T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BLOK ŞİFRELER İÇİN CEBİRSEL İKİLİ DOĞRUSAL DÖNÜŞÜM TASARIMI VE MODERN BİR BLOK ŞİFREYE UYGULANMASI

Bora ASLAN

DOKTORA TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ

ANABİLİM DALI

DANIŞMAN

YRD. DOÇ. DR. M. TOLGA SAKALLI

EDİRNE, 2013

BLOK ŞİFRELER İÇİN CEBİRSEL İKİLİ DOĞRUSAL DÖNÜŞÜM TASARIMI VE MODERN BİR BLOK ŞİFREYE UYGULANMASI

Bora ASLAN

DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

2013
TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

T.Ü. Fen Bilimleri Enstitüsü onayı

Prof. Dr. Mustafa ÖZCAN
Fen Bilimleri Enstitüsü Müdürü

Bu tezin Doktora tezi olarak gerekli şartları sağladığını onaylarım.

Yrd.Doç.Dr. M. Tolga SAKALLI
Anabilim Dalı Başkanı

Bu tez tarafımca (tarafımızca) okunmuş, kapsamı ve niteliği açısından bir Doktora tezi olarak kabul edilmiştir.

Yrd.Doç.Dr. M. Tolga SAKALLI
Anabilim Dalı Başkanı

Bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından Bilgisayar Mühendisiliği Anabilim Dalında bir Doktora tezi olarak oy birliği/oy çokluğu ile kabul edilmiştir.

Jüri Üyeleri (Ünvan, Ad, Soyad):	İmza
Prof. Dr. Mustafa ÖZCAN	……
Doç.Dr. Yılmaz KILIÇASLAN	……
Doç.Dr. Ertan ARDA	……
Yrd.Doç.Dr. M. Tolga SAKALLI	……
Yrd.Doç.Dr. Sedat AKLEYLEK	……

Tarih: 03/05/2013

T.Ü.FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DOKTORA PROGRAMI

DOĞRULUK BEYANI

İlgili tezin akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin kaynak gösterilerek ilgili tezde yer aldığını beyan ederim.

07.05.2013

Bora ASLAN

Doktora Tezi

Blok Şifreler için Cebirsel İkili Doğrusal Dönüşüm Tasarımı ve Modern Bir Blok Şifreye Uygulanması

Trakya Üniversitesi Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

Blok şifreleme algoritmaları orijinal metni veya şifreli metni bloklara bölerek şifreleme/şifre çözme işlemi yaparlar. Blok şifreler, karıştırma (confusion) ve yayılım (diffusion) tekniklerine dayanır. Karıştırma, şifreli metin ve açık metin arasındaki ilişkiyi gizlemeyi amaçlarken, yayılım açık metindeki izlerin şifreli metinde sezilmemesini sağlamak için kullanılır. Karıştırma ve yayılım, sırasıyla yer değiştirme kutuları (S-kutuları) ve doğrusal dönüşüm işlemleri ile gerçekleştirilir. Yayılım elemanları olan doğrusal dönüşümler şifre tasarımında kullanılan önemli elemanlardır.

Bu tez çalışmasında kriptografik özellikleri iyi involutif veya involutif olmayan 8×8 ve 16×16 boyutunda doğrusal dönüşümlerin elde edilebilmesi için bir cebirsel yöntem geliştirilecektir. Bu yöntem ile kriptografik özellikleri iyi olan olası tüm doğrusal dönüşümlerin listelenmesi ve modern bir şifreleme algoritmasına uygulanması amaçlanmaktadır. Bunun yanında elde edilecek doğrusal dönüşümlerin önemli saldırı tekniklerine karşı dayanıklılığının incelenmesi de amaçlanmaktadır. Bu tasarlanacak doğrusal dönüşümlerin ikili elemanlardan oluşacak doğrusal dönüşümler olarak seçilmesinin amacı, geliştirilecek şifreleme algoritmasının yazılım performansının üst düzeyde tutmaktır.

Tez on bölüme ayrılmıştır. 1. bölümünde kriptografi bilimine kısa bir giriş yapılmıştır. 2.bölümde, tezin anlaşılabilmesi için gerekli olan matematik alt yapı verilmiştir. 3. bölümde, literatürde yer alan bazı önemli şifreleme algoritmalarında kullanılan doğrusal dönüşümler incelenmiştir. 4. bölümde İkili Doğrusal dönüşüm

tasarımı için geliştirilen yeni bir yöntem tanıtılmaktadır. 5. bölümde, geliştirilen yeni yöntem ile kriptografik özellikleri iyi olan 8×8 ikili doğrusal dönüşüm tasarımı gerçekleştirilmiştir. 6. bölümde, yine geliştirilen yeni yöntem ile 16×16 ikili doğrusal dönüşüm tasarımı gerçekleştirilmiştir. 7. bölümde 16×16 boyutunda, dallanma sayısı 7 ve sabit nokta sayısı 1 olan ikili doğrusal dönüşüm tasarımı yapılmıştır. 8. bölümde, ARIA blok şifreleme algoritması tanıtılmış ve önceki bölümlerde üretilen bir doğrusal dönüşüm, ARIA blok şifresi içerisine yerleştirilmiştir. 9. bölümde tez esnasında geliştirilen ikili doğrusal dönüşümler kullanılarak geliştirilen sadece XOR tabanlı, 128-bit blok uzunluğuna sahip TRAKYA blok şifre uygulaması tanıtılmıştır. 10. bölümde ise tez çalışmasında elde edilen sonuçlar verilmektedir.

Yıl: 2013

Sayfa Sayısı:136

Anahtar Kelimeler: Kriptografi; Blok Şifreler; İkili Matrisler; Dallanma Sayısı; Sabit Noktalar; Doğrusal Dönüşümler

Phd Thesis

Algebraic Construction of Binary Linear Transformations for Block Ciphers and Implementation of a Constructed Binary Linear Transformation in a Modern Block Cipher

Trakya University Institute of Natural Sciences

Department of Computer Engineering

ABSTRACT

Block ciphers encrypt or decrypt the plain text or cipher text by dividing them into blocks. Block ciphers are based on confusion and diffusion techniques. Confusion aims at concealing the relation between cipher text and plain text while diffusion is used for the marks on plain text not to be found in the cipher text. Confusion and diffusion are enabled with the help of S- boxes and linear transformation processes, respectively. Linear transformations are diffusion elements which are important elements used in code design.

In this thesis an algebraic method is developed to construct 8×8 and 16×16 binary linear transformations, which have cryptographic good features, with involution or without involution properties. By means of this method, it is aimed to generate and list all possible binary linear transformations, which have good cryptographic properties and to apply one of them to a modern encryption algorithm. Also, the resistance of the proposed binary linear transformations against some attacks such as linear cryptanalysis, differential cryptanalysis is also analyzed/studied. The reason for choosing these binary linear transformations as binary elements is to keep the level of software performance of the encryption algorithm at its highest level.

The thesis is composed of ten parts. In the first part, there is a brief introduction to the science of cryptography. In the second part, there is mathematical information for the study to be understood. In the third part, linear transformations that are used in

significant encoding algorithms in the literature are analyzed. In the fourth part, a new method developed for constructing binary linear transformations is presented. In the fifth part, the construction of 8×8 binary linear transformations with good cryptographic features is actualized via the method proposed. In the sixth part, with the help of this new method, the construction of 16×16 binary linear transformations is actualized. In the seventh part, the construction of 16×16 binary transformations of branch number 7 with one fixed point is presented. In the eighth part, the ARIA block encryption algorithm is introduced and one of the linear transformations produced in the previous parts is placed inside the ARIA block cipher. In the ninth part, the TRAKYA block cipher application, which is developed using the binary linear transformations made in the course of thesis, is presented. It is only XOR based and has 128-bit block length. In the tenth part, there is the list of results gained at the end of this study

Year: 2013

Number of Pages: 136

Keywords: Cryptography; Block cipher; Binary matrix; Branch number; Fixedpoints; Linear Transformation

TEŞEKKÜR

Tez çalışmam sırasında bana destek olan ve yardımlarını esirgemeyen kişilere buradan teşekkür etmeyi bir borç bilirim.

Öncelikle değerli hocam ve danışmanım Yrd. Doç. Dr. M. Tolga SAKALLI' ya bana yaptığı tüm katkılardan ve dostluğundan dolayı çok teşekkür ederim.

Bu tezin izleme komitesinde yer alan Doç. Dr. Yılmaz KILIÇASLAN'a ve Doç. Dr. Ertan ARDA'ya tezime yaptıkları değerli katkılarından dolayı teşekkürlerimi sunarım. Tez jürimde görev alarak beni onure eden ve tezime yaptılıkları bilimsel katkılardan dolayı Prof.Dr. Mustafa ÖZCAN'a ve Yrd.Doç.Dr. Sedat AKLEYLEK'e teşekkür ederim.

Tezimi proje olarak kabul eden (2011/163) ve sağladığı maddi katkılardan dolayı Trakya Üniversitesi yönetimine ve Bilimsel Araştırma Projeleri Birimine teşekkür ederim.

Eğitim öğretim hayatım boyunca beni sürekli motive eden ve her türlü desteği veren canım annem Zeynep ASLAN'a ve canım babam Esvet ASLAN'a sonsuz sevgi ve teşekkürlerimi sunarım.

Son olarak, 2000 yılından beri hayatını benimle paylaşan değerli eşim Füsun YAVUZER ASLAN'a tezime sağladığı manevi destek için, hayatıma kattığı anlam için ve iyi-kötü tüm zamanlarda yanımda olduğu için sonsuz teşekkürlerimi ve sevgimi sunarım.

İÇİNDEKİLER

ÖZET i
ABSTRACT iii
TEŞEKKÜR v
İÇİNDEKİLER vi
SİMGELER VE KISALTMALAR ix
ŞEKİLLER LİSTESİ x
TABLOLAR LİSTESİ xii
BÖLÜM 1 1
GİRİŞ 1
1.1. Kriptografi 1
1.2. Kriptografi Tarihi 2
1.3. Modern Kriptografi 5
1.4. Blok Şifreler 6
1.4.1. Anahtar Büyüklüğü 7
1.4.2. S-Kutuları (Substitution-Boxes) 8
1.4.3. Doğrusal Dönüşümler 8
1.5. Kriptanaliz 8
1.5.1. Doğrusal Kriptanaliz 9
1.5.2. Diferansiyel Kriptanaliz 10
1.6. Tezin Önemi 10
BÖLÜM 2 12
MATEMATİK ALT YAPI 12
BÖLÜM 3 27
Blok Şifrelerde Kullanılan Doğrusal Dönüşümlerin İncelenmesi 27
3.1. Giriş 27
3.2. AES Şifresinde Kullanılan Doğrusal Dönüşüm 27
3.3. Khazad Şifresinde Kullanılan Doğrusal Dönüşüm 30
3.4. Camellia Şifresinde Kullanılan Doğrusal Dönüşüm 31
3.5. ARIA Şifresinde Kullanılan Doğrusal Dönüşüm 34

3.6. SERPENT Şifresinde Kullanılan Doğrusal Dönüşüm 36
3.7. PRESENT Şifresinde Kullanılan Doğrusal Dönüşüm 39
3.8. Sonuçlar 41
BÖLÜM 4 43
İkili Doğrusal Dönüşüm Tasarımı için Yeni Bir Yöntem 43
4.1. Giriş 43
4.2. 2×2 Boyutundaki Matrislerin 8×8 Boyutundaki İkili Matrislere Dönüşümü 48
4.3. 4×4 Boyutundaki Matrislerin 16×16 Boyutundaki İkili Matrislere Dönüşümü 51
4.4. Sonuçlar 59
BÖLÜM 5 60
8×8 Boyutunda Kriptografik Özellikleri İyi İkili Doğrusal Dönüşüm Tasarımı 60
5.1. Giriş 60
5.2. 8×8 Boyutunda İkili Doğrusal Dönüşüm Tasarımı 60
5.3. 8×8 Boyutunda İnvolutif İkili Doğrusal Dönüşüm Tasarımı 63
5.4. Sonuçlar 68
BÖLÜM 6 71
16×16 Boyutunda Kriptografik Özellikleri İyi İkili Doğrusal Dönüşüm Tasarımı 71
6.1. Giriş 71
6.2. 16×16 Boyutunda İkili Doğrusal Dönüşüm Tasarımı 71
6.3. Sonuçlar 81
BÖLÜM 7 83
16×16 Boyutunda Dallanma Sayısı 7 ve Sabit Nokta Sayısı 1 olan İkili Doğrusal Dönüşüm Tasarımı 83
7.1. Giriş 83
7.2. 16×16 Boyutunda Dallanma Sayısı 7 ve Sabit Nokta Sayısı 1 Olan İkili Doğrusal Dönüşüm Tasarımı 86
7.3. Sonuçlar 89
BÖLÜM 8 94
Üretilen Matrislerin ARIA Blok Şifresine Uygulanması 94
8.1. Giriş 94
8.2. ARIA Blok Şifreleme Algoritması 94
8.2.1. Yer Değiştirme Katmanı 95
8.2.2. Yayılım Katmanı 98

