

**AKIŞ ŞİFRELEME ALGORİTMALARININ
KARŞILAŞTIRMALI OLARAK İNCELENMESİ**

**A COMPARATIVE INVESTIGATION OF
STREAM CIPHER ALGORITHMS**

BÜKE ÇAKIR

Hacettepe Üniversitesi
Lisansüstü Eğitim – Öğretim ve Sınav Yönetmeliğinin
ELEKTRİK ve ELEKTRONİK Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.

2012

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma jürimiz tarafından **ELEKTRİK ve ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI 'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan :
Yrd. Doç. Dr. Semih BİNGÖL

Üye (Danışman) :
Yrd. Doç. Dr. Yakup ÖZKAZANÇ

Üye :
Yrd. Doç. Dr. Umut SEZEN

Üye :
Yrd. Doç. Dr. Mehmet DEMİRER

Üye :
Yrd. Doç. Dr. Harun ARTUNER

ONAY

Bu tez Hacettepe Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği'nin ilgili maddeleri uyarınca yukarıdaki jüri üyeleri tarafından/...../..... tarihinde uygun görülmüş ve Enstitü Yönetim Kurulunca/...../..... tarihinde kabul edilmiştir.

Prof. Dr. Fatma Sevin DÜZ
Fen Bilimleri Enstitüsü Müdürü

AKIŞ ŞİFRELEME ALGORİTMALARININ KARŞILAŞTIRMALI OLARAK İNCELENMESİ

Büke ÇAKIR

ÖZ

Akış şifreleme yöntemi, simetrik anahtar şifreleme yöntemlerinden olup bilgi güvenliğinin önemli olduğu alanlarda kullanılan bir şifreleme yöntemidir. Özellikle askeri ve haberleşme alanlarında güvensiz bir kanaldan, kritik bilginin güvenli bir şekilde iletilmesini sağlamak amacıyla geliştirilmiştir. LFSR ve PRNG tabanlı algoritmalar ile mobil cihaz ve baz istasyonu arasındaki veri iletişimi, anahtar dizisi üretme, Monte Carlo benzetimleri, internet trafiğini koruma (SSL/TLS), kablosuz ağ güvenliğini sağlama (WEP) gibi alanlarda akış şifreleme yöntemi kullanılmaktadır.

Bu tez çalışması kapsamında, akış şifreleme yöntemleri karşılaştırmalı olarak incelenmiş, yaygın olarak kullanılan LFSR ve PRNG tabanlı algoritmaların çalışma prensipleri araştırılmıştır. Akış şifreleme algoritmalarından; LFSR tabanlı A5/1, A5/2, RC4 algoritmaları ve PRNG tabanlı Blum Blum Shub, LCG, ICG, Mersenne Twister algoritmaları MATLAB ortamında programlanmış ve uygulanmıştır. Bu algoritmaların çıktıları; Frekans, Koşum, Spektral (Ayrık Fourier Dönüşümü) ve Ki-Kare testi gibi istatistiksel yöntemlerle test edilmiştir. LFSR ve PRNG tabanlı akış şifreleme algoritmaları tarafından üretilen dizilerin istatistiksel karakteristikleri gözlemlenmiş ve algoritmalarda kullanılan parametrelerin test sonuçlarına etkileri değerlendirilmiştir. Test sonuçları incelenerek, akış şifreleme algoritmalarının uygulanması sırasında dikkat edilmesi gereken hususlar vurgulanmıştır.

Anahtar Kelimeler: Akış Şifreleme, Anahtar dizisi, LFSR, PRNG, A5/1, A5/2, RC4, Blum Blum Shub, LCG, ICG, Mersenne Twister, Frekans Testleri, Koşum Testleri, Spektral Testleri, Ki-Kare Testi.

Danışman: Yrd. Doç. Dr. Yakup ÖZKAZANÇ, Hacettepe Üniversitesi, Elektrik ve Elektronik Mühendisliği Bölümü

A COMPARATIVE INVESTIGATION OF STREAM CIPHER ALGORITHMS

Büke ÇAKIR

ABSTRACT

Stream ciphering is a cryptographic method that is used in areas where information security is crucial. It is specifically developed for transmitting critical information in military and telecommunication applications where insecure communication channels are used. Stream ciphering is used in applications as LFSR and PRNG based algorithms, data encryption between mobile and base station communications, keystream generation, Monte Carlo simulation, internet security (SSL/TLS) and wireless network security (WEP).

In this thesis, stream ciphering methods are comparatively investigated. LFSR and PRNG based algorithms that are used with stream ciphering methods are analyzed. LFSR based algorithms such as A5/1, A5/2, RC4 and PRNG based algorithms such as Blum Blum Shub, LCG, ICG, Mersenne Twister are programmed and implemented in MATLAB. The outputs of the algorithms implemented are tested via statistical methods such as Frequency Test, Runs Test, Spectral Test (Discrete Fourier Transform) and Chi Square Goodness of Fit Test. Statistical characteristics of the outputs of the algorithms are observed. The algorithm specific parameters that specify the nature of the random output are tailored and the corresponding changes in test results are evaluated. Key aspects of stream ciphering algorithms are emphasized by utilizing the test results.

Keywords : Stream Cipher, Keystream, LFSR, PRNG, A5/1, A5/2, RC4, Blum Blum Shub, LCG, ICG, Mersenne Twister, Frequency Test, Runs Test, Spectral Test, Chi-Square Test.

Advisor : Asst. Prof. Yakup ÖZKAZANÇ, Hacettepe University, Department of Electrical and Electronics Engineering.

TEŞEKKÜR

Tez çalışmam süresince bilgisini, tecrübelerini ve yardımını esirgemeyen değerli tez danışmanım Sayın Yrd. Doç. Dr. Yakup Özkazanç'a çok teşekkür ederim.

Bana hayatın her alanında daima destek olan canım annem Dilek ve babam Raif Çakır ile anneannem Ayten Ladik'e sonsuz teşekkürü bir borç bilirim.

Berrak ve Ozan başta olmak üzere tez çalışmam boyunca gösterdikleri anlayış ve desteklerinden dolayı tüm arkadaşlarıma teşekkür ederim.

İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
ÖZ	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER DİZİNİ	iv
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER DİZİNİ	ix
SİMGELER VE KISALTMALAR DİZİNİ	xi
SÖZLÜKÇE	xvi
1. GİRİŞ	1
1.1 Akış Şifreleme	2
1.2 Tez Raporunun Yapısı	5
2 RASTSALLIK TESTLERİ	6
2.1 Rastsallık	6
2.2 Tahmin Edilemezlik	6
2.3 Test Etme	7
2.4 Hipotezler	7
2.4.1 Sıfır Hipotezi (H_0)	7
2.4.2 Karşıt Hipotez (H_1/H_a)	7
2.5 Testler	9
2.5.1 Uygulanan Testler	11
2.5.1.1 Frekans Testi	11
2.5.1.2 Koşum Testi	12
2.5.1.3 Ayırık Fourier Dönüşümü (Spektral) Testi	14
2.5.1.4 Ki-Kare Testi	15
2.5.1.5 Histogram	16
2.5.2 Test Uygulama Stratejileri ve Sonuçların Değerlendirilmesi	16
3 DOĞRUSAL GERİ BESLEMELİ KAYAN YAZMAÇ TABANLI YÖNTEMLER 21	
3.1 A5/1 Algoritması	23
3.1.1 A5 Algoritmasının Zayıf Yönleri	26
3.1.1.1 A5 Şifreleme Algoritmalarına Karşı Kaba Kuvvet Saldırısı	27
3.1.1.2 A5 Şifreleme Algoritmalarına Karşı Böl ve Fethet Saldırısı	27
3.1.2 A5/1 Algoritmasının Uygulanması	28
3.1.3 Uygulanan Testlerin Sonuçları ve Değerlendirmeler	31
3.1.3.1 Strateji-1 Testleri	32
3.1.3.2 Strateji-2 Testleri	33
3.2 A5/2 Algoritması	36
3.2.1 A5 Algoritmasının Zayıf Yönleri	38
3.2.2 A5/2 Algoritmasının Uygulanması	39
3.2.3 Uygulanan Testlerin Sonuçları ve Değerlendirmeler	41
3.2.3.1 Strateji-1 Testleri	42
3.2.3.2 Strateji-2 Testleri	43
3.3 RC4 ALGORİTMASI	47
3.3.1 RC4 Algoritmasının Uygulanması	50

3.3.2	Uygulanan Testlerin Sonuçları ve Değerlendirmeler	52
3.3.2.1	Strateji-1 Testleri	53
3.3.2.2	Strateji-2 Testleri	54
3.4	LFSR Algoritmalarının Karşılaştırılması	57
4	SÖZDE RASTSAL SAYI DİZİSİ TABANLI YÖNTEMLER	59
4.1	BLUM BLUM SHUB ALGORİTMASI	60
4.1.1	BBS Algoritmasının Uygulanması	65
4.1.2	Uygulanan Testlerin Sonuçları ve Değerlendirmeler	67
4.1.2.1	Strateji-1 Testleri	68
4.1.2.2	Strateji-2 Testleri	72
4.2	Doğrusal Eşleniksel Üreteç (LCG)	82
4.2.1	Çarpımsal Doğrusal Eşleniksel Üreteç (MLCG)	83
4.2.2	Karışık Eşleniksel Üreteç (MCG)	84
4.2.3	LCG Algoritmasının Uygulanması	86
4.2.4	Uygulanan Testlerin Sonuçları ve Değerlendirmeler	87
4.2.4.1	Strateji-1 Testleri	88
4.2.4.2	Strateji-2 Testleri	91
4.3	Ters Eşleniksel Üreteç (ICG)	99
4.3.1	ICG Algoritmasının Uygulanması	101
4.3.2	Uygulanan Testlerin Sonuçları ve Değerlendirmeler	101
4.3.2.1	Strateji-1 Testleri	103
4.3.2.2	Strateji-2 Testleri	105
4.4	Mersenne Twister (MT)	112
4.4.1	MT Algoritmasının Uygulanması	116
4.4.2	Uygulanan Testlerin Sonuçları ve Değerlendirmeler	117
4.4.2.1	Strateji-1 Testleri	118
4.4.2.2	Strateji-2 Testleri	120
4.5	PRNG Algoritmalarının Karşılaştırılması	126
5	SONUÇ	128
	KAYNAKLAR DİZİNİ	131
	ÖZGEÇMİŞ	138

ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
Şekil 1.1 Şifreleme Yöntemleri ve Türleri	2
Şekil 1.2 Eşzamanlı Akış Şifrelemede Şifreleme ve Şifre Çözme İşlemleri	3
Şekil 1.3 Eşzamansız Akış Şifrelemede Şifreleme ve Şifre Çözme İşlemleri	3
Şekil 1.4 Eşzamanlama Akış Şifrelemede Eşzamanlama Yapılması	4
Şekil 2.1 Ret Bölgeleri	9
Şekil 2.2 Testlerin Uygulanması	17
Şekil 2.3 Örnek Bir P-değeri Histogram Grafiği	20
Şekil 3.1 Doğrusal Olmayan Filtre ile Doğrusal Karmaşıklık Artırma.....	22
Şekil 3.2 Doğrusal Olmayan Fonksiyon ile Doğrusal Karmaşıklık Artırma	22
Şekil 3.3 A5/1 Algoritmasının İç Yapısı	24
Şekil 3.4 A5/1 Algoritmasının Veri Akışı	30
Şekil 3.5 A5/1 Algoritması ile Düz Metnin Şifrenmesi	30
Şekil 3.6 A5/1 Algoritması Strateji-1 FFT Grafiği	33
Şekil 3.7 A5/1 Algoritması Strateji-2 Frekans Testi Histogramı	34
Şekil 3.8 A5/1 Algoritması Strateji-2 Koşum Testi Histogramı	35
Şekil 3.9 A5/1 Algoritması Strateji-2 Spektral Testi Histogramı	35
Şekil 3.10 A5/2 Algoritmasının İç Yapısı	37
Şekil 3.11 A5/2 Algoritması Strateji-1 FFT Grafiği	43
Şekil 3.12 A5/2 Algoritması Strateji-2 Frekans Testi Histogramı	44
Şekil 3.13 A5/2 Algoritması Strateji-2 Koşum Testi Histogramı	45
Şekil 3.14 A5/2 Algoritması Strateji-2 Spektral Testi Histogramı	45
Şekil 3.15 RC4 Algoritmasının WEP Uygulaması.....	48
Şekil 3.16 RC4 Algoritması ile Şifreleme	48
Şekil 3.17 RC4 Algoritması S Dizisi.....	49
Şekil 3.18 RC4 Akış Diagramı	51
Şekil 3.19 Sırasıyla 3 Üretecın FFT Grafiği	54
Şekil 3.20 RC4 Algoritması Strateji-2 Frekans Testi Histogramı	55
Şekil 3.21 RC4 Algoritması Strateji-2 Koşum Testi Histogramı	56
Şekil 3.22 RC4 Algoritması Strateji-2 Spektral Testi Histogramı	56
Şekil 4.1 BBS Algoritmasının Fiziksel Yapısı.....	61
Şekil 4.2 BBS Algoritması Strateji-1 CDF Grafikleri.....	70

Şekil 4.3 BBS Algoritması Strateji-1 Histogramı	71
Şekil 4.4 BBS Algoritması Strateji-1 FFT Grafikleri (Eşlik Biti Çıktı Yöntemi)	71
Şekil 4.5 BBS Algoritması Strateji-1 FFT Grafikleri	72
Şekil 4.6 BBS Algoritması Strateji-2, Üreteç-1 48. , 115. ve 193. Örneklemeleri....	73
Şekil 4.7 BBS Algoritması Strateji-2, Üreteç-1 48. , 115. ve 193. Örneklemeleri....	74
Şekil 4.8 BBS Algoritması Strateji-2, Üreteç-2 48. , 115. ve 193. Örneklemeleri....	74
Şekil 4.9 BBS Algoritması Strateji-2, Üreteç-2 48. , 115. ve 193. Örneklemeleri....	74
Şekil 4.10 BBS Algoritması Strateji-2, Üreteç-3 48. , 115. ve 193. Örneklemeleri..	75
Şekil 4.11 BBS Algoritması Strateji-2, Üreteç-3 48. , 115. ve 193. Örneklemeleri..	75
Şekil 4.12 BBS Algoritması Strateji-2 Frekans Testi Histogramı	76
Şekil 4.13 BBS Algoritması Strateji-2 Frekans Testi Histogramı	76
Şekil 4.14 BBS Algoritması Strateji-2 Koşum Testi Histogramı	77
Şekil 4.15 BBS Algoritması Strateji-2 Koşum Testi Histogramı	77
Şekil 4.16 BBS Algoritması Strateji-2 Spektral Testi Histogramı	78
Şekil 4.17 BBS Algoritması Strateji-2 Spektral Testi Histogramı	79
Şekil 4.18 BBS Algoritması Strateji-2 Ki-Kare Testi Histogramı	80
Şekil 4.19 LCG Fiziksel Gösterimi	84
Şekil 4.20 LCG Algoritması Strateji-1 CDF Grafikleri	90
Şekil 4.21 LCG Algoritması Strateji-1 Histogramı	91
Şekil 4.22 LCG Algoritması Strateji-1 FFT Grafikleri	91
Şekil 4.23 LCG Algoritması Strateji-2, Üreteç-1 48. , 115. ve 193. Örneklemeleri..	92
Şekil 4.24 LCG Algoritması Strateji-2, Üreteç-2 48. , 115. ve 193. Örneklemeleri..	93
Şekil 4.25 LCG Algoritması Strateji-2, Üreteç-3 48. , 115. ve 193. Örneklemeleri..	93
Şekil 4.26 LCG Algoritması Strateji-2 Frekans Testi Histogramı	94
Şekil 4.27 LCG Algoritması Strateji-2 Koşum Testi Histogramı	95
Şekil 4.28 LCG Algoritması Strateji-2 Spektral Testi Histogramı	96
Şekil 4.29 LCG Algoritması Strateji-2 Ki-Kare Testi Histogramı	97
Şekil 4.30 ICG Algoritması Strateji-1 CDF Grafikleri	104
Şekil 4.31 ICG Algoritması Strateji-1 Histogramı	105
Şekil 4.32 ICG Algoritması Strateji-1 FFT Grafikleri	105
Şekil 4.33 ICG Algoritması Strateji-2, Üreteç-1 48. , 115. ve 193. Örneklemeleri.	106
Şekil 4.34 ICG Algoritması Strateji-2, Üreteç-2 48. , 115. ve 193. Örneklemeleri.	107
Şekil 4.35 ICG Algoritması Strateji-2, Üreteç-3 48. , 115. ve 193. Örneklemeleri.	107
Şekil 4.36 ICG Algoritması Strateji-2 Frekans Testi Histogramı	108

Şekil 4.37 ICG Algoritması Strateji-2 Koşum Testi Histogramı	109
Şekil 4.38 ICG Algoritması Strateji-2 Spektral Testi Histogramı	110
Şekil 4.39 ICG Algoritması Strateji-2 Ki-Kare Testi Histogramı	110
Şekil 4.40 MT Algoritması Strateji-1 CDF Grafiği	119
Şekil 4.41 MT Algoritması Strateji-1 Histogramı	120
Şekil 4.42 MT Algoritması Strateji-1 FFT Grafiği	120
Şekil 4.43 MT Algoritması Strateji-2, Üretecın 48. , 115. ve 193. Örneklemeleri ..	121
Şekil 4.44 MT Algoritması Strateji-2 Frekans Testi Histogramı	122
Şekil 4.45 MT Algoritması Strateji-2 Koşum Testi Histogramı	123
Şekil 4.46 MT Algoritması Strateji-2 Spektral Testi Histogramı	124
Şekil 4.47 MT Algoritması Strateji-2 Ki-Kare Testi Histogramı	125

ÇİZELGELER DİZİNİ

	<u>Sayfa</u>
Çizelge 2.1 Gerçek Durum ve Karar Çizelgesi	8
Çizelge 3.1 Maksimum Periyodu Sağlayan Polinom Seçimi Örneği.....	21
Çizelge 3.2 A5/1 Algoritmasının Matematiksel Yapısı	25
Çizelge 3.3 A5 Algoritmasına Yapılan Saldırıları.....	26
Çizelge 3.4 A5/1 Algoritması Strateji-1 Test Sonuçları-1.....	32
Çizelge 3.5 A5/1 Algoritması Strateji-2 Test Sonuçları-1.....	33
Çizelge 3.6 A5/1 Algoritması Strateji-2 Test Sonuçları-2.....	34
Çizelge 3.7 A5/2 Algoritmasının Matematiksel Yapısı	38
Çizelge 3.8 A5/2 Algoritması Strateji-1 Test Sonuçları-1.....	42
Çizelge 3.9 A5/2 Algoritması Strateji-2 Test Sonuçları-1.....	43
Çizelge 3.10 A5/2 Algoritması Strateji-2 Test Sonuçları-2.....	44
Çizelge 3.11 RC4 Algoritması Strateji-1 Test Sonuçları-1	53
Çizelge 3.12 RC4 Algoritması Strateji-2 Test Sonuçları-1	54
Çizelge 3.13 RC4 Algoritması Strateji-2 Test Sonuçları-2	55
Çizelge 3.14 LFSR Algoritmalarının Karşılaştırılması.....	57
Çizelge 4.1 BBS Algoritması Strateji-1 Test Sonuçları-1.....	69
Çizelge 4.2 BBS Algoritması Strateji-1 Eşlik Biti Çıktı Yöntemi Test Sonuçları-2 .	69
Çizelge 4.3 BBS Algoritması Strateji-1 Test Sonuçları-2	69
Çizelge 4.4 BBS Algoritması Strateji-2 Eşlik Biti Çıktı Yöntemi Test Sonuçları-1 .	72
Çizelge 4.5 BBS Algoritması Strateji-2 Test Sonuçları-1	73
Çizelge 4.6 BBS Algoritması Strateji-2 Frekans Testi	75
Çizelge 4.7 BBS Algoritması Strateji-2 Koşum Testi	76
Çizelge 4.8 BBS Algoritması Strateji-2 Spektral Testi	78
Çizelge 4.9 BBS Algoritması Strateji-2 Ki-Kare Testi	79
Çizelge 4.10 BBS Algoritması Strateji-2 Uygulanan Testlerin Sonuçları	80
Çizelge 4.11 BBS Algoritması Strateji-2 Uygulanan Testlerin Sonuçları	80
Çizelge 4.12 LCG Algoritması Strateji-1 Test Sonuçları-1.....	89
Çizelge 4.13 LCG Algoritması Strateji-1 Test Sonuçları-2.....	89
Çizelge 4.14 LCG Algoritması Strateji-2 Test Sonuçları-1.....	92
Çizelge 4.15 LCG Algoritması Strateji-2 Frekans Testi	94
Çizelge 4.16 LCG Algoritması Strateji-2 Koşum Testi	95

Çizelge 4.17 LCG Algoritması Strateji-2 Spektral Testi	95
Çizelge 4.18 LCG Algoritması Strateji-2 Ki-Kare Testi	96
Çizelge 4.19 LCG Algoritması Strateji-2 Uygulanan Testlerin Sonuçları	97
Çizelge 4.20 ICG Algoritması Strateji-1 Test Sonuçları-1.....	103
Çizelge 4.21 ICG Algoritması Strateji-1 Test Sonuçları-2.....	103
Çizelge 4.22 ICG Algoritması Strateji-2 Test Sonuçları-1.....	106
Çizelge 4.23 ICG Algoritması Strateji-2 Frekans Testi	108
Çizelge 4.24 ICG Algoritması Strateji-2 Koşum Testi	108
Çizelge 4.25 ICG Algoritması Strateji-2 Spektral Testi	109
Çizelge 4.26 ICG Algoritması Strateji-2 Ki-Kare Testi	110
Çizelge 4.27 ICG Algoritması Strateji-2 Test Sonuçları.....	111
Çizelge 4.28 MT Algoritması Strateji-1 Test Sonuçları-1	118
Çizelge 4.29 MT Algoritması Strateji-2 Test Sonuçları-1	121
Çizelge 4.30 MT Algoritması Strateji-2 Frekans Testi	122
Çizelge 4.31 MT Algoritması Strateji-2 Koşum Testi	123
Çizelge 4.32 MT Algoritması Strateji-2 Spektral Testi	123
Çizelge 4.33 MT Algoritması Strateji-2 Ki-Kare Testi	124
Çizelge 4.34 MT Algoritması Strateji-2 Test Sonuçları	125
Çizelge 4.35 PRNG Algoritmalarının Karşılaştırılması	126

SİMGELER VE KISALTMALAR DİZİNİ

Simgeler

α	: Anlam Düzeyi, Yanılgı Payı, I. Tür Hata
a	: LCG Algoritmasında Çarpan
β	: II. Tür Hata
c	: LCG Algoritmasında Artan
d	: Düzgelenmiş %95 Eşiğinin Dışı
e	: Euler Ölçütü
erfc	: Hata Fonksiyonu
$\mathbb{F}_2$	: İkili Alan
F_i	: i. Altaralıktaki P-değerlerinin Sayısı
F_n	: Çerçeve Numarası
FI	: KASUMI Fonksiyonu
FL	: KASUMI Fonksiyonu
FO	: KASUMI Fonksiyonu
gcd	: En Büyük Ortak Bölen (EBOB)
H_0	: Sıfır Hipotezi
H_1	: Karşıt Hipotez
H_a	: Karşıt Hipotez
i	: Altaralık
K	: Anahtar
K_c	: Oturum Anahtarı
KI	: Alt Anahtar
KL	: Alt Anahtar
KO	: Alt Anahtar

LC	: Doğrusal Karmaşıklık
lcm	: En Küçül Ortak Kat (EKOK)
M	: Blum Tamsayısı
M_p	: Mersenne Asalı
m	: Örneklem Sayısı
μ	: Ortalama
N_0	: Beklenen Tepe Sayısı
N_1	: Gözlenen Tepe Sayısı
n	: Dizi Uzunluğu
σ^2	: Varyans
ϵ	: Bit Dizisi
o_i	: Gözlenen Frekans
e_i	: Beklenen Frekans
π	: BBS Algoritmasında Periyot
π	: Bit Dizisindeki 1'lerin Sayısı
λ	: Carmichael Fonksiyonu
$\hat{p}$	: Doğru Karar Çıkarımı (1- α)
p	: Asal sayı
q	: Asal sayı
R1	: 1. Yazmaç
R2	: 2. Yazmaç
R3	: 3. Yazmaç
R4	: 4. Yazmaç
S_{obs}	: Gözlenen Değer
S_n	: Dizideki 0 ve 1'lerin Toplamı

s	: Örneklem Sayısı
S	: Permütasyon İşlemi
T	: Tepe Yüksekliği
TR	: Kırpma (Truncate)
x_0	: Başlangıç Değeri
x_i	: i . Sözde Rastsal Dizi Elemanı
x_i^{-1}	: $x_i \bmod p$ 'nin Çarpımsal Tersi
x_k^u	: x_k Vektörünün En Önemli $\omega - r$ Biti
x_{k+1}^l	: x_{k+1} Vektörünün En Önemsiz r Biti
χ^2	: Test İstatistiği
$V_n(\text{obs})$	: Koşumların Sayısı

Kısaltmalar

3 GPP	: 3rd Generation Partnership Project
AOCE	: Apple Open Collaboration Environment
BBS	: Blum Blum Shub
CDF	: Cumulative Distribution Function (Birikimli Dağılım Fonksiyonu)
CDMA	: Code Division Multiple Access (Kod Bölmeli Çoklu Erişim)
CSPRNG	: Cryptographically Secure Pseudo-Random Number Generator (Kriptografik olarak Güvenli Sözde Rastsal Sayı Dizisi)
EBOB	: En Büyük Ortak Bölen
EDGE	: Enhanced Data Rates for GSM Evolution
GPRS	: General Packet Radio Service
GPS	: Global Positioning System
GSMA	: GSM Association
HSCSD	: High Speed Circuit Switched Data
ICG	: Inverse Congruential Generator (Ters Eşleniksel Üretici)
IMP	: Inverse Maximal Period (Ters Maksimum Periyot)
IV	: Initial Vector (Başlangıç Vektörü)
KSA	: Key Scheduling Algorithm (Anahtar Çizelgesi Algoritması)
LCG	: Linear Congruential Generator (Doğrusal Eşleniksel Üretici)
LSB	: Least Significant Bit (En Önemsiz Bit)
LSFR	: Linear Feedback Shift Register (Doğrusal Geri Beslemeli Kayan Yazmaç)
MCG	:Mixed Linear Congruential Generator (Karışık Eşleniksel Üreteç)
MLCG	: Multiplicative Linear Congruential Generator (Çarpımsal Doğrusal Eşleniksel Üreteç)
MSB	: Most Significant Bit (En Önemli Bit)
NIST	: National Institute of Standards and Technology

OFB	: Output Feedback Mode (Çıktı Geribeslemeleli Mod)
OTP	: One Time Pad (Tek Kullanımlık Şifre)
PRGA	: Pseudo Random Number Algorithm (Sözde Rastsal Üreteç Algoritması)
PRNG	: Pseudo Random Number Generator (Sözde Rastsal Sayı Dizisi)
QR	: Quadratic Residue (Karesel Kalıntı)
RDC	: Remote Desktop Connection
RDP	: Remote Desktop Protocol
RNG	: Random Number Generator (Rastsal Sayı Üreteci)
SAGE	: Security Algorithms Group of Experts
SASL	: Simple Authentication and Security Layer
SSH	: Secure Shell
SSL	: Secure Sockets Layer
TGFSR	: Twisted Generalized Feedback Shift Register
TSL	: Transport Layer Security
UMTS	: Universal Mobile Telecommunications System
WELL	: Well Equidistributed Long-period Linear
WEP	: Wired Equivalent Privacy
WPA	: Wi-Fi Protected Access
XOR	: Exclusive OR

SÖZLÜKÇE

Altaralık	: Sub-interval
Akış Şifreleme	: Stream Cipher
Anahtar Çizelgesi Algoritması	: Key Scheduling Algorithm
Anahtar Dizisi	: Keystream
Asal Polinom	: Primitive Polynomial
Ayrık Fourier Dönüşümü	: Discrete Fourier Transform
Başlangıç Değeri	: Seed
Benzetim Dili	: Simulation Language
Birikimli Toplam Testi	: The Cumulative Sums (Cusums) Test
Birörnek Dağılım	: Uniform Distribution
Bloktaki En Uzun Birler Testi	: Tests for the Longest-Run-of-Ones in a Block
Böl ve Fethet Saldırısı	: Divide and Conquer Attacks
Çevrim	: Cycle
Çift Eşlik Biti	: Even Parity Bit
Çıktı Dizisi	: Output Stream
Çıktı Geri Beslemeli Mod	: Output Feedback Mode
Çoğulortam	: Multimedia
Dağılımsal	: Distributional
Derece	: Order
Doğrusal Karmaşıklık Testi	: Linear Complexity Test
Döngü	: Round
Düzgeleme	: Normalization
Eşdağılım	: Equidistributed
Gecikmeli Fibonacci Üreteç	: Lagged Fibonacci Generators

Geniş Anahtar Arama	: Exhaustive Key Search
Gizlice dinleme yöntemine	: Eavesdropping
Göreceli Asal	: Co-prime, Relatively Prime
Hesaba dayalı zamanın	: Computational Time
Hesaplama Karmaşıklığı	: Computational Complexity
İkili Alan	: Binary Field
İkili Matris Sırası Testi	: Binary Matrix Rank Test
İlinti	: Correlation
Kaba Kuvvet Saldırısı	: Brute Force Attack
Ki-Kare	: Chi-square
Koşum Testi	: Runs Test
Kriptografik olarak güvenli	: Cryptographically Secure
Kuyruk Olasılığı	: Tail Probability
Matrisli Doğrusal Tekrarlama	: Matrix Linear Recurrence
Maurer Evrensel İstatistiksel Testi	: Maurer's "Universal Statistical" Test
Modüler Çarpım Tersisi	: Modular Multiplicative Inverse
Ölçeklenebilirlik	: Scalability
Örneklem Sayısı	: Sample Size
Örüntü	: Pattern
Polinomsal Zaman	: Polynomial Time
Rastsal Ayrılma Testi	: The Random Excursions Test
Rastsal Değişimli Ayrılma Testi	: The Random Excursions Variant Test
Rastsallık	: Randomness
Rastsal Sayı Dizisi	: Random Number Generator
Seri Testi	: The Serial Test
Sözde Rastsal Dizi	: Pseudo-Random Sequence

Şablon ile Örtüşme Uyum Testi	: The Overlapping Template Matching Test
Şablon ile Örtüşmeyen Uyum Testi	: The Non-overlapping Template Matching Test
Tahmin Edilemezlik	: Unpredictability
Tekdüzelik	: Uniformity
Tek Eşlik Biti	: Odd Parity Bit
Tek Kullanımlık Şifre	: One Time Pad
Tutarlılık	: Consistency
Uyumun İyiliği	: Goodness of Fit
Veri dizisi	: Data Stream
Yaklaşık Entropi Testi	: The Approximate Entropy Test
Zaman-Hafıza Ödünleşimi	: Time-memory trade off

1. GİRİŞ

Haberleşme ve bilgi çağı olarak nitelendirilen günümüzde, bilginin paylaşımı ve bilgi güvenliğinin sağlanması bir gereklilik haline gelmiştir. Eski çağlardan beri, önem seviyesi yüksek bilgilerin, sadece alıcısı tarafından anlaşılabilirliğini sağlamak amacıyla çeşitli şifreleme yöntemleri kullanılmaktadır. Teknoloji ile birlikte, bilgi iletimini güvenli bir şekilde sağlamak için yeni şifreleme sistemleri ve yeni yöntemler geliştirilmektedir.

Akış şifreleme (stream cipher) algoritmaları, askeri ve haberleşme alanları, bilgi işleme uygulamaları gibi geniş bir yelpazede kullanılmaktadır. Kişisel ve kurumsal bilgilerin aktarımı sırasında, bilgilerin dışarıya karşı korunması için güvenli haberleşme altyapılarına ihtiyaç duyulmaktadır.

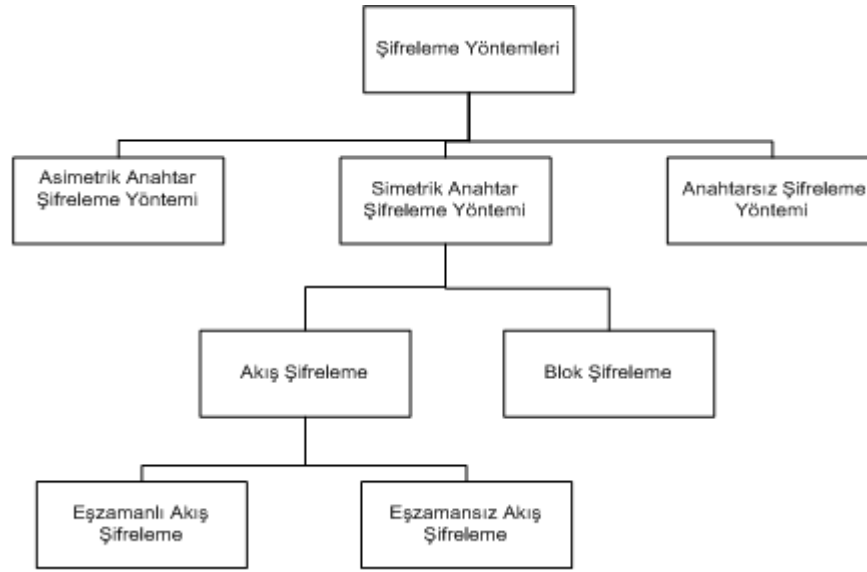
Akış şifrelemenin çıkış noktası Tek Kullanımlık Şifre (OTP: One-Time Pad) yöntemidir. OTP tek kullanımlık şifreleme yöntemi olup Vernam şifreleme yöntemi olarak da bilinmektedir. 1. Dünya Savaşı sırasında Gilbert Vernam tarafından akış şifreleme bulunmuş ve Joseph Mauborgne ile OTP şifreleme metodu geliştirilmiştir. Bu yapının güvenilirliği 1949 yılında Claude Shannon tarafından kanıtlanmıştır [Başkök, 2007].

OTP yönteminde, şifrelenecek mesajın uzunluğunda tam rastsal bir anahtar dizisi seçilir. Bu nedenle koşulsuz güvenli bir sistemdir. Ancak uzun bir anahtar üretmek, bu anahtarı güvenli bir şekilde karşı tarafa iletmek ve saklamak zordur. Bu nedenlerden dolayı yöntemin kullanım alanı kısıtlıdır. Bu yöntem yerine tercih edilen sözde rastsal diziler, kısa bir anahtar kullanılarak bir dizi üretici tarafından üretilir.

Bu tezin konusu akış şifreleme yöntemine ait farklı algoritmaların incelenmesi, gerçekleştirilmesi ve karşılaştırılmasıdır. Bu amaca yönelik olarak, yedi farklı algoritma incelenmiş ve karşılaştırılmıştır.

1.1 Akış Şifreleme

Şifreleme algoritmalarını temelde simetrik, asimetrik ve anahtarsız şifreleme (hash) algoritmaları olmak üzere üç gruba ayırabiliriz. Simetrik şifrelemede, şifreleme ve şifre çözme işlemleri aynı anahtar ile yapılır. Simetrik şifreleme algoritmaları, blok ve akış şifreleme algoritmaları olmak üzere iki gruba ayrılabilir [Turan, 2008].



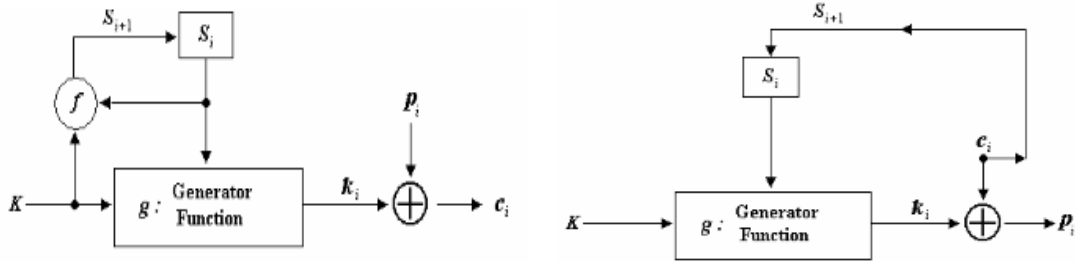
Şekil 1.1 Şifreleme Yöntemleri ve Türleri

Akış şifreleme algoritmalarında kullanılan rastsal dizi üreteçlerinin yapısı sayesinde, akış şifreleme blok şifrelemeye göre daha hızlıdır. Bu sayede akış şifrelemenin yazılım verimliliği daha üst seviyededir. Buna ek olarak donanıma uygulanabilirliğindeki karmaşıklık daha düşük seviyededir. Bu özelliklerine rağmen güvenlik açıklıkları oluşturabilecekleri nedeniyle, başlangıç değeri ve anahtar yalnızca bir kez kullanılmalıdır.

Akış şifrelemenin iç durumları ve seri işlemleri ile akış şifreleme çıktısı olan anahtar dizisi elde edilir. Anahtar dizisi, sözde rastsal dizi, düz metin ile bit-bit şifrelenerek (XOR işleminden geçirilerek) şifreli metin oluşturulur, şifreleme gerçekleştirilir. Şifreli metin ikinci bir kez XOR işleminden geçirilerek de düz metin elde edilir, şifre çözme gerçekleştirilir.

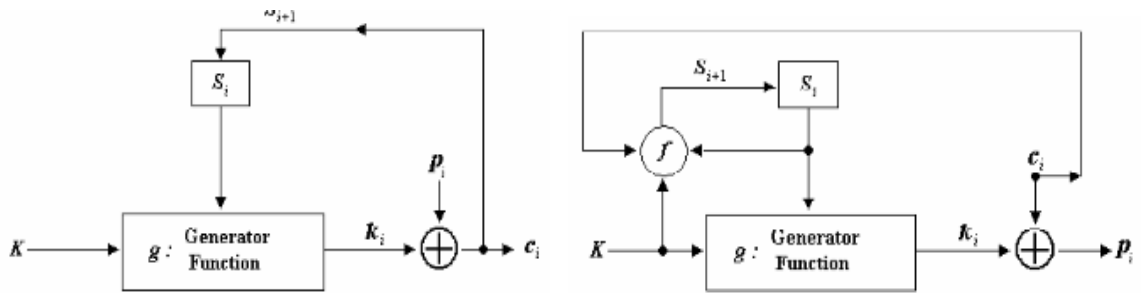
Bu şifreleme yönteminin gücü ve güvenliği, akış üreteçlerinin ürettikleri bit dizilerinin karakteristiğine bağlıdır. Şifreleme sırasında bir hata oluştuğunda, blok şifrelemede olduğu gibi hatanın yayılması söz konusu değildir. Her dizi elemanının, birbirinden bağımsız bir şekilde şifrlenmesi ya da şifresinin çözülmesi, hatanın yayılımına engel olur [Fischer, 2008] .

Akış şifreler eşzamanlı (senkron) ve eşzamansız (asenkron) olmak üzere iki türe ayrılırlar. Eşzamanlı akış şifrelemede anahtar dizisi, düz metin ve gizli anahtardan bağımsız olarak üretilir. Şekil 1.2'de eşzamanlı akış şifreleme ile düz metnin şifrlenmesini ve şifreli metnin çözülmesi işlemleri gösterilmektedir.



Şekil 1.2 Eşzamanlı Akış Şifrelemede Şifreleme ve Şifre Çözme İşlemleri

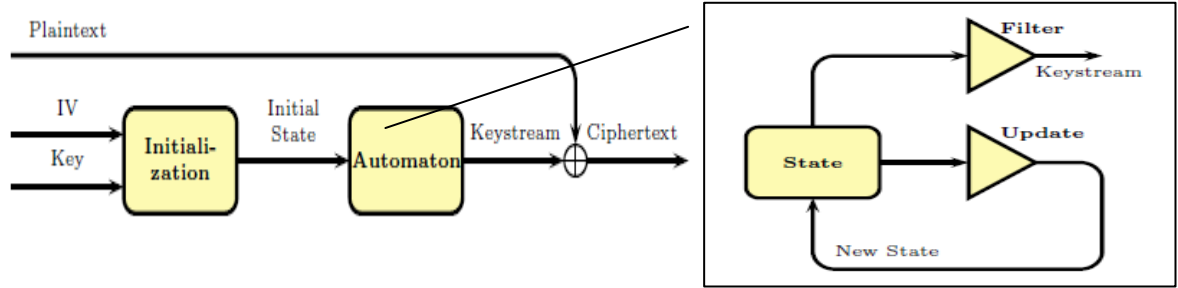
Şekil 1.3'te eşzamansız akış şifreleme ile düz metnin şifrlenmesini ve şifreli metnin çözülmesi işlemleri gösterilmektedir.



Şekil 1.3 Eşzamansız Akış Şifrelemede Şifreleme ve Şifre Çözme İşlemleri

Her iki şifreleme türü de sonlu durum otomatıdır; ancak eşzamansız akış şifrelemede anahtar dizisi, sabit uzunluktaki bir önceki şifreli metinlerin ve anahtarın bir fonksiyonu ile elde edilir. Eşzamansız akış şifrelemede, bir iletim hatası olduğunda şifrenin tekrar eşzamanlanması mümkün olacaktır. Ancak

eşzamanlı akış şifrelemelerde bir hata olduğunda (şifreli metnin bir dizi elemanı kaybolabilir, silinebilir) şifre çözülemez, tekrar eşzamanlama yapılır. Bunun için Şekil 1.4'te olduğu gibi başlangıç vektörü kullanılarak anahtar dizisi üretilir. Bu sayede eşzamanlı şifreleme de hata yayılımı olmamaktadır [Fischer, 2008].



Şekil 1.4 Eşzamanlama Akış Şifrelemede Eşzamanlama Yapılması

Bu tez çalışması kapsamında, akış şifreleme yöntemlerinden; LFSR (Linear Feedback Shift Register) ve PRNG (Pseudo Random Number Generators) tabanlı yöntemler üzerinde çalışılmıştır. Temelde bakıldığında akış şifreleme algoritmaları, donanım ve yazılım uygulamalarına uygunlukları bakımından iki kategoride incelenebilir. LFSR tabanlı algoritmalar donanım uygulamalarında tercih edilirken, PRNG tabanlı algoritmalar da yazılım uygulamalarında tercih edilmektedir.

Donanım uygulamalarında kullanılan akış üreteçlerinin yapı taşları olarak LFSR gösterilebilir. Bunun nedeni olarak donanımsal uygulamalardaki uygunlukları, üretilen dizinin uzun periyoda sahip olması ve iyi istatistiksel özellikler göstermesi verilebilir. Bu tez çalışması kapsamında, LFSR tabanlı yaygın olarak kullanılan A5/1, A5/2 ve RC4 algoritmaları incelenmiştir.

Yazılım uygulamalarında tercih edilen akış üreteçleri PRNG tabanlı algoritmalarlardır. Bunun nedeni ise sözde rastsal sayı üreteçleri (PRNG) rastsal sayıların özelliklerine yakınsayan sayı dizisi üretmektedir. Sayı dizileri rastsal başlangıç değerleri kullanılarak üretilen ve gerçekte rastsal olmayan sayılardan oluşmaktadır [Büyüksaraçoğlu, 2011]. Bu tez çalışması kapsamında, PRNG tabanlı yaygın olarak kullanılan Blum Blum Shub, LCG, ICG ve Mersenne Twister algoritmaları incelenmiştir.

Bu yöntemlerin, şifreleme uygulamalarında kullanılabilmesi için seçilen üreteçlerin rastsal diziler üretmesi gerekmektedir. Bunun nedeni üreteçler tarafından üretilen sözde rastsal sayı dizilerinin anahtar dizi olarak kullanılmasıdır. Anahtar dizisi düz metin ile XOR işleminden geçirilip şifreli metnin oluşmasını sağlamaktadır. Bu nedenle güvenli ve tahmin edilemez olmasını sağlamak gerekir, bu da anahtar dizinin rastsal bir dizi olması ile mümkündür. Aksi takdirde anahtar dizisine kolayca ulaşılabilir ve şifreleme uygulamaları önemi yitirir. Bu nedenle akış şifreleme yöntemlerinde üreteçler tarafından üretilen sözde sayı dizilerinde rastsalılık aranmaktadır. Bu tez çalışması kapsamında LFSR ve PRNG tabanlı yöntemler incelenirken ürettikleri sayı dizilerinin rastsallıklarının incelenmesi için üreteçlere testler uygulanmıştır.

1.2 Tez Raporunun Yapısı

Tezin ilk bölümünde tez konusu hakkında genel bilgi verilmekte, geçmiş çalışmalar ve temel bilgiler sunulmaktadır. Konunun uygulama alanları ve geliştirilen sistemin genel yapısı da kabaca anlatılmaktadır.

İkinci bölümde, algoritmaların ürettikleri dizilerin kalitelerinin karşılaştırılabilmesi için, kullanılan istatistiksel testler ile ilgili bilgiler sunulmaktadır.

Üçüncü bölümde kullanılacak LFSR tabanlı akış şifreleme algoritmaları anlatılmakta ve istatistiksel test sonuçları değerlendirilmektedir.

Tezin dördüncü bölümünde ise PRNG tabanlı akış şifreleme algoritmaları anlatılmakta ve istatistiksel test sonuçları değerlendirilmektedir.

Beşinci bölümde, yapılan çalışmalar değerlendirilmekte ve sonuçlar tartışılmaktadır.

2 RASTSALLIK TESTLERİ

Rastsal dizilerin üretilmesi için kullanılan iki temel yöntem vardır. Bunlar rastsal sayı dizisi (RNG: Random Number Generator) ve sözde rastsal sayı dizi (PRNG: Pseudo Random Number Generator) olarak sınıflandırılır. RNG ve PRNG yöntemleri şifreleme, modelleme ve benzetim uygulamaları gibi birçok alanda kullanılır. Bu yöntemlerin, şifreleme uygulamalarında kullanılabilmesi için seçilen üreteçlerin rastsal diziler üretmesi gerekmektedir. Üretilen dizilerin rastsal olup olmadıklarına, dizilere uygulanan istatistiksel testler ile anlaşılmaktadır.

Bu tezde akış şifreleme algoritmaları kapsamında, LFSR tabanlı ve PRNG tabanlı anahtar dizisi üretme algoritmaları ele alınmıştır. Her iki yöntemde de sözde rastsal sayılar üretilmektedir. Bu nedenle sözde rastsal sayı üreteçleri, özellikleri ve testleri üzerinde durulmuştur.

2.1 Rastsallık

Ana kütleyle ait gözlemlerden rastgele birinin çekilmesi durumunda ana kütledeki her bir elemanın eşit çekilme şansının olmasına rastsal denir [Moore, 1990].

2.2 Tahmin Edilemezlik

Şifreleme uygulamalarında kullanılacak sözde rastsal sayı üreticinin tahmin edilemez olması gerekir. Sözde rastsal sayı üreticinin başlangıç değeri bilinmiyor ancak dizinin önceki rastsal sayıları biliniyorsa bile, dizinin bir sonraki elemanı tahmin edilememelidir. Bu özelliğin sağlanabilmesi için başlangıç değeri ile üretilen dizi elemanları arasında bir ilintinin kurulamaması gerekir. Ancak başlangıç değeri ve seçilen algoritma biliniyor ise üretilen dizinin elemanları tahmin edilebilir. Buna ek olarak, başlangıç değeri kayıt edilirse sözde rastsal dizi tekrar elde edilebilir veya doğrulanabilir.

2.3 Test Etme

Çeşitli testler ile üretilen dizilerin rastsallığı sorgulanmaktadır. Rastsallık olasılıksal bir özellik olduğundan; tanımlamalar olasılıksal ifadeler üzerinden yapılmaktadır. İstatistiksel testler ile dizinin elemanları arasında ilinti olup olmadığı, dizinin periyodik özellikler taşıyıp taşımadığı belirlenir. Uygulanan testler sonucunda bir ilintiye rastlanırsa bu durum dizinin rastsal olmadığını gösterir.

