http://www.newtheory.org **ISSN: 2149-1402**

Journal of New Theory

Received: 20.04.2015 *Year*: **2016**, *Number*: **14**, *Pages*: **46-57**
Published: 22.07.2016 *Original Article*[**]

FRAGMENTED CAESAR CIPHER

Yunus Aydoğan <yunus.programmer@gmail.com>
Naim Çağman <naim.cagman@gop.edu.tr>
Irfan Şimşek[*] <irfan.simsek@gop.edu.tr>

Department of Mathematics, Faculty of Arts and Sciences, Gaziosmanpaşa University
60240 Tokat, Turkey

Abstract — In this study, we define a cipher method that is called fragmented Caesar cipher method that based on the basic logic of Caesar cipher. This new method has more possibility then the classical Caesar cipher because of the fragmented alphabet is used to cipher. We then construct a mathematical modeling and make a computer programs of the method.

Keywords — *Caesar cipher, encryption, decryption, plaintext, ciphertext, fragmented Caesar cipher.*

1 Introduction

One of the earliest known cryptographic systems was used by Julius Caesar. In the Caesar cipher, each letter in a plaintext is shifted by a letter a certain number of positions down the alphabet. The Caesar cipher can be decrypted an easy way with the brute-force attack. Since then a lots of technics of cipher have been developed to obtain an unbroken cipher technic. For examples, Omolara *et al.* [5], Patni [7, 8], Dey *et al.* [2] and Mishra [4]. More detailed explanations related to the Caesar cipher can be found in [9] and [6].

In this study, we define a cipher method that is called fragmented Caesar cipher method that is based on the Caesar cipher. In the fragmented Caesar cipher, the alphabet broken into small fragments and each letter in each fragment is replaced by a letter some arbitrary number of positions down. We then construct a mathematical

[**] *Edited by* Oktay Muhtaroğlu *(Area Editor)*
[*] *Corresponding Author.*

modeling and make a computer programs of the method. We finally give an example to show the cipher method is working successfully.

The present paper is a condensation of part of the dissertation [1].

2 Fragmented Caesar Cipher Method

In this section, we define a new cipher method which depends on the Caesar cipher method. In this method, we firstly divide the alphabet arbitrary fragments. And then each letter in each fragment is replaced by a letter some fixed number of positions down. Therefor, we call this method as fragmented Caesar cipher method or in sort FCC-method.

2.1 Mathematical Model of FCC-method

In this subsection, we first give a mathematical model of the FCC-method. We then write an algorithm of the FCC-method to make a computer program.

Throughout this paper, ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange) is used, $I_n = \{1, 2, ..., n\}$ for all $n \in \mathbb{N}$ is an index set and U is a set of using characters which is ordered according to the ASCII.

Definition 1. Let $|U| = n$ and $X = \{x_i : i \in I_n\}$ be an ordered set according to the index set I_n. Then,

$$\alpha : U \to X, \quad \alpha(i\text{-th element}) = x_i, i \in I_n,$$

is called **indexing function** of U. Here, x_i is called **indexed element** of ith element of U and the set X is called **indexed character set** of U.

Example 2. Let 1, 9, b, M, $<$ and $+$ be using characters. Then, the character set U is written as $U = \{+, 1, 9, <, M, b\}$ since

x	$+$	1	9	$<$	M	b
$ASCII$	43	49	57	60	77	98

Therefore the indexed character set of U is obtained as $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ since

x	$+$	1	9	$<$	M	b
$\alpha(x)$	x_1	x_2	x_3	x_4	x_5	x_6

Definition 3. Let X be an indexed character set and $|X| = n$. Then, for $a_p \in \mathbb{N}$, $p \in I_n$, a fragmentation algorithm is set up as follows:

Algorithm of Fragmentation:

Step 1: Choose a_1 such that $2 \leq a_1 \leq n_1 = n - 2$

Step 2: Let $n_2 = n_1 - a_1$. If $n_2 > 4$, choose a_2 such that $2 \leq a_2 \leq n_2 - 2$, if not $a_2 = n_2$ which means the process is terminated.

