T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HÜCRESEL DÖNÜŞÜMLERLE ŞİFRELEME VE ANAHTAR ÜRETİMİ

FATİH TEMİZ

YÜKSEK LİSANS TEZİ
MATEMATİK ANABİLİM DALI

DANIŞMAN
PROF. DR. İRFAN ŞİAP

İSTANBUL, 2012

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

HÜCRESEL DÖNÜŞÜMLERLE ŞİFRELEME ve ANAHTAR ÜRETİMİ

Fatih TEMİZ tarafından hazırlanan tez çalışması 27.07.2012 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Prof. Dr. İrfan ŞİAP

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Prof. Dr. İrfan ŞİAP

Yıldız Teknik Üniversitesi

Doç. Dr. Bayram Ali ERSOY

Yıldız Teknik Üniversitesi

Yrd. Doç. Dr. Bahattin YILDIZ

Fatih Üniversitesi

ÖNSÖZ

Çalışmalarım sırasında bana her zaman yol gösteren, bilgi ve tecrübesiyle önümde ışık tutan fakat en önemlisi bana ilham veren saygıdeğer hocam Prof. Dr. İrfan ŞİAP beyfendiye teşekkürlerimi sunarım. Değerli fikir ve önerileriyle bu teze katkıda bulunan jüri üyeleri Sayın Yrd. Doç. Dr. Bahattin YILDIZ ve Doç. Dr. Bayram Ali ERSOY'a da teşekkürlerimi sunarım.

Eğitim ve öğretim hayatımı anlamlı kılan ve tüm imkânlarını benim için seferber eden kıymetli anneme, babama ve kardeşlerime de sonsuz teşekkürler.

Son olarak matematiğin eğlenceli taraflarıyla birlikte çokça vakit geçirdiğimiz değerli arkadaşım Durmuş YANBUL'a ve bilgisayar programlama konusunda önemli katkılarından dolayı değerli arkadaşım Halil İbrahim TAYFUROĞLU'na teşekkür ederim.

Haziran, 2012

Fatih TEMİZ

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ

$\mathbb{Z}_k$	$\{0,1,\ldots,k-1\}$ elemanlarından oluşan sonlu halka
$\mathbb{Z}_k^r$	Bileşenleri $\mathbb{Z}_k$ halkasından alınmış r uzunluklu vektörlerin uzayı
e_k	Şifreleme fonksiyonu
d_k	Deşifreleme fonksiyonu
K^i	$i.$ döngü anahtarı
IP	Başlangıç permütasyonu
E	Genişletme fonksiyonu
$c_i^{(t)}$	i konumundaki hücrenin t anındaki durumu
$C^{(t)}$	Hücresel dönüşümün t anındaki konfigürasyonu
$< i >$	Bir boyutlu hücresel dönüşümde $i.$ hücre
$\{u,v\}$	u ve v kurallarının oluşturduğu hibrit hücresel dönüşüm kuralı

KISALTMA LİSTESİ

DES	Data Encryption Standard
NIST	National Institute of Standards and Technology
IBM	International Business Machine
LCG	Linear Congruential Generator
CA	Cellular Automata (Hücresel Dönüşümler)
HCA	Hybrid Cellular Automata (Hibrit Hücresel Dönüşümler)
Çek	Çekirdek

ŞEKİL LİSTESİ

TABLO LİSTESİ

Sayfa

ÖZET

HÜCRESEL DÖNÜŞÜMLERLE ŞİFRELEME VE ANAHTAR ÜRETİMİ

Fatih TEMİZ

Matematik Anabilim Dalı
Yüksek Lisans Tezi

Tez Danışmanı: Prof. Dr. İrfan ŞİAP

Herhangi bir mesajı ya da bilgiyi istenmeyen kişilerden saklamak için kullanılagelen şifreleme, çağımızda özellikle elektronik ortamlarda çok fazla bilginin dolaşması ve bu bilgilerin her an ele geçirilmesi istenecek derecede önemli olmaları sebebiyle hayati bir hal almıştır.

Şifrelemenin önemi arttıkça matematikçiler başta olmak üzere bilim insanları, sürekli olarak daha iyi şifreleme sistemleri geliştirmeye çalıştılar. Ancak geliştirilen sistemler ne kadar iyi olursa olsun bir şifreleme sisteminin güvenliği anahtarda gizlidir. Bunun için de anahtar üretimi ayrı bir önem arz eder.

Şifrelenmiş bilgilerin ele geçirilmemesi için kullanılan anahtar kırılamamalı ya da tahmin edilememelidir. Bunu sağlayabilmek için de çeşitli yöntemler geliştirilmiş ve bu yöntemlerin gerçekten istenilen güvenilirliği sağlayıp sağlamadıklarına karar verebilecek testler öne sürülmüştür.

Anahtar üretimi için kullanılan yöntemlerin bir tanesi hücresel dönüşüm olarak adlandırılan ayrık dinamik sistemlerdir. Temel olarak dört sınıfa ayrılan hücresel dönüşümlerden üçüncü sınıfı teşkil eden ve kaotik dinamik sistemlere benzeyen hücresel dönüşümler, başlangıç koşullarına oldukça duyarlı oldukları ve rastgelelikte önemli bir rol oynadıkları için anahtar üretimi için kullanılmışlardır.

Bu tezde iki boyutlu melez (hibrit) hücresel dönüşümlerden anahtar üretilmeye çalışılmış, üretilen anahtarlar istatistiksel testlere tabi tutularak güvenilirlikleri ölçülmüştür.

Ayrıca, literatürde ikilik sistemde verilen dizilerin güvenilirliğini ölçen testlerin istatistiksel özellikleri incelenerek bazı testlerin üçlük sistemde verilen dizilerin güvenilirliğini de ölçebilmesi amacıyla çalışmalar yapılmıştır.

Anahtar Kelimeler: Şifreleme, Anahtar Üretimi, Hücresel Dönüşüm

ABSTRACT

CRYPTOLOGY AND GENERATING KEYS VIA CELLULAR AUTOMATA

Fatih TEMİZ

Department of Mathematics
MSc. Thesis

Advisor: Prof. Dr. İrfan ŞİAP

Cryptology, used to hide a message or data from opponents, has been very vital in our age, since there are lots of important data over electronic platforms and opponents try to seize these data constantly.

As the importance of cryptology increases, scientists, especially mathematicians, have studied to develop more secure algorithms and cryptosystems. Even though these algorithms are really secure, whenever the key used in these cryptosystems is vulnerable to some attacks, the cipher will not be secure. Therefore, generating a key is especially important.

To avoid seizing the cipher text, the key used must be unpredictable and not be broken. To provide this, several methods have been proposed. Also some tests have been proposed to decide whether these methods satisfy the security.

One of these methods which are used for generating a key is a dynamical system, called cellular automata. There are four fundamental classes of cellular automata and the third class which are analogous to chaotic dynamical systems have been used for generating key since they are very sensitive to initial conditions and exhibit random behavior.

In this work, generating random keys has been attempted and the generated keys have been tested whether they are secure or not.

Besides, the statistical properties of some basic tests for given binary strings have been studied and some of these tests have been tried to adapt to given ternary strings.

Key words: Cryptology, Generating Key, Cellular Automata

BÖLÜM 1

GİRİŞ

1.1 Hücresel Dönüşümler ve Şifreleme (Literatür Özeti)

Tarihi milattan önceye dayanan şifreleme (kriptoloji) bilimi, özelllikle son yüzyıllarda cebir ve sayılar teorisinin hızlı gelişmesine paralel olarak önemli bir ilerleme kaydetti. Teknolojik ilerlemelerle birlikte telekomünikasyondan banka güvenliğine kadar birçok alanda hayatımızın bir parçası haline geldi.

Temel olarak iki modern şifreleme sistemi kullanılmaktadır: Açık anahtarlı ve gizli anahtarlı (simetrik) şifreleme sistemleri. Şifreleme sistemlerinin güvenliği anahtarda gizlidir [1]. Bir sistemde kullanılan şifreleme algoritması ne kadar iyi olursa olsun, kullanılan anahtar tahmin edilebilir ya da kırılabilirse, bu sistem güvenli olmayacaktır.

Şifreleme sistemlerinde anahtar üretimi, kriptografinin başlı başına bir problemidir. Anahtar üretmek amacıyla çeşitli yöntemler geliştirilmiştir. Bazı çalışmalarda anahtar üretimi için hücresel dönüşümler kullanılmışır. 2004 yılında C. F. Rubio ve arkadaşları, kriptolojide kullanılmak üzere lineer melez (hibrit) hücresel dönüşümleri kullanarak sözde rastgele bit dizisi ürettiler [2].

Çeşitli anahtar üretme metodları üzerinde çalışmalar devam etmektedir.

1.2 Tezin Amacı

Gizli anahtarlı blok şifreleme sistemlerinde anahtar üretiminde hücresel dönüşümlerin kullanımı çalışılmıştır. Bu alanda daha önceden yapılan çalışmalardan yola çıkarak iki boyutlu hücresel dönüşümlerle anahtar üretimi amaçlanmıştır.

Üretilen anahtarların, sözde rastgele sayılabilmeleri için bazı istatistiksel testlerden başarıyla geçmesi gerekmektedir. Fakat bu testler kaynaklarda $\mathbb{Z}_2$ cisminin elemanlarıyla oluşturulan dizilerin rastgeleliğini ölçmek üzere verilmiştir. Bu tezde ise bu testlerden bazıları $\mathbb{Z}_3$ cisminin elemanlarıyla üretilen diziler için uygulanarak, bu dizilerin sözde rastgele dizi olarak kabul edilip edilemeyeceğine karar verebilme amaçlanmıştır.

1.3 Hipotez

İki boyutlu uygun hücresel dönüşümlerin birlikte kullanılmasıyla sözde rastgele bit dizileri elde etmek mümkündür.

$\mathbb{Z}_2$ cisminin elemanlarıyla oluşturulan dizilerin rastgeleliğini ölçen bazı istatistiksel testleri $\mathbb{Z}_3$ cisminin elemanlarıyla elde edilen diziler için de kullanmak mümkündür.

BÖLÜM 2

KRİPTOGRAFİ

Kriptografi, kriptoloji (şifreleme) ve kriptanaliz (şifre çözme) bilimlerinin bilikte meydana getirdikleri bilim dalıdır. Kriptografinin temel amacı güvenli olmayan kanallar üzerinde iki kişi arasındaki iletişimi, iletilen bilgiyi üçüncü bir kişinin elde edemeyeceği şekilde sağlamaktır [3].

Tarihi milattan önce 2000 yıl öncesine kadar uzanan kriptografi son yüzyıllarda özellikle cebir ve sayılar teorisindeki gelişmelerle birlikte günümüzdeki anlamına kavuşmaya başlamıştır. Modern anlamda ilk kez 18. yüzyılın sonunda Thomas Jefferson tarafından icat edilen, Jefferson Diski veya Tekerlek Şifresi olarak adlandırılan bir silindir kullanılarak uygulanmaya başlanan kriptografi o yıllardaki sürekli savaşlar sebebiyle önemini gittikçe artırmıştır [4].

Birinci Dünya Savaşı sonunda bu önem iyice kendini hissettirdi ve 1919 yılında ticari pazar için Enigma makinesi geliştirildi ancak 1932 yılında Polonyalı matematikçi Marian Rejewski tarafından Enigma'nın şifresi çözüldü. İkinci Dünya Savaşı öncesinde Enigma'yı kullanmaya başlayan Almanlar İkinci Dünya Savaşı sırasında Enigma'yı geliştirseler de Polonyalı ve Fransız şifreciler Enigmanın şifresini çözerek büyük mesajları deşifre ettiler. Enigma bu yüzden kötü bir üne sahiptir [5].

20. yüzyılın sonuna yaklaşıldığında ise kriptografi artık yalnızca askerî ve diplomatik alanlarda değil, teknolojinin ve özellikle bilgisayarların gelişmesi ve hayatımızda önemli bir yer edinmesiyle birlikte sivil alanlarda da önemli bir yer buldu. Bankacılık işlemleri,

telekomünikasyon, elektronik ortamlara yahut veritabanlarına erişim ve daha bir çok alanda kriptografi kullanılmaktadır [4],[3],[5].

2.1 Kriptografinin Temel İlkeleri

Şifreleme sistemlerinin, amacı gereği, aşağıdaki kriterlere uygun olması gerekir [4].

- **i.** Güvenlik Seviyesi: Aslında bu kriteri ölçebilmek çok kolay değildir. Genelde, hâlihazırda bilinen en iyi yöntemlerle, ele geçirilmek istenen bilgiye ulaşmak için gereken işlem sayısı olarak verilir. Tipik olarak, bilgiye ulaşmak için gerekli işlem miktarında bir üst sınır ile tanımlanır.
- **ii.** İşlevsellik: Şifreleme sistemlerinin yapıtaşları, değişik bilgi güvenliği hedeflerini gerçekleştirmek için birbirleriyle kombine olmalıdır.
- **iii.** İşlem Metodları: Şifreleme sistemlerinin temel yapıtaşları değişik şekillerde ve değişik girdilerle uygulandığında değişik karakteristikler sergilemelidir.
- **iv.** Performans: Şifrelemedeki temel yapıların belirli bir işlem sırasındaki verimidir. Örneğin, bir şifreleme algoritmasının performansı, saniyede şifreleyebildiği bit sayısıyla ölçülebilir.
- **v.** Uygulamada Kolaylık: Bu kriter, temel yapıların, donanım ya da yazılım ortamlarına uygulanmasındaki karmaşıklıkta olduğu gibi, pratikte uygulamaya geçirilebilme zorluğuna işaret eder.

2.2 Temel Şifreleme Sistemleri

Telefon hattı veya bilgisayar ağı gibi güvenli olmayan bir kanal üzerinde, iki kişi arasındaki iletişimi, üçüncü bir kişinin anlayamayacağı bir şekilde sağlamayı amaçlayan bir şifreleme sistemi matematiksel olarak aşağıdaki gibi tanımlanır [3].

Tanım 2.1 Bir şifreleme sistemi (kriptosistem, *cryptosystem*) aşağıdaki özellikleri sağlayan bir (P, C, K, E, D) beşlisidir [3]:

- **i.** P olası açık metinlerin sonlu bir kümesi;
- **ii.** C olası şifreli metinlerin sonlu bir kümesi;
- **iii.** K, anahtar uzayı, olası anahtarların sonlu bir kümesi;

iv. Her $k \in K$ için bir $e_k \in E$ şifreleme kuralı ve buna karşılık bir $d_k \in D$ deşifreleme kuralı vardır. Her $e_k : P \to C$ ve $d_k : C \to P$ bir fonksiyondur ve her $x \in P$ açık metni için $d_k(e_k(x)) = x$ olur.

2.2.1 Öteleme Şifrelemesi (Shift Cipher)

$P = C = K = \mathbb{Z}_{29}$ olsun. $0 \le K \le 28$ ve $x, y \in \mathbb{Z}_{29}$ için

$$e_k(x) = (x + k) \bmod 29$$

$$d_k(y) = (y - k) \bmod 29$$

olarak tanımlansın. Türk Alfabesindeki harfler ile modülo 29 un kalanlarını aşağıdaki gibi bire bir eşleyerek sıralı bir açık metni şifreleyebiliriz:

A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K
0	1	2	3	4	5	6	7	8	9	10	11	12	13

L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

Şekil 2. 1 Türkçe Alfabenin modülo 29 ile eşlenmesi

Örnek 2.1 Öteleme şifrelemesi için anahtarımızın $k = 7$ olduğunu ve şifrelenmek istenen açık metnin

"cebir"

olduğunu kabul edelim. Öncelikle açık metni tam sayılarından oluşan diziye çevririz:

$$2 \quad 5 \quad 1 \quad 11 \quad 20$$

Sonra her bir değere modulo 29 da 7 ekleyerek dizinin şifrelenmiş halini elde ederiz.

$$9 \quad 12 \quad 8 \quad 18 \quad 27$$

Son olarak elde ettiğimiz tam sayı dizisini alfabetik harflere çevirir ve şifreli metni elde ederiz:

"HJĞÖY"

Anahtarı bilen kişi, aldığı şifreli metni tam sayılara çevirdikten sonra her bir değerden modülo 29 da k çıkartarak açık metne ulaşır. Yani

$$d_k(y) = (y-k) \bmod 29$$

deşifreleme kuralıyla şifrelenmiş metni deşifre eder. Burada anahtar uzayımız K, 29 elemana sahip olduğu için en fazla 29 deneme ile şifrelenmiş metnin kırılabileceği açıktır. Dolayısıyla bu şifreleme sistemi güvenli değildir.