Tüm veriden örneklemeler alınır ve bu örneklemelere istatistiksel testlerin uygulanmasıyla hipotez testi yapılmaktadır. Hipotez testi, örneklemden elde edilen bilgilere bağlı olarak belirli bir hata payı ile hipotezin doğrulanmasıdır. Sıfır hipotezi ve alternatif hipotez olmak üzere iki hipotez birbirine karşı sınanır[NIST, 2010].

2.4 Hipotezler

2.4.1 Sıfır Hipotezi (H_0)

Literatürde yokluk ya da istatistiksel hipotez olarak da bilinmektedir. İki ya da daha fazla veri setine ait incelenen parametrelerin birbirlerine eşit olduğu ve parametrelerin birbirinden bağımsız olduğunu savunan istatistiksel hipoteze sıfır hipotezi H_0 denir. Özetle H_0 , beklenen ve gözlenen veriler arasında bir farklılık ve ilişkinin olmadığını savunur. Bu görüş savunulurken istatistiksel anlamlılık denilen yanılğı payı göz önüne alınır.

2.4.2 Karşıt Hipotez (H_1/H_a)

Literatürde alternatif ya da araştırma hipotezi olarak da bilinmektedir. İki ya da daha fazla veri setine ait incelenen parametrelerin birbirlerine eşit olmadığı, parametrelerin birbirine bağımlı olduğunu savunan hipoteze karşıt hipotezi denir. H_1 veya H_a ile ifade edilir. Özetle H_1 , beklenen ve gözlenen veriler arasında bir farklılık olduğu savunulur. Bir hipotez testinde iki tip hata olabilir.

Çizelge 2.1’de yer alan parametreler, anlamları ve test sonuçlarına göre alınan kararlar aşağıda detaylı olarak açıklanmıştır.

Çizelge 2.1 Gerçek Durum ve Karar Çizelgesi

Gerçek Durum	Karar	
	H_0 Kabul	H_0 Ret
H_0 Doğru	Doğru karar çıkarımı ($1 - \alpha$)	I.Tür hata (α)
H_0 Doğru Değil	II.Tür hata (β)	Doğru karar çıkarımı ($1 - \beta$)

α : Gerçekte doğru olan H_0 hipotezinin reddedilmesine ve H_1 hipotezinin kabul edilmesine I. tip hata denir. Sıfır hipotezinin reddedilme ölçütüdür. α parametresi anlam düzeyi, risk düzeyi, yanlış payı ya da hata payı olarak adlandırılmaktadır.

P (I. Tür Hata) = Birinci Tür Hata Olasılığı = α

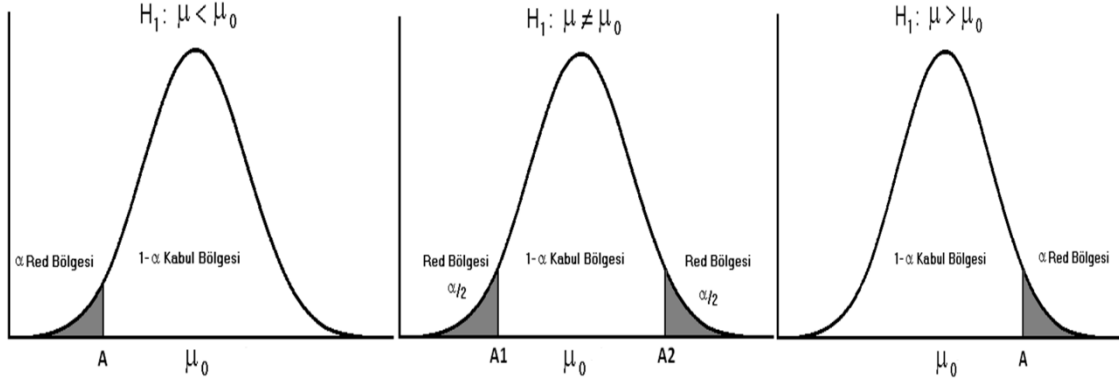
β : Gerçekte yanlış olan H_0 hipotezinin kabul edilmesine II. tip hata denir.

$(1 - \alpha)$: Gerçekte doğru olan H_0 hipotezinin kabul edilme olasılığı olup, testin güvenilirlik düzeyini ifade etmektedir.

$(1 - \beta)$: Gerçekte yanlış olan H_0 hipotezinin reddedilme olasılığı olup, testin gücünü ifade etmektedir. Güç, bir hipotez testinin isabetliliği için önemli bir kriter olup, maksimize edilmek istenir. Testin gücünün 1 olması o testin ideal olduğunu gösterir ancak gerçekte bu duruma çok nadir rastlanır.

I. Tür- α ve II. Tür- β tipi hatalar bilinçli olarak yapılan hatalardır. Hataların bilinçli yapılmasının sebebi olaylara birde tersinden bakma gereksinimidir. Hipotez testi yaparken α ve β hatalarını en aza indirmek için örneklem sayısı artırılabilir. İki hatayı aynı anda indirmek mümkün değildir. Çünkü α hatası ile β hatası ters ilişkilidir, iki hatadan biri azalırken diğeri artar. Aynı testte hem α hem de β hatası beraber yapılamaz. β hatasında dizi rastsal olmadığından, β hatası çok farklı değerler alabilir. Bu nedenle hesaplanması α hatasına göre zordur. Testlerin genellikle anlam düzeyinin α üzerinden incelenmesi ve sonuçların yorumlanmasının temelinde bu neden bulunmaktadır [NIST, 2010].

Seilen rneklemenin, daėılım eėrisinin saė ve solundaki ret blgelerinde olup olmadıėına bakılır. Ret blgesi demek; H_0 'ın kabul alanı dıėında kalan, H_0 'ın reddedilmesini saėlayan blgedir. Ret alanı ift taraflı veya tek taraflı olabilir.



ėekil 2.1 Ret Blgeleri

α deėerlerine baėlı olarak kritik A noktalarının belirlenmesi ile uygulanan istatistiksel testlerin hipoteze baėlı kararları oluėmaktadır. rneklemelerde yanlış bir iddiada bulunmamak ve sıfır hipotezini korumak iin anlam dzeyi (α) kk tutulur. Genellikle 0.05, 0.01 veya 0.001 deėerleri kullanılır. őifreleme uygulamalarında genellikle anlam dzeyi iin 0.01 deėeri tercih edilir [Sulak, 2011].

P-deėeri, H_0 kabul edildiėinde en az test iin hesaplanan deėer kadar byk bir deėer bulma olasılıėı olarak tanımlanır. Testin sonucunda oluėan P-deėeri incelendiėinde; H_0 kabul ediliyorsa retilen dizinin rastsal olduėu, H_0 reddediliyorsa retilen dizinin rastsal olmadıėı deėerlendirilir.

2.5 Testler

Rastsal sayı dizisi test edilirken aėaėıda belirtilen zelliklere sahip olması beklenir [NIST, 2010].

- Tekdzelik: PRNG ile retilen rastsal dizideki 0 ve 1 sayılarının eėit olması yani 0 ve 1 gelme olasılıėlarının 0.5 olması beklenmektedir.
- leklenebilirlik: Test edilebilecek olan herhangi bir dizi alt dizilere ayrılabilir ve eėer bir dizi rastsal ise ayrılan alt dizilerin de rastsal olması beklenir.
- Tutarlılık: Baėlangı deėerine gre retecin davranıėı deėiėmemelidir.

Rastsallığın test edilmesi, istatistiksel hipotez testlerine dayanır. Yazılım ya da donanım tabanlı PRGN ve RNG yöntemleri ile üretilen dizilerin rastsallığını test etmek için geliştirilmiş 15 testi içeren istatistiksel pakete NIST paketi denir. NIST paketi içindeki testler, rastsal olmayan farklı türde dizilerin tespit edilmesi için çeşitlendirilmiştir. Testler aşağıda sıralanmıştır [NIST, 2010].

- Frekans Testi
- Blok Frekans Testi
- Koşum (Öbek) Testi
- Bloktaki En Uzun Birler Testi
- İkili Matris Sırası Testi (Matris Rank)
- Ayırık Fourier Dönüşümü (Spektral) Testi
- Şablon ile Örtüşmeyen Uyum Testi
- Şablon ile Örtüşme Uyum Testi
- Maurer Evrensel İstatistiksel Testi
- Doğrusal Karmaşıklık Testi
- Seri Testi
- Yaklaşık Entropi Testi
- Birikimli Toplam Testi
- Rastsal Ayrılma Testi
- Rastsal Değişimli Ayrılma Testi

Belirtilen testlerden ilk olarak Frekans testinin yapılması önerilmektedir. Çünkü bu testten kalan dizi, diğer testlerde büyük olasılıkla kalacaktır. İstatistiksel testler arasında en uzun zamanda sonuç veren test ise Doğrusal Karmaşıklık Testidir.

İstatistiksel testlerden geçen dizilerin birörnek dağılıma sahip olup olmadığı incelenir. Dizinin ortalama μ ve varyans σ^2 değerlerine bakılarak dağılım ile arasındaki fark saptanır [L'Ecuyer, 2004].

2.5.1 Uygulanan Testler

Dizi üreteçlerinin rastsallığını ölçmek için bir istatistiksel test yeterli değildir. Bu konuda birçok test paketi üretilmiştir. NIST paketinden seçilen ve bu tezde uygulanan testlerin açıklaması örnekleri ile birlikte aşağıda verilmiştir.

2.5.1.1 Frekans Testi

Test edilecek dizide bulunan 0 ve 1'lerin oranı kontrol edilir. Testin geçerli sayılabilmesi için dizi uzunluğunun en az 100 bit olması gerekmektedir. Testin herhangi bir parametresi bulunmamaktadır.

Test sonucunda elde edilen P-değeri $\geq \alpha$ olduğunda, H_0 hipotezi kabul edilir. P-değeri $< \alpha$ olduğunda, test edilen dizideki 1 veya 0 sayılarının beklenenden fazla olduğunu göstermektedir. Bu durumda H_0 hipotezi reddedilir [Knuth, 1981].

n : Bit dizisinin uzunluğunu,

ε : PRNG ile üretilen bit dizisini simgelemektedir.

$$S_{\text{obs}} = \frac{|S_n|}{\sqrt{n}} \quad (2.1)$$

$$\text{Hata fonksiyonu } \text{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_z^{\infty} e^{-u^2} du \quad (2.2)$$

ve

$$\text{P-değeri} = \text{erfc}\left(\frac{S_{\text{obs}}}{\sqrt{2}}\right) \quad (2.3)$$

olarak tanımlanmıştır.

Örnek [NIST, 2010],

$$S_n = X_1 + X_2 + \dots + X_n \quad X_i = 2\varepsilon_i - 1 \quad (2.4)$$

$$\varepsilon = 1011010101 \quad n = 10$$

$$S_n = 1 + (-1) + 1 + 1 + (-1) + 1 + (-1) + 1 + (-1) + 1 = 2$$

Test istatistiği $S_{obs} = \frac{|S_n|}{\sqrt{n}} \rightarrow S_{obs} = \frac{|2|}{\sqrt{10}} = 0.632455532$ olarak hesaplanmıştır.

P-değeri $= \text{erfc}\left(\frac{S_{obs}}{\sqrt{2}}\right) \rightarrow 0.527089 \geq 0.01$ olduğundan üretilen dizi rastsal kabul edilir.

P-değeri < 0.01 olsaydı, bunu nedeni $|S_n|$ veya $|S_{obs}|$ değerlerinin büyük olması olabilir. S_n değerinin büyük pozitif değerlerde olması dizide çok sayıda 1 olduğunu, S_n değerinin büyük negatif değerlerde olması dizide çok sayıda 0 olduğunu göstermektedir.

2.5.1.2 Koşum Testi

Bir koşulda yan yana gelen 0'ların ya da 1'lerin sıralanışını ifade eder. Bu testin, Frekans testinden geçen dizilere uygulanması önerilir. 0 ve 1'ler arasındaki salınımlar incelenerek üretilen bit dizisinin yavaş ya da hızlı olabileceği konusunda fikir verir. Dizideki salınımlar ne çok hızlı (örn. 01010101), ne de çok yavaş (örn. 00001111) olmalıdır. Test edilen dizi uzunluğu en az 100 bit olmalıdır.

n : Bit dizisinin uzunluğunu,

ε : RNG ve PRNG ile üretilen bit dizisini,

π : Bit dizisindeki 1'lerin sayısını,

$V_n(\text{obs})$: Koşumların sayısını simgelemektedir.

$$\pi = \frac{\sum j\varepsilon_j}{n} \quad \text{olarak tanımlanır.} \quad (2.5)$$

$$\left| \pi - \frac{1}{2} \right| \geq \tau \quad (2.6)$$

ve

$$\tau = \frac{2}{\sqrt{n}} \quad (2.7)$$

olarak ifade edilir.

$$V_n(\text{obs}) = \sum_{k=1}^{n-1} r(k) + 1 \quad (2.8)$$

olduğu bilinmektedir.

Örnek [NIST, 2010],

$$\varepsilon = 1001101011 \quad n=10 \quad \pi = 6/10 = 3/5$$

Frekans testinden geçmiş olması ön şartı ile test edilecek dizinin bu teste uygun olup olmadığı değerlendirilir. $|\pi - 1/2| \geq \tau$ gösterilebilir ise Koşum testinin uygulanmasına gerek yoktur; çünkü Frekans testini geçememiş demektir. Bu durumda P-değeri sıfıra eşitlenir. Bu örnek için $\tau = 2/\sqrt{n}$ önceden tanımlanmıştır. Bu durumda aşağıdaki eşitliği sağladığından test yapılır.

$\tau = 2/\sqrt{10} \approx 0.63246$, $|\pi - 1/2| = |3/5 - 1/2| = 0.1 < \tau$ olduğundan frekans testini geçtiği anlaşılmaktadır.

$$V_n(\text{obs}) = \sum_{k=1}^{n-1} r(k) + 1$$

Test istatistiğini hesaplanması,

$r(k) = 0$, eğer $\varepsilon_k = \varepsilon_{k+1}$ ise

$r(k) = 1$, eğer $\varepsilon_k \neq \varepsilon_{k+1}$ ise olarak tanımlanmaktadır.

$$\varepsilon = 1001101011 \quad V_n(\text{obs}) = (1 + 0 + 1 + 0 + 1 + 1 + 1 + 0) + 1 = 7$$

$$P\text{-değeri} = \text{erfc} \left(\frac{V_n(\text{obs}) - (2 \cdot n \cdot a)}{2 \cdot \sqrt{2 \cdot n \cdot a}} \right) \quad (2.9)$$

ve

$a = \pi(1 - \pi)$ olarak ifade edilir.

$$P\text{-değeri} = \text{erfc} \left(\frac{7 - \left(2 \cdot 10 \cdot \frac{3}{5} \left(1 - \frac{3}{5} \right) \right)}{2 \cdot \sqrt{2 \cdot 10 \cdot \frac{3}{5} \cdot \left(1 - \frac{3}{5} \right)}} \right) = 0.147232 \geq 0.01 \text{ olduğundan üretilen dizi}$$

rastsal kabul edilir.

$V_n(\text{obs})$ değeri büyük ise dizideki salınım çok küçük olacaktır, bu da salınımın çok yavaş gerçekleştiğini göstermektedir.

2.5.1.3 Ayırık Fourier Dönüşümü (Spektral) Testi

Ayrık Fourier Dönüşüm testin amacı dizi elemanları arasında ilinti olup olmadığını ve dizinin periyodik özellikler taşıyıp taşımadığının tespit edilmesidir. Testin yapılmasındaki bir diğer amaç da dizinin tepe yüksekliklerine bakılarak %95 eşiğini geçen tepe sayısının, % 5'den daha farklı olup olmadığının tespit etmektir. Daha farklı bir bakış açısına göre Fourier analizine göre n sayılar kümesinin bağımsızlığı ölçülür. Bu test Knuth testi olarak da bilinir. Test edilen dizi uzunluğu en az 1000 bit olmalıdır. Diğer testlerden farklı olarak bu test için eklenmesi gereken bir "d" parametresi vardır.

n : Bit dizisinin boyunu,

$\mathcal{E}$: RNG ve PRNG ile üretilen bit dizisini,

d :Gözlenen ve beklenen frekans bileşenleri arasındaki fark, düzgelenmiş %95 eşiğinin dışını simgelemektedir.

Örnek [NIST, 2010],

$$X = x_1 + x_2 + \dots + x_n \quad X_i = 2\varepsilon_i - 1 \text{ olduğunda,}$$

$$\mathcal{E} = 1001010011 \quad n = 10$$

$$X = 1, -1, -1, 1, -1, 1, -1, -1, 1, 1$$

- X üzerine Ayırık Fourier Dönüşümü uygulanır ve aşağıdaki değer üretilir.

$$S = \text{DFT}(X) \quad (2.10)$$

- $M = |S'|$ hesaplanır. S' 'in ilk n/2 elemanından oluşan bir alt dizidir. Büyüklük işlemi tepe yüksekliklerinin dizisini üretir.

- $$T = \sqrt{\left(\log \frac{1}{0.05}\right)n} \quad (2.11)$$

T ile %95 tepe yüksekliği eşik değeri hesaplanır. Rastsallık varsayımı altında testten elde edilen değer % 95'i T değerini geçmemelidir.

- Rastsallık varsayımı altında beklenen tepe sayısı (N_0), T değerinden azdır. Bu örnek için $N_0 = 4.75$ değerindedir.

- M içerisindeki gözlenen tepe sayısı(N_1) T değerinden azdır. Bu örnek için $N_1 = 4$ alınmıştır.

$$d = \frac{(N_1 - N_0)}{\sqrt{n(.95)(.05)/4}} \quad (2.12)$$

Bu örnek için $d = \frac{(4 - 4.75)}{\sqrt{10(.95)(.05)/4}} = -2.176429$ olarak hesaplanmıştır.

- P-değeri = $\text{erfc}\left(\frac{|d|}{\sqrt{2}}\right)$ (2.13)

Bu örnek için P-değeri = $\text{erfc}\left(\frac{2.176429}{\sqrt{2}}\right) = 0.029523 \geq 0.01$ olduğundan üretilen dizi rastsal kabul edilir.

d değeri çok küçük ise az sayıda tepe noktasının T değerinin altında kaldığını (<%95), d değeri çok büyük ise çok sayıda tepe noktasının T değerinin üzerinde (>%5) kaldığını göstermektedir.

2.5.1.4 Ki-Kare Testi

Ki-Kare testi 1912 yılında Karl Pearson tarafından geliştirilmiştir. Ki-Kare; iki veya daha fazla veri seti arasında önemli farkın olup olmadığını belirlemede kullanabilecek bir istatistiksel testtir. Ki-Kare testinde gözlenen frekanslar (o_i) ile belli bir hipoteze göre elde edilen beklenen frekansları (e_i) karşılaştırılıp, arada önemli bir fark olup olmadığına, frekansların uyumuna bakılır [Özkaynak, 2006].

$$\text{Beklenen frekans} \rightarrow e_k = \frac{n}{m} = e_i \quad (2.14)$$

$$\text{Gözlenen frekans} \rightarrow f_k = o_i \quad (2.15)$$

$$X^2 = \sum_{k=1}^m \frac{(f_k - e_k)^2}{e_k} = \frac{m}{n} \sum_{k=1}^m \left(f_k - \frac{n}{m}\right)^2 \quad (2.16)$$

$$X^2 = \frac{(o_1 - e_1)^2}{e_1} + \frac{(o_2 - e_2)^2}{e_2} + \dots + \frac{(o_k - e_k)^2}{e_k} = \sum_{j=1}^k \frac{(o_j - e_j)^2}{e_j} \quad (2.17)$$

Test istatistiği olan $X^2 = 0$ ise gözlenen ve beklenen frekanslar arasında fark yoktur. $X^2 > 0$ ise gözlenen ve beklenen frekanslar arasında fark vardır ve uyumlu değildirler.

Beklenen frekansın (e_i), X^2 değerine etkisi büyük olacağından, testin uygulanabilmesi için beklenen frekansın $e_i \geq 5$ olması beklenir. e_i değeri küçüldükçe X^2 değeri yani gözlenen ve beklenen frekanslar arasındaki fark artmaktadır. X^2 büyüdükçe, farkın örneklem hatasından kaynaklanma olasılığı azalmaktadır. Bu durum da H_0 hipotezinin reddedilme olasılığını artırmaktadır [Bircan, 2003].

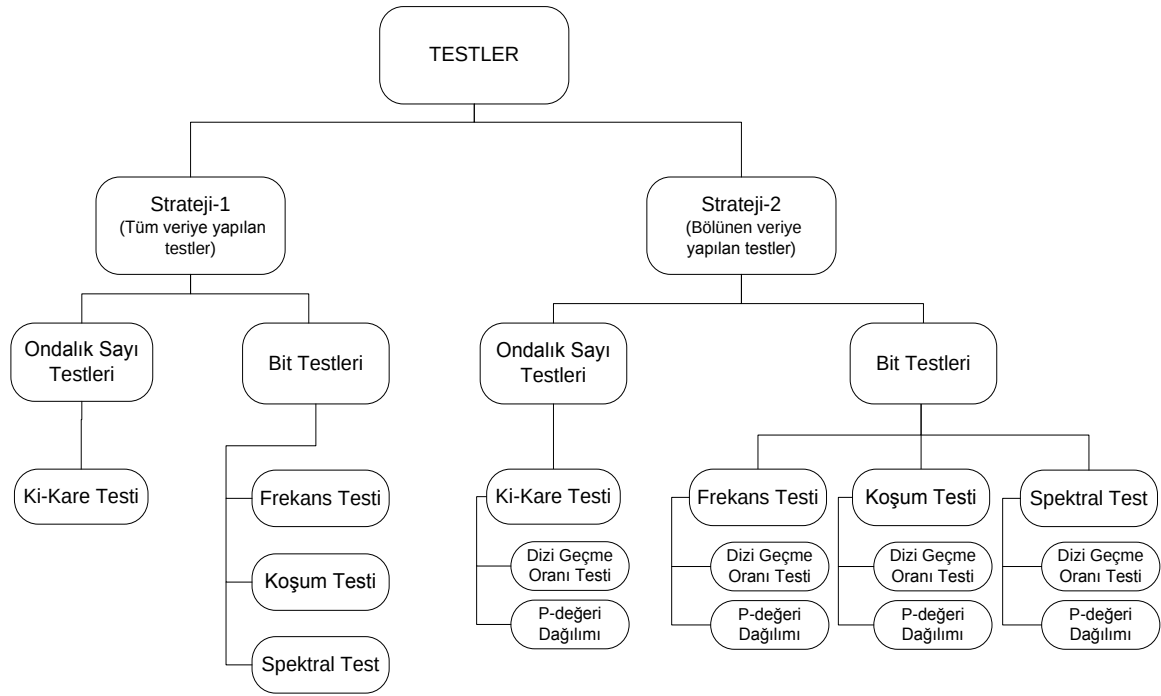
2.5.1.5 Histogram

İstatistiksel testler ve hipotezler yardımı ile elde edilen P-değerinin bir örnek dağılım sergileyip sergilemediğini grafiksel gösterimi için Histogram kullanılmaktadır [Law, 1991]. Histogram kullanımı ile üretilen dizinin dağılımının düzgün olup olmadığına bakılarak sayı üreticinin kalitesi hakkında bilgi edinilir. Bu tezde de histogram kullanılarak test sonuçları değerlendirilmiştir.

2.5.2 Test Uygulama Stratejileri ve Sonuçların Değerlendirilmesi

Testler iki stratejiye ayrılarak üreticilerin ürettikleri dizilerin rastsallıklarını gözlemlemek için uygulanır. Strateji-1 ve Strateji-2 olarak ayrılır. Strateji-1 de üreticiler tarafından üretilen tüm veriye (1000000 uzunluğundaki veri) testler yapılırken; Strateji-2 de ise bölünmüş veriye yani 5000 örneklem dizi uzunluğuna sahip 200 örnekleme test yapılır.

Şekil 2.2’de stratejilere göre yapılan testler yer almaktadır. Bu testler de kendi içlerinde bit ve ondalık sayılar üzerinden uygulanan testler olarak ayrılmaktadır.



Şekil 2.2 Testlerin Uygulanması

Bu tezde izlenen adımlar aşağıda belirtilmiştir.

1) Dizi Üreticisinin Seçimi

LFSR tabanlı ve PRNG tabanlı bir dizi üretici yöntemi seçilir. Bu üretici ile n uzunluğunda bir dizi üretilir.

2) Dizi Üretme

Seçilen dizi üretici ile n uzunluğunda bir dizi üretilir. n uzunluğundaki dizi m , örneklem sayısında alt dizilere bölünür. Testlerden doğru yorumlanabilecek sonuçlar elde etmek için dizi uzunluğu (n) ve örneklem sayısı (m) ile ilgili birtakım öneriler aşağıda bulunmaktadır.

a) Dizi Uzunluğu

Üretilen dizilerin uzunluğu n , en az 20000 bit civarında olmalıdır. Ancak bu kural Maurer Evrensel İstatistiksel Testi gibi bazı testler için sorun oluşturabilir. Daha az uzunluktaki dizilerin test istatistiklerini ele almak, daha büyük uzunluktaki dizilerinkine göre zordur. Kısıtlı bilgi ile bir değerlendirme yapmak gerekir [NIST, 2010].

b) Örneklem Sayısı

Örneklem sayısı seçilecek test ile bağlantılı olup, anlam düzeyinin tersi α^{-1} olarak alınabilir. Örneğin $\alpha = 0.001$ olduğunda, en az 1000 örneklemden oluşan n uzunluğundaki dizi test edilir.

3) Test Seçimi

Uygulanacak istatistiksel testler seçilir, teste uygun girdi sağlanır.

4) P-değerinin Analizi

P-değeri analiz edilerek, üretilen dizinin ve üreticinin kalitesi hakkında bilgi elde edilir. Deneysel test sonuçları incelenirken iki farklı çıktı üzerinden P-değeri incelenecektir.

a) Dizilerin Geçme Oranı (DGO)

Örneklem sayısı (m) içinde kaç dizinin testi geçtiği belirlenir ve testi geçme oranının, istenilen güvenli aralıkta olup olmadığı incelenir. Test, dizide rastsalılık olmadığını yakalamaya çalışmaktadır. Dizinin en azından belirli bir bölümünün testten geçmesi gerekir, bunun için dizilerin testi geçme oranı üstlü-altlı bir sınır olarak tanımlanmalıdır. Kabul edilecek oranların aralığı güvenli aralık olarak tanımlanır ve Eş. 2.18'de belirtildiği gibi örneklem sayısı ve anlam düzeyine bağlıdır. Testi geçme oranı, güvenli aralığın içinde ise dizi rastsal değil denemez; ancak bu oran güven aralığının dışında ise dizi rastsal değil denebilir. Güvenli aralığı aşağıdaki denklem ile bulunmaktadır [NIST, 2010].

$$\hat{p} \pm 3\sqrt{\frac{\hat{p}(1-\hat{p})}{m}} \quad (2.18)$$

ve

$$\hat{p} = 1 - \alpha \quad (2.19)$$

olarak tanımlanmaktadır.

Örnek,

$\alpha = 0.01$ anlam düzeyinde seçildiği durumda, ikili diziden oluşan 1000 örneklem ($m=1000$) test ediliyorsa, testin sonucunda da 996 dizinin P-değeri $\geq \alpha = 0.01$ eşitliğini sağlıyorsa dizilerin $996/1000 = 0.996$ oranındaki çoğunluğu testi geçmiştir.

$$0.99 \pm 3 \sqrt{\frac{0.99(1-0.99)}{1000}} = 0.99 \pm 0.0094392$$

0.996 oranı güvenli aralık içinde olduğundan 1000 örnekleme yapılan test sonucunda dizi için rastsal değil denemez.

Bu tezde farklı üreticiler tarafından üretilen dizilerin karşılaştırılmasında kullanılan yöntemlerden biri olup, her test için örneklem sayısı kadar h değeri elde edilir, testi geçme oranı ve güvenli aralık bulunur. Testi geçme oranı ve test sonucu (1 veya 0) çıktı olarak alınmaktadır. Bu değerlendirme, üreticinin kalitesi hakkında bilgi vermektedir.

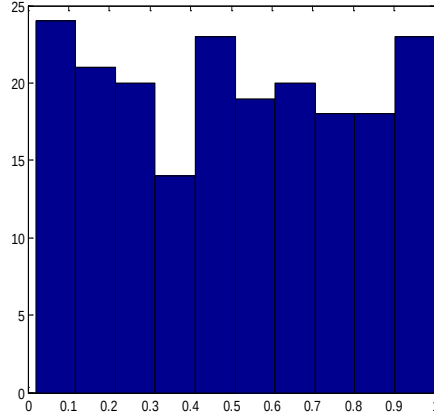
b) P-değerlerinin Dağılımı

P-değerlerinin analizinde kullanılan diğer bir yöntem de; dizinin birörnek dağılıma sahip olup olmadığının incelenmesidir. Testin sonucunda elde edilen P-değerleri görsel olarak histogram grafikleri ile değerlendirilmektedir. Tekdüzelik, üretilen diziye Ki-Kare testinin uygulanması ile elde edilen P-değeri ile belirlenmektedir. Bu ikinci yöntem için aşağıda örnek verilmiştir.

$$\chi^2 = \sum_{i=1}^{10} \frac{(F_i - s/10)^2}{s/10} \quad (2.20)$$

- i : Alt aralığı,
- s : Örneklem sayısını,
- F_i : i . Alt aralıktaki P-değerlerinin sayısını ifade etmektedir.

Bu tezde farklı üreteçler tarafından üretilen dizilerin karşılaştırılmasında kullanılan yöntemlerden biri olup, her test için örneklem sayısı kadar P-değeri elde edilir. P-değerlerinin Şekil 2.3'teki gibi histogramı çizdirilerek dizinin rastsal olup olmadığı değerlendirilir.



Şekil 2.3 Örnek Bir P-değeri Histogram Grafiği

5) Geçti/Kaldı Kararı

Yapılan tüm testler için P-değeri elde edilir. P-değerlerinin istenilen aralıkta olup olmamasına göre üretilen dizinin testten geçip geçmediğine karar verilir.

P- değeri $\geq \alpha$ olduğunda, H_0 hipotezi kabul edilir ve

- Test edilen dizinin ana kütleden çekilmiş rastsal örneklem olduğunu ve rastsallıktan herhangi bir sapma olmadığını
- Örneklem ortalaması ile örneklemin alındığı ana kütle ortalaması arasında gözlenen farkın önemli olmayan küçük bir fark olduğunu gösterir.

P- değeri $< \alpha$ değeri olduğunda, H_0 hipotezini reddedilir ve

- H_1 hipotezi kabul edilir.
- Test edilen dizinin ana kütleden çekilmiş rastsal örneklem olmadığını ve rastsallıktan sapma olduğunu gösterir.

3 DOĞRUSAL GERİ BESLEMELİ KAYAN YAZMAÇ TABANLI YÖNTEMLER

Kriptografik algoritmaların donanımsal uygulamalara entegrasyonu, gelişen yüksek hız ve yüksek güvenilirlik seviyesinde iletişim ihtiyaçları doğrultusunda önemli bir rol oynamaktadır. Ancak yüksek işlemci gücü isteği, yüksek hızlı ağ yapısına engel olmaktadır. İletişim ağına uygulanacak şifreleme yüksek veri işleme hızı gerektirdiğinden ağın bant genişliğini verimli kullanılmalıdır. FPGA gibi ürünlere şifreleme algoritmalarını entegre etmek, yazılım tabanlı ürünlere veya genel amaçlı bilgisayarlara entegre etmekten daha başarılı bir performans sağlamaktadır [Galanis, 2005].

Boyu L olan bir LFSR'ın ürettiği dizinin periyodu, uygun bit seçimleri yapıldığı takdirde en fazla $2^L - 1$ değerindedir. Maksimum periyotta dizi üretmek için kullanılan polinomunun $C(D)$ çarpanlarına ayrılamaz olması gerekir. Dizinin periyodunun maksimum olması için $C(D)$ polinomunun böldüğü en küçük dereceli polinom $1 + D^{2^L - 1}$ olmalıdır. Bunu sağlayan $C(D)$ polinomuna ilkel polinom denir.

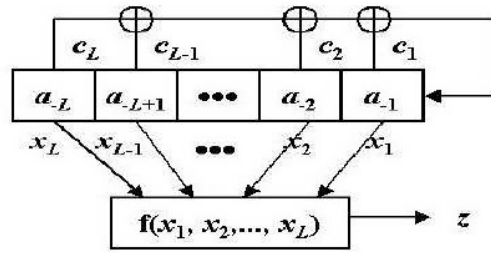
Örnek,

Çizelge 3.1 Maksimum Periyodu Sağlayan Polinom Seçimi Örneği

n bit LFSR	Geri Beslemeli Polinomlar ($C(D)$)	Periyod ($2^n - 1$)
4	$x^4 + x^3 + 1$	15
11	$x^{11} + x^9 + 1$	2047
17	$x^{17} + x^{14} + 1$	131071

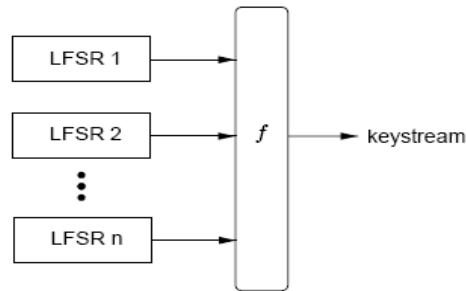
LFSR algoritmalarında önemli olan diğer bir parametre doğrusal karmaşıklıktır (LC: Linear Complexity). Maksimum uzunlukta bir LFSR, doğrusal karmaşıklığı en fazla kendi boyuna eşit bir dizi üretebilir. Bu da doğrusal karmaşıklık (LC) için küçük bir değerdir. Dizinin doğrusal karmaşıklığını arttırmak için çeşitli yollar vardır:

- LFSR'a doğrusal olmayan bir fonksiyon bağlanabilir. Bu filtre doğrusal olmayan yani derecesi en az iki olan bir fonksiyondur.



Şekil 3.1 Doğrusal Olmayan Filtre ile Doğrusal Karmaşıklığı Artırma

- Birden fazla LFSR doğrusal olmayan bir fonksiyonla bağlanabilir.



Şekil 3.2 Doğrusal Olmayan Fonksiyon ile Doğrusal Karmaşıklığı Artırma

Eğer LFSR'lar maksimum uzunlukta (L) ise ve ürettikleri dizilerin periyodu ikiden büyük ve birbirlerinden farklı ise z dizisinin doğrusal karmaşıklığı $F(L_1, L_2, \dots, L_n)$ dir.

- Birden fazla LFSR saat kontrollü ile tetiklenir. Bunlara Saat Kontrollü Üreteçler denir.

➤ Değişen Adımlı Üreteçler: Üç tane LFSR kullanılır.

LFSR1 (L_1), LFSR2 (L_2) ve LFSR3(L_3) maksimum periyoda sahip diziler üretiyorsa ve $\text{OBEB}(L_1, L_2) = 1$ ise üretilecek z dizisinin;

→ Periyodu $2^{L_1} (2^{L_2} - 1) (2^{L_3} - 1)$,

→ Doğrusal karmaşıklığı $(L_2 + L_3) 2^{L_1-1} < \text{LC}(z) \leq (L_2 + L_3) 2^{L_1}$

şeklinde ifade edilir.

➤ Küçülen Üreteçler (Shrinking Generator): İki LFSR aynı anda çalışır.

→ L_1 ve L_2 aralarında asal ise oluşan dizinin periyodu $(2^{L_1} - 1)(2^{L_2} - 1)$,

→ Doğrusal karmaşıklığı ise $L_2(2^{L_1-2}) < L.C.(z) \leq L_2(2^{L_1-1})$ dir.

Bu tezde LFSR tabanlı sözde rastsal sayı üretici algoritmalarından en yaygın olan A5/1, A5/2 ve RC4 yöntemleri incelenmiş ve karşılaştırılmıştır. Algoritmaların rastsallıkları ve şifreleme uygulamalarına uygunlukları çeşitli istatistiksel testler yapılarak incelenmiştir. Test sonuçları değerlendirilerek, elde edilen sonuçlar her algoritmanın uygulama bölümünün sonunda verilmiştir.

3.1 A5/1 Algoritması

Tüm dünyada geniş bir kullanıcı kitlesi tarafından kullanılan GSM sisteminde, baz istasyonu ile mobil cihaz arasındaki veri alış-verişinin şifrlenmesinde A5 algoritması kullanılmaktadır. A5 akış şifreleme algoritması A5/1 ve A5/2 olmak üzere iki ana türden oluşmaktadır.

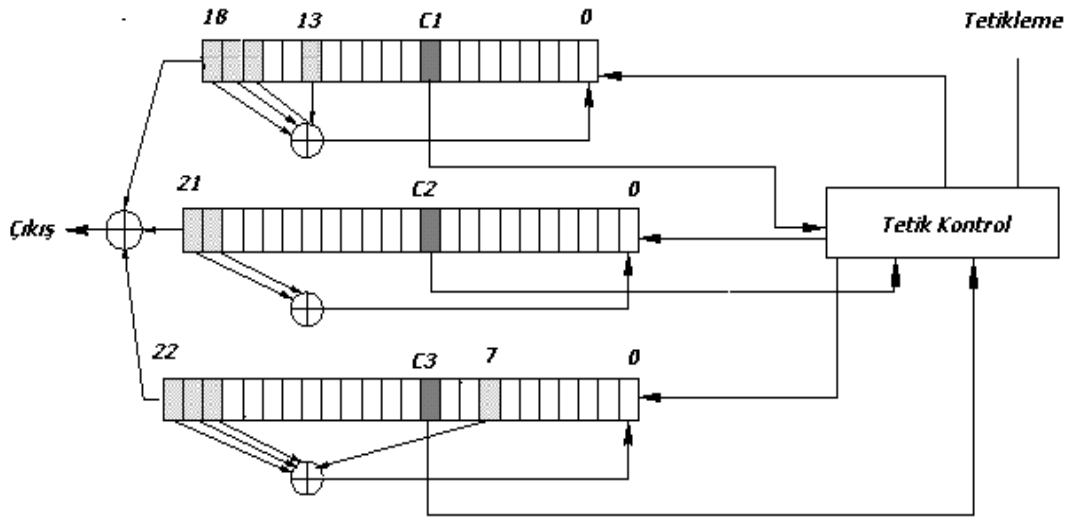
Hava kanalı üzerinden ses şifrelemede A5 akış şifreleme yöntemi kullanılmaktadır; şifrelemede kullanılan algoritmaya göre bit veya bayt üzerinden şifreleme yapılmaktadır. 1987 yılında geliştirilen A5/1 akış şifreleme algoritması, hava kanalı üzerinden ses şifrelemede kullanılan bir algoritmadır. 2000 yılında 130 milyon, 2011 yılında 4 milyar civarında olan GSM müşterisinin sesli iletişim gizliliği için A5/1 şifreleme algoritması kullanılmaktadır. LFSR tabanlı bir algoritma olduğundan donanım uygulamalarında rahatlıkla ve kolaylıkla kullanılmaktadır.

GSM de her bir konuşma çerçeveler dizisi halinde 4.6 milisaniyede gönderilir. Her bir çerçeve, mobil cihaz-baz istasyonu iletişimi için 114 bit, baz istasyonu-mobil cihaz için 114 bit olmak üzere 228 bitten oluşmaktadır [Ahmed, 2009].

Kullanıcıya ait bilgiler 64 bit K_c gizli oturum anahtarı ile şifrelenir. K_c anahtarı, çerçeve numarası F_n ile karıştırılır. A5/1 algoritmasının başlangıç durumunun tamamlanması ile 228 bitlik sözde rastsal bitlerden oluşan bir dizi oluşmuş olur [Erguler, 2005]. Her konuşma için bu işlemler tekrarlanır.

Her çerçeve için o çerçeveyi tanımlayan ve 22 bittten oluşan çerçeve numarası F_n verilir, her çerçevede bu numara değişir. Aynı K_c anahtarı arama süresince kullanılır, fakat 22 bit çerçeve numarası (F_n) arama süresince değişir. Her 228 bit bir çerçeveyi oluşturur. Bir çerçevede 228 bit tamamlandığında, yeni bir çerçeve oluşur ve tekrar başlangıç durumuna getirilerek yeni bir çerçeve numarası verilir. Bu sayede her bir çerçeve için benzersiz bir anahtar dizisi kullanılır.

A5/1 akış şifreleme, ikili LFSR tabanında bir anahtar dizisi üreticidir. A5/1 şifreleme üç LFSR ve bu LFSR'ların düzensiz olarak tetiklemesi temeline dayanan bir algoritmadır. A5/1 şifreleme algoritmasının yapısı Şekil 3.3'te gösterilmektedir.



Şekil 3.3 A5/1 Algoritmasının İç Yapısı

Bu algoritmada üç farklı uzunlukta yazmaç kullanılmakta olup, üç yazmacın toplam uzunluğu 64 bittir. Yazmaçlar sırasıyla R1, R2, R3 olmak üzere 19, 22 ve 23 bit uzunluğunda geri beslemeli polinomlar ile ifade edilir. R1'in maksimum periyodu $2^{19} - 1$, R2'nin $2^{22} - 1$ ve R3'ün maksimum periyodu $2^{23} - 1$ uzunluğundadır. Üç yazmaç aynı anda tetiklenirken yazmaçların toplam periyodu $(2^{19} - 1) \times (2^{22} - 1) \times (2^{23} - 1)$ uzunluğundadır. Ancak deneysel olarak üç yazmacın toplam periyodu, $(4/3) \times (2^{23} - 1)$ olarak gösterilebilmektedir.

Çizelge 3.2 A5/1 Algoritmasının Matematiksel Yapısı

LFSR No	LFSR Uzunlukları	Geri Beslemeli Polinomlar	Tetikleme Tap Biti	Tap Biti
R1	19 bit	$x^{19} + x^{18} + x^{17} + x^{14} + 1$	8	13, 16, 17, 18
R2	22 bit	$x^{22} + x^{21} + 1$	10	20, 21
R3	23 bit	$x^{23} + x^{22} + x^{21} + x^8 + 1$	10	7, 20, 21, 22

Her bir yazmacın en önemsiz biti sıfır olarak etiketlenir. Çizelge 3.2’de belirtildiği gibi R1 yazmacındaki 13., 16., 17. ve 18. bit pozisyonları, R2’deki 20. ve 21. bit pozisyonları, R3’deki 7., 20., 21. ve 22. bit pozisyonları önemli bit grupları olup tap olarak isimlendirilir. Yazmaçlar her tetiklendiğinde her bir yazmaca ait tap bitleri XOR operasyonuna girer ve sonuç en önemsiz bitte tutulur. Sola kaydırma işlemi sonucunda R1, R2 ve R3’den çıkan en önemli bitler (MSB) XOR operasyonundan geçirilerek çıktı oluşturulur.

Her bir yazmaçta tek bir tetikleme tap biti vardır. R1 için 8. bit, R2 için 10. bit ve R3 için 10. bit tetikleme tap bitleridir. Tetiklemenin kontrolü çoğunluk kuralına göre yapılır. Her bir saat çevriminde tetiklenen tap bitleri çoğunluk fonksiyona girdi olur ve fonksiyon sonucu olan çoğunluk biti ile uyuşan tap bitlerine ait yazmaçlar tetiklenir. Çoğunluk kuralının A5/1 şifreleme algoritmasında nasıl işlediği aşağıda belirtilmiştir. Eğer tetikleme bitlerinden iki ya da daha fazlası 0 değerindeyse çoğunluk fonksiyonun çıktısı $m=0$, 1 değerinde ise $m=1$ değerindedir.

$$m = \text{Majority}(R1[8], R2[10], R3[10]) \rightarrow \text{maj}(a,b,c) = a \cdot b \oplus b \cdot c \oplus c \cdot a \quad (3.1)$$

$$\bullet \quad R1[8] = m \quad \text{ise} \quad R1 \text{ tetiklenir} \quad (3.2)$$

$$\bullet \quad R2[10] = m \quad \text{ise} \quad R2 \text{ tetiklenir} \quad (3.3)$$

$$\bullet \quad R3[10] = m \quad \text{ise} \quad R3 \text{ tetiklenir} \quad (3.4)$$

Çoğunluk fonksiyonun yapısı nedeniyle; bir yazmacın tetiklenme olasılığı $3/4$ iken, tetiklenmeme olasılığı $1/4$ olmaktadır [Erguler, 2005].

3.1.1 A5 Algoritmasının Zayıf Yönleri

A5/1 ve A5/2 şifreleme algoritmaları uzun bir süre gizli kalsa da 1994 yılında tasarımı dışarıya sızdırıldı. İlk saldırı 1994 yılında Ross Anderson tarafından gerçekleştirilmiştir. A5/1 ve A5/2 algoritmalarının tasarımı, Marc Briceno tarafından ters mühendislikle 1999 yılında elde edilmiştir [Pandya, 2001] . Diğer algoritmalar gibi A5/1 de bugüne kadar birçok teknik kullanılarak kırılmıştır. Bu saldırıların çoğu bilinen metin saldırılarıdır. Son yıllarda Biryukov, Shamir ve Wagner tarafından yapılan iki tip saldırıda birkaç dakika veri alınması ile saniyeler süresince saldırı gerçekleştirilmiştir. Bilinen Metin saldırıları aşağıda özetlenmiştir.

Çizelge 3.3 A5 Algoritmasına Yapılan Saldırıları

Saldıran	Saldırı Yılı	Saldırı Tipi	Önişleme Adımları	Mevcut Veriler	İhtiyaç Duyulan Bellek	Saldırı süresi
Golic [Golic, 2002]	1997	Doğrusal denklemlerin çözümü	$2^{40.16}$			
		"Biased Birthday"	2^{42}	2 dakika	300 GB	1 saniye
		"Biased Birthday"	2^{48}	2 dakika	150 GB	1 saniye
Biryukov	2000	Zaman-Hafıza Ödünleşimi	2^{48}	2 dakika	300 GB	1 saniye
Biryukov, Shamir, Wagner [Biryukov, 2001]	2000	Rastsal Altçizge	2^{48}	2 dakika anahtar dizisi bilgisi	300 GB	Birkaç dakika da K_c
Biham	2000		2^{38}	$2^{20.8}$ bit	32 GB	$2^{39.91}$ tetiklenme süresi
Ekdahl, Johansson	2003	İklendirme yok	-	2-5 dakika		Birkaç dakika
Maximov	2004		-	Birkaç saniye		<1 dakika
Timo Gendrulis ve ekibi [Gendrulis, 2008]	2008	Tahmin Et ve Belirle	COPACO BANA isimli özel amaçlı donanım ürünü kullanımı	64 bit anahtar dizisi		<6 saat İçteki tüm durumlar [Kumar, 2006]

A5/1 algoritmasında tetikleme kontrolünü yapan R1,R2 ve R3 yazmaçlarının tap bitlerinin yerleri bilindiğinden doğrusal denklemler ile bazı bitler tahmin edilebilir. A5/1 algoritmasının tek bir bilgisayarın birkaç dakikasını ve 150-300 GB hafızasını kullanarak kırılabilirdiği farklı çalışmalar ile kanıtlanmıştır [Erguler, 2005]. 2003 yılında Barkan, Biham ve Keller tarafından A5/1 ve A5/2 algoritmalarına saldırı gösterimi yapılmıştır. Saldıranlar GSM mobil telefon konuşmalarını kaçak olarak dinlemiş, gerçek zamanlı ve sonraki bir zaman için tüm veriyi çözmeyi başarmışlardır.

