Ceylon Journal of Science 48(4) 2019: 327-330
DOI: http://doi.org/10.4038/cjs.v48i4.7672

RESEARCH ARTICLE

A symmetric and a transposition cipher using the Euler's totient function

A.P. Madushani and P.G.R.S. Ranasinghe*

Department of Mathematics, Faculty of Science, University of Peradeniya, Peradeniya, Sri Lanka.

Received: 30/04/2019 ; Accepted: 25/10/2019

Abstract: Cryptography provides a method of exchanging sensitive information in a secured form while assuring its confidentiality. Encryption and decryption are the two steps in which the process gets completed. In the present work, we introduce two algorithms using modular arithmetic and ASCII symbols to encrypt and decrypt messages. Our schemes are based on symmetric ciphers, in which a secret key is exchanged between the sender and the receiver for the encoding and decoding process to take place.

Keywords: Modular arithmetic, Euler's totient function, ASCII, Symmetric cipher.

INTRODUCTION

Throughout history, mankind has always had the need to communicate in secrecy for numerous reasons. With the advent of technology, the information security is becoming a decisive factor in the modern world. Hence, the importance of cryptography has grown leaps and bounds which has created an imperative need for novel encrypting and decrypting algorithms which are more secure as well as user friendly.

The base for our two algorithms is modular arithmetic which plays a vital role in cryptography from classical crypto systems to modern asymmetric systems. The earliest records of the use of modular arithmetic are found in the Caesar cipher which dated back to the era of the Roman emperor Julius Caesar, in which each letter is shifted by a fixed amount under some modulo value (interested readers may refer, Holden, 2018). It is also used in the historically significant, well-known cryptographic protocol in practice today, the RSA algorithm, which is based on modular exponentiation. In our algorithms, we use the *Euler's totient function* of a positive integer as the modulo value in the encryption and decryption processes, which is defined to be the number of integers j with $1 \leq j \leq n$ such that $\gcd(j,n)=1$ (Kraft and Washington, 2018). Throughout this paper, $\gcd(\alpha,\beta)$ stands for the greatest common divisor of the two numbers α and β.

In 2013, A. Mishra has introduced some techniques to provide more security to classical cryptosystems using the Caesar cipher as the representative and in the same year, Q-A. Kester has implemented a hybrid cryptosystem based on Vigènere cipher and columnar transposition cipher. Incorporating those ideas we have developed two cryptographic schemes that use the Euler's totient function which is a novel approach for symmetric cryptography. Our first algorithm is a transposition cipher which shifts the position of each letter in the plaintext to obtain the ciphertext under some modulo value. We also introduce a symmetric cipher using the ASCII characters, which are abbreviated from the "American Standard Code for Information Interchange" is a character encoding standard for electronic communication. It allows the user to encrypt all the characters in a plaintext including the spaces, which will increase the security of the system.

METHODOLOGY

Here we introduced the general procedure for the two proposed algorithms.

A transposition cipher using the Euler's totient function:

In this section we introduced a transposition cipher based on modular arithmetic which used the Euler's totient function of a positive integer to shift the positions of each letter in a plaintext under some modulo value.

We first stated a couple of standard results from literature.

Theorem (Rosen, 2005):

Let a,b and m be integers such that $m>0$ and $\gcd(a,m)=d$. If d does not divide b, then $ax\equiv b(\text{mod } m)$ has no solution. If d divides b, then $ax\equiv b(\text{mod } m)$ has exactly d incongruent solutions.

Corollary (Kraft and Washington, 2018):

The integer a has an inverse modulo m if and only if $\gcd(a,m)=1$.

Encrypting Algorithm:

Step 1*:* Ignore the spaces in the plaintext.

Step 2*:* Assign a number to position each letter as they appear in the plaintext.

Step 3*:* Determine the number of distinct letters, n and find the Euler's totient function of n, $\varphi(n)$.