2.2.2 Yer Değiştirme Şifrelemesi (Substitution Cipher)

Bir başka ünlü temel şifreleme sistemi yer değiştirme (substitution) şifrelemesidir ve yüzlerce yıl kullanılmıştır[3]. Bu sistemi matematiksel olarak aşağıdaki gibi ifade edebiliriz:

$P = C = \mathbb{Z}_{29}$ olsun. K, $0,1,...,28$ sayılarının olası tüm permütasyonlarını içersin. Her bir $\pi \in K$ için

$$e_\pi(x) = \pi(x)$$

$$d_\pi(y) = \pi^{-1}(y)$$

olarak tanımlansın. Burada π^{-1}, verilen π' ye ters permütasyondur. Bu şifreleme sisteminde, aslında P ve C kümelerini doğrudan 29 harfli Türkçe alfabe olarak alabiliriz. Öteleme şifrelemesinde $\mathbb{Z}_{29}$ u kullanmamızın sebebi şifreleme ve deşifreleme kurallarının cebirsel işlemler olmasıydı. Fakat yer değiştirme şifrelemesinde, şifreleme ve deşifrelemenin, harflerin permütasyonu olarak düşünülmesi daha uygun olur. Aşağıda bir şifreleme fonksiyonu oluşturan rasgele bir π permütasyonu örneği verilmiştir:

a	b	c	ç	d	e	f	g	ğ	h	ı	i	j	k
P	L	Ğ	D	V	K	O	Ş	Z	B	G	Y	Ö	R

l	m	n	o	ö	p	r	s	ş	t	u	ü	v	y	z
M	A	T	H	I	S	F	U	N	Ü	C	E	J	İ	Ç

Şekil 2. 2 π Permütasyonu

Örnek 2.2 Yukarıdaki permütasyon fonksiyonu ile şifrelenmiş

"İGMVGÇ"

metnini deşifre edelim. Deşifreleme için π fonksiyonun tersini kullanarak her bir harfi deşifre ederiz. π^{-1} fonksiyonu aşağıdaki gibi olacağından

A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K
m	h	u	z	ç	ü	r	ı	c	o	ö	y	v	e

L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z
b	l	ş	f	j	a	k	p	g	n	s	t	d	i	ğ

Şekil 2. 3 π^{-1} permütasyonu

şifrelenmiş her bir harfin π^{-1} fonksiyonu altındaki görüntüsünü alarak açık metni elde ederiz:

"yıldız"

Yer değiştirme şifreleme sisteminde 29 harfin tüm olası permütasyonlarıyla $29!$ farklı anahtar oluşturabiliriz. Bu sayının $4 \cdot 10^{29}$ gibi büyük bir sayıdan daha büyük olduğunu düşünürsek, öteleme şifrelemesinde olduğu gibi anahtarı deneyerek bulmaya çalışmak, bilgisayar kullanarak bile sonuç vermeyecektir. Bu noktada yer değiştirme şifrelemesi güvenli gibi görülebilir ancak, her dilde bulunan harflerin kelimelerde kullanılma sıklığının istatistikleri tutulduğunu göz önünde bulundurursak aslında bu sistemin de güvenli olmadığını görürüz. Tablo 2.1' de Türkçe'deki harflerin kullanım sıklıkları (binde cinsinden) verilmiştir [6].

Harf	Görülme Sıklığı	Harf	Görülme Sıklığı	Harf	Görülme Sıklığı
A	121	I	48	R	68
B	25	İ	107	S	28
C	9	J	0,5	Ş	16
Ç	10	K	47	T	31
D	41	L	62	U	29
E	95	M	37	Ü	15
F	5	N	71	V	8
G	13	O	22	Y	31
Ğ	11	Ö	7	Z	14
H	11	P	8		

Tablo 2. 1 Türkçe Alfabenin frekans tablosu

Benzer tablo veya istatistiklerin, harflerin ikili ya da daha fazla kombinasyonları için de çıkarıldığını göz önünde bulundurursak, fazla tekrar eden harf ya da harf gruplarının ne olduklarını tahmin ederek şifreyi kırmak oldukça kolay olacaktır.

2.2.3 Afin Şifreleme Sistemi (Affine Cipher)

Öteleme şifrelemesi, anahtar uzayı olarak yer değiştirme şifrelemesinin, 29 harfin $29!$ permütasyonundan yalnızca 29 unu içeren özel bir halidir. Yer değiştirme şifrelemesinin bir başka özel hali ise şimdi tanımlayacağımız Afin şifrelemesidir. Afin şifrelemesinde, şifreleme fonksiyonu olarak

$$e(x) = ax + b \mod 29,$$

şeklindeki fonksiyonlar kullanılır. Bu fonksiyonlara Afin fonksiyonları denir. Dikkat edilirse $a = 1$ alınması durumunda öteleme şifrelemesinin elde edileceği açıktır.

Deşifrelemeyi mümkün kılabilmek için, bir afin fonksiyonun ne zaman bire bir olduğunu belirlememiz gerekir. Ya da başka bir ifadeyle, herhangi bir $y \in \mathbb{Z}_{29}$ için

$$ax + b \equiv y \pmod{29}$$

kongrüansının, x için bir tek çözümü olmasını isteriz. Bu kongrüans

$$ax \equiv y - b \pmod{29}$$

kongrüansına denktir. Burada y, $\mathbb{Z}_{29}$ üzerinde gezerken $y-b$ de $\mathbb{Z}_{29}$ üzerinde gezeceğinden $ax \equiv y \pmod{29}$ kongrüansında çalışmak yeterli olacaktır.

Bu kongrüansın, tüm y değerleri için tek çözümünün olmasının gerek ve yeter koşulu $ebob(a,29)=1$ olmasıdır. Aslında bu da $a \in \mathbb{Z}_{29}$'un bu halkada bir tersinin olması demektir[7]. İddiamızı kanıtlamak için önce $ebob(a,29)=d>1$ olduğunu kabul edelim. O halde bu kongrüans $\mathbb{Z}_{29}$ üzerinde en az iki farklı çözüme sahip olacaktır. Bunlar, $x=0$ ve $x=29/d$ dir. Bu durumda ise, $e(x)=ax+b \mod 29$ bire bir fonksiyon olamayacağından geçerli bir şifreleme fonksiyonu olamaz.

Şimdi $ebob(a,29)=1$ olduğunu kabul edelim. Bazı x_1 ve x_2 ler için

$$ax_1 \equiv ax_2 \pmod{29}$$

olsun. O halde

$$a(x_1 - x_2) \equiv 0 \pmod{29},$$

ve dolayısıyla

$$29 \mid a(x_1 - x_2).$$

$ebob(a,29)=1$ olduğu için

$$29 \mid (x_1 - x_2)$$

olmak zorundadır. Yani $x_1 \equiv x_2 \pmod{29}$ dur.

Bire birliği sağladıktan sonra artık değişfreleme için kongrüansın her iki tarafını a^{-1} ile çarparız. $ebob(a,29)=1$ olduğu için böyle bir a^{-1} bulmak mümkündür [7].

$$a^{-1}(ax) \equiv a^{-1}(y-b) \pmod{29}$$

Sonuçta $x \equiv a^{-1}(y-b) \pmod{29}$ deşifreleme fonksiyonunu elde ederiz.

$K = \{(a,b) \in \mathbb{Z}_{29} \times \mathbb{Z}_{29} \mid ebob(a,29)=1\}$ anahtar uzayıdır. 29 asal sayı ve dolayısıyla $\mathbb{Z}_{29}$ bir cisim olduğundan [7] ve cisimde sıfırdan başka her elemanın tersi olduğundan,

anahtar seçimi için bu cisimde bir kısıtlama ($a = 0$ hariç) olmayacaktır. O halde toplam $28 \times 29 = 812$ anahtar kullanılabilir ki bu da güvenilirlik için çok düşük bir sayıdır.

Örnek 2.3 $K = (10, 20)$ anahtarı ile "şifre" açık metni afin şifrelemesine göre aşağıdaki gibi şifrelenir:

$\mathbb{Z}_{29}$ halkasında $10^{-1} = 3$ olduğundan, şifreleme ve deşifreleme fonksiyonları

$$e_k(x) = 10x + 20 \pmod{29}$$

$$d_k(y) = 3(y - 20) = 3y - 27 \pmod{29}$$

olarak elde edilir. Şekil 2.1. deki eşlemeyi kullanarak açık metin matematiksel ifadeye dönüştürülür:

ş i f r e

22 11 6 20 5

Bu değerlerin, $e_k(x) = 10x + 20 \pmod{29}$ şifreleme fonksiyonundaki görüntülerini hesaplayarak şifreli metnin matematiksel ifadesini elde ederiz.

$$e_k(22) = 8, \quad e_k(11) = 14, \quad e_k(6) = 22, \quad e_k(20) = 17, \quad e_k(5) = 12$$

O halde $K = (10, 20)$ anahtarı için,

"şifre"

açık metni, afin şifrelemesinde

"ĞLŞOJ"

olarak şifrelenir.

2.2.4 Vigenere Şifrelemesi

Şimdiye kadar verilen şifreleme sistemlerinde, seçilen anahtarla her harf tek bir harfe dönüştürldü. Şimdi ise faklı bir şifreleme sistemini sunuyoruz [3]. Vigenere şifrelemesi, 16. yüzyılda Blaise de Vigenere tarafından geliştirildi. Vigenere şifrelemesinde, daha önce tanımladığımız gibi, harfler ve sayılar arasındaki eşlemeyi kullanarak, *anahtar*

sözcük adı verilen anahtarımızı, m uzunluğundaki bir harf dizisi şeklinde oluşturabiliriz. Vigenere şifrelemesi m tane alfabetik karakteri aynı anda şifreler; her açık metin m tane alfabetik karaktere eşittir. Küçük bir örnekle açıklayalım:

Örnek 2.4 Farz edelim ki $m = 5$ ve anahtar sözcük "CEBİR" olsun. Bu, sayısal olarak $K = (2,5,1,11,20)$ beşlisine karşılık gelir. Farz edelim ki açık metin aşağıdaki gibi olsun:

buşifresistemigüvenlideğil

Açık metin elemanlarını modülo 29 kalanlarına çevirip, beşerli gruplar halinde yazdıktan sonra, modülo 29 da anahtarsözüğü aşağıdaki gibi ekleriz:

1	24	22	11	6	20	5	21	11	21	23	5	15
2	5	1	11	20	2	5	1	11	20	2	5	1
3	0	23	22	26	22	10	22	22	12	25	10	16
11	7	25	26	5	16	14	11	4	5	8	11	14
11	20	2	5	1	11	20	2	5	1	11	20	2
22	27	27	2	6	27	5	13	9	6	19	2	16

Şekil 2. 4 Vigenere kriptosisteminde şifreleme

Öyleyse, şifreli metnin alfabetik karşılığı şöyle olacaktır:

ÇATŞVŞIŞŞJÜINŞYYCFYEKHFPCN

Deşifreleme işlemi, aynı anahtarsözcük ile bu kez modülo 29 da toplama yerine çıkarma işlemi ile yapılır. Görüldüğü gibi, vigenere şifrelemesinde iki ayrı karakter aynı karaktere veya bir karakter, farklı yerlerde, farklı karakterlere şifrelenebilir. Vigenere şifrelemesinde olası anahtar sayısı 29^m dir ki m nin küçük değerlerinde bile deneyerek anahtara ulaşma oldukça uzun bir zaman gerektirir. Örneğin, $m = 5$ olduğunda anahtar uzayının eleman sayısı 2×10^7 den daha büyük olacaktır. Fakat bu, elle denemek için büyük olsa bile bilgisayar yardımıyla anahtarı bulmak için yeterince büyük bir sayı değildir [3].

2.3 Açık ve Gizli Anahtarlı (Simetrik) Şifreleme Sistemleri

Şifreleme sistemleri açık anahtarlı ve gizli anahtarlı (simetrik) olmak üzere ikiye ayrılır.

Açık anahtarlı şifreleme sistemlerinde herkesin açık ve gizli olmak üzere bir çift anahtarı vardır. Açık anahtarlı şifreleme sistemlerinde amaç verilen e_k şifreleme fonksiyonunu (açık anahtar) kullanarak d_k deşifreleme fonksiyonunu (gizli anahtar) elde etmeyi imkânsız kılmaktır. Bu sayede, e_k şifreleme fonksiyonu açık bir şekilde herkese dağıtılabilir. Aslında açık anahtar ile kapalı anahtar arasında matematiksel bir ilişki vardır ancak bu ilişki çözülememiş matematik problemlerine dayandığı için açık anahtarı kullanarak gizli anahtarı elde etmek mümkün görülmemiştir[1]. Örneğin RSA şifreleme sistemi iki çok büyük asal sayının çarpımının, çarpanlarına ayrılmasındaki zorluğa dayanmaktadır. Bu çarpım açık anahtarı, çarpanları ise gizli anahtarları oluşturur. Bu çarpımı çarpanlarına ayırmak mümkün görülmediğinden, dağıtılmasında bir sakınca yoktur. Dolayısıyla bu açık anahtarı kullanarak herkes şifreleme yapabilir. Ancak deşifreleme için ayrı ayrı çarpanlara ihtiyaç duyulduğundan, yalnızca bu gizli anahtarlara sahip olan kişi mesajı deşifreleyebilecektir.

Gizli anahtarlı şifreleme sistemlerinde, şifreleme ve deşifreleme için kullanılan anahtarlar hem mesajı gönderen kişi tarafından hem de mesajı alan kişi tarafından bilinmektedir çünkü d_k deşifreleme fonksiyonunu bulmak için e_K şifreleme fonksiyonunu bilmek yeterlidir. Modern gizli anahtarlı şifreleme sistemlerinin çoğunda $e_K = d_K$ olduğundan bu şifreleme sistemlerine *simetrik* adı verilmiştir.

Simetrik şifreleme sistemlerinde şifreleme ve deşifreleme için aynı anahtar kullanıldığından, bu anahtar yalnızca haberleşecek kişiler arasında kalmalı ve başkalarına dağıtılmamalıdır. Bu da, haberleşecek olan kişilerin, anahtar belirlemek üzere bir araya gelmesini zorunlu kılmaktadır. Bu açık anahtarlı şifreleme sistemlerine nazaran bir dezavantaj olmakla birlikte simetrik sistemler, açık anahtarlı sistemelere göre daha hızlıdır [1].

2.3.1 Simetrik Şifrelemeye Giriş ve Blok Şifreleme Sistemleri

Simetrik şifreleme sistemlerinin büyük çoğunluğu blok şifrelemele sistemleridir. Günümüzde kullanılan modern blok şifreleme sistemlerinin çoğu ise birden fazla şifreleme sisteminin birlikte kullanımına dayanmaktadır. Örneğin, permütasyon ve yer değiştirme (substitution) gibi. Genel olarak *yinelemeli (iterated)* şifreleme kullanılır. Yinelemeli şifrelemelerde bir *döngü fonksiyonu (round function)* vardır ve her bir döngüde kullanılmak üzere başlangıçtaki anahtardan sabit bir algoritma kullanılarak döngü anahtarları elde edilir. Bu döngü anahtarları $K^1,...,K^{Nr}$ ile gösterilir ve açık metnin şifrelenmesi N_r tane benzer döngü (round) çalıştırılarak yapılır [3].

Döngü fonksiyonunu g ile gösterelim. g iki girdi alır; döngü anahtarı (K^r) ve şimdiki durum (w^{r-1}). Sonraki durum $w^r = g(w^{r-1}, K^r)$ olarak tanımlanır. Başlangıç durumu, w^0, açık metin olarak tanımlanır. y şifreli metni ise N_r döngüden sonraki durum olarak tanımlanır. Yani şifreleme işlemi aşağıdaki gibi gerçekleşir [3]:

$$
\begin{aligned}
w^0 &\leftarrow x \\
w^1 &\leftarrow g(w^0, K^1) \\
w^2 &\leftarrow g(w^1, K^2) \\
&\vdots \qquad\quad \vdots \\
w^{N_r} &\leftarrow g(w^{N_r-1}, K^{N_r}) \\
y \quad &\leftarrow w^{N_r}.
\end{aligned}
$$

Deşifrelemeyi mümkün kılmak için, g fonksiyonu, ikinci girdisi sabit olduğunda bire bir olmalıdır. Yani tüm w ve y ler için aşağıdaki özelliği sağlayacak bir g^{-1} fonksiyonu var olmalıdır;

$$g^{-1}(g(w, y), y) = w.$$

Bu takdirde deşifreleme işlemi ise aşağıdaki gibi yapılır [3]:

$$
\begin{aligned}
w^{N_r} &\leftarrow y \\
w^{N_r-1} &\leftarrow g^{-1}(w^{N_r}, K^{N_r}) \\
\vdots &\qquad \vdots \\
w^1 &\leftarrow g^{-1}(w^2, K^2) \\
w^0 &\leftarrow g^{-1}(w^1, K^1) \\
x &\leftarrow w^0.
\end{aligned}
$$

2.3.2 DES

15 Mayıs 1973'te, o zamanki adı Ulusal Standartlar Bürosu (National Bureau of Standards) olan Ulusal Standartlar ve Teknoloji Enstitüsü (National Institute of Standards and Technology, NIST) ABD resmi gazetesinde (Federal Register) kriptosistemler için bir talep yayınladı. Bu talep, sonunda tüm dünyada en çok kullanılan kriptosistem olan Veri Şifreleme Standardı'nın (Data Encryption Standard, DES) benimsenmesine yol açtı. DES, International Business Machines (IBM) de Lucifer olarak bilinen önceki bir kriptosistemin modifiyesi olarak geliştirildi ve ilk olarak 17 Mart 1975'te resmi gazetede yayınlandı. Önemli ölçüde tartışmalardan sonra DES, Veri Şifreleme Standardı seçilmiştir [3].