8.2.3. Örnek Şifreleme İşlemleri 100
8.3. Tez Çalışmasında Üretilen Doğrusal Dönüşümlerin ARIA Blok Şifresine Uygulanması 102
8.3.1. Örnek şifreleme İşlemleri 103
BÖLÜM 9 106
Yeni Bir Blok Şifre Uygulaması: TRAKYA 106
9.1. Giriş 106
9.2. Blok Şifre TRAKYA 106
9.2.1. Yer Değiştirme Katmanı 110
9.2.2. Yayılım Katmanı 114
9.2.3. Anahtar Planlama 117
9.2.4. Örnek Şifreleme İşlemleri 122
9.3. Geliştirilen Blok Şifrenin Diferansiyel ve Doğrusal Kriptanalize Karşı Güvenlik Değerlendirmesi 127
9.4. Sonuçlar 128
BÖLÜM 10 129
SONUÇLAR 129
KAYNAKLAR 132
TEZ SIRASINDA YAPILAN ÇALIŞMALAR 136

SİMGELER VE KISALTMALAR

AES: Advanced Encryption Standard

DES: Data Encryption Standard

ECC: Elliptic Curve Cryptography

S-Box: Yer Değiştirme Kutuları (Substitution Box)

FIPS: Federal Information Processing Standard

FPN: Sabit Nokta Sayısı (Fixed Point Number)

IDEA: International Data Encryption Algorithm

MD: Message-Digest Algorithm

MDBL: Maximum Distance Binary Linear

MDS: Maximum Distance Seperable

NIST: National Institute of Standards and Technology

NTT: Nippon Telegraph and Telephone Corporation

SHA: Secure Hash Algorithm

SPN: Substitution-Permutation Network

XOR: Exclusive Or

GF(2): Elemanları 0 veya 1 olan Galois Cismi

$GF(2^n)$: Elemanları 0 veya 1 olan *n* elemanlı Galois Cismi

$\{0,1\}^n$: Elemanları 0 veya 1 olan *n*-bit vektör

$wt()$: Hamming ağırlığı

$Had()$: Hadamard matris

$circ()$: Dairesel matris

$\beta()$: Dallanma sayısı (Branch number)

LAT: Doğrusal Yaklaşım Tablosu (Linear Approximation Table)

DDT: Fark Dağılım Tablosu (Difference Distribution Table)

NLM: Doğrusal Olmama Ölçüsü (Nonlinear Measurement)

DP: Diferansiyel Olasılık

LP: Doğrusal Olasılık

ŞEKİLLER LİSTESİ

Şekil 1.1 Şifreleme ve Şifre çözme İşlemleri..... 2
Şekil 1.2 Blok Şifreleme ve Şifre Çözme..... 6
Şekil 2.1 Test için Geliştirilen Yazılımın Arayüzü..... 24
Şekil 3.1 AES Blok Şifresindeki Tek Döngülük SPN Mimarisi..... 27
Şekil 3.2 AES Şifreleme Doğrusal Dönüşüm İşlemi..... 29
Şekil 3.3 KHAZAD Blok Şifresindeki Tek Döngülük SPN Mimarisi..... 30
Şekil 3.4 CAMELLIA Şifreleme Algoritması..... 32
Şekil 3.5 SERPENT Blok Şifresindeki Tek Döngülük SPN Mimarisi..... 36
Şekil 3.6 SERPENT Blok Şifresindeki Doğrusal Dönüşüm Katmanı..... 37
Şekil 3.7 PRESENT Blok Şifresinin Blok Diyagramı..... 39
Şekil 3.8 PRESENT İçerisindeki PLayer Dönüşümü..... 40
Şekil 3.9 PRESENT İçerisindeki PLayer Dönüşümü için Örnek..... 41
Şekil 4.1 2×2 Boyutundaki Matrisin 8×8 Boyutundaki Matrislere Dönüştürülmesi..... 49
Şekil 4.2 4×4 Boyutundaki Matrisin 16×16 Boyutundaki Matrislere Dönüştürülmesi..... 52
Şekil 8.1 ARIA Şifreleme Algoritmasının Blok Diyagramı 95
Şekil 8.2 ARIA'da Kullanılan Yer Değiştirme Katmanları..... 98
Şekil 8.3 Programın Arayüzü..... 102
Şekil 8.4 Dosya Şifreleme..... 102
Şekil 9.1 TRAKYA Algoritmasının Tek Döngülük SPN Mimarisi..... 106
Şekil 9.2 Geliştirilen Şifrenin Şifreleme Blok Diyagramı..... 108
Şekil 9.3 Geliştirilen Şifrenin Şifre Çözme Blok Diyagramı..... 109
Şekil 9.4 S-kutusu Tasarımı..... 111
Şekil 9.5 Yer Değiştirme İşlemi..... 114
Şekil 9.6 Doğrusal Dönüşüm İşlemi..... 116
Şekil 9.7 Şifrenin Anahtar Planlama Evresi..... 118
Şekil 9.8 Şifreleme Yazılımının Arayüzü..... 126

Şekil 9.9 Dosya Şifreleme Yazılımının Arayüzü..... 126
Şekil 9.10 Dosya Şifreleme İşlemi..... 126

TABLOLAR LİSTESİ

Tablo 1.1 Sezar Şifreleme Algoritması için Örnek Tablo..... 3
Tablo 1.2 Vigenere Tablosu..... 4
Tablo 1.3 Üç Gruba göre Şifreleme Algoritmalarına Örnekler..... 5
Tablo 1.4. Bazı Blok Şifrelerin Anahtar Uzunlukları..... 7
Tablo 2.1: [2*n*, *n*] Doğrusal Kodların Maksimum Uzaklığının Alt ve Üst Sınırları 22
Tablo 3.1: AES ShiftRows Dönüşümü..... 28
Tablo 3.2: PRESENT PLayer Dönüşüm Tablosu..... 40
Tablo 3.3: İncelenen Blok Şifrelerin Özellikleri..... 41
Tablo 4.1: İndirgenemez Polinom x^4+x+1 ile Tanımlı $GF(2^4)$ Cisminin Elemanlarının x Sonlu Cisim Elemanı ile Çarpımı Sonucu Üretilen 4×4 Büyüklüğünde İkili Doğrusal Dönüşümler 45
Tablo 4.2: İndirgenemez Polinom x^4+x^3+1 ile Tanımlı $GF(2^4)$ Cisminin Elemanlarının x Sonlu Cisim Elemanı ile Çarpımı Sonucu Üretilen 4×4 Büyüklüğünde İkili Doğrusal Dönüşümler..... 46
Tablo 4.3: İndirgenemez Polinom $x^4+x^3+x^2+x+1$ ile Tanımlı $GF(2^4)$ Cisminin Elemanlarının x Sonlu Cisim Elemanı ile Çarpımı Sonucu Üretilen 4×4 Büyüklüğünde İkili Doğrusal Dönüşümler..... 47
Tablo 5.1: x^4+x+1 İndirgenemez Polinomuna göre 2×2 Boyutundaki İnvolutif Matrisler..... 69
Tablo 5.2: x^4+x+1 İndirgenemez Polinomuna göre 2×2 Boyutundaki İnvolutif Olmayan Matrisler..... 70
Tablo 5.3: x^4+x^3+1 İndirgenemez Polinomuna göre 2×2 Boyutundaki İnvolutif Matrisler..... 70
Tablo 5.4 : x^4+x^3+1 indirgenemez polinomuna göre 2×2 boyutundaki involutif olmayan matrisler..... 71
Tablo 6.1 : x^4+x+1 İndirgenemez Polinomu Kullanılarak Elde Edilen 76

Dallanma Sayısı 8 ve *(A-I)* Matris Rank Değeri 8 Olan İnvolutif İkili Doğrusal Dönüşümlerin Sınıflandırılması.....

Tablo 6.2 : x^4+x+1 İndirgenemez Polinomuna göre Dallanma Sayısı 8 Olan İnvolutif Olmayan İkili Matrisler (Dairesel Matrisler) 77

Tablo 6.3: x^4+x^3+1 İndirgenemez Polinomu Kullanılarak Elde Edilen Dallanma Sayısı 8 ve *(A-I)* Matris Rank Değeri 8 Olan İnvolutif İkili Doğrusal Dönüşümlerin Sınıflandırılması..... 78

Tablo 6.4: x^4+x^3+1 İndirgenemez Polinomuna Göre Dallanma Sayısı 8 Olan İnvolutif Olmayan İkili Matrisler (Dairesel Matrisler) 78

Tablo 6.5: $x^4+x^3+x^2+x+1$ İndirgenemez Polinomu Kullanılarak Elde Edilen Dallanma Sayısı 8 ve *(A-I)* Matris Rank Değeri 8 Olan İnvolutif İkili Doğrusal Dönüşümlerin Sınıflandırılması..... 80

Tablo 6.6: $x^4+x^3+x^2+x+1$ İndirgenemez Polinomuna Göre Dallanma Sayısı 8 Olan İnvolutif Olmayan İkili Matrisler (Dairesel Matrisler) 80

Tablo 7.1: Üretilen 1 Sabit Noktalı Dallanma Sayısı 7 Olan 16×16 Boyutundaki Matris Sayıları..... 89

Tablo 7.2: 1 Sabit Noktalı Dallanma Sayısı 7 Olan 16×16 Boyutundaki Matrise Dönüştürülebilecek x^4+x+1 İndirgenemez Polinomu ile Tanımlı Olan 4×4 Boyutundaki Matris Sınıfları..... 90

Tablo 7.3: had(1_h;2_h;4_h;E_h) Sınıfının Elemanları..... 91

Tablo 7.4: 1 Sabit Noktalı Dallanma Sayısı 7 Olan 16×16 Boyutundaki Matrise Dönüştürülebilecek x^4+x^3+1 İndirgenemez Polinomu ile Tanımlı Olan 4×4 Boyutundaki Matris Sınıfları..... 92

Tablo 8.1: S-kutusu S_1..... 96

Tablo 8.2: S-kutusu S_1^{-1}..... 96

Tablo 8.3: S-kutusu S_2..... 97

Tablo 8.4: S-kutusu S_2^{-1}..... 97

Tablo 9.1: S-kutusu..... 110

Tablo 9.2: Şifrede Kullanılan S-kutusunun "01" Giriş Farkı için DDT Dağılımı.... 113

Tablo 9.3: S-kutusunun "01" Giriş Maskesi için LAT Dağılımı..... 113

BÖLÜM 1

GİRİŞ

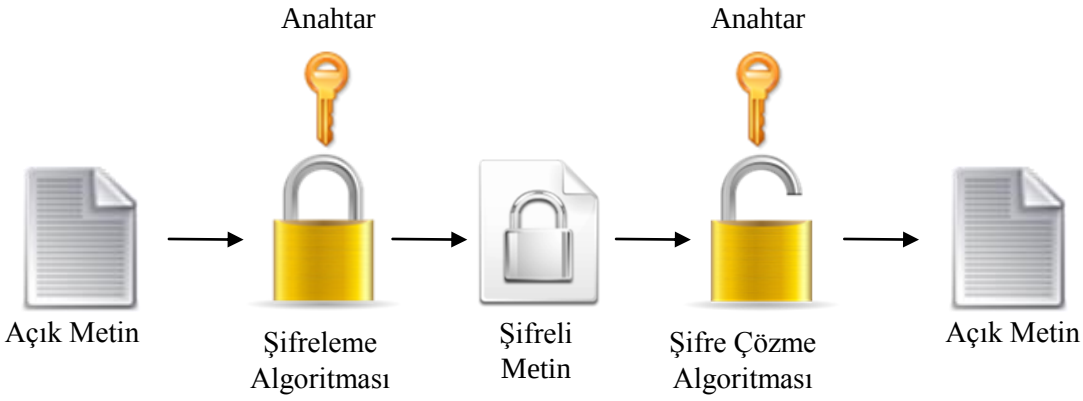
İnsanoğlu var olduğundan beri haberleşme için yeni yöntemler geliştirmiş ve iletilerini başka insanlardan saklama gerekliliği ortaya çıktığından beri de gizlilik, haberleşmedeki en önemli kıstas olmuştur. Bilgileri kodlama, binlerce yıl önce devletlerin ve imparatorlukların gizli ve önemli bilgileri üçüncü kişilerin eline geçmeden iletebilmesi için ortaya çıkmıştır. İleti ne şekilde gönderilirse gönderilsin her zaman başkalarının eline geçme ihtimali vardır. Bu sebeple iletiyi kodlayarak göndermek, mesajın başkaları tarafından anlaşılmasını zorlaştırmaktadır.

