Decryption)

Message	**tvdRog wi z uljnfthsqrgKugltmsojVjkjfrG yl**		
Keyword	**CIpheRS**	RSA Keys	**d(5, 26)**

Sum of character ASCII values: 67 + 73 + 112 + 104 + 101 + 82 + 83 = 622

Ending algorithm: 622 mod 7 = 6 (Algorithm 7)

Interval: (6 + 2 + 2) mod 6 = 2 (-2)

Sequence: 5 -> 3 -> 1 -> 6 -> 4 -> 2 -> 7

5. Vigenere cipher – reverse key with 2x value per character (srehpic -> 38 36 10 16 32 18 6)

L J J P F R F

```
tvdRog wi z uljnfthsqrgKugltmsojVjkjfrG yl
ljjpfr fl j jpfrfljjpfrfljjpfrfljjpfrfl jj    (SUBTRACTION)
```

Round 1 result: `imuCjp rx q lwewaiyjbmpFjxcehbjyMaveomV pc`

3. Caesar Cipher (distance to shift = length(key) = -7)

Round 2 result: `fjrZgm ou n itbtxfvgyjmCguzbeygvJxsbljS mz`

1. RSA Decryption – d(5,26)

Decryption key generated from previous session:

d=5, N=26

$M = C^5 \bmod 26$

```
INPUT KEYS: 5 26
Cipher:     6 10 18 26  7 13 15 21 14  9 20  2 20 24  6 22  7 25 10 13  3  7 21 26  2  5 25  7 22 10 24 19  2 12 10 19 13 26
Original:   2  4 18  0 11 13 19 21 14  3 24  6 24 20  2 16 11 25  4 13  9 11 21  0  6  5 25 11 16  4 20 15  6 12  4 15 13  0
```

Round 3 result: `bdrZkm su n cxfxtbpkydmIkuzfeykpDtofldO mz`

6. Transpositional cipher – original key (ciphers)

C (1)	E (5)	H (4)	I (2)	P (3)	R (6)	S (7)
b		c	p	k	k	l
d	s	x	k	u	p	d
r	u	f	y	z	D	O
Z		x	d	f	t	
k	n	t	m	e	o	m
m		b	I	y	f	z

Round 4 result: `bpkc kldkuxspdryzfuDoZdfx t kmetnommIyb fz`

4. Vigenere cipher – original key (ciphers)

`bpkc kldkuxspdryzfuDoZdfx t kmetnommIyb fz`

`ciph erscipherscipherscip h ersciphersc ip` **(SUBTRACTION)**

Round 5 result: `zhvv gulimillmzwrqnZxHbxi m gvmrfzfiRgz xk`

2. Atbash cipher

Round 6 result: `asee tfornroonadijmAcSycr n teniuaurIta cp`

7. Transpositional cipher – reverse key (srehpic)

C (7)	E (3)	H (4)	I (6)	P (5)	R (2)	S (1)
a	f	c	m	r	n	I
s	o	x	A		i	t
e	r	f	c	n	u	a
e	n	x	S		a	
	r	t	y	t	u	c
t	o	b	c	e	r	P

FINAL RESULT (DECRYPTED): Information Assurance and Security Project