DES'in eksiksiz olarak tanımı 1977'de FIPS (Federal Information Processing Standards) de yapılmıştır. DES, yinelemeli (iterated) bir şifreleme olan *Fiestel* şifrelemesinin özel bir çeşitidir. Öncelikle Bölüm 2.3.1' deki terminolojiyle Fiestel şifrelemesinin temellerinden bahsedelim [3]: Bir Fiestel şifrelemesinde, her bir u^i durumu, L^i ve R^i olmak üzere iki eşit uzunluklu parçaya bölünür. g döngü fonksiyonu,

$$
\begin{aligned}
L^i &= R^{i-1} \\
R^i &= L^{i-1} \oplus f(R^{i-1}, K^i)
\end{aligned}
$$

olmak üzere $g(L^{i-1}, R^{i-1}, K^i) = (L^i, R^i)$ formundadır.

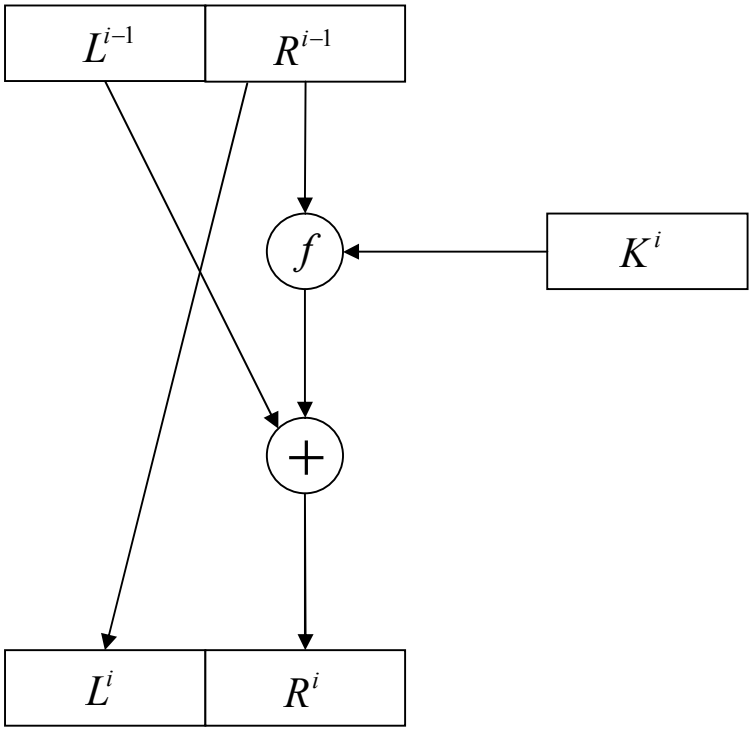
DES, blok uzunluğu 64 olan 16 döngülü bir Fiestel şifrelemesidir; 56-bit uzunluğunda bir anahtar kullanarak 64-bit uzunluğundaki x açık metnini 64-bit uzunluğundaki y şifreli metnine şifreler. Şifrelemenin 16 döngüsünden önce, başlangıç permütasyonu denilen $(\mathbf{IP})$ ve açık metne uygulanan sabit bir permütasyon kullanılır ve şu şekilde gösterilir:

$$\mathbf{IP}(x) = L^0 R^0.$$

16 döngüden sonra, $R^{16}L^{16}$ bit dizisine (bitstring) $\mathbf{IP}^{-1}$ ters permütasyonu uygulanarak y şifreli metni elde edilir. Yani

$$y = \mathbf{IP}^{-1}(R^{16}L^{16})$$

Elbette $\mathbf{IP}^{-1}$ uygulanmadan önce L^{16} ile R^{16} nın yer değiştirmesi gerekir. $\mathbf{IP}$ ve $\mathbf{IP}^{-1}$ permütasyonlarının uygulanmasında kriptografik açıdan bir önem yoktur, bu sebeple DES in güvenilirliği tartışıldığında genellikle görmezden gelinir. DES in bir döngüsü şekil 2.5'de verilmiştir [3].



Şekil 2. 5 DES'in bir döngüsü

Her bir L^i ve R^i 32-bit uzunluğundadır.

$$f : \{0,1\}^{32} \times \{0,1\}^{48} \to \{0,1\}^{32}$$

fonksiyonu, şimdiki durumunun sağ yarısı ile döngü anahtarını iki girdi olarak alır. $K^1, K^2, ..., K^{16}$ döngü anahtarları, 56-bit K anahtarından elde edilen 48-bitlik anahtarlardır. Her bir K^i anahtarı, K' dan belirli bir permütasyonla seçilmiş bitlerdir.

f fonksiyonu Şekil 2.6' da gösterilmiştir. Temel olarak, önce bir yer değiştirme (substitution) ve ardından sabit bir permütasyon $(\mathbf{P})$ ihtiva eder. f fonksiyonunun ilk değişkeninin A ile, ikinci değişkeninin ise J ile gösterildiğini kabul edelim. O halde $f(A,J)$ yi hesaplamak için aşağıdaki yol izlenmelidir.

i. $A,$ sabit bir $\mathbf{E}$ genişletme fonksiyonuna göre 48 uzunluklu bir bit dizisine genişletilir. $\mathbf{E}(A)$, 16 sı iki defa görünmek üzere, A'nın 32 bitini içerir.

ii. $\mathbf{E}(A) \oplus J$ hesaplanır ve sekiz tane 6-bit uzunluğundaki dizi bir araya getirilerek $B = B_1B_2B_3B_4B_5B_6B_7B_8$ dizisi sonuç olarak yazılır.

iii. Sıradaki adımda $S_1,...,S_8$ şeklinde gösterilen sekiz tane $S-$ kutusu kullanılır.

Her bir $S-$ kutusu

$$S_i : \{0,1\}^6 \rightarrow \{0,1\}^4$$

şeklinde bir fonksiyon olmak üzere, altı biti dört bite resmeder ve bileşenleri $0,1,...,15$ tam sayılarından oluşan 4×16 tipinde iki boyutlu bir dizi olarak gösterilir. Altı uzunluğunda bir bit dizisi verilmiş olsun;

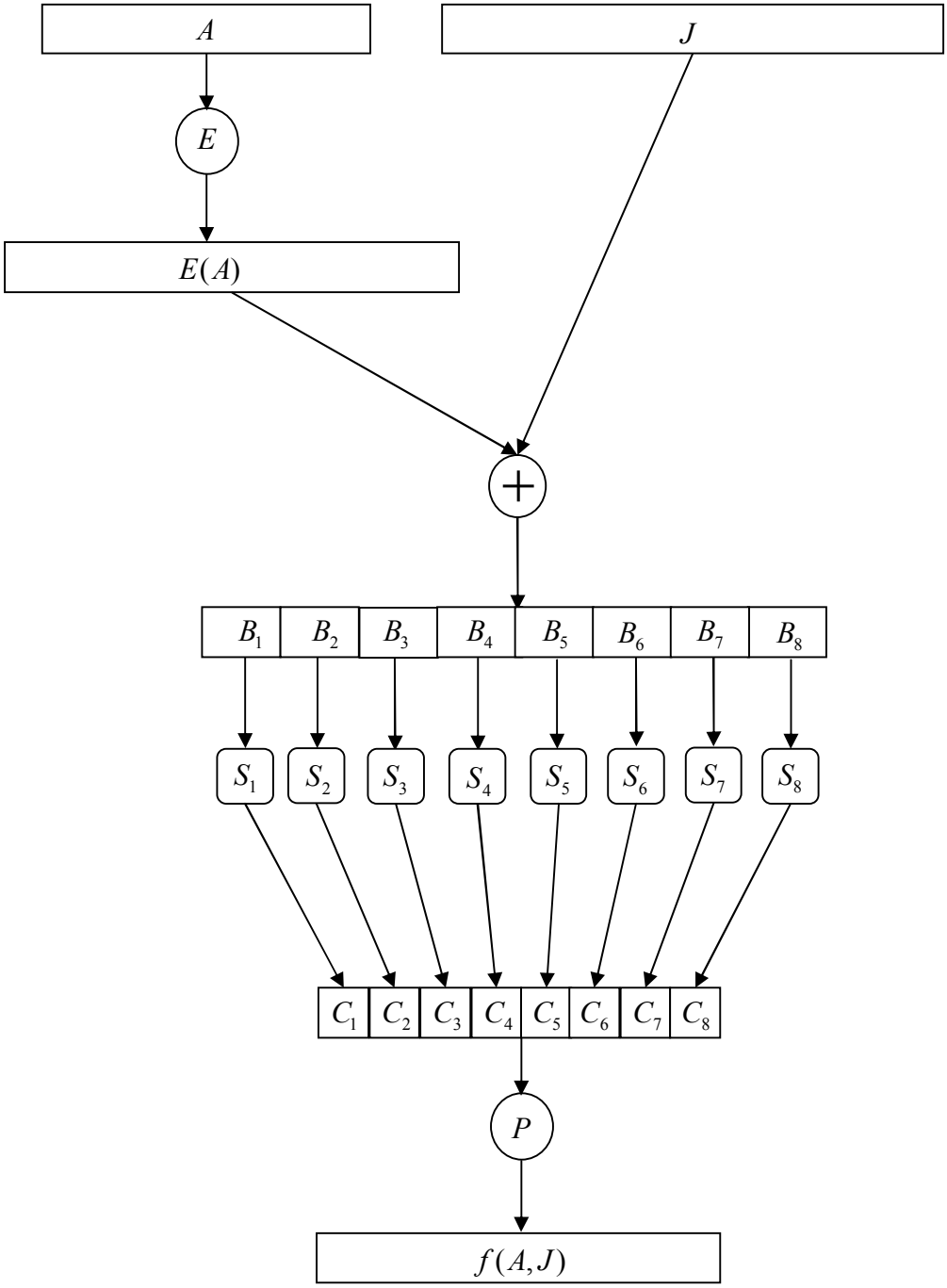
$$B_j = b_1b_2b_3b_4b_5b_6,$$

$S_j(B_j)$ yi şu şekilde hesaplarız: b_1b_6 bitleri S_j'de satır r'nin ikili (binary) gösterimini belirler $(0 \le r \le 3)$ ve $b_2b_3b_4b_5$ bitleri S_j'de kolon c'yi belirler $(0 \le r \le 3)$. Bu takdirde $S_j(B_j)$, $S_j(r,c)$ bileşeninin dört uzunluğundaki ikilik tabanda yazılmış bit dizisidir. Bu şekilde $C_j = S_j(B_j)$' yi hesaplarız $(1 \le j \le 8)$.

iv. 32-bit uzunluğundaki

$$C = C_1C_2C_3C_4C_5C_6C_7C_8$$

bit dizisine $\mathbf{P}$ permütasyonu uygulanır. Sonuçta elde edilen $\mathbf{P}(C)$ bit dizisi $f(A,J)$ olarak tanımlanır [3].



Şekil 2. 6 Döngülerde kullanılan f fonksiyonu

DES şifrelemesinde kullanılan sekiz adet S – kutusu şu şekildedir:

S_1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	2	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	12	8	14	12	0	1	10	6	9	11	15
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	13	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7															
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Örnek 2.4 Şimdi bir örnekle $S-$kutularının çıktılarının nasıl belirlendiğini gösterelim. S_1 kutusunu ele alalım ve 101000 dizisinin girdi olduğunu kabul edelim. İlk ve son bitleri yanyana getirirsek 10 dizisini elde ederiz ki bu 2 tam sayısının ikilik tabandaki gösterimidir. Ortadaki dört biti yanyana getirirsek 0100 i elde ederiz ki bu da 4 tam sayısının ikilik tabandaki karşılığıdır. Satır $2,$ S_1'in üçüncü satırı (çünkü satırlar $0,1,2,3$ ile numaralandırılmıştır) ve benzer şekilde kolon 4, beşinci kolondur. S_1'de, satır $2,$ kolon 4 de 13 tamsayısı bulunur ve ikilik tabanda 1101 ile gösterilir. O halde 101000 dizisi S_1 kutusuna girdiğinde 1101 olarak çıkar.

DES şifrelemesindeki $S-$kutuları permütasyon değildir ancak her bir $S-$kutusunun her bir satırı $0,1,...,15$ tam sayılarının birer permütasyondur.

Döngülerdeki f fonksiyonunda kullanılan $\mathbf{E}$ genişletme fonksiyonu ise aşağıdaki tabloda verilmiştir:

E bit-seçim tablosu					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Tablo 2. 2

32 uzunluğunda $A = (a_1, a_2, ..., a_{32})$ dizisi verilsin, o halde yukarıdaki tabloya göre $\mathbf{E}(A)$ aşağıdaki gibi olur:

$$\mathbf{E}(A) = (a_{32}, a_1, a_2, a_3, a_4, a_5, a_4, ..., a_{31}, a_{32}, a_1).$$

Yine döngülerdeki f fonksiyonunda kullanılan $\mathbf{P}$ permütasyonu ise aşağıdaki gibidir:

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Tablo 2. 3

Verilen bir $C = (c_1, c_2, ..., c_{32})$ dizisi için yuarıdaki tabloya göre $\mathbf{P}(C)$ şöyledir:

$$\mathbf{P}(C) = (c_{16}, c_7, c_{20}, ..., c_4, c_{25})$$

BÖLÜM 3

HÜCRESEL DÖNÜŞÜMLER

Bu bölümün içeriği hazırlanırken Cellular Automata: A Discrete View of the World [8], Cellular Automata, A Discrete Universe [9], Additive Cellular Automata: Theory and Applications [10] isimli kitaplardan ve Mehmet Emin KÖROĞLU'nun "Hücresel Dönüşümlerle Hata Düzelten Kodlar" isimli yüksek linsans tezinden yararlanılmıştır.

Bugünkü anlamıyla hücresel dönüşümler *(cellular automata)* ilk olarak Stanislaw Marcin Ulam ve John von Neumann' ın çalışmaları sonucu ortaya çıkmıştır. Ulam, 1940'lı yılların sonunda kristallerin gelişimini ya da donmayı matematiksel bir modelle ifade etmek için bazı çalışmalar yaptı [11]. Aynı yıllarda Neumann, DNA'nın kendini kopyalamasından hareketle iki boyutlu, 29 durumlu özel bir hücresel dönüşüm sundu. Neumann'ın kendi kendini yeniden üreten makine (self reproducing machine) olarak adlandırılan bu dönüşümü gerçek manada modellenememiştir [8]. Ancak sonraki yıllarda bu dönüşümün daha basit (az sayıda durum içeren) örnekleri modellenmeye çalışılmıştır [12]. 1960'lı yıllara gelindiğinde bu alandaki çalışmaların sayısında ciddi bir artış yaşandı. Bu yıllarda hücresel dönüşümler bir tür özel dinamik sistem olarak çalışılmakla beraber hücresel dönüşümlerin matematiksel özellikleri de incelenmeye başladı [13]. 1970'li yıllarda da bu alandaki çalışmalar ciddi şekilde artış gösterdi. 1980'li yılların başından itibaren Stephen Wolfram hücresel dönüşümlerle ilgili olarak düzenli bir şekilde makaleler yayımladı. Bu makalelerde, bir boyutlu temel hücresel dönüşümler (Wolfram kuralları) detaylı bir şekilde tanıtılmakla beraber bu dönüşümlerin matematiksel özellikleri incelenmiş ve çeşitli uygulama alanlarıyla ilişkilendirilmeye çalışılmıştır. 1980'li yıllardan itibaren hücresel dönüşümler ile birçok bilim dalı arasında ilişki kurulmaya çalışılmıştır.

3.1 Bir Boyutlu Hücresel Dönüşümler

Herhangi bir geometrik şekilden oluşan hücrelerin dizisini (latis) düşünelim. Her bir hücre $k \geq 2$ sonlu tamsayısı kadar farklı durumda var olabilsin. Burada k tane durumdan her biri, geriye kalan $k-1$ durumdan farklı olmalıdır. Olası durumların kümesini $\mathbb{Z}_k = \{0,1,2,...,k-1\}$ ile gösterelim. Bu kümedeki her bir eleman ile renkler, veya birbirinden ayırt edilebilen olgular arasında birebir eşleşme yapılabilir. En basit durum her bir hücrenin, 0 ve 1 ile (görsel olarak beyaz ve siyah ile) gösterilen iki olası durumda var olabileceği $\mathbb{Z}_2 = \{0,1\}$ durumdur.

Tanım 3.1 Bir hücresel dönüşümde her zaman adımı (Şekil 3.1) aynı türden geometrik şekillerin bir dizisinden oluşur. Bu dizilerin her birine o zaman adımındaki *konfigürasyon* denir ve $C^{(t)}$ ile gösterilir. Özel olarak $C^{(0)}$ *başlangıç konfigürasyonu* olarak adlandırılır.

...	$c_{i-1}^{(t)}$	$c_i^{(t)}$	$c_{i+1}^{(t)}$	...