1.1. Kriptografi

Günümüzde bilgi güvenliği, iletişim teknolojilerinin gelişimi ile birlikte oldukça önemli hale gelmiştir. Kriptografi, verinin güvenli bir şekilde iletilmesi ile ilgilidir. Dolayısıyla güvenli şifreleme algoritması tasarımı kriptografide çok önemli bir yer tutar.

Yunanca gizli anlamına gelen “kryptos” ve yazmak anlamına gelen “graphien” kelimelerinden türetilen kriptografi anlaşılır bir mesajı anlaşılmaz hale dönüştürme ve tekrar anlaşılmaz mesajı anlaşılır hale geri dönüştürme işlemlerini kapsayan bir bilimdir. Bir başka deyişle kriptografi, anlaşılabilecek durumdaki bilginin, istenmeyen taraflarca anlaşılamayacak bir hale dönüştürülmesinde kullanılan tekniklerin tümüdür.

Bir kriptosistem; şifreleme algoritması, açık metin, şifreli metin ve anahtardan oluşmaktadır. Açık metin, bir anahtar kullanılarak şifreleme algoritması aracılığı ile şifreli metne çevrilir. Bu işleme şifreleme (encryption) denir. Şifre çözme (decryption) aşamasında ise yine bir anahtar kullanılarak şifreleme algoritmasının tersi olan bir algoritma ile açık metin elde edilir. Şekil 1.1 şifreleme ve şifre çözme işlemlerini göstermektedir.



Şekil 1.1: Şifreleme ve Şifre Çözme İşlemleri [1]

Kriptografi ile ilgilenen bilim adamlarına kriptograf adı verilir. Kriptanaliz ise ele geçirilen şifreli metinleri bazı teknikler kullanılarak açık metinleri elde etme işlemidir. Kriptanaliz ile ilgilenen kişilere kriptanalist denir. İyi bir kriptanalist aynı zamanda kriptografi alanında da bilgi sahibi olmalıdır çünkü bir şifrenin kırılabilmesi için şifre tasarımı hakkında yeterli bilgiye ihtiyaç vardır. Tarih boyunca kriptograflar ile kriptanalistler arasında çekişmeli bir yarış olduğunu söylemek yanlış olmayacaktır. Kriptanalistlerin başarıları kriptografları daha güçlü şifreler tasarlamaya zorlamıştır. Bu iki alanın birleşmesi ile kriptoloji bilimi ortaya çıkmaktadır.

1.2. Kriptografi Tarihi

Yazının icadıyla birlikte bilgi toplanabilir, saklanabilir ve iletilebilir hale gelmiştir. Her ne kadar insanlar ilk başlarda yazdıkları yazıları gizleme ihtiyacı duymamış olsa da, yazma becerisinin gelişmesiyle bu ihtiyaç ön plana çıkmıştır. Kripto tarihçisi David Khan'a göre ilk kriptografik belge yaklaşık olarak M.Ö. 1900 yılında yazıldığı tahmin edilen hiyerogliftir. Bu yazıtta bir lordun hayatından bahsedilmektedir. Aynı şekilde M.Ö. 1500 yıllarına ait olan bir Mezopotamya tabletinde çömleklerin cilalanması hakkındaki bilgiler şifrelenmiş olarak verilmiştir [2].

M.Ö. 457 yılında Anadolu'da Spartalılar tarafından Skytale adındaki tarihteki ilk şifreleme cihazının kullanıldığı görülmektedir. Uzun bir şerit şeklindeki parşömen kâğıdı, belirli bir şekildeki veya boyuttaki silindire sarılır ve gizlenecek olan metin silindirin bölümüne yazılır. Şerit üzerindeki diğer bölümler karışık harfler ile doldurulur. Şerit silindirden çıkarıldığında karışık harflerin olduğu bir kâğıt halini alır.

Mesajın deşifrelenebilmesi için parşömenin aynı boyutlarda olan bir başka silindire sarılması gerekmektedir.

Tarihteki önemli şifreleme algoritmalarından bir tanesi M.Ö. 60-50 yılları arasında Roma lideri Julius Ceasar tarafında kullanılan ve adına Sezar şifresi denilen bir şifreleme sistemidir. Bu şifreleme sisteminde; her harf, alfabede kendisinden sonra gelen üçüncü harfle yer değiştirilerek şifreli metin elde edilmektedir. Türkçe alfabe düşünüldüğünde, şifreleme işlemi için aşağıdaki tablo kullanılabilir. Örneğin "SEN DE Mİ BRUTUS" açık metni şifrelenerek "UĞP GĞ ÖL DTYVYU" şifreli metnine ulaşılabilir. Sezar şifresinde şifreleme algoritmasının kaydırma tekniği, anahtarının da üç harf kaydırma olarak düşünülebilir.

Tablo 1.1 Sezar Şifreleme Algoritması için Örnek Tablo

Açık Metin	A	B	C	Ç	.	.	.	V	Y	Z
Şifreli Metin	Ç	D	E	F	.	.	.	A	B	C

Sezar şifresine benzer birçok kaydırma şifresi çeşitli anahtar şekilleri ile tarih boyunca kullanılmıştır. Fakat bu şifrelerin güvenliği çok azdır. Alfabenin harf sayısı kadar kaydırma işlemi denemesi sonucunda şifre sistemi rahatlıkla kırılabilir.

1404-1472 yıllarında yaşamış olan Leon Alberti, sezar şifreleme yönteminden esinlenmiştir. Alberti, kaydırma miktarını sabit olarak değil değişken olarak belirlemiştir. Şifrelenecek olan metindeki her harfin şifreli karşılığı Alberti Diski yardımıyla belirlenerek şifreli metine ulaşılıyordu.

Öte yandan Blaise de Vigenere, 1586 yılında yine Sezar algoritmasının türevi olan "Vigenere Sistemi" ni geliştirdi. Şifreleme yapılacak olan metindeki her harf, tek bir alfabeye göre değil farklı alfabelere göre şifreleniyordu. Bu işlem oldukça karışık bir iş yükü getirdiğinden dolayı Tablo1.2 deki Vigenere Tablosu kullanılıyordu.

Tablo 1.2 Vigenere Tablosu

	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z
A	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z
B	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A
C	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B
C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C
D	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç
E	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D
F	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E
G	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F
Ğ	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G
H	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ
I	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H
İ	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I
J	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ
K	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J
L	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K
M	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L
N	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M
O	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N
Ö	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O
P	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö
R	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P
S	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R
S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S
T	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş
U	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T
Ü	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U
V	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü
Y	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V
Z	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y

Örnek olarak Vigenere şifreleme algoritması kullanılarak “TRAKYA ÜNİVERSİTESİ” açık metni “EDİRNE” anahtar kelimesi ile aşağıdaki gibi şifrelenebilir.

T	R	A	K	Y	A	Ü	N	İ	V	E	R	S	İ	T	E	S	İ
E	D	İ	R	N	E	E	D	İ	R	N	E	E	D	İ	R	N	E
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Z	U	İ	D	L	E	B	R	Ş	O	S	Ü	V	M	E	Ü	Ğ	N

Vigenere şifresi 1800’lü yılların ortasına kadar kırılması imkânsız olarak nitelendirildi. 1854’te Charles Babbage anahtar uzunluğuna dayalı frekans testi ile şifreyi kırdı ve “Gizli Yazma ve Şifre Çözme Sanatı” adlı kitabında yayınladı. 1864’te Kasiski, Babbage’dan bağımsız olarak aynı saldırı tekniğini geliştirdi. Amerikan iç savaşında Vigenere şifresinin kullanıldığı bilinmektedir.

İkinci dünya savaşı yaklaşırken ülkeler bilgi güvenliği ve gizliliği kavramlarına ağırlık vermişlerdir. İkinci dünya savaşının önemli bir bölümünün kriptograflar ile kriptanalistler arasında geçtiğini söylemek pek de yanlış olmayacaktır. Savaşın Almanların aleyhine dönüşünde İngiliz, Polonyalı ve Amerikalı kriptanalistlerin büyük çabalar ile kırdığı Enigma mesajlarının rolü büyüktür [3].

1.3. Modern Kriptografi

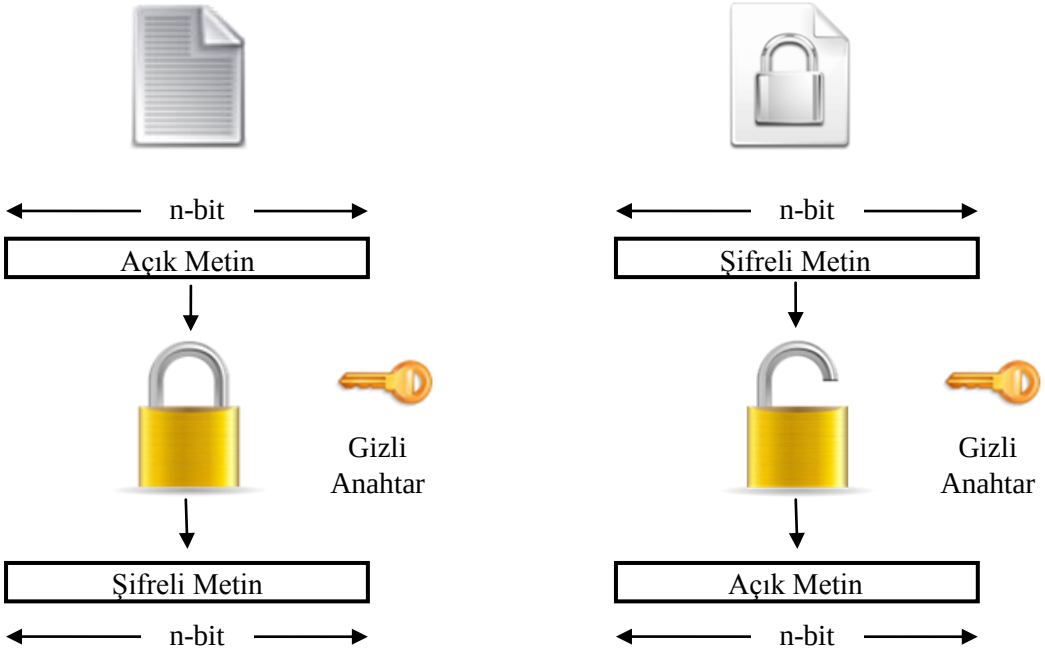
Günümüzde, şifreleme algoritmaları simetrik, asimetrik ve hash (özet) algoritmaları olarak üç gruba ayrılabilir. Bu şifreleme algoritmalarından simetrik algoritmalar şifreleme ve şifre çözme işlemlerinde aynı anahtarı, gizli anahtarı, kullanır. Asimetrik şifreleme algoritmaları ise şifreleme için herkesin ulaşabileceği açık bir anahtar kullanılırken şifre çözme işlemi için gizli bir anahtar kullanır. Son olarak Hash algoritmalar ise verinin bir temsilini oluşturmak için kullanılırlar ve kimlik denetiminin sağlanmasında büyük rol oynarlar. Bu üç gruba göre bazı şifreleme algoritmaları Tablo 1.3'de gösterilmiştir. Diğer yandan kriptografide blok şifreleme ve akan (stream) şifreleme olmak üzere iki temel simetrik algoritma tipi vardır. Bunlardan blok şifreleme algoritmaları orijinal metni veya şifreli metni bloklara bölerek şifreleme/şifre çözme işlemini yaparken akan şifreleme algoritmaları bir bit veya byte anahtar üreterek şifreleme ve şifre çözme işlemlerini gerçekleştirirler.

Tablo 1.3 Üç Gruba göre Şifreleme Algoritmalarına Örnekler

Simetrik Şifreleme Algoritmaları		Asimetrik Şifreleme Algoritmaları	Hash Algoritmaları
Blok Şifreler	Akan Şifreler		
-DES [4] -Present [5] -Serpent [6] -AES [7] -Camellia[8] -ARIA[9] -Khazad[10]	-RC4 [11] -Trivium[12] - HC-256 [13]	- RSA [11] - ElGamal [11] - ECC [14]	- MD4 [11] - MD5 [11] - SHA [11] -RIPEMD-160 [15]

1.4. Blok Şifreler

Blok şifreleme algoritmaları açık metni sabit uzunluklu blok adı verilen bit grupları halinde işler. Bloklar bir anahtar aracılığı ile şifrelenerek şifreli metin ortaya çıkar. Şifre çözme işleminde yine aynı anahtar sayesinde şifreli metin açık metin haline getirilir.



(a) Şifreleme (b) Şifre Çözme

Şekil 1.2: Blok Şifreleme ve Şifre Çözme

Blok şifreler, Shannon'un [16] önerdiği karıştırma (confusion) ve yayılma (diffusion) teknikleri üzerine kuruludur. Karıştırma şifreli metin ve açık metin arasındaki ilişkiyi gizlemeyi amaçlarken, yayılma açık metindeki izlerin şifreli metinde sezilmemesini sağlamak için kullanılır. Karıştırma yer değiştirme işlemi ile gerçekleştirilirken yayılma ise doğrusal dönüşüm işlemleri ile gerçeklenir.