Step 4*:* Choose m to be the total number of letters in the

**Corresponding Author's Email:* rajitharana@pdn.ac.lk
https://orcid.org/0000-0002-7465-4328

plaintext or the smallest odd number greater than that to satisfy the condition gcd($\varphi(n)$,m)=1.

Remark: $\varphi(n)$ is even for n>2 (Rosen, 2005).

Step 5*:* Obtain the new positions q under modulo m such that,

$$q \equiv p \cdot \varphi(n) (\text{mod } m).$$

Decrypting Algorithm:

Reverse the encryption process with an inverse modulo m. Since we obtain gcd(m,$\varphi(n)$)=1, the existence of an inverse modulo m is guaranteed by the corollary.

If x is the plaintext position that we need to determine it should satisfy,

$$\varphi(n) \cdot x \equiv q (\text{mod } m).$$

A Symmetric cipher using ASCII codes:

In this section, we introduce a symmetric cryptosystem based on ASCII codes using the Euler's totient function. The salient feature is the use of two random keys to encrypt all the characters in the plaintext which increases the security of the scheme.

Encrypting Algorithm:

Step 1: Assign a positional value to each character in the plaintext and group them as odd and even according to those positions.

Step 2: Considering each character as an ASCII symbol get the corresponding decimal value, p.

Step 3: Determine the number of characters n, in each group and find the Euler's totient function of n, φ (n).

Step 4: Identify the maximum decimal value for both odd and even cases. Let m be the maximum value, if gcd($\varphi(n)$,m)=1;; otherwise choose the nearest prime number greater than the maximum value to satisfy that condition.

Step 5: Obtain a new value q under modulo m for each p such that,

$$q \equiv p \cdot \varphi(n) (\text{mod } m).$$

Step 6: Select two random key streams for odd and even cases using ASCII symbols and let x be the decimal value corresponding to each of these characters.

Step 7: Determine the decimal value of each ciphertext character by adding q and x.

Step 8: Convert each character into ASCII symbols and combine the odd and even groups to obtain the ciphertext.

Decrypting Algorithm:

We shall reverse the encryption process with an inverse modulo m. Since we choose m to satisfy the condition gcd(m,$\varphi(n)$)=1 the existence of an inverse modulo m is guaranteed by the corollary.

If the decimal values corresponding to plaintext characters that we need to determine are x, we shall solve,

$$\varphi(n) \cdot x \equiv q (\text{mod } m).$$

RESULTS AND DISCUSSION

First, we will illustrate the encryption process for the transposition cipher using the plaintext message "HELLO EVERYONE".

Example 01:

Step 1: Ignore the space and note down the position of each distinct letter. (Refer Table 1)

Step 2*:*Number of distinct letters, n=8 and $\varphi(8)$=4.

Step 3*:* Total number of letters, m=13. Note that gcd($\varphi(8)$,13)=1.

Step 4*:* Compute $q_{*\#} \equiv p \cdot \varphi(n)$ (mod m) where * and # stands for each distinct letter and its positional value respectively, to find the ciphertext positions $q_{*\#}$ of each letter.