Şekil 3. 1 Bir boyutlu hücresel dönüşüm için latis gösterimi

Hücrelerin latisi $n(\geq 1)$ boyutlu olabilir fakat bu alandaki çalışmalar çoğunlukla bir ve iki boyutludur. Bir boyutlu durumda bu latis, yukarıdaki gibi bitişik kutulardan meydana gelen bir satır halindedir. Temelde, kutular sonsuz sayıdadır fakat uygulanabilirlik açısından kolay olması için dönüşümün davranışını sergileyebilecek büyüklükte almak yeterlidir. Bu, çoğunlukla aşağıda tanımlanacak belirli sınır şartlarıyla sağlanır.

Hücrelerin durumlarının değişmesi için zamanın değişmesi gerekir. Burada zaman ayrık (discrete) kabul edilir ve bu değişim zamanın ayrık bir anında, yani saatin tik-taklarında olduğu gibi $t = 0,1,2,3...,$ *zaman adımlarında* gerçekleşir. Bu zaman adımlarında hücrelerin değişmesine *evrilme (evolution)* denir.

Hücrelerin evrilmesi için gereken bir diğer unsur da *yerel kural (local rule)* ya da *yerel geçiş fonksiyonu (local transition function)* dur. Yerel geçiş fonksiyonu, her bir hücre için bu hücrenin ve bu hücrenin komşu hücrelerinin halihazırdaki durumlarını hesaba

katarak, bir sonraki zaman adımındaki durumunu nasıl değiştireceğini belirleyen fonksiyondur. Elbette burada bir hücrenin komşu hücrelerinin tanımını yapmak gerekir.

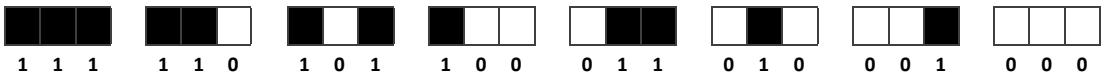
Tanım 3.2 Bir hücre durumunu başka bir duruma evirirken çevresindeki eşit sayıda hücreden etkilenir. Etkileşim halinde bulunulan bu hücrelere *komşu*, sağdaki ya da soldaki hücrelerin sayısına da *komşuluk yarıçapı* denir. Komşuluk yarıçapını r ile göstereceğiz.

Hücrelerin bu değişimi (evrilmesi) latisteki her hücre için eş zamanlı bir şekilde gerçekleşir. Geçiş fonksiyonu deterministik ya da olasılıksal olabilir. Ancak daha önce bahsettiğimiz (anahtar üretimi) sebeplerden bu çalışmada deterministik geçiş fonksiyonlarını kullanacağız. Hücrelerin latisi, olası durumların kümesi, geçiş fonksiyonuyla birlikte bir hücresel dönüşüm *(cellular automaton)* olarak adlandırılır.

r, komşuluk yarıçapı olmak üzere; bir boyutlu hücresel dönüşümler için bir komşulukta toplam $2r+1$ tane hücre vardır. Bu durumda k bir hücrenin bulunabileceği olası durumların sayısı iken toplam k^{2r+1} tane olası komşuluk durumu vardır. Örneğin; olası durumların sayısı $k=2$ ve komşuluk yarıçapı $r=1$ olarak alınırsa toplam $2^{(2\times1)+1}=8$ tane olası komşuluk durumu vardır. Bu durumlar;

$$N=\{\{1,1,1\},\{1,1,0\},\{1,0,1\},\{1,0,0\},\{0,1,1\},\{0,1,0\},\{0,0,1\},\{0,0,0\}\}$$

kümesi ile gösterilebilir. Bu sekiz komşuluk durumunu aşağıdaki gibi de gösterebiliriz.



Şekil 3. 2 Bir boyutlu hücresel dönüşümler için $r=1$ iken olası komşuluk durumları

Şimdi yerel geçiş fonksiyonu tanımını verelim:

Tanım 3.3 $k \geq 2$ olası durumların sayısı, r komşuluk yarıçapı ve $t = 0,1,2,...,n$ ayrık zaman adımları olmak üzere; $\mathbb{Z}_k = \{0,1,2,...,k-1\}$ olsun.

$$f : \mathbb{Z}_k^{2r+1} \to \mathbb{Z}_k$$

$$c_i^{(t+1)} = f(c_{i-r}^{(t)}, c_{i-r+1}^{(t)}, ..., c_{i-1}^{(t)}, c_i^{(t)}, c_{i+1}^{(t)}, ..., c_{i+r-1}^{(t)}, c_{i+r}^{(t)})$$

olarak tanımlanan fonksiyona bir boyutlu hücresel dönüşümler için *yerel geçiş fonksiyonu* denir. Bu tanım d -boyuta da genelleştirilebilir.

Tanım 3.4 $k \geq 2$ olası durumların sayısı, r komşuluk yarıçapı ve t ayırık zaman adımları olsun. $i = 1,...,2r+1$ için $a_i \in \mathbb{Z}_k = \{0,1,\ldots,k-1\}$ olmak üzere

$$\begin{aligned} c_i^{(t+1)} &= f(c_{i-r}^{(t)}, ..., c_{i-1}^{(t)}, c_i^{(t)}, c_{i+1}^{(t)}, ..., c_{i+r}^{(t)}) \\ &= a_1 c_{i-r}^{(t)} + \cdots + a_r c_{i-1}^{(t)} + a_{r+1} c_i^{(t)} + a_{r+2} c_{i+1}^{(t)} + \cdots + a_{2r+1} c_{i+r}^{(t)} \pmod{k} \end{aligned}$$

şeklinde tanımlı

$$f : \mathbb{Z}_k^{2r+1} \to \mathbb{Z}_k$$

fonksiyonuyla belirli kurallara lineer kurallar denir. Bir boyutta toplam k^{2r+1} tane lineer kural vardır.

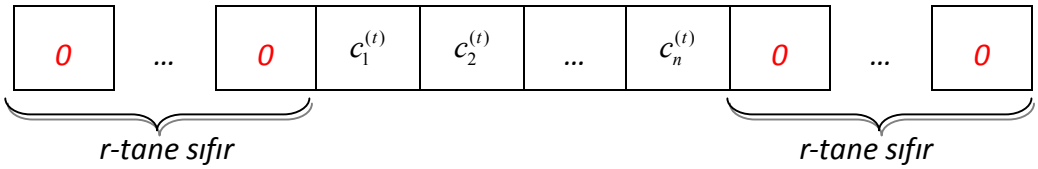
Tanım 3.5 Eğer bir başlangıç dizisindeki (konfigürasyonundaki) hücreler bir sonraki zaman adımında durumlarını belirlerken farklı yerel geçiş fonksiyonları kullanıyorlarsa ya da global geçiş fonksiyonunda bulunan yerel geçiş fonksiyonlarından en az bir tanesi diğerlerinden farklıysa bu hücresel dönüşüme *melez (hybrid)* hücresel dönüşüm denir.

3.1.1 Bir Boyutlu Hücresel Dönüşümler İçin Sınır Şartları

Hücresel dönüşümlerde, verilen bir başlangıç konfigürasyonunun evrilebilmesi için verilen her hücrenin komşularının belirli olması gerekir. Bu sebeple herhangi bir boyutta evirilmenin olabilmesi için başlangıç dizisi sonlu olmalıdır. Bu durumda verilen bir başlangıç dizisinde, sınırda bulunan hücrelerin komşuları için özel bir tanımlama gereklidir. Şimdi bir boyutlu hücresel dönüşümler için sınır şartlarını *(boundary conditions)* tanımlayalım.

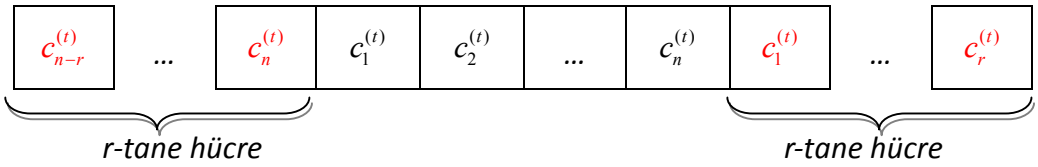
Tanım 3.6 $r < n$ komşuluk yarıçapı olmak üzere $C^{(t)} = \left[c_1^{(t)}, c_1^{(t)}, ..., c_n^{(t)}\right]$ t-zaman adımındaki konfigürasyon olsun.

i. Eğer her bir konfigürasyonun sol baştaki teriminin soluna r tane sıfır ve sağ baştaki teriminin yanına da r tane sıfır eklenerek sırasıyla sol ve sağ baştaki hücrelerin sola doğru ve sağa doğru komşulukları belirleniyorsa bu sınır şartına *sıfır sınır şartı (null boundary condition)* denir.



Şekil 3. 3 Bir boyutlu hücresel dönüşümler için sıfır sınır şartı

ii. Eğer her bir konfigürasyonun sol baştaki terimi ile sağ baştaki terimi bitişikmiş gibi düşünülerek dizinin uç kısmında yer alan hücrelerin r tane komşusu belirleniyorsa bu sınır şartına *periyodik sınır şartı (periodic boundary condition)* denir.



Şekil 3. 4 Bir boyutlu hücresel dönüşümler için periyodik sınır şartı

3.1.2 Bir Boyutlu Elementer Hücresel Dönüşümler (Wolfram Kuralları)

Bir boyutlu hücresel dönüşümlerde $k = 2$ ve komşuluk yarıçapı $r = 1$ için toplam $k^{k^{2r+1}} = 2^{2^3} = 256$ tane kural (yerel geçiş fonksiyonu) vardır. Bu kurallar *temel hücresel dönüşüm (elementary cellular automata)* olarak adlandırılır. Bu kuralların şu şekilde numaralamdırılmıştır:

Şekil 3. 2 deki komşuluk durumlarının her birinin belirlenen kuralla $\mathbb{Z}_2$'de gittiği elemanlar biraraya getirerek bit dizisi şeklinde yazılır. Bu sekiz uzunluğundaki bit dizisinin onluk tabandaki karşılığı o kuralın ismini verir.

Örnek 3.1 f kuralı aşağıdaki gibi tanımlansın:

$$f\left(c_{i-1}^{(t)}, c_i^{(t)}, c_{i+1}^{(t)}\right) = c_i^{(t+1)} = c_{i-1}^{(t)} + c_i^{(t)} + c_{i+1}^{(t)} \pmod 2$$

$c_{i-1}^{(t)}$	$c_i^{(t)}$	$c_{i+1}^{(t)}$	$c_i^{(t+1)}$
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	0

Şekil 3. 5 Kural 150

f kuralının çıktılarını yanyana yazarak elde edilen dizi 10010110 olduğundan ve bu onluk tabanda 150' ye eşit olduğundan f fonksiyonu kural 150 olarak adlandırılır.

3.1.3 Bir Boyutlu Lineer Hücresel Dönüşümler

Lineer kuralların tanımını hatırlarsak, 256 kuraldan

$$f : \mathbb{Z}_2^3 \to \mathbb{Z}_2$$

$$f\left(c_{i-1}^{(t)}, c_i^{(t)}, c_{i+1}^{(t)}\right) = c_i^{(t+1)} = \alpha c_{i-1}^{(t)} + \beta c_i^{(t)} + \theta c_{i+1}^{(t)} \pmod 2;\ \alpha, \beta, \theta \in \mathbb{Z}_2$$

formundaki, komşu hücrelerin doğrusal bir fonksiyonu olan $2^3 = 8$ tane kural lineerdir. Bu kurallar aşağıdaki tabloda gösterilmiştir.

Kural Numarası	Geçiş Fonksiyonu
0. Kural	$c_i^{(t+1)} = 0$
60. Kural	$c_i^{(t+1)} = c_{i-1}^{(t)} + c_i^{(t)} \pmod 2$
90. Kural	$c_i^{(t+1)} = c_{i-1}^{(t)} + c_{i+1}^{(t)} \pmod 2$
102. Kural	$c_i^{(t+1)} = c_i^{(t)} + c_{i+1}^{(t)} \pmod 2$
150. Kural	$c_i^{(t+1)} = c_{i-1}^{(t)} + c_i^{(t)} + c_{i+1}^{(t)} \pmod 2$
170. Kural	$c_i^{(t+1)} = c_{i+1}^{(t)}$
204. Kural	$c_i^{(t+1)} = c_i^{(t)}$
240. Kural	$c_i^{(t+1)} = c_{i-1}^{(t)}$

Şekil 3. 6 Bir boyutlu temel lineer hücresel dönüşümler

3.2 İki Boyutlu Hücresel Dönüşümler (2D-CA)

İki boyutlu hücresel dönüşümü, düzlemdeki aynı geometrik şekillerin iki boyutlu bir dizisi olarak düşünebiliriz.

...	.	.	.	...
...	$c_{(i-1,j+1)}^{(t)}$	$c_{(i,j+1)}^{(t)}$	$c_{(i+1,j+1)}^{(t)}$	...
...	$c_{(i-1,j)}^{(t)}$	$c_{(i,j)}^{(t)}$	$c_{(i+1,j)}^{(t)}$	...
...	$c_{(i-1,j-1)}^{(t)}$	$c_{(i,j-1)}^{(t)}$	$c_{(i+1,j-1)}^{(t)}$	...
...	.	.	.	...

Şekil 3. 7 İki boyutlu hücresel dönüşüm için latis gösterimi

İki boyutlu hücresel dönüşümler, bir boyutlu hücresel dönüşümlerin bazı özelliklerini aynen sergiler. İki boyutlu hücresel dönüşümlerde temel olarak iki komşuluk tanımlıdır. Bu iki komşuluk aşağıdaki şekillerde verilmiştir.

(a) (b)

Şekil 3. 8 (a) Moore komşuluğu, (b) Neumann komşuluğu ($r = 1$)

(a) (b)

Şekil 3. 9 (a) Moore komşuluğu, (b) Neumann komşuluğu ($r = 2$)

Yukarıdaki şekillerde (Şekil 3.8, Şekil 3.9) açık gri renkli hücreler merkez hücreyi, koyu gri renkli hücreler ise komşu hücreleri göstermektedir.

3.2.1 İki Boyutlu Hücresel Dönüşümler İçin Sınır Şartları

Bir boyutlu hücresel dönüşümler için verilen sınır şartları iki boyutta da geçerlidir. Bu sınır şartları, $r = 1$ komşuluk yarıçapı için iki boyutlu hücresel dönüşümlerde şu şekilde gösterilebilir:

i. Sıfır (Null) Sınır Şartı:

0	0	0	0	0
0	$c^{(t)}_{(i-1,j+1)}$	$c^{(t)}_{(i,j+1)}$	$c^{(t)}_{(i+1,j+1)}$	0
0	$c^{(t)}_{(i-1,j)}$	$c^{(t)}_{(i,j)}$	$c^{(t)}_{(i+1,j)}$	0
0	$c^{(t)}_{(i-1,j-1)}$	$c^{(t)}_{(i,j-1)}$	$c^{(t)}_{(i+1,j-1)}$	0
0	0	0	0	0

Şekil 3. 10 İki boyutlu hücresel dönüşümler için sıfır (null) sınır şartı

ii. Periyodik Sınır Şartı:

$c^{(t)}_{(i+1,j-1)}$	$c^{(t)}_{(i-1,j-1)}$	$c^{(t)}_{(i,j-1)}$	$c^{(t)}_{(i+1,j-1)}$	$c^{(t)}_{(i-1,j-1)}$
$c^{(t)}_{(i+1,j+1)}$	$c^{(t)}_{(i-1,j+1)}$	$c^{(t)}_{(i,j+1)}$	$c^{(t)}_{(i+1,j+1)}$	$c^{(t)}_{(i-1,j+1)}$
$c^{(t)}_{(i+1,j)}$	$c^{(t)}_{(i-1,j)}$	$c^{(t)}_{(i,j)}$	$c^{(t)}_{(i+1,j)}$	$c^{(t)}_{(i-1,j)}$
$c^{(t)}_{(i+1,j-1)}$	$c^{(t)}_{(i-1,j-1)}$	$c^{(t)}_{(i,j-1)}$	$c^{(t)}_{(i+1,j-1)}$	$c^{(t)}_{(i-1,j-1)}$
$c^{(t)}_{(i+1,j+1)}$	$c^{(t)}_{(i-1,j+1)}$	$c^{(t)}_{(i,j+1)}$	$c^{(t)}_{(i+1,j+1)}$	$c^{(t)}_{(i-1,j+1)}$

Şekil 3. 11 İki boyutlu hücresel dönüşümler için periyodik (periodic) sınır şartı

3.2.2 İki Boyutlu Elementer Hücresel Dönüşümler (Moore Komşuluklu)

Komşuluk (Moore) yarıçapı iki olan, iki boyutlu hücresel dönüşümde (i, j) konumundaki merkez hücrenin $t+1$ anındaki durumu, çevresindeki sekiz hücrenin ve kendisinin t anındaki durumunun bir fonksiyonu

$$c_{(i,j)}^{(t+1)} = f(c_{(i,j)}^{(t)}, c_{(i+1,j)}^{(t)}, c_{(i+1,j-1)}^{(t)}, c_{(i,j-1)}^{(t)}, c_{(i-1,j-1)}^{(t)}, c_{(i-1,j)}^{(t)}, c_{(i-1,j+1)}^{(t)}, c_{(i,j+1)}^{(t)}, c_{(i+1,j+1)}^{(t)}) \pmod 2$$

olacağından, $\alpha, \beta, \delta, \gamma, \psi, \phi, \mu, \theta, \omega \in \mathbb{Z}_2$ için lineer bir kural aşağıdaki gibidir.

$$c_{(i,j)}^{(t+1)} = \alpha.c_{(i,j)}^{(t)} + \beta.c_{(i+1,j)}^{(t)} + \delta.c_{(i+1,j-1)}^{(t)} + \gamma.c_{(i,j-1)}^{(t)} + \psi.c_{(i-1,j-1)}^{(t)} + \varphi.c_{(i-1,j)}^{(t)} + \mu.c_{(i+1,j-1)}^{(t)} + \theta.c_{(i+1,j)}^{(t)} + \omega.c_{(i+1,j+1)}^{(t)}$$

Bu şekilde $2^9 = 512$ lineer kural bulunabilir. Şekil 3.12 iki boyutlu hücresel dönüşümlerin temel kurallarını gösterir [14].