Blok şifresi tasarımı için en çok tercih edilen iki mimari, Feistel ağları ve Yer değiştirme-Permütasyon ağlarıdır (SPN) [17]. Örneğin DES algoritması Feistel ağları tabanlı iken, AES (Advanced Encryption Standard) algoritması SPN (Substitution-permutation network) mimarisi tabanlı olarak tasarlanmıştır. Her iki mimari de yer değiştirme ve doğrusal dönüşüm yapılarını kullanır. Ayrıca her iki mimari ürün

şifrelerinin örneklerindendir. Yani birden fazla şifreleme işleminin birleşmesi ile oluşturulurlar. Tekrarlanan şifreler yine ürün şifreleridir ve aynı şifreleme adımının tekrarlanan uygulamasını içerir ve her şifreleme adımına döngü denir. Bir döngü birden fazla şifreleme adımı içerebilir. Genellikle her döngüde farklı anahtar materyali kullanılmaya özen gösterilir. Diğer yandan bu mimarilerin arasındaki en temel fark döngü içerisinde bir bloğun işlenmesinde ortaya çıkmaktadır. Örneğin Feistel mimarisinde bir döngüde o anki bloğun yarısı işlenirken SPN mimarisinde o anki bloğun tümü işlenir. Buna ek olarak bir blok şifrenin genel tasarımında bir döngü içindeki yer değiştirme, S-kutuları (S-Boxes) yardımıyla yayılım ise doğrusal dönüşüm veya dönüşümler ile sağlanır. Her döngüde, döngünün sonunda anahtar planlamadan gelen o döngü için elde edilen bir anahtar değeri ile XOR'lama işlemi gerçekleştirilir.

Blok şifrelerin gücünü belirleyen bazı faktörler aşağıdaki gibidir. Bunlar anahtar büyüklüğü, S-kutuları ve doğrusal dönüşümlerdir.

1.4.1. Anahtar Büyüklüğü

Blok şifrelerde anahtarın uzunluğu saldırılara karşı güçlü olacak şekilde seçilmelidir. Bunun için anahtar uzunluğu önemlidir. Bunun sayesinde şifrenin kaba kuvvet (brute-force) saldırısına karşı kırılabilirliği zorlaşmaktadır. Bazı blok şifrelerin kullandıkları anahtar uzunlukları Tablo 1.4'te verilmiştir. Örneğin anahtar uzunluğu kısa olan DES algoritması kırılarak günümüzde güvenirliliğini yitirmiştir.

Tablo 1.4. Bazı Blok Şifrelerin Anahtar Uzunlukları

Blok Şifreler	Anahtar Uzunluğu
DES	56-bit
IDEA	128-bit
AES	128, 192, 256-bit
Camellia	128, 192, 256-bit
ARIA	128-bit
Khazad	128-bit
Present	80 veya 128-bit
Serpent	128, 192, 256-bit

1.4.2. S-Kutuları (Substitution-Boxes)

Blok şifreleme algoritmalarının en önemli elemanı S-kutularıdır ve karıştırma işlevini üstlenirler. Algoritmanın tek doğrusal olmayan elemanıdır. Bu yüzden iyi bir S-kutusu seçimi şifrenin karmaşıklığını doğrudan etkiler.

1.4.3. Doğrusal Dönüşümler

Blok şifrelerin önemli bir özelliği olan yayılma işlemini sağlayan yapılar doğrusal dönüşümlerdir. Doğrusal dönüşümler sabit uzunluktaki bir giriş bloğunu doğrusal olarak karıştırarak aynı uzunlukta bir çıkış bloğu elde etmeyi sağlar.

1.5. Kriptanaliz

Kriptanaliz açık metni veya anahtarı elde etme bilimidir. Kısacası şifre kırma bilimidir. Kriptoloji içerisinde oldukça önemli bir yere sahiptir. Ortaya sürülen bir şifreleme sisteminin zayıf ve güçlü yönlerini ortaya çıkarmak için kullanılabileceği gibi kötü niyetli olarak bir şifrenin kırılıp açık metne ulaşmak içinde kullanılabilir. Bilinen kayıtlı ilk kriptanaliz açıklaması 9.yy da Matematikçi olan Ebu Yusuf Yakup tarafından yazılan "A Manuscript on Deciphering Cryptographic Messages" eserinde yer almıştır. Bu yöntem frekans analizine dayanmaktadır [18]. İkinci dünya savaşı ile beraber kriptanalizin önemi daha çok ortaya çıkmıştır. Bu şekilde Almanların Enigma şifresi kırılmış ve savaşın yönü büyük oranda değişmiştir.

Kriptanalizde düşmanın saldırı yapılan kripto sistemi bildiği kabul edilir (Kerckhoffs'un prensibi) ve bu koşul altında düşman kripto sistemin en önemli öğesi olan şifreleme algoritmasına saldırı gerçekleştirir. Düşmanın bir kripto sisteme saldırabilmesi için sahip olması gereken veriler vardır. Bu sahip olduğu verilere göre saldırı modellerinden birini seçebilir. Saldırı modelleri aşağıdaki gibi sınıflandırılabilir [19]:

- Sadece şifreli metin saldırısı: Düşman şifreli metin dizisine sahiptir,
- Bilinen açık metin saldırısı: Düşman açık metin dizisine ve bunların şifreli metin dizisine sahiptir,

- Seçilmiş açık metin saldırısı: Düşman bir açık metin dizisini seçebilir ve bunların şifreli metinlerini oluşturabilir,
- Seçilmiş şifreli metin saldırısı: Düşman bir şifreli metin dizisi seçebilir ve bunların açık metinlerini oluşturabilir.

Blok şifrelerin kırılması, zaaflarının ve güçlü yanlarının anlaşılması için birden fazla kriptanaliz yöntemi kullanılabilir. Bunlardan en önemlileri doğrusal ve diferansiyel kriptanalizdir. Bu saldırılar, şifrenin doğrusal olmayan tek yapısı olan S-kutularını hedef alır. Örneğin AES şifresi doğrusal ve diferansiyel saldırılara karşı dayanıklı olması amacıyla geliştirilmiş bir şifredir.

1.5.1. Doğrusal Kriptanaliz

1993 yılında Matsui [20] tarafından teorik bir saldırı olarak keşfedilmiştir. Daha sonra DES algoritmasına karşı başarı ile uygulanmıştır. Modern sifreleme algoritmalarının tasarımında dikkate alınması gereken önemli bir unsurdur.

Doğrusal kriptanaliz, S-kutularının doğrusal ifadelere dönüştürülmesi ve doğrusal ifadeleri birleştirerek bilinmeyen anahtar bitlerini elde etme prensibine dayanır. Doğrusal kriptanaliz, şifreli metin bitleri ile açık metin bitleri arasındaki yüksek olasılıkta doğrusal ifadelerin meydana gelme avantajını kullanır. Bunun yolu da S-kutularından geçer. Saldırganın algoritmayı bildiği (Kerchoffs kuralı) ve belli sayıda açık metin ve şifreli metinlere sahip olduğu varsayılır. S-kutularının büyüklüğü, aktif S-kutularının (doğrusal ifade içinde olan) sayısının artışı ve doğrusal sapması küçük S-kutularının tasarımı doğrusal kriptanalizin uygulanmasını engelleyici faktörlerdir. Bu işlem için son döngüde yerine getirilen yer değiştirmelerden önceki durum bitleri ile açık metin bitleri arasında doğrusal bir ilişki bulunması gereklidir. Bu doğrusal ifade olası tüm anahtar bitleri ile test edilir ve anahtar bitlerinin sapması teorik olarak elde edilen sapma değeri ile karşılaştırılır. En yüksek sapma ($\frac{1}{2}$ den uzaklaşma) değerine sahip anahtar aranılan hedef anahtardır. Eğer hedef anahtar yanlış ise sapma 0 değerine yakın olacaktır.

1.5.2. Diferansiyel Kriptanaliz

1991 yılında Biham [21] tarafından keşfedilmiş bir kriptanaliz yöntemidir. Doğrusal kriptanalize benzemekle beraber seçilmiş açık metin saldırısı modeline dayanmaktadır. Yani açık metin çiftlerindeki özel farkların sonuçlanan şifreli metinlerde oluşturduğu farkın etkisini analiz eder. Bu farklar mümkün olan anahtarların olasılıklarını ve en yüksek mümkün anahtarı ortaya koymak için tayin edilir. Kısacası bu saldırıda birçok sayıda açık metin ve şifreli metin çiftlerini üretilir. Bu çiftler arasındaki özel farklara karşılık şifreleme algoritmasının son döngüsündeki S-kutusundan önceki durum bitleri farkı bulunur. Bu farka göre her açık ve şifreli metin çiftleri için olası anahtar değerleri denenir ve eğer uygun bir değer yakalanır ise sayaç değeri bulunan anahtar değeri için 1 arttırılır. Yüksek olasılığı yakalayan anahtar değeri aranan hedef anahtar olarak kabul edilir.

İki saldırı yönteminde de S-kutularının ve doğrusal dönüşümlerin önemi açıktır. Eğer şifreleme algoritması için kriptografik özellikler açısından iyi S-kutuları ve doğrusal dönüşümler seçilirse hem doğrusal hem de diferansiyel saldırılara karşı daha dayanıklı ve güvenli şifreleme algoritmaları tasarlanabilir.

Bu iki kriptanaliz yöntemlerinin dışında imkânsız diferansiyel kriptanaliz [22], kesilmiş diferansiyel kriptanaliz [23], çokluset saldırıları [24], interpolasyon saldırısı [25] gibi cebirsel saldırılar, boomerang saldırısı [26], kare (rectangular) saldırısı [27], yan kanal saldırıları [28] gibi çeşitli saldırı teknikleri mevcuttur.

1.6. Tezin Önemi

Blok şifreler, karıştırma (confusion) ve yayılım (diffusion) tekniklerine dayanır. Karıştırma şifreli metin ve açık metin arasındaki ilişkiyi gizlemeyi amaçlarken, yayılım açık metindeki izlerin şifreli metinde sezilmemesini sağlamak için kullanılır. Karıştırma ve yayılım, sırasıyla S-kutuları ve doğrusal dönüşüm işlemleri ile gerçekleştirilir.

Yayılım elemanları olan doğrusal dönüşümler şifre tasarımında kullanılan önemli elemanlardır. Bu tez çalışması blok şifrelerde kullanılan doğrusal dönüşümler üzerinedir. Tez çalışmasında mevcut önemli blok şifrelerde kullanılan doğrusal dönüşümler incelenmiş ve kriptografik özellikleri iyi involutif veya involutif olmayan

8×8 ve 16×16 boyutunda doğrusal dönüşümlerin elde edilebilmesi için bir cebirsel yöntem geliştirilmiştir. Bu yöntem ile elde edilen ve kriptografik özellikleri iyi olan olası tüm ikili doğrusal dönüşümlerin listelenmesi amaçlanmaktadır. Buna ek olarak bu elde edilen ikili doğrusal dönüşümlerden birinin modern bir şifreleme algoritmasına uygulanması da tezde gerçekleştirilmektedir. Bunun yanında elde edilecek doğrusal dönüşümlerin önemli saldırı tekniklerine karşı dayanıklılığının incelenmesi de amaçlanmaktadır. Bu tasarlanacak doğrusal dönüşümlerin ikili elemanlardan oluşacak doğrusal dönüşümler olarak seçilmesinin amacı geliştirilecek şifreleme algoritmasının 8-bit, 32-bit ve 64-bit işlemciler için sadece XOR işlemleri tabanlı olmasıdır. Dolayısıyla yazılım performansı iyi ve güçlü bir şifreleme algoritması tasarımı gerçekleştirilecektir.

BÖLÜM 2

MATEMATİK ALT YAPI

Bu bölümde, tez çalışması boyunca kullanılacak olan matematiksel ifadelerin anlaşılması açısından temel bir matematik alt yapı verilecektir. Verilen tanım ve teorilerin ispatlarına [29, 30, 31]'den ulaşılabilir.

Tanım 2.1. Bir sonlu cisim, sıfır değerinden farklı elemanlarının çarpmaya göre tersi olduğu değişmeli bir halkadır. Sonlu cisim $GF(2^m)$, $GF(2)$ sonlu cisminin genişletilmiş bir cismidir ve *m* pozitif bir tamsayı olmak üzere, 2^m elemana sahiptir. Sonlu cisim $GF(2^m)$'in 2^m elemanının her biri katsayıları $GF(2)$ cisminden olan (0 ya da 1) ve derecesi *m-1* olan bir polinom şeklinde ifade edilebilir. Örneğin, *x*, $GF(2^m)$ cisminin bir elemanı ise o zaman *x*'in polinom tabanlı gösterimi (2.1) ifadesindeki gibi verilebilir.

$$x_{m-1}\alpha^{m-1} + x_{m-2}\alpha^{m-2} + ... + x_1\alpha + x_0 \qquad (2.1)$$

(2.1) ifadesinde α, $GF(2^m)$ cismini oluşturmak için kullanılan ilkel elemanı temsil etmektedir. $GF(2^m)$ cisminde iki elemanın toplama sonucu basitçe XOR işlemi ile elde edilirken çarpma sonucu, iki polinomun çarpımının bir indirgenemez polinom ile indirgenmesi sonucu elde edilir. Örneğin $GF(2^4)$, $GF(2)$ cisminin 4. dereceden genişletilmiş cismi olarak isimlendirilir ve $GF(2)$ cismine, $GF(2^4)$ cisminin taban cismi adı verilir.