$\vdots$

Step p: Let $n_p = n_{p-1} - a_{p-1}$. If $n_p > 4$, choose a_p such that $2 \leq a_p \leq n_p - 2$, if not $a_p = n_p$ which means the process is terminated.

Here, p is called a **fragment number**, a_p is number of characters in a fragment and $P = (a_1, a_2, ..., a_p)$ is called **fragment key** of X.

We can briefly choose values a_p as follow, for $p \in I_n$ and $i \in I_p$,

$$\begin{cases} 2 \leq a_1 \leq n_1, & \text{if } p = 1,\ n_1 = n - 2 \\ 2 \leq a_p \leq n_i - 2, & \text{if } p > 1,\ 4 < n_p,\ n_p = n_{p-1} - a_{p-1} \\ n_p = a_p, & \text{if } p > 1,\ n_p < 4,\ n_p = n_{p-1} - a_{p-1} \end{cases}$$

Example 4. Let X be an indexed character set and $|X| = 13$. If the fragmentation algorithm is working as follows,

Step 1: Choose $a_1 = 5$ such that $2 \leq a_1 \leq n_1 = 13 - 2 = 11$,

Step 2: Choose $a_2 = 6$ such that $2 \leq a_2 \leq 8 - 2$, because of $n_2 = 13 - 5 = 8$ and $8 > 4$,

Step 3: Choose $a_3 = 2$ because of $n_3 = 8 - 6 = 2$ and $2 < 4$.

Then, we obtain that $p = 3$ and $P = (5, 6, 2)$.

Definition 5. Let $X = \{x_1, x_2, ..., x_n\}$ be an indexed character set. For all $i \in I_n$ and $k \in I_{n-i}$, the set $W = \{x_i, x_{i+1}, ..., x_{i+k}\}$ is called as a **block subset** of X and denoted by $W \sqsubseteq X$.

Definition 6. Let X be an indexed character set, p be a number of fragment of X. If $X_i \sqsubseteq X$ has the following conditions, then family of set $\{X_i : i \in I_p\}$ is called an **ordered fragmentation** of X.

1. $|X_i| = a_i$,
2. $X_i \cap X_j = \emptyset$ for $i, j \in I_p$, $i \neq j$,
3. $X = \bigcup_{i \in I_p} X_i$,
4. $x_{max(X_i)+1} = x_{min(X_{i+1})}$ for $i \in I_p$, where $x_{min(X_i)}$ and $x_{max(X_i)}$ be the first and the last element of X_i, respectively.

Here, the X_i is called a **fragment** of X for $i \in I_p$.

Example 7. Let us consider Example 4 where $X = \{x_1, x_2, ..., x_{13}\}$ and $P = (5, 6, 2)$. Then, for $a_1 = 5$, $a_2 = 6$ and $a_3 = 2$ the ordered fragmentation of X are respectively as follow,

$$\begin{aligned} X_1 &= \{x_1, x_2, x_3, x_4, x_5\} \\ X_2 &= \{x_6, x_7, x_8, x_9, x_{10}, x_{11}\} \\ X_3 &= \{x_{12}, x_{13}\} \end{aligned}$$

Therefor, the ordered fragmentation of X is obtained as $\{X_1, X_2, X_3\}$.

Definition 8. Let X_i be a fragment of X and a_i be the number of X_i for all $i \in I_p$. If $0 < r_i < a_i$, then $R = (r_1, r_2, ..., r_p)$ is called **rotation key** of X.

Here, the r_i is called a **number of rotation** of X_i for all $i \in I_p$.

Note that the key of this method has two part, one of them is a fragment key P and the other is a rotation key R.

Example 9. Let us consider Example 4, if we choose number of rotations $r_1 = 3$, $r_2 = 4$ and $r_3 = 1$ for X_1, X_2, X_3, respectively. Then, the rotation key of X would be $R = (3, 4, 1)$.