H: $q_{H1} \equiv 1 \cdot 4(\text{mod}13) \equiv 4(\text{mod } 13)$

E: $q_{E2} \equiv 2 \cdot 4(\text{mod}13) \equiv 8(\text{mod } 13)$,

$q_{E6} \equiv 6 \cdot 4(\text{mod}13) \equiv 11(\text{mod } 13)$,

$q_{E8} \equiv 8 \cdot 4(\text{mod}13) \equiv 6(\text{mod } 13)$,

$q_{E13} \equiv 13 \cdot 4(\text{mod}13) \equiv 0(\text{mod } 13)$

L: $q_{L3} \equiv 3 \cdot 4(\text{mod}13) \equiv 12(\text{mod } 13)$,

$q_{L4} \equiv 4 \cdot 4(\text{mod}13) \equiv 3(\text{mod } 13)$

O: $q_{O5} \equiv 5 \cdot 4(\text{mod}13) \equiv 7(\text{mod } 13)$,

$q_{O11} \equiv 11 \cdot 4(\text{mod}13) \equiv 5(\text{mod } 13)$

V: $q_{V7} \equiv 7 \cdot 4(\text{mod}13) \equiv 2(\text{mod } 13)$

R: $q_{R9} \equiv 9 \cdot 4(\text{mod}13) \equiv 10(\text{mod } 13)$

Y: $q_{Y10} \equiv 10 \cdot 4(\text{mod}13) \equiv 1(\text{mod } 13)$

N: $q_{N12} \equiv 12 \cdot 4(\text{mod}13) \equiv 9(\text{mod } 13)$

Table 1: Each distinct letter and the corresponding position(s) in the plaintext.

Distinct letter	H	E	L	O	V	R	Y	N
Position(s) in the plaintext (p)	1	2, 6, 8, 13	3, 4	5, 11	7	9	10	12

Table 2: Each distinct letter and the corresponding position(s) in the ciphertext.

Position in the plaintext.	1	2	3	4	5	6	7	8	9	10	11	12	13
Assigned letter	Y	V	L	H	O	E	O	E	N	R	E	L	E

So the ciphertext is YVLHOEOENRELE.

Remark: The ciphertext is an anagram of the plaintext.

We will illustrate the decryption process using the same ciphertext. To get the positions in the plaintext, we solve the congruence

$$4x \equiv q \text{ (mod 13)},$$

where q is obtained from Table 2.

For example, to find the new position(s) of the letter 'O', we solve the congruences,

$4x \equiv 5$ (mod 13) and $4x \equiv 7$ (mod 13)

to get, $x \equiv 11$ (mod 13) and $x \equiv 5$ (mod 13) respectively, which are exactly the two positions of the letter 'O' in Table 1 as expected. Continuing this process the plaintext can be deciphered.

Now, we will illustrate the encryption process for the symmetric cipher with the plaintext,

"HELLO EVERYONE!".

Example 02:

Step 1: Group the characters as odd and even according to their positional value.

Step 2: Considering each character as an ASCII symbol obtain their relevant decimal value. (Refer Tables 3 and 4 below)

Step 3: Find m,n,and φ (n) for the two cases.

For "odd" characters: max = 89, $n = 8$, and φ (8) = 4; note that gcd(max,φ (8)) = (89, 4) = 1, thus let $m = 89$.

For "even" characters: max = 86, $n = 7$, and φ (7) = 6; note that gcd(max,φ (7)) = (86, 6) = 2≠1, thus let $m = 89$.

Step 4: Compute $q_* \equiv p \cdot \varphi(n)$(mod m) for the corresponding positional value of each character * in the plaintext and find the value q.

Odd positions: H:$q_H \equiv 4 \cdot 72$(mod 89)≡21(mod 89)
Even positions: E:$q_E \equiv 6 \cdot 69$(mod 89)≡58(mod 89)

Odd positions: L:$q_L \equiv 4 \cdot 76$(mod 89)≡37(mod 89)
Even positions: L:$q_L \equiv 6 \cdot 76$(mod 89)≡11(mod 89)

Odd positions: O: $q_O \equiv 4 \cdot 79$(mod 89)≡49(mod 89)
Even positions: (sp):$q \equiv 6 \cdot 32$(mod 89)≡14(mod 89)

Odd positions: E:$q_E \equiv 4 \cdot 69$(mod 89)≡9(mod 89)
Even positions: V:$q_V \equiv 6 \cdot 86$(mod 89)≡71(mod 89)

Odd positions: Y: $q_Y \equiv 4 \cdot 89$(mod 89)≡0(mod 89)
Even positions: R: $q_R \equiv 6 \cdot 82$(mod 89)≡47(mod 89

Odd positions: N:$q_N \equiv 4 \cdot 78$(mod 89)≡45(mod 89)
Even positions: O: $q_O \equiv 6 \cdot 79$(mod 89)≡29(mod 89)

Odd positions: !:$q_! \equiv 4 \cdot 33$(mod 89)≡43(mod 89)

Step 5: Obtain the decimal equivalents (x) of each character of the two key streams and obtain the ciphertext as below (Table 5,6).