Tanım 2.2. Z_m bir cisim olmak üzere set $Z_m[x] := \left\{ \sum_{i=0}^{n} a_i x^i : a_i \in Z_m, n \geq 0 \right\}$, Z_m üzerine bir polinom halka olarak isimlendirilir. $Z_m[x]$'in bir elemanı Z_m üzerine polinom olarak isimlendirilir. Pozitif dereceli bir polinom $f(x) = \sum_{i=0}^{n} a_i x^i$ için

derece$(g(x)) <$ derece$(f(x))$, derece$(h(x)) <$ derece$(f(x))$ ve $f(x) = g(x)h(x)$ şartlarını sağlayacak şekilde iki polinom varsa *f(x)* polinomu Z_m üzerine indirgenebilir aksi takdirde pozitif dereceli *f(x)* polinomu Z_m üzerine indirgenemez polinom olarak tanımlanabilir.

$GF(2^4)$ cisminin tanımlanması için kullanılabilecek 3 indirgenemez polinom $x^4 + x + 1$, $x^4 + x^3 + 1$ ve $x^4 + x^3 + x^2 + x + 1$ şeklindedir. Verilen indirgenemez polinomlar ile tanımlanan $GF(2^4)$ cismi tezde kullanıldığı için bu cisimlerin elemanları aşağıda verilmiştir:

$x^4 + x + 1$ ile tanımlanan $GF(2^4)$ cisminin üreteç elemanı α olmak üzere, cismin elemanlarının ikili ve hexadecimal gösterimleri aşağıda verilmiştir:

α^0	= 1	0001	1_h
α^1	= α^1	0010	2_h
α^2	= α^2	0100	4_h
α^3	= α^3	1000	8_h
α^4	= $\alpha + 1$	0011	3_h
α^5	= $\alpha^2 + \alpha$	0110	6_h
α^6	= $\alpha^3 + \alpha^2$	1100	C_h
α^7	= $\alpha^3 + \alpha + 1$	1011	B_h
α^8	= $\alpha^2 + 1$	0101	5_h
α^9	= $\alpha^3 + \alpha$	1010	A_h
α^{10}	= $\alpha^2 + \alpha + 1$	0111	7_h
α^{11}	= $\alpha^3 + \alpha^2 + \alpha$	1110	E_h
α^{12}	= $\alpha^3 + \alpha^2 + \alpha + 1$	1111	F_h
α^{13}	= $\alpha^3 + \alpha^2 + 1$	1101	D_h
α^{14}	= $\alpha^3 + 1$	1001	9_h
α^{15}	= 1		

$x^4 + x^3 + 1$ ile tanımlanan $GF(2^4)$ cisminin üreteç elemanı α olmak üzere, cismin elemanlarının ikili ve hexadecimal gösterimleri aşağıda verilmiştir:

α^0	=	1	0001	1_h
α^1	=	α^1	0010	2_h
α^2	=	α^2	0100	4_h
α^3	=	α^3	1000	8_h
α^4	=	$\alpha^3 + 1$	1001	9_h
α^5	=	$\alpha^3 + \alpha + 1$	1011	B_h
α^6	=	$\alpha^3 + \alpha^2 + \alpha + 1$	1111	F_h
α^7	=	$\alpha^2 + \alpha + 1$	0111	7_h
α^8	=	$\alpha^3 + \alpha^2 + \alpha$	1110	E_h
α^9	=	$\alpha^2 + 1$	0101	5_h
α^{10}	=	$\alpha^3 + \alpha$	1010	A_h
α^{11}	=	$\alpha^3 + \alpha^2 + 1$	1101	D_h
α^{12}	=	$\alpha + 1$	0011	3_h
α^{13}	=	$\alpha^2 + \alpha$	0110	6_h
α^{14}	=	$\alpha^3 + \alpha^2$	1100	C_h
α^{15}	=	1		

$x^4 + x^3 + x^2 + x + 1$ ile tanımlanan $GF(2^4)$ cisminin üreteç elemanı $\alpha + 1$ olmak üzere, cismin elemanlarının ikili ve hexadecimal gösterimleri aşağıda verilmiştir:

α^0	=	1	0001	1_h
α^1	=	$\alpha + 1$	0011	3_h
α^2	=	$\alpha^2 + 1$	0101	5_h
α^3	=	$\alpha^3 + \alpha^2 + \alpha + 1$	1111	F_h
α^4	=	$\alpha^3 + \alpha^2 + \alpha$	1110	E_h

α^5	=	$\alpha^3+\alpha^2+1$	1101	D_h
α^6	=	α^3	1000	8_h
α^7	=	$\alpha^2+\alpha+1$	0111	7_h
α^8	=	α^3+1	1001	9_h
α^9	=	α^2	0100	4_h
α^{10}	=	$\alpha^3+\alpha^2$	1100	C_h
α^{11}	=	$\alpha^3+\alpha+1$	1011	B_h
α^{12}	=	α	0010	2_h
α^{13}	=	$\alpha^2+\alpha$	0110	6_h
α^{14}	=	$\alpha^3+\alpha$	1010	A_h
α^{15}	=	1		

Sonlu cisimde iki elemanın toplanması ya da çıkarılması işlemi, standart polinomların toplama ve çıkarma işlemi gibidir. Sonlu cisim aritmetiğinde elemanlar $\{0,1\}$ katsayılarına sahip polinomlar olarak temsil edilebildiğinden, toplama işlemi basitçe katsayılarının modulo 2 aritmetiğine göre toplamıdır. Bir başka deyişle *GF*(2)'de aynı dereceli elemanlara XOR işlemi uygulanır.

Örnek 2.1. $x^4+x^3+x^2+x+1$ ile tanımlanan $GF(2^4)$ cisminin elemanlarından olan A_h ($\alpha^3+\alpha$) ve 7_h ($\alpha^2+\alpha+1$) değerlerinin toplamı

$$A_h + 7_h = \alpha^3+\alpha+\alpha^2+\alpha+1 = \alpha^3+\alpha^2+1$$

olarak bulunur. Bu değer de cisimde D_h elemanına karşılık gelir yani $A_h + 7_h = D_h$ olur.

Sonlu cisim aritmetiğinde çarpma ise polinomların birbirleri ile aritmetik çarpımı şeklinde yapılır. Elbette ki çarpma sonucunda sonlu cismin derecesinden daha yüksek dereceli elemanlar oluşabilir. Bu durumda indirgenemez polinoma göre indirgeme ya da mod alma işlemi gerçekleştirilir.

Örnek 2.2. $x^4+x^3+x^2+x+1$ ile tanımlanan $GF(2^4)$ cisminin elemanlarından olan A_h ($\alpha^3+\alpha$) ve 7_h ($\alpha^2+\alpha+1$) değerlerinin çarpımının 5_h (α^2+1) olarak hesaplandığı aşağıdaki şekilde gösterilebilir:

$$\begin{aligned} A_h \times 7_h &= (\alpha^3+\alpha)\times(\alpha^2+\alpha+1) \\ &= \alpha^5+\alpha^4+\alpha^3+\alpha^3+\alpha^2+\alpha \\ &= \alpha^3+\alpha^2+1+\alpha^3+\alpha^2+\alpha+\alpha^2+\alpha \\ &= \alpha^2+1. \end{aligned}$$

Bölüm 1'de verildiği gibi bir blok şifrenin tasarımında kullanılan iki önemli yapı bulunmaktadır: S-kutuları ve doğrusal dönüşümler. S-kutuları *n*-bit'ten *m*-bit'e dönüşüm yapan ve doğrusal olmayan yapılardır. Tezde *n*-bit'ten *n*-bit'e dönüşüm yapan S-kutuları gözönüne alınmaktadır.

Tanım 2.3. $S: GF(2^n) \to GF(2^n)$ olmak üzere *n*-bit giriş ve *n*-bit çıkışa sahip bir S-kutusu olsun. O zaman herhangi verilen $a,b,\Gamma_a,\Gamma_b \in GF(2^n)$ için $XOR\,(a,b)$, herhangi $a!=0$ ve b için $S(x)+s(x+a)=b$ denklemindeki b değerlerinin sayısını tanımlar ve denklem (2.2) deki gibi gösterilebilir [32]. S için denklem (2.2)'de a ve b değerleri sırasıyla giriş farkı ve çıkış farkı olarak isimlendirilir.

$$XOR(a,b) = \#\left\{x \in GF(2^n) \mid S(x)+S(x+a)=b\right\} \tag{2.2}$$

Tanım 2.4. $N_L(\Gamma_a,\Gamma_b)$, herhangi $\Gamma_a \setminus 0$ ve Γ_b için $x \in GF(2^n)$ olmak üzere $\Gamma_a \bullet x = \Gamma_b \bullet S(x)$ denklemini sağlayan değerlerin sayısını tanımlar ve (2.3) denklemindeki gibi gösterilebilir [32]. S için denklem (2.4) de Γ_a ve Γ_b değerleri sırasıyla giriş maskesi ve çıkış maskesi olarak isimlendirilir. Denklem (2.4)'te herhangi bir giriş ve çıkış maskesi değerine göre LAT (Linear Approximation Table - Doğrusal Yaklaşım Tablosu) tablosu değerinin nasıl elde edileceği verilmiştir ($x \bullet y$: x ve y'nin nokta ürünü olarak isimlendirilir).

$$N_L(\Gamma_a,\Gamma_b) = \#\left\{x \in GF(2^n) \mid \Gamma_a \bullet x = \Gamma_b \bullet S(x)\right\} \tag{2.3}$$

$$LAT(\Gamma_a,\Gamma_b)=\#\left\{x\in GF(2^n)\mid \Gamma_a\bullet x=\Gamma_b\bullet S(x)\right\}-2^{n-1} \tag{2.4}$$

Tanım 2.5. $GF(q)$, q elemanlı sonlu bir cisim ve q, p^n olacak şekilde asal bir sayının üssü olsun. $f:GF(q)\to GF(q)$ olan fonksiyonları ele alalım. $a,b\in GF(q)$ olmak üzere $\nabla_f(q)$ değeri q

$$\nabla_f(q)=\text{maks}\left\{XOR\ (a,b):a,b\in GF(q),\ a\neq 0\right\}$$

değerinden az ise fonksiyon için doğrusal değildir deriz. Diğer yandan bir S-kutusu için doğrusal olmama ölçüsü NLM_S değeri LAT değeri ile ilişkili olarak ifade (2.5)'te verilmiştir.

$$NLM_S=2^{n-1}-\text{maks}\left|LAT_S(\Gamma_a,\Gamma_b)\right| \tag{2.5}$$

Yukarıda verilen tanımlara ilişkin olarak diferansiyel ve doğrusal saldırılara karşın S-kutusunun iyi davranış gösterebilmesi için S-kutusunun hem XOR hem de LAT değerlerinin maksimum değerinin olduğunca küçük olması istenen özellikler arasındadır.