A5/1 algoritmasının zaman karmaşıklığı 2^{54} civarındadır. A5/2 algoritması 2^{17} zaman karmaşıklığında olduğundan kolay ve hızlı bir şekilde kırılabilir [Petrovic, 2000]. A5/1 şifreleme algoritması, A5/2 şifreleme algoritmasına göre daha güçlü ve saldırılara daha dayanıklı bir algoritmadır.

3.1.1.1 A5 Şifreleme Algoritmalarına Karşı Kaba Kuvvet Saldırısı

Gerçek zamanda GSM güvenliğine karşı yapılan Kaba Kuvvet saldırısı pratik değildir. Çünkü algoritmanın zaman karmaşıklığı 2^{54} (64 bitlik anahtar uzunluğunun son 10 biti sıfır değilse 2^{64}) olduğundan yüksektir. Bu saldırıda A5/1 algoritması ile şifrelenen metnin çözülebilmesi için, gerçek zamanda GSM aramasının uzun bir süre dinlemesi gerekmektedir. Kaba Kuvvet saldırıları, aktif saldırı için uygun değildir. Ama baz istasyonu ile mobil cihaz arasındaki çerçeveler kaydedilerek bundan sonra yapılacak saldırı için kullanılırsa, pasif saldırı yapılmış olur.

3.1.1.2 A5 Şifreleme Algoritmalarına Karşı Böl ve Fethet Saldırısı

Bu saldırı tipi bilinen metin saldırısı sınıfına girmektedir. Böl ve fethet saldırısı, zaman karmaşıklığını 2^{54} 'den 2^{45} 'e düşürmektedir. Bu saldırı, Kaba Kuvvet saldırısından 2^9 defa daha hızlıdır.

Saldırgan bilinen anahtar dizisinden LFSR'ların başlangıç durumunu belirlemeye çalışır. Saldırgan eğer şifreli metnin ve bunun ilişkili olduğu orijinal metni bilirse 64 bitlik anahtar dizisini ele geçirebilir. Bu büyük oranda ön ve arka GSM çerçevelerinin formatına bağlıdır. 64 bitlik anahtar dizisinin daima bilinmesine gerek yoktur; fakat 32'den 48 bite kadar olan kısım genelde bilinir. Dikkat edilmesi

gereken konu, saldırganın yalnızca 64 bit orijinal metin parçasına ihtiyacı olduğudur. Kısa Böl ve Fethet saldırısı iki LFSR'ın içeriği tahmin edilerek üçüncü LFSR'ın bilinen anahtar dizisinin hesaplanmasıyla gerçekleştirilir. Eğer R1 ve R2'nin tetiklenmesi R3'e bağlı değilse 2^{40} saldırı, bağlı ise 2^{45} saldırı gerekir.

Golic'in önerdiği aynı varsayıma dayanan bir böl ve fethet saldırısının ortalama karmaşıklığı $2^{40.16}$ 'dır. LFSR'lerdeki n bitin doğrusal eşitliklerle nasıl tahmin edileceği tanımlanmaktadır. Bu doğrusal eşitliklerin çözülmesiyle üç LFSR'nin başlangıç durumları ortaya çıkarılabilir. Doğrusal eşitliklerin çözüm karmaşıklığı $2^{41.16}$ 'dır. Ortalama olarak iç durum $2^{40.16}$ işlemde yüzde elli olasılıkla çözülür.

Golic ayrıca "Birthday Paradox" tabanlı Zaman-Hafıza Getiri-Götürü saldırısını da önermiştir. Saldırının amacı bilinen anahtar dizisine karşı gelen bilinen çerçeve numarası için bilinen zamanda üç LFSR'nin iç durumunun ortaya çıkarılmasıdır, böylece oturum anahtarı (K_c) yeniden oluşturulur.

Bunlara ek olarak A5/x algoritmalarında kullanılan yazmaçların uzunluklarının kısa olması nedeni ile algoritmaların geniş anahtar arama saldırılarına karşı zayıf [Sankaliya, 2011] olduğu bilinmektedir.

Saldırılara karşı önerilen yazılımsal ve donanımsal tedbirler aşağıda tanımlanmıştır.

- Donanımı ve tetikleme kontrolünü zorlaştırma, LFSR yapısını karmaşıklaştırma
- Yazılım uygulaması ile ek bir şifreleme yöntemi ekleme

3.1.2 A5/1 Algoritmasının Uygulanması

A5/1 algoritması hakkında yukarıdaki bölümde genel olarak bilgi verilmiştir. A5/1 algoritmasının kodlanması, MATLAB programlama dili ile gerçekleştirilmiştir. A5/1 algoritmasının kodlanma sırasında dikkat edilen algoritma kıstasları ve algoritma adımları aşağıda belirtilmiştir.

1. R1, R2 ve R3 yazmaçların her bir biti sıfırlanır.

$$R1 = R2 = R3 = 0 \quad (3.5)$$

Tetikleme kontrolü yapılmadan yazmaçlar tetiklenir.

For i = 0 to 63 do

$$R1[0] = R1[0] \oplus K_c[i] \quad (3.6)$$

$$R2[0] = R2[0] \oplus K_c[i] \quad (3.7)$$

$$R3[0] = R3[0] \oplus K_c[i] \quad (3.8)$$

K_c anahtarının her bir biti ile her bir yazmacın en önemsiz bitine XOR işlemi uygulanır, tetikleme kontrolü uygulanmadan yazmaçlar tetiklenir. K_c anahtarı 64 bitten oluştuğundan bu işlem 64 çevrimde tamamlanır. XOR operasyonu K_c anahtarının en önemsiz bitten (LSB) en önemli bitine (MSB) doğru yapılır.

2. 1. Adımda K_c anahtarı ile yapılan çevrim, 2. adımda 22 bitlik F_n için 22 çevrimde tamamlanır.

For i = 0 to 21 do

$$R1[0] = R1[0] \oplus F_n[i] \quad (3.9)$$

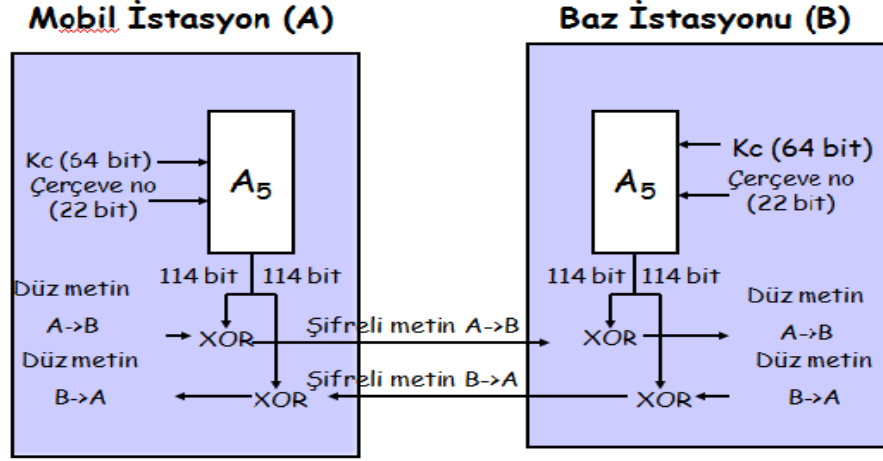
$$R2[0] = R2[0] \oplus F_n[i] \quad (3.10)$$

$$R3[0] = R3[0] \oplus F_n[i] \quad (3.11)$$

Çerçeve numarasının (F_n) her bir biti ile her bir yazmacın en önemsiz bitine XOR işlemi uygulanır, tetikleme kontrolü uygulanmadan yazmaçlar tetiklenir. F_n , 22 bitten oluştuğundan bu işlem 22 çevrimde tamamlanır. XOR operasyonu, F_n çerçeve numarasının en önemsiz bitten (LSB) en önemli bitine (MSB) doğru yapılır.

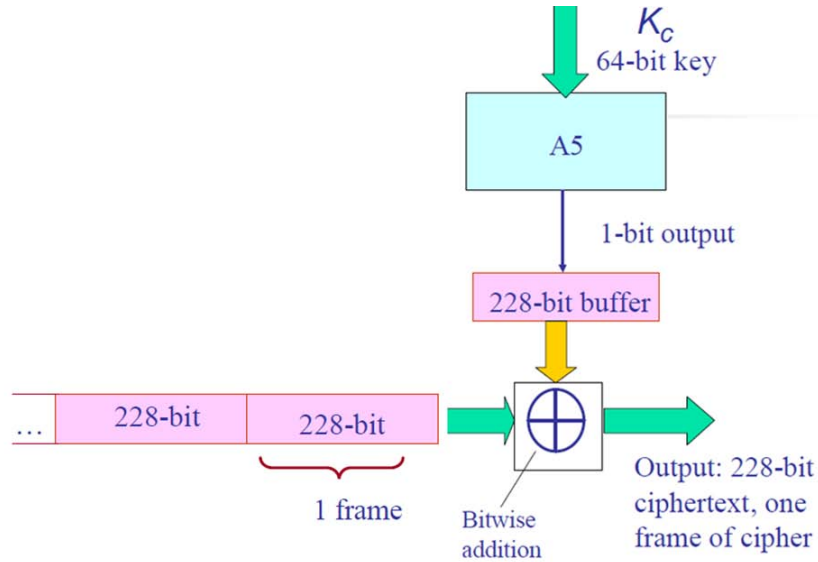
3. Her yazmaç, çoğunluk kuralı ile tetik kontrolü yapılarak 100 çevrim için tetiklenir. Elde edilen 100 bitlik çıktı kullanılmaz.
4. Üç yazmaç, tetik kontrolü yapılarak 228 çevrim için tetiklenir. Her bir çevrimde üç yazmacın en önemli bitine XOR işlemi uygulanarak tek bir çıkış biti üretilir.

Her çevrimde çıkan bir bitlik çıktı 228 bitlik arabelleğe yazılır ve 228 çevrim sonunda 228 bitlik A5/1 algoritma çıktısı elde edilir. 114 biti mobil istasyondan baz istasyonuna, diğer 114 biti ise baz istasyonundan mobil istasyona gönderimde kullanılır.



Şekil 3.4 A5/1 Algoritmasının Veri Akışı

K_c ve F_n kullanılarak A5/1 algoritması ile anahtar dizisi üretilir ve çıktı 228 bit olana kadar arabellekte depolanır. 228 bit boyutundaki bir düz metin ile XOR işleminden geçirilerek şifreli metin oluşturulur.



Şekil 3.5 A5/1 Algoritması ile Düz Metnin Şifrelenmesi

5. Testler iki stratejiye ayrılarak yapılır. Strateji-1 de üret  tarafından  retilen t m veriye (1000000 uzunlu undaki veri) testler yapılırken; Strateji-2 de ise b l nm   veriye yani 5000  rneklem dizi uzunlu una sahip 200  rnekleme test yapılır.

A5/1 algoritmasının  ifreleme uygulamalarında kullanılması i in iyi istatistiksel  zelliklere sahip diziler  retmesi beklenir. Bunun sa lanıp sa lanmadı ı istatistiksel testler ile g zlemlenir.

3.1.3 Uygulanan Testlerin Sonu ları ve De erlendirmeler

T m  rete lere

- Frekans Testi,
- Ko um Testi,
- Ayrık Fourier D n   m  (Spektral) Testi

yapılarak elde edilen test sonu ları  izelge ve grafik halinde bu b l mde sunulmu  ve test sonu lar yorumlanmı tır.

Bu testlerin sıfır hipotezini g steren h ve P -de erleri incelenir. $h = 0$ ise sıfır hipotezinin kabul edildi ini g sterir. Yapılan testler anlamlılık d zeyi α olarak %1 se ilmi tir. Bu nedenle P -de eri ≥ 0.01 ise  retilen dizi rastsaldır.

Algoritma girdisi olarak [Briceno, 1990];

- Anahtar (K_c): 0x12, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF
-  er eve numarası (F_n): 0x134

Algoritma  ıktısı olarak [Briceno, 1990];

- Anahtar dizisi: 0x53, 0x4E, 0xAA, 0x58, 0x2F, 0xE8, 0x15, 0x1A, 0xB6, 0xE1, 0x85, 0x5A, 0x72, 0x8C, 0x00

alınmı tır.

Test girdileri olarak;

- Algoritma çıktısı
- Üretilen dizi uzunluğu : 1000000
- Örneklemdeki dizi uzunluğu : 5000
- Örneklem sayısı : 200
- $\alpha = 0.01$ değerleri seçilmiştir.

A5/1 algoritmasının çıktıları testlerden geçirilmiştir. A5/1 algoritması üretici için test sonuçları aşağıdaki çizelge ve grafikler ile verilmiştir.

3.1.3.1 Strateji-1 Testleri

Seçilen üretici ile üretilen 1000000 dizi uzunluğundaki veriye Frekans, Koşum ve Spektral testleri yapılmıştır.

Tüm testlerden elde edilen h ve p değerleri incelenmiştir. Anlam düzeyi (α) %1 seçildiği durumda Çizelge 3.4'te görüldüğü gibi seçilen üretici için tüm veriye uygulanan Frekans, Koşum ve Spektral testlerinin h değerleri sıfırdır. Testlerin sonucunda h=0 olması, A5/1 algoritması ile oluşturulan üreticinin tüm testlerden geçtiğini ve sıfır hipotezinin kabul edildiğini göstermektedir. Bu sonuçlara göre üreticinin birörnek dağılıma sahip rastsal diziler oluşturduğu değerlendirilmektedir.

Çizelge 3.4 A5/1 Algoritması Strateji-1 Test Sonuçları-1

h_frek	p_frek	h_koşum	p_koşum	h_spek	p_spek
0	0,44506075	0	0,39933301	0	0,748072451

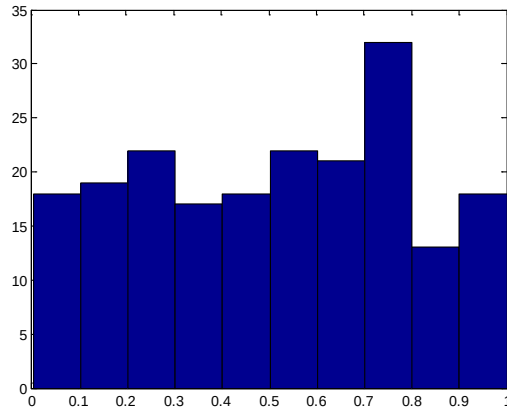
A5/1 algoritmasına ait üretici ile üretilen tüm verinin Spektral testi yapılmış ve bu test sonuçları Şekil 3.6'da verilmiştir. Çizelge 3.4'te yer alan test sonucuna göre üreticinin Spektral testten geçtiği bilinmektedir. Üreticinin tüm verilerine uygulanan Spektral testten geçtiği Şekil 3.6'daki FFT grafiği ile desteklenmektedir.

testleri sonucunda elde edilen P-değerlerine Ki-Kare testi uygulanarak P-değerlerinin dağılımına bakılır. Üretcin 200 örnekleme yapılan testlerden elde edilen P-değerlerine Ki-Kare testi uygulandığında, Çizelge 3.6'da görüldüğü gibi üretçi sadece Spektral testten kalmış, Frekans ve Koşum testlerinden ise geçmiştir.

Çizelge 3.6 A5/1 Algoritması Strateji-2 Test Sonuçları-2

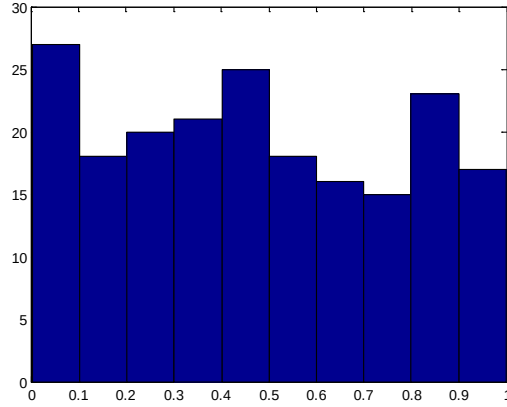
P_Değeri HT-frek	P_Değeri p_değeri-frek	P_Değeri HT-koşum	P_Değeri p_değeri-koşum	P_Değeri HT-spek	P_Değeri p_değeri-spek
0	0,245365694	0	0,66169123	1	< 0,01

Üretcin 200 örneklemin yapılan Frekans testinden elde edilen P-değerleri Ki-Kare testi uygulandığında, Çizelge 3.6'da görüldüğü gibi üretçi Frekans testinden geçmiştir. P-değerleri üzerinde yapılan Ki-Kare testi sonuçları ve P-değerlerinin histogramları birlikte değerlendirildiğinde; Şekil 3.7'de yer alan histogramda görünen iniş-çıkışların; bir örnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturmadığı anlaşılmaktadır.



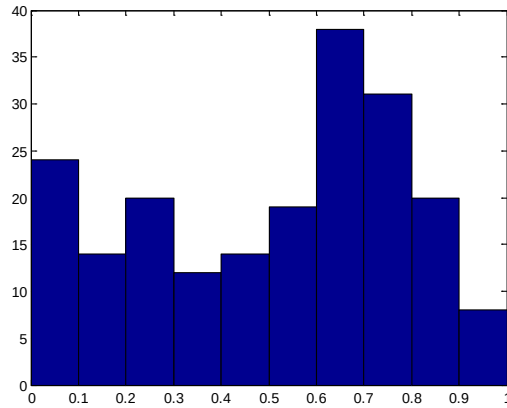
Şekil 3.7 A5/1 Algoritması Strateji-2 Frekans Testi Histogramı

A5/1 algoritmasına ait üretcin 200 örnekleme yapılan Koşum testinden geçtiğini Çizelge 3.6'da gözlemlenmektedir. P-değerleri üzerinde yapılan Ki-Kare testi sonuçları ve P-değerlerinin histogramları birlikte değerlendirildiğinde; Şekil 3.8'de yer alan histogramda gözlenen iniş-çıkışların; bir örnek dağılıma sahip olmadığı söylenemez.



Şekil 3.8 A5/1 Algoritması Strateji-2 Koşum Testi Histogramı

Çizelge 3.6'da yer alan P-değerleri üzerinde yapılan Ki-Kare testi sonuçlarında da görüldüğü gibi üretcin 200 örneklemini Spektral testten kalmıştır. Test sonuçları ve P-değerlerinin histogramları birlikte değerlendirildiğinde; Şekil 3.9'da yer alan histogramda görünen iniş-çıkışların; birörnek dağılıma sahip olmadığı değerlendirilmektedir.



Şekil 3.9 A5/1 Algoritması Strateji-2 Spektral Testi Histogramı

Üretcin 200 örneklemine yapılan Spektral testin sonucu başarısızdır; ancak tüm veriye yapılan Spektral testten başarı ile geçilmiştir. Bu durum üretcin örneklemlerinin bazı testlerden geçemeyebileceğini; ancak üretcin ürettiği tüm veriye aynı test uygulandığında, dizinin o testten başarı ile geçebileceğini göstermektedir. Örneklemlerin, dizinin genel davranışını yansıtmaması beklenir ancak testten kalma oranı ve P-değerleri testi geçmeye oldukça yakın olabilir. Bu da, tüm veriye test yapıldığında başarılı sonuç alınmasına neden olabilir. Bu sonuçlara göre üretcin örneklemlerinin rastsal dizilerden oluştuğu değerlendirilmektedir.

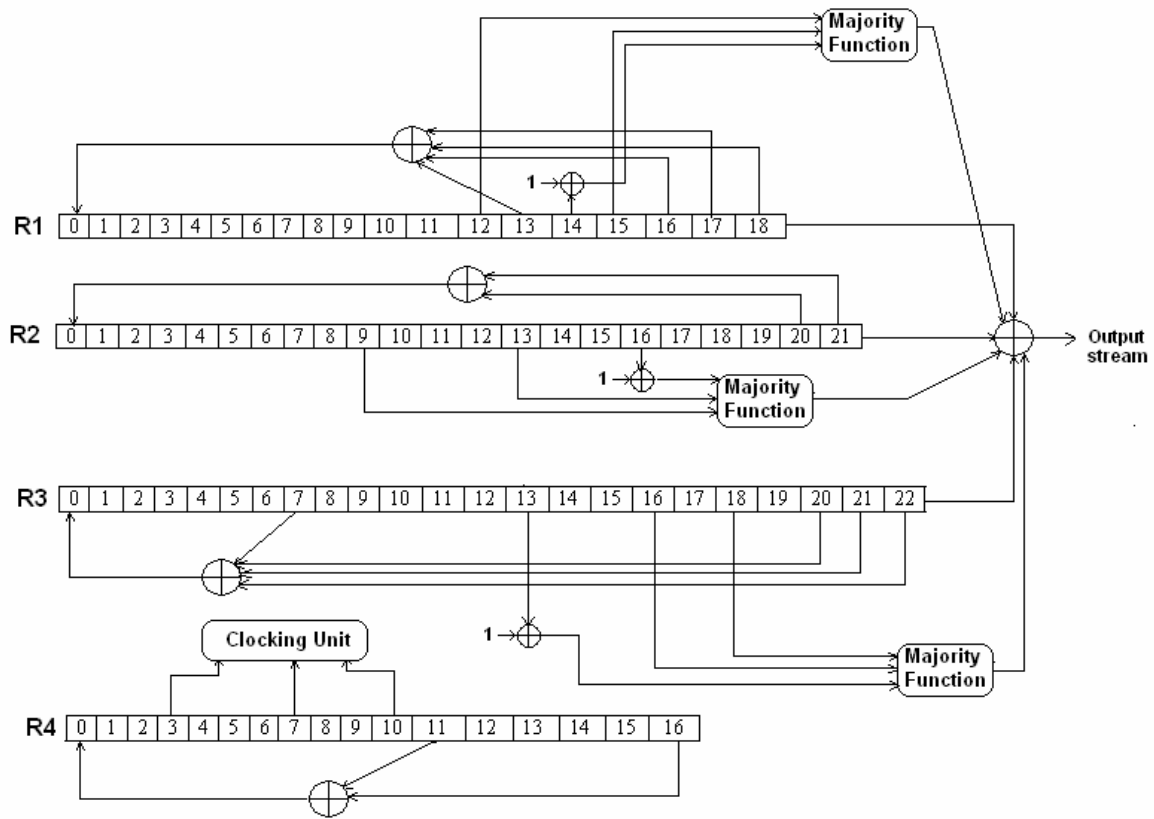
3.2 A5/2 Algoritması

A5/2 şifreleme algoritması, hava kanalı üzerinden ses şifrelemesinde kullanılan A5/1 algoritmasından 1989 yılında türetilmiştir. Soğuk Savaş sırasında şifreleme teknolojisi dışsarı kısıtlamaları yüzünden, A5/1 şifreleme algoritması kasten zayıflatılarak A5/2 algoritması türetilmiştir [Sankaliya, 2011]. A5/1 algoritması ABD ve Avrupa da kullanılırken, A5/2 algoritması zayıf bir versiyon olarak Avrupa dışında kullanılmaktadır. LFSR tabanlı bir algoritma olduğundan donanım uygulamalarında oldukça etkili ve kolay kullanılır.

A5/2 algoritması hava kanalı iletimini şifrelemede kullanılan akış şifreleme yöntemidir. A5/1 şifreleme algoritmasında olduğu gibi A5/2 şifreleme algoritmasında da oturum anahtarı (K_c) ve çerçeve numarası (F_n) ile başlatılır. GSM de her bir konuşma, çerçeveler dizisi halinde 4.6 milisaniyede gönderilir. Her bir çerçeve, mobil cihaz-baz istasyonu iletişimi için 114 bit, baz istasyonu-mobil cihaz iletişimi için 114 bit olmak üzere 228 bitten oluşmaktadır [Ahmed, 2009].

A5/2 akış şifreleme, ikili LFSR tabanında bir anahtar dizisi üreticidir. A5/2 algoritması özellikle tetikleme ve tetikleme kararının alınması durumlarında A5/1 algoritmasından farklılaşmaktadır. A5/2 şifreleme, dört LFSR ve bu LFSR'ların düzensiz olarak tetiklenmesi temeline dayanan bir algoritmadır. A5/2 şifreleme algoritması çıktısı doğrusal olmayan bir yapıdadır, bu nedenle doğrusal yapılara göre algoritmaların karmaşıklığı fazladır; ancak tetikleme mekanizması nedeni ile zayıf bir türevidir.

A5/1 şifreleme algoritmasında yer alan üç LFSR aynı şekilde A5/2 algoritmasında yer alırken, A5/2 algoritmasına özel dördüncü bir LFSR bulunmaktadır. Bu alırtmada dört farklı uzunlukta yazmaç kullanılmakta olup yazmaçlar sırasıyla R1, R2, R3, R4 olmak üzere 19, 22, 23 ve 17 bit uzunluğunda geri beslemeli polinomlar ile ifade edilir. R1'in maksimum periyodu $2^{19} - 1$, R2'nin $2^{22} - 1$, R3'ün $2^{23} - 1$ ve R4'ün maksimum periyodu $2^{17} - 1$ uzunluğundadır. A5/2 şifreleme algoritmasının yapısı Şekil 3.10'da gösterilmektedir.



Şekil 3.10 A5/2 Algoritmasının İç Yapısı

Bu algoritmaya özel olan 17 bit uzunluğa sahip R4 yazmacının kullanılma amacı diğer LFSR'lerden farklıdır. R4 yazmacı R1, R2 ve R3 LFSR'larından hangilerinin tetikleneceğine karar veren bir mekanizmadır. A5/2 algoritması A5/1 şifreleme algoritmasından farklı bir çoğunluk fonksiyonu temeline dayalı olarak, R1, R2 ve R3 yazmaçlarını tetiklemektedir. Üç yazmacın tetiklenme işlemleri çoğunluk fonksiyonu ile olurken, R4 yazmacı sürekli tetiklenmektedir. Her bir çevrimde R4'ün 11. ve 16. bitleri XOR operasyonundan geçirilerek elde edilen sonuç, sola kaydırmalı yazmacın en önemsiz bitinde depolanır. Bu çevrim işlemi R4 yazmacı için sürekli olduğundan R1, R2 ve R3 yazmaçlarının tetiklenme sıraları, R4 yazmacın tetiklenme bitinin sürekli olarak değişmesi ile değişkenlik göstermektedir.

Çizelge 3.7'de belirtildiği gibi R1 yazmacındaki 13., 16., 17. ve 18. bit pozisyonları, R2'deki 20. ve 21. bit pozisyonları, R3'deki 7., 20., 21. ve 22. bit pozisyonları ve R4'deki 11. ve 16. bit pozisyonları önemli bit grupları olup tap olarak isimlendirilir. Başlangıç durumunda her bir yazmacın en önemsiz biti sıfır olarak etiketlenir.

Yazmaçlar her tetiklendiğinde, her bir yazmaca ait tap bitleri kendi aralarında XOR operasyonuna girer ve sonuç en önemsiz bitte depolanır.

Çizelge 3.7 A5/2 Algoritmasının Matematiksel Yapısı

LFSR No	LFSR Uzunlukları	Geri Beslemeli Polinomlar	Tetikleme Tap Biti	Tap Biti
R1	19 bit	$x^{19} + x^{18} + x^{17} + x^{14} + 1$	8	13, 16, 17, 18
R2	22 bit	$x^{22} + x^{21} + 1$	10	20, 21
R3	23 bit	$x^{23} + x^{22} + x^{21} + x^8 + 1$	10	7, 20, 21, 22
R4	17 bit	$x^{17} + x^{12} + 1$	3, 7, 10	11,16

Her bir yazmaçta tek bir tetikleme tap biti vardır. R1 için 8. bit, R2 için 10. bit ve R3 için 10. bit tetikleme tap bitleridir. R4 yazmacının 3., 7. ve 10. bitleri tetikleme biti olup bu bitler çoğunluk fonksiyona girdi olur; fonksiyon sonucu olan çoğunluk bitine göre R1, R2 ve R3 yazmaçları tetiklenir. Çoğunluk kuralının A5/2 şifreleme algoritmasında nasıl işlediği aşağıda belirtilmiştir [Barkan, 2003].

$$m = \text{Majority}(R4[3], R4[7], R4[10]) \rightarrow \text{maj}(a,b,c) = a \cdot b \oplus b \cdot c \oplus c \cdot a \quad (3.12)$$

$$\bullet \quad R4[10] = m \quad \text{ise} \quad R1 \text{ tetiklenir} \quad (3.13)$$

$$\bullet \quad R4[3] = m \quad \text{ise} \quad R2 \text{ tetiklenir} \quad (3.14)$$

$$\bullet \quad R4[7] = m \quad \text{ise} \quad R3 \text{ tetiklenir} \quad (3.15)$$

A5/1 algoritmasına göre A5/2 şifreleme algoritmasında çıktı alma yöntemi farklılıklar göstermektedir. A5/1 algoritmasında sola kaydırma işlemi sonucunda R1, R2 ve R3 yazmaçlarının en önemli bitleri XOR operasyonuna girdi olurken, A5/2 algoritmasında buna ek olarak Şekil 3.10'da görüldüğü gibi R1, R2 ve R3 yazmaçların çoğunluk fonksiyonu çıktıları da XOR operasyonuna girdi olmaktadır.

3.2.1 A5 Algoritmasının Zayıf Yönleri

A5/2 algoritması birçok teknik kullanılarak kırılmıştır. A5/2 algoritmalarına yapılan saldırıların çoğu tetikleme mekanizmalarındaki zayıflıklardan kaynaklanmaktadır. A5/2 algoritmasında tetikleme kontrolünü yapan R4 yazmacın bilinen bir çıktı biti için başlangıç durumu belirlenirse yazmacın ürettiği çıktı biti bilinebilir.

Saldırıların çoğu bilinen metin saldırılarıdır. Biryukov, Shamir ve Wagner tarafından yapılan iki tip saldırıda birkaç dakika veri alınması ile saniyelerce saldırı düzenlenmiştir. Bu saldırılar sonucunda algoritmaların temelini oluşturan bir öz alma fonksiyonu olan COMP128 algoritması ile A5/2 tamamen kırılmıştır. A5/2'nin tahmin edilen zaman karmaşıklığı 2^{17} civarındadır [Petrovic, 2002].

2006 yılında bilinen metin saldırıları ile A5/1 ve A5/2 algoritmalarının telefon görüşmeleri sırasında yani gerçek zaman veya sonraki herhangi bir zamanda kırılabileceği Barkan, Biham ve Keller tarafından gösterilmiştir. Algoritmaların zayıflığından ötürü oturma anahtarı elde edilebilmiştir [Barkan, 2003].

1 Temmuz 2006 tarihinden bu yana GSMA (GSM Birliği) A5/2 algoritmasını zayıflığından dolayı daha fazla desteklemeyeceğini ve A5/1 algoritmasının 3GPP Birliği tarafından zorunlu olarak kabul edilebilir olması gerektiğini bildirdi. Temmuz 2007 tarihinde 3GPP Birliği yeni üretilen mobil telefonda A5/2 algoritmasının uygulanmasını yasaklayan değişiklik isteğini onayladı.

3.2.2 A5/2 Algoritmasının Uygulanması

A5/2 algoritması hakkında yukarıdaki bölümde genel olarak bilgi verilmiştir. A5/2 algoritmasının kodlanması, MATLAB programlama dili ile gerçekleştirilmiştir. A5/2 algoritmasının kodlanma sırasında dikkat edilen algoritma kıstasları ve algoritma adımları aşağıda belirtilmiştir.

1. R1, R2, R3 ve R4 yazmaçların her bir biti sıfırlanır.

$$R1 = R2 = R3 = R4 = 0 \quad (3.16)$$

Tetikleme kontrolü yapılmadan yazmaçlar tetiklenir.

For i = 0 to 63 do

$$R1[0] = R1[0] \oplus K_c[i] \quad (3.6)$$

$$R2[0] = R2[0] \oplus K_c[i] \quad (3.7)$$

$$R3[0] = R3[0] \oplus K_c[i] \quad (3.8)$$

$$R4[0] = R4[0] \oplus K_c[i] \quad (3.17)$$

K_c anahtarının her bir biti ile her bir yazmacın en önemsiz bitine XOR işlemi uygulanır, tetikleme kontrolü uygulanmadan yazmaçlar tetiklenir. K_c anahtarı 64 bitten oluştuğundan bu işlem 64 çevrimde tamamlanır. XOR operasyonu K_c anahtarının en önemsiz bitten (LSB) en önemli bitine (MSB) doğru yapılır.

2. 1. Adımda K_c anahtarı ile yapılan çevrim 2. adımda 22 bitlik F_n için 22 çevrimde tamamlanır.

For i = 0 to 21 do

$$R1[0] = R1[0] \oplus \text{COUNT}[i] \quad (3.9)$$

$$R2[0] = R2[0] \oplus \text{COUNT}[i] \quad (3.10)$$

$$R3[0] = R3[0] \oplus \text{COUNT}[i] \quad (3.11)$$

$$R4[0] = R4[0] \oplus \text{COUNT}[i] \quad (3.18)$$

Çerçeve numarasının her bir biti ile her bir yazmacın en önemsiz bitine XOR işlemi uygulanır, tetikleme kontrolü uygulanmadan yazmaçlar tetiklenir. F_n çerçeve numarası 22 bitten oluştuğundan bu işlem 22 çevrimde tamamlanır. XOR operasyonu F_n çerçeve numarasının en önemsiz bitten (LSB) en önemli bitine (MSB) doğru yapılır.

3. $R1[14] = R2[16] = R3[13] = R4[10] = 1$ (3.19)
atanır.

4. Her yazmaç, çoğunluk kuralı ile tetik kontrolü yapılarak 100 çevrim için tetiklenir. Elde edilen 100 bitlik çıktı kullanılmaz.

5. Üç yazmaç, R4 yazmacına bağlı tetik kontrolü yapılarak 228 çevrim için tetiklenir. R4 yazmacı sürekli tetiklenir. Her bir çevrimde üç yazmacın en önemli bitleri ve üç yazmacın çoğunluk fonksiyonu çıktıları, XOR operasyonundan geçirilerek tek bir çıkış biti üretilir. Çoğunluk fonksiyonu çıktıları aşağıdaki belirtildiği gibi üretilir.

R1 yazmacı çoğunluk fonksiyonu çıktısı;

$$\text{maj} = \text{majority}(R1(12,1), \text{not}(R1(14,1)), R1(15,1)) \quad (3.20)$$

R2 yazmacı çoğunluk fonksiyonu çıktısı;

$$\text{maj} = \text{majority}(R2(9,1), R2(13,1), \text{not}(R2(16,1))) \quad (3.21)$$

R3 yazmacı çoğunluk fonksiyonu çıktısı;

$$\text{maj} = \text{majority}(\text{not}(R3(13,1)), R3(16,1), R3(18,1)) \quad (3.22)$$

6. Her çevrimde çıkan bir bitlik çıktı 228 bitlik arabelleğe yazılır; 228 çevrim sonunda 228 bitlik A5/2 algoritma çıktısı elde edilir. 228 bitlik verinin; 114 biti mobil istasyondan baz istasyonuna, diğer 114 biti ise baz istasyonundan mobil istasyona gönderimde kullanılır.
7. Testler iki stratejiye ayrılarak yapılır. Strateji-1 de üreteç tarafından üretilen tüm veriye (1000000 uzunluğundaki veri) testler yapılırken; Strateji-2 de ise bölünmüş veriye yani 5000 örneklem dizi uzunluğuna sahip 200 örnekleme test yapılır.

3.2.3 Uygulanan Testlerin Sonuçları ve Değerlendirmeler

Tüm üreteçlere

- Frekans Testi,
- Koşum Testi,
- Ayırık Fourier Dönüşümü (Spektral) Testi

yapılarak elde edilen test sonuçları çizelge ve grafik halinde bu bölümde sunulmuş ve test sonuçlar yorumlanmıştır.

Bu testlerin sıfır hipotezini gösteren h ve P - değerleri incelenir. $h = 0$ ise sıfır hipotezinin kabul edildiğini gösterir. Yapılan testler anlamlılık düzeyi α olarak %1 seçilmiştir. Bu nedenle P -değeri ≥ 0.01 ise üretilen dizi rastsaldır.

Algoritma girdisi olarak [Briceno, 1990];

- Anahtar (K_c): 0x00, 0xFC, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
- Çerçeve numarası (F_n): 0x21

Algoritma çıktısı olarak [Briceno, 1990];

- Anahtar dizisi: 0xF4, 0x51, 0x2C, 0xAC, 0x13, 0x59, 0x37, 0x64, 0x46, 0x0B, 0x72, 0x2D, 0xAD, 0xD5, 0x00

alınmıştır.

Test girdileri olarak;

- Algoritma çıktısı
- Üretilen dizi uzunluğu : 1000000
- Örneklemdaki dizi uzunluğu : 5000
- Örneklem sayısı : 200
- $\alpha = 0.01$ değerleri seçilmiştir.

A5/2 algoritmasının çıktıları testlerden geçirilmiştir. A5/1 algoritması üretici için test sonuçları aşağıdaki çizelge ve grafikler ile verilmiştir.

3.2.3.1 Strateji-1 Testleri

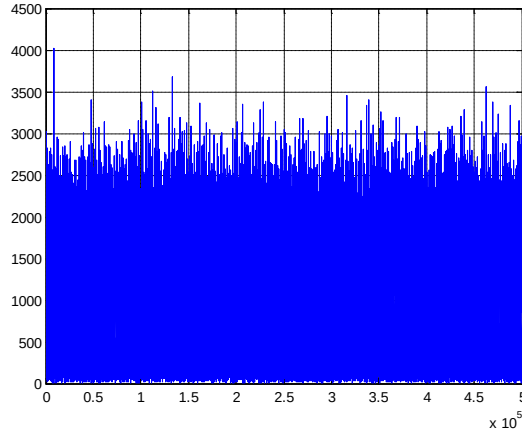
Seçilen üretici ile üretilen 1000000 dizi uzunluğundaki veriye Frekans, Koşum ve Spektral testleri yapılmıştır.

Tüm testlerden elde edilen h ve p değerleri incelenmiştir. Anlam düzeyi (α) %1 seçildiği durumda Çizelge 3.8'de görüldüğü gibi seçilen üretici için tüm veriye uygulanan Frekans, Koşum ve Spektral testlerinin h değerleri sıfırdır. Testlerin sonucunda h=0 olması, sıfır hipotezinin kabul edildiğini göstermektedir. Buna göre A5/2 algoritması ile oluşturulan üreticinin tüm testlerden geçtiği gözlenmiştir. Bu sonuçlara göre üreticinin birörnek dağılıma sahip rastsal diziler oluşturduğu değerlendirilmektedir.

Çizelge 3.8 A5/2 Algoritması Strateji-1 Test Sonuçları-1

h_fre	p_fre	h_koşum	p_koşum	h_spek	p_spek
0	0,843053	0	0,329384	0	0,408863

A5/2 algoritmasına ait üretici ile üretilen tüm verinin Spektral testleri yapılmış ve bu test sonuçları Çizelge 3.8’de verilmiştir. Bu sonuca göre üreticinin Spektral testten geçtiği bilinmektedir. Üreticinin tüm verilerine uygulanan Spektral testten geçtiğini Şekil 3.11’deki FFT grafiği de doğrulamaktadır.



Şekil 3.11 A5/2 Algoritması Strateji-1 FFT Grafiği

3.2.3.2 Strateji-2 Testleri

Örnekleme sayısı ve bir örneklemdaki dizi uzunluğu seçimi ile algoritma tarafından üretilen dizinin tümü kullanılarak testler gerçekleştirilmiştir. Bu kısımda üretilen tüm veri, örnekleme sayısı ve bir örneklemdaki dizi uzunluğu göz önüne alınarak bölünmüştür. Bölünen veriler üzerinde Frekans, Koşum, Spektral ve Ki-Kare testleri yapılmıştır.

A5/2 algoritmasına ait üreticinin 200 örneklemine yapılan testler sonucunda elde edilen P-değerine analiz yapılır. P-değeri analizi kapsamında dizi geçme oranı hesaplanır. 200 örnekleme arasında kaçının sıfır hipotezini geçtiğine dair bir oran elde edilir ve sonuç olarak testten geçip geçmediğine bakılır. Test sonuçları yer Çizelge 3.9’da almaktadır.

Çizelge 3.9 A5/2 Algoritması Strateji-2 Test Sonuçları-1

Frekans DGO	Frekans Testi Sonucu	Koşum DGO	Koşum Testi Sonucu	Spektral DGO	Spektral Test Sonucu
0,985	1	0,965	0	0,99	1

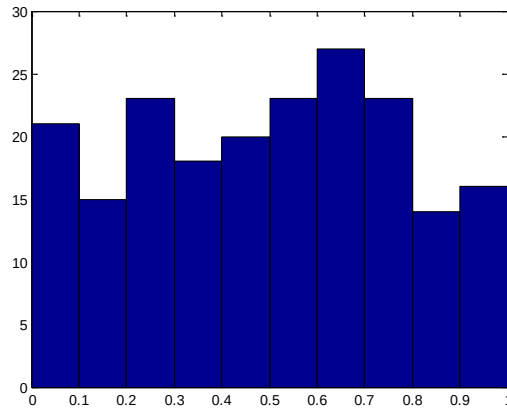
Çizelge 3.9’da görüldüğü gibi farklı oranlarla da olsa test sonuçları 1 olduğundan, üretcin 200 örnekleme uygulanan Frekans ve Spektral testlerin geçtiği, Koşum testinden ise kaldığı gözlenmektedir.

Tüm testler için P-değeri analizi kapsamında P-değerinin dağılımı hesaplanır. A5/2 algoritmasına ait üretcin 200 örnekleme uygulanan testlerin sonucunda elde edilen P-değerlerine Ki-Kare testi uygulanarak P-değerlerinin dağılımına bakılır.

Çizelge 3.10 A5/2 Algoritması Strateji-2 Test Sonuçları-2

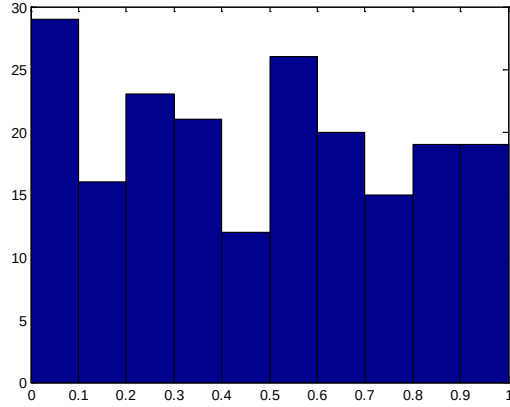
P_Değeri HT-frek	P_Değeri p_değeri-frek	P_Değeri HT-koşum	P_Değeri p_değeri-koşum	P_Değeri HT-spek	P_Değeri p_değeri-spek
0	0,544741	0	0,237367	1	< 0,01

Üretcin örneklemlerine testler yapıldığında, Çizelge 3.10’da görüldüğü gibi üretçi sadece Spektral testten kalmıştır. Frekans ve Koşum testlerinden başarı ile geçmiştir.



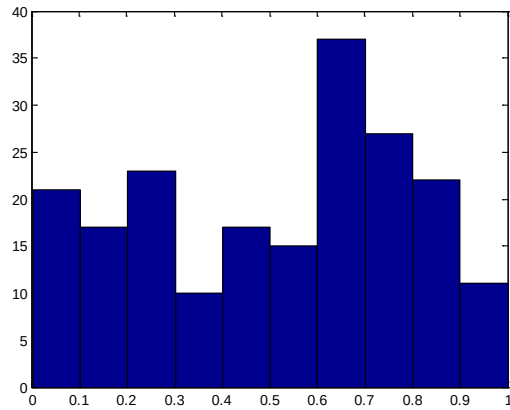
Şekil 3.12 A5/2 Algoritması Strateji-2 Frekans Testi Histogramı

P-değerleri üzerinde yapılan Ki-Kare testi sonuçları ve P-değerlerinin histogramları birlikte değerlendirildiğinde; Şekil 3.12 ve Şekil 3.13’te yer alan histogramda görünen iniş-çıkışların; birörnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturmadığı anlaşılmaktadır.



Şekil 3.13 A5/2 Algoritması Strateji-2 Koşum Testi Histogramı

Ancak Spektral testten kaldığını bildiğimiz üretcin Şekil 3.14'te yer alan histogramda gözlenen iniş-çıkışların; birörnek dağılıma sahip olmadığına dair göstergeler oluşturmaktadır.



Şekil 3.14 A5/2 Algoritması Strateji-2 Spektral Testi Histogramı

A5/2 algoritmasına ait üretcin 200 örnekleme yapılan Spektral testin sonucu başarısızdır; ancak tüm veriye yapılan Spektral testten başarı ile geçilmiştir. Bu durum örneklemelerin bazı testlerden geçemeyebileceği ancak tüm veriye aynı test uygulandığında üretcin o testten başarı ile geçebileceğini göstermektedir.

Bu sonuçlara göre üretcin örneklemelerinin rastsal dizilerden oluştuğu değerlendirilmektedir.

A5/1 ve A5/2 algoritmasına uygulanan istatistiksel testlerin sonucunda görüldüğü gibi benzer istatistiksel özelliklere sahip algoritmalar. A5/1 algoritmasından türetildiği için beklenildiği gibi benzer istatistiksel özellikler sergilemektedir. A5/2

algoritmasının A5/1 algoritmasına göre güçsüzlüğü yapılan testler ile gözlemlenmemiştir. Kriptografik uygulamalarda zayıf bir algoritma olduğunu farklı testler ve saldırı yöntemleri ile gözlenebileceği değerlendirilmektedir.

3.3 RC4 ALGORİTMASI

RC4, RSA Security kurucularından Ron Rivest tarafından 1987 yılında geliştirilen bir simetrik akış şifreleme metodudur. RC4'ün birden fazla açılımı mevcuttur; Ron's Code 4, Rivest Cipher 4. RC4 şifreleme algoritması başlangıçta bir sırđı; ancak 1994 Eylül ayında bu algoritmanın kaynak kodları internet ortamında yayınlanmıştır. Buna rağmen RC4 ticari bir marka haline getirildi ve ARCFOUR ve ARC4 gibi isimler takıldı.

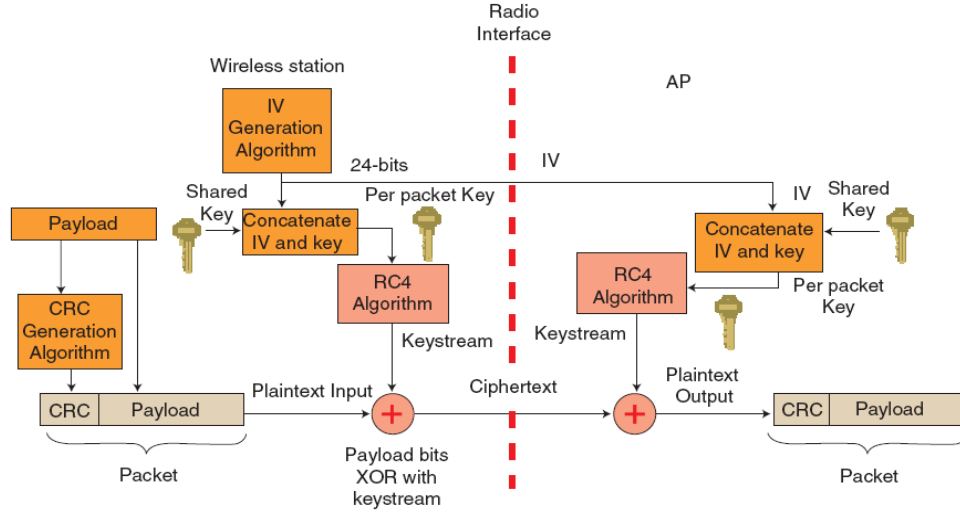
Birçok akış şifreleme algoritması, yazılımdan çok donanımda uygulanabilirliği kolay olması için yazmaç tabanlı tasarlanmıştır. RC4 algoritması ise yazılım uygulamalarında kullanılması için ideal bir algoritmadır, işlemler bayt'lar üzerinden yapılır. Durum dizilerinin tutulması için (S[0] ...S[255]) 256 baytlık bir hafıza kullanılır. Hafızanın k baytı (anahtar[0] ...anahtar[k-1]) anahtar ve tam sayı değişkenleri (i,j) için ayrılır.