64	128	256
32	1	2
16	8	4

Şekil 3. 12 İki Boyutlu Temel Hücresel Kurallar

Tablodaki her bir hücrenin içindeki sayı, hücrenin kural numarasını verir. Eğer hücreler bir hücreye değil de birden fazla hücreye bağlı olarak değişiyorsa kural numarası, ilgili hücrelerdeki sayıların toplamı olur. Örneğin 2D-CA 170 (=2+8+32+128) kuralı 2,8,32 ve 128 ile numaralandırılmış hücrelerle ilgilidir ve merkezdeki hücrenin bir sonraki zaman adımındaki durumu etrafındaki bu dört hücrenin modülo 2 deki toplamı olacaktır [14].

Yukarıdaki eşitlikten faydalanarak kural numarasını yerel geçiş fonksiyonundan çıkarmak mümkündür ve tersi de geçerlidir.

$$\textit{Kural Numrası} = \alpha \cdot 2^0 + \beta \cdot 2^1 + \delta \cdot 2^2 + \gamma \cdot 2^3 + \psi \cdot 2^4 + \varphi \cdot 2^5 + \mu \cdot 2^6 + \theta \cdot 2^7 + \omega \cdot 2^8$$

Burada $\alpha, \beta, \delta, \gamma, \psi, \phi, \mu, \theta, \omega \in \mathbb{Z}_2$ değişkenlerinden 1 olanların çarpıldığı kural numaralarının toplamı yerel geçiş kuralının numarasını verir.

3.3 Hücresel Dönüşümlerin Matrislerle İfade Edilmesi

Yukarıda lineer kuralları tanımlamıştık. Matrisler aslında lineer transformasyonlara karşılık geldikleri için, lineer kuralları matrislerle ifade etmek mümkündür. Üstelik bu sayede hücreleri tek tek evirmek yerine bir konfigürasyonu bir seferde evirebiliriz. Örneğin,

$$f\left(c_{i-1}^{(t)}, c_i^{(t)}, c_{i+1}^{(t)}\right) = c_i^{(t+1)} = \alpha c_{i-1}^{(t)} + \beta c_i^{(t)}, + \theta c_{i+1}^{(t)} \left(\mathrm{mod}\, 2\right);\ \alpha, \beta, \theta \in \mathbb{Z}_2$$

lineer Wolfram kurallarını ele alalım. Doğrusallık sayesinde t anındaki konfigürasyonun evrilmesi aşağıdaki ifadedeki gibidir:

$$C^{(t+1)} = T \cdot C^{(t)T} \ (\mathrm{mod}\ 2).$$

Burada $C^{(t)T}$, t anındaki konfigürasyonun matris şeklindeki yazımının transpozudur ve T matrisi aşağıda verilmiştir. Bu T matrisine *temsili matris (representation matrix)* ya da *geçiş matrisi (transition matrix)* denir.

$$T = \begin{pmatrix} \beta & \theta & 0 & \cdots & 0 & \alpha \\ \alpha & \beta & \theta & \cdots & 0 & 0 \\ 0 & \alpha & \beta & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \beta & \theta \\ \theta & 0 & 0 & \cdots & \alpha & \beta \end{pmatrix}$$

Şimdi de periyodik sınır şartı altında iki boyutlu lineer hücresel dönüşümlerin geçiş matrisini ve bu matriste yer alan blok matrisleri verelim [15]:

$$T = \begin{pmatrix} K(\alpha,\beta,\phi) & K(\gamma,\delta,\psi) & 0 & \cdots & 0 & K(\theta,\omega,\mu) \\ K(\theta,\omega,\mu) & K(\alpha,\beta,\phi) & K(\gamma,\delta,\psi) & \cdots & 0 & 0 \\ 0 & K(\theta,\omega,\mu) & K(\alpha,\beta,\phi) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & K(\alpha,\beta,\phi) & K(\gamma,\delta,\psi) \\ K(\gamma,\delta,\psi) & 0 & 0 & \cdots & K(\theta,\omega,\mu) & K(\alpha,\beta,\phi) \end{pmatrix}_{n^2 \times n^2}$$

$$K(\alpha,\beta,\phi)=\begin{pmatrix}\alpha & \beta & 0 & . & . & . & 0 & \phi\\ \phi & \alpha & \beta & 0 & . & . & . & 0\\ 0 & \phi & \alpha & \beta & 0 & . & . & .\\ . & 0 & \phi & . & . & . & . & .\\ . & . & 0 & . & . & . & . & .\\ . & . & . & . & . & . & . & 0\\ 0 & . & . & . & . & . & . & \beta\\ \beta & 0 & . & . & . & 0 & \phi & \alpha\end{pmatrix}_{n\times n},$$

$$K(\gamma,\delta,\psi)=\begin{pmatrix}\gamma & \delta & 0 & . & . & . & 0 & \psi\\ \psi & \gamma & \delta & 0 & . & . & . & 0\\ 0 & \psi & \gamma & \delta & 0 & . & . & .\\ . & 0 & \psi & . & . & . & . & .\\ . & . & 0 & . & . & . & . & .\\ . & . & . & . & . & . & . & 0\\ 0 & . & . & . & . & . & . & \delta\\ \delta & 0 & . & . & . & 0 & \psi & \gamma\end{pmatrix}_{n\times n},$$

$$K(\theta,\omega,\mu)=\begin{pmatrix}\theta & \omega & 0 & . & . & . & 0 & \mu\\ \mu & \theta & \omega & 0 & . & . & . & 0\\ 0 & \mu & \theta & \omega & 0 & . & . & .\\ . & 0 & \mu & . & . & . & . & .\\ . & . & 0 & . & . & . & . & .\\ . & . & . & . & . & . & . & 0\\ 0 & . & . & . & . & . & . & \omega\\ \omega & 0 & . & . & . & 0 & \mu & \theta\end{pmatrix}_{n\times n}.$$

BÖLÜM 4

SÖZDE RASTGELE SAYILAR VE BİT DİZİLERİ

Şifreleme sistemlerinde kullanılan algoritmalar ne kadar iyi olursa olsun, eğer tahmin edilebilir ya da kırılabilir bir anahtar kullanılırsa bu şifreleme sistemi güvenli olmayacaktır. Bu yüzden kullanılan anahtar tahmin edilememeli dolayısıyla belirli bir şablon halinde olmamalı yani *rastgele (random)* olmalıdır. Rastgele bir anahtar ya da sayı dizisi üretmenin çok basit ve etkili yolları olabilir. Örneğin hilesiz madeni bir para ile yazı tura atmak veya hilesiz bir zar atmak vs. Fakat bu, hem uygulanması hem de elektronik ortama ve bilgisayar ortamına adaptesi zor bir yöntemdir. Ayrıca simterik sistemlerde, şifreleme ve deşifreleme için aynı anahtar kullanıldığından, bu yöntem işe yaramayacaktır. O halde, aynı girdilerle aynı sonucu verecek olan bir anahtar üreteci kullanmamız gerekir. Fakat bir fonksiyon ya da algoritma ile elde edeceğimiz bir dizi gerçekte rastgele olmayacaktır. Bu yüzden, *sözde rastgele (pseudorandom)* fikri ortaya atılmıştır. Sözde rastgele bir sayı (ya da dizi) gerçekte rastgele olmayan fakat rastgele bir sayının (dizinin) tüm özelliklerini taşıyan bir sayıdır (dizidir) [4].

Bu bölümde sözde rastgele sayı veya bit dizisi üretmek için bazı tekniklerden bahsedilecek ve elde edilen dizisinin rastgele olup olmadığını kontrol etmek için uygulanan testler hakkında bilgi verilecektir.

4.1 Tanımlar ve Örnekler

Tanım 4.1 İstatistiksel olarak bağımsız ve tarafsız ikili (binary) rakamların bir dizisini çıktı veren bir aygıt ya da algoritmaya *rastgele bit üreteci* denir [3].

Tanım 4.2 Bir *sözde rastgele bit üreteci (pseudorandom bit generator)* k uzunluğunda gerçek rastgele ikili (binary) bir diziyi girdi olarak alıp $l \gg k$ uzunluğunda rastgele gibi "görünen" ikili bir diziyi çıktı olarak veren deterministik bir algoritma veya fonksiyondur. Daha formal olarak şu şekilde ifade edebiliriz:

$l \geq k+1$ olmak üzere k ve l iki pozitif tam sayı olsun. O halde üretecimiz

$$f : \mathbb{Z}_2^k \to \mathbb{Z}_2^l$$

fonksiyonudur. Burada $s_0 \in \mathbb{Z}_2^k$ girdi ya da çekirdek *(seed)*, $f(s_0) \in \mathbb{Z}_2^l$ ise üretilen bit dizisi olarak adlandırılır.

Örnek 4.1 Lineer kongrüansal üreteç (linear congruential generator) $n \geq 1$ olmak üzere aşağıdaki lineer rekürans ile $x_1, x_2, x_3, \ldots$ sayılarının sözde rastgele bir dizisini üretir;

$$x_n = ax_{n-1} + b \pmod{\mathrm{m}}.$$

Burada a, b ve m üreteci karakterize eden parametrelerdir ve x_0 gizlidir *(çekirdek)*.

Örneğin $a = 3,\ b = 5$ ve $m = 31$ alalım ve verilen 5 uzunluklu diziden 10 uzunluklu bir dizi elde edelim. Yukarıdaki eşitlikten $s \mapsto 3s + 5 \mod 31.$ O halde $13 \mapsto 13$ ve diğer 30 kalan sınıfı 30 uzunluğunda bir devir içinde permüt edilir, yani

$$0,5,20,3,14,16,22,9,1,8,$$
$$29,30,2,11,7,26,21,6,23,12,$$
$$10,4,17,25,18,28,27,24,15,19.$$

Eğer girdi olarak 13 haricinde birşey alınırsa, girdi bu devirde bir başlangıç noktası belirler ve modülo 2 ye indirgenmiş sonraki on eleman sözde rastgele dizi olarak alınır. Bu yolla elde edilmiş 31 olası bit dizisi Tablo 4.1 de gösterilmiştir. Örneğin 0'ın girdi olarak alınması ile elde edilen dizi $5,20,3,14,16,22,9,1,8$ tam sayılarının modülo 2 deki değerlerinin dizisidir [3].

çek	dizi
0	1010001101
1	0100110101
2	1101010001
3	0001101001
4	1100101101
5	0100011010
6	1000110010
7	0101000110
8	1001101010
9	1010011010
10	0110010110
11	1010100011
12	0011001011
13	1111111111
14	0011010011
15	1010100011

çek	dizi
16	0110100110
17	1001011010
18	0101101010
19	0101000110
20	1000110100
21	0100011001
22	1101001101
23	0001100101
24	1101010001
25	0010110101
26	1010001100
27	0110101000
28	1011010100
29	0011010100
30	0110101000

Tablo 4. 1 LCG ile Üretilen Bit Dizileri

4.2 Golomb'un Rastgelelik Esasları

Golomb'un rastgelelik esasları (Golomb's randomness postulates) periyodik sözde rastgele bir dizinin rastgele olarak görülebilmesi adına bazı gerek şartlar vermek için yapılan ilk denemelerden biridir. Bu şartların, dizilerin rastgele olarak görülebilmesi için yeterli şartlar olmaktan uzak olduğunu belirtelim. Şimdi ileride kullanacağımız bazı tanımlar ve notasyonları verelim. Bu kısımda aksi belirtilmedikçe diziler ikilik sistemde alınacaktır [4].

Tanım 4.3 $s = s_0, s_1, s_2, \ldots$ sonsuz bir dizi olsun. s dizisinin ilk n teriminden oluşan alt dizileri $s^n = s_0, s_1, \ldots, s_{n-1}$ ile gösterilir [4].

Tanım 4.4 Eğer bütün $i \geq 0$ için $s_i = s_{i+N}$ oluyorsa $s = s_0, s_1, s_2, \ldots$ dizisine $N-$periyodik denir. Eğer s dizisi bazı pozitif N tamsayıları için $N-$periyodik oluyorsa s dizisine *periyodik (periodic)* denir. Periyodik bir s dizisinin *periyod*u, s dizisinin $N-$periyodik olduğu en küçük pozitif tam sayıdır. Eğer s dizisi, periyodu N olmak üzere periyodik ise s dizisinin *deviri* s^N altdizisidir [4].

Tanım 4.5 s bir dizi olsun. s dizisinin ardışık 0'lar ya da ardışık 1'lerden oluşan alt dizisine *"run"* denir. 0'ların oluşturduğu "run"a *boşluk (gap),* 1'lerin oluşturduğu "run"a ise *blok (block)* denir [4].

Tanım 4.6 $s = s_0, s_1, s_2, \ldots$ periyodu N olan periyodik bir dizi olsun. s dizisinin tam sayı değerli otokorelasyon fonksiyonu (autocorrelation function) $C(t)$ ile gösterilir ve aşağıda verilmiştir

$$C(t) = \frac{1}{N}\sum_{i=0}^{N-1}(2s_i - 1)(2s_{i+t} - 1), \quad 0 \le t \le N-1.$$

$C(t)$ otokorelasyon fonksiyonu s dizisi ile s dizisinin t pozisyon ötelenmiş (shifted) hali arasındaki benzerlik miktarını ölçer [4].

Tanım 4.7 s, periyodu N olan periyodik bir dizi olsun. Golomb'un rastgelelik esasları aşağıda verilmiştir:

- **i.** s dizisinin s^N deviri içinde 0'ların sayısı ile 1'lerin sayısının farkı en fazla 1'dir.
- **ii.** s^N deviri içindeki "run"ların en az yarısının uzunluğu 1, en az dörtte birinin uzunluğu 2, en az sekizde birinin uzunluğu 3'tür...
- **iii.** $C(t)$ otokorelasyon fonksiyonu iki değerlidir. Yani bazı K tam sayıları için,

$$N \cdot C(t) = \sum_{i=0}^{N-1}(2s_i - 1)(2s_{i+t} - 1) = \begin{cases} N, & t = 0 \\ K, & 1 \le t \le N-1 \end{cases}$$

biçiminde ifade edilir [4].

Örnek 4.2 $N = 15$ periyodlu, deviri

$$s^{15} = 0,1,1,0,0,1,0,0,0,1,1,1,1,0,1$$

olan s dizisini düşünelim [4].

- **i.** s^{15} dizisinde yedi tane 0 ve sekiz tane de 1 var.
- **ii.** s^{15} dizisinde $1-$uzunluklu dört tane "run" var (2 blok ve 2 boşluk), $2-$uznuluklu iki tane "run" (1 blok ve 1 boşluk), $3-$uzunluklu bir "run" (1 boşluk) ve $4-$uzunluklu bir tane "run" (1 blok) var.
- **iii.** $C(0) = 1$ ve $1 \le t \le 14$ için $C(t) = -1/15$.

4.3 İstatistiksel Testler

Bir bit dizisinin iyi sözde rastgelelik özelliklerine sahip olabilmesi için geçmesi gereken çeşitli istatistiksel testler vardır [4]. Genellikle bit dizilerinin, rastgele bir bit dizisinin sergilmesi gereken özelliklere sahip olup olmadıklarınını belirlemek amacıyla kullanılan beş istatistiksel test, *frekans testi (frequency test), seri testi (serial test), poker testi (poker test), run testi (run test)* ve *otokorelasyon testi (autocorrelation test)*dir. Bu testler Golomb'un rastgelelik esasları temel alınıp özel olarak *(ad hoc)* geliştirilmiştir.

Tanım 4.8 İstatistikte *serbestlik derecesi*, bir istatistiğin kesin olarak hesaplanmasında kullanılan değerlerin sayısının ne kadar değişme serbestisi olduğunu sayısal olarak verir. Örneğin sarı, kırmızı ve mavi toplardan oluşan bir torbada toplam 10 tane top olduğunu biliyorsak herhangi iki renge ait olan top sayısını bilmemiz üçüncü renge ait top sayısını bilmemizi garantiler. Yani serbestlik derecesi ikidir.

Tanım 4.9 Değişkenlerin gözlenen değerlerinin, olasılıkları ile tanımlanmış beklenen değerlerinden ne kadar saptığını gösteren ölçüme *ki-kare testi (chi-squared test)* denir ve bu ölçüm k olası durumların sayı, O_i ler değişkenlerin gözlenen değeri ve E_i ler beklenen değer olmak üzere şu şekilde yapılır:

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

Çeşitli güven aralığındaki eşik değerleri Tablo 4.3' de verilmiştir. Uygun serbestlik derecesi için χ^2 değeri eşik değerinin altında ise değişkenler ki-kare dağılmıştır. Rastgeleliği ölçmek için kullanacağımız beş testten dördü ki-kare testine dayalıdır.

4.3.1 Frekans Testi (Frequency Test)

Bu testin amacı s dizisi içinde, 0 ve 1 lerin sayısının, rastgele bir diziden bekleneceği gibi aşağı yukarı aynı olup olmadığını belirlemektir. n_0, n_1 sırasıyla s dizisindeki 0 ve 1'lerin sayısı olsun. Bu testte

$$X_1 = \frac{(n_0 - n_1)^2}{n}$$

istatistiği kullanılır ve 1 serbestlik derecesi ile eğer $n \geq 10$ ise χ^2 dağılımı gösterir [4].

4.3.2 Seri Testi (Serial Test)

Seri testinin amacı s dizisi içinde $00, 01, 10$ ve 11 altdizilerinin sayısının, rastgele bir diziden bekleneceği gibi aşağı yukarı aynı olup olmadığını belirlemektir. $n_{00}, n_{01}, n_{10}, n_{11}$ sırasıyla $00, 01, 10$ ve 11 altdizilerinin s dizisi içindeki sayısı olsun. Dikkat edilirse burada alt dizilerin kesişmesine izin verildiği için $n_{00} + n_{01} + n_{10} + n_{11} = (n-1)$ olduğu görülür [4]. Bu testte

$$X_2 = \frac{4}{n-1}\left(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2\right) - \frac{2}{n}\left(n_0^2 + n_1^2\right) + 1$$

istatistiği kullanılır ve 2 serbestlik derecesiyle $n \geq 21$ olduğunda χ^2 dağılımı gösterir.

4.3.3 Poker Testi (Poker Test)

Poker testinde s dizisi, $\lfloor n/m \rfloor \geq 5 \cdot 2^m$ olacak şekilde k-tane m-uzunluğunda kesişmeyen parçaya bölünür. n_i, $1 \leq i \leq 2^m$ olmak üzere $i.$ tip dizilerin sayısı olsun. Poker testi s dizisi içinde, m-uzunluğundaki dizilerin, rastgele bir diziden bekleneceği gibi aşağı yukarı aynı sayıda olup olmadığını belirler. Bu test için

$$X_3 = \frac{2^m}{k}\left(\sum_{i=1}^{2^m} n_i^2\right) - k$$

istatistiği kullanılır ve $2^m - 1$ serbestlik derecesiyle χ^2 dağılımı gösterir [4].

4.3.4 Run Testi (Runs Test)

Bu testin amacı değişik uzunluktaki "run"ların (0 veya 1'lerden oluşan) sayısının rastgele bir diziden bekleneceği kadar olup olmadığını belirlemektir. n-uzunluklu rastgele bir dizideki i-uzunluklu blokların (ya da boşlukların) beklenen değeri

$$e_i = \frac{n - i + 3}{2^{i+2}}$$

dir. Şimdi k, $e_i \geq 5$ olacak şekilde en büyük i tamsayısına eşit olsun. Ayrıca s dizisinde $1 \leq i \leq k$ olacak şekilde her i için, i-uzunluklu blok ve boşlukların (gap) sayısını B_i ve G_i ile gösterelim. Bu test için

$$X_4 = \sum_{i=1}^{k} \frac{(B_i - e_i)^2}{e_i} + \sum_{i=1}^{k} \frac{(G_i - e_i)^2}{e_i}$$

istatistiği kullanılır ve $2k-2$ serbestlik derecesiyle χ^2 dağılımı gösterir [4].

4.3.5 Otokorelasyon Testi (Autocorrelation Test)

Bu testin amacı s dizisi ile devirsel olmayan ötelenmiş hali arasında ilişkileri (korelasyon) kontrol etmektir. d, $1 \leq d \leq \lfloor n/2 \rfloor$ olacak şekilde sabit bir tamsayı olsun. s dizisinde, $d-$ ötenlenmiş haline eşit olmayan bitlerin sayısı

$$A(d) = \sum_{i=0}^{n-d+1} s_i \oplus s_{i+d}$$

dir. Bu test için

$$X_5 = \frac{2A(d) - n + d}{\sqrt{n-d}}$$

istatistiği kullanılır ve $n-d \geq 10$ olduğunda $N(0,1)$ dağılım gösterir [4].

Örnek 4.3 Rastgele olmayan, aşağıdaki dizinin dört kez tekrarıyla elde edilen $n = 160$ uzunluklu s dizisini ele alalım:

$$11100\ 01100\ 01000\ 10100\ 11101\ 11100\ 10010\ 01001.$$

- **i.** *(frekans testi)* $n_0 = 84,\ n_1 = 76$ ve X_1 istatistik değeri $0,4.$
- **ii.** *(seri testi)* $n_{00} = 44,\ n_{01} = 40,\ n_{10} = 40,\ n_{11} = 35$ ve X_2 değeri $0,6252.$
- **iii.** *(poker testi)* $m = 3$ ve $k = 53.$ $000, 001, 010, 011, 100, 101, 110, 111$ bloklarının görülme sayıları sırasıyla $5, 10, 6, 4, 12, 3, 6$ ve 7'dir. Öyleyse $X_3 = 9,6415$
- **iv.** *(run testi)* Burada $e_1 = 20,2500,\ e_2 = 10,0625,\ e_3 = 5$ ve $k = 3.$ Sırayla $25,\ 4$ ve 5 tane $1,\ 2$ ve 3 uzunluklu blok ve de sırayla $8,\ 20$ ve 12 tane $1,\ 2$ ve 3 uzunluklu boşluk var. O halde $X_4 = 31,7913.$

v. *(otokorelasyon testi)* Eğer $d = 8$ alınırsa $A(8) = 100$ bulunur. O halde $X_5 = 3,8933.$

α	0.1	0.05	0.025	0.01	0.005	0.0025	0.001	0.0005
x	1.2816	1.6449	1.9600	2.3263	2.5758	2.8070	3.0902	3.2905

Tablo 4. 2 Normal Dağılım Tablosu [4]

	α					
v	0.100	0.050	0.025	0.010	0.005	0.001
1	2.7055	3.8415	5.0239	6.6349	7.8794	10.8276
2	4.6052	5.9915	7.3778	9.2103	10.5966	13.8155
3	6.2514	7.8147	9.3484	11.3449	12.8382	16.2662
4	7.7794	9.4877	11.1433	13.2767	14.8603	18.4668
5	9.2364	11.0705	12.8325	15.0863	16.7496	20.5150
6	10.6446	12.5916	14.4494	16.8119	18.5476	22.4577
7	12.0170	14.0671	16.0128	18.4753	20.2777	24.3219
8	13.3616	15.5073	17.5345	20.0902	21.9550	26.1245
9	14.6837	16.9190	19.0228	21.6660	23.5894	27.8772
10	15.9872	18.3070	20.4832	23.2093	25.1882	29.5883
11	17.2750	19.6751	21.9200	24.7250	26.7568	31.2641
12	18.5493	21.0261	23.3367	26.2170	28.2995	32.9095
13	19.8119	22.3620	24.7356	27.6882	29.8195	34.5282
14	21.0641	23.6848	26.1189	29.1412	31.3193	36.1233
15	22.3071	24.9958	27.4884	30.5779	32.8013	37.6973
16	23.5418	26.2962	28.8454	31.9999	34.2672	39.2524
17	24.7690	27.5871	30.1910	33.4087	35.7185	40.7902
18	25.9894	28.8693	31.5264	34.8053	37.1565	42.3124
19	27.2036	30.1435	32.8523	36.1909	38.5823	43.8202
20	28.4120	31.4104	34.1696	37.5662	39.9968	45.3147
21	29.6151	32.6706	35.4789	38.9322	41.4011	46.7970
22	30.8133	33.9244	36.7807	40.2894	42.7957	48.2679
23	32.0069	35.1725	38.0756	41.6384	44.1813	49.7282
24	33.1962	36.4150	39.3641	42.9798	45.5585	51.1786
25	34.3816	37.6525	40.6465	44.3141	46.9279	52.6197
26	35.5632	38.8851	41.9232	45.6417	48.2899	54.0520
27	36.7412	40.1133	43.1945	46.9629	49.6449	55.4760
28	37.9159	41.3371	44.4608	48.2782	50.9934	56.8923
29	39.0875	42.5570	45.7223	49.5879	52.3356	58.3012
30	40.2560	43.7730	46.9792	50.8922	53.6720	59.7031
31	41.4217	44.9853	48.2319	52.1914	55.0027	61.0983
63	77.7454	82.5287	86.8296	92.0100	95.6493	103.4424
127	147.8048	154.3015	160.0858	166.9874	171.7961	181.9930
255	284.3359	293.2478	301.1250	310.4574	316.9194	330.5197
511	552.3739	564.6961	575.5298	588.2978	597.0978	615.5149
1023	1081.3794	1098.5208	1113.5334	1131.1587	1143.2653	1168.4972

Tablo 4. 3 Ki-Kare Tablosu [4]

$\alpha = 0,05$ güven aralığında eşik değerleri X_1 için $3,8415$, X_2 için $5,9915$, X_3 için $14,0671$, X_4 için $9,4877$ ve X_5 için $1,96$ dır (Tablo 4.2 ve Tablo 4.3). O halde verilen s dizisi frekans, seri ve poker testlerinden geçerken "run" ve otokorelasyon testlerinden geçemez [4].

4.4 $\mathbb{Z}_2$ Üzerindeki İstatistiksel Testlerin $\mathbb{Z}_3$ Halkasına Uyarlanması

Bu bölümde yukarıda $\mathbb{Z}_2$ halkasının elemanlarıyla oluşturulan dizilerin rastgeleliğini ölçmek için kullanılan testlerin $\mathbb{Z}_3$ halkasının elemanlarıyla oluşturulan dizilerin de rastgeleliğini ölçebilmesi için yapılan çalışmalar, bölüm 4.3 ün notasyonları kullanılarak aktarılacaktır.

4.4.1 $\mathbb{Z}_3$ Üzerinde Frekans Testi

Üçlük sistemde frekans testi, dizide bulunan $0,1$ ve 2 lerin sayısının hemen hemen aynı olup olmadığını inceler. Bunun için değişkenlerin beklenen değerlerini ve serbestlik derecesini belirlememiz gerekir. Öncelikle serbestlik derecesinden bahsedelim. $\mathbb{Z}_3$ halkasında üç eleman (değişken) vardır fakat bu üç değişkenden herhangi ikisinin sayısını bilirsek üçüncüyü de bileceğimiz için serbestlik derecesi $3-1=2$ dir. Dolayısıyla frekans testi χ^2 dağılımı gösterdiğinden

$$\chi^2 = \sum_{i=1}^{3} \frac{(n_i - e_i)^2}{e_i}$$

değerini hesaplayıp eşik değerini aşıp aşmadığını χ^2 tablosunda kontrol etmemiz gerekir. Burada n_i, dizide bulunan i' lerin $(0 \le i \le 2)$ sayısı ve e_i de beklenen değeridir. Eğer dizi n uzunluğunda ise $e_i = \frac{1}{3} \cdot n = \frac{n}{3}$ olur.

4.4.2 $\mathbb{Z}_3$ Üzerinde Seri Testi

Üçlük sistemde seri testini uygulayabilmemiz için yine gerekli serbestlik derecesini ve beklenen değerleri hesaplamamız gerekir. Seri testi iki uzunluğundaki alt dizilerin sayısıyla ilgileniyor ve bu alt dizilerin kesişmesine izin veriliyordu. Eğer dizimiz n uzunluğunda ise ve dizide bulunan $00, 01, 02, 10, 11, 12, 20, 21$ ve 22 altdizilerinin sayısı sırayla $n_{00}, n_{01}, n_{02}, n_{10}, n_{11}, n_{12}, n_{20}, n_{21},$ ve n_{22} ise, yine

$$n_{00} + n_{01} + n_{02} + n_{10} + n_{11} + n_{12} + n_{20} + n_{21} + n_{22} = n - 1$$

olacaktır. Burada dokuz değişken olduğu için serbestlik derecesinin sekiz olması gerektiği düşünülebilir fakat bu bizi yanılgıya düşürür çünkü burada bahsedilen alt dizilerin kesişmesine izin verilir ve bu da bu alt dizilerin sayılarının birbiriyle ilintili yapar. Örneğin 01 alt dizisini ele alalım. Eğer dizi bundan sonra tamamen 1 ile devam etmiyorsa bir yerde mutlaka 10 ya da 12 alt dizisi olmak zorundadır. Aynı şey 21 alt dizisini düşündüğümüzde de geçerlidir; eğer 21 den sonra dizi tamamen 1 ile devam etmiyorsa bir yerde mutlaka 10 ya da 12 alt dizisi olmak zorundadır. Bu bize aynı sayıyla başlayan alt dizilerin toplamının, o sayıyla biten alt dizilerin toplamına eşit olduğunu gösterir. O halde

$$\begin{aligned} n_{01} + n_{02} &= n_{10} + n_{20} \\ n_{10} + n_{12} &= n_{01} + n_{21} \\ n_{20} + n_{21} &= n_{02} + n_{12} \end{aligned}$$

eşitliklerini elde ederiz. Bu denklem sistemini matrisler ile ifade edecek olursak

$$\begin{pmatrix} 1 & 1 & -1 & -1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & -1 \\ 0 & -1 & 0 & 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} n_{01} \\ n_{02} \\ n_{10} \\ n_{20} \\ n_{12} \\ n_{21} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

eşitliğini elde ederiz. Dikkat edilirse soldaki matrisin satırları lineer bağılmıdır. O halde

$$\begin{pmatrix} 1 & 1 & -1 & -1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} n_{01} \\ n_{02} \\ n_{10} \\ n_{20} \\ n_{12} \\ n_{21} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

sistemini elde ederiz. Altı değişken ve iki eşitliğimiz olduğu için parametre sayısı dörttür. $00,11,22$ alt dizilerinin sayıları birbirlerinden ve diğer alt dizilerden bağımsız olacağından serbestlik derecesi $4+3-1=6$ dır.

$n_{00}=o_o,\ n_{01}=o_1,\ldots,\ n_{22}=o_8$ olarak ifade edelim. O halde

$$\chi^2=\sum_{i=0}^{8}\frac{(o_i-e_i)^2}{e_i}$$

istatistiği hesaplanır ve tablodan kontrol edilir. Burada $e_0,\ldots,e_8$, sırayla $n_{00},\ldots,n_{22}$ nin beklen değeri olup dokuz değişken ile elde edilen toplam sayı $n-1$ olduğuna göre $e_i=\frac{1}{9}(n-1)$'dir.

4.4.3 $\mathbb{Z}_3$ Üzerinde Poker Testi

Poker testinde $m-$uzunluğundaki alt dizilerin kesişmesine izin verilmediği için alt dizi sayıları biri hariç diğerlerinden bağımsız olacaktır. O halde serbestlik derecesi 3^m-1 olarak hesaplanır. Beklenen değer ise her bir altdizi için $e_i=\frac{n}{m}\cdot\frac{1}{3^m}=\frac{k}{3^m}$ olacaktır.