Definition 10. Let X be an indexed character set, $\{X_1, X_2, ..., X_p\}$ be an ordered fragmentation of X and $P = (a_1, a_2, ..., a_p)$ be a fragment key of X. Then, m_i is defined by

$$m_i = \begin{cases} 0, & i = 0 \\ m_{i-1} + a_i, & i \in I_p \end{cases}$$

and called a **module** of X_i for all $i \in I_p$.

It is clear to see that $x_{m_i} = x_{max(X_i)}$ for $i \in I_p$.

Definition 11. Let X be an indexed character set and $\{X_1, X_2, ..., X_p\}$ be an ordered fragmentation of X. If m_i is a module of X_i for all $i \in I_p$ and $R = (r_1, r_2, ..., r_p)$ is a rotation key of X, then X_i**-rotation function**, denoted by β_i, is defined by

$$\beta_i : X_i \to X_i,\ \beta_i(x_t) = \begin{cases} x_{t+r_i}, & t + r_i \leq m_i \\ x_{(t+r_i)(mod\, m_i)+m_{i-1}}, & t + r_i > m_i \end{cases}$$

where $t \in I_{a_i}$.

Definition 12. Let X be an indexed character set and $\{X_1, X_2, ..., X_p\}$ be an ordered fragmentation of X. If β_i is an X_i-rotation function for all $i \in I_p$, then the following function

$$\beta : X \to X,\ \beta(x) = \begin{cases} \beta_1(x), & x \in X_1 \\ \beta_2(x), & x \in X_2 \\ \quad \vdots & \\ \beta_p(x), & x \in X_p \end{cases}$$

is called a **rotation function** of X.

Definition 13. Let $\alpha : U \to X$ be an indexing function. Then for all $t \in I_n$, inverse of α is called a **characterization function** and defined by

$$\alpha^{-1} : X \to U, \quad \alpha^{-1}(x_t) = \text{"}t\text{-th element of U "}$$

Definition 14. If $\alpha : U \to X$, $\alpha^{-1} : X \to U$ and $\beta : X \to X$ be indexing, characterization and rotation functions, respectively, then an **encryption function** on U is defined by

$$\gamma : U \to U, \quad \gamma(x) = \alpha^{-1}(\beta(\alpha(x)))$$

Definition 15. Let X be an indexed character set and $\{X_1, X_2, ..., X_p\}$ be an ordered fragmentation of X. If $R = (r_1, r_2, ..., r_p)$ is a rotation key of X, $\beta_i : X_i \to X_i$ is an X_i-rotation function and m_i is a module of X_i for all $i \in I_p$, then **inverse of rotation function** of X_i, denoted by β_i^{-1}, is defined by

$$\beta_i^{-1} : X_i \to X_i,$$
$$\beta_i^{-1}(x_t) = \begin{cases} x_{t+m_i-(r_i+m_{i-1})}, & t+m_i-(r_i+m_{i-1}) \leq m_i \\ x_{(t+m_i-(r_i+m_{i-1}))(mod\, m_i)+m_{i-1}}, & t+m_i-(r_i+m_{i-1}) > m_i \end{cases} \text{ where } t \in I_{a_i}.$$

Definition 16. Let X be an indexed character set and $\{X_1, X_2, ..., X_p\}$ be an ordered fragmentation of X. If β_i^{-1} is an inverse of rotation function of X_i for all $i \in I_p$, then the following function

$$\beta^{-1} : X \to X,\ \beta^{-1}(x) = \begin{cases} \beta_1^{-1}(x), & x \in X_1 \\ \beta_2^{-1}(x), & x \in X_2 \\ \quad \vdots & \\ \beta_p^{-1}(x), & x \in X_p \end{cases}$$

is called a **inverse of rotation function** of X.

Definition 17. If $\alpha : U \to X$, $\alpha^{-1} : X \to U$, $\beta^{-1} : X \to X$ and $\gamma : U \to U$ be indexing, characterization, inverse of rotation and encryption functions, respectively, then a **decryption function** on U is defined by

$$\gamma^{-1} : U \to U, \quad \gamma^{-1}(x) = \alpha^{-1}(\beta^{-1}(\alpha(x)))$$

It is clear to see that $\gamma^{-1}(x) = \alpha^{-1}(\beta^{-1}((\alpha^{-1})^{-1}(x))) = \alpha^{-1}(\beta^{-1}(\alpha(x)))$.