Step 6: Combine the odd and even cases to obtain the complete ciphertext,

"Acknowledgement"

Table 3: Odd characters and the corresponding decimal values.

Character(ASCII)	H	L	O	E	E	Y	N	!
Decimal value (p)	72	76	79	69	69	89	78	33

Table 4: Even characters and the corresponding decimal values.

Character(ASCII)	E	L	(SP)*	V	R	O	E
Decimal value (p)	69	76	32	86	82	79	69

*Here 'SP' stands for 'space'.

Table 5: Obtaining the ciphertext for "odd" characters.

Plaintext Character(ASCII)	H	L	O	E	E	Y	N	!
New value (q)	21	37	49	9	9	0	45	43
Key stream	,	F	>	c	[	e	8	I
Decimal equivalents (x)	44	70	62	99	91	101	56	73
$x + q$	65	107	111	108	100	101	101	116
Ciphertext	A	k	o	l	d	e	e	t

Table 6: Obtaining the ciphertext for "even" characters.

Plaintext Character(ASCII)	E	L	(sp)	V	R	O	E
New value (q)	58	11	14	71	47	29	58
Keystream	)	c	i	30	8	P	4
Decimal equivalents (x)	41	99	105	30	56	80	52
$x + q$	99	110	119	101	103	109	110
Ciphertext	c	n	w	e	g	m	n

We will illustrate the decryption process using the same ciphertext.

To get the decimal equivalents of the plaintext characters, reverse the encryption process and solve the congruences,

$4x \equiv q \pmod{89}$ and $6x \equiv q \pmod{89}$,

where q is obtained from the Table 3 and Table 4 respectively.

For example, to find the plaintext character of the letter '*m*' in the ciphertext, first note that '*m*' belongs to the "even" case and $q = 29$. By solving the congruence, $6x \equiv 29 \pmod{89}$, we obtain $x \equiv 79 \pmod{89}$. So 79 is the decimal equivalent of the ASCII character 'O' in the plaintext. Continuing this process for both "odd" and "even" cases, the plaintext can be deciphered.

In the transposition cipher that we have introduced, no key is shared between the authorized parties. Thus, if the method of encryption is unknown, secrecy of the plaintext can be assured. To decrypt a message that has been encrypted using the proposed symmetric cipher one must possesses two secret keys. This will increase the confidentiality in communication across unsecure channels in contrast to that of the single key classical cryptosystems.

CONCLUSION

In this paper, we have introduced a couple of cryptographic schemes; a transposition cipher and a symmetric cipher (using the ASCII characters). The key ingredient to our method is the use of the Euler's totient function with its intrinsic properties. Our work is just the beginning of many avenues to be explored in future such as comparing the optimality with respect to the existing algorithms and we believe that this will inspire researchers with the same interest.

REFERENCES

Holden, J. (2018). The Mathematics of Secrets: Cryptography from Caesar ciphers to Digital encryption. *Princeton University Press*, pp. 1-28, 207.

Kester, Q.A., (2013). A Hybrid Cryptosystem based on Vigenere cipher and Columnar Transposition cipher. *arXiv preprint arXiv*:1307.7786.

Kraft, J. and Washington, L. (2018). An Introduction to Number Theory with Cryptography. *Chapman and Hall/CRC*, pp. 127-168.

Mishra, A., (2013). Enhancing security of caesar cipher using different methods. *International Journal of Research in Engineering and Technology*, **2**(09), pp.327-332.

Rosen, K. H. (2005). Elementary Number Theory and It's Applications. *Addison-Wesley*, pp. 102, 161-165.