4.4.4 $\mathbb{Z}_3$ Üzerinde Run Testi

İkilik sistemde çalışırken blok ve boşlukların beklenen değerlerini verilen formülle hesaplamıştık. Bu formül, üzerinde çalıştığımız halkanın eleman sayısından bağımsızdı yani ikilik sistem için özel olarak verilmişti. O halde üçlük sistemde bu testi doğrudan kullanamayız. Öyleyse $\mathbb{Z}_3$ halkasının elemanlarıyla oluşturulan bir dizideki blok ve boşlukların beklenen değerlerini bulmamız gerekir. Aslında bu beklenen değerler, n uzunluğundaki tüm olası diziler için, o dizinin içindeki i uzunluklu blok sayılarının

toplamı ile o dizinin tüm olası dizilerin içinden seçilmesi olasılığının çarpımlarının toplamlarıdır. Diğer bir ifadeyle, i uzunluğundaki blokların beklenen değerleri, n uzunluğundaki tüm olası dizidelerde, i uzunluğundaki bütün olası blokların toplam sayısının n uzunluğunda yazılabilecek tüm dizilerin sayısına $\left(3^n\right)$ bölümüdür. O halde bu beklenen değerleri hesaplayabilmek için $\mathbb{Z}_3$ halkası üzerinde n uzunluğundaki tüm olası dizilerde bulunan tüm i uzunluklu blokların sayısını bulmamız gerekir. Bunun için özyinelemeli (recursive) bağıntı kullanacağız. Boşluklar için gösterecek olursak öncelikle $n=1$ uzunluklu dizileri ele alalım:

$$0$$
$$1$$
$$2$$

İki uzunluğundaki tüm dizileri elde etmek için bir uzunluğundaki tüm dizilerin başına olası durumları getirme yolunu izleyebiliriz. Bu yalnızca 1 uzunluğundan 2 uzunluğuna geçerken değil, genel olarak n uzunluğundan $n+1$ uzunluğuna geçerken de yapılabilir. İki uzunluğundaki tüm dizilerin bu yolla elde edilişi aşağıdaki gibidir:

$$\begin{array}{ccc} 0 & 0 & 0 \\ 0\ 1 & 1\ 1 & 2\ 1 \\ 2 & 2 & 2 \end{array}$$

$$\begin{array}{ccc} 00 & 10 & 20 \\ 01 & 11 & 21 \\ 02 & 12 & 22 \end{array}$$

Pekala 1 uzunluklu dizilerdeki boşluk sayısı ile 2 uzunluklu dizilerdeki boşlukların sayısı arasında genel bir ilişki var mıdır? Bu sorunun cevabını vermek 2 uzunluğundan 3 uzunluğuna geçerken daha kolay olacaktır. Yukarıdaki gibi iki uzunluğundaki tüm dizileri alalım ve hepsinin başına $0,1$ ve 2 eklediğimiz durumları inceleyelim:

Öncelikle hepsinin başına 0 eklediğimiz durumları ele alalım:

Bu durumu da iki durum içerisinde inceleyebiliriz:

- **i.** 0 ile başlayan dizilerin başına 0 ekleme.
- **ii.** 1 ya da 2 ile başlayan dizilerin başına 0 ekleme.

0	00
0	01
0	02
0	10
0	11
0	12
0	20
0	21
0	22

0 ile başlayan dizilerin başına 0 getirdiğimizde bir uzunluklu boşlukların sayısında bir artış olmaz aksine azalma olur. Çünkü bir uzunluklu boşlukların başına 0 getirdiğimizde bunlar artık iki uzunluklu boşluğa dönüşür. Peki bu azalmanın miktarı ne kadardır? Bu azalma elbette 0 dan sonra 1 veya 2 nin geldiği durumlarda olacaktır. Yani bu örnek için $2,$ genelde ise $3^{n-1}\cdot\frac{2}{3}$ olur. Örneğin, yukarıdaki dizileri incelersek 2 uzunluğundaki 0 ile başlayan dizilerin içinde, 1-uzunluklu toplam boşluk sayısı 2 iken, hepsinin başına 0 eklendiğinde artık 1 uzunluğunda boşluk kalmamıştır.

İkinci durumda yani 1 veya 2 ile başlayan dizilerin başına 0 eklediğimizde ise her dizi için 1 tane 1 uzunluğunda boşluk meydana gelir. O halde ikinci durum için 1 uzunluklu boşlukların sayısında bu örnek için altı, genel olarak ise $3^n\cdot\frac{2}{3}$ tane artış olacaktır. Öyleyse bu örnek için iki uzunluğundaki tüm dizilerin başına 0 eklediğimizde altı tane yeni 1 uzunluğunda boşluk oluşacak ve iki tane de eksilecektir. 2-uzunluğundaki tüm dizilerde toplam dört tane 1uzunluklu boşluk olduğu için $4+6-2=8$ tane boşluk elde ederiz. Bu 3 uzunluğunda 0 ile başlayan dizilerdeki toplam boşluk sayısıdır. Pekala 2 uzunluklu tüm dizilerin başına 1 veya 2 getirdiğimizde boşluk sayısı hakkında ne söyleyebiliriz?

Bu durumu da yine iki durumda inceleyelim:

i. 0 ile başlayan dizilerin başına 1 veya 2 ekleme.

ii. 1 veya 2 ile başlayan dizilerin başına 1 veya 2 ekleme.

1	00
1	01
1	02
1	10
1	11
1	12
1	20
1	21
1	22

İki durum için de boşluk sayısında değişme olmadığı kolaylıkla görülebilir. Fakat burada dikkat edilmesi gereken şey önceki adımda (iki uzunluklu dizilerde) toplam ne kadar boşluk varsa, hepsinin başına 1 eklediğimizde –ki bu yeni uzunluk için tüm olası durumların üçte biridir- yine o kadar boşluğun üç uzunluğundaki dizilerde de olacağıdır. Aynı durum iki uzunluğundaki dizilerin başına 2 eklediğimiz durumda da geçerlidir. O halde olası tüm durumlar için 1 uzunluğundaki boşluk sayısı $+6-2+4+4=12$ artmıştır. İki uzunluğundaki olası tüm dizilerde dört tane 1-uzunluğunda boşluk vardı. Şimdi üç uzunluğundakilere bakalım.

200	100	000
201	101	001
202	102	002
210	110	010
211	111	011
212	112	012
220	120	020
221	121	021
222	122	022

Görüldüğü gibi bir uzunluğundaki boşlukarın sayısı 12 artmış ve 16 olmuştur. Şimdi genel durum için formül verelim. g_i^n n-uzunluklu dizilerde bulunabilecek i uzunluklu toplam boşluk sayısı ve $g_i^{n_k}$ n uzunluklu, k ile başlayan dizilerde bulunabilecek toplam boşluk sayısı olsun. O halde

$$g_1^n = \begin{cases} g_1^{(n-1)_0} - 2\cdot 3^{n-3} & + g_1^{(n-1)_1} + 3^{n-2} & + g_1^{(n-1)_2} + 3^{n-2} \\ + g_1^{(n-1)_0} & + g_1^{(n-1)_1} & + g_1^{(n-1)_2} \\ + g_1^{(n-1)_0} & + g_1^{(n-1)_1} & + g_1^{(n-1)_2} \end{cases}$$

Bu eşitlikten de

$$g_1^n = 3\left(g_1^{(n-1)}\right) + 2\left(3^{n-2} - 3^{n-3}\right) = 3\left(g_1^{(n-1)}\right) + 4\cdot 3^{n-3}$$

elde edilir. O halde özyinelemeli (recursive) olarak

$$g_2^n = 3\left(g_2^{(n-1)}\right) + 2\left(3^{n-3} - 3^{n-4}\right) = 3\left(g_2^{(n-1)}\right) + 4\cdot 3^{n-4}$$
$$g_3^n = 3\left(g_3^{(n-1)}\right) + 2\left(3^{n-4} - 3^{n-5}\right) = 3\left(g_3^{(n-1)}\right) + 4\cdot 3^{n-5}$$

eşitliklerini elde ederiz. Boşluklar için yaptığımız işlemlerin hepsi 1 ve 2 den oluşan bloklar için de geçerlidir. Artık herhangi bir uzunluğa sahip dizilerdeki boşlukları veya blokları sayabildiğimiz için bu sayıları $\frac{1}{3^n}$ ile çarparak beklenen değerini bulabiliriz.

Serbestlik derecesinden de bahsedecek olursak, ikilik sistemde $2k-2$ olarak verilmişti. Bunun nedeni aslında k ve daha düşük uzunluklu bloklarla ilgileniyor olmamızdı. Dolayısıyla ikilik sistemde iki sayı olduğundan $2k$ tane değişken ile ilgileniyorduk:

$$\begin{array}{cccccc} 0 & 00 & 000 & \cdots & \overbrace{00\cdots0}^{k-\text{tane}} \\ 1 & 11 & 111 & \cdots & 11\cdots1 \end{array}$$

Dikkat edersek 0'dan oluşan "run" sayısının, 1'lerden oluşan "run" sayısına eşit olması gerektiğini (ya da aralarında bir fark olması gerektiğini) görürüz. Çünkü her 1 bloğu için eğer dizinin geri kalan kısmı 1'lerden oluşmuyorsa 0'lardan oluşan bir "run" vardır. Bu yüzden de serbestlik derecesi $2k-2$ idi. Bu bağlamda üçlük sistemde verilen bir dizi için de serbestlik derecesi $3k-1$ olacaktır. Son olarak 2 den oluşan blokları D ile gösterirsek

$$\chi^2 = \sum_{i=1}^{k} \frac{\left(G_i - e_i\right)^2}{e_i} + \sum_{i=1}^{k} \frac{\left(B_i - e_i\right)^2}{e_i} + \sum_{i=1}^{k} \frac{\left(D_i - e_i\right)^2}{e_i}$$

hesaplanmalı ve χ^2 tablosundan kontrol edilmelidir.

BÖLÜM 5

ANAHTAR ÜRETİMİ ve UYGULAMALARI

Bu bölümde simetrik şifrelemelerde anahtar olarak kullanılmak üzere rastgele dizilerin elde edilmesi için hücresel dönüşümlerin kullanılması ve bazı uygulamalarından bahsedilecektir.

5.1 Hücresel Dönüşümlerle Anahtar Üretimi

Hücresel dönüşümler, çok basit yollarla sözderastgele bit üreteci olarak kullanılabilirler. Örneğin, n-uzunluğundaki $C^{(0)} = \left(c_0^{(0)}, \ldots, c_{n-1}^{(0)}\right)$ başlangıç konfigürasyonunu $k-1$ defa evirip $C^{(0)}, \ldots, C^{(k-1)}$ konfigürasyonlarını bir araya getirerek $k \cdot n \gg n$ uzunluğunda bir dizi elde edebilir. Fakat elbette bu kriptografik açıdan güvenli değildir. Eğer üçüncü bir şahıs, oluşturulan bu dizinin bir kısmını ele geçirir ya da öğrenirse ve kullanılan hücresel dönüşüm kuralını biliyorsa, dizinin geri kalan kısmını da kolaylıkla elde edebilir [2].

Bunun yerine her bir konfigürasyondan sabit bir hücre seçip, bu hücreleri bir araya getirerek anahtar üretmek daha güvenli olacaktır; $\left(c_i^{(0)}, c_i^{(1)}, c_i^{(2)}, \ldots\right)$. Bu yöntem öncekine göre daha zahmetli ve pahalıdır ancak dizinin bir kısmını ele geçiren üçüncü şahıslar için dizinin geri kalan kısmını elde etmek daha zor olacaktır. Çünkü her bir konfigürasyondan yalnızca bir hücrenin durumu bilinmektedir [2].

Örneğin, Wolfram [1986] tek bir siyah hücreyi Kural 30'a göre 960 defa evirerek merkezdeki hücrenin $t=0, t=1, \ldots, t=959$ anındaki durumlarından meydana getirdiği $1101110011\ldots$ dizinisinin rastgele olduğunu görmüştür [8].

Şekil 5. 1 Kural 30

C. Fraile Rubio ve arkadaşları bu yöntemi kullanarak bir boyutlu lineer hibrit hücresel dönüşümleri test etmişlerdir [2]. Şimdi kullandıkları yöntemi inceleyelim:

C. Fraile Rubio ve arkadaşları iki Wolfram kuralının kombinasyonunu kullandılar. $u<v$ ve $\alpha_u, \beta_u, \gamma_u, \alpha_v, \beta_v, \gamma_v \in \mathbb{Z}_2$ olmak üzere hücreleri evirmek için

$$WCA(u) \equiv c_i^{(t+1)} = \alpha_u c_{i-1}^{(t)} + \beta_u c_i^{(t)} + \gamma_u c_{i+1}^{(t)} \pmod 2,$$
$$WCA(v) \equiv c_i^{(t+1)} = \alpha_v c_{i-1}^{(t)} + \beta_v c_i^{(t)} + \gamma_v c_{i+1}^{(t)} \pmod 2,$$

kuralları kullanılmış ve bu hibrit kural $\{u,v\}$ ile gösterilmiştir. Eğer hibrit hücresel dönüşümdeki hücrelerin sayısı n ise $(\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{n-1})$ konfigürasyonu şu şekilde evrilir: Eğer $\varepsilon_i = 0$ ise $\langle i \rangle$ hücresi $WCA(u)$ kuralına göre, $\varepsilon_i = 1$ ise $\langle i \rangle$ hücresi $WCA(v)$ kuralına göre evrilir. Bu hibrit kuralın periyodik sınır şartı altında matrisle gösterimi şu şekildedir:

$$M = \begin{pmatrix} \beta_{\xi_0} & \gamma_{\xi_0} & 0 & \cdots & 0 & \alpha_{\xi_0} \\ \alpha_{\xi_1} & \beta_{\xi_1} & \gamma_{\xi_1} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma_{\xi_{n-1}} & 0 & 0 & \cdots & \alpha_{\xi_{n-1}} & \beta_{\xi_{n-1}} \end{pmatrix}$$

ve

$$\xi_i = (1-\varepsilon_i)u + \varepsilon_i v, \quad 0 \le i \le n\text{-}1.$$

Bu yöntemle 512-bit uzunluğunda rastgele bir bit dizisi alınmış ve 1024-bit uzunluğunda sözde rastgele bit dizsi üretilmiştir. 28 adet lineer hibrit hücresel dönüşümden 10 tanesinin ürettiği bit dizileri bu testlerden geçmeyi başarmıştır.

Biz de iki boytulu hücresel dönüşümlerde lineer kuralları kullanarak sözde rastgele dizi elde etmeye çalıştık. İki boyutlu (Moore komşuluklu) toplam $\binom{512}{2} \approx 2^{17}$ tane hibrit lineer kural bulunabileceğinden, bu kuralların hepsini test etmek yerine, sözsderastgele dizi üretebilen özel kurallar aradık. Örneğin kural 45 ve kural 95'i ele alalım:

$$45 = 1.2^0 + 0.2^1 + 1.2^2 + 1.2^3 + 0.2^4 + 1.2^5 + 0.2^6 + 0.2^7 + 0.2^8$$

$$95 = 1.2^0 + 1.2^1 + 1.2^2 + 1.2^3 + 1.2^4 + 0.2^5 + 1.2^6 + 0.2^7 + 0.2^8$$

Bu, tablo 5.1 de gösterildiği gibi, kural 45 kullanıldığında, merkez hücrenin $t+1$ zaman adımındaki durumunun, kendisinin, solundaki hücrenin, altındaki hücrenin ve sağ alt çaprazındaki hücrenin modülo 2 deki toplamı olduğunu gösterir.

64	128	256
32	1	2
16	8	4

Tablo 5. 1 Kural 45

Benzer şekilde kural 95 kullanıldığında merkez hücrenin $t+1$ zaman adımındaki durumu da aşağıdaki tabloda kırmızı ile yazılmış sayıların içinde bulunduğu hücrelerin durumlarının modülo 2 deki toplamıdır.

64	128	256
32	1	2
16	8	4

Tablo 5. 2 Kural 95

5×5 lik bir başlangıç konfigürasyonunu evirmek için 25×25 tipinde geçiş matrisi oluşturulur. 5×5 lik konfigürasyon 25×1 tipinde bir kolon vektörü biçimde yazılarak T geçiş matrisiyle çarpılır, çıkan 25×1 tipindeki kolon vektörü tekrar 5×5 lik bir latis şeklinde yazıldığında bir sonraki zamın adımının konfigürasyonu elde edilmiş olur.

Hibrit kural olarak, hücreleri sırasıyla kural 45, kural 95 ve sonra yine kural 45' e göre evirir ve böyle devam edersek T geçiş matrisinin ilk satırı kural 45'in ilk satırı, T'nin ikinci satırı kural 95'in ikinci satırı ve T'nin üçüncü satırı kural 45'in üçüncü satırı olmalı ve T matrisinin inşası bu şekilde devam etmelidir. O halde kural 45 ve kural 95'in hibrit geçiş matrisi şöyledir:

$$\begin{pmatrix}
1&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&1&1&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&1&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&0&0&0&1&1&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&1&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&1&1&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&1&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&1&0&0&0&1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&1&1&0&0&1&1&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&1&1&0&0&1&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&1&1&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&1&0&0&1&1&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&1&1&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1\\
1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&1&0&0\\
0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&1\\
1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1
\end{pmatrix}$$

Aldığımız rastgele 5×5 lik başlangıç konfigürasyonunun $t=0, t=1, \ldots, t=511$ zaman adımındaki durumlarını hesaplayıp, her bir konfigürasyonun onüçüncü, yani $0 \le t \le 511$ olmak üzere $c_{(3,1)}^{(t)}$ hücresini sabitleyerek elde ettiğimiz bit dizilerini yukarıda bahsedilen testlere tabi tuttuk. Yüz farklı başlangıç konfigürasyonundan elde ettiğimiz yüz adet dizinin 97 tanesi frekans testinden, 96 tanesi seri testinden, 95 tanesi poker testinden, 94 tanesi run testinden ve 97 tanesi de otokorelasyon testinden geçmeyi başardı. Burada bir tanesini örnek olarak verelim:

$$100111010001011010000 1011$$

başlangıç konfiürasyonu ile elde ettiğimiz 512-bit uzunluğundaki anahtar şöyledir:

$$\begin{aligned}
&11101101001000011000001100100100100011011101010011 0\\
&00110011110110001100011011101101111000101011110000 0\\
&11001101100110011110001011111101111111101111011000 1\\
&11011001000010000110101111010100010101000000111011 1\\
&11101101100010000010010001111001110101010001101000 0\\
&10111011100010000010000101011110110010011111101010 1\\
&01001101010000011010001111101111011110001001001100 0\\
&01010011000000110000111111000111111000101010001000 0\\
&11111101101010110101100101000001101100000111100111 11\\
&00111001111110011001010000011100000011010110011110 11
\end{aligned}$$

Bu bit dizisi için istatistiksel sonuçlar ise aşağıdaki tabloda verilmiştir:

Dizi	Adet
0	250
1	262
00	130
01	120
10	120
11	141
000	21
001	19
010	22
011	27
100	23
101	10
110	22
111	23
0000	6

Dizi	Adet
0001	10
0010	7
0011	8
0100	8
0101	8
0110	5
0111	12
1000	12
1001	8
1010	4
1011	9
1100	5
1101	12
1110	6
1111	8

Tablo 5. 3

Test sonuçları ise şöyledir:

- **i.** Frekans test için elde edilen değer $X_1 = 0,2812$.
- **ii.** Seri testi için elde edilen değer $X_2 = 2,0729$.
- **iii.** Poker testi için elde edilen değer $X_3 = 9,0117$.
- **iv.** Run testi için elde edilen değer $X_4 = 13,2565$.
- **v.** Otokorelasyon testi için elde edilen değer $X_5 = 0,5345$.