Definition 18. Let U be a character set, P be a fragment key, R be a rotation key and γ be an encryption function. The four tuple (U, P, R, γ) is called an **FCC-encryption** on U. The four tuple (U, P, R, γ^{-1}) is called an **FCC- decryption** on U.

2.2 FCC-Encryption Algorithm

Assume that U is a character set and X is an indexed character set. Then, an algorithm of the FCC-encryption is set up as follows:

Algorithm of FCC-Encryption:

Step 1: Find the fragment number p and the $P = (a_1, a_2, ..., a_p)$,
Step 2: Choose the $R = (r_1, r_2, ..., r_p)$ according to the P,
Step 3: Find the $\{X_i : i \in I_p\}$ and the module m_i for each X_i,
Step 4: Find the $\beta_i(x_t)$ for $x_t \in X_i$ $t \in I_{a_i}$ and $i \in I_p$,
Step 5: Find the $\alpha^{-1}(x_t)$ for $x_t \in X_i$, $t \in I_{a_i}$ and $i \in I_p$,
Step 6: Find the $\gamma(x)$ for $x \in U$.

Example 19. Let

$$U = \{\text{ç, d, e, f, g, ğ, h, ı, a, b, c, m, n, i, j, k, l, u, ü, o, ö, p,r, s, ş, t, z, v, y}\}$$

and

$$X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, ..., x_{29}\}$$

Then,

Step 1: By using the algorithm of fragmentation, we can obtain the fragment number $p = 4$ and the $P = (11, 6, 9, 3)$ where $a_1 = 11$, $a_2 = 6$, $a_3 = 9$ and $a_4 = 3$.

Step 2: For $a_1 = 11$, $a_2 = 6$, $a_3 = 9$ and $a_4 = 3$ the rotation key is obtained as $R = (3, 4, 7, 2)$ since $0 < r_1 = 3 < a_1 = 11, \quad 0 < r_2 = 4 < a_2 = 6, \quad 0 < r_3 = 7 < a_3 = 9, \quad 0 < r_1 = 2 < a_4 = 3$.

Step 3: For a_i $(i = 1, 2, 3, 4)$ the fragments of X, X_i, are obtained as,

$$\begin{array}{ll} \text{for } a_1 = 11, & X_1 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}\} \\ \text{for } a_2 = 6, & X_2 = \{x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}\} \\ \text{for } a_3 = 9, & X_3 = \{x_{18}, x_{19}, x_{20}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}\} \\ \text{for } a_4 = 3, & X_4 = \{x_{27}, x_{28}, x_{29}\} \end{array}$$

and value of m_i $(i = 0, 1, 2, 3, 4)$ can also choose as,

$$\begin{array}{ll} \text{for } i = 0, & m_0 = 0 \\ \text{for } i = 1, & m_1 = (m_0 = 0) + (a_1 = 11) = 11 \\ \text{for } i = 2, & m_2 = (m_1 = 11) + (a_2 = 6) = 17 \\ \text{for } i = 3, & m_3 = (m_2 = 17) + (a_3 = 9) = 26 \\ \text{for } i = 4, & m_4 = (m_3 = 26) + (a_4 = 3) = 29. \end{array}$$