RC4 şifreleme algoritmasının kullanıldığı bazı uygulamalar aşağıda belirtilmiştir:

- WEP (kablosuz ağ güvenliği, 802.11b)
- WPA
- Bit Torrent Protokol Şifreleme
- Microsoft Point-to-Point Şifreleme
- SSL/TLS (İnternet trafiğini korumayı amaçlar)
- SSH
- Uzaktan Masaüstü Erişim Protokolü
- Kerberos
- Oracle Secure SQL,
- Lotus Notes,
- Apple – AOCE, RDC, RDP, SASL

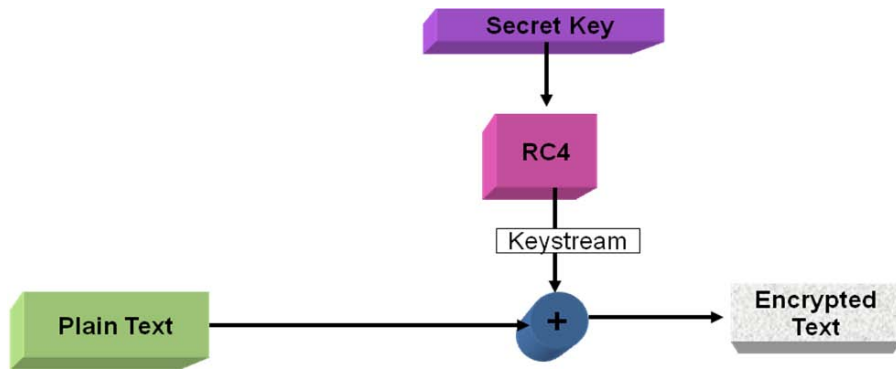
RC4 algoritması doğrusal olmayan çıktılar ürettiğinden kırılması güçtür. Aktif saldırılara karşı dayanıklıdır; ancak aktif olmayan bir saldırı ile kırılabilir. RC4 algoritmasının durum tablosu saldırılara karşı savunmasız değildir. Ancak anahtarın tek kullanımlık olması gerekmektedir. Güçsüz ya da kısa bir anahtar kullanımında algoritmanın kırılması kolaylaşır. Güçsüz anahtardan kasıt, seçilen

anahtarlar arasında bir ilintinin var olmasıdır. Bazı RC4 uygulamalarında 24 bitlik başlangıç vektörleri (IV) kullanılır. Örneğin Şekil 3.15'te görüldüğü gibi RC4 algoritmasının WEP uygulamasında, 24-bit IV ve 40-104-bit uzunluğunda anahtar kullanılarak anahtar dizisi oluşturulur. Bu gibi uygulamalarda da, aynı anahtar veya aynı başlangıç vektörü ikinci kez kullanılırsa, RC4 algoritması savunmasız kalır.



Şekil 3.15 RC4 Algoritmasının WEP Uygulaması

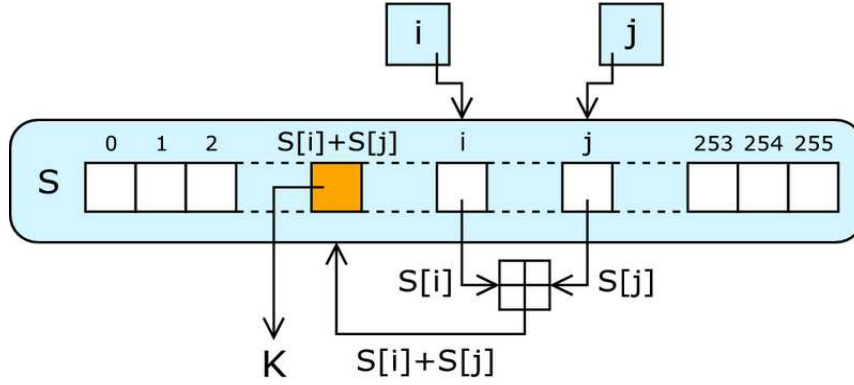
Buna ek olarak RC4 algoritmasının hızlı ve kolay uygulanabilir olmasını sağlayan XOR işlemlerinin yapısı, algoritmayı kolaylaştırdığı gibi saldırıları da kolaylaştırmakta ve algoritmanın dayanıklılığını düşürmektedir.



Şekil 3.16 RC4 Algoritması ile Şifreleme

Şekil 3.16'da görüldüğü gibi gizli anahtar RC4 algoritmasına girdi olur ve RC4 algoritması ile anahtar dizisi üretilir. Anahtar dizisi ile düz metin bayt bayt XOR işleminden geçirilir; şifreli metin oluşturulur. Şifreli metin güvenli olmayan bir kanal

ile alıcı tarafa ulaştırılır. Aşağıdaki şekilde S dizisi için RC4 algoritmasının nasıl çalıştığı gösterilmiştir.



Şekil 3.17 RC4 Algoritması S Dizisi

RC4 algoritmasının çıktısı olan anahtar dizisinin üretilmesi iki safhadan oluşur.

1) Anahtar Çizelgesi Algoritması (KSA)

40 ile 2048 bit arasında bir uzunluğa sahip olan anahtarı, bir başlangıç S permütasyonuna dönüştürür. Bu kapsamda KSA, durum ve anahtar dizilerinin yaratılması işlemlerini içerir; bu diziler için iki tane 256 bayt uzunluğunda dizi kullanılır. 256 olası tüm baytlar için permütasyon işlemi yapılır. Permütasyon değişken anahtar uzunluğunun bir fonksiyonudur. Permütasyon işlemi (S) 40 ile 256 arasında değişken bir sayıdaki anahtar ile ilklendirilir. İşlemler sırasında kullanılan 8 bitlik indeks işaretleyicisi, sayıcısı i ve j vardır.

Dizi Başlangıç Atamaları ile ilgili algoritma adımları aşağıda belirtilmiştir.

- $S[0], S[1], S[2], \dots, S[255] \rightarrow$ S durum dizisi,
- $T[0], T[1], T[2], \dots, T[255] \rightarrow$ T geçici vektör,
- K : 1 ile 256 bayt arasında değişken uzunluğuna sahip gizli anahtar

olarak tanımlanır.

2) Sözde Rastsal Üreteç Algoritması (PRGA)

KSA aşamasında üretilen permütasyonu kullanarak bir sözde rastsal dizi üretimini gerçekleştirir. Bu sözde rastsal dizi, anahtar dizisini oluşturmaktadır ve anahtar dizisi ile düz metnin XOR işleminden geçirilerek şifreli metin elde edilir.

3.3.1 RC4 Algoritmasının Uygulanması

RC4 algoritması hakkında yukarıdaki bölümde genel olarak bilgi verilmiştir. RC4 algoritmasının kodlanması, programlama dili olan MATLAB ile gerçekleştirilmiştir. RC4 algoritmasının kodlanma sırasında dikkat edilen algoritma kıstasları ve algoritma adımları aşağıda belirtilmiştir.

1. KSA, başlangıç bölümünde tüm muhtemel kelime değerleri dizisini oluşturmaya başlar. Şifreleme bölümünde bu dizinin içeriği rastsal sıralanmış değerler ile değiştirilir. Her adımda dizinin i indeksi bir artırılır ve her adımda j indeksi hesaplanır ve burada KSA'ya giriş olan K anahtarı kullanılır.

Başlangıç:

$S[i] \leftarrow i, 0 \leq i \leq 255$, durum dizi ataması
 $T[i] \leftarrow K[i \bmod \text{anahtar uzunluğu}], 0 \leq i \leq 255$
 $T[0 \dots 255]$ dizisi anahtar K ile doldurularak anahtar dizi atanmış olur.
 $j \leftarrow 0$

Şifreleme:

for $i \leftarrow 0$ to 255 do
 $j \leftarrow (j + S[i] + T[i]) \bmod 256$ (3.23)
swap $S[i], S[j]$

2. Bu kısımda birçok iterasyon sunucu, PRGA durum ve anahtar dizisinin bayt halindeki çıktıların oluşturulması sağlanır.

Başlangıç:

$i = 0$
 $j = 0$

Üretme döngüsü:

$i \leftarrow (i + 1) \bmod 256$
swap $S[i], S[j]$ $S[i]$ 'nin değeri $S[j]$ değerine, $S[j]$ 'nin değeri $S[i]$ değerine aktarılır.

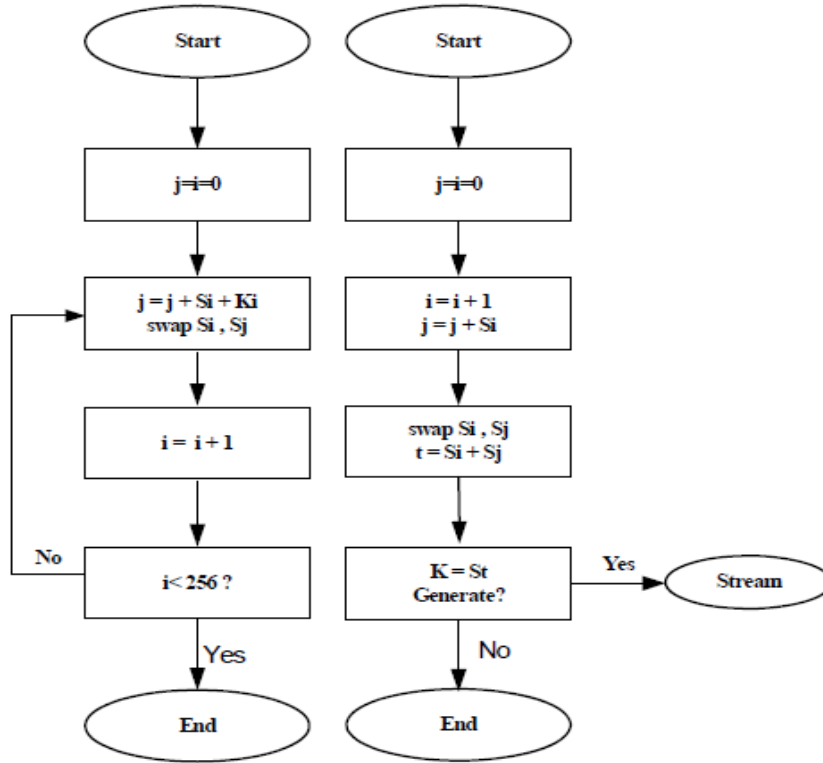
$$\text{temp} \leftarrow (j + S[i]) \bmod 256 \quad (3.24)$$

$$k \leftarrow S[\text{temp}]$$

$C_i = M_i \oplus S[k]$ düz metin ile anahtar dizisi XOR işleminden geçirilerek şifreli metin elde edilir.

Üretme döngüsü, veriyi şifrelemede kullanılan n bitlik rastsal çıktıyı (anahtar dizisi) üretmek için kullanılır. Her bir döngüde i indeksi bir artırılır ve N=256 adımda dizi taranır; her bir eleman en az bir kez takas edilir [Gürkaş, 2005].

Her yeni anahtarda bu iki safha tekrarlanır. Bu safhaların akış diagramları de gösterilmiştir.



Şekil 3.18 RC4 Akış Diagramı

- Testler iki stratejiye ayrılarak yapılır. Strateji-1 de üretç tarafından üretilen tüm veriye (1000000 uzunluğundaki veri) testler yapılırken; Strateji-2 de ise bölünmüş veriye yani 5000 örneklem dizi uzunluğuna sahip 200 örnekleme test yapılır.

3.3.2 Uygulanan Testlerin Sonuçları ve Değerlendirmeler

Tüm üreteçlere

- Frekans Testi,
- Koşum Testi,
- Ayırık Fourier Dönüşümü (Spektral) Testi

yapılarak elde edilen test sonuçları çizelge ve grafik halinde bu bölümde sunulmuş ve test sonuçlar yorumlanmıştır.

Bu testlerin sıfır hipotezini gösteren h ve P -değerleri incelenir. $h = 0$ ise sıfır hipotezinin kabul edildiğini gösterir. Yapılan testler anlamlılık düzeyi α olarak %1 seçilmiştir. Bu nedenle P -değeri ≥ 0.01 ise üretilen dizi raststaldır.

RC4 algoritmasında sadece anahtarlar değiştirilerek üç üreteç oluşturulmuştur. Anahtarları farklılaştırmanın, başlangıç değerlerinin seçilmesinde bir hassasiyete neden olup olmadığı ve test sonuçlarına bir etkisi olup olmadığı incelenmiştir.

Algoritma girdisi olarak [Wikipedia, 2011];

- Üreteç-1'in Anahtarı : "Wiki"
- Üreteç-2'in Anahtarı : "Key"
- Üreteç-3'in Anahtarı: "Secret"

Algoritma çıktısı olarak [Briceno, 1990];

- Anahtar dizisi-1: "6044db6d41b7"
- Anahtar dizisi-2: "eb9f7781b734ca72a719"
- Anahtar dizisi-3: "04d46b053ca87b59"

alınmıştır.

Test girdileri olarak;

- Algoritma çıktısı
- Üretilen dizi uzunluğu : 1000000
- Örneklemdeki dizi uzunluğu : 5000

- Örneklem sayısı : 200
- $\alpha = 0.01$ değerleri seçilmiştir.

RC4 algoritmasının ait üç üretelin çıktılarına belirtilen testler uygulanmıştır. RC4 algoritması üreteli için test sonuçları çizelge ve grafikler ile sunulmuştur.

3.3.2.1 Strateji-1 Testleri

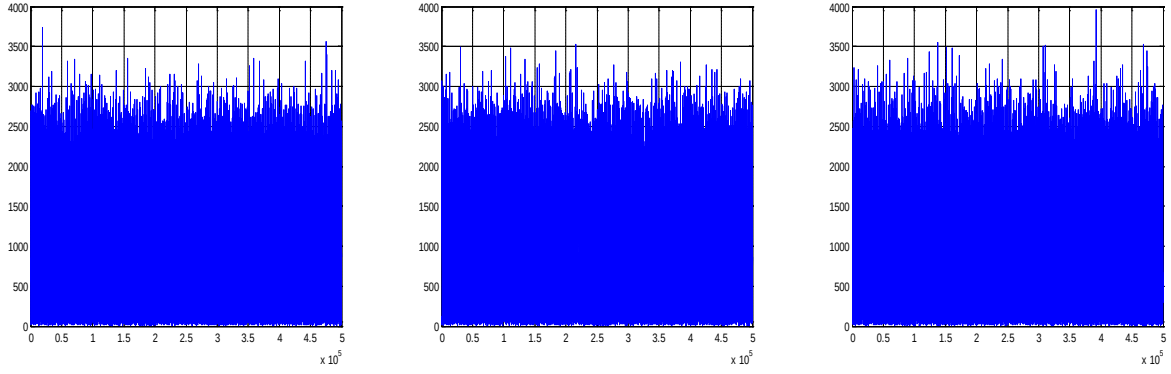
Seçilen üreteli ile üretilen 1000000 dizi uzunluğundaki veriye Frekans, Koşum ve Spektral testleri yapılmıştır.

Üç üreteli için tüm testlerden elde edilen h ve p değerleri incelenmiştir. Anlam düzeyi (α) %1 seçildiği durumda Çizelge 3.11'de görüldüğü gibi seçilen üreteli için tüm veriye uygulanan Frekans, Koşum ve Spektral testlerinin h değerleri sıfırdır. Testlerin sonucunda h=0 olması, sıfır hipotezinin kabul edildiğini göstermektedir. Buna göre RC4 algoritması ile oluşturulan üç üretelinde tüm testlerden geçtiği gözlenmiştir. Bu sonuçlara göre üretelin birörnek dağılıma sahip rastsal diziler oluşturduğu değerlendirilmektedir.

Çizelge 3.11 RC4 Algoritması Strateji-1 Test Sonuçları-1

Üretilen	h_fre	p_fre	h_koşum	p_koşum	h_spek	p_spek
1	0	0,777297	0	0,358509	0	0,941477
2	0	0,079495	0	0,141349	0	0,39853
3	0	0,257899	0	0,244409	0	0,992678

RC4 algoritmasına ait üç üreteli ile üretilen tüm verinin Çizelge 3.11'deki Spektral test sonuçlarına göre, üç üretele ait tüm verinin Spektral testten geçtiği bilinmektedir. Her bir üretelin tüm verilerine uygulanan Spektral testten geçtiği; Şekil 3.19'da sırası ile yer alan üreteli-1, üreteli-2 ve üreteli-3'ün FFT grafiği ile desteklenmektedir.



Şekil 3.19 Sırasıyla 3 Üretecin FFT Grafiği

3.3.2.2 Strateji-2 Testleri

Örnekleme sayısı ve bir örneklemdaki dizi uzunluğu seçimi ile algoritma tarafından üretilen dizinin tümü kullanılarak testler gerçekleştirilmiştir. Bu kısımda üretilen tüm veri, örnekleme sayısı ve bir örneklemdaki dizi uzunluğu göz önüne alınarak bölünmüştür. Bölünen veriler üzerinde Frekans, Koşum, Spektral ve Ki-Kare testleri yapılmıştır.

RC4 algoritmasına ait üreteçlerin 200 örneklemine yapılan testler sonucunda elde edilen P-değerine analiz yapılır. P-değeri analizi kapsamında dizi geçme oranı (DGO) hesaplanır. 200 örnekleme arasında kaçının sıfır hipotezini geçtiğine dair bir oran elde edilir ve sonuç olarak testten geçip geçmediğine bakılır. Çizelge 3.12’de görüldüğü gibi farklı oranlarla da olsa test sonuçları 1 olduğundan, üç üretecin 200 örneklemine uygulanan tüm testlerden geçtiği gözlenmektedir.

Çizelge 3.12 RC4 Algoritması Strateji-2 Test Sonuçları-1

Üreteç	Frekans DGO	Frekans Testi Sonucu	Koşum DGO	Koşum Testi Sonucu	Spektral DGO	Spektral Test Sonucu
1	0,99	1	0,995	1	0,99	1
2	0,995	1	1	1	0,985	1
3	0,985	1	0,995	1	0,99	1

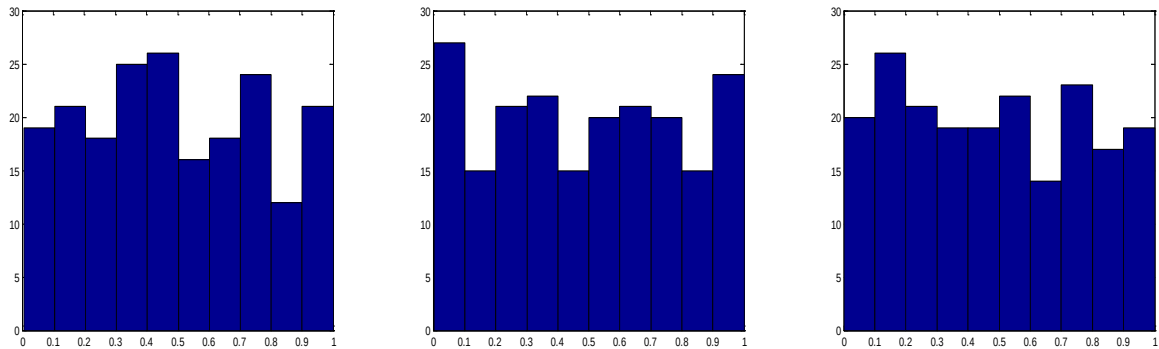
Tüm testler için P-değeri analizi kapsamında P-değerinin dağılımı hesaplanır. RC4 algoritmasına ait üretecin 200 örneklemine uygulanan testlerin sonucunda, elde edilen P-değerlerine Ki-Kare testi uygulanarak P-değerlerinin dağılımına bakılır.

Çizelge 3.13 RC4 Algoritması Strateji-2 Test Sonuçları-2

Üreteç	P_Değeri HT-frek	P_Değeri p_değer-frek	P_Değeri HT-koşum	P_Değeri p_değeri - koşum	P_Değeri HT-spek	P_Değeri p_değeri -spek
1	0	0,478319	0	0,277032	1	< 0,01
2	0	0,651831	0	0,393674	1	< 0,01
3	0	0,842878	0	0,512315	1	< 0,01

RC4 algoritmasına ait üç üretcin örneklemelerine testler yapıldığında, Çizelge 3.13'de görüldüğü gibi üç üretilen aynı testten yani Spektral testten kalmıştır. Frekans ve Koşum testlerinden ise üç üretilen başarı ile geçmiştir. Bu test sonuçları esas alınarak üç üretcin 200 örneklemin Frekans test sonuçlarının P-değerleri dağılımına Şekil 3.20'de, Koşum testi sonuçlarına Şekil 3.21'de ve Spektral test sonuçlarına Şekil 3.22'de görüldüğü gibi histogram ile de bakılmıştır.

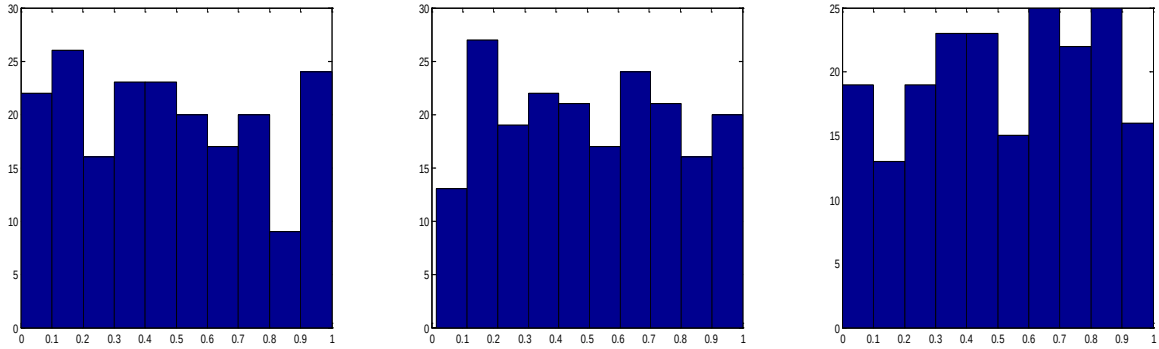
Üç üretcin 200 örnekleminin Frekans testinin, Şekil 3.20'de sırası ile yer alan üretci-1, üretci-2 ve üretci-3'ün histogramlarında iniş-çıkışlar gözlenmektedir. Çizelge 3.13'teki sonuçlar ve histogramlar birlikte incelendiğinde; üretcin örnekleminin P-değerlerinin birörnek dağılıma sahip olmadığına dair bir gösterge oluşturmamaktadır.



Şekil 3.20 RC4 Algoritması Strateji-2 Frekans Testi Histogramı

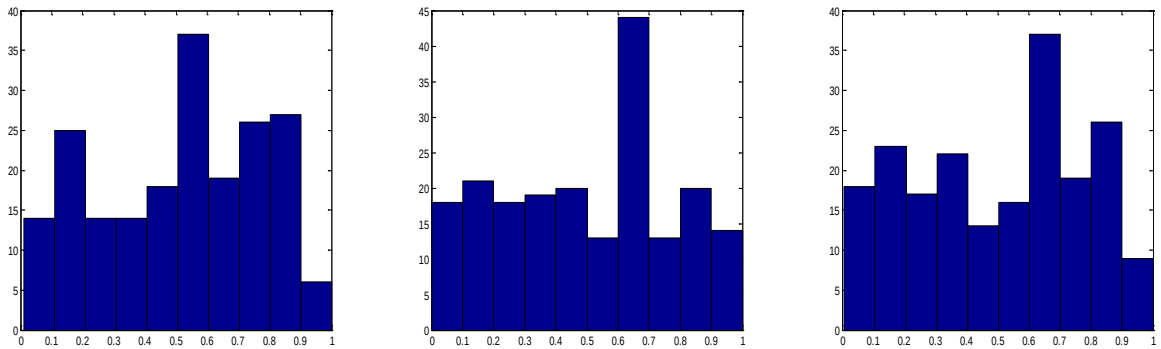
Üç üretcin 200 örneklemelerine yapılan Koşum testinin histogram grafiklerinde 1. ve 3. üretcin, inişli-çıkışlı davranışlar sergilediği gözlenmektedir. Çizelge 3.13'teki sonuçlara göre üç üretcin 200 örnekleminin Koşum testinden geçtiği bilinmektedir. P-değerleri üzerinde yapılan Ki-Kare testi sonuçları ve Şekil 3.21'de

sırası ile yer alan üreteç-1, üreteç-2 ve üreteç-3'ün P-değerlerinin histogramları birlikte değerlendirildiğinde; birörnek dağılıma sahip olduğu değerlendirilmektedir.



Şekil 3.21 RC4 Algoritması Strateji-2 Koşum Testi Histogramı

Çizelge 3.13'te yer alan sonuçlarda, RC4 algoritmasına ait üç üretelin 200 örnekleminin Spektral testten kaldığını gözlenmektedir. Şekil 3.22'de sırası ile yer alan üreteç-1, üreteç-2 ve üreteç-3'ün histogram grafiklerindeki dağılımın inişli-çıkışlı yapısı, örneklemdaki sayı dizisinin birörnek dağılıma sahip olmadığına dair bir göstergedir.



Şekil 3.22 RC4 Algoritması Strateji-2 Spektral Testi Histogramı

RC4 algoritmasına ait üç üretelin 200 örnekleminin yapılan Spektral testin sonucu başarısız olmasına rağmen üç üreteçte her üreteç için tüm veriye yapılan Spektral testten başarı ile geçilmiştir. Bu durum örneklemelerin bazı testlerden geçemeyebileceği ancak tüm veriye aynı test uygulandığında üretelin o testten başarı ile geçebileceğini göstermektedir. Bu sonuçlara göre üretelin örneklemelerinin rastsal dizilerden oluştuğu değerlendirilmektedir.

3.4 LFSR Algoritmalarının Karşılaştırılması

Bu tez kapsamında akış şifreleme uygulamalarında kullanılacak LFSR tabanlı anahtar dizisi üretme algoritmalarından önde gelen A5/1, A5/2 ve RC4 algoritmaları olmak üzere 3 algoritma incelenmiştir. Çizelge 3.14'te LSFR tabanlı algoritmaların karşılaştırmaları verilmiştir.

Çizelge 3.14 LFSR Algoritmalarının Karşılaştırılması

	A5/1	A5/2	RC4
Geliştirilme Yılı	1987	1989	1987
Anahtar Uzunluğu	64 bit (bazı durumlarda anahtar 54 bittir, son 10 bit 0 ile doldurulur)	64 bit (bazı durumlarda anahtar 54 bittir, son 10 bit 0 ile doldurulur)	40-2048 bit arasında değişken anahtar uzunluğu vardır. Ancak genelde 64 bit ve 128 bit kullanılır
Alt fonksiyon kullanımı	Çoğunluk fonksiyonu	Çoğunluk fonksiyonu	S permütasyon işlemi
Gereken Parametreler	Kc ve Fn	Kc ve Fn (22 bit)	K, IV (WEP uygulamaları için)
Zaman Karmaşıklığı	2^{54} civarında	2^{17} civarında	
Kombinasyonel Karmaşıklığı	2^{40} ya da 2 saniye	4.6 milisaniye (1 çerçeve iletim zamanı)	2^{13} ya da 2^{33}
Periyodu	Teorik olarak; $(2^{19}-1) (2^{22}-1) (2^{23}-1)$ Deneyisel olarak; $(4/3) \times (2^{23}-1)$		128 bit anahtar için; $>10^{100}$
Senkron / Kendi Kendine Senkron Şifreleme	Senkron	Senkron	Senkron
LFSR	3	4	-

	A5/1	A5/2	RC4
Mantıksal Kapılar	Var	Var	Var
Düzensiz Tetikleme	Var	Var	-
Yazılım/Donanım	Donanım	Donanım	Yazılım ve Donanım
Uygulama alanı	GSM sisteminde baz istasyonu ile mobil cihazı arasındaki veri alış-verişinin şifrlenmesinde	GSM sisteminde baz istasyonu ile mobil cihazı arasındaki veri alış-verişinin şifrlenmesinde	WEP, WPA, SSL/TLS, Bit Torrent Protokol Şifreleme vs.
Çıktısı	Doğrusal olmayan	Doğrusal olmayan	Doğrusal olmayan
Türediği Algoritma	-	A5/1	-

4 SÖZDE RASTSAL SAYI DİZİSİ TABANLI YÖNTEMLER

PRNG'lerin uygulama alanları oldukça geniştir. Fizik, benzetim, yazılım testleri, küresel pozisyon sistemleri (GPS), radar sistemleri, akış şifreleme ve Kod Bölmeli Çoklu Erişim (CDMA) uygulamalarından bazılarıdır. İyi bir PRNG'lerde beklenen özellikler aşağıda belirtilmiştir [Kahaner, 1977].

- Tekdüzeliliğin olması,
- İlinti olmaması : Dizi elemanları arasında ilinti olmaması, dizi elemanlarının periyodik özellikler göstermemesi,
- Verimli olması : Koşumların kısa sürede tamamlanması ve üretilen veriler için minimum bellek kullanılması,
- Kaliteli olması : Uygulanacak istatistiksel testlerden geçmesi, üretilen dizilerin rastsal olması ve üreticinin güvenli olması,
- Periyot : Yeterince uzun periyoda sahip diziler üretmesi,
- Tekrarlanabilir olması: Başlangıç değeri üreteceye tekrar verildiğinde önceden üretilen dizinin tekrar üretilmesi,
- Basitlik : Uygulaması ve kullanımı kolay olması,
- Taşınabilirlik : Farklı sistemlere entegre edildiğinde aynı sonuçları vermesi

Birçok PRNG kriptografik olarak güvenli değildir bu nedenle hepsi aynı hassasiyetteki şifreleme uygulamalarında kullanılmaz. Şifreleme uygulamalarında kullanılacak üreticinin ürettiği dizilerin tahmin edilemez olması gerekir [Knuth, 1981]. Bunun içinde daha özel tasarım güçlü bir PRNG algoritmasına ihtiyaç vardır. Çıktılar arasında herhangi bir doğrusallık veya ilinti olmamalıdır; çünkü birçok istatistiksel test yardımı ile ilinti var ise tespit edilir. Güçlü bir algorithmadan kasıt hesaplama zorluğu içeren Blum Blum Shub, Fortuna, RSA ve Blum Micali gibi algoritmalarıdır. Bunlar kriptografik olarak güvenli (CSPRNG) algoritmalar olarak nitelendirilir. İyi istatistiksel özelliklere sahip çıktılar üreten ancak buna rağmen şifreleme uygulamaların haricindeki alanlarda tercih edilen Doğrusal Eşleniksel Üreteç (LCG) ve Gecikmeli Fibonacci Üreteç gibi algoritmalar da mevcuttur.

İdeal bir sözde rastsal sayı üretici, kısa bir başlangıç değerinden uzun bir bit dizisini hızlı bir şekilde üretendir [Blum, 1986]. Ancak kritik önem seviyesine sahip uygulamalarda kullanılacak dizi için dizi üretici algoritmalarında sadece rastsallık ve iyi istatistiksel özelliklere sahip olması değil aynı zamanda uzun periyoda sahip olması da önem taşımaktadır [Glen, 2002].

Bu tezde PRNG tabanlı sözde rastsal sayı üretici algoritmalarından yaygın olarak kullanılan Blum Blum Shub (BBS), Doğrusal Eşleniksel Üreteç (LCG), Park-Miller, Ters Eşleniksel Üreteç (ICG) ve Mersenne Twister (MT) yöntemleri incelenmiştir. Algoritmaların periyotları, rastsallıkları ve şifreleme uygulamalarına uygunlukları çeşitli istatistiksel testler yapılarak incelenmiştir. Test sonuçları değerlendirilerek, elde edilen sonuçlar her algoritmanın uygulama bölümünün sonunda verilmiştir.

4.1 BLUM BLUM SHUB ALGORİTMASI

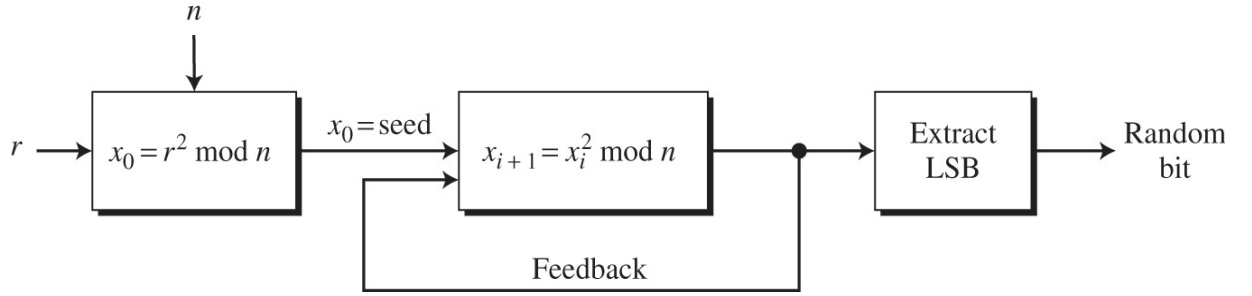
Blum Blum Shub (BBS) algoritması 1980-1986 yılında Lenore Blum, Manuel Blum ve Michael Shub tarafından geliştirilen bir sözde rastsal sayı üreticidir. BBS, sözde rastsal sayı üretici (PRNG) algoritmalar arasında kriptografik olarak güvenli olan en yaygın algoritmadır. BBS algoritması hesaplama yoğunluğuna dayandığından algoritmanın anlaşılabilmesi için ciddi bir matematik alt yapısı gerekir. Blum Blum Shub denklemi aşağıdaki şekilde ifade edilmiştir.

$$x_{n+1} = x_n^2 \bmod M \quad (4.1)$$

İki büyük asal sayı olan p ve q çarpımından oluşan $M = p \times q$ parametresi “Blum Tamsayısı” olarak tanımlanmaktadır [Cusik, 1994]. Algoritmanın her bir döngüsünde x_{n+1} ’den çıktı üretilir. Bu çıktı genellikle eşlik biti ya da en önemsiz bitten oluşur. Eşlik biti yöntemi, tek eşlik ve çift eşlik biti olmak üzere ikiye ayrılır.

Tek eşlik biti (ya da çift eşlik biti), ikili gösterimdeki 1’lerin toplamının her zaman tek (ya da çift) olmasını sağlayacak şekilde diziye eklenen bittir. Veri iletiminde herhangi bir nedenle hata olursa, alıcı taraf eşlik bitine bakarak hata olduğunu anlar; ancak hangi bitinde ya da kaç bitinde hatalı olduğunu anlayamaz. Eşlik biti yöntemi, güvenli bir hata denetimi imkanı sağlamasa da bir bit kullanarak hata kontrolünün yapılması bu yöntemi avantajlı kılar.

Blum Blum Shub algoritmasının çıktısından 0 ve 1 değerlerini çoğunlukla eşit sıklıkta alınması, verinin rastgele olduğunu ve periyodunun yeterince uzun olduğunu göstermektedir [Griffin, 2000]. Dengede tanımından kastedilen uzun bir koşulda 0 ve 1 değerlerinin eşit frekansta olmasıdır [Cusik, 1994]. BBS algoritmasının fiziksel yapısı aşağıdaki şekilde verilmiştir.



Şekil 4.1 BBS Algoritmasının Fiziksel Yapısı

İncelediğimiz diğer algoritmalarından farklı olarak kriptografik anlamda güvenli bir algoritma olma özelliğini taşımaktadır. Güvenli olmasını sağlayan en önemli etken algoritmanın Karesel Kalıntı (QR) problemine dayanıyor olmasıdır.

QR problemi, tamsayıları asal çarpanlarına ayırma problemi olarak tanımlanır. Hesap zorluğu içerdiği için BBS'in güvenli bir algoritma olmasını sağlamaktadır. QR problemi ikiye ayrılarak ayrık logaritma problemi ve çarpanlara ayırma problemi olarak ele alınabilir [Blum, 1984]. BBS algoritması çarpanlara ayırma problem nedeni ile hesaplama karmaşıklığı teorisinden yararlandığından, büyük bir M değeri için problemin çözümü oldukça zordur [Cusik, 1994]. $M = p \times q$ değeri ne kadar büyürse asal çarpanlara ayırma işlemi zorlaşacaktır; çünkü büyük asal çarpanlarla işlem yapmak hesap karmaşıklığını artıracak dolayısıyla da hesap zamanını uzatacaktır. Bu problemin geçerli olması için parametrelerin uygun seçilmesi gerekir.

BBS algoritması açık anahtar ile şifrelemeye dayanır; bu nedenle çok yavaştır. M (Blum tamsayı) değeri açık anahtar, $M = p \times q$ oluşturan p ve q asal değerleri gizli anahtar olarak ele alınmaktadır [Blum, 1986]. Yavaşlığı nedeni ile çoğul ortam ve benzetim uygulamalarında tercih edilmezken, kriptografik alanlarda anahtar

retme uygulamalarında kullanılır. Kriptografik anlamda gvenli szde rastsal sayı retici (CSPRNG) bir algoritma olarak sınıflandırılmaktadır. Bunun nedeni ise yksek gvenlik dzeyine sahip olması ve bir sonraki durumun tahmin edilemez olmasıdır. Tasarımındaki bu yapı sayesinde saldırılara karřı dayanıklıdır. Bu zellikleri nedeni ile askeri ve diplomatik haberleřme alanlarında BBS algoritması ne çıkmaktadır [Ding, 1997].

Birok PRNG'den farklı olarak ama her CSPRNG de olduęu gibi tm istatistiksel testleri geerek bir sonraki durumun polinomsal zamanda da tahmin edilemez olduęunu kanıtlamıř bir algoritma olma zellięini tařımaktadır. [Blum, 1986]. Algoritmanın gvenli ve tahmin edilemez zelliklere sahip olması iin kritik parametrelerin uygun seilmesi gerekmektedir. BBS algoritmasında olması gereken tasarım kriterleri ve algoritma adımları ařaęıda belirtilmiřtir.

Adımlar;

- 1) p ve q ayrı ayrı asal tek sayılar olarak seilir.
- 2) $M=N = p \times q$ Blum Tamsayısı olarak tanımlanır, M ya da N olarak ifade edilir.
- 3) $p \equiv q \pmod{N} \rightarrow p = q = 3 \pmod{4}$ (4.2)

p ve q eřleniksel sayılar olması gerekir. Bu denklemin saęlanması p ve q deęerlerinin mod N ile QR olmasını saęlar.

$$4) \quad \gcd(x_0, N) = 1 \quad (4.3)$$

Bařlangı deęeri x_0 dahil olmak zere tm x_i deęerleri, Blum tamsayısı ile greceli asal yani En Byk Ortak Bleni (EBOB) = 1 olmalıdır.

- 5) λ , Carmichael fonksiyonu tanımlanır.

$$\lambda(N) = \lambda(p \times q) = \text{lcm}(p-1, q-1) \quad (4.4)$$

Euler teoreminin Carmichael uzantısına gre; [Knuth, 1997]

$$\gcd(x_0, N) = 1 \text{ iken } x_0^{\lambda(N)} = 1 \pmod{N} \quad (4.6)$$

Bu denklemi saęlayan en kk deęer aranır.

$$6) \quad p_1 = (p-1)/2, \quad q_1 = (q-1)/2 \quad (4.7)$$

$$p_2 = (p-3)/4, \quad q_2 = (q-3)/4 \quad (4.8)$$

p_1, q_1, p_2 ve q_2 değerlerinin tümü tek ve asal sayılar olmalıdır [Moreau, 1996].

- 7) 2, mod N ile bir QR olmalıdır. Aşağıdaki denklemi sağladığında ikinin QR olduğu kanıtlanır. Denklemden bulunan derece, $\text{ord}_N x_i = \text{ord}_N x_{i+1}$ bu şekilde tanımlanır.

$$\text{ord}_{\lambda(N)/2}(2) = \lambda(\lambda(N)) \quad (4.9)$$

Daha kolay bir matematiksel ifade olarak aşağıdaki denklemi kullanmamız mümkündür. 2 en fazla p_1 veya q_1 parametrelerinden biri ile QR olmalıdır. Aşağıdaki ifadelerden en fazla birinin sağlandığı durumda geçerli olup x_p ve x_q aynı anda var olamayacak şekilde parametreler seçilir [Moreau, 1996].

$$2 = x_p \bmod p_1 \quad \text{ve} \quad 2 = x_q \bmod q_1 \quad (4.10)$$

$e = (p_1 - 1)/2$ ile Euler ölçütü kullanılarak $x = 2^e \bmod p_1$ ile 2'nin QR olup olmadığı çözülebilir. p_1 için olan gösterim aynı şekilde q_1 için de düzenlenebilir.

- 8) $x_0 \bmod M$ ile bir QR olmalıdır.

$$\text{ord}_N(x_0) = \lambda(N)/2 \quad (4.11)$$

$x_0 \bmod M$ ile bir QR olduğu durumda dizinin periyodu aşağıda belirtildiği şekilde tanımlanır.

$$\pi = \pi(x_0) = x_0, x_1, x_2, \dots \quad (4.12)$$

BBS dizisinin periyodu, parametrelerin uygun seçildiği durumda oluşabilecek en büyük periyodu ifade edilen $\lambda(\lambda(N))$ değerinin böleni olarak tanımlanmaktadır.

$$\pi | \lambda(\lambda(N)) \quad (4.13)$$

Eş. 4.13 ifadesi aşağıdaki tamsayılardan birine eşittir.

- $\text{ord}_{\lambda(N)/2}(2) = p_2 q_2$
- $\text{ord}_{\lambda(N)/2}(2) | 2p_2$
- $\text{ord}_{\lambda(N)/2}(2) | 2q_2$

9) Maksimum periyotta bir dizi elde etmek için x_0 ve 2'nin bir QR olma koşullarının sağlanması gerekir. Bu koşulların sağlandığı durumda ise;

$$\pi | \lambda(\lambda(N)) = \lambda(\lambda(N)) | \pi(x_0) = 1 \quad (4.14)$$

Bu ifadeler birlikte değerlendirildiğinde maksimum periyodu gösteren eşitlik ortaya çıkmaktadır.

$$\lambda(\lambda(N)) = \pi(x_0) = \text{maksimum periyot} \quad (4.15)$$

10) N yukarıda belirtilen özelliklere sahip olduğu durumda;

$$\lambda(N) = \text{lcm}(2p_1, 2q_1) = 2p_1 q_1 \quad (4.16)$$

$$\lambda(\lambda(N)) = \text{lcm}(2p_1, 2q_1) = 2p_2 q_2 \quad (4.17)$$

Eş. 4.16 ve Eş. 4.17'de belirtildiği şekilde ifade edilebilir. Bu denklemler Eş. 4.4, Eş. 4.7 ve Eş. 4.8'de belirtilen denklemler kullanılarak çıkarılmıştır. Eş. 4.16 ve Eş. 4.17'deki denklemlerden aşağıda belirtilen eşitliğin rahatlıkla ortaya çıktığı görülmektedir.

$$\lambda(\lambda(N)/2) = \lambda(\lambda(N)) \rightarrow \text{ord}_{\lambda(N)/2}(2) | \lambda(\lambda(N)) = \text{ord}_{\lambda(N)/2}(2) | 2p_2 q_2 \quad (4.18)$$

11) BBS algoritması ile başlangıç değerinden x_0 , herhangi bir i. duruma ait x_i değeri Euler teoremi yardımı ile elde edilebilir.

$$x_i = (x_0^{2^{\text{imod}\lambda(N)}}) \bmod N \quad (4.19)$$

12) Sözde rastsal dizinin periyodunu belirlemek üzere yapılması gereken adımlar aşağıda tanımlanmıştır.

$$x_i = (x_0^{2^{\text{imod}\lambda(N)}}) \bmod N \rightarrow x_\pi = (x_0)^{2^\pi} \bmod N = x_0 \quad (4.20)$$

Eş. 4.13'de ve Eş. 4.6'da yer alan denklemler kullanılarak aşağıdaki denklem elde edilir.

$$(x_0)^{2^{\lambda(\lambda(N)) \bmod \lambda(N)}} \bmod N = x_0 \quad (4.21)$$

Sonuç olarak $\lambda(N)$, $\lambda(\lambda(N))$ ve $\lambda(\lambda(N))$ değerinin çarpanları bilindiğinden periyot belirlenebilir. π periyot için d değerinin aranır.

$$d \mid \lambda(\lambda(N)) \rightarrow (x_0)^{2^{\lambda(\lambda(N))/d \bmod \lambda(N)}} \bmod N = x_0 \quad (4.22)$$

Periyodun hesaplanmasında aşağıdaki denklem ile gerçekleşir.

$$\pi = \lambda(\lambda(N))/d \quad (4.23)$$

Eş. 4.9 ve Eş. 4.11'deki denklemlerinin sağlanması ile maksimum periyotta bir sözde rastsal dizi üretilebilir. Bu durumda $d = 1$ değerini alarak maksimum periyot Eş. 4.14'te olduğu gibi ifade edilir.

Görüldüğü gibi BBS algoritması karmaşık bir matematik yapısına ve kısıtlarına dayanmaktadır. Hesap yoğunluğu çok olduğu için, iyi bir matematik altyapısı gerektirmektedir. BBS algoritmasında hesap yoğunluğu, seçilen parametrelerle polinomsal zamanda çözülemez hale getirilmektedir. Polinomsal zamandan kasıt, bir problemin büyüklüğü doğrusal artarken hesaplanması için gereken vaktin bir polinom fonksiyonuna bağlı olarak artmasıdır. Bu nedenle BBS algoritması güvenli ancak yavaş bir algoritmadır.

4.1.1 BBS Algoritmasının Uygulanması

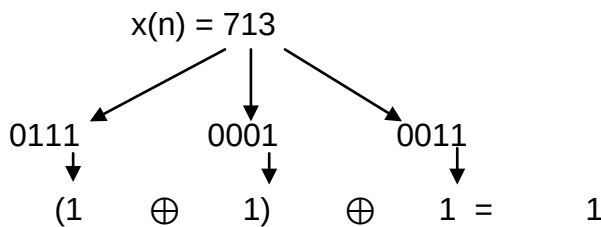
BBS sözde rastsal sayı üretme algoritması hakkında yukarıdaki bölümde algoritma adımları tek tek anlatılarak detaylı bilgi verilmiştir. BBS algoritmasının kodlanması, programlama dili olan MATLAB ile gerçekleştirilmiştir. BBS algoritması kodlanma sırasında yapılan kontroller ve algoritma adımları aşağıda belirtilmiştir.

- 1) İterasyonun başladığı sayı M olarak düşünülür ve iki çarpandan mı oluşuyor kontrol edilir.
- 2) Çarpanlar ayrı ayrı bir dizi elemanı olarak alınır.

- 3) Elde edilen iki çarpan ayrı ayrı eşleniksel değerler mi kontrol edilir.
- 4) İstenen koşulu sağlıyor ise dizi elemanları p ve q test verisi olarak atanır. (Diğer koşulları sağladığından emin olmadan p ve q değeri olarak atanamayacağından test verisi olarak değerlendirilmektedir.)
- 5) p ve q test verileri için p_1 , q_1 , p_2 ve q_2 değerleri hesaplanır.
- 6) p_1 , q_1 , p_2 ve q_2 değerlerinin tek ve asal sayılar olması beklenir. Bunun için ayrı bir kontrol yapılır.
- 7) Bu koşulda sağlanıyor ise 2 en fazla p_1 ve q_1 'den biri ile QR mı kontrol edilir.
- 8) Bu koşulun sağlanmasının ardından p ve q test verileri, p ve q değerleri olarak atanır.
- 9) Seçilen p ve q değeri ile başlangıç değeri aranır. Başlangıç değerinin Blum tamsayısı (M) ile göreceli asal olması ve başlangıç değerinin bir QR olması beklenir.
- 10) Koşulları sağlayan başlangıç değeri ve p, q, p_1 , q_1 , p_2 ve q_2 periyot bulmayı sağlar.
- 11) Çıktı üretmenin 2 farklı yöntemi vardır.

I. Normal Çıktı Alma Yöntemi

Sözde rastsal sayı dizinin oluşturulması için elde edilen x_n sayısının onluk tabandaki her bir basamağı ikilik sisteme çevrilir ve en önemsiz bitleri XOR işleminden geçirilir. Bu sayede x_n değeri için sözde rastsal sayı dizinin bir çıktısı elde edilir.



II. Eşlik Biti ile Çıktı Alma Yöntemi

Bu yöntem seçildiğinde tek eşlik ya da çift eşlik bitleri kullanılarak çıktı alınmaktadır. Tek çıktı alma yöntemi ile elde edilen çıktı çift eşlik biti ile elde edilen çıktının ikili sistemde tersidir. Bu nedenle BBS algoritmasında bu çıktı alma yöntemini seçildiğinde çift eşlik biti yöntemi seçilmiş ve normal çıktı alma yöntemi ile kıyaslanmıştır.

- 12) Üretilen dizinin periyodu hesaplanır. Maksimum periyoda sahip bir dizi üretilip üretilmediği kontrol edilir.
- 13) Testler iki stratejiye ayrılarak yapılır. Strateji-1 de üreteç tarafından üretilen tüm veriye, 1000000 uzunluğundaki veri, testler yapılırken; Strateji-2 de ise bölünmüş veriye yani 5000 örneklem dizi uzunluğuna sahip 200 örnekleme test yapılır.

BBS algoritması için örnek olarak iterasyon başlangıcı $2^{26} = 67108864$ değeri seçilmiştir. İterasyon bu değerden başlayarak istenilen sayıda koşullara uygun p ve q seti aranır. Koşullara uygun p ve q seti bulduktan sonra arama durdurulur ve başlangıç değeri, periyot bulma işlemleri gerçekleştirilir.

4.1.2 Uygulanan Testlerin Sonuçları ve Değerlendirmeler

Tüm üreteçlere

- Ki-Kare Testi,
- Frekans Testi,
- Koşum Testi,
- Ayrık Fourier Dönüşümü (Spektral) Testi

yapılarak elde edilen test sonuçları çizelge ve grafik halinde bu bölümde sunulmuş ve test sonuçlar yorumlanmıştır.

Bu testlerin sıfır hipotezini gösteren h ve P- değerleri incelenir. $h = 0$ ise sıfır hipotezinin kabul edildiğini gösterir. Yapılan testler anlam düzeyi (α) olarak %1 seçilmiştir. Bu nedenle P-değeri ≥ 0.01 ise üretilen dizi rastsalıdır.

Örnek olarak algoritma tarafından seçilmiş koşullara uygun üç p ve q veri setine, belirtilen testler uygulanmış, sonuçlar çizelge ve grafik olarak sunulmuştur. Algoritma girdisi olarak iterasyonun başlayacağı 2^{26} değeri verilmiştir. Algoritma çıktısı olarak 3 p ve q seti diğer bir deyişle 3 üreteç alınmıştır.

Test girdileri olarak;

- Algoritma çıktısı
- Üretilen dizi uzunluğu : 1000000
- Örneklemdaki dizi uzunluğu : 5000
- Örneklem sayısı : 200
- $\alpha = 0.01$ değerleri seçilmiştir.

Tüm algoritmalarda aynı çıktı elde etme yöntemi ile çıktılar alınmış ve bu çıktılar testlerden geçirilmiştir. Diğer algoritmalarından farklı olarak BBS algoritmasında çift eşlik biti çıktı alma yöntemi kullanılarak da çıktı elde edilmiştir. BBS algoritmasının üç üretici için her iki çıktı alma yöntemi kullanılarak testler yapılmıştır. İlk olarak çift eşlik biti yönteminin çizelge ve grafikleri, daha sonra normal çıktı alma yöntemine ait çizelge ve grafikler sunularak ortak bir değerlendirme yapılmıştır.

BBS algoritmasına ait üreteçlerin, maksimum periyoda sahip diziler üretip üretmedikleri kontrol edilmiş; maksimum periyoda sahip diziler ise max_flag=1 değerine, değil ise max_flag=0 değerine eşitlenmiştir.

4.1.2.1 Strateji-1 Testleri

Seçilen üreteç ile üretilen 1000000 dizi uzunluğundaki veriye Frekans, Koşum, Spektral ve Ki-Kare testleri yapılmıştır.

Çizelge 4.1'de görüldüğü gibi üreteçler en yüksek periyoda sahip olandan en düşük periyoda sahip olan üretece doğru sıralanmıştır. Çizelgede görüldüğü gibi en büyük p ve q değerine ya da M değerine sahip üreteç en yüksek periyoda sahip demek değildir. Örneğin 2. sırada yer alan setin M değeri (67073353) en büyük M değeridir; ancak görüldüğü gibi en uzun periyoda sahip olduğu anlamına gelmemektedir. Max_flag parametresi ile üreteçlerin tümünün maksimum periyoda sahip diziler ürettiği gözlenmektedir.

Çizelge 4.1 BBS Algoritması Strateji-1 Test Sonuçları-1

Üreteç	p	q	M	x_0	Periyot	Max Flag	h_ ki-kare	p_ ki-kare
1	5927	11279	66850633	14791716	8349878	1	0	0,445553
2	719	93287	67073353	29008996	8348918	1	0	0,714214
3	359	186479	66945961	31400928	8298182	1	0	0,234955

Tüm testlerden elde edilen h ve p değerleri incelenmiştir. Anlam düzeyi (α) %1 seçildiği durumda Çizelge 4.1’de yer alan sonuçlara göre, üretilen verinin tümüne uygulanan Ki-Kare testinin h değerleri üç üreteç için de sıfırdır. Testlerin sonucunda h=0 olması, sıfır hipotezinin kabul edildiğini göstermektedir. Buna göre BBS algoritması ile oluşturulan üç üretecın Ki-Kare testinden geçtiği gözlenmiştir.

Çizelge 4.2’de eşlik biti ile çıktı üretildiğinde ve bu çıktılara Frekans, Koşum ve Spektral testleri uygulandığında elde edilen test sonuçları verilmiştir. Çizelge 4.3’te ise normal çıktı üretme yöntemi ile çıktı üretildiğinde ve testler yapıldığında elde edilen sonuçlar verilmiştir.

Çizelge 4.2 BBS Algoritması Strateji-1 Eşlik Biti Çıktı Yöntemi Test Sonuçları-2

Üreteç	h_fre	p_fre	h_koşum	p_koşum	h_spek	p_spek	ortalama	varyans
1	0	0,094524	0	0,158478	0	0,479815	0,500152	0,083208
2	0	0,370253	0	0,821048	0	0,659591	0,499815	0,083421
3	0	0,847742	0	0,273533	0	0,222279	0,499869	0,083305

Çizelge 4.2 ve Çizelge 4.3’te görüldüğü gibi her iki çıktı yöntemi kullanılarak üretilen verinin tümüne uygulanan Frekans, Koşum ve Spektral testlerinden üç üreteç de geçmektedir. Buna göre BBS algoritması ile oluşturulan üç üretecın her iki çıktı yönteminde de tüm testlerden geçtiği gözlenmiştir.

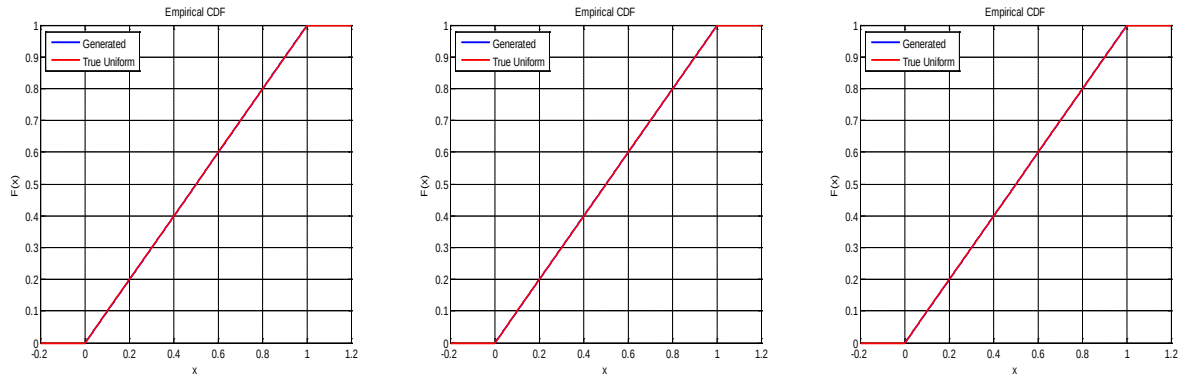
Çizelge 4.3 BBS Algoritması Strateji-1 Test Sonuçları-2

Üreteç	h_fre	p_fre	h_koşum	p_koşum	h_spek	p_spek	Ortalama	varyans
1	0	0,730846	0	0,968796	0	0,093089	0,500152	0,083208
2	0	0,89024	0	0,862666	0	0,299755	0,499815	0,083421
3	0	0,505411	0	0,314683	0	0,1254	0,499869	0,083305

Çizelge 4.1, Çizelge 4.2 ve Çizelge 4.3'te yer alan test sonuçlarına göre üç üreticinin ürettiği dizilerin rastsal olduğu değerlendirilmektedir. Birikimli Dağılım Fonksiyonu (CDF: Cumulative Distribution Function) ve histogram grafikleri incelenerek nihai bir sonuca varılabilecektir.

Birörnek dağılıma sahip bir rastsal dizinin ortalamasının 0.5, varyansının da 0.08 olması gerektiği bilinmektedir [Gentle, 2003]. Her iki çıktı yöntemi ile üç üreticinin ürettiği dizilerin ortalama ve varyans değerlerine bakılarak birörnek dağılıma sahip olup olmadıkları incelenmiştir. Çizelge 4.2 ve Çizelge 4.3'te görüldüğü gibi her iki çıktı alma yönteminde üç üretici için ortalama ve varyans değerleri aynıdır. Buna ek olarak üreticilere ait ortalama ve varyans değerlerinin 0.5 ve 0.08 civarında olduğu ve çok küçük farklılıklar sergiledikleri gözlenmektedir.

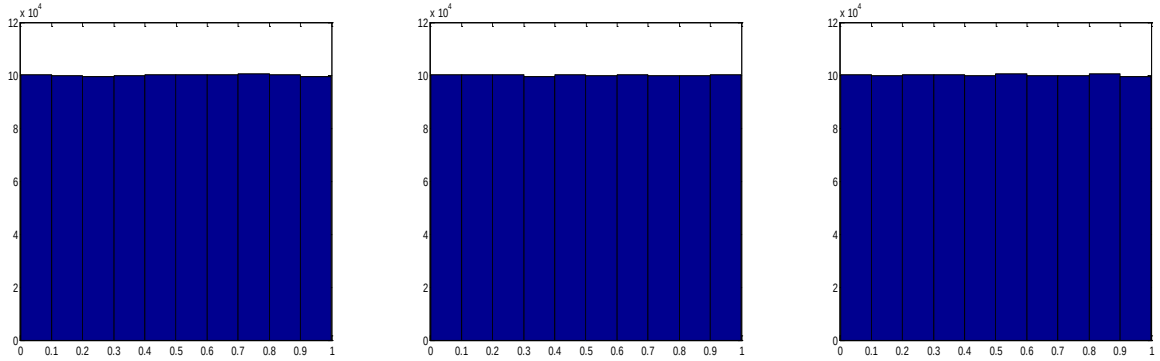
CDF grafiği ile her üretici tarafından üretilen tüm verinin (1000000 uzunluğunda dizinin) birörnek dağılım ile ne kadar örtüştüğünü gözlemek amaçlanır. Şekil 4.2'de sırası ile yer alan üretici-1, üretici-2 ve üretici-3'e ait verinin, birörnek dağılıma ait veri ile tam olarak örtüştüğü gözlenmektedir. Birörnek dağılıma ait beklenen ve üreticilere ait gözlenen veriler örtüştüğü için, üç üreticinin birörnek dağılıma sahip diziler oluşturduğu değerlendirilmektedir.



Şekil 4.2 BBS Algoritması Strateji-1 CDF Grafikleri

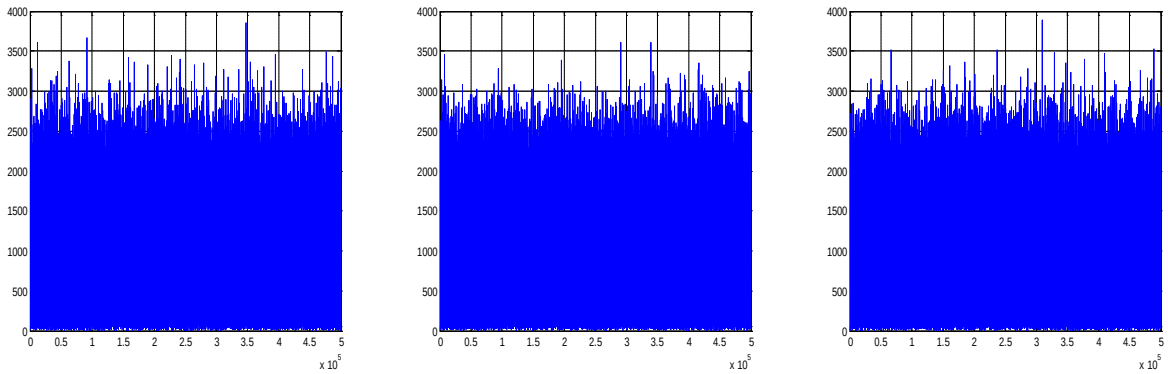
Şekil 4.3'te sırası ile yer alan üretici-1, üretici-2 ve üretici-3 için üretilen verilerin grafiklere yansıyan dalgalanmalarını gözlemleyip kıyaslayabilmek adına CDF ve histogram grafikleri birlikte gösterilmiştir. Ancak BBS algoritmasına ait üç üretici ile üretilen tüm verinin histogramında için bir dalgalanma gözlenmemektedir. Bu

nedenle CDF ve Histogram grafikleri ile de birörnek dağılıma sahip diziler üretildiği görülmektedir.



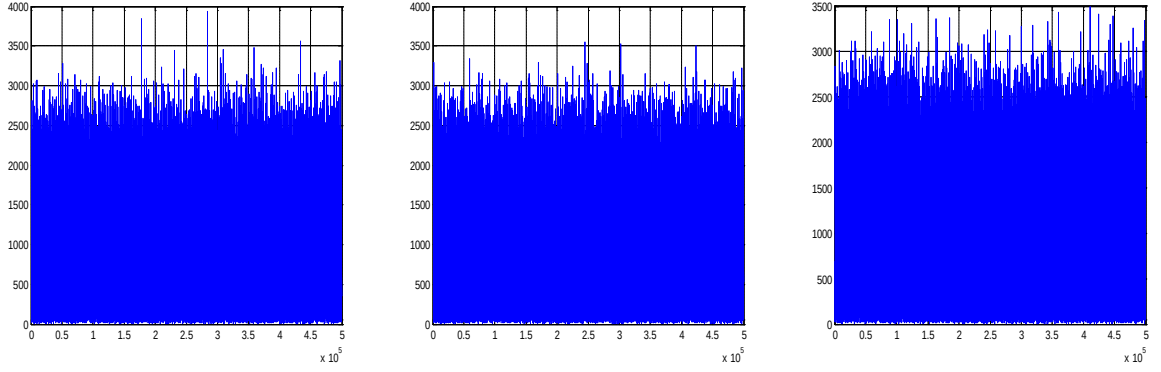
Şekil 4.3 BBS Algoritması Strateji-1 Histogramı

Üç üreteç ile üretilen tüm verinin (her bir üreteç için 1000000 dizi uzunluğundaki veri) Spektral testleri yapılmıştır. Bu test sonuçları çift eşlik biti ile çıktı alma yöntemi için Çizelge 4.2 ve diğer yöntem için Çizelge 4.3'te verilmiştir. Her iki yöntemle ait çizelgelerde yer alan sonuçlara göre üç üreticinin Spektral testten geçtiği bilinmektedir. Şekil 4.4 ve Şekil 4.5'te üç üreteç için sırası ile yer alan FFT grafikleri, her bir üreteç için tüm veriye uygulanan Spektral testin geçildiğini desteklemektedir.



Şekil 4.4 BBS Algoritması Strateji-1 FFT Grafikleri (Eşlik Biti Çıktı Yöntemi)

Bu sonuca ek olarak çıktı alma yönteminin, üç üreticinin tüm verilerine yapılan Spektral testi sonucuna ciddi bir etkisi olmadığı FFT grafikleri ile gözlemlenmiştir.



Şekil 4.5 BBS Algoritması Strateji-1 FFT Grafikleri

BBS algoritması tarafından seçilen üç üretelin tüm verilerine uygulanan testlerin sonuçları, CDF ve histogram grafikleri incelendiğinde; her üretel tarafından üretilen 1000000 uzunluğundaki dizinin, birörnek dağılıma sahip rastsal bir dizi olduğu anlaşılmaktadır.

4.1.2.2 Strateji-2 Testleri

Örneklem sayısı ve bir örneklemdeki dizi uzunluğu seçimi ile algoritma tarafından üretilen dizinin tümü kullanılarak testler gerçekleştirilmiştir. Bu kısımda üretilen tüm veri, örneklem sayısı ve bir örneklemdeki dizi uzunluğu göz önüne alınarak bölünmüştür. Bölünen veriler üzerinde Frekans, Koşum, Spektral ve Ki-Kare testleri yapılmıştır.

Üç üretelin 200 örneklemine yapılan testler sonucunda elde edilen P-değerinin analiz yapılır. P-değeri analizi kapsamında Dizi Geçme Oranı (DGO) hesaplanır. 200 örneklem arasında kaçının sıfır hipotezini geçtiğine dair bir oran elde edilir ve sonuç olarak örneklemelerin testten geçip geçmediğine bakılır. Test sonuçları eşlik biti çıktı yöntemi için Çizelge 4.4'te, normal çıktı alma yöntemi için Çizelge 4.5'te yer almaktadır.

Çizelge 4.4 BBS Algoritması Strateji-2 Eşlik Biti Çıktı Yöntemi Test Sonuçları-1

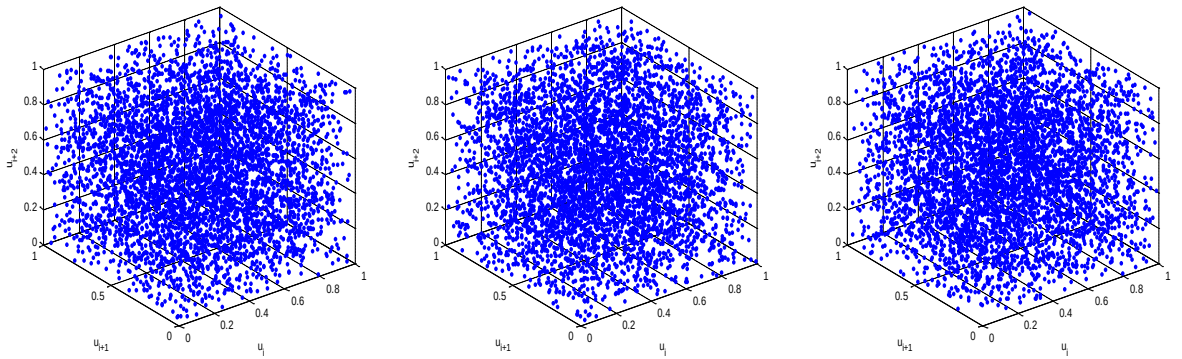
Üreteç	Ki- Kare DGO	Ki-Kare Test Sonucu	Frekans DGO	Frekans Testi Sonucu	Koşum DGO	Koşum Testi Sonucu	Spektral DGO	Spektral Test Sonucu
1	0,98	1	1	1	0,995	1	0,99	1
2	1	1	0,98	1	0,995	1	0,99	1
3	0,985	1	1	1	0,995	1	0,99	1

Her iki çıktı alma yöntemi için Çizelge 4.4 ve Çizelge 4.5'te görüldüğü gibi testi geçme oranları farklı olsa da test sonuçları 1 olduğundan, her iki çıktı yöntemi ile üç üretcin 200 örnekleme uygulanan Frekans, Koşum, Spektral ve Ki-Kare testlerinin tümünden geçildiği gözlenmektedir.

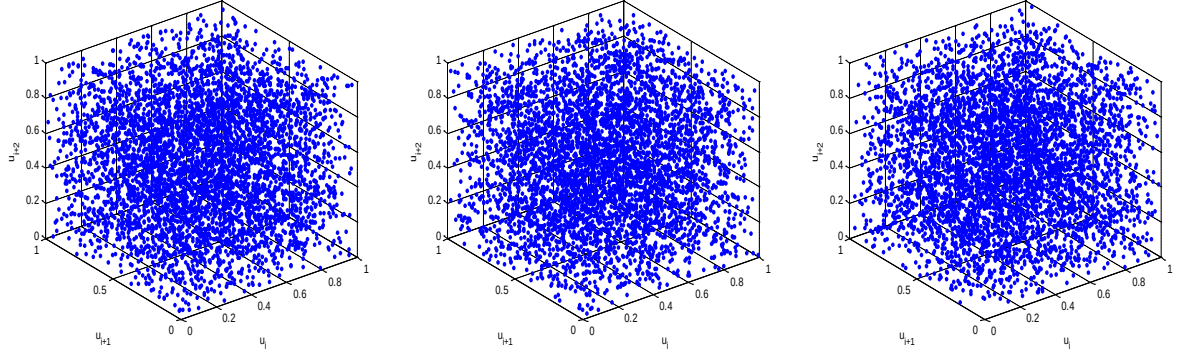
Çizelge 4.5 BBS Algoritması Strateji-2 Test Sonuçları-1

Üreteç	Ki- Kare DGO	Ki-Kare Test Sonucu	Frekans DGO	Frekans Testi Sonucu	Koşum DGO	Koşum Testi Sonucu	Spektral DGO	Spektral Test Sonucu
1	0,98	1	0,985	1	0,985	1	1	1
2	1	1	0,985	1	0,985	1	0,99	1
3	0,985	1	0,99	1	0,99	1	1	1

Üç üretecten rastsal örneklemler alınarak dizi elemanları arasında bir ilinti olup olmadığı, dizi elemanlarının periyodik özellikler sergileyip sergilemediğini incelemiştir. Rastsal olarak 48. , 115. ve 193. örneklemleri seçilerek olan 3-çokuzlu grafikler üzerinden gözlem yapılmıştır. 2-çokuzlu grafiklerde var olan ilintiler gözlemlenemeyebilirken, aynı veriler için 3-çokuzlu grafikler çizildiğinde ilinti gözlemlenmektedir. Dizi elemanları arasındaki ilinti, arka arkaya olan üç duruma ait veri (u_i, u_{i+1}, u_{i+2}) incelenerek tespit edilmeye çalışılmıştır. Dizi elemanları arasındaki ilintinin gözlemlenebilmesi için Şekil 4.6 ile Şekil 4.11 arasında yer alan 3-boyutlu grafikler incelenmiştir.

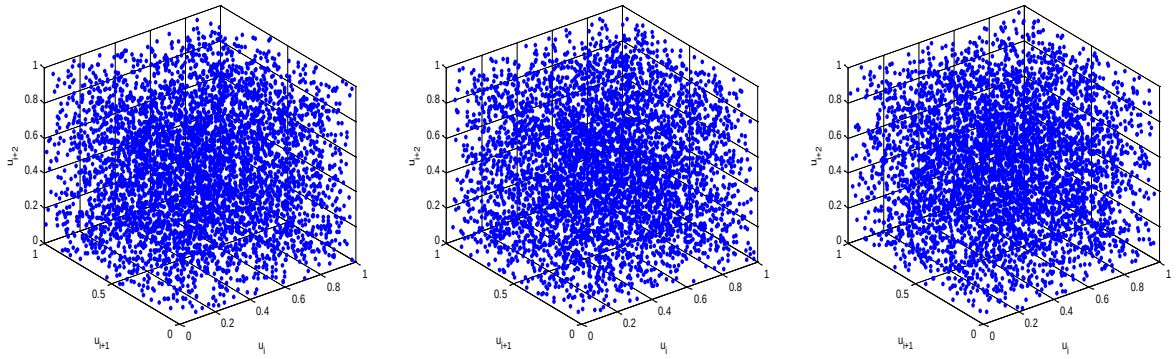


Şekil 4.6 BBS Algoritması Strateji-2, Üreteç-1 48. , 115. ve 193. Örneklemleri
(Eşlik Biti Çıktı Yöntemi)

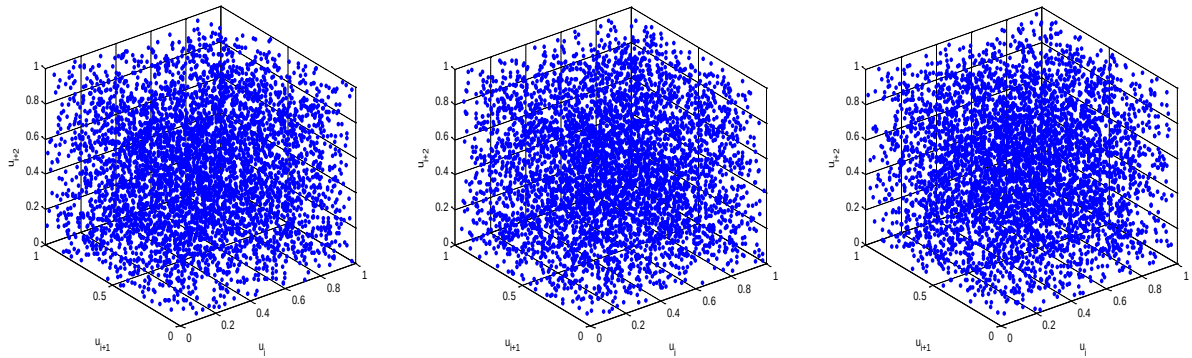


Şekil 4.7 BBS Algoritması Strateji-2, Üreteç-1 48. , 115. ve 193. Örneklemeleri

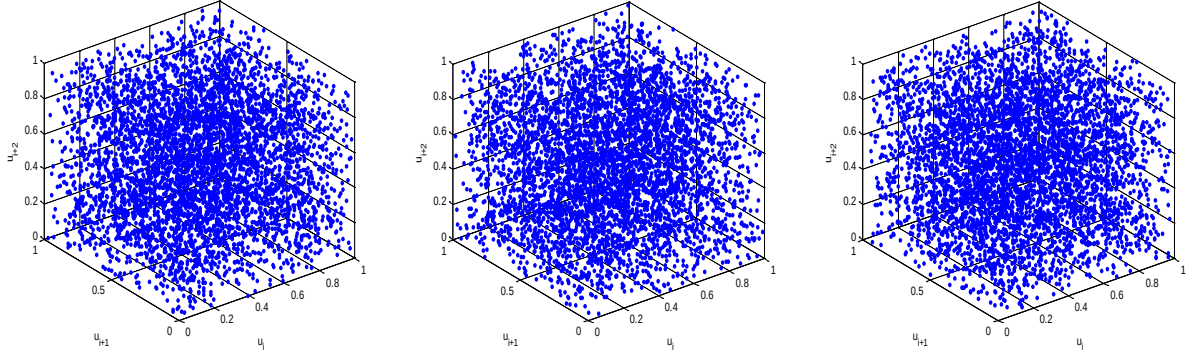
Şekil 4.6 ile Şekil 4.11 arasında görüldüğü gibi her iki çıktı yönteminde üç üreticinin rastsal seçilen hiçbir örnekleminde, dizi elemanları arasında bir ilinti ya da periyodik özellik gözlenmemiştir. Bu durum BBS algoritmasının güçlü bir algoritma olduğunu bir kez daha göstermektedir.



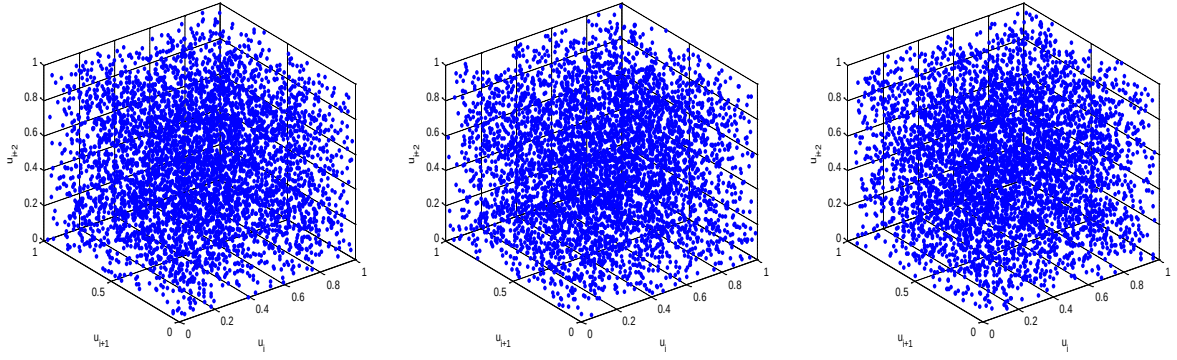
Şekil 4.8 BBS Algoritması Strateji-2, Üreteç-2 48. , 115. ve 193. Örneklemeleri
(Eşlik Biti Çıktı Yöntemi)



Şekil 4.9 BBS Algoritması Strateji-2, Üreteç-2 48. , 115. ve 193. Örneklemeleri



Şekil 4.10 BBS Algoritması Strateji-2, Üreteç-3 48. , 115. ve 193. Örneklemeleri
(Eşlik Biti Çıktı Yöntemi)



Şekil 4.11 BBS Algoritması Strateji-2, Üreteç-3 48. , 115. ve 193. Örneklemeleri

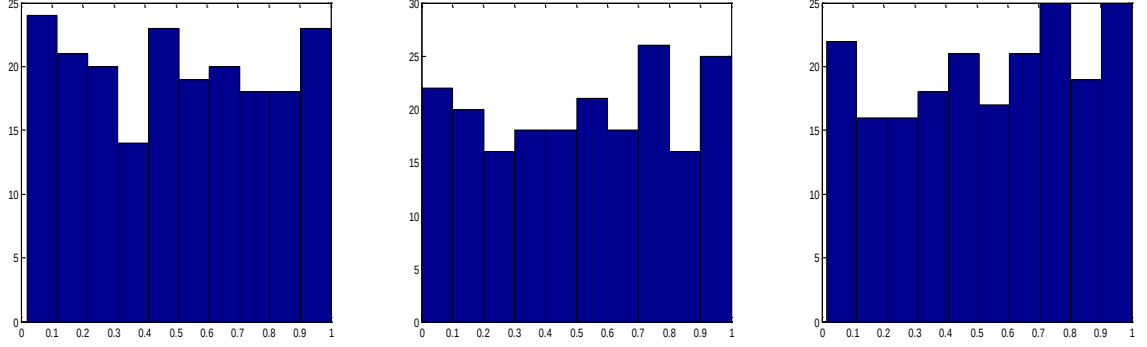
Tüm testler için P-değeri analizi kapsamında P-değerinin dağılımı hesaplanır. Her bir üretcin 200 örneğine uygulanan Frekans, Spektral, Koşum ve Ki-Kare testleri sonucunda elde edilen P-değerlerine Ki-Kare testi uygulanarak P-değerlerinin dağılımına bakılır.

Üreteçlerin P-değerleri dağılımına bakıldığında, her iki çıktı alma yöntemi için üç üretcin örneklemelerinin Frekans testinden geçtiği Çizelge 4.6'da görülmektedir.

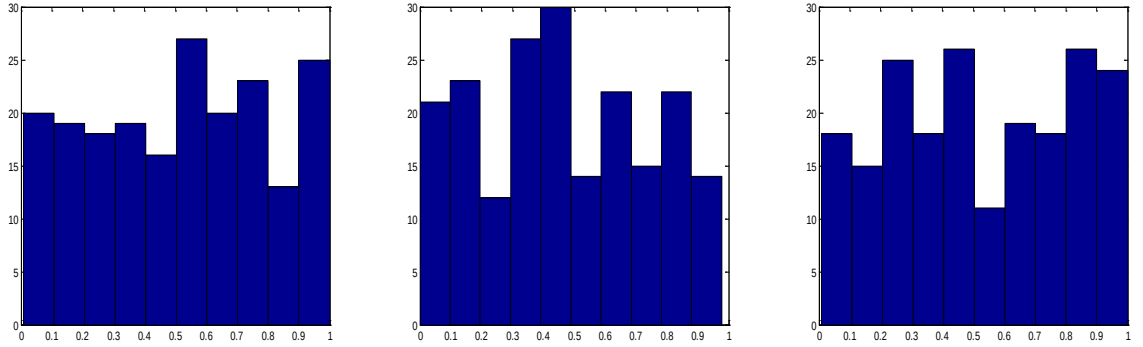
Çizelge 4.6 BBS Algoritması Strateji-2 Frekans Testi

Üreteç	Eşlik Biti Çıktı Yöntemi		Normal Çıktı Yöntemi	
	PD Frekans Testi H değeri	PD Frekans Testi P-değeri	PD Frekans Testi H değeri	PD Frekans Testi P-değeri
1	0	0,956355	0	0,547793
2	0	0,791781	0	0,021174
3	0	0,837896	0	0,209027

Çizelge 4.6'da yer alan Ki-Kare testi sonuçları ve P-değerlerinin histogramları birlikte değerlendirilmektedir. Şekil 4.12 ve Şekil 4.13'te sırası ile yer alan üreteç-1, üreteç-2 ve üreteç-3 histogramlarında görünen iniş-çıkışların, P-değerlerinin birörnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturmadığı anlaşılmaktadır.



Şekil 4.12 BBS Algoritması Strateji-2 Frekans Testi Histogramı
(Eşlik Biti Çıktı Yöntemi)



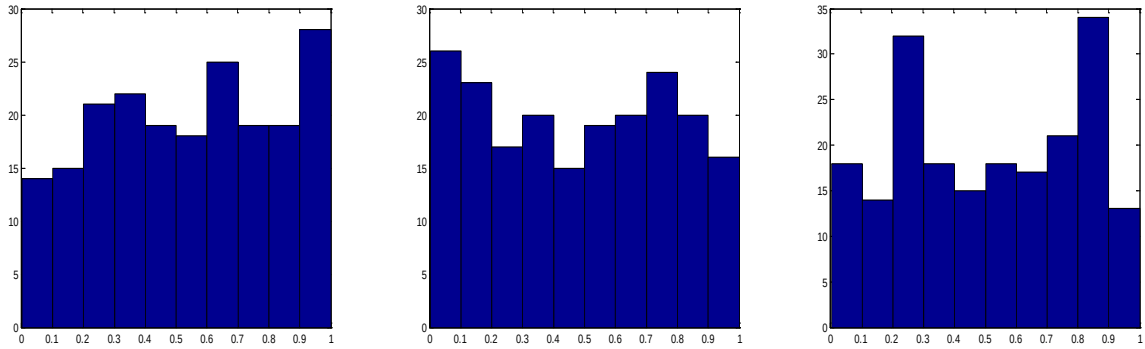
Şekil 4.13 BBS Algoritması Strateji-2 Frekans Testi Histogramı

Eşlik biti ile çıktı alma yönteminde ilk iki üretecin Koşum testinden geçtiği, 3. üretecin ise kaldığı Çizelge 4.7'de görülmektedir.

Çizelge 4.7 BBS Algoritması Strateji-2 Koşum Testi

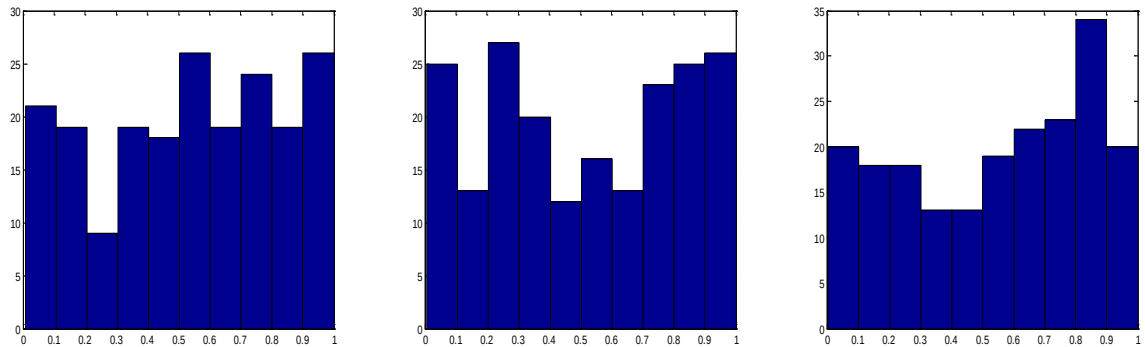
	Eşlik Biti Çıktı Yöntemi		Normal Çıktı Yöntemi	
Üreteç	PD Koşum Testi H değeri	PD Koşum Testi P-değeri	PD Koşum Testi H değeri	PD Koşum Testi P-değeri
1	0	0,490529	0	0,280484
2	0	0,806942	0	0,074812
3	1	< 0,01	0	0,069327

Çizelge 4.7’de yer alan P-değerleri üzerinde yapılan Ki-Kare testi sonuçları ve Şekil 4.14’te sırası ile yer alan üreteç-1, üreteç-2 ve üreteç-3’ün P-değerlerinin histogramları birlikte değerlendirilmektedir. İlk iki üreteç için Şekil 4.14’te yer alan histogramda görünen iniş-çıkışların, P-değerlerinin birörnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturmadığı anlaşılmaktadır. 3. üreteç için ise histogramda görünen iniş-çıkışların, P-değerlerinin birörnek dağılıma sahip olmadığına dair göstergeler oluşturduğu anlaşılmaktadır.



Şekil 4.14 BBS Algoritması Strateji-2 Koşum Testi Histogramı
(Eşlik Biti Çıktı Yöntemi)

Normal çıktı alma yönteminde ise üreteçlerin Koşum testinden geçtiği, Çizelge 4.7’de görülmektedir. Üç üreticinin P-değerleri üzerinde yapılan Ki-Kare testi sonuçları ve P-değerlerinin histogramları birlikte değerlendirilmektedir. Şekil 4.15’te sırası ile yer alan üreteç-1, üreteç-2 ve üreteç-3’e ait histogramlarda görünen iniş-çıkışların, P-değerlerinin birörnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturmadığı anlaşılmaktadır.



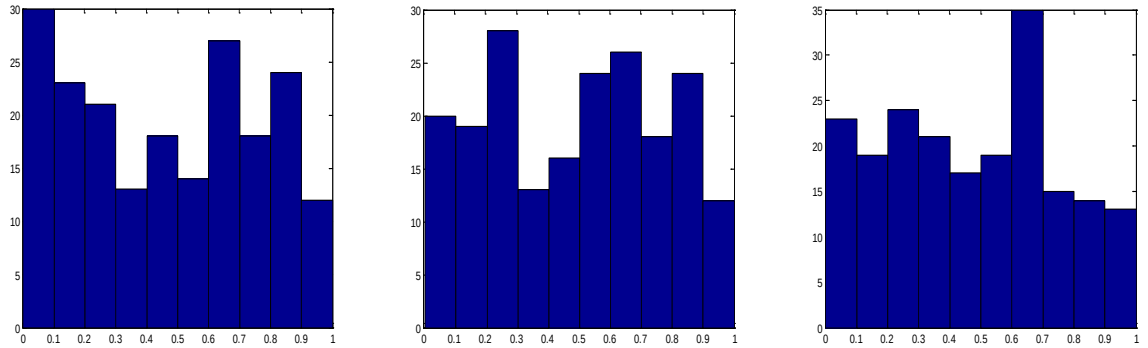
Şekil 4.15 BBS Algoritması Strateji-2 Koşum Testi Histogramı

Üreteçlerin P-değerleri dağılımına bakıldığında, Spektral test sonuçları Çizelge 4.8'de görüldüğü gibi çıktı alma yöntemine göre değişmektedir.

Çizelge 4.8 BBS Algoritması Strateji-2 Spektral Testi

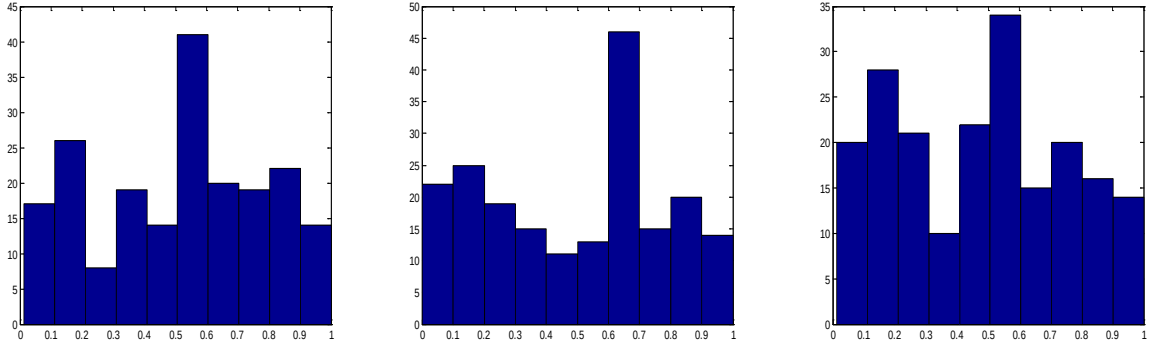
Üreteç	Eşlik Biti Çıktı Yöntemi		Normal Çıktı Yöntemi	
	PD Spektral Testi H değeri	PD Spektral Testi P-değeri	PD Spektral Testi H değeri	PD Spektral Testi P-değeri
1	0	0,063159	1	< 0,01
2	0	0,143008	1	< 0,01
3	0	0,029566	1	< 0,01

Eşlik biti ile çıktı alma yönteminde, üç üreticinin Spektral testinden elde edilen P-değerlerine Ki-Kare testi uygulandığında; Şekil 4.16'da sırası ile yer alan üretic-1, üretic-2 ve üretic-3'ün P-değerlerinin histogramları elde edilmektedir. Bu histogramlardan ve Çizelge 4.8'deki P-değerleri üzerinde yapılan Ki-Kare testi sonuçlarından görüldüğü gibi üç üretic de Spektral testten geçmiştir.



Şekil 4.16 BBS Algoritması Strateji-2 Spektral Testi Histogramı
(Eşlik Biti Çıktı Yöntemi)

Normal çıktı yönteminde ise Şekil 4.17 ve Çizelge 4.8'de görüldüğü gibi üç üretic de Spektral testten geçememiştir. Bu durum, P-değeri dağılımı incelenirken eşlik biti yönteminde, P-değerlerinin birörnek dağılıma sahip olmadığına dair bir göstergeye rastlanmazken, normal çıktı alma yönteminde birörnek dağılıma sahip olmadığına dair bir göstergeye rastlanmıştır.



Şekil 4.17 BBS Algoritması Strateji-2 Spektral Testi Histogramı

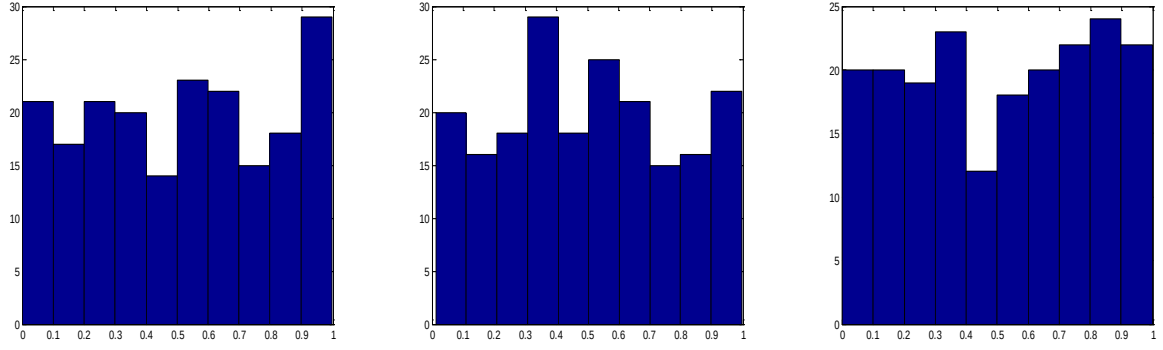
Çift eşlik biti ile çıktı alma yöntemi ile normal çıktı alma yöntemi arasındaki en büyük fark, üç üreticinin 200 örnekleme yapılan testlerde ortaya çıkmıştır. Çift eşlik biti yöntemi ile çıktı alındığında üç üreticinin örneklemeleri Spektral testten geçerken, normal çıktı yönteminde üç üreticinin de bu testten kalmıştır. Buna ek olarak Şekil 4.16 ve Şekil 4.17’de üreticinin sırasıyla yer alan histogram grafikleri kıyaslandığından iki yöntem arasındaki farkın Spektral test sonucunda ne kadar belirginleştiği görülmektedir. Histogram grafikleri ve çizelgeden anlaşıldığı gibi çift eşlik biti ile çıktı alma yöntemi uygulandığında üreticiler tarafından istatistiksel özellikleri daha iyi olan, testlerden başarı ile geçen çıktılar üretildiği gözlemlenmiştir.

Üç üreticinin 200 örneklemin P-değerleri üzerine yapılan Ki-Kare testi sonuçları Çizelge 4.9’da yer almaktadır.

Çizelge 4.9 BBS Algoritması Strateji-2 Ki-Kare Testi

Üreteç	PD Ki-Kare Testi H değeri	PD Ki-Kare Testi P-değeri
1	0	0,564377
2	0	0,434758
3	0	0,830877

Ki-Kare testi sonuçları ve P-değerlerinin histogramları birlikte değerlendirildiğinde; Şekil 4.18’de üç üreticinin için sırasıyla yer alan histogramlarda iniş-çıkışlar gözlemlense de, Çizelge 4.9’dan da anlaşıldığı gibi üç üreticinin de Ki-Kare testinden geçmiştir. Bu durumda P-değerlerinin bir örnek dağılıma sahip olduğu değerlendirilmektedir.



Şekil 4.18 BBS Algoritması Strateji-2 Ki-Kare Testi Histogramı

Çizelge 4.10 ve Çizelge 4.11’de görüldüğü gibi her iki çıktı alma yöntemi ile Strateji-2’ye uygulanan Frekans, Koşum, Spektral ve Ki-Kare test sonuçları bir bütün halinde verilmiştir.

Çizelge 4.10 BBS Algoritması Strateji-2 Uygulanan Testlerin Sonuçları
(Eşlik Biti Çıktı Yöntemi)

Üreteç	P_Değeri HT-ki- kare	P_Değeri p_değeri- ki-kare	P_Değeri HT-frek	P_Değeri p_değeri- frek	P_Değeri HT- koşum	P_Değeri p_değeri- koşum	P_Değeri HT-spek	P_Değeri p_değeri- spek
1	0	0,564377	0	0,956355	0	0,490529	0	0,063159
2	0	0,434758	0	0,791781	0	0,806942	0	0,143008
3	0	0,830877	0	0,837896	1	0,004349	0	0,029566

Eşlik biti ile çıktı alma yöntemi ile üreteçlerden çıktı alınarak örneklemeline testler yapıldığında Çizelge 4.10’da görüldüğü gibi sadece 3. üreteç Koşum testinden kalmıştır. Normal çıktı alma yöntemi ile üreteçlerden çıktı alınarak örneklemeline testler yapıldığında ise Çizelge 4.11’de görüldüğü gibi üç üreteçinde Spektral testten kaldığı gözlemlenmiştir. Bunun dışındaki testlerin tümünden üç üreteç de geçmiştir.

Çizelge 4.11 BBS Algoritması Strateji-2 Uygulanan Testlerin Sonuçları

Üreteç	P_Değeri HT-ki- kare	P_Değeri p_değeri- ki-kare	P_Değeri HT-frek	P_Değeri p_değeri- frek	P_Değeri HT- koşum	P_Değeri p_değeri- koşum	P_Değeri HT-spek	P_Değeri p_değeri- spek
1	0	0,564377	0	0,547793	0	0,280484	1	< 0,01
2	0	0,434758	0	0,021174	0	0,074812	1	< 0,01
3	0	0,830877	0	0,209027	0	0,069327	1	< 0,01

Çift eşlik biti ile çıktı alma yöntemi ile normal çıktı alma yöntemi arasındaki en büyük fark üç üretcin 200 örnekleme yapılan testlerde ortaya çıktığını çizelgeleri incelenirken belirtilmiştir. Normal çıktı alma yönteminde üç üretcin 200 örnekleme yapılan Spektral testten geçilememesine rağmen, tüm veriye yapılan Spektral testten başarı ile geçilmiştir.

Örneklemlerin genelde tüm verinin istatistiksel özelliklerini yansıtmaması beklenir. Ancak bu durumda da gözlemlendiği gibi örneklemler tüm verinin istatistiksel özelliklerini yansıtmayabilir. Örneklemlere yapılan testin sonucu, örneklemin büyüklüğü ve uzunluğuna bağlı olarak değişebilir [Hellekalek, 1995].

.

4.2 Doğrusal Eşleniksel Üreteç (LCG)

Doğrusal Eşleniksel Üreteç (LCG: Linear Congruential Generator) en eski ve en bilinen sözde rastsal sayı üretici algoritma olup 1951 yılında D. H. Lehmer tarafından geliştirilmiştir. LCG algoritmasının dayandığı teorem nedeni ile anlaşılması ve uygulaması kolay bir algoritmadır.

LCG algoritması matematiksel olarak Eş. 4.24'te yer aldığı şekilde ifade edilir [Challis, 2008]. Belirtilen eşitlikten de anlaşıldığı gibi bir durumun çıktısı, diğer durumun girdisi olduğundan tekrarlı bir ilişki yapısı mevcuttur. LCG algoritmasının doğrusal bir üreteç olduğu görülmektedir.

$$x_{i+1} = ax_i + c \pmod{m} \quad (4.24)$$

$$u_i = \frac{x_i}{m} \quad (4.25)$$

Parametrelerin tanımları ve değer aralıkları;

- x_0 , $0 \leq x_0 < m - 1$ → Başlangıç değeri
- m , $0 < m$ → Mod
- a , $0 < a < m$ → Çarpan
- c , $0 \leq c < m$ → Artan
- x_i , $0 \leq x_0 < m - 1$ → sözde rastsal sayı dizisi
- u_i , bir örnek sözde rastsal sayı dizisi olarak tanımlanmaktadır [Mills, 2003].

Algoritmanın başarısını ve periyodunu m , a , c ve x_0 parametreleri belirlemektedir. LCG algoritması parametre ve örneklem seçimine karşı çok hassastır [Hellekalek, 1994]. Uygun seçilen parametreler doğrultusunda maksimum periyot elde edilebilir. Bu kapsamda m değerinin büyük bir asal sayı seçilmesi gerekmektedir. LCG algoritmasının verimli olması ve Eş. 4.24'teki denkleminin bilgisayar üzerindeki uygulamasının kolay ve hızlı olması bakımından, m değerinin 2'nin üssü seçilmesi gerekmektedir [L'Ecuyer, 1999].

LCG algoritması artan (c) parametresinin seçimine göre iki tip sayı üretici algoritmaya ayrılır.

1. Çarpımsal Doğrusal Eşleniksel Üreteç (MLCG)
2. Karışık Eşleniksel Üreteç (MCG)

4.2.1 Çarpımsal Doğrusal Eşleniksel Üreteç (MLCG)

LCG denkleminde $x_{i+1} = ax_i + c \pmod{m}$ artan olarak tanımlı parametrenin, $c = 0$ eşit olduğu durumda Eş. 4.26'da yer aldığı şekilde ifade edilen sayı üretici, Çarpımsal Doğrusal Eşleniksel Üreteç, Lehmer RNG ya da Park-Miller olarak tanımlanmaktadır [Challis, 2008].

$$x_{i+1} = ax_i \pmod{m} \quad (4.26)$$

Bu üreteç kullanılarak üretilecek sayıların alabileceği değer aralığı $[0, m-1/m]$ olarak tanımlanmaktadır.

1981 yılında Knuth, 1987 yılında Bratley, 1988 yılında Park ve Miller, 1990 yılında L'Ecuyer, 1992 yılında Niederreiter tarafından konu ile ilgili çalışmalar yapılmış ve maksimum periyoda ulaşılması için uygun parametre setleri önerilmiştir.

Uygun parametre seçimi ile periyot elde edildiğinde MLCG algoritması için maksimum periyot $m-1$ değerine eşittir. İyi bir MLCG algoritması için m parametresinin 2^{31} , 2^{47} ve 2^{63} sayılarına yakın seçilmesi önerilmektedir [L'Ecuyer, 1999]. M değeri büyük bir asal seçildiğinde ve sözde rastsal dizinin elemanları arasındaki ilintinin az olduğu durumda, MLCG algoritması rastsal sayı üreticisi olarak benzetim uygulamalarında kullanılabilir. MLCG basit, uygulaması kolay ve hızlı koşan bir algoritmadır [L'Ecuyer, 1993].

Park-Miller algoritmasında, artan değer $c=0$ eşitliğinde olmasına ek olarak sözde rastsal sayı üretici algoritmalar için "Minimum Standart" yani uyulması gereken minimum koşullar önermiştir [Li, 2012]. Park-Miller ile maksimum periyoda ulaşmak için kritik parametreler; $m=2^{31}-1=2147483647$ ve $a=7^5=16807$ şeklinde seçilmiştir [Park, 1988].

Park-Miller algoritmasında önerildiği ve genel olarak m parametresi için tercih edilen $2^{31}-1$ değerinin seçilmesinin nedeni, $2^{31}-1$ sayısının bir Mersenne asal sayılarından olmasıdır [L'Ecuyer,1999]. Uygun parametreler ile çalışıldığında birörnek dağılıma sahip sözde rastsal sayı dizisinin elde edilebildiği görülmektedir.

MATLAB programlama dilinin 5. versiyonu çıkana kadar (1995) rastsal sayı üretimi için Park-Miller algoritması kullanılmıştır. 5. versiyondan itibaren “subtract with borrow” sayı üretici kullanılmaya başlanmıştır. “Subtract with borrow” algoritması, Park-Miller algoritmasına göre daha hızlı ve daha uzun periyoda sahip bir algoritma olması nedeni ile tercih edilmiştir [Mills, 2003].

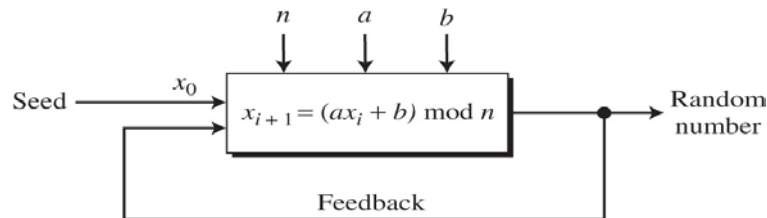
4.2.2 Karışık Eşleniksel Üreteç (MCG)

LCG algoritması Eş. 4.24 denkleminde artan olarak tanımlı parametrenin $c \neq 0$ olduğu durumda, sayı üretici Karışık Eşleniksel Üreteç olarak tanımlanmaktadır [Challis, 2008].

$$x_{i+1} = ax_i + c \pmod{m} \quad (4.24)$$

Bu üreteç kullanılarak üretilecek sayıların alabileceği değer aralığı $[1/m, (m-1)/m]$ olarak tanımlanmaktadır. Uygun parametre seçimi ile periyot elde edildiğinde MCG algoritması için maksimum periyot m değerine eşittir.

Bu tezde MLCG ve MCG algoritması, LCG algoritmasının altında parametre seçimlerini farklılaştırarak birlikte ele alınmıştır. Her iki üreteç için önerilen parametre setleri denenmiş; ancak her iki algoritma LCG sayı üreticisi olarak ele alınmıştır. LCG algoritmasının fiziksel yapısı aşağıdaki gibidir.



Şekil 4.19 LCG Fiziksel Gösterimi

LCG algoritmasında; a, c ve m parametrelerinin seçimi periyodu belirlediğinden ve maksimum periyota sahip bir sonuç elde etmek için a, c ve m kritik parametrelerin uygun seçilmesi gerektiğinden bahsedilmiştir. LCG algoritmasının maksimum periyotta sayı üretmesi için gereken tasarım kriterleri aşağıda belirtilmiştir.

- 1) $\gcd(c,m) = 1$, artan ve m değeri göreceli asalıdır.
- 2) $a - 1$, m değerinin tüm asal çarpanlarına bölünebilir olmalıdır.
- 3) m değeri 4 sayısının katı ise $a - 1$ de 4 sayısının katı olmalıdır.
- 4) c, artan, değerinin sıfır seçilmediği durumlarda tek sayı seçilmesi önerilmektedir.
- 5) m değerinin asal sayı olması önerilmektedir.

Örnek,

$x_{i+1} = 13x_i + 5 \pmod{18}$ ve $x_0 = 7$, $a = 13$, $c = 5$, $m = 18$ olarak belirlenmiştir.

$x_1 = 13 \cdot 7 + 5 \pmod{18} = 6$ başlangıç değerinin kullanılması ile elde edilen dizi elemanıdır.

$x_2 = 13 \cdot 6 + 5 \pmod{18} = 11 \dots$ her durumda elde edilen değerler ile yaratılan dizi aşağıdaki gibidir.

7,6,11,4,3,8,10,5,16,15,2,13,12,17,10,9,14,7,6,11,4,...
periyot=18=m

- 1) $\gcd(5,18) = 1$ ile 1. koşul sağlanmıştır.
- 2) $a - 1 = 12$, 18'in asal çarpanları olan 2 ve 3 sayılarına bölünebilir. Bu sayede 2. koşul da sağlanmıştır.
- 3) $m = 18$, 4 'e bölünmediği için $a - 1$ 'inde 4'e bölünmesi beklenmez.
- 4) $c = 5$ tek sayı seçilerek öneriye uyulmuştur.
- 5) $m=18$ asal değil ancak bu koşul bir öneri olduğu için sorun teşkil etmemektedir.

Görüldüğü gibi koşulların sağlanması ile maksimum periyoda ($m=18$) sahip dizi elde edilmiştir [Glen, 2002].

LCG algoritmasının tahmin edilebilir olduğunu John von Neumann ortaya koymuştur. LCG algoritmasının tahmin edilebilir olduğu ilk olarak Jim Reeds daha sonra da Joan Boyar tarafından kanıtlanmıştır [Li, 2012]. LCG algoritmasının ürettiği sözde rastsal sayı dizilerinin elemanları arasında ilinti vardır. Üretilen sayılara Spektral test ile bakıldığında kafes yapısı gözlenmektedir [Eichenauer, 1986]. 2-çokuzlu olarak bakıldığında rastsal görünen sayılar, 3- çokuzlu ile çizdirildiğinde rastsal olmadıkları gözlenebilir. Algoritma için dezavantaj olan bu özelliğe rağmen algoritmanın kullanım alanı oldukça geniştir. Kritik parametre seçimine göre uzun periyoda sahip istatistiksel testlerde iyi performans veren LCG kombinasyonları, bazı benzetimlerde kullanılabilir; ancak Monte Carlo benzetimlerinde kullanımı ilinti nedeni ile uygun değildir. İlintili yapısı nedeni ile şifreleme uygulamaları için kullanımı uygun değildir [Challis, 2008]. Bu tip uygulamalar için BBS gibi kriptografik olarak güvenli sözde rastsal sayı üretici algoritmaları tercih etmek daha uygundur. Çünkü LCG, BBS algoritması gibi polinomsal zamanda tahmin edilemez değildir, başlangıç değerine ulaşılabilir [Frieze, 1984].

4.2.3 LCG Algoritmasının Uygulanması

LCG sözde rastsal sayı üretme algoritması hakkında yukarıdaki bölümde detaylı olarak bilgi verilmiştir. LCG algoritmasının kodlanması, MATLAB programlama dili ile gerçekleştirilmiştir. LCG algoritması kodlanma sırasında yapılan kontroller ve algoritma adımları aşağıda belirtilmiştir.

$$x_{i+1} = ax_i + c \pmod{m} \quad (4.24)$$

- 1) a ve c değerlerinin uzunluklarının eşit olması beklenir. Bu koşul sağlanmıyorsa ekranda hatalı girdi olduğuna dair uyarı çıkar.
- 2) m değeri 4 sayısının katı ise a-1 de 4 sayısının katı mı kontrol edilir. Bu koşul sağlanmıyorsa ekranda koşulun sağlanmadığı uyarısı çıkar.
- 3) m değeri faktörlerine ayrılarak birbirinden farklı çarpanları elde edilir.
- 4) Elde edilen birbirinden farklı çarpanlar ile a-1, m değerinin tüm asal çarpanlarına bölünebilir mi kontrol edilir.

5) $a - 1$, çarpanların hepsine bölünebiliyorsa, maksimum periyot elde edilir. Maksimum periyot hesabı aşağıdaki gibidir.

$c > 0 \rightarrow$ maksimum periyot = m

$c = 0 \rightarrow$ maksimum periyot = $m-1$

6) Maksimum periyodu sağlamayan a , c ve m değerleri için başlangıç değeri ile elde edilen sonraki değerlere (x_{next}) bakılır. Başlangıç değeri ile sözde rastsal sayı dizisindeki elemanın aynı olup olmadığı her durum (i) için kontrol edilir. Başlangıç değeri ile sonraki değer (x_{next}) aynı olduğu durumda aralarındaki iterasyon sayısı periyodu vermektedir.

7) Sözde rastsal sayı dizinin oluşturulması için elde edilen x_n sayısının onluk tabandaki her bir basamağı ikilik sisteme çevrilir ve en önemsiz bitleri XOR işleminden geçirilir. Bu sayede x_n değeri için sözde rastsal sayı dizinin bir çıktısı elde edilir.

8) Testler iki stratejiye ayrılarak yapılır. Strateji-1 de üretici tarafından üretilen tüm veriye, 1000000 uzunluğundaki veri, testler yapılırken; Strateji-2 de ise bölünmüş veriye yani 5000 örneklem dizi uzunluğuna sahip 200 örnekleme test yapılır.

4.2.4 Uygulanan Testlerin Sonuçları ve Değerlendirmeler

Tüm üreteçlere

- Ki-Kare Testi,
- Frekans Testi,
- Koşum Testi,
- Ayırık Fourier Dönüşümü (Spektral) Testi

yapılarak elde edilen test sonuçları çizelge ve grafik halinde bu bölümde sunulmuş ve test sonuçlar yorumlanmıştır.

Bu testlerin sıfır hipotezini gösteren h ve P - değerleri incelenir. $h = 0$ ise sıfır hipotezinin kabul edildiğini gösterir. Yapılan testler anlam düzeyi (α) olarak %1 seçilmiştir. Bu nedenle P -değeri ≥ 0.01 ise üretilen dizi rastsaldır.

Örnek olarak seçilen üç üretece belirtilen testler uygulanmıştır. Algoritma girdisi olarak 3 üretcin parametreleri verilmiştir.

1. Üreteç $[a \ c \ m \ x_0] = [65539 \ 0 \ 2^{31} \ 1]$ [Challis, 2008]
2. Üreteç $[a \ c \ m \ x_0] = [4561 \ 51349 \ 243000 \ 5]$ [Scheiner, 1996]
3. Üreteç $[a \ c \ m \ x_0] = [7141 \ 54773 \ 2^{31} \ 5]$ [Schneier, 1996]

IBM tarafından 1968 yılında geliştirilen ve uygulanan sözde rastsal sayı üretme algoritması RANDU, 1. üreteç olarak seçilmiştir. RANDU üretcinin periyodu $2^{29} = 0.25 \ m$ olarak bilinmektedir. RANDU oldukça güçsüz ve dizi elemanları arasında ilinti olduğu bilinen bir üreteçtir. Bu özelliklerini gözlemek ve parametre seçiminin önemi göstermek amacıyla RANDU örnek olarak seçilmiştir [Challis, 2008].

Algoritma çıktısı olarak sözde rastsal sayı dizisi alınmıştır.

Test girdileri olarak;

- Algoritma çıktısı
- Üretilen dizi uzunluğu : 1000000
- Örneklemdaki dizi uzunluğu : 5000
- Örnekleme sayısı : 200
- $\alpha = 0.01$ değerleri seçilmiştir.

Tüm algoritmalarda aynı çıktı elde etme yöntemi ile çıktılar alınmış ve bu çıktılar testlerden geçirilmiştir. LCG algoritması üç üretci için test sonuçları aşağıdaki çizelge ve grafikler ile verilmiştir.

LCG algoritmasına ait üreteçlerin maksimum periyoda sahip diziler üretip üretmedikleri kontrol edilmiş; maksimum periyoda sahip diziler ise max_flag=1 değerine, değil ise max_flag=0 değerine eşitlenmiştir.

4.2.4.1 Strateji-1 Testleri

Seçilen üreteç ile üretilen 1000000 dizi uzunluğundaki veriye Frekans, Koşum, Spektral ve Ki-Kare testleri yapılmıştır; test sonuçları Çizelge 4.12 ve Çizelge 4.13'te yer almaktadır.

Çizelge 4.12 LCG Algoritması Strateji-1 Test Sonuçları-1

Üreteç	a	c	M	x_0	Periyot	Max Flag	h_ki-kare	p_ki-kare
1	65539	0	2147483648	1	536870912	0	0	0,861132
2	4561	51349	243000	5	243000	1	0	1
3	7141	54773	2147483648	5	2147483648	1	0	0,596686

Seçilen üreteçlerin arasında en büyük periyoda sahip olan 3. üreteçtir. Max_flag parametresi ile üreteçlerin maksimum periyoda sahip diziler üretip üretmediğini göstermektedir. Bu kapsamda 2. ve 3. üreteçlerin maksimum periyoda sahip dizi ürettikleri gözlenirken 1. üreticinin maksimum periyoda ulaşamadığını anlaşılmaktadır.

Tüm testlerden elde edilen h ve P-değerleri incelenmiştir. Anlam düzeyi (α) %1 seçildiği durumda Çizelge 4.12 ve Çizelge 4.13'te görüldüğü gibi 1. ve 3. üreteçler ile üretilen verinin tümüne uygulanan; Frekans, Koşum, Spektral ve Ki-Kare testlerinin h değerleri üç üreteç için de sıfırdır. 2. üreteç de ise Spektral test dışındaki testlerin h değerleri sıfırdır. Testlerin sonucunda h=0 olması, sıfır hipotezinin kabul edildiğini göstermektedir. Buna göre LCG algoritmasına ait 1. ve 3. üreticinin ürettiği dizilerin tüm testlerden geçtiği, 2. üretece ait dizinin ise Spektral test dışındaki testlerden geçtiği gözlenmiştir.

Çizelge 4.12 ve Çizelge 4.13'te yer alan test sonuçlarına göre üç üreticinin ürettiği dizilerin rastsal olduğu değerlendirilmektedir. CDF ve histogram grafikleri incelenerek nihai bir sonuca varılabilecektir.

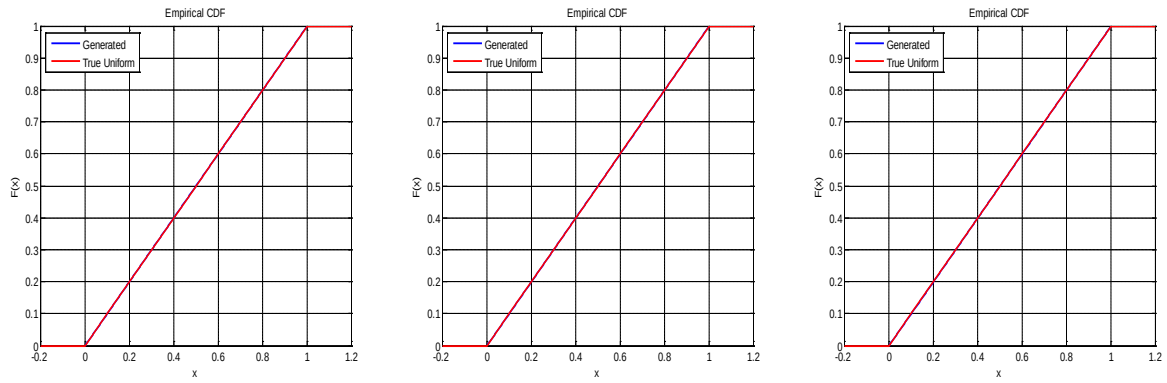
Çizelge 4.13 LCG Algoritması Strateji-1 Test Sonuçları-2

Üreteç	h_fre	p_fre	h_koşum	p_koşum	h_spek	p_spek	ortalama	varyans
1	0	0,8855	0	0,130783	0	0,383329	0,499662	0,08341
2	0	0,802587	0	0,488349	1	< 0,01	0,499965	0,08333
3	0	0,7414	0	0,724007	0	0,440804	0,500286	0,083233

Birörnek dağılıma sahip bir rastsal dizinin ortalamasının 0.5, varyansının da 0.08 olması gerektiği bilinmektedir [Gentle, 2003]. Her iki çıktı yöntemi ile üç üreticinin ürettiği dizilerin ortalama ve varyans değerlerine bakarak birörnek dağılıma sahip

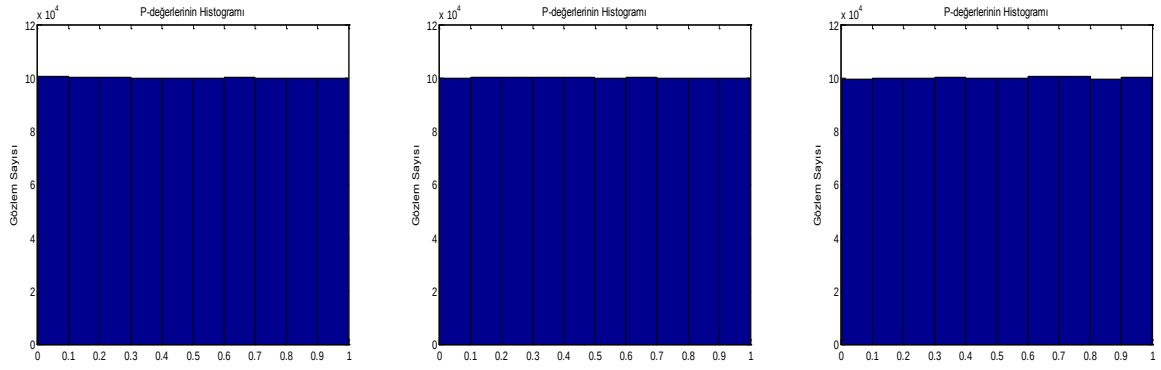
olup olmadıkları incelenmiştir. Çizelge 4.13'te görüldüğü gibi üreteçlere ait ortalama ve varyans değerlerinin, 0.5 ve 0.08 civarında olduğu ve çok küçük farklılıklar sergiledikleri gözlenmektedir.

CDF grafiği ile her üreteçte üretilen dizinin (1000000 uzunluğunda) birörnek dağılımı ile ne kadar örtüştüğünü gözlemek amaçlanır. Şekil 4.20'de sırası ile yer alan üreteç-1, üreteç-2 ve üreteç-3'e ait verilerin, birörnek dağılıma ait veri ile tam olarak örtüştüğü gözlenmektedir. Birörnek dağılıma ait beklenen ve üreteçlere ait gözlenen veriler örtüştüğü için, üç üreticinin birörnek dağılıma sahip diziler oluşturduğu değerlendirilmektedir.



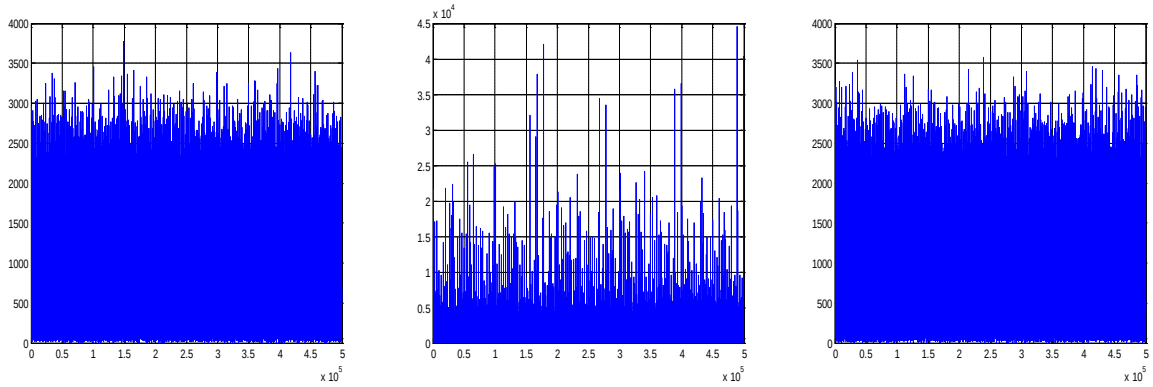
Şekil 4.20 LCG Algoritması Strateji-1 CDF Grafikleri

Şekil 4.21'de sırası ile yer alan üreteç-1, üreteç-2 ve üreteç-3 için üretilen tüm verinin, grafiklere yansıyan dalgalanmalarını gözlemleyip kıyaslayabilmek adına CDF ve histogram grafiklerinin birlikte gösterilmiştir. LCG algoritmasına ait üç üreteçte üretilen tüm verinin histogramında dalgalanma gözlenmemektedir. Çizelge 4.12'de yer alan Ki-Kare test sonuçlarından 2. ve 3. üreteçlerin birörnek dağılıma sahip rastsal diziler ürettikleri anlaşılmaktadır. Bu sonuç CDF ve histogram grafikleri ile de gözlenmiştir. 1. üreteç içinde benzer bir sonuç gözlemlense de Strateji-2 incelendiğinde, dizi elemanları arasında ilinti olduğu gözlenmiştir. Bu nedenle 2. üreticinin rastsal dizi ürettiği söylenemez.



Şekil 4.21 LCG Algoritması Strateji-1 Histogramı

Üç üreteç ile üretilen tüm verinin Spektral testleri yapılmıştır ve bu test sonuçları Çizelge 4.13'te verilmiştir. Çizelgede yer alan sonuçlara göre ve 3. üreteçlerin Spektral testten geçtiği, 2. üreticinin ise Spektral testten kaldığı bilinmektedir. Şekil 4.22'de üç üreteç için sırası ile yer alan FFT grafikleri; 1. ve 3. üretece ait dizilerin Spektral testten geçtiğini, 2. üretece ait dizinin ise Spektral testten kaldığını desteklemektedir.



Şekil 4.22 LCG Algoritması Strateji-1 FFT Grafikleri

4.2.4.2 Strateji-2 Testleri

Örneklem sayısı ve bir örneklemdeki dizi uzunluğu seçimi ile algoritma tarafından üretilen dizinin tümü kullanılarak testler gerçekleştirilmiştir. Bu kısımda üretilen tüm veri, örneklem sayısı ve bir örneklemdeki dizi uzunluğu göz önüne alınarak bölünmüştür. Bölünen veriler üzerinde Frekans, Koşum, Spektral ve Ki-Kare testleri yapılmıştır.

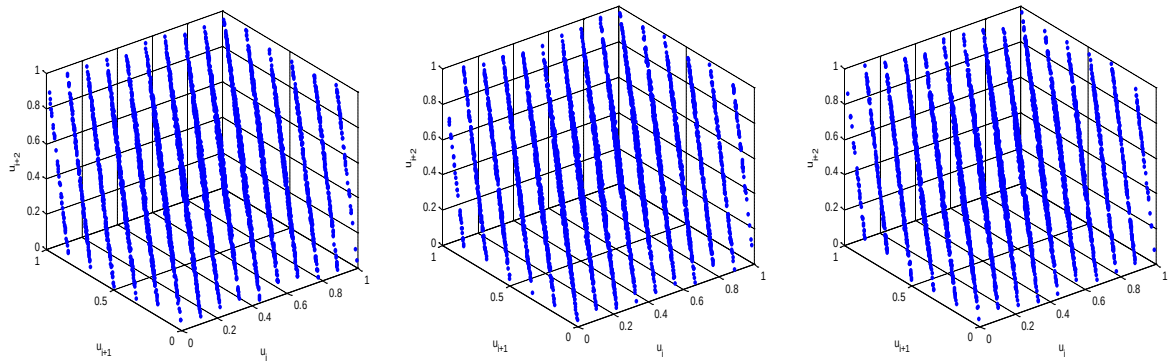
Üç üretcin 200 örnekleme yapılan testler sonucunda elde edilen P-değerinin analiz yapılır. P-değeri analizi kapsamında Dizi Geçme Oranı (DGO) hesaplanır. 200 örneklem arasında kaçının sıfır hipotezini geçtiğine dair bir oran elde edilir ve sonuç olarak testten geçip geçmediğine bakılır. Test sonuçları Çizelge 4.14'te yer almaktadır.

Çizelge 4.14 LCG Algoritması Strateji-2 Test Sonuçları-1

Üreteç	Ki- Kare DGO	Ki-Kare Test Sonucu	Frekans DGO	Frekans Testi Sonucu	Koşum DGO	Koşum Testi Sonucu	Spektral DGO	Spektral Test Sonucu
1	0,995	1	1	1	0,98	1	0,98	1
2	1	1	0,92	0	1	1	0,925	0
3	0,99	1	0,99	1	1	1	0,98	1

Çizelge 4.14'te görüldüğü gibi testi geçme oranları farklı olsa da test sonuçları 1 olduğundan; 1. ve 3. üretcin 200 örnekleme uygulanan Frekans, Koşum ve Spektral testlerin tümünden geçtiği, 2. üretcin ise Frekans ve Spektral testinden geçemediği gözlenmektedir.

Üç üretçten rastsal örneklemeler alınarak dizi elemanları arasında bir ilinti olup olmadığı, dizi elemanlarının periyodik özellikler sergileyip sergilemediğini incelemiştir. Dizi elemanları arasındaki ilinti, arka arkaya olan üç duruma ait veri (u_i, u_{i+1}, u_{i+2}) incelenerek tespit edilmeye çalışılmıştır. Rastsal olarak 48. , 115. ve 193. örneklemeleri seçilerek olan 3-çokuzlu grafikler üzerinden gözlem yapılmıştır. Şekil 4.23'te görüldüğü gibi 1. üretçte seçilen örneklemelerde dizi elemanları arasında ilinti vardır; farklı örneklemeler incelendiğinde ilinti gözlenmiştir.

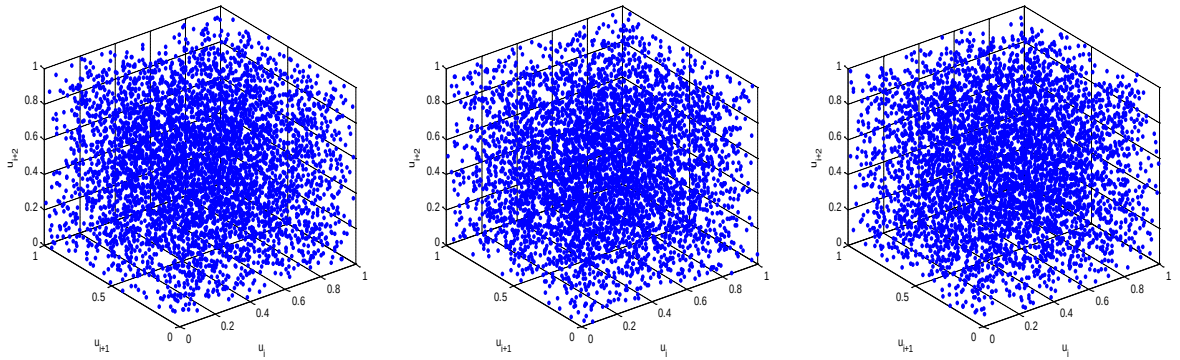


Şekil 4.23 LCG Algoritması Strateji-2, Üreteç-1 48. , 115. ve 193. Örneklemeleri

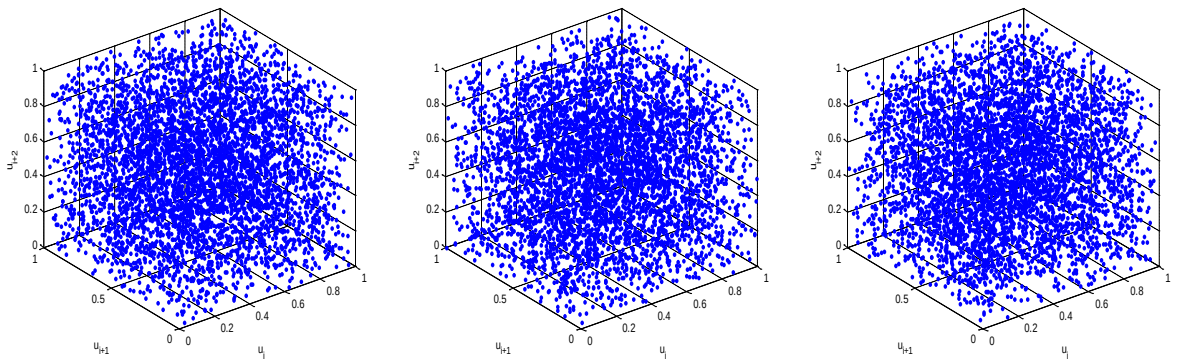
Diğer algoritmalarından farklı olarak LCG algoritmasının ilinti nedeni ile oluşan kafes yapısı bilinmektedir, bu yapı Şekil 4.23'teki grafikler ile gözlenmiştir. Bu nedenle 1. üretcin rastsal sayı dizi ürettiği söylenemez. Bu üretcin seçilmesindeki en önemli etken uygun parametre seçimi yapılmadığında tüm testlerden geçen, bir üretcin ilintili yapısı nedeni ile güvensiz olabileceğini ve kolay bir şekilde kırılacağını göstermektedir.

Tahmin edilebilirliğe neden olan kafes yapısından parametre seçimi ile uzak kalınmaya çalışılsa da, bu yönü nedeni ile LCG'nin diğerlerine göre daha zayıf bir algoritma olduğu anlaşılmaktadır.

Şekil 4.24 ve Şekil 4.25'te görüldüğü gibi 2. ve 3. üretcin rastsal seçilen hiçbir örnekleminde, dizi elemanları arasında bir ilinti ya da periyodik özellik gözlenmemiştir. Bu grafikler uygun parametre seçimi yapıldığında LCG algoritmasına ait üretçilerin dizi elemanları arasında ilintinin oluşmayacağını göstermektedir.



Şekil 4.24 LCG Algoritması Strateji-2, Üreteç-2 48. , 115. ve 193. Örneklemeleri



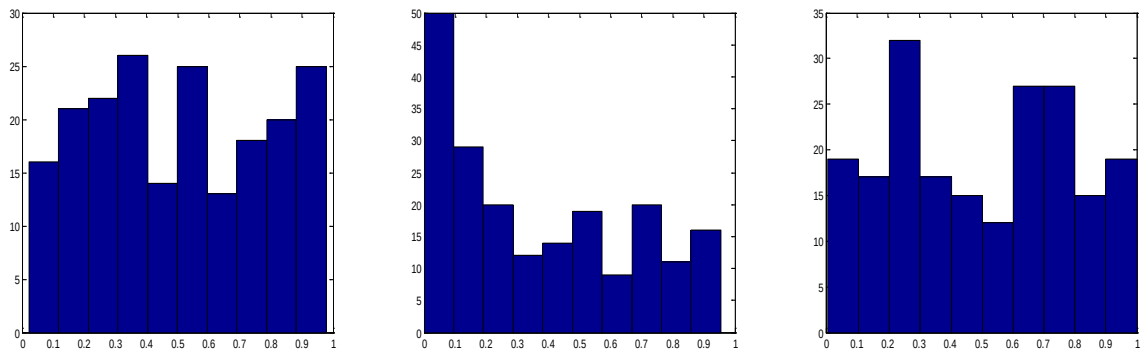
Şekil 4.25 LCG Algoritması Strateji-2, Üreteç-3 48. , 115. ve 193. Örneklemeleri

Tüm testler için P-değeri analizi kapsamında P-değerinin dağılımı hesaplanır. Her bir üretcin 200 örnekleme uygulanan Frekans, Spektral, Koşum ve Ki-Kare testleri sonucunda elde edilen P-değerlerine Ki-Kare testi uygulanarak P-değerlerinin dağılımına bakılır. LCG algoritmasına ait üreticilerin P-değerleri dağılımına bakıldığında, 1. ve 3. üretcin örneklemelerinin Frekans testinden geçtiği, 2. üretcin ise Frekans testinden kaldığı Çizelge 4.15'te görülmektedir.

Çizelge 4.15 LCG Algoritması Strateji-2 Frekans Testi

Üreteç	Normal Çıktı Yöntemi	
	PD Frekans Testi H değeri	PD Frekans Testi P-değeri
1	0	0,287901751
2	1	< 0,01
3	0	0,024662549

1. ve 3. üretcin için P-değerleri üzerinde yapılan Ki-Kare testi sonuçları ve P-değerlerinin histogramları birlikte değerlendirildiğinde; Şekil 4.26'da üreticilerle ilgili yer alan histogramlarda görülen iniş-çıkışların, P-değerlerinin birörnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturmadığı anlaşılmaktadır. 2. üreticinin için ise histogramda görülen iniş-çıkışların, P-değerlerinin birörnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturduğu anlaşılmaktadır.



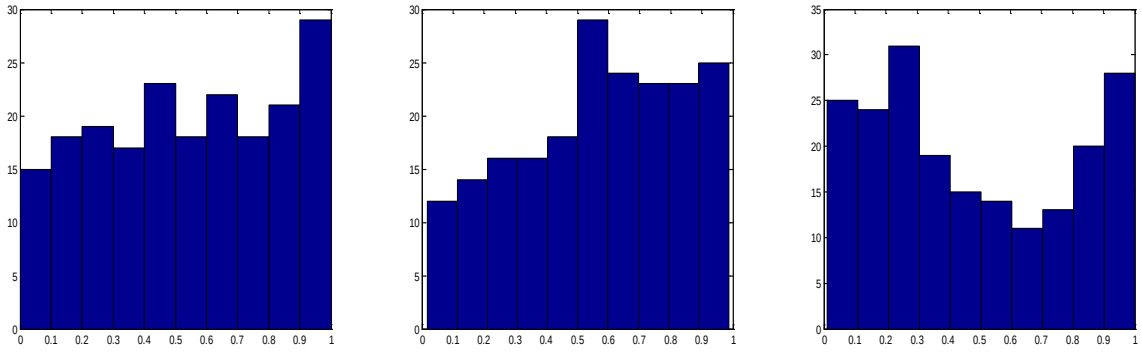
Şekil 4.26 LCG Algoritması Strateji-2 Frekans Testi Histogramı

Üç üretcin 200 örnekleme yapılan Koşum testinden elde edilen P-değerlerine, Ki-Kare testi uygulanmış; elde edilen test sonuçları Çizelge 4.16'da verilmiştir. Bu çizelgedeki sonuçlara göre üç üretcin de Koşum testinden geçtiği görülmektedir.

Çizelge 4.16 LCG Algoritması Strateji-2 Koşum Testi

Üreteç	Normal Çıktı Yöntemi	
	PD Koşum Testi H değeri	PD Koşum Testi P-değeri
1	0	0,623679
2	0	0,077883
3	0	0,016592

P-değerleri üzerinde yapılan Ki-Kare testi sonuçları ve P-değerlerinin histogramları birlikte değerlendirildiğinde; Şekil 4.27’de sırası ile yer alan üreteç-1, üreteç-2 ve üreteç-3’e ait histogramlarda iniş-çıkışlar gözlense de, Çizelge 4.16’dan da anlaşıldığı gibi üç üreteç de Koşum testinden geçmiştir. Bu durumda üç üreteç için de P-değerlerinin birörnek dağılıma sahip olduğu değerlendirilmektedir.



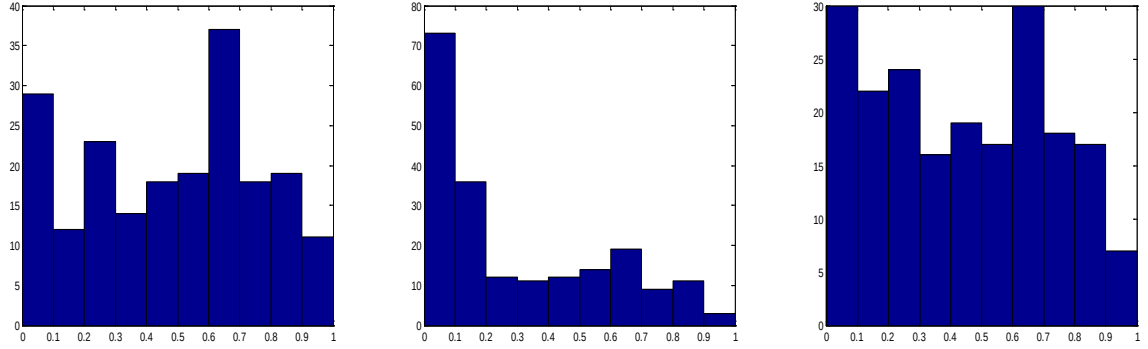
Şekil 4.27 LCG Algoritması Strateji-2 Koşum Testi Histogramı

Üç üreticinin 200 örnekleminin Spektral testten elde edilen P-değerleri üzerine yapılan Ki-Kare testi sonuçları Çizelge 4.17’de yer almaktadır.

Çizelge 4.17 LCG Algoritması Strateji-2 Spektral Testi

Üreteç	Normal Çıktı Yöntemi	
	PD Spektral Testi H değeri	PD Spektral Testi P-değeri
1	1	< 0,01
2	1	< 0,01
3	0	0,013657

Şekil 4.28'deki P-değerlerinin histogramlarından ve Çizelge 4.17'deki P-değerleri üzerinde yapılan Ki-Kare testi sonuçlarından anlaşıldığı gibi; 1. ve 2. üreteç Spektral testten kalmış, 3. üretcin ise testten geçmiştir. Bu durumda, 1. ve 2. üreteçler için P-değerlerinin birörnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturduğu anlaşılmaktadır. 3. üreteç için P-değerlerinin birörnek dağılıma sahip olduğu değerlendirilmektedir.



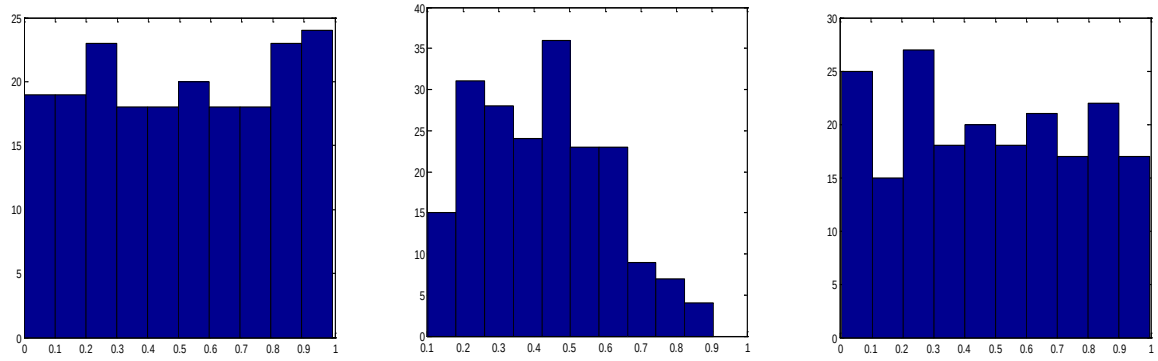
Şekil 4.28 LCG Algoritması Strateji-2 Spektral Testi Histogramı

LCG algoritmasına ait üreteçlerin P-değerleri üzerinde yapılan Ki-Kare testinin sonuçları Çizelge 4.18'de verilmiştir.

Çizelge 4.18 LCG Algoritması Strateji-2 Ki-Kare Testi

Üreteç	Normal Çıktı Yöntemi	
	PD Ki-Kare Testi H değeri	PD Ki-Kare Testi P-değeri
1	0	0,989757916
2	1	< 0,01
3	0	0,72698459

Çizelge 4.18'de yer alan test sonuçları ve Şekil 4.29'da sırası ile yer alan üreteç-1, üreteç-2 ve üreteç-3'e ait P-değerlerinin histogramları incelendiğinde; 1. ve 3. üretcin ise Ki-Kare testten geçtiği, 2. üretcin ise testten kaldığı gözlemlenmektedir. 1. ve 3. üreteç için P-değerlerinin birörnek dağılıma sahip olmadığına dair, 2. üretcin ise P-değerlerinin birörnek dağılıma sahip olduğuna dair gösterge bulunamamıştır.



Şekil 4.29 LCG Algoritması Strateji-2 Ki-Kare Testi Histogramı

Strateji-2'ye uygulanan Frekans, Koşum, Spektral ve Ki-Kare test sonuçları bir bütün halinde Çizelge 4.19'da verilmiştir.

Çizelge 4.19 LCG Algoritması Strateji-2 Uygulanan Testlerin Sonuçları

Üreteç	P_Değeri HT- ki-kare	P_Değeri p_değeri- ki-kare	P_Değeri HT-frek	P_Değeri p_değeri- frek	P_Değeri HT- koşum	P_Değeri p_değeri- koşum	P_Değeri HT-spek	P_Değeri p_değeri- spek
1	0	0,9897579	0	0,28790175	0	0,623679	1	< 0,01
2	1	< 0,01	1	< 0,01	0	0,077883	1	< 0,01
3	0	0,7269845	0	0,02466254	0	0,016592	0	0,013657

Çizelge 4.19'da görüldüğü gibi 3. üreticinin 200 örnekleme yapılan testlerden elde edilen P-değerlerine Ki-Kare testi uygulandığında; P-değerlerine uygulanan Ki-Kare testinden başarı ile geçmiştir. Ancak 1. üreticinin Spektral testten, 2. üreticinin ise Koşum testi dışındaki testlerin tümünden kalmıştır.

2. üreticinin 200 örnekleme yapılan testlerden elde edilen P-değerlerinin birörnek dağılıma sahip olmadığı, 3. üreticinin P-değerlerinin birörnek dağılıma sahip rastsal diziler ürettiği değerlendirilmektedir.

Strateji-1 de 2. üreticinin değerlendirildiğinde, üreticinin tarafından üretilen tüm verinin rastsal bir dizi olduğu CDF ve histogram grafiklerine bakarak söylenebilir. Ancak Strateji-2 de 2. üreticinin örneklemelerinin, birörnek dağılıma sahip rastsal diziler olmadığı değerlendirilmektedir. 2. üreticinin tüm verisine yapılan testte tüm verinin birörnek dağılıma sahip rastsal diziden oluştuğu gözlenmektedir.

1. üret  te g  zlenen ilintili yapısından nedeniyle rastsal bir dizi olu  turmadı  ı anla  ılmı  tır. Buna ek olarak. dizi elemanları arasındaki ilintili yapıdan dolayı tahmin edilebilir oldu  u g  zlemlenmi  tir. 1. üret    ile   retilen dizilerin maksimum periyoda ula  maması da   retec  n yeterince g     l   olmadı  ını g  stermektedir. Bu   rnek ile dolay   uygun parametreler se  ilmedi  inde dizinin tekrarı, ilintili ve g  vensiz bir yapıda olabilece  i g  zlenmi  tir.

3.   rete   ise hem   rneklemelere hem de t  m veriye uygulanan testler g    n  ne alındı  ında,   retec  n hi  bir testten kalmadı  ı, bir  rnek da  ılıma sahip rastsal ve maksimum periyoda sahip diziler   retti  in g  zlenmi  tir. Bu nedenle   rnek olarak se  ilen LCG   rete  lerinden en iyi istatistiksel   zelliklere sahip   rete   oldu  u s  ylenebilir. Parametre se  iminin hassas oldu  u bilinen LCG algoritmasında, parametrelere dayalı olarak g  zlenebilecek durumlar verilen   rete     rnekleri ile yansıtılmı  tır.

4.3 Ters Eşleniksel Üreteç (ICG)

Ters Eşleniksel Üreteç (ICG: Inverse Congruential Generator) 1986 yılında Lehn ve Eichenauer tarafından geliştirilmiştir. ICG algoritmaları bir çeşit doğrusal olmayan eşleniksel sözde rastsal sayı üreteçlerindendir. Modüler çarpım tersi olduğunda doğrusal olmama özelliği sağlanabilir. ICG algoritmasının doğrusal olmayan yapısı ile sayı dizisi üretilir. ICG algoritmasının denklemi Eş. 4.27 ile ifade edilmektedir [Hellekalek,1994].

- x_0 , $0 \leq x_0 < q$ → Başlangıç değeri
- q , $q \geq 3$ → Mod, asal sayı kullanılır.
- a , $0 < a < q$ → Çarpan
- c , $0 \leq c < q$ → Artan
- x_i , $0 < x_i < q$ → Sözde rastsal sayı dizisi
- x_i^{-1} , $x_i \bmod q$ 'nin çarpımsal tersi
- u_i , sözde rastsal sayı dizisi düzgelendiğinden $[0, 1]$ aralığında tanımlanmaktadır [Hellekalek,1995].

$$x_{i+1} \equiv \begin{cases} (ax_i^{-1} + c) \bmod q & , x_i \neq 0, \\ c & , x_i = 0, \end{cases} \quad i \geq 0 \quad (4.27)$$

$$u_i = \frac{x_i}{q} \quad (4.28)$$

ICG, doğrusal olmayan modüler çarpım tersinin alınması işlemi dışında LCG algoritmasına benzer bir algoritmadır. Ancak ICG algoritmalarında parametre ve örneklem seçimi LCG algoritmasında olduğu gibi hassas ve kritik değildir. LCG de örneklem seçimi dizinin istatistiksel özelliklerini yansıtmayabilirken, ICG algoritmalarında rastsal olarak seçilen örneklem dizinin istatistiksel özelliklerini yansıtmaktadır.

ICG algoritmasında a ve c parametrelerinin seçimi, periyot açısından LCG algoritmasında olduğu kadar kritik değildir. Uygun parametre seçimi ile periyot elde

edildiğinde, ICG algoritması için ulaşılabilecek maksimum periyot değeri q asal sayısına eşittir. ICG algoritmasının maksimum periyotta sayı üretebilmesi için Eş. 4.27’de yer alan denklemin, asal polinom $f(x) = x^2 - cx - a \in \mathbb{F}_q[x]$ olarak ifade edilebilmesi gerekir [Eichenauer, 1991]. Bu tip polinomlara Ters Maksimum Periyot (IMP) polinomlar denir.

Örnek,

$$x_{i+1} \equiv 2x_i^{-1} + 3 \pmod{7} \text{ ve } x_0 = 1, a = 2, c = 3, q = 7$$

$$x^2 - 3x - 2 \equiv x^2 + 4x + 5 \pmod{7}, \mathbb{F}_7 \text{ asal polinomu bu şekilde ifade edilebilir.}$$

ICG tarafından yaratılan dizi aşağıdaki gibidir.

$$\underbrace{1, 5, 2, 4, 0, 3, 6}_{\text{periyot}=7=q}, 1, 5, \dots$$

Asal polinom koşulunun sağlanması ile maksimum periyoda= $q=7$ sahip dizi elde edilmiştir [Glen, 2002].

ICG algoritmalarında kafes yapısının gözlenmediği, Eichenauer-Herrmann ve Lehn (1986), Grothe ve Niederreiter (1989) tarafından yapılan çalışmalar ile gösterilmiştir. ICG algoritmasının doğrusal olmayan güçlü yapısı sayesinde dizi elemanları arasında ilinti gözlenmemektedir; dolayısıyla kafes yapısı bulunmamaktadır [Lendl, 2001]. LCG algoritmasından en belirgin farkı doğrusal olmayan yapısı ile kafes yapısının oluşmamasıdır. Bu da ICG algoritmasının birörnek dağılıma sahip diziler ürettiği, üretilen dizi elemanları sayılar arasında ilintinin gözlenmediği anlamına gelmektedir [Gentle, 2003].

ICG algoritması modüler çarpım tersi işlemi nedeniyle LCG algoritmasına göre oldukça yavaş bir algoritmadır. Ancak hesaplama gücü ve verimliliği ile LCG algoritmasına karşı bir alternatiftir. Algoritmayı hızlandırmak için üç veya dört bağımsız kaynak kullanılarak uzun periyotlara sahip diziler üretilir. Büyük q asalı yerine küçük asal sayılar belirleyip, her kaynakta bir ICG algoritması koşturup sonra her biri için u_i hesaplanabilir [Melanson, 2010].

$$u_i = \left(\frac{x_i}{q_1} + \frac{y_i}{q_2} + \frac{z_i}{q_3} \right) \quad (4.29)$$

Örnek,

- x_i ICG (a=55, q=1031, c=1) , 1. kaynak için parametreler seçilmiştir.
 y_i ICG (a=103, q=1033, c=1) , 2. kaynak için parametreler seçilmiştir.
 z_i ICG (a=66, q=2027, c=1) , 3. kaynak için parametreler seçilmiştir.

$$u_i = \left(\frac{x_i}{1031} + \frac{y_i}{1033} + \frac{z_i}{2027} \right)$$

Periyotları ise üç ICG kaynağının çarpımıdır. Periyotları $1031 \times 1033 \times 2027 = 2158801621$ olup 2^{31} değerinden büyüktür.

4.3.1 ICG Algoritmasının Uygulanması

ICG sözde rastsal sayı üretme algoritması hakkında yukarıdaki bölümde detaylı olarak bilgi verilmiştir. ICG algoritmasının kodlanması, programlama dili olan MATLAB ile gerçekleştirilmiştir.

ICG algoritmasında kullanılan parametreler aşağıdaki gibi tanımlanmıştır. ICG algoritmasının kodlanması sırasında yapılan kontroller ve algoritma adımları modüler çarpım tersi işlemi dışında LCG algoritması ile aynıdır. Bu nedenle ayrıca belirtilmemiştir.

ICG algoritmasına uygulanacak testler iki stratejiye ayrılarak yapılır. Strateji-1 de üreteç tarafından üretilen tüm veriye, 1000000 uzunluğundaki veri, testler yapılırken; Strateji-2 de ise bölünmüş veriye yani 5000 örneklem dizi uzunluğuna sahip 200 örnekleme test yapılır.

4.3.2 Uygulanan Testlerin Sonuçları ve Değerlendirmeler

Tüm üreteçlere

- Ki-Kare Testi,
- Frekans Testi,
- Koşum Testi,
- Ayırık Fourier Dönüşümü (Spektral) Testi

yapılarak elde edilen test sonuçları çizelge ve grafik halinde bu bölümde sunulmuş ve test sonuçlar yorumlanmıştır.

Bu testlerin sıfır hipotezini gösteren h ve P - değerleri incelenir. $h = 0$ ise sıfır hipotezinin kabul edildiğini gösterir. Yapılan testler anlamlılık düzeyi α olarak %1 seçilmiştir. Bu nedenle P -değeri ≥ 0.01 ise üretilen dizi raststaldır.

Seçilen üç üretece belirtilen testler uygulanmıştır. Algoritma girdisi olarak üç üreticinin parametreleri verilmiştir [Hellekalek,1994].

1. Üreteç $[a \ c \ q \ x_0] = [9102 \quad 36884165 \quad 2^{31} - 1 \quad 5]$ [Lendl, 2001]
2. Üreteç $[a \ c \ q \ x_0] = [21916 \quad 71499791 \quad 2^{31} - 1 \quad 127]$ [Lendl, 2001]
3. Üreteç $[a \ c \ q \ x_0] = [95985493 \quad 1804787115 \quad 2^{31} - 1 \quad 195]$ [Hellekalek, 1994]

Algoritma çıktısı olarak sözde rastsal sayı dizisi alınmıştır.

Test girdileri olarak;

- Algoritma çıktısı
- Üretilen dizi uzunluğu : 1000000
- Örneklemdeki dizi uzunluğu : 5000
- Örneklem sayısı : 200
- $\alpha = 0.01$ değerleri seçilmiştir.

Tüm algoritmalarda aynı çıktı elde etme yöntemi ile çıktılar alınmış ve bu çıktılar testlerden geçirilmiştir. ICG algoritmasına ait üç üreteç için test sonuçları aşağıdaki çizelge ve grafikler ile verilmiştir.

ICG algoritmasına ait üreteçlerin maksimum periyoda sahip diziler üretilip üretmedikleri kontrol edilmiş; maksimum periyoda sahip diziler ise $\text{max_flag}=1$ değerine, değil ise $\text{max_flag}=0$ değerine eşitlenmiştir.

4.3.2.1 Strateji-1 Testleri

Seçilen üreteç ile üretilen 1000000 dizi uzunluğundaki veriye Frekans, Koşum, Spektral ve Ki-Kare testleri yapılmıştır. Test sonuçları Çizelge 4.20 ve Çizelge 4.21’de yer almaktadır.

Çizelge 4.20 ICG Algoritması Strateji-1 Test Sonuçları-1

Üreteç	a	c	M	X_0	Periyot	Max Flag	h_ki-kare	p_ki-kare
1	9102	36884165	2147483647	5	2147483647	1	0	0,4695396
2	21916	71499791	2147483647	127	2147483647	1	0	0,8491923
3	95985493	1804787115	2147483647	195	2147483647	1	1	< 0,01

Seçilen üreteçlerin periyodu aynıdır ve Max_flag parametresi ile üreteçlerin tümünün maksimum periyoda sahip diziler ürettiği gözlenmektedir.

Tüm testlerden elde edilen h ve p değerleri incelenmiştir. Anlam düzeyi (α) %1 seçildiği durumda Çizelge 4.20 ve Çizelge 4.21’de görüldüğü gibi, 1. ve 2. üreteç ile üretilen verinin tümüne uygulanan; Frekans, Koşum, Spektral ve Ki-Kare testlerinin sonucunda elde edilen h değerleri sıfırdır. Testlerin sonucunda h=0 olması sıfır hipotezinin kabul edildiğini göstermektedir. Bu duruma göre ICG algoritması ile oluşturulan ilk iki üretelin tüm testlerden geçtiği gözlenmiştir. 3. üretelin ise Frekans, Koşum, Spektral ve Ki-Kare testlerinden kaldığı gözlenmiştir.

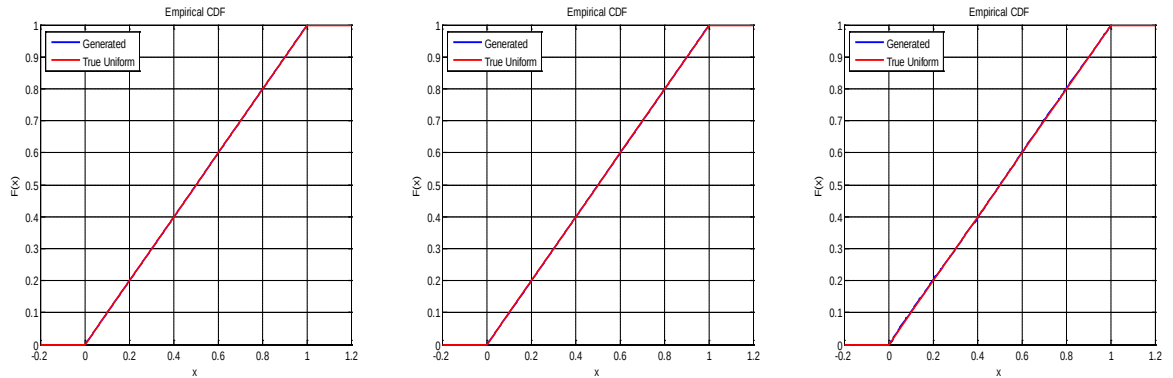
Çizelge 4.20 ve Çizelge 4.21’de yer alan test sonuçlarına göre 1. ve 2. üretelin ürettiği dizilerin rastsal olduğu değerlendirilmektedir. CDF ve histogram grafikleri incelenerek nihai bir sonuca varılabilecektir.

Çizelge 4.21 ICG Algoritması Strateji-1 Test Sonuçları-2

Üreteç	h_fre	p_fre	h_koşum	p_koşum	h_spek	p_spek	ortalama	varyans
1	0	0,6701078	0	0,6218841	0	0,71357	0,500052	0,083326
2	0	0,3182794	0	0,771819	0	0,028959	0,50028	0,083335
3	1	< 0,01	1	< 0,01	1	< 0,01	0,498459	0,083538

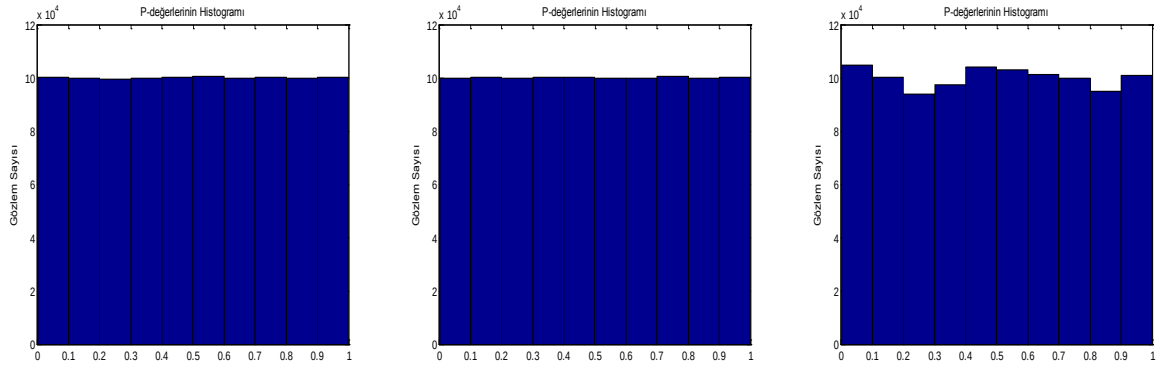
Birörnek dağılıma sahip bir rastsal dizinin ortalamasının 0.5, varyansının da 0.08 olması gerektiği bilinmektedir [Gentle, 2003]. Üç üretcin ürettiği dizilerin ortalama ve varyans değerlerine bakarak birörnek dağılıma sahip olup olmadıkları incelenmiştir. Çizelge 4.21’de görüldüğü gibi üretçilere ait ortalama ve varyans değerlerinin 0.5 ve 0.08 civarında olduğu ve çok küçük farklılıklar sergiledikleri gözlenmektedir.

CDF grafiği ile her üretçte üretilen dizinin (1000000 veri) birörnek dağılımı ile ne kadar örtüştüğünü gözlemek amaçlanır. Şekil 4.30’da üç üretç için sırasıyla yer alan CDF grafiklerinde görüldüğü gibi; 1. ve 2. üretcin birörnek dağılıma ait veri ile tam olarak çakıştığı, 3. üretcin diğer üretçler kadar olmasa da neredeyse çakıştığı gözlenmektedir. Birörnek dağılıma ait beklenen ve üretçlere ait gözlenen veriler örtüştüğü için, üç üretcin birörnek dağılıma sahip diziler oluşturduğu değerlendirilmektedir.



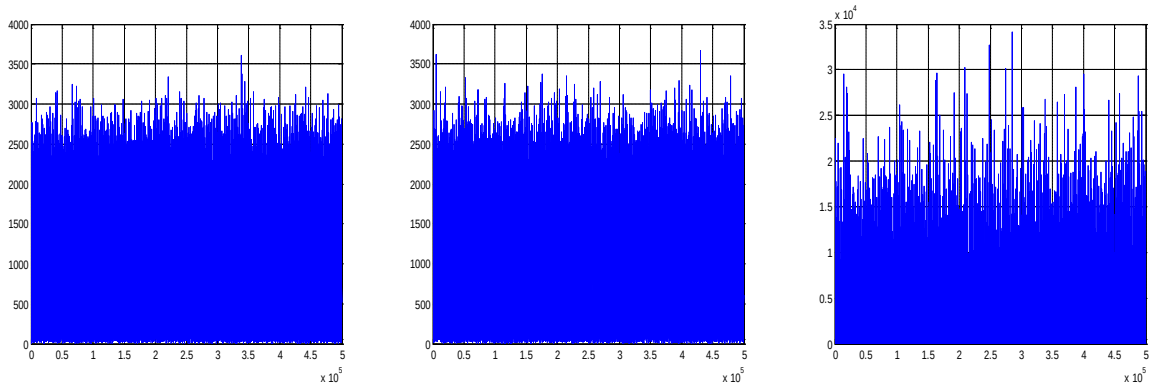
Şekil 4.30 ICG Algoritması Strateji-1 CDF Grafikleri

Her bir üretç için üretilen tüm verinin grafiklere yansıyan dalgalanmalarını gözlemleyip kıyaslayabilmek adına CDF ve histogram grafiklerinin birlikte gösterilmiştir. 3. üretcin ilk iki üretç kadar birörnek dağılıma yakın olmadığı ve birörnek dağılımdan farklı olduğu, Şekil 4.31’de sırası ile yer alan üretç-1, üretç-2 ve üretç-3’e ait histogramlar ile anlaşılmaktadır. Çizelge 4.21’de yer alan 1. ve 2. üretcin, birörnek dağılıma sahip rastsal diziler ürettikleri CDF ve histogram ile de gözlenmektedir. 3. üretcin ise birörnek dağılıma sahip rastsal diziler üretmediği değerlendirilmektedir.



Şekil 4.31 ICG Algoritması Strateji-1 Histogramı

Üç üretici ile üretilen tüm verinin Spektral testleri yapılmıştır ve bu test sonuçları Çizelge 4.21’de verilmiştir. Çizelgede yer alan sonuçlara göre 1. ve 2. üreticinin Spektral testten geçtiği, 3. üreticinin ise kaldığı bilinmektedir. Şekil 4.32’de üç üretici için sırası ile yer alan FFT grafikleri, 1. ve 2. üreticilerin Spektral testten geçtiği, 3. üreticinin ise Spektral testten kaldığını desteklemektedir.



Şekil 4.32 ICG Algoritması Strateji-1 FFT Grafikleri

4.3.2.2 Strateji-2 Testleri

Örneklem sayısı ve bir örneklemdaki dizi uzunluğu seçimi ile algoritma tarafından üretilen dizinin tümü kullanılarak testler gerçekleştirilmiştir. Bu kısımda üretilen tüm veri, örneklem sayısı ve bir örneklemdaki dizi uzunluğu göz önüne alınarak bölünmüştür. Bölünen veriler üzerinde Frekans, Koşum, Spektral ve Ki-Kare testleri yapılmıştır.

Üç üreticinin 200 örneklemine yapılan testler sonucunda elde edilen P-değerinin analiz yapılır. P-değeri analizi kapsamında Dizi Geçme Oranı (DGO) hesaplanır.

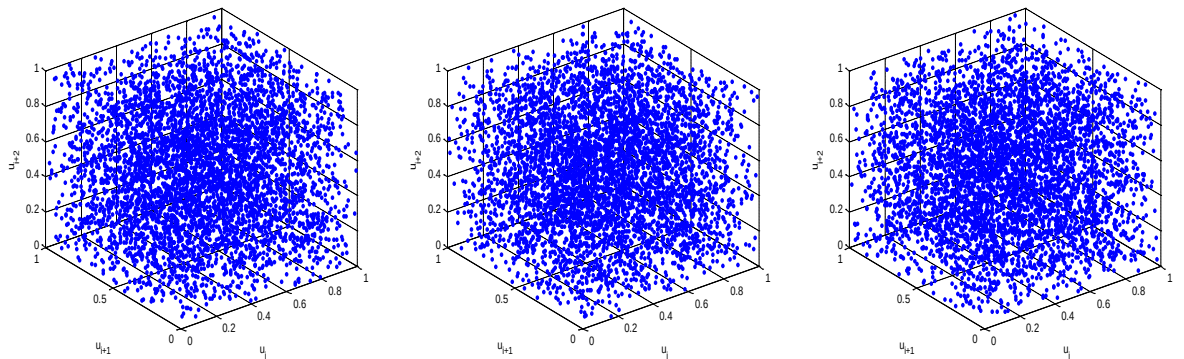
200 örneklem arasında kaçının sıfır hipotezini geçtiğine dair bir oran elde edilir ve sonuç olarak testten geçip geçmediğine bakılır. ICG algoritmasına ait üreticilerin test sonuçları Çizelge 4.22’de yer almaktadır.

Çizelge 4.22 ICG Algoritması Strateji-2 Test Sonuçları-1

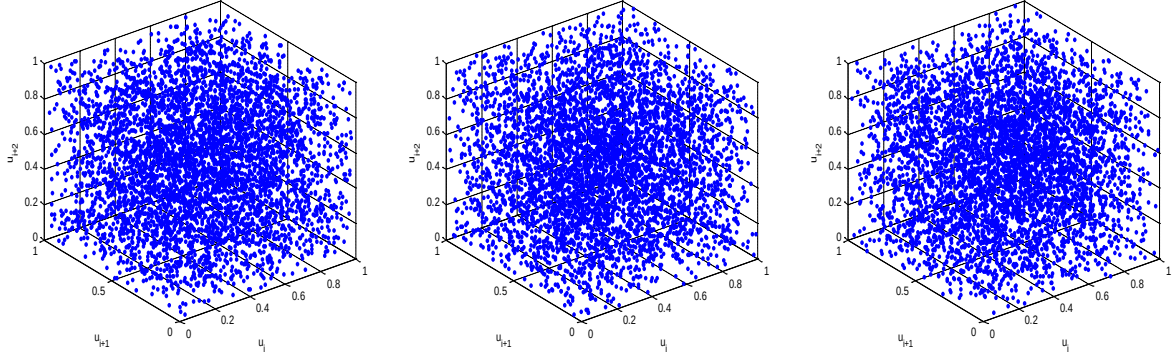
Üreteç	Ki- Kare DGO	Ki-Kare Test Sonucu	Frekans DGO	Frekans Testi Sonucu	Koşum DGO	Koşum Testi Sonucu	Spektral DGO	Spektral Test Sonucu
1	0,985	1	0,985	1	0,99	1	0,995	1
2	0,975	1	1	1	0,995	1	0,985	1
3	1	1	1	1	0,33	0	0,955	0

Çizelge 4.22’de görüldüğü gibi farklı oranlarla da olsa test sonuçları 1 olduğundan, 1. ve 2. üreticinin 200 örneklemine uygulanan Frekans, Koşum, Ki-Kare ve Spektral testlerin tümünden geçtiği, 3. üreticinin ise Koşum ve Spektral testlerden geçemediği gözlenmiştir.

Üç üreticiden rastsal örneklemeler alınarak dizi elemanları arasında bir ilinti olup olmadığı, dizi elemanlarının periyodik özellikler sergileyip sergilemediğini incelemiştir. Dizi elemanları arasındaki ilinti, arka arkaya olan üç duruma ait veri (u_i, u_{i+1}, u_{i+2}) incelenerek tespit edilmeye çalışılmıştır. Rastsal olarak 48. , 115. ve 193. örneklemeleri seçilerek olan 3-çokuzlu grafikler üzerinden gözlem yapılmıştır. Dizi elemanları arasındaki ilintinin gözlemlenebilmesi için Şekil 4.33 ile Şekil 4.35 arasında yer alan 3-boyutlu grafikler incelenmiştir.

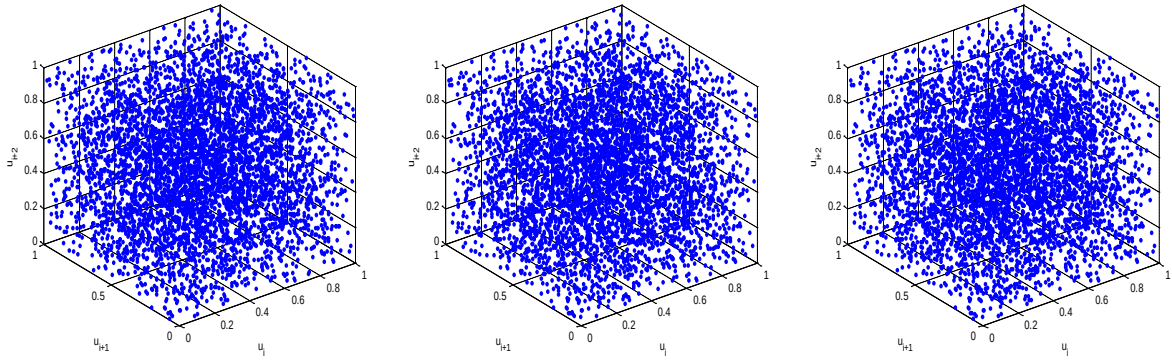


Şekil 4.33 ICG Algoritması Strateji-2, Üreteç-1 48. , 115. ve 193. Örneklemeleri



Şekil 4.34 ICG Algoritması Strateji-2, Üreteç-2 48. , 115. ve 193. Örneklemeleri

Şekil 4.33, Şekil 4.34 ve Şekil 4.35'te görüldüğü gibi üç üreticinin rastsal seçilen hiçbir örnekleminde, dizi elemanları arasında bir ilinti ya da periyodik özellik gözlenmemiştir. Bu sonuç ICG algoritmasında kafes yapısının oluşmadığını göstermektedir.



Şekil 4.35 ICG Algoritması Strateji-2, Üreteç-3 48. , 115. ve 193. Örneklemeleri

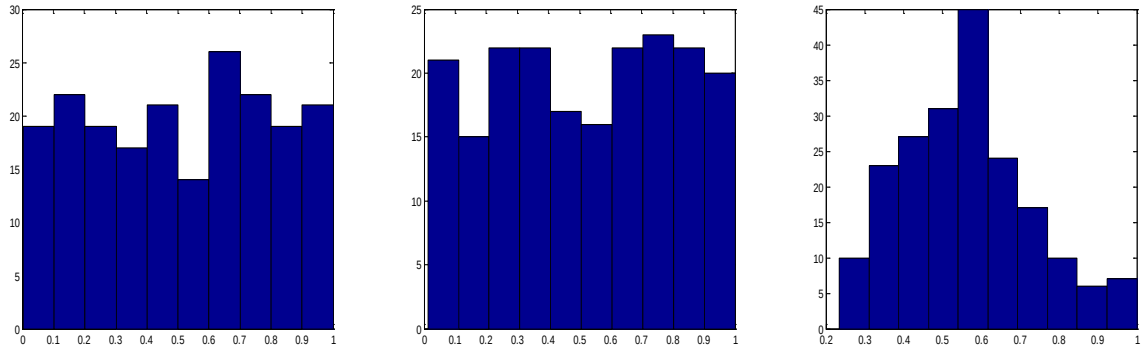
Tüm testler için P-değeri analizi kapsamında P-değerinin dağılımı hesaplanır. Her bir üreticinin 200 örneklemine uygulanan Frekans, Spektral, Koşum ve Ki-Kare testleri sonucunda elde edilen P-değerlerine Ki-Kare testi uygulanarak P-değerlerinin dağılımına bakılır.

Frekans testinden elde edilen P-değerleri üzerine yapılan Ki-Kare testi sonuçları Çizelge 4.23'te yer almaktadır.

Çizelge 4.23 ICG Algoritması Strateji-2 Frekans Testi

	Normal Çıktı Yöntemi	
Üreteç	PD Frekans Testi H değeri	PD Frekans Testi P-değeri
1	0	0,8525162
2	0	0,9207574
3	1	< 0,01

Çizelge 4.23'te belirtilen Ki-Kare testi sonuçları ve P-değerlerinin histogramları birlikte değerlendirilmektedir. Şekil 4.36'da sırası ile yer alan üreteç-1, üreteç-2 ve üreteç-3'e ait histogramlarda görünen iniş-çıkışların; 1. ve 2. üreticinin P-değerlerinin birörnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturmadığı anlaşılmaktadır. 3. üreticinin ise P-değerlerinin birörnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturduğu değerlendirilmektedir.



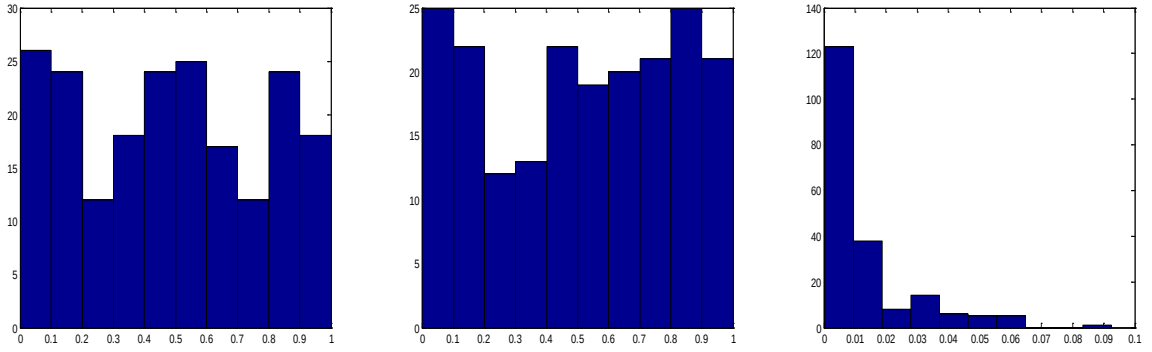
Şekil 4.36 ICG Algoritması Strateji-2 Frekans Testi Histogramı

Üreteçlerin örneklemelerine yapılan Koşum testinden elde edilen P-değerlerine, Ki-Kare testi uygulanarak elde edilen test sonuçları Çizelge 4.24'te verilmiştir.

Çizelge 4.24 ICG Algoritması Strateji-2 Koşum Testi

	Normal Çıktı Yöntemi	
Üreteç	PD Koşum Testi H değeri	PD Koşum Testi P-değeri
1	0	0,184029
2	0	0,48212
3	1	< 0,01

Çizelge 4.24'te yer alan üreticilerin P-değerleri üzerinde yapılan Ki-Kare testi sonuçları ve P-değerlerinin histogramları birlikte değerlendirilmektedir. Şekil 4.37'de sırası ile yer alan üretic-1, üretic-2 ve üretic-3'e ait histogramlarda görünen iniş-çıkışların, 1. ve 2. üreticinin P-değerlerinin birörnek dağılıma sahip olmadığına dair göstergeler oluşturmadığı anlaşılmaktadır. 3. üretic için ise histogramda görünen iniş-çıkışların, P-değerlerinin birörnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturduğu anlaşılmaktadır.



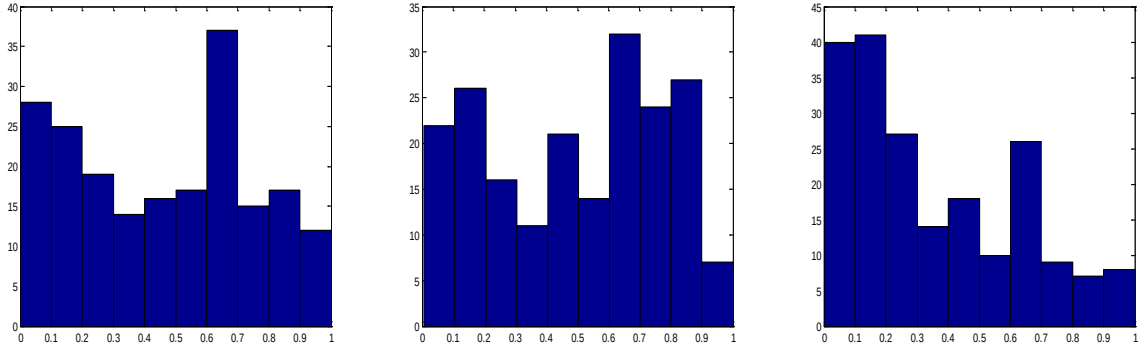
Şekil 4.37 ICG Algoritması Strateji-2 Koşum Testi Histogramı

Üç üreticinin 200 örneklemine yapılan Spektral testten elde edilen P-değerlerine, Ki-Kare testi uygulanmış; elde edilen test sonuçları Çizelge 4.25'te verilmiştir. Bu çizelgedeki sonuçlara göre üç üreticinin de Spektral testten kaldığı görülmektedir.

Çizelge 4.25 ICG Algoritması Strateji-2 Spektral Testi

	Normal Çıktı Yöntemi	
Üreteç	PD Spektral Testi H değeri	PD Spektral Testi P-değeri
1	1	< 0,01
2	1	< 0,01
3	1	< 0,01

Şekil 4.38'de üç üretic için sırasıyla yer alan P-değerlerinin histogramları, Çizelge 4.25'te yer alan Ki-Kare testi sonuçlarını desteklemektedir. P-değerlerinin histogramları ve çizelgedeki test sonuçları birlikte değerlendirildiğinde, üç üreticinin de Spektral testten kaldığı anlaşılmaktadır. Bu durumda, üç üretic için P-değerlerinin birörnek dağılıma sahip olmadığı anlaşılmıştır.



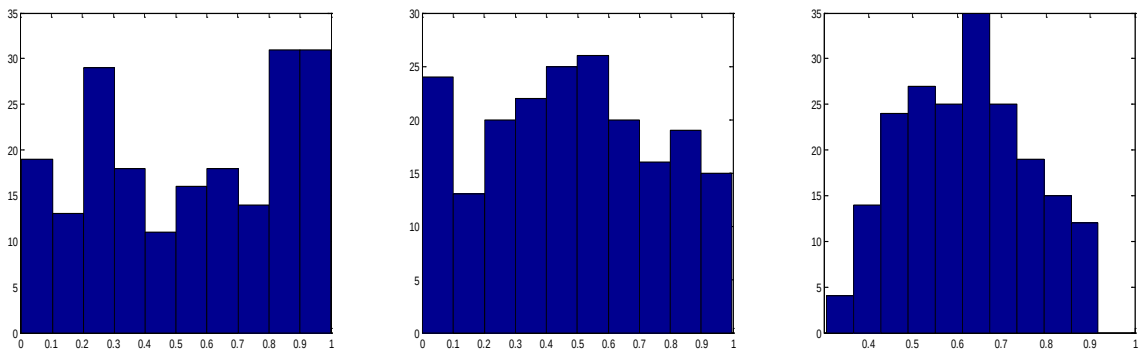
Şekil 4.38 ICG Algoritması Strateji-2 Spektral Testi Histogramı

ICG algoritmasına ait üreteçlerin P-değerleri dağılımına bakıldığında, 1. ve 3. üreticinin örneklemelerinin Ki-Kare testinden kaldığı, 2. üreticinin ise Ki-Kare testinden geçtiği Çizelge 4.26'da görülmektedir.

Çizelge 4.26 ICG Algoritması Strateji-2 Ki-Kare Testi

Üreteç	Normal Çıktı Yöntemi	
	PD Ki-Kare Testi H değeri	PD Ki-Kare Testi P-değeri
1	1	< 0,01
2	0	0,4634716
3	1	< 0,01

Çizelge 4.26'da yer alan Ki-Kare testi sonuçlarının ve Şekil 4.39'da yer alan histogramda görünen iniş-çıkışların, 1. ve 3. üreteçlerin P-değerlerinin birörnek dağılıma sahip olmadığını dair göstergeler oluşturduğu değerlendirilmektedir. 2. üreteç için ise histogramda görünen iniş-çıkışların, P-değerlerinin birörnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturmadığı anlaşılmaktadır.



Şekil 4.39 ICG Algoritması Strateji-2 Ki-Kare Testi Histogramı

Strateji-2'ye uygulanan Frekans, Koşum, Spektral ve Ki-Kare test sonuçları bir bütün halinde Çizelge 4.27'de verilmiştir.

Çizelge 4.27 ICG Algoritması Strateji-2 Test Sonuçları

Üreteç	P_Değeri HT- ki-kare	P_Değeri p_değeri- ki-kare	P_Değeri HT-frek	P_Değeri p_değeri- frek	P_Değeri HT- koşum	P_Değeri p_değeri- koşum	P_Değeri HT-spek	P_Değeri p_değeri- spek
1	1	0,01	0	0,8525162	0	0,184029	1	< 0,01
2	0	0,4634716	0	0,9207574	0	0,48212	1	< 0,01
3	1	< 0,01	1	< 0,01	1	0	1	< 0,01

Çizelge 4.27'de görüldüğü gibi sıfır hipotezine göre 1. üreticin 200 örnekleme yapılan testlerin P-değerlerine uygulanan Ki-Kare testi sonucunda Spektral ve Ki-Kare testi dışındakilerden geçilmiştir. 2. Üreticin 200 örnekleme yapılan testlerin P-değerlerine bakıldığında Spektral testi dışındakilere uygulanan Ki-Kare testinden başarı ile geçilmiştir. 3. üreticin 200 örnekleme uygulanan tüm testlere ait P-değerlerin Ki-Kare testinden kalmıştır. 3. üreticin örneklemelerin birörnek dağılıma sahip olmadığı gözlemlenmektedir.

1. üreticin 200 örnekleme uygulanan Ki-Kare testinden kalmış olması, örneklemelerin rastsal dizilerden oluşmadığını gösterir; ancak tüm verinin rastsal olmadığını söylenmesi mümkün değildir. Çünkü Çizelge 4.20'de görüldüğü gibi 1. üreticin tüm verilerine yapılan Ki-Kare testinden geçmiştir. Bu nedenle 1. üretece ait tüm verinin birörnek dağılıma sahip rastsal bir dizi olmadığına dair belirgin göstergeler oluşturmadığı değerlendirilmektedir.

2. üreticin örneklemelerinin ve tüm verilerinin, diğer üreteçlere göre en iyi istatistiksel özelliklere sahip diziyi ürettiği anlaşılmaktadır. 2. üreticin birörnek dağılıma sahip rastsal dizi ürettiği kararına varılmıştır.

3. üreticin 200 örnekleme uygulanan testlerden kaldığı, örneklemelerin birörnek dağılıma sahip rastsal dizilerden oluşmadığı gözlenmiştir. Buna bağlı olarak beklendiği gibi, 3. üretece ait, Strateji-1 olarak ele alınan, tüm verinin de testlerden kaldığı ve birörnek dağılıma sahip rastsal diziler olmadığı yapılan incelemeler sonucu anlaşılmıştır.

4.4 Mersenne Twister (MT)

Mersenne Twister (MT) algoritması Makoto Matsumoto ve Takuji Nishimura tarafından 1997 yılında tasarlanmış, hızlı ve yüksek kalitede sözde rastsal sayı üreten güçlü bir algoritmadır. Algoritmada yer alan mantıksal kapılar ve kaydırma işlemleri nedeni ile MT algoritması “Twisted LFSR (TGFSR)” algoritmasının türevi olarak da tanımlanmaktadır [Matsumoto, 1992].

Algoritmanın adı, periyot boyunu etkileyen kritik parametre olan Mersenne asalından gelmektedir; Mersenne asalı aşağıda belirtildiği gibi tanımlanmaktadır.

$$M_p = 2^p - 1 \rightarrow p \text{ ve } M \text{ değerlerinin asal olması gerekmektedir.} \quad (4.30)$$

MT algoritması ikili alan $\mathbb{F}_2$ üzerine “Matrisli Doğrusal Tekrarlama” tabanına dayanmaktadır [Matsumoto, 1998]. Diğer bir deyişle MT algoritması ikili tabanda sabit bir vektörün doğrusal dönüşümler ile bir döngü içerisinde tekrarlanmasıdır. Algoritmanın matematiksel ifadesi aşağıda verilmiştir. n tekrar derecesini, ω ise vektör uzunluğunu göstermektedir.

$$x_{k+n} := x_{k+m} \oplus \left(x_k^u \mid x_{k+1}^l \right) A, \quad (k = 0, 1, \dots) \quad (4.31)$$

$$\text{periyot} = 2^{n\omega-r} - 1 = 2^p - 1 = 2^{624 \cdot 32 - 31} - 1 = 2^{19937} - 1 \quad (4.32)$$

Başlangıç değerinin seçimi MT algoritmasında önem taşımamaktadır. Özel bir teknik kullanılmadığında başlangıç işlemleri hızlı ilerler [Matsumoto, 1992]. Bu nedenle başlangıç değeri dışındaki parametrelerin uygun seçilmesi ile maksimum periyot $2^{n\omega-r} - 1$ olarak tanımlanmaktadır [Matsumoto, 1998]. Bu algoritma; sadece 624 sözcük gibi küçük bir çalışma alanı kullanarak, $2^{19937} - 1$ değerinde asal ve çok uzun periyoda sahip diziler üretmeye imkan sağlamaktadır.

MT algoritması, uzun periyot ve büyük k -dağılım özelliğinden dolayı diğer algoritmalarla göre ön plana çıkmaktadır. Sözde rastsal sayı dizilerinin rastsallıklarının ölçüldüğü testler, Spektral testlerde olduğu gibi büyük boyutlu birörnek dağılım baz alınarak yapılır. Bu kapsamda MT Algoritmasında $2^{19937} - 1$ uzunluğunda periyoda sahip bir dizi, 623 boyutlu eşdağılım özelliğine sahipken 32

bitlik doğruluktur. Bu özellikteki bir algoritma 10.18 saniye CPU zamanı kullandığından, pek çok sözde rastsal sayı üretici algoritmadan daha hızlı çalışmaktadır [Matsumoto, 1998]. MT algoritması Eş. 4.31'de yer alan denklemde görüldüğü gibi yoğun matematik denklemleri ve teoremleri içermediğinden hızlı bir algoritmadır.

MT algoritması istatistiksel testlerin birçoğundan geçtiği gibi, MT algoritması ile üretilen sözde rastsal sayıların elemanları arasında ilinti de gözlenmemektedir [L'Ecuyer, 2007].

MT algoritması, doğrusal çıktı ürettiği için Blum Blum Shub algoritması gibi kriptografik anlamda güvenli bir algoritma değildir. Ancak MT algoritmasının çıktısı güvenli bir özetleme algoritması ile yeni bir hale çevrilirse, MT algoritması kriptografik alanlarda da kullanılabilir. Uzun bir iterasyon periyoduna sahip olsa da bir iterasyonun tahmin edilmesi ile sonraki tüm iterasyonlar tahmin edilebilir. Tahmin edilemezlik özelliğine sahip olmadığından CSPRNG (Kriptografik olarak Güvenli Sözde Rastsal Sayı Üreteci) algoritmalar sınıfında yer almamaktadır. Bu nedenle algoritma şifreleme uygulamaları yerine Monte Carlo benzetim uygulamalarında tercih edilmektedir [Ferrenberg, 1992]. C derleyicisinde [Matsumoto, 1998] ve MATLAB programlama dilinin 7.4. versiyonunda sözde rastsal sayı üreten standart algoritma olarak MT19937 tipi MT algoritması kullanılmaktadır [Challis, 2008].

Gecikmeli Fibonacci ve LCG algoritmaları ile başlangıç durumuna hızlı dönülebildiğinden, MT algoritmalarında başlangıç değerleri Fibonacci ve LCG algoritmaları ile atanabilir.

$$x_{k+n} := x_{k+m} \oplus (x_k^u | x_{k+1}^l) A, \quad (k = 0, 1, \dots) \quad (4.31)$$

Algoritmada kullanılan parametreler;

- $m, \quad 1 \leq m \leq n$
- $r, \quad 0 < r < \omega - 1$ aralığında bir tamsayı olup x_k^u içinde gizli bir parametre
- n , tekrar derecesi

- 32 bit doğruluk k-dağılımında, $1 \leq k \leq 623$
- x_k^u , x_k vektörünün en önemli $\omega - r$ bitlik kısmı
- x_{k+1}^l , x_{k+1} vektörünün en önemsiz r bitlik kısmı
- A , $\omega \times \omega$ boyutunda sabit bir matris
- x ve a , ω boyutunda satır vektörleri olup ikili sonlu $\mathbb{F}_2 = \{0, 1\}$ alanda tanımlanmaktadır.
- l , s , t ve u kaydırmayı sağlayan tamsayı parametreler
- b ve c uygun sözcüğün uygun boyutta maskelenmesini sağlayan vektör parametreler
- T , $\omega \times \omega$ boyutunda tersi alınabilir bir geçici matris
- $(x \gg u)$: u bit sağa kaydırma yazmacı
- $(x \ll u)$: u bit sola kaydırma yazmacı olarak tanımlanmaktadır [Matsumoto,1998].

MT algoritmasında kullanılan parametreler; periyodun belirlenmesinde sağlayan parametreler (ω, n, a) ve k-dağılımında v bit doğruluk için ayarlanan geçici parametreler (l, s, t, u, b, c) olarak sınıflara ayrılır. MT algoritmasında olması gereken tasarım kriterleri ve algoritma adımları aşağıda belirtilmiştir.

1) Başlangıç değeri verilir; $k=0$ durumu için x_n yaratılır.

MT algoritmasına verilen başlangıç değerini kullanarak başlangıç durumu yaratılır. Algoritma tarafından Eş. 4.31 kullanılarak $k=0$ durumu için x_n 'i yaratır.

$k = 1, 2, \dots$, değerleri verilerek $x_n, x_{n+1}, x_{n+2} \dots$ bulunur.

2) $x = (x_{\omega-1}, x_{\omega-2}, \dots, x_0)$ şeklinde tanımlı olduğu durumda,

$x_{k\omega-1}^u = (x_{\omega-2}, x_{\omega-3}, \dots, x_0)$ ve $x_{k+1}^l = (x_{r-1}, \dots, x_0)$ vektörlerinin oluşturulması

$$u \leftarrow \underbrace{1\dots1}_{\omega-r} \underbrace{0\dots0}_r \text{ (}\omega - r \text{ bit için bitmask uygulanır)} \rightarrow x_k^u = (x_{\omega-1}, x_{\omega-2}, \dots, x_r) \quad (4.33)$$

$$l \leftarrow \underbrace{0\dots0}_{\omega-r} \underbrace{1\dots1}_r \text{ (} r \text{ bit için bitmask uygulanır)} \rightarrow x_{k+1}^l = (x_{r-1}, \dots, x_0) \quad (4.34)$$

$$a \leftarrow a_{\omega-1} a_{\omega-2} \dots a_1 a_0 \text{ (} A \text{ matrisinin son satırı)} \quad (4.35)$$

3) $(x_k^u | x_{k+1}^l)$ vektörünün oluşturulması

$$(x_k^u | x_{k+1}^l) = \underbrace{(x[k] \text{ AND } u)}_{x_k^u} \text{ OR } \underbrace{(x[(k+1) \bmod n] \text{ AND } II)}_{x_{k+1}^l} \quad (4.36)$$

4) $(x_k^u | x_{k+1}^l)$ A vektörünün oluşturulması [Matsumoto, 1992]

$$x[k] \leftarrow x[(k+m) \bmod n] \text{ XOR } ((x_k^u | x_{k+1}^l) \gg 1) \quad (4.37)$$

$$\text{XOR} \begin{cases} 0, & (x_k^u | x_{k+1}^l)' \text{in LSB} = 0 \\ a, & (x_k^u | x_{k+1}^l)' \text{in LSB} = 1 \end{cases} \quad (4.38)$$

A matrisi, x vektörü ile çarpılması işleminin kolay ve hızlı olması dikkate alınarak seçilir. Bu nedenle aşağıdaki gibi bir örnek seçilebilir.

$$A = \begin{pmatrix} 1 & \dots & & \\ & & \ddots & \\ & \vdots & 1 & \\ & & & \ddots & 1 \\ a_{\omega-1} & a_{\omega-2} & \dots & & a_0 \end{pmatrix} \rightarrow \omega \times \omega \text{ boyutunda sabit bir matris}$$

5) $x[k]T$ oluşturulması (4.39)

k dağılımında v bit doğruluğunu sağlamak için $\omega \times \omega$ boyutunda tersi alınabilir bir geçici T matris, Eş. 4.40'da belirtildiği gibi ayarlanır. Aşağıdaki ardışık dönüşümler uygulanır [Matsumoto,1998].

$$x \rightarrow z = xT \quad (4.40)$$

$$y := x \oplus (x \gg u) \quad (4.41)$$

$$y := y \oplus ((y \ll s) \text{ AND } b) \quad (4.42)$$

$$y := y \oplus ((y \ll t) \text{ AND } c) \quad (4.43)$$

$$z := y \oplus (y \gg l) \quad (4.44)$$

6) $k \leftarrow (k + 1) \bmod n$

7) 3. adıma gidilir ve Eş. 4.36'da yer alan işlemde devam edilir.

4.4.1 MT Algoritmasının Uygulanması

MT sözde rastsal sayı üretme algoritması hakkında yukarıdaki bölümde detaylı olarak bilgi verilmiştir. MT algoritmasının kodlanması, programlama dili olan MATLAB ile gerçekleştirilmiştir.

Maksimum periyoda ulaşabilmek için uygun parametreler seçilmiştir. Maksimum periyot $= 2^{n\omega-r} - 1 = 2^{624 \cdot 32 - 31} - 1 = 2^{19937} - 1$ dir. Başlangıç değeri bu algoritma için kritik bir parametre olmadığından rastsal seçilmiştir. Seçilen parametreler [Matsumoto,1998] ;

- $(\omega, n, m, r) = (32, 624, 397, 31)$
- $u = 11, l = 18, s = 7, t = 15$
- $a = 9908B0DF = 2567483615$
- $b = 9D2C5680 = 2636928640$
- $c = EFC60000 = 4022730752$
- $mt_p1 = 6C078965 = 1812433253$
- $x_0 = 5489$
- Çıktı boyutu = 1000000

MT algoritması kodlanma sırasında yapılan kontroller ve algoritma adımları aşağıda belirtilmiştir.

- 1) Başlangıç değeri x_0 atanır.
- 2) Başlangıç değeri kullanılarak algoritmanın başlangıç durumu oluşturulur. Bir durum 624 boyutlu diziden oluşmaktadır. Elde edilen başlangıç durumu kaydedilir.
- 3) Algoritmanın çıktı boyutu atanır.

- 4) 624 boyutlu diziden oluşan başlangıç durumu yeni durumun oluşturulmasında kullanılır.
- 5) Bir durumda bulunan 624 boyutlu dizinin her bir elemanı kullanılarak çıktı elde edilir. 624 kez aynı döngü tekrarlanarak bir durum için çıktı üretilir.
- 6) Yeni bir durumun üretilmesi için 4. adımdaki işlem mevcut durumu yansıtan dizi ile tekrarlanır; böylelikle yeni bir durum üretilir.
- 7) Bu işlemler girilen çıktı boyutu elde edilene kadar tekrarlanır. Bir durum için çıktı üretilirken 5. adım 624 kez tekrarlanır. Yeni bir durum üretmek için 4. adım 1 kez tekrarlanır. Bu işlemler tamamlanınca ana döngüde 1 döngü tamamlanmış olur. Çıktı boyutu ne girildi ise ana döngü o sayı kadar dönecektir.
- 8) Testler iki stratejiye ayrılarak yapılır. Strateji-1 de üreteç tarafından üretilen tüm veriye, 1000000 uzunluğundaki veri, testler yapılırken; Strateji-2 de ise bölünmüş veriye yani 5000 örneklem dizi uzunluğuna sahip 200 örnekleme test yapılır.

4.4.2 Uygulanan Testlerin Sonuçları ve Değerlendirmeler

Tüm üreteçlere

- Ki-Kare Testi,
- Frekans Testi,
- Koşum Testi,
- Ayırık Fourier Dönüşümü (Spektral) Testi

yapılarak elde edilen test sonuçları çizelge ve grafik halinde bu bölümde sunulmuş ve test sonuçlar yorumlanmıştır.

Bu testlerin sıfır hipotezini gösteren h ve P - değerleri incelenir. $h = 0$ ise sıfır hipotezinin kabul edildiğini gösterir. Yapılan testler anlamlılık düzeyi α olarak %1 seçilmiştir. Bu nedenle P -değeri ≥ 0.01 ise üretilen dizi rastsaldır.

Algoritma girdisi olarak, 4.4.1 MT Algoritmasının Uygulanması bölümünde üretecin parametreleri verilmiştir.

Algoritma çıktısı olarak sözde rastsal sayı dizisi alınmıştır.

Test girdileri olarak;

- Algoritma çıktısı
- Üretilen dizi uzunluğu : 1000000
- Örneklemdeki dizi uzunluğu : 5000
- Örneklem sayısı : 200
- $\alpha = 0.01$ değerleri seçilmiştir.

Tüm algoritmalarda aynı çıktı elde etme yöntemi ile çıktılar alınmış ve bu çıktılar testlerden geçirilmiştir. MT algoritması üretici için test sonuçları aşağıdaki çizelge ve grafikler ile verilmiştir.

4.4.2.1 Strateji-1 Testleri

Seçilen üreteç ile üretilen 1000000 dizi uzunluğundaki veriye Frekans, Koşum, Spektral ve Ki-Kare testleri yapılmıştır. Test sonuçları Çizelge 4.28'de yer almaktadır.

Çizelge 4.28 MT Algoritması Strateji-1 Test Sonuçları-1

h_ki-kare	p_ki-kare	h_frek	p_frek	h_koşum	p_koşum	h_spek	p_spek	ortalama	varyans
0	0,94299	0	0,86524	0	0,723957	0	0,53866	0,500123	0,083379

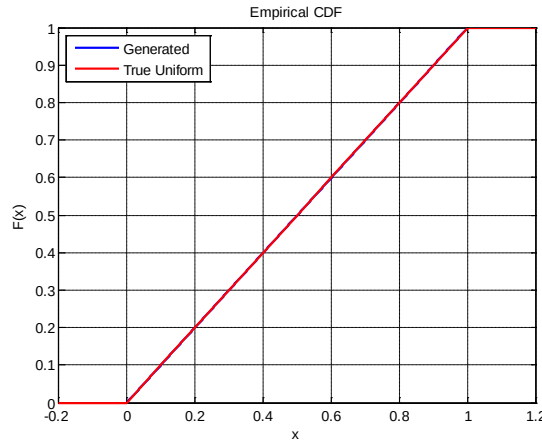
MT19937 algoritması için $2^{19937} - 1$ periyodunda diziler üretecek bir üreteç seçilmiştir, parametrelerin istenen periyodu sağladığı bilinmektedir [Matsumoto,1998].

Tüm testlerden elde edilen h ve p değerleri incelenmiştir. Anlam düzeyi (α) %1 seçildiği durumda Çizelge 4.28'de görüldüğü gibi seçilen üreteç için tüm veriye uygulanan Frekans, Koşum, Spektral ve Ki-Kare testlerinin h değerleri sıfırdır. Testlerin sonucunda h=0 olması, sıfır hipotezinin kabul edildiği ve buna göre MT algoritması ile oluşturulan üreticinin tüm testlerden geçtiği gözlenmiştir.

Çizelge 4.28’de yer alan test sonuçlarına göre MT algoritmasına ait üreticinin ürettiği dizinin rastsal olduğu değerlendirilmektedir. CDF ve histogram grafikleri incelenerek nihai bir sonuca varılacaktır.

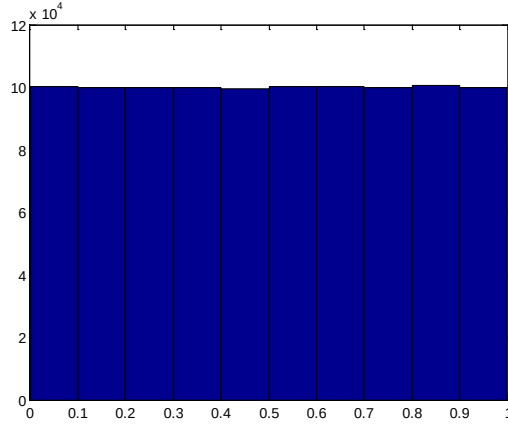
Birörnek dağılıma sahip bir rastsal dizinin ortalamasının 0.5, varyansının da 0.08 olması gerektiği bilinmektedir [Gentle, 2003]. MT algoritmasına ait üreticinin ürettiği dizilerin ortalama ve varyans değerlerine bakarak birörnek dağılıma sahip olup olmadıkları incelenmiştir. Çizelge 4.28’de görüldüğü gibi üretece ait ortalama ve varyans değerinin 0.5 ve 0.08 civarında olduğu gözlenmektedir.

CDF grafiği ile, üreteçte üretilen dizinin (1000000 veri) birörnek dağılımı ile ne kadar örtüştüğü gözlenmek istenir. Şekil 4.40’da yer alan CDF grafiğinde görüldüğü gibi, üreticinin birörnek dağılıma ait veri ile çakıştığı gözlenmektedir. Birörnek dağılıma ait beklenen ve üreteçlere ait gözlenen veriler örtüştüğü için, üreticinin birörnek dağılıma sahip diziler oluşturduğu değerlendirilmektedir.



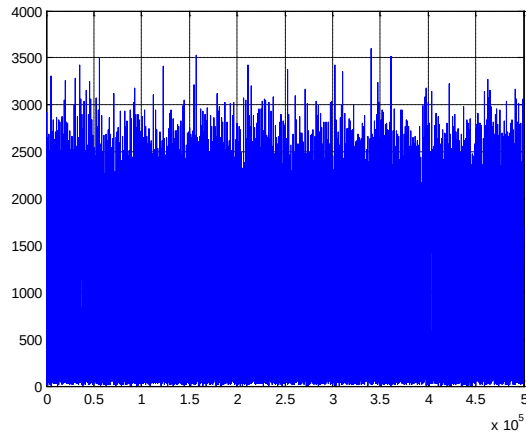
Şekil 4.40 MT Algoritması Strateji-1 CDF Grafiği

MT algoritmasına ait bir üreteç için üretilen tüm verinin grafiklere yansıyan dalgalanmalarını gözlemleyip kıyaslayabilmek adına CDF ve histogram grafiklerinin birlikte gösterilmiştir. Üreticinin birörnek dağılıma sahip rastsal diziler ürettiği, Çizelge 4.28’deki test sonuçları, Şekil 4.41’deki histogram ve Şekil 4.40’daki CDF ile anlaşılmıştır.



Şekil 4.41 MT Algoritması Strateji-1 Histogramı

MT algoritmasına ait üretici ile üretilen tüm veriye Spektral test yapılmıştır. Çizelge 4.28’de yer alan test sonucuna göre üreticinin Spektral testten geçtiği bilinmektedir. Üreticinin tüm verilerine uygulanan Spektral testten geçtiği Şekil 4.42’deki FFT grafiği ile desteklenmektedir.



Şekil 4.42 MT Algoritması Strateji-1 FFT Grafiği

4.4.2.2 Strateji-2 Testleri

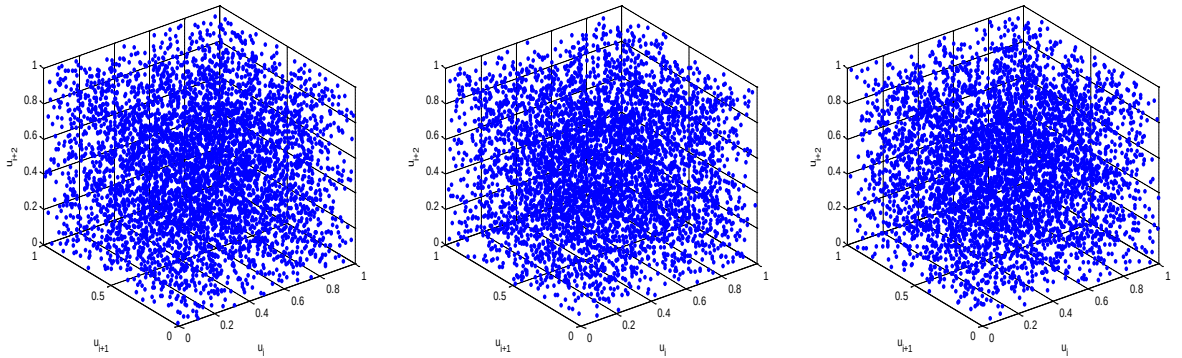
Örneklem sayısı ve bir örneklemdeki dizi uzunluğu seçimi ile algoritma tarafından üretilen dizinin tümü kullanılarak testler gerçekleştirilmiştir. Bu kısımda üretilen tüm veri, örneklem sayısı ve bir örneklemdeki dizi uzunluğu göz önüne alınarak bölünmüştür. Bölünen veriler üzerinde Frekans, Koşum, Spektral ve Ki-Kare testleri yapılmıştır.

Üretcin 200 örnekleme yapılan testler sonucunda elde edilen P-değerinin analiz yapılır. P-değeri analizi kapsamında Dizi Geçme Oranı (DGO) hesaplanır. 200 örneklem arasında kaçının sıfır hipotezini geçtiğine dair bir oran elde edilir ve sonuç olarak testten geçip geçmediğine bakılır. Çizelge 4.29’da görüldüğü gibi MT algoritmasına ait üretcin 200 örnekleme uygulanan; Frekans, Koşum, Spektral ve Ki-Kare testlerinin tümünden geçtiği gözlenmiştir.

Çizelge 4.29 MT Algoritması Strateji-2 Test Sonuçları-1

Ki- Kare DGO	Ki-Kare Test Sonucu	Frekans DGO	Frekans Testi Sonucu	Koşum DGO	Koşum Testi Sonucu	Spektral DGO	Spektral Test Sonucu
0,985	1	0,99	1	0,98	1	0,985	1

Üreteçten rastsal örneklem alınıp dizi elemanları arasında bir ilinti olup olmadığı, dizi elemanlarının periyodik özellikler sergileyip sergilemediğini incelemiştir. Dizi elemanları arasındaki ilinti, arka arkaya olan üç duruma ait veri (u_i, u_{i+1}, u_{i+2}) incelenerek tespit edilmeye çalışılmıştır. Rastsal olarak 48. , 115. ve 193. örneklem seçilerek olan 3-çokuzlu grafikler üzerinden gözlem yapılmıştır. Dizi elemanları arasındaki ilintinin gözlemlenebilmesi için Şekil 4.43’te yer alan 3-boyutlu grafikler incelenmiştir.



Şekil 4.43 MT Algoritması Strateji-2, Üretcin 48. , 115. ve 193. Örneklem

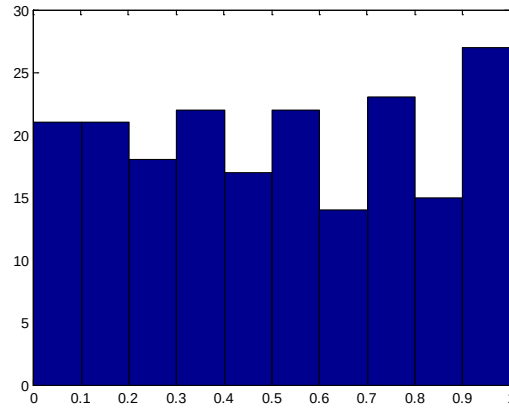
Şekil 4.43’te görüldüğü gibi üretcin rastsal seçilen hiçbir örnekleminde, dizi elemanları arasında bir ilinti ya da periyodik özellik gözlenmemiştir. Bu grafikler MT algoritması ile üretilen sözde rastsal dizi elemanları arasında ilinti gözlenmediği bilgisini desteklemektedir.

Tüm testler için P-değeri analizi kapsamında P-değerinin dağılımı hesaplanır. Üretcin 200 örneğine uygulanan Frekans, Spektral, Koşum ve Ki-Kare testleri sonucunda elde edilen P-değerlerine Ki-Kare testi uygulanarak P-değerlerinin dağılımına bakılır. MT algoritmasına ait üretcin P-değerleri dağılımına bakıldığında, üretcin örneklerinin Frekans testinden geçtiği Çizelge 4.30'da görülmektedir.

Çizelge 4.30 MT Algoritması Strateji-2 Frekans Testi

Normal Çıktı Yöntemi	
PD Frekans Testi H değeri	PD Frekans Testi P-değeri
0	0,628605

Üretcin 200 örneğine yapılan Frekans testinden elde edilen P-değerlerine Ki-Kare testi uygulandığında, testten geçilmiştir. P-değerleri üzerinde yapılan Ki-Kare testi sonuçları ve P-değerlerinin histogramı birlikte değerlendirildiğinde; Şekil 4.44'te yer alan histogramda görünen iniş-çıkışların; dizinin birörnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturmadığı anlaşılmaktadır. Bu sonuç Çizelge 4.30 ile de desteklenmektedir.



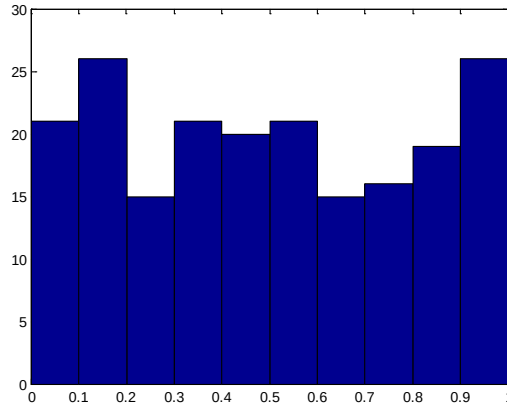
Şekil 4.44 MT Algoritması Strateji-2 Frekans Testi Histogramı

MT algoritmasına ait üretcin 200 örneğine yapılan Koşum testinden elde edilen P-değerlerine Ki-Kare testi uygulanmıştır. Test sonuçları Çizelge 4.31'de yer almaktadır.

Çizelge 4.31 MT Algoritması Strateji-2 Koşum Testi

Normal Çıktı Yöntemi	
PD Koşum Testi H değeri	PD Koşum Testi P-değeri
0	0,628329

P-değerleri dağılımına bakıldığında, üretcin örneklemelerinin Koşum testinden geçtiği Çizelge 4.31'de görülmektedir. Üretcin 200 örneğine yapılan Koşum testinin histogramında iniş-çıkışlar gözlenmektedir. Şekil 4.45'te gözlenen iniş-çıkışların; birörnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturmadığı anlaşılmaktadır.



Şekil 4.45 MT Algoritması Strateji-2 Koşum Testi Histogramı

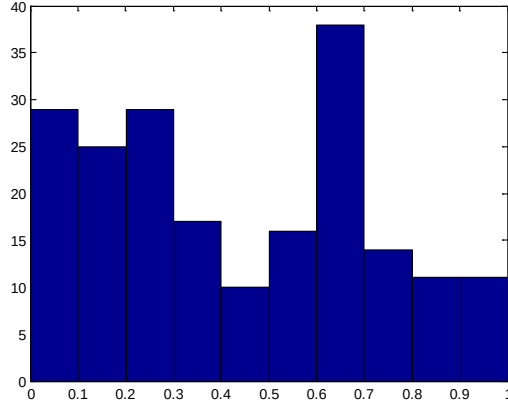
Üretcin örneklemelerine yapılan Spektral testinden elde edilen P-değerlerine Ki-Kare testi uygulandığında, testten geçilememiştir. P-değerleri üzerinde yapılan Ki-Kare testi sonuçları Çizelge 4.32'de yer almaktadır.

Çizelge 4.32 MT Algoritması Strateji-2 Spektral Testi

Normal Çıktı Yöntemi	
PD Spektral Testi H değeri	PD Spektral Testi P-değeri
1	< 0,01

Çizelge 4.32'deki test sonuçları ve P-değerlerinin histogramları birlikte değerlendirildiğinde; Şekil 4.46'da yer alan histogramda görünen iniş-çıkışların;

birörnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturduğu anlaşılmaktadır.



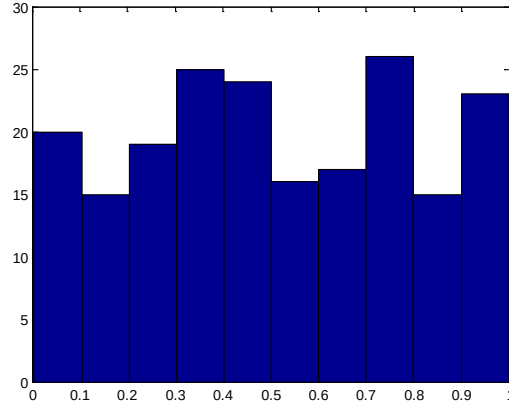
Şekil 4.46 MT Algoritması Strateji-2 Spektral Testi Histogramı

Ki-Kare testinden elde edilen P-değerleri üzerine yapılan Ki-Kare testi sonuçları Çizelge 4.33'te yer almaktadır.

Çizelge 4.33 MT Algoritması Strateji-2 Ki-Kare Testi

Normal Çıktı Yöntemi	
PD Ki-Kare Testi H değeri	PD Ki-Kare Testi P-değeri
0	0,527083

Çizelge 4.33'te görüldüğü gibi üreteç Ki-Kare testinden geçmiştir. P-değerleri üzerinde yapılan Ki-Kare testi sonuçları ve P-değerlerinin histogramları birlikte değerlendirilmektedir. Şekil 4.47'de yer alan histogramda görünen iniş-çıkışların, P-değerlerinin birörnek dağılıma sahip olmadığına dair belirgin göstergeler oluşturmadığı anlaşılmaktadır.



Şekil 4.47 MT Algoritması Strateji-2 Ki-Kare Testi Histogramı

Strateji-2'ye uygulanan Frekans, Koşum, Spektral ve Ki-Kare test sonuçları bir bütün halinde Çizelge 4.34'te verilmiştir.

Çizelge 4.34 MT Algoritması Strateji-2 Test Sonuçları

P_Değeri HT- ki-kare	P_Değeri p_değeri- ki-kare	P_Değeri HT-frek	P_Değeri p_değeri- frek	P_Değeri HT- koşum	P_Değeri p_değeri- koşum	P_Değeri HT-spek	P_Değeri p_değeri- spek
0	0,527083	0	0,628605	0	0,628329	1	< 0,01

Üretcin 200 örneğine yapılan testlerin P-değerlerine uygulanan Ki-Kare testi sonucunda, Çizelge 4.34'te görüldüğü gibi Spektral test dışındaki testlerden geçilmiştir.

Üretcin 200 örneğine uygulanan Spektral testten kalması örneklemelerin rastsal dizilerden oluşmadığını gösterir; ancak tüm verinin rastsal olmadığını söylememiz mümkün değildir. Çünkü Çizelge 4.28'de görüldüğü gibi üretcin tüm verilerine yapılan Ki-Kare testinden geçtiğinden, üretee ait tüm verinin birörnek dağılıma sahip rastsal bir dizi olmadığına dair belirgin göstergeler oluşturmadığı değerlendirilmektedir.

4.5 PRNG Algoritmalarının Karşılaştırılması

Bu tez kapsamında akış şifreleme uygulamalarında kullanılacak PRNG tabanlı anahtar dizisi üretme algoritmalarından önde gelen BBS, LCG, ICG ve MT algoritmaları olmak üzere 4 algoritma incelenmiştir. Çizelge 4.35'te LSFR tabanlı algoritmaların karşılaştırmaları verilmiştir.

Çizelge 4.35 PRNG Algoritmalarının Karşılaştırılması

	BBS	LCG	ICG	MT
Geliştirilme Yılı	1986	1951	1986	1997
Kriptografik anlamda güvenli mi (CSPRNG)	Evet	Hayır	Hayır	Hayır
Maksimum Periyot	$2p_2q_2$	m $m-1$	q	$2^{n\omega-r} - 1$, $2^{19937} - 1$ (MT19937)
Başlangıç parametresi seçimine hassasiyet	Maksimum periyoda sahip diziler elde etmek için hassastır	En hassas algoritmadır	Parametre ve örneklem seçimine hassas değildir	Başlangıç parametresinin seçimi kritik değildir ancak algoritmanın içinde kullanılan diğer parametreler maksimum periyodu sağlaması için kritik öneme sahiptir
LFSR / Mantıksal Kapıları	-	-	-	var
Gereken Parametreler	x_0, p, q	a, c, m, x_0	a, c, q, x_0	$n, m, r, \omega, a, b, c, x_0, l, s, t$ ve u
Algoritmanın taradığı aralık	2^{26}	2^{31}	2^{31}	2^{32}

İlinti gözlemleme	İlinti gözlenmez	İlinti gözlenebilir, kafes yapısı oluşabilir	İlinti gözlenmez	İlinti gözlenmez
Tahmin Edilebilirlik	Polinomsal zamanda tahmin edilemezlik özelliğine sahip	Tahmin edilebilir	Tahmin edilebilir	Tahmin edilebilir
Türevleri	Blum Micali	-	-	-
Çeşitleri	-	MCG MLCG Park-Miller	-	MT11213A, MT11213B, TT800
Algoritma hızı	Yavaş	Hızlı	Hızlı	Hızlı
Hesaplama Zorluğu	Matematiksel teoremlere dayandığından hesaplama yoğunluğu çoktur	Yoktur	Modüler çarpım tersinin alınması işlemi nedeni ile LCG'ye oranla hesaplama zorluğu vardır ancak BBS ile karşılaştırılmaz.	Yoğun matematiksel denklemler içermez, hesaplama zorluğu yoktur
Uygulama alanı	Kriptografik uygulamalarda, askeri ve diplomatik haberleşme alanlarında kullanılabilir	Bazı benzetim uygulamalarında, sözde rastsal sayı üretimi uygulamalarında kullanılabilir	Benzetim uygulamalarında kullanılabilir	Monte Carlo benzetim uygulamalarında, algoritma özet fonksiyonlar ile düzenlenerek kriptografik uygulamalarda, bazı programlama dillerinde sözde sayı üretici standart algoritma olarak kullanılabilir
Çıktısı	Doğrusal olmayan	Doğrusal	Doğrusal olmayan	Doğrusal
Türediği Algoritma	-	-	LCG	Twisted LFSR (TGFSR)

5 SONUÇ

Bu tez çalışması kapsamında, akış şifrelemede kullanılan LFSR ve PRNG tabanlı algoritmaların çalışma prensipleri incelenmiş, bu algoritmaların verimli çalışması için uygun parametreler araştırılmıştır. Seçilen parametreler doğrultusunda algoritmalarından elde edilen anahtar dizisine, istatistiksel testler uygulanarak dizinin istatistiksel özellikleri incelenmiştir. Parametre seçimine göre üretilen dizilerin periyotları, istatistiksel özellikleri ve dizi elemanları arasındaki ilinti gözlenmiştir.

Akış şifreleme algoritmaları LFSR tabanlı üreteçler ve PRNG tabanlı üreteçler olarak iki farklı grupta ele alınmıştır. LFSR tabanlı üreteçlerden; A5/1, A5/2 ve RC4 olmak üzere yaygın olarak kullanılan üç algoritma seçilerek incelenmiştir. PRNG tabanlı üreteçlerden ise BBS, LCG, ICG ve MT olmak üzere dört algoritma seçilerek incelenmiştir.

Bu algoritmaları modellemek için MATLAB yazılım ortamı kullanılmıştır. Algoritmaların ürettikleri dizilerin istatistiksel özelliklerini gözlemlediğimiz ve üreteçlerin kaliteleri hakkında bilgi edindiğimiz istatistiksel testler de yine bu yazılım geliştirme aracı ile kodlanmıştır.

LFSR algoritmalarında tanımlanan tap biti, tetiklenme biti ve geri beslemeli polinomlarda belirtilen uygun bitler seçildiğinde ve algoritma bu parametreler üzerine oturtulduğunda diğer parametrelerin seçimi kritik önem taşımamaktadır. Örneğin, A5/x algoritmalarında anahtarın ve çerçeve numarasının farklılaştırılması, elbette ki çıktıyı değiştirmektedir; ancak kritik bir önem taşımamaktadır. Farklı anahtarlar istatistiksel testler de farklı sonuçlara neden olmayacaktır. Bu sonuç RC4 algoritmasında anahtarın farklılaştırılması ile gözlemlenmiştir.

RC4 algoritması haricinde kalan A5/x algoritmaları birçok mantıksal kapı ve yazmaç içerdiğinden donanım tabanlı gerçekleştirmeye uygun algoritmalarlardır. RC4 algoritmasının ise veri üretim hızı ve kodlamadaki kolaylığı nedeni ile yazılım uygulamalarında tercih edilebileceği değerlendirilmiştir.

PRNG tabanlı üreteçlerde üretilen dizilerin, maksimum periyoda sahip olması için uygun parametre seçimi kritik önem taşımaktadır. Parametre seçimi dizilerin sadece periyodunu değil, istatistiksel özelliklerini ve birörnek dağılıma sahip olup

olmama özelliklerini de etkilemektedir. PRNG tabanlı üreteçler ile maksimum periyoda sahip diziler üretebilmesi için, LFSR tabanlı üreteçlere nazaran uyulması gereken daha çok kural ve matematiksel kısıt bulunmaktadır.

BBS haricindeki algoritmalar ile maksimum periyoda sahip dizilerin üretilmesini sağlayacak uygun parametre setleri literatür taraması sonucu elde edilmiştir. Ancak BBS algoritması için böyle bir bilgiye ulaşılammıştır. Algoritmanın maksimum periyodu sağlayacak kurallara uygun parametre setleri bulunmuştur. Elde edilen parametre setlerinden üç tanesine testler uygulanmıştır. Yapılan istatistiksel testler sonucunda BBS algoritmasının güçlü, tahmin edilebilirlikten uzak, örneklem seçimine diğer algoritmalar kadar hassas olmayan, ilintisiz yapısı gözlenmiş ve başlangıç parametrelerine göre dizilerin periyotları hesaplanmıştır. Buna ek olarak BBS algoritmasında eşlik biti ile çıktı alarak diğer algoritmalarda kullanılandan farklı bir yöntem ile çıktı alınmış ve bu çıktılara testler yapılmıştır. Örneklemelere yapılan Spektral testten, eşlik biti çıktı alma yöntemi ile alınan çıktılar geçerken, normal çıktı yöntemi ile alınan çıktılar geçememektedir. Bu durum, çıktı alma yöntemi farklılaştığında, çıktıların istatistiksel özelliklerinin değiştiğini göstermektedir.

LCG algoritması, BBS algoritması gibi güçlü ve kriptografik anlamda güvenilir bir yapıya sahip değildir. LCG algoritmasında parametre ve örneklem seçiminin; çıktının istatistiksel özellikleri, tahmin edilebilirliği, ilintili yapısı ve periyodu üzerindeki etkisi test sonuçları ile gözlemlenmiştir. LCG algoritmasında, dizi elemanları arasında ilinti oluşturacağı bilinen bir üreteç ile iyi istatistiksel özelliklere sahip bir üreteç karşılaştırılarak parametre seçimlerinin önemi gözlenmiştir.

LCG algoritmasında oluşan kafes yapısını önlemek için ICG algoritması geliştirilmiştir. ICG algoritmasının yapısı sayesinde üç farklı üreteçte ilinti yapısı gözlenmemiştir. Uygun parametreler seçilerek istatistiksel özellikleri iyi olan diziler üretilbileceği anlaşılmıştır. Bunun yanı sıra, uygun parametreler seçilmeden de; maksimum periyoda sahip ancak istatistiksel özellikleri iyi olmayan diziler üretilbileceği gözlemlenmiştir. Bu da periyot uzunluğunun etkin olduğu uygulamalarda parametre seçiminin çok kritik olmadığını göstermektedir. Bununla beraber ICG algoritmasına ait 3. üreteçte olduğu gibi, örneklemeleri tüm testlerden

kalan ancak tüm veriye uygulanan testlerin bir kısmından geçen bir üreticinin birörnek dağılıma sahip rastsal diziden oluşmadığına dair belirtilere rastlanmıştır.

MT algoritması ile üretilen dizilerin maksimum periyoda ulaşabilmesi için algoritma içinde yer alan parametrelerin uygun seçilmesi kritik önem taşırken, başlangıç parametresinin seçimi kritik önem taşımamaktadır. Bu durum esas alındığında üretilen dizilerin $2^{19937}-1$ periyoduna ulaşılacağı bilinmektedir. Üretilen dizilerin istatistiksel özellikleri iyi olduğundan farklı programlama dillerinde sözde rastsal sayı dizisi üreten standart algoritma olarak yer almaktadır. Üretilen diziye yapılan testler sonucunda, bu algoritma tarafından üretilen dizilerin istatistiksel özellikleri, periyodu ve parametre seçimi ile ilgili yaklaşımın doğru olduğu gözlemlenmiştir.

Özetle, PRNG tabanlı üreteçlerde algoritmaların güçleri, hesaplama zorlukları, tahmin edilebilirlikleri değişmektedir. Buna ek olarak, algoritmalara ait üreteçlerin parametrelerinin seçimi; üretilen dizinin periyodunu, dizi elemanları arasındaki ilintiyi, dolayısıyla dizinin tahmin edilebilirliğini ve istatistiksel özelliklerini değiştirmektedir. Uygun parametrelerin seçimi ile birörnek dağılıma sahip rastsal sayılar üretmek mümkündür. Bu nedenle BBS algoritması dışında kalan algoritmaların çıktıları doğrudan şifreleme uygulamalarında olmasa da anahtar dizisi üretiminde kullanılmaktadır. BBS algoritması ise CSPRNG olduğundan şifreleme uygulamalarında da kullanılabilir. PRNG tabanlı üreteçler, mantıksal kapı veya yazmaç gibi donanımsal yapılardan uzak olduklarından; gelişmiş bir yazılım ortamında gerçekleştirmeye uygun algoritmalarlardır.

LFSR tabanlı üreteçler ile birörnek dağılıma sahip sözde rastsal diziler elde edildiği gözlemlenmiştir. Bu algoritmalara ait üreteçlerden iyi istatistiksel özelliklere sahip diziler üretilse de; LFSR yapısının kısıtlayıcı (19 bit, 22 bit, 23 bit) olması periyodunun çok uzun olmamasına neden olmaktadır. Buna ek olarak tetikleme kontrolündeki zayıflıklar nedeni ile algoritmaların kırılabilirliği de bilinmektedir. Bu nedenle doğrudan şifreleme uygulamalarında olmasa da; anahtar dizisi üretme algoritmaları olarak kullanılmaktadır. Mantıksal kapı veya yazmaç gibi bileşenler nedeni ile donanıma uygun algoritmalarlardır.

KAYNAKLAR DİZİNİ

[Ahmad, 2009] Musheer Ahmad and Izharuddin, 2009, Enhanced A5/1 Cipher with Improved Linear Complexity, IEEE

[Ahmed, 2009] Fayyaz Ahmed, Mudassar Imran, 2009, Cryptographic Analysis of GSM Network, IEEE

[Akkemik, 2008] Esen Akkemik, Orhun Kara, 2008, Düzensiz Şifreleme Algoritmasının Gerçek Zamanlı Kripto Analizi

[Altan, 2004] K. Altan, K. Kaşkaloğlu, N. Kındap, Ç. Özakin, Z. Saygı, M. Yıldırım, S. Yıldız, Kriptoloji Seminer Notları, 2004, Uygulamalı Matematik Enstitüsü, Kriptografi Bölümü ODTÜ

[Bakhtiari, 2011] M. Bakhtiari, A. Maarof, 2011, An Efficient Stream Cipher Algorithm for Data Encryption, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1

[Barkan, 2003] E. Barkan, E. Biham, N. Keller, 2003, Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication, in Advances in Cryptology - CRYPTO 2003, Springer Berlin / Heidelberg. p. 600-616

[Başkök, 2007] Melek Dirayet Başkök, 2007, AES Şifreleme Algoritmasının Modellenmesi, Gazi Üniversitesi

[Biham, 2000] E. Biham, O. Dunkelman, 2000, Cryptanalysis of the A5/1 GSM stream cipher, Proceedings of Indocrypt'2000, Lecture Notes in Computer Science, vol. 1977, pp. 43–51

[Bircan, 2003] H. Bircan, Y. Karagöz, Y. Kasapoğlu, 2003, Ki-Kare ve Kolmogorov Smirnov Uygunluk Testlerinin Simulasyonu ile Elde Edilen Veriler Üzerinde Karşılaştırılması, Cumhuriyet Üniversitesi, İktisadi ve İdari Bilimler Dergisi, Cilt 4, Sayı 1

[Biryukov, 2001] A. Biryukov, A. Shamir, D. Wagner, 2001, Real Time Cryptanalysis of A5/1 on a PC, Proceedings of Fast Software Encryption, Editors. 2001, Springer Berlin / Heidelberg. p. 37-44

[Blum, 1986] L. Blum, M. Blum and M. Shub, 1986, A Simple Unpredictable Pseudo-Random Number Generator, SIAM J.COMPT. Vol. 15 No.2

[Blum, 1984] Manuel Blum and Silvio Micali, 1984, How To Generate Cryptographically Strong Sequence Of Pseudo-Random Bits, SIAM J.COMPT. Vol.13 No.4.

[Boyar, 1989] Joan Boyar, 1989, Inferring Sequences Produced by Pseudo-Random Number Generators, Journal of the Association for Computing Machinery. No.1, pg 129-141

[Briceno, 1999] Marc Briceno; Ian Goldberg; David Wagner, 1999, A pedagogical implementation of the GSM A5/1 and A5/2 "voice privacy" encryption algorithms

[Büyüksaraçoğlu, 2011] Fatma Büyüksaraçoğlu, Ercan Buluş, 2011, Sözde Rastsal Sayı Üretiminin Kriptografik Açından İncelenmesi

[Challis, 2008] Vivien Challis, 2008, Pseudo Random Number Generators

[Cusik, 1994] Thomas W. Cusik, 1994, Properties of the Pseudorandom Number Generator, IEEE Transaction On Information Theory Vol.41 No.4, State University of New York at Buffalo, IEEE Log Number 9412179

[Ding, 1997] C. Ding, 1997, Blum Blum Shub Generator, IEE Electronics Letters Online No: 19970440, Turku Centre for Computer Science, DataCity, Lemminkaisenkatu 14 A, FIN-20252 Turku, Finland

[Eichenauer, 1991] Jürgen Eichenauer, G. Herrmann, 1991, Inverse Congruential Pseudorandom Numbers Avoid The Planes, American Mathematical Society, Mathematics of Computation Vol. 56 No.193, pg. 297-301

[Eichenauer, 1986] J. Eichenauer, J. Lehn, 1986 A non linear congruential pseudorandom number generator Statist. Papers 27, pg 315-326.

[Ek Dahl, 2003] P. Ek Dahl, T. Johansson, 2003, Another Attack on A5/1, IEEE Transactions on Inform. Theory, vol. 49, pp. 1-7

[Erguler, 2005] Imran Erguler, Emin Anarim, 2005, A Modified Stream Generator for the GSM Encryption Algorithms A5/1 and A5/2, 13th European Signal Processing Conference (EUSIPCO 2005)

[Ferrenberg, 1992] A. M. Ferrenberg, D. P. Landau, Y. J. Wong, 1992, monte carlo simulations: hidden error from “good” random number generator, PHYs. Rev. Lett. 69, 3382-3384

[Fischer, 2008] Simon Fischer, 2008, Analysis Of Lightweight Stream Ciphers

[Frieze, 1984] A.M. Frieze, R. Kannan, J.C. Lagarias, 1984, Linear Congruential Generators Do Not Produce Random Sequences, pp.480-484, 25th Annual Symposium on Foundations of Computer Science (FOCS 1984)

[Galanis, 2005] M. Galanis, P. Kitsos, K. Giorgos, N. Sklavos, C. Goutis, 2005, Comparison of the Hardware Implementation of Stream Ciphers , The International Arab Journal of Information Technology, Vol.2, No.4

[Gendrullis, 2008] T. Gendrullis, M. Novotný, A. Rupp, 2008, A Real-World Attack Breaking A5/1 within Hours, Cryptographic Hardware and Embedded Systems – CHES 2008, E. Oswald and P. Rohatgi, Editors. 2008, Springer Berlin / Heidelberg. p. 266-282.

[Gentle, 2003] James E. Gentle, 2003, Random Number Generation and Monte Carlo Methods, 2nd Edition, Statistics and Computing, Springer

[Glen, 2002] Amy Glen, 2002, On The Period Length of Pseudorandom Number Sequences

[Golic, 1997] J. Golic, 1997, Cryptanalysis of Alleged A5 Stream Cipher, Proceedings of Eurocrypt'97, Lecture Notes in Computer Science, vol. 1233, pp. 239-255.

[Golic, 2002] J. Golic, V. Bagini, and G. Morgari, 2002, Linear Cryptanalysis of Bluetooth Stream Cipher, in Advances in Cryptology — EUROCRYPT 2002, L. Knudsen, Editor. 2002, Springer Berlin / Heidelberg. p. 238-255.

[Griffin, 2000] Frances Griffin and Igor E. Shparlinski, 2000, On the Linear Complexity Profile of the Power Generator, IEEE TRANSACTION ON INFORMATION THEORY, VOL.46, NO.6

[Gürkaş, 2005] Zeynep Gürkaş, 2005, Kablosuz Güvenlik Protokollerinin Karşılaştırmalı Analizi, İstanbul Üniversitesi

[Hellekalek, 1994] Peter Hellekalek, Marcus Mayer, Andreas Weingartner, 1994, Implementation of Algorithms for IMP Polynomials, Research Institute for Software Technology, University of Salzburg, Austria

[Hellekalek, 1995] Peter Hellekalek, 1995, Inverse Pseudorandom Number Generators: Concepts, Results And Links, Proceedings of the 1995 Winter Simulation Conference pg. 255-262

[Kahaner, 1977] D. Kahaner, 1977, Numerical Methods and Software, Prentice Hall Series in Computational Mathematics

[Knuth, 1981] D. Knuth, 1981, The Art of computer programming, Addison-Wesley Vol. 2 Seminumerical Algorithms

[Kumar, 2006] S. Kumar, et al., 2006, Breaking Ciphers with COPACOBANA –A Cost-Optimized Parallel Code Breaker, Cryptographic Hardware and Embedded Systems - CHES 2006, L. Goubin and M. Matsui, Editors. 2006, Springer Berlin / Heidelberg. p. 101-118

[Law, 1991] Averil M. Law, W. David Kelton, 1991, Simulation Modeling and Analysis pg 248-250, Mc Graw-Hill Series in Industrial Engineering and Management Science

[L'Ecuyer, 2007] P. L'Ecuyer, R. Simard, 2007, TestU01: A C Library for Empirical Testing of Random Number Generators

[L'Ecuyer, 2004] P. L'Ecuyer, 2004, Uniform Random Number Generation, Universite de Montral, Canada

[L'Ecuyer, 1999] Pierre L'Ecuyer, Richard J. Simard, 1999, Beware of Linear Congruential Generators with Multipliers of the form $a = \pm 2^q \pm 2^r$, ACM Trans. Math. Softw. 25(3): 367-374, Universite de Montral, Canada

[L'Ecuyer, 1999] Pierre L'Ecuyer, 1999, Tables of Linear Congruential Generators of Different Size and Good Lattice Structure, American Mathematical Society

[L'Ecuyer, 1998] P. L'Ecuyer, 1998, Good Parameters and Implementations for Combined Multiple Recursive, Universite de Montral, Canada

[L'Ecuyer, 1993] P. L'Ecuyer, F. Blouin, R. Couture, 1993, A Search for Good Multiple Recursive Generators, ACM Trans. on Modeling and Computer Simulation, pg 87—98

[Lendl, 2001] Otmar Lendl, Josef Leydold, 2001, PRNG, Generating Random Numbers Ver: 3.0.2

[Li, 2012] Chung-Chih Li, Bo Sun, 2012, Using Linear Congruential Generators for Cryptographic Purpose

[Matsumoto, 2012] M. Matsumoto, T. Nishimura, 2012, Mersenne Twister Homepage <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html>

[Matsumoto, 1998] Makoto Matsumoto, Takuji Nishimura, 1998, Mersenne Twister: A 623 Dimensionally Equidistributed Uniform Pseudo Random Number Generators, ACM Transaction on Modeling and Computer Simulation, Vol. 8, No. 1 pg 3-30.

[Matsumoto, 1992] Makoto Matsumoto, Yoshihura Kurita, 1992, Twisted GFSR Generators, Rev No. 15

[Melason, 2010] Steven Melason, H. P. Yered-Juarez, 2010, European Option Pricing Using A Combined Inversive Congruential Generator, Dep. of Mathematics and Statics, California State University

[Moore, 1990] D. S. Moore, 1990, Uncertainty In L. S. Steen (Ed.), On the shoulders of giants: New approaches to numeracy pg. 95-137. Washington, DC: National Academy Press

[Mills, 2003] Anna Mills, 2003, Random Number Generators – a brief assessment of those available

[Moreau, 1996] Thierry Moreau, 1996, A practical “perfect” pseudo-random number generator, Article submitted to Computer in Physics,

[NIST, 2010], A Statistical Test Suite For Random And Pseudorandom Number Generators For Cryptographic Applications, Special Publication 800-22 Revision 1a

[Özkaynak, 2006] Fatih Özkaynak, A. Bedri Özer, 2006, Lojistik Harita ile Rasgele Sayı Üretilmesi ve İstatistikî Yöntemlerle Sınanması, Journal of İstanbul Kültür University, Vol.4, No.3, 129-133

[Pacati, 2012] Claudio Pacati, Uniform Random Number Generators

[Pandya, 2001] Raj Pandya, 2001, Mobile and Personal Communication Systems and Services, IEEE PRESS

[Park, 1988] Stephen K. Park, Keith. W. Miller, 1988, Random Number Generators : Good Ones are Hard to Find, Communications of the ACM/ Vol. 31/ Number 10

[Petrovic, 2002] S. Petrovic, A. Sabater-Fuster, 2002, An improved Cryptanalysis of the A5/2 Algorithm for Mobile Communications

[Petrovic, 2000] S. Petrovic, A. Sapater- Fuster, 2000, Cryptanalysis of the A5/2 Algorithm, Cryptology ePrint Archive, Report 2000/052,

[Sankaliya, 2011] Alpesh R. Sankaliya, V. Mishra (Senior MIEEE), Abhilash Mandloi, 2011, Implementation of Cryptographic Algorithms for GSM Cellular Standard, Ganpat University Journal of Engineering & Technology Vol. 1, Issue-1

[Sankaliya, 2011] Alpesh R. Sankaliya, V. Mishra, Abhilash Mandloi, 2011, Implementation of Cryptographic Algorithms for GSM UMTS Systems, International Journal of Network Security & Its Applications (IJNSA) Vol. 3, No. 6

[Schneier, 2003] Bruce Schneier, Neils Ferguson, 2003, Practically Cryptography, Wiley Publishing, Inc, Indianapolis, Indiana

[Schneier, 1996] Bruce Schneier, 1996, Applied Cryptography, Protocols, Algorithms, Source Code in C

[Stallings, 1991] William Stallings, 1991, Cryptography and Network Security, Pearsen Education

[Sulak, 2011] Fatih Sulak, 2011, Statistical Analysis Of Block Ciphers And Hash Functions, Philosophy of Doctorate in Department of Cryptography, Middle East Technical University

[Turan, 2008] M. S. Turan, 2008, On Statistical Analysis of Synchronous Stream Ciphers. PhD thesis, Middle East Technical University

[Wikipedia, 2011] Wikipedia Homepage, 2011, RC4 and Test Vectors section (2.4)
<http://en.wikipedia.org/wiki/RC4>

ÖZGEÇMİŞ

Adı Soyadı : Büke ÇAKIR

Doğum Yeri : Ankara

Doğum Yılı : 8 Ekim 1986

Medeni Hali : Bekar

Eğitim ve Akademik Durumu:

Lise : 2000 – 2003 TED Ankara Koleji

Lisans : 2004 – 2008 Başkent Üniversitesi, Elektrik ve Elektronik Mühendisliği
Bölümü

Yüksek Lisans : 2009 – 2012 Hacettepe Üniversitesi, Elektrik ve Elektronik
Mühendisliği Bölümü

Yabancı Dil : İngilizce (iyi seviye), Almanca (temel seviye)

İş Tecrübesi:

2008 - 2009 Selex Kominkasyon A.Ş. Konfigürasyon Mühendisi (5 ay)

2009 - 2012 ROKETSAN A.Ş., Sistem Mühendisi