$\alpha = 0,05$ güven aralığındaki eşik değerleri ise X_1 için $3,8415$, X_2 için $5,9915$, X_3 için $14,0671$, X_4 için $15,5073$ ve X_5 için $1,96$ dır (Tablo 4.2 ve Tablo 4.3). O halde elde ettiğimiz dizi tüm testlerden geçmeyi başarmıştır.

Yapılan bir başka çalışmada da hibrit kuadratik (quadratic) hücresel dönüşümler kullanılarak $\mathbb{Z}_3$ üzerinde sözde rastgele dizi elde edilmeye çalışılmıştır.

$$C^0 = [012021012000102012021012000102012021010200102112000102020000102011121200100100112012000102001000102212210102011000110012000102222]$$

başlangıç konfigürasyonu ile elde edilen 567-bit uzunluğundaki dizi $\mathbb{Z}_3$ cismine uyarladığımız testlere tabi tutulmuş ve şu sonuçlar elde edilmiştir.

Dizi	Adet	Dizi	Adet	Dizi	Adet
0	196	001	5	112	7
1	185	002	5	120	7
2	186	010	13	121	9
00	69	011	8	122	6
01	64	012	9	200	6
02	62	020	7	201	6
10	70	021	1	202	5
11	55	022	15	210	8
12	60	100	9	211	4
20	57	101	9	212	3
21	65	102	6	220	4
22	64	110	7	221	12
000	7	111	4	222	7

Tablo 5. 4

Test	Serbestlik Derecesi	χ^2-değeri
Frekans	2	0,391
Seri	6	3,19
Poker	26	34
Run	10	7,93

Tablo 5. 5

Ki-kare tablosu kontrol edildiğinde bu değerlerin eşik değerlerini aşmadığı görülür.

5.2 Üretilen Anahtarla Görüntü Şifreleme

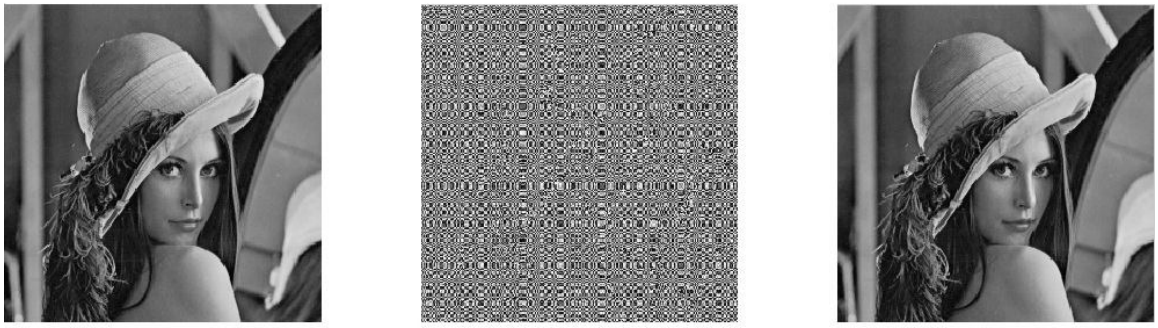
Bu bölümde basit bir görüntü şifreleme algoritması öne sürüp bu algoritmayı kullanarak anahtarımızın şifrelemede nasıl bir rol izlediğini test edeceğiz.

Bir fotoğraf aslında temel olarak bir matristir ve bu matrisin her bir bileşeni belirli sayıda bitin meydana getirdiği piksellerden *(pixel, picture element)* oluşur. Örneğin siyah beyaz bir resimde her bir piksel için 0 (ikilik olarak 00000000) ile 255 (ikilik olarak 11111111) arasında 256 tane renk tonu belirlidir. O halde 256×256 boyutlarındaki siyah beyaz bir resim temel olarak 256×256 tipinde ve her bir bileşeni 1 byte (8 bit) büyüklüğündeki pikseller meydana gelen bir matristir.

Görüntü şifreleme ile metin şifreleme arasında önemli farklılıklar vardır. Çünkü bir resimde aynı ya da çok yakın renklerin peşpeşe gelmesi ya da bir alana birikmesi oldukça doğal iken metin şifrelemede bu söz konusu değildir. Biz de bu yüzden şu şekilde bir şifreleme algoritması öne sürdük:

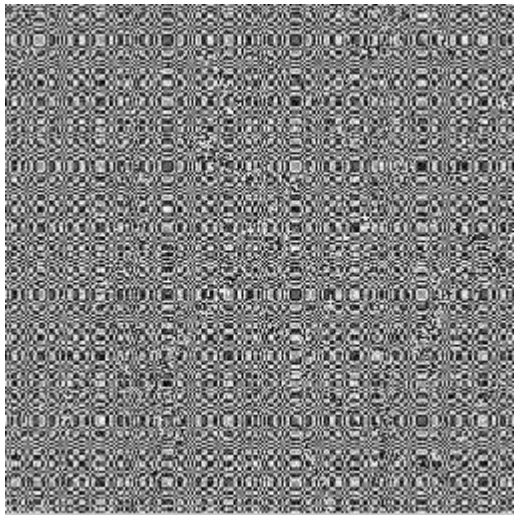
Resmin matrisini elde ettikten sonra üretilen anahtarla ilk satırdan başlanarak soldan sağa doğru sırayla her bir bit, anahtarın karşılık gelen biti ile modülo 2 de toplanır. Bu şekilde yeni resim yani matris elde edildikten sonra bu kez ilk sütundan başlanarak yukarıdan aşağıya sırayla her pikselin ikilik sistemdeki değeri, anahtarın karşılık gelen

byte değeri ile modülo 2 de toplanır ve şifrelenmiş görüntü elde edilir. Aşağıdaki örnekte, Bölüm 5. 1' de elde ettiğimiz anahtarı kullanarak gerçekleştirilen şifreleme ve deşifreleme işlemleri sonrasında oluşan resimler verilmiştir:



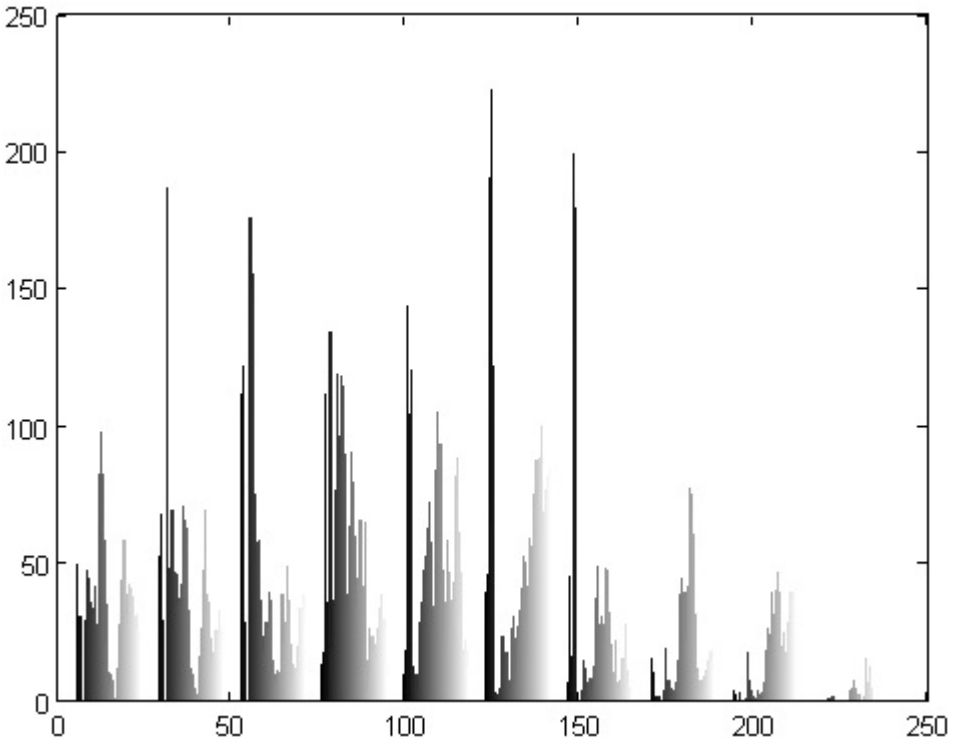
Resim 5. 1 Lena; orijinal resim, şifrelenmiş resim ve deşifrelenmiş resim

Resim 5. 2 Lena

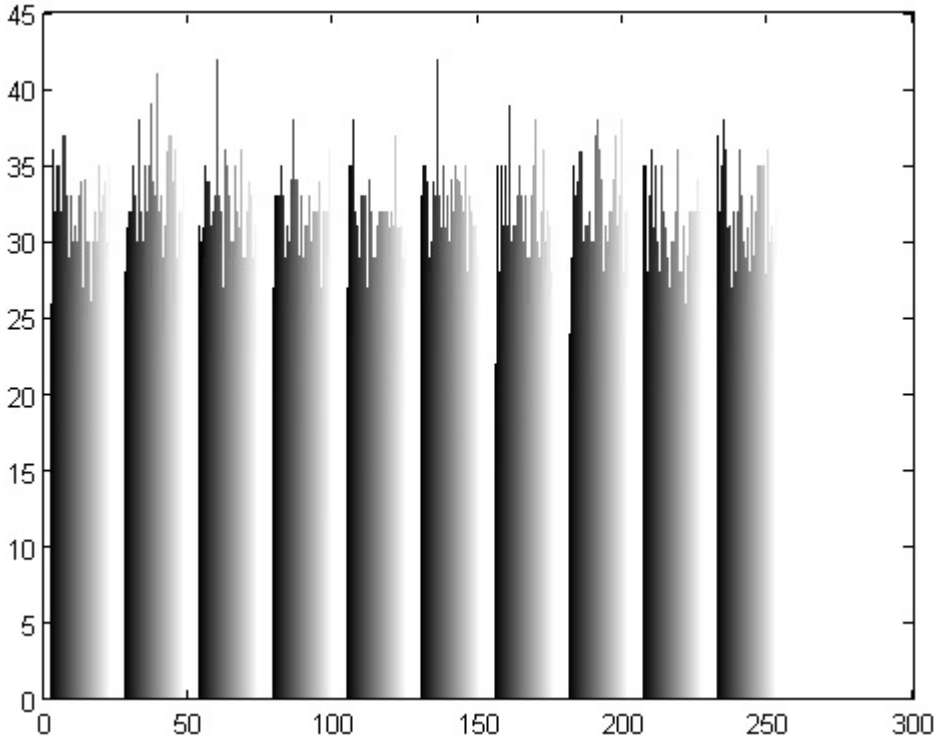


Resim 5. 3 Şifrelenmiş Lena resmi

Orijinal resim ve şifrelenmiş resmin histogramları ise aşağıdaki gibidir:



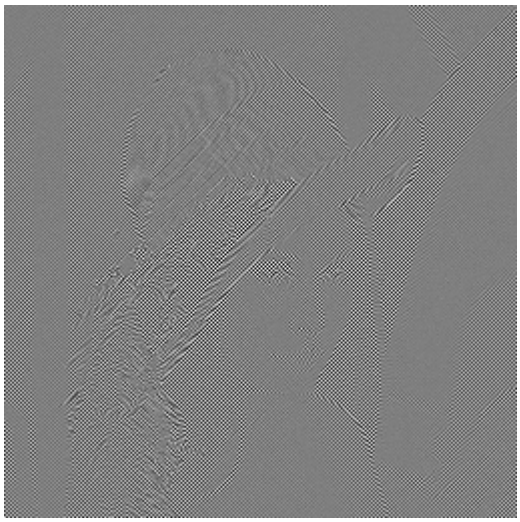
Şekil 5. 2 Orijinal resmin histogramı



Şekil 5. 3 Şifrelenmiş resmin histogramı

Görüldüğü gibi orijinal resimde renklerin dağılımı düzgün değilken şifrelendikten sonra renkler yaklaşık olarak düzgün dağılmıştır. Bu ise etkili bir görüntülü şifrelemede gerekli olan bir koşuldur [16].

Aynı şifreleme algoritması ile bu kez zayıf anahtarlarla yaptığımız şifreleme işlemi sonucunda oluşan iki resim ise aşağıda verilmiştir. Kullanılan anahtarlar Örnek 4. 3'te elde edilen bit dizisi ve "010101101010…" dizisidir.



Resim 5. 4

Resim 5. 5

BÖLÜM 6

SONUÇ ve ÖNERİLER

Bu tezde şifrelemenin temel ilkeleri, şifreleme için gerekli anahtar üretimi ve anahtarların güvenli sayılabilmesi için gerekli olan kriterlerden bahsedilmiş, hücresel dönüşümlerin tanımı yapılarak iki boyutlu lineer hibrit hücresel dönüşümler yardımıyla güvenli anahtarlar elde edilmiştir. Ayrıca ikilik sistemde verilen bazı istatistiksel testlerin özellikleri incelenerek üçlük sistemde kullanılmak üzere bazı sonuçlar elde edilmiştir.

KAYNAKLAR

[1] Akyıldız, E., (2004). “Kriptolojiye Giriş Ders Notları”, ODTÜ, Ankara.

[2] Rubio, C. F., Encinas, L. H., White, S. H., Rey, A. M. ve Sanchez, G. R., (2004). “The Use of Linear Hybrid Cellular Automata as Pseudorandom Bit Generators in Cryptography”, Neural, Parallel & Scientific Computations, 12: 175–192.

[3] Stinson, D. R., (2005). “Cryptography: Theory and Practice”, Chapman and Hall/CRC Press, Ontario.

[4] Menezes, A. J., Oorschot P. C. ve Vanstone S. A., (1996), “Handbook of Applied Cryptography”, CRC Press.

[5] Kahn, D., (1996). “The Codebreakers”, Scribner Press.

[6] Çimen, C., Akleylek S. ve Akyıldız E., (2009). “Şifrelerin Matematiği: Kriptografi”, ODTÜ Yayıncılık, Ankara.

[7] Fraleigh, J. B., (2003). “A First Course in Abstract Algebra”, Pearson Education.

[8] Schiff, J. L., (2008). “Cellular Automata: A Discrete View of the World”, Wiley &Sons, Inc. Hoboken, New Jersey.

[9] Ilachiski, A., (2001). “Cellular Automata, *A Discrete Universe”*, First Edition, World Scientific, New York.

[10] Chaudhuri, P.P., Chowdhury, D.R., Nandi S. ve Chattopadhyay S., (1997). “Additive Cellular Automata: Theory and Applications”, IEEE Press, New York.

[11] Ulam, S. M., (1950). “Random Processes and Transformations”, Proc. International Congress of Mathematicians, Cambridge MA. 2: 264-275.

[12] Codd, E. F., (1968). “Cellular Automata”, Academic Press, New York.

[13] Hedlund, G. A., (1969). “Endomorphisms and Automorphisms of The Shift Dynamical System”, Math. Syst. Theory 3: 320–375.

[14] Panda, S.P., Sahu, M. ve Swain, M. K., (2010). “Applications of Two Dimensional Cellular Automata rules for Block Cipher”, Special Issue of IJCCT, 2: 166-172.

[15] Khan, A. R., (2010). “On Two Dimensional Cellular Automata and its VLSI Applications”, IJECS-IJENS, 10: 124-129.

[16] Abdullah, A. H., Enayatifar, R. ve Lee, M., (2012). “A hybrid genetic algorithm and chaotic function model for image encryption”, International Journal of Electronics and Communications, 66: 806-816.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı **:** Fatih Temiz

Doğum Tarihi ve Yeri **:** 10/04/1986, Erbaa

Yabancı Dili: : İngilizce

E-posta: **:** ftemiz@yildiz.edu.tr

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Lisans	Matematik (İng)	Dokuz Eylül Üniversitesi	2010
Lise	Fen Bilimleri	Yılmaz Kayalar Anadolu Lisesi	2004

İŞ TECRÜBESİ

Yıl	**Firma/Kurum**	**Görevi**
2011- …	Yıldız Teknik Üniversitesi	Araştırma Görevlisi