Tanım 2.6. $S:GF(2^m)\to GF(2^m)$ bir S-kutusu olsun. O zaman *a* ve *b* değerleri S-kutusuna giriş ve çıkış farkları; Γ_a ve Γ_b ise S-kutusu için giriş ve çıkış maskesi olmak üzere herhangi verilen $a,b,\Gamma_a,\Gamma_b\in GF(2^n)$ değerleri için S-kutusunun diferansiyel ve doğrusal olasılığı (2.6) ifadesinde verilebilir.

$$\begin{aligned} DP_S(a,b)&=\frac{\#\left\{x\in GF(2^m)\mid S(x)\oplus S(x\oplus a)=b\right\}}{2^m}\\ LP_S(a,b)&=\left(\frac{\#\left\{x\in GF(2^n)\mid \Gamma_a\bullet x=\Gamma_b\bullet S(x)\right\}}{2^{m-1}}\right)^2 \end{aligned} \tag{2.6}$$

Tanım 2.7: Bir S-kutusunun maksimum diferansiyel ve lineer olasılığı ifade (2.7)'deki gibi tanımlanır.

$$\begin{aligned} p&=\max_{a\neq 0,b} DP_s(a,b)\\ q&=\max_{\Gamma_a,\Gamma_b\neq 0} LP_s(\Gamma_a,\Gamma_b) \end{aligned} \tag{2.7}$$

Örneğin AES şifreleme algoritmasında kullanılan S-kutusunun maksimum diferansiyel olasılığı ve maksimum doğrusal olasılığı 2^{-6} olduğu aşağıdaki gibi gösterilebilir:

$$DP_S(a,b) = \frac{\#\left\{x \in GF(2^m) \mid S(x) \oplus S(x \oplus a) = b\right\}}{2^m}$$
$$= \frac{4}{2^8} = 2^{-6}$$
$$LP_S(a,b) = \left(\frac{\#\left\{x \in GF(2^n) \mid \Gamma_a \bullet x = \Gamma_b \bullet S(x)\right\}}{2^{m-1}}\right)^2$$
$$= \left(\frac{16}{2^7}\right)^2 = 2^{-6}$$

Tanım 2.8. Bir doğrusal dönüşüm elemanı $A:(\{0,1\}^m)^n \to (\{0,1\}^m)^n$ olmak üzere (2.8) ifadesindeki gibi tanımlanabilir.

$$A(x) = A.x^T = \begin{pmatrix} a_{11} & a_{12} & . & . & . & a_{1n} \\ a_{21} & a_{22} & . & . & . & a_{2n} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ a_{n1} & a_{n2} & . & . & . & a_{nn} \end{pmatrix} . \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{pmatrix} \qquad (2.8)$$

İfade (2.8)'de $x = (x_1, x_2, ..., x_n)^T$, $x_i \in \{0,1\}^m$, $i = 1,...,n$ olmak üzere *n*-bit doğrusal dönüşüme giriş olan *m*-bit giriş ve çıkışlı S-kutularını (yer değiştirme kutuları) temsil etmektedir [33]. Doğrusal dönüşümün matris elemanları $GF(2^8)$ ya da $GF(2)$'nin elemanları olabilir. Eğer elemanlar $GF(2)$'den seçilir ise bu doğrusal dönüşüme ikili doğrusal dönüşüm adı verilir.

Tanım 2.9. *c* kod kelimesinin Hamming ağırlığı wt(*c*) olarak temsil edilebilir ve kod kelimesinin 0 olmayan elemanlarının sayısı olarak tanımlanır. Buna ek olarak $(GF(2^m))^n$ vektör uzayından *n* boyutlu iki vektör arasındaki Hamming uzaklığı da vektörlerin farklılaştığı pozisyon sayısı olarak tanımlanır [34].

Tanım 2.10. $n \times n$ boyutunda bir A matrisinin diferansiyel dallanma sayısı ifade (2.9)'deki gibi tanımlanabilir.

$$\beta_d(A) = \min\{wt(x) + wt(A.x^T) \mid x \in (\{0,1\}^m)^n, x \neq 0\} \tag{2.9}$$

Tanım 2.11. $n \times n$ boyutunda bir A matrisinin doğrusal dallanma sayısı ifade (2.10)'deki gibi tanımlanabilir.

$$\beta_l(A) = \min\{wt(x) + wt(A^T.x^T) \mid x \in (\{0,1\}^m)^n, x \neq 0\} \tag{2.10}$$

İfade (2.9) ve (2.10)'a göre ikili bir doğrusal dönüşüm matrisinin dallanma sayısı, giriş bitlerinin 0 olmayan elemanları ile çıkış bitlerinin 0 olmayan elemanlarının toplamı olarak ifade edilebilir [35].

Bir doğrusal dönüşümün dallanma sayısı o matrisin en kötü durumu için bir ölçü verir. Bu ölçü de takip eden iki döngüde bir doğrusal yaklaşım ya da fark yaklaşım karakteristiğinde bulunan aktif S-kutularının en düşük sınırıdır. Bu tezde üretilen ikili matrislerin doğrusal ve diferansiyel dallanma sayıları eşittir. Dolayısıyla üretilen ikili bir matrisin ve transpozunun dallanma sayısı eşittir.

Tanım 2.12. Bir $GF(2^m)$ üzerine bir $[n, k, d]$ kod, vektör uzayı $(GF(2^m))^n$ 'in k boyutlu bir alt uzayıdır ve n elemanlı iki vektör arasındaki Hamming uzaklığı minimum d dir. Bu özellik ile d en büyük değerdir. Doğrusal bir $[n, k, d]$ kod c için bir G üreteç matris satırları c için bir taban oluşturan $k \times n$ boyutunda bir matristir. Doğrusal $[n, k, d]$ kodlar Singleton sınırı olan $d \leq n - k + 1$ eşitsizliğini sağlar [34].

Önerme 2.1. Bir doğrusal $[n, k, d]$ kod Singleton sınırı olan $d \leq n - k + 1$ eşitsizliğini sağlıyor ise MDS (Maximum Distance Seperable) koddur. Alternatif olarak, bir matrisin MDS matris olabilmesi için satır ve sütunlarından oluşturulan tüm alt matrislerinin determinantının 0'dan farklı olması gerekir [34].

Yukarıdaki önermeden yola çıkarak elemanları $GF(2^m)$ 'den olan bir $n \times n$ matrisin tüm hesaplanması gereken alt matris determinantlarının sayısı

$$\sum_{k=1}^{n-2} \left[\binom{n}{n-k} \right]^2 \tag{2.11}$$

şeklinde verilebilir [36]. Örneğin 4×4 bir matrisin MDS matris olup olmadığını incelemek için hesaplanması gereken alt determinantlarının sayısı 2×2 alt determinantlarının sayısı ile 3×3 alt determinantlarının sayısının toplamı şeklinde elde edilebilir. Bu değer de yukarıdaki ifadeden

$$\left(\binom{4}{2}\right)^2+\left(\binom{4}{3}\right)^2=36+16=52$$

şeklinde bulunabilir. (2.11) ifadesinde 1×1 alt matrisler ve determinantları gözardı edilmiştir. Bunun nedeni bir MDS matrisin tüm elemanlarının 0'dan farklı olması gereğidir. Buna ek olarak tez çalışmasında incelenen ve sonlu cisim elemanlarından oluşan tüm matrisler 0 haricindeki sonlu cisim elemanlarını kullanmaktadır.

Tanım 2.13. $A=circ(a_1,a_2,...,a_n)$ notasyonu A matrisinin dairesel ve her satırının sağa 1 pozisyon hareket ettirilerek elde edildiğini ifade eder. Dolayısıyla dairesel formdaki $n\times n$ boyutlu bir A matrisi ifade (2.12)'te verilmiştir.

$$A=circ(a_1,a_2,...,a_n)=\begin{bmatrix} a_1 & a_2 & ... & a_n \\ a_n & a_1 & ... & a_{n-1} \\ . & . & ... & . \\ a_2 & a_3 & ... & a_1 \end{bmatrix}_{n\times n} \tag{2.12}$$

Tanım 2.14. $A=had(a_1,a_2,...,a_n)$ notasyonu A matrisinin bir Hadamard matris olduğunu ifade etmektedir. Buna göre 4×4 sonlu cisim Hadamard matris (2.13) ifadesindeki gibi verilebilir.

$$A=had(a_1,a_2,a_3,a_4)=\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix} \tag{2.13}$$

Tanım 2.15. Eğer bir *A* matrisinin tersi kendisine eşitse ($A = A^{-1}$), *A* matrisine involutif matris denir.

Önerme 2.2. $A = had(a_1, a_2, a_3, a_4) = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix}$ matrisi elemanları $GF(2^n)$ cismine ait 4×4 bir matris olsun. Buna ek olarak elemanları birbirinden ve 0'dan farklı olsun. O zaman *A* matrisi, sadece ve sadece $\sum_{i=1}^{4} a_i = 1$ (indisler arası toplama işlemi mod 2 toplama veya XOR işlemi anlamına gelmektedir) ve her kare alt matrislerinin determinantı 0'dan farklı ise involutif MDS matristir.

İspat. Yardımcı Önerme 2.1'den *A*'nın her kare alt matrisinin 0'dan farklı olması *A* matrisinin MDS matris olması için yeterli bir şarttır. Bununla beraber A^2 matrisinin sonucu birim matris olarak eğer $\sum_1^4 {a_i}^2 = 1$ ise elde edilebilmektedir.

$$A^2 = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix} \cdot \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix}$$

$$= \begin{bmatrix} \sum_1^4 {a_i}^2 & 0 & 0 & 0 \\ 0 & \sum_1^4 {a_i}^2 & 0 & 0 \\ 0 & 0 & \sum_1^4 {a_i}^2 & 0 \\ 0 & 0 & 0 & \sum_1^4 {a_i}^2 \end{bmatrix}$$

Önerme 2.3 A_{11} herhangi bir tekil olmayan (non-singular) elemanları $GF(2^m)$ 'den olan $\frac{n}{2} \times \frac{n}{2}$ olan herhangi bir matris olsun. O zaman *n*×*n* boyutunda $A_i = \begin{bmatrix} A_{11} & A_{11}^{-1} \\ A_{11}^3 + A_{11} & A_{11} \end{bmatrix}$ matrisi $GF(2^m)$ üzerine involutif bir matristir [37].

Tanım 2.16. *n*×*n* ikili matrislerin maksimum dallanma sayısı ikili [*2n, n*] doğrusal kodların maksimum uzaklığına eşittir [33].

Tanım 2.16'ya ek olarak ikili bir doğrusal dönüşümün maksimum dallanma sayısına sahip olabilmesi için bu kodların maksimum uzaklığının alt ve üst (lower and upper bounds) sınırlarının eşit olması gerekir. Bununla beraber $n \times n$ ikili matrislerin ($n \leq 18$) maksimum dallanma sayısı (maksimum uzaklığı) bilinmektedir çünkü bu ikili matrislerin alt ve üst sınırlarının maksimum uzaklığı birbirine eşittir. Bu tür kodlara Maksimum uzaklıkta ikili doğrusal kodlar (Maximum Distance Binary Linear Codes-MDBL) denmektedir. Örneğin 8×8 boyutundaki bir ikili matrisin maksimum dallanma sayısı 5'tir. Çünkü bu matrislerin alt ve üst sınırları için maksimum mesafe birbirine eşittir ve bu değer 5'tir. Ancak $n \times n$ ikili matrislerin ($n > 18$) maksimum dallanma sayısı bilinmemektedir. Bunun nedeni ise bu ikili matrislerin maksimum uzaklığının alt ve üst sınırları birbirlerine eşit olabilmekte ya da olamamaktadır [33]. Tablo 2.1'de [$2n$, n] doğrusal kodların maksimum uzaklığının alt ve üst sınırları ($1 \leq n \leq 32$) için gösterilmektedir.

Tablo 2.1: [$2n$, n] Doğrusal Kodların Maksimum Uzaklığının Alt ve Üst Sınırları

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

Tanım 2.17. *A* bir doğrusal dönüşüm ve *x* bu doğrusal dönüşüme giriş bloğu olmak üzere *A(x)=x* ise *x* bu doğrusal dönüşümdeki bir sabit noktadır ve bu durum ifade (2.14)'de gösterilmiştir. Bir başka deyişle *A* doğrusal dönüşümü giriş bitlerini aynı çıkış bitlerine eşleştiriyor ise bir sabit nokta oluşmuş olur [38].

$$A(x)=\begin{pmatrix} a_{11} & a_{12} & . & . & . & a_{1n} \\ a_{21} & a_{22} & . & . & . & a_{2n} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ a_{n1} & a_{n2} & . & . & . & a_{nn} \end{pmatrix} . \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{pmatrix} \qquad (2.14)$$

Bir doğrusal dönüşüme giriş bloğunun $GF(2^m)$ den *m*-bit değerlerden oluştuğunu varsayalım. Buna ek olarak doğrusal dönüşüm matrisinin $n \times n$ boyutunda ve *I* matrisinin $n \times n$ boyutunda birim matris olduğunu varsayalım. O zaman doğrusal dönüşüm matrisi A için tüm sabit noktaların sayısı ifade (2.15)'deki denklemin çözülmesi ile elde edilebilir:

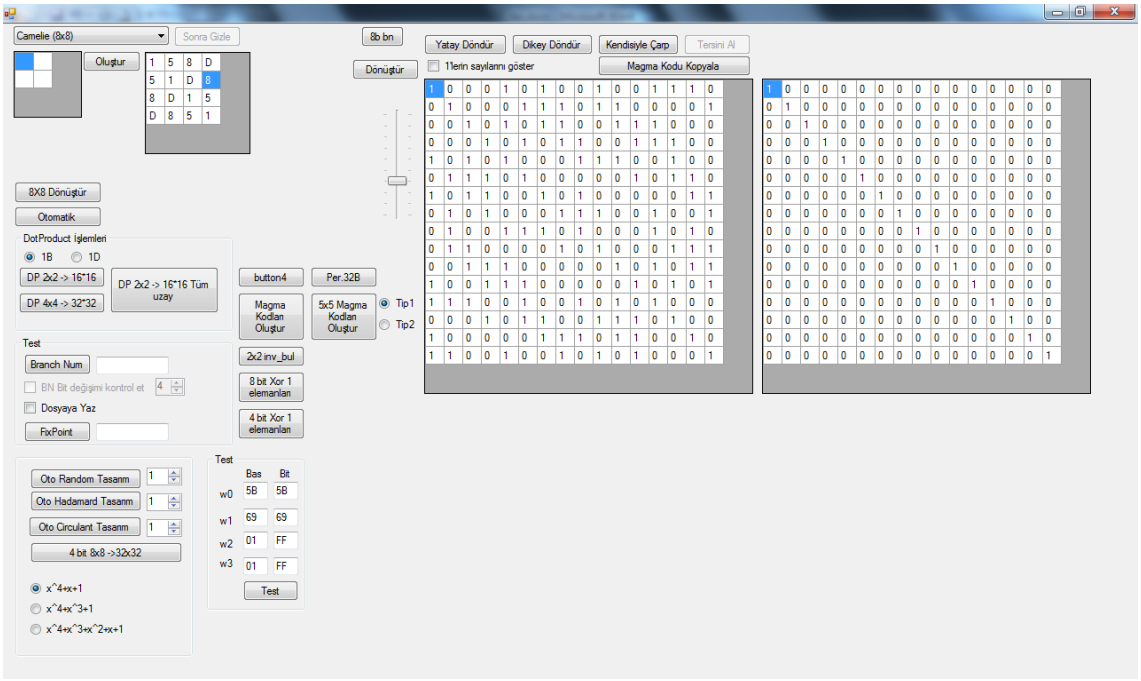
$$(A-I)x^T = 0 \qquad (2.15)$$

(2.15) ifadesindeki 0, *n* uzunluğunda tüm elemanları 0 olan vektörü temsil etmektedir. (2.15) ifadesinden A doğrusal dönüşümündeki sabit noktaların sayısı (2.16) ifadesindeki gibi verilebilir [38].

$$F_A = 2^{m(rank(A)-rank(A-I))} = 2^{m(n-rank(A-I))} \qquad (2.16)$$

(2.16) ifadesinden görülmektedir ki *A* matrisi için *(A-I)* matrisinin rank değeri yüksek ise doğrusal dönüşümdeki sabit nokta sayısının daha az olacaktır.

Tez esnasında kullanılan sonlu cisim işlemleri ve geliştirilen doğrusal dönüşümlerin dallanma sayıları ve sabit nokta sayıları gibi özellikleri geliştirilen bir test yazılımı incelenmiştir (Şekil 2.1). Ayrıca sonuçlar MAGMA [39] yazılımı ile de doğrulanmıştır.



Şekil 2.1: Test için Geliştirilen Yazılımın Arayüzü

Magma Yazılımı

Magma yazılımı, Sydney Üniversitesi'ndeki bir araştırma grubu tarafından geliştirilen bir yazılımdır. C programlama diline benzer bir dil yapısına sahip olan MAGMA, ticari bir yazılımdır. Fakat belirli sınırlamaları olmakla beraber ücretsiz bir hesaplama aracına http://magma.maths.usyd.edu.au/calc/ adresinden ulaşılabilir. Tez sırasında geliştirilen bazı yapılar kendi geliştirdiğimiz yazılım ile test edilmesine ek olarak MAGMA yazılımı üzerinde de test edilerek sonuçlar doğrulanmıştır.

Aşağıda bazı işlemlerin örnek MAGMA kodları verilmiştir:

Örnek 2.3: $M = had(1_h, 2_h, A_h, B_h) = \begin{bmatrix} 1_h & 2_h & A_h & B_h \\ 2_h & 1_h & B_h & A_h \\ A_h & B_h & 1_h & 2_h \\ B_h & A_h & 2_h & 1_h \end{bmatrix}$ 4×4 boyutunda elemanları $x^4 + x + 1$ ile tanımlanan $GF(2^4)$ cisminden seçilmiş hadamard biçiminde olan bir matris olsun. M matrisi MAGMA kodları ile aşağıdaki şekilde kodlanır:

```
P<z> := PolynomialRing(GF(2));
p := z^4+z+1;
F<x> := ext <GF(2)|p>;
M:=Matrix(F,4,4,[
1, x, x^3+x, x^3+x+1,
x,1, x^3+x+1, x^3+x,
x^3+x, x^3+x+1, 1, x,
x^3+x+1, x^3+x, x,1]);
```

M matrisinin determinantını bulmak için aşağıdaki kod kullanılmaktadır:

```
Determinant(M);
```

M matrisinin dallanma sayısını bulabilmek için aşağıdaki kod kullanılmaktadır:

```
I:=IdentityMatrix(F,4);
C := LinearCode(HorizontalJoin(I,M));
MinimumWeight(C);
```

M matrisinin sabit nokta sayısını bulabilmek için aşağıdaki kod kullanılmaktadır:

```
I:=IdentityMatrix(F,4);
2^(4-Rank(M-I));
```

Örnek 2.4: 8×8 boyutunda elemanları A ikili matrisi için MAGMA kodları ile aşağıdaki şekilde kodlanır:

$$A=\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

```
n:=8;
A:=Matrix(GF(2),n,n,[
1,0,1,1,0,1,1,1,
1,1,0,1,1,0,1,1,
1,1,1,0,1,1,0,1,
0,1,1,1,1,1,1,0,
1,1,0,0,0,1,1,1,
0,1,1,0,1,0,1,1,
0,0,1,1,1,1,0,1,
1,0,0,1,1,1,1,0]);
```

A ikili matrisinin determinantını bulmak için aşağıdaki kod kullanılmaktadır:

```
Determinant(A);
```

A ikili matrisinin ve A^T matrisinin dallanma sayısını bulabilmek için aşağıdaki kod kullanılmaktadır:

```
B:=HorizontalJoin(IdentityMatrix(GF(2),8),A);
C:=LinearCode(B);
MinimumWeight(C);
B:=HorizontalJoin(IdentityMatrix(GF(2),8),Transpose(A));
C:=LinearCode(B);
MinimumWeight(C);
```

A ikili matrisinin sabit nokta sayısını bulabilmek için aşağıdaki kod kullanılmaktadır:

```
I:= IdentityMatrix(GF(2),8);
2^(8-Rank(A-I));
```

BÖLÜM 3

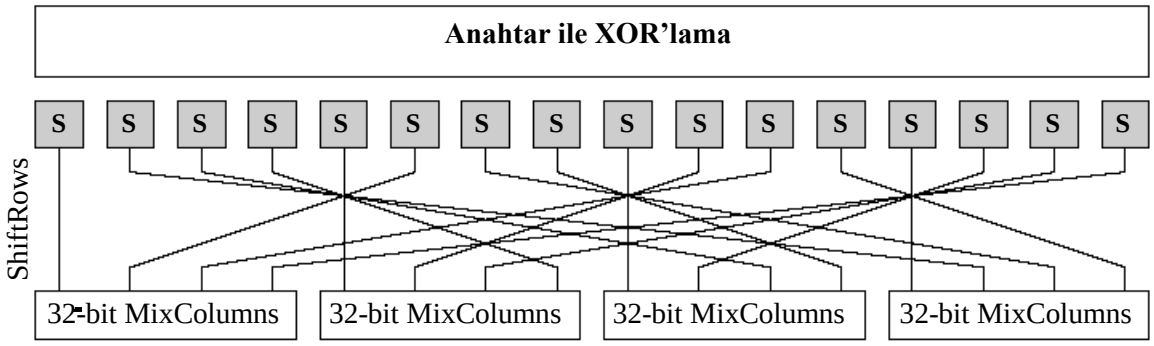
Blok Şifrelerde Kullanılan Doğrusal Dönüşümlerin İncelenmesi

3.1. Giriş

Tezin bu bölümünde önemli blok şifrelerin içyapılarında kullanılmış olan doğrusal dönüşüm elemanları Bölüm 2'de verilen matematiksel alt yapı ışığında incelenmiştir.

3.2.AES Şifresinde Kullanılan Doğrusal Dönüşüm

AES (Advanced Encryption Standard; Gelişmiş Şifreleme Standardı), 128-bit veri bloklarını 128, 192, 256-bit anahtar seçenekleri ile şifreleyen bir algoritmadır [7]. SPN mimarisi tabanlıdır. Döngü sayısı anahtar uzunluğuna göre değişmektedir. 128-bit anahtar uzunluğu için AES şifreleme algoritması 10 döngüde şifreleme yaparken 192 ve 256-bit anahtar uzunlukları için sırasıyla 12 ve 14 döngüde şifreleme yapmaktadır.



Şekil 3.1: AES Blok Şifresindeki Tek Döngülük SPN Mimarisi

Şekil 3.1'de AES blok şifreleme algoritmasının bir döngüsü gösterilmiştir. Şifre içerisinde iki adet doğrusal dönüşüm kullanılmaktadır. Bunlardan ilki ShiftRows dönüşümüdür. ShiftRows aşamasının byte permütasyonudur. Permütasyon işlemi Tablo 3.1'de verilmiştir.

Tablo 3.1: AES ShiftRows Dönüşümü

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p(i)	1	6	11	16	5	10	15	4	9	14	3	8	13	2	7	12

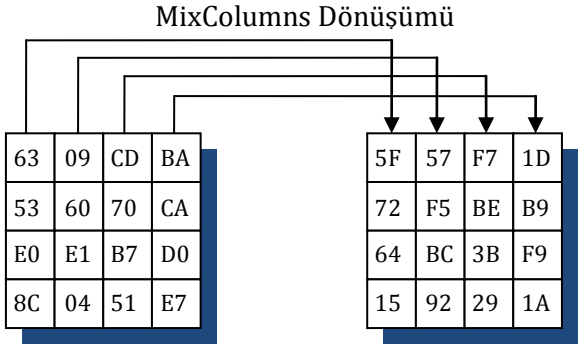
Diğer dönüşüm ise 32-bitten 32-bite dönüşüm yapan sütunları karıştırma (MixColumns) doğrusal dönüşümüdür. Bu dönüşüm doğrusal ve diferansiyel kriptanalizi zorlaştırıcı etki yapma amacındadır ve sonlu cisimde çarpma tabanlıdır. $GF(2^8)$ de elemanlar içeren sütunları karıştırma dönüşümü çarpma işlemlerinde indirgeme işlemleri için $x^8+x^4+x^3+x+1$ polinomunu kullanmaktadır. Bir MDS matris olan bu doğrusal dönüşüm involutif değildir. Dolayısı ile şifreleme ve şifre çözme işlemleri için farklı dönüşümler kullanılmaktadır. AES şifreleme algoritmasında sırasıyla şifreleme ve şifre çözme işlemlerinde kullanılan doğrusal dönüşüm matrisleri (3.1)'de verilmiştir.

$$A_{AES}=\begin{bmatrix} 02_h & 03_h & 01_h & 01_h \\ 01_h & 02_h & 03_h & 01_h \\ 01_h & 01_h & 02_h & 03_h \\ 03_h & 01_h & 01_h & 02_h \end{bmatrix} \rightarrow \text{Şifreleme Doğrusal Dönüşümü}$$

$$A_{AES}^{-1}=\begin{bmatrix} 0\text{E}_h & 0\text{B}_h & 0\text{D}_h & 09_h \\ 09_h & 0\text{E}_h & 0\text{B}_h & 0\text{D}_h \\ 0\text{D}_h & 09_h & 0\text{E}_h & 0\text{B}_h \\ 0\text{B}_h & 0\text{D}_h & 09_h & 0\text{E}_h \end{bmatrix} \rightarrow \text{Deşifreleme Doğrusal Dönüşümü} \tag{3.1}$$

AES'in dairesel formda tasarlanan MDS matrisinin dallanma sayısı 5'tir. Dolayısı ile giriş/çıkış değerlerinden 1 byte'ın değişmesi sonucunda giriş/çıkış byte'larında minimum 4 byte, giriş/çıkış değerlerinden 2 byte'ın değişmesi sonucunda giriş/çıkış byte'larında minimum 3 byte bu durumdan etkilenecektir. Bu doğrusal dönüşümün $(A-I)$ matrisinin rank değeri 3 olduğundan $2^{b(rank(A)-rank(A-I)}=2^{8(4-3)}=2^8$ adet sabit nokta içerir.

Örnek 3.1: 6353E08C0960E104CD70B751BACAD0E7 128-bitlik değerinin yayılım katmanından sonra 5F72641557F5BC92F7BE3B291DB9F91A değerine dönüştüğü Şekil 3.2'deki gibi gösterilebilir.