Step 4: For $i = 1, 2, 3, 4$ values of the X_i-rotation function β_i are obtained as follows. Here, we first obtain the values of β_1 as,

$$\begin{array}{ll} \beta_1(x_1) = x_4, & \text{since } 1 + r_1 = 1 + 3 = 4 \text{ because of } 1 + 3 < 11 \\ \beta_1(x_2) = x_5, & \text{since } 2 + r_1 = 1 + 3 = 5 \text{ because of } 2 + 3 < 11 \\ \beta_1(x_3) = x_6, & \text{since } 3 + r_1 = 1 + 3 = 6 \text{ because of } 3 + 3 < 11 \\ \beta_1(x_4) = x_7, & \text{since } 4 + r_1 = 1 + 3 = 7 \text{ because of } 4 + 3 < 11 \\ \beta_1(x_5) = x_8, & \text{since } 5 + r_1 = 1 + 3 = 8 \text{ because of } 5 + 3 < 11 \\ \beta_1(x_6) = x_9, & \text{since } 6 + r_1 = 1 + 3 = 9 \text{ because of } 6 + 3 < 11 \\ \beta_1(x_7) = x_{10}, & \text{since } 7 + r_1 = 1 + 3 = 10 \text{ because of } 7 + 3 < 11 \\ \beta_1(x_8) = x_{11}, & \text{since } 8 + r_1 = 1 + 3 = 11 \text{ because of } 8 + 3 = 11 \\ \beta_1(x_9) = x_1, & \text{since } (9 + 3)(mod\, 11) + 0 = 1 \text{ because of } 9 + 3 > 11 \\ \beta_1(x_{10}) = x_2, & \text{since } (10 + 3)(mod\, 11) + 0 = 2 \text{ because of } 10 + 3 > 11 \\ \beta_1(x_{11}) = x_3, & \text{since } (11 + 3)(mod\, 11) + 0 = 3 \text{ because of } 11 + 3 > 11 \end{array}$$

and for $i = 2, 3, 4$ the values of β_i are obtained similarly. Hence

X_1	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
β_1	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
X_1	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_1	x_2	x_3

X_2	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}
β_2	↓	↓	↓	↓	↓	↓
X_2	x_{16}	x_{17}	x_{12}	x_{13}	x_{14}	x_{15}

X_3	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}
β_3	↓	↓	↓	↓	↓	↓	↓	↓	↓
X_3	x_{25}	x_{26}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}

X_4	x_{27}	x_{28}	x_{29}
β_4	↓	↓	↓
X_4	x_{29}	x_{27}	x_{28}

and therefore,

X	x_1	x_2	...	x_{11}	x_{12}	x_{13}	...	x_{17}	x_{18}	x_{19}	...	x_{26}	x_{27}	x_{28}	x_{29}
β	↓	↓		↓	↓	↓		↓	↓	↓		↓	↓	↓	↓
X	x_4	x_5	...	x_3	x_{16}	x_{17}	...	x_{15}	x_{25}	x_{26}	...	x_{24}	x_{29}	x_{27}	x_{28}

Step 5: Values of the characterization function α^{-1} are obtained as following list:

X	x_4	x_5	...	x_3	x_{16}	x_{17}	...	x_{15}	x_{25}	x_{26}	...	x_{24}	x_{29}	x_{27}	x_{28}
α^{-1}	↓	↓		↓	↓	↓		↓	↓	↓		↓	↓	↓	↓
U	ç	d	...	c	m	n	...	l	u	ü	...	t	z	v	y

Step 6: Values of the encryption function γ are obtained as following list:

U	a	b	c	ç	d	e	f	g	ğ	h	ı	i	j	k	l	m	n	...	v	y	z
γ	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		↓	↓	↓
U	ç	d	e	f	g	ğ	h	ı	a	b	c	m	n	o	i	j	k	...	z	v	y

In this example we showed that the plaintext "ankara" is encrypted as "çliçöç" according to the method which can be seen in Figure 1.

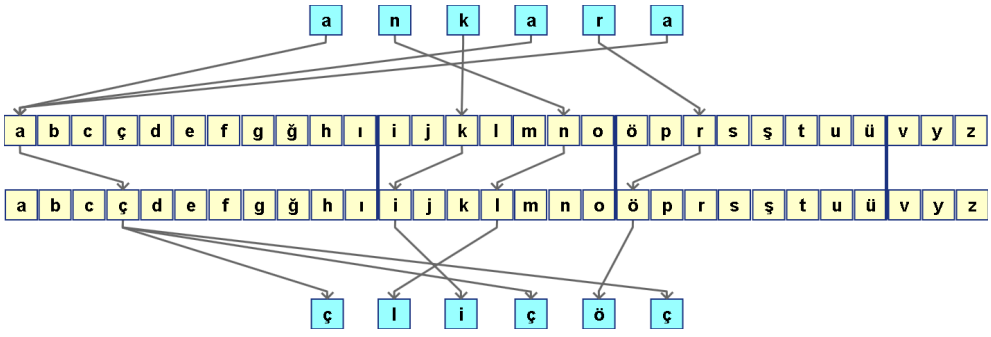

Figure 1: Encryption of "ankara" by FCEA

2.3 FCC-Decryption Algorithm

Assume that U is a character set and X is an indexed character set. Then, an algorithm of the FCC-Decryption is set up as follows:

Algorithm of FCC-Decryption:

Step 1: Use the $\{X_i : i \in I_p\}$ and the module m_i for each X_i,
Step 2: Find the $\beta_i^{-1}(x_t)$ for $x_t \in X_i$, $t \in I_{a_i}$ and $i \in I_p$,
Step 3: Find the $\alpha^{-1}(x)$ for $x \in U$,
Step 4: Find the $\gamma^{-1}(x)$ for $x \in U$.

Example 20. Let us consider the result of Example 19 where

$$U = \{\text{ç, d, e, f, g, ğ,h, ı, a, b, c, m, n, i, j, k, l, u, ü, o, ö, p, r, s, ş, t, z, v, y}\}$$

and

$$X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, ..., x_{29}\}$$

Then,

Step 1 : In Example 19, for a_i $(i = 1, 2, 3, 4)$ the fragments of X, X_i, and was obtained as

$$\begin{array}{ll} \text{for } a_1 = 11, & X_1 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}\} \\ \text{for } a_2 = 6, & X_2 = \{x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}\} \\ \text{for } a_3 = 9, & X_3 = \{x_{18}, x_{19}, x_{20}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}\} \\ \text{for } a_4 = 3, & X_4 = \{x_{27}, x_{28}, x_{29}\} \end{array}$$

and value of m_i $(i = 0, 1, 2, 3, 4)$ was also choosen as,

$$\begin{array}{ll} \text{for } i = 0, & m_0 = 0 \\ \text{for } i = 1, & m_1 = (m_0 = 0) + (a_1 = 11) = 11 \\ \text{for } i = 2, & m_2 = (m_1 = 11) + (a_2 = 6) = 17 \\ \text{for } i = 3, & m_3 = (m_2 = 17) + (a_3 = 9) = 26 \\ \text{for } i = 4, & m_4 = (m_3 = 26) + (a_4 = 3) = 29 \end{array}$$

Step 2 : For $i = 1, 2, 3, 4$ values of the X_i-rotation function β_i^{-1} are obtained as follows. Here, we first obtain the values of β_1^{-1} as,

$$\begin{array}{ll} \beta_1^{-1}(x_1) = x_9, & \text{since } 1 + m_1 - r_1 = 1 + 11 - 3 = 9 \text{ because of } 1 + 11 - 3 < 11 \\ \beta_1^{-1}(x_2) = x_{10}, & \text{since } 2 + m_1 - r_1 = 2 + 11 - 3 = 10 \text{ because of } 2 + 11 - 3 < 11 \\ \beta_1^{-1}(x_3) = x_{11}, & \text{since } 3 + m_1 - r_1 = 3 + 11 - 3 = 11 \text{ because of } 3 + 11 - 3 = 11 \\ \beta_1^{-1}(x_4) = x_1, & \text{since } (4 + 11 - 3)(mod\,11) + 0 = 1 \text{ because of } 4 + 11 - 3 > 11 \\ \beta_1^{-1}(x_5) = x_2, & \text{since } (5 + 11 - 3)(mod\,11) + 0 = 2 \text{ because of } 5 + 11 - 3 > 11 \\ \beta_1^{-1}(x_6) = x_3, & \text{since } (6 + 11 - 3)(mod\,11) + 0 = 3 \text{ because of } 6 + 11 - 3 > 11 \\ \beta_1^{-1}(x_7) = x_4, & \text{since } (7 + 11 - 3)(mod\,11) + 0 = 4 \text{ because of } 7 + 11 - 3 > 11 \\ \beta_1^{-1}(x_8) = x_5, & \text{since } (8 + 11 - 3)(mod\,11) + 0 = 5 \text{ because of } 8 + 11 - 3 > 11 \\ \beta_1^{-1}(x_9) = x_6, & \text{since } (9 + 11 - 3)(mod\,11) + 0 = 6 \text{ because of } 9 + 11 - 3 > 11 \\ \beta_1^{-1}(x_{10}) = x_7, & \text{since } (10 + 11 - 3)(mod\,11) + 0 = 7 \text{ because of } 10 + 11 - 3 > 11 \\ \beta_1^{-1}(x_{11}) = x_8, & \text{since } (11 + 11 - 3)(mod\,11) + 0 = 8 \text{ because of } 11 + 11 - 3 > 11 \end{array}$$

and for $i = 2, 3, 4$ the values of β_i^{-1} are obtained similarly. Hence,

X_1	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
β_1^{-1}	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
X_1	x_9	x_{10}	x_{11}	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8

X_2	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}
β_2^{-1}	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
X_2	x_{14}	x_{15}	x_{16}	x_{17}	x_{12}	x_{13}

X_3	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}
β_3^{-1}	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
X_3	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{18}	x_{19}

X_4	x_{27}	x_{28}	x_{29}
β_4^{-1}	$\downarrow$	$\downarrow$	$\downarrow$
X_4	x_{29}	x_{27}	x_{28}

and therefore,

X	x_1	x_2	...	x_{11}	x_{12}	x_{13}	...	x_{17}	x_{18}	x_{19}	...	x_{26}	x_{27}	x_{28}	x_{29}
β^{-1}	$\downarrow$	$\downarrow$		$\downarrow$	$\downarrow$	$\downarrow$		$\downarrow$	$\downarrow$	$\downarrow$		$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
X	x_9	x_{10}	...	x_8	x_{14}	x_{15}	...	x_{13}	x_{20}	x_{21}	...	x_{19}	x_{29}	x_{27}	x_{28}

Step 3: Values of the characterization function α^{-1} are obtained as following list:

X	x_9	x_{10}	...	x_8	x_{14}	x_{15}	...	x_{13}	x_{20}	x_{21}	...	x_{19}	x_{29}	x_{27}	x_{28}
α^{-1}	$\downarrow$	$\downarrow$		$\downarrow$	$\downarrow$	$\downarrow$		$\downarrow$	$\downarrow$	$\downarrow$		$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
U	a	b	...	ı	i	j	...	n	o	ö	...	ü	v	y	z

Step 4: Values of the decryption function γ^{-1} are obtained as following list:

U	ç	d	e	f	g	ğ	h	ı	a	b	c	m	n	o	i	j	k	...	z	v	y
γ^{-1}	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$		$\downarrow$	$\downarrow$	$\downarrow$
U	a	b	c	ç	d	e	f	g	ğ	h	ı	i	j	k	l	m	n	...	v	y	z

In this example we showed that the ciphertext "çliçöç" is decrypted as "ankara".

3 FC Encryption Program Codes

In this section, the FCC-method is programmed by using C# as follows:

```
private static string[] alf_tex()
{
   string[] alphabet = { "a", "b",..,"Y", "Z"}; //caharacter set
   return alpabet;
}
private static int _rnd(int bas, int bit)
```

```
{
   Random rnd = new Random();
   int deger = rnd.Next(bas, bit);
   return deger;
}
private static string _key(int alphabet_number)
{
   string key = "";
   int n = alphabet_number;
   int a,r;
   do
   {
       if (n >= 6)
       {
           a = _rnd(3, n - 3); //Fragment key
       }
       else
       {
           a = n;
       }
       r = _rnd(2, a);//Rotation key
       key += a.ToString() + "," + r.ToString() + '-';
       n = n - a;
   }
   while (n > 0);
   return key;
}
private void btn_creat_alphabet_Click(object sender, EventArgs e)
{
   //key function
   if (rdsifre.Checked)
   {
       string key = _key(alp_tex().Length);
       txtanahtar.Text = key;
   }
   int alfabe_sayac = 1;
   string[] fragment = key.Substring(0,key.Length-1).Split('-');
   string[][] U = new string[fragment.Length][];//(Açık U)
   string[][] SU = new string[fragment.Length][];//(encrypted U)
   //be divided into sets of the alphabet
   for (int j = 0; j < parca.Length; j++)
   {
       string[] parca_a = fragment[j].ToString().Split(',');
```

```
        int P = int.Parse(fragment_a[0]); //fragment key
        U[j] = new string[alp_tex().Length + 1];
        SU[j] = new string[alp_tex().Length + 1];
        for (int x = 0; x < P; x++)
        {
            U[j][alfabe_sayac] = alp_tex()[alphabet_sayac - 1];
            alphabet_sayac++;
        }
    }
    //encrypting alphabet
    int m = 0;
    int index = 1;
    for (int j = 0; j < U.Length; j++)
    {
        string[] parca_a = fragment[j].ToString().Split(',');
        int P = int.Parse(fragment_a[0]); //fragment key
        int R = int.Parse(fragment_a[1]); //rotation key
        m += P;
        for (int x = 0; x < P; x++)
        {
            int k = 0;
            if ((index + R) <= m) //rotation function
            {
                k = index + R;
            }
            else if ((index + R) > m)
            {
                k = ((index + R) % m) + (m - P);
            }
            SU[j][index] = U[j][k];
            index++;
        }
    }
}
```

4 Conclusions

In this work, a cipher method so called fragmented Caesar cipher method is defined. A mathematical modeling and then a computer programs of the method have done. The method is based on the basic logic of Caesar cipher. The classical Caesar cipher is a type of substitution cipher in which each letter is replaced by a letter some

fixed number of positions in the alphabet. The Fragmented Caesar cipher has more possibility then the classical Caesar cipher because of the fragmented alphabet is used to cipher. We finally give an example to show the cipher method is working successfully.

References

[1] Y. Aydoğan, Multi Fragmented Caesar Cipher Method and its Applications (In Turkish), MSc Thesis, Gaziosmanpasa University, Graduate School of Natural and Applied Science, 2014.

[2] S. Dey, Nath and A. J. Nath, An Integrated Symmetric Key Cryptographic Method - Amalgamation of TTJSA Algorithm , Advanced Caesar Cipher Algorithm, Bit Rotation and Reversal Method: SJA Algorithm, I. J. Modern Edu. and Comp. Sci., 5 (2012), 1-9.

[3] J. Hoffstein, J. Pipher and J. H. Silverman, An Introduction to Mathematical Cryptograph, Springer-Newyork, 2008.

[4] A. Mishra, Enhancing Security of Caesar Cipher Using Different Methods, I. J. of Res. in Eng. and Tech., 2(9) (2013), 954-959.

[5] O. E. Omolara, A. I. Oludare and S. E. Abdulahi, Developing a Modified Hybrid Caesar Cipher and Vigenere Cipher for Secure Data Communication, Comp. Eng. and I. Syst., 5(5) (2014).

[6] C. Paar, J. Pelzl, Understanding Cryptograph, Springer-Verlag, 2010.

[7] P. Patni, A Poly-alphabetic Approach to Caesar Cipher Algorithm, I. J. of Comp. Sci. and Info. Tech., 4(6) (2013), 954-959.

[8] P. Patni, Implementation and Result Analysis of Polyalphabetic Approach to Caesar Cipher, IOSR J. of Com. Eng., 16(4) (2014), 100-106.

[9] A. Sinkov, Elementary Cryptanalysis – A Mathematical Approach, New Mathematical Library, No. 22, Mathematical Association of America, 1966.