Şekil 3.2: AES Şifreleme Doğrusal Dönüşüm İşlemi

Örnek 3.1'de verilen giriş değerinin çıkış değerine dönüşmesi aşağıda verilmiştir.

$$\begin{bmatrix} 02_h & 03_h & 01_h & 01_h \\ 01_h & 02_h & 03_h & 01_h \\ 01_h & 01_h & 02_h & 03_h \\ 03_h & 01_h & 01_h & 02_h \end{bmatrix} \cdot \begin{bmatrix} 63_h \\ 53_h \\ E0_h \\ 8C_h \end{bmatrix} = \begin{bmatrix} 5F_h \\ 72_h \\ 64_h \\ 15_h \end{bmatrix}$$

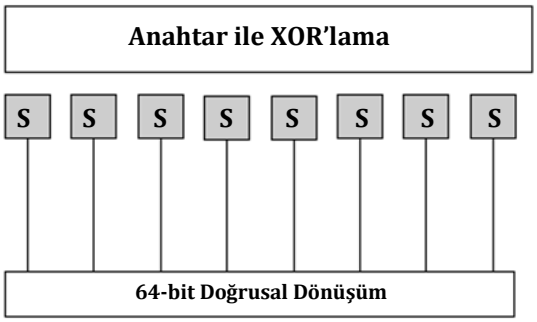
$$\begin{bmatrix} 02_h & 03_h & 01_h & 01_h \\ 01_h & 02_h & 03_h & 01_h \\ 01_h & 01_h & 02_h & 03_h \\ 03_h & 01_h & 01_h & 02_h \end{bmatrix} \cdot \begin{bmatrix} 09_h \\ 60_h \\ E1_h \\ 04_h \end{bmatrix} = \begin{bmatrix} 57_h \\ F5_h \\ BC_h \\ 92_h \end{bmatrix}$$

$$\begin{bmatrix} 02_h & 03_h & 01_h & 01_h \\ 01_h & 02_h & 03_h & 01_h \\ 01_h & 01_h & 02_h & 03_h \\ 03_h & 01_h & 01_h & 02_h \end{bmatrix} \cdot \begin{bmatrix} CD_h \\ 70_h \\ B7_h \\ 51_h \end{bmatrix} = \begin{bmatrix} F7_h \\ BE_h \\ 3B_h \\ 29_h \end{bmatrix}$$

$$\begin{bmatrix} 02_h & 03_h & 01_h & 01_h \\ 01_h & 02_h & 03_h & 01_h \\ 01_h & 01_h & 02_h & 03_h \\ 03_h & 01_h & 01_h & 02_h \end{bmatrix} \cdot \begin{bmatrix} BA_h \\ CA_h \\ D0_h \\ E7_h \end{bmatrix} = \begin{bmatrix} 1D_h \\ B9_h \\ F9_h \\ 1A_h \end{bmatrix}$$

3.3.Khazad Şifresinde Kullanılan Doğrusal Dönüşüm

KHAZAD, Barreto and Rijmen tarafından 2000 yılında 128-bit anahtarla çalışan 64-bit bir blok şifre olarak geliştirilmiştir. 8 döngü ile şifreleme yapmaktadır [10]. KHAZAD, Wide Trail (geniş iz) stratejisine göre tasarlanmıştır. Wide Trail stratejisinde, bir blok şifrenin döngü dönüşümü tersi alınabilir farklı dönüşümlerin birleşiminde oluşur. Her birinin kendi fonksiyonları ve gereksinimleri vardır. Doğrusal yayılım katmanı birkaç döngüden sonra tüm çıkış bitlerinin tüm giriş bitlerine bağımlı olmasını sağlarken doğrusal olmayan katman ise karmaşıklığı ve doğrusal olmamayı sağlar. Anahtar ekleme safhası diğer şifrelerde olduğu gibi o anki döngü çıkışının anahtar ile XOR'lama işlemine tabi tutulmasıdır. Wide Trail stratejinin diğer bir avantajı ise farklı bileşenlerin birbirlerinden tamamen farklı bir şekilde belirlenebilmesidir.



Şekil 3.3. KHAZAD Blok Şifresindeki Tek Döngülük SPN Mimarisi.

KHAZAD şifresinin doğrusal dönüşümü 8×8 boyutunda $GF(2^8)$ cisim elemanlarına sahip sonlu cisim Hadamard ve MDS bir matristir. Dolayısıyla dallanma sayısı 9'dur. Çarpma işlemleri sonucunda indirgeme işlemleri için $x^8 + x^4 + x^3 + x + 1$ polinomunu kullanmaktadır. Bu dönüşüm (3.2) ifadesinde verilmiştir:

$$A_{Khazad} = \begin{bmatrix} 01_h & 03_h & 04_h & 05_h & 06_h & 08_h & 0B_h & 07_h \\ 03_h & 01_h & 05_h & 04_h & 08_h & 06_h & 07_h & 0B_h \\ 04_h & 05_h & 01_h & 03_h & 0B_h & 07_h & 06_h & 08_h \\ 05_h & 04_h & 03_h & 01_h & 07_h & 0B_h & 08_h & 06_h \\ 06_h & 08_h & 0B_h & 07_h & 01_h & 03_h & 04_h & 05_h \\ 08_h & 06_h & 07_h & 0B_h & 03_h & 01_h & 05_h & 04_h \\ 0B_h & 07_h & 06_h & 08_h & 04_h & 05_h & 01_h & 03_h \\ 07_h & 0B_h & 08_h & 06_h & 05_h & 04_h & 03_h & 01_h \end{bmatrix} \quad (3.2)$$

KHAZAD blok şifresinde kullanılan bu doğrusal dönüşüm verilen özelliklerine ek olarak involutif olarak tasarlanmıştır. Bu sebeple şifreleme ve şifre çözme işlemlerinde aynı matris kullanılmaktadır. Böylelikle bu işlemler arasındaki fark kaldırılmış olur. Bu matrisin, $(A-I)$ matrisinin rank değeri 4 olduğundan sabit nokta sayısı değeri aşağıdaki gibi 2^{32} olarak hesaplanabilir:

$$2^{b(rank(A)-rank(A-I)} = 2^{8(8-4)} = 2^{32}$$

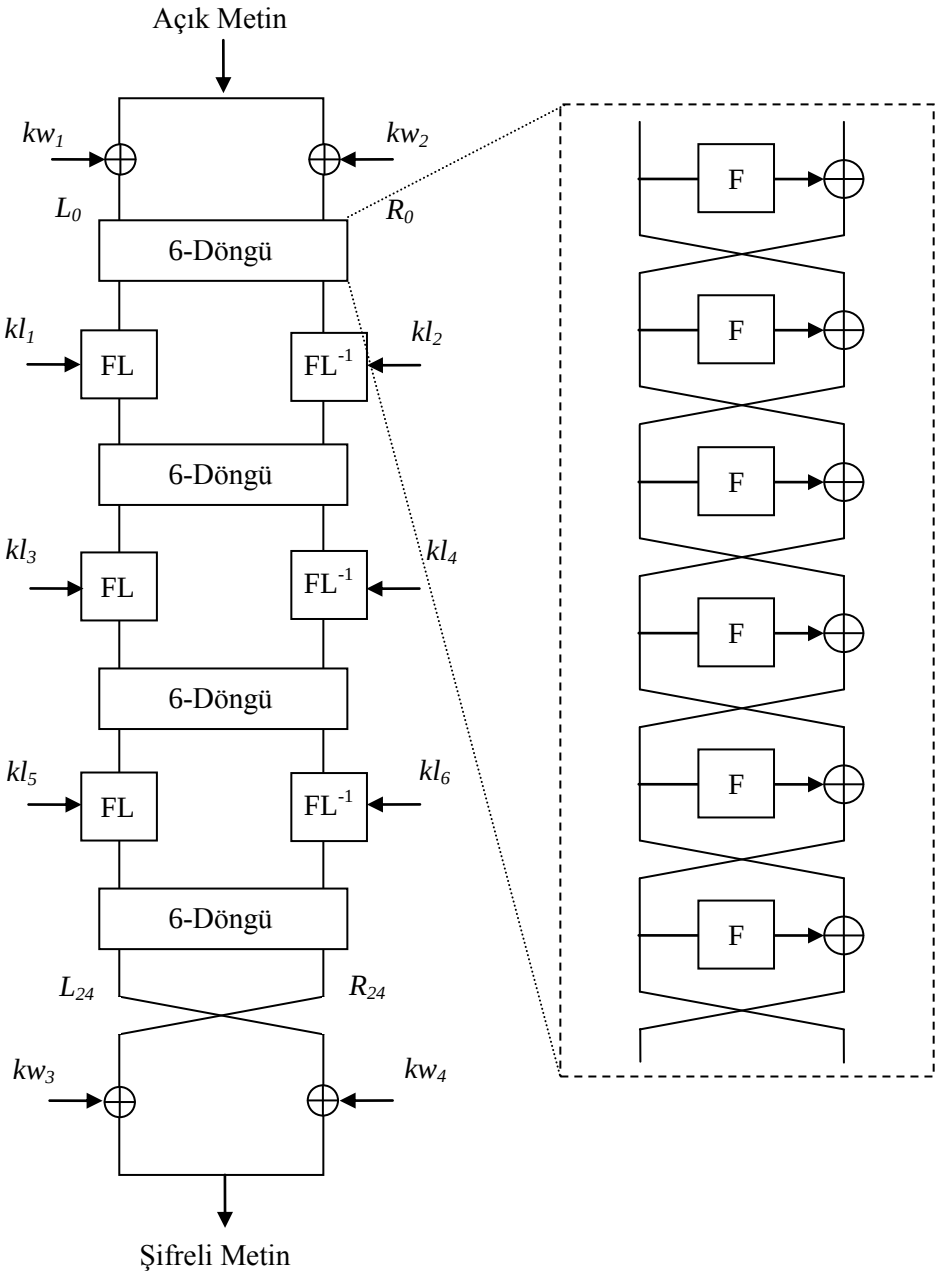
Örnek 3.2: 6353E08C0960E104 64-bitlik değerinin yayılım katmanından sonra 6353E08C0960E104 değerine dönüştüğü ifade (3.3)'te gösterilmiştir.

$$\begin{bmatrix} 01_h & 03_h & 04_h & 05_h & 06_h & 08_h & 0B_h & 07_h \\ 03_h & 01_h & 05_h & 04_h & 08_h & 06_h & 07_h & 0B_h \\ 04_h & 05_h & 01_h & 03_h & 0B_h & 07_h & 06_h & 08_h \\ 05_h & 04_h & 03_h & 01_h & 07_h & 0B_h & 08_h & 06_h \\ 06_h & 08_h & 0B_h & 07_h & 01_h & 03_h & 04_h & 05_h \\ 08_h & 06_h & 07_h & 0B_h & 03_h & 01_h & 05_h & 04_h \\ 0B_h & 07_h & 06_h & 08_h & 04_h & 05_h & 01_h & 03_h \\ 07_h & 0B_h & 08_h & 06_h & 05_h & 04_h & 03_h & 01_h \end{bmatrix} \cdot \begin{bmatrix} 63 \\ 53 \\ E0 \\ 8C \\ 09 \\ 60 \\ E1 \\ 04 \end{bmatrix} = \begin{bmatrix} C7 \\ 13 \\ C4 \\ F7 \\ 18 \\ F2 \\ 7A \\ A7 \end{bmatrix} \quad (3.3)$$

3.4.Camellia Şifresinde Kullanılan Doğrusal Dönüşüm

Camellia, 128-bit veri bloklarını 128, 192 veya 256-bit anahtar seçenekleri ile şifreleyen bir şifreleme algoritmasıdır [8]. 2000 yılında NTT ve Mitsubishi Electric Corporation tarafından ortak geliştirilmiştir. Özellikle düşük maliyetli akıllı kartlar ve

yüksek hızlı ağ sistemleri için uygun bir algoritmadır. Şifreleme işlemi, 128-bit anahtar ile 18 döngüde yapılırken; 192 ve 256-bit anahtar ile 24 döngüde yapılır. Camellia şifreleme algoritması Feistel mimarisi ile tasarlanmıştır. Algoritmanın blok diyagramı Şekil 3.4'te gösterilmiştir.



Şekil 3.4: CAMELLIA Şifreleme Algoritması

18 döngü olarak Şekil 3.4'te verilen algoritmada her 6 döngünün içerisinde bulunan *F* fonksiyonu ifade (3.4)'te verilmiştir. İfade (3.4)'te *x* giriş verisi, *k* döngü anahtarı, *S* s-kutusu ve *P* şifrede kullanılan doğrusal dönüşümdür.

$$F = P(S(x \oplus k)) \tag{3.4}$$

Camellia şifresinde kullanılan doğrusal dönüşüm matrisi MDBL (Maximum Distance Binary Linear) kod olarak tasarlanmıştır. Dallanma sayısı 5 olan dönüşüm (3.5)'te verilmiştir.

$$P = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{3.5}$$

Yukarıda verilen Camellia doğrusal dönüşümünün $(A-I)$ matrisinin rank değeri 7 olduğundan $2^{8(8-7)} = 2^8$ adet sabit nokta içerir. Elemanların *GF*(2) den seçildiği için XOR tabanlı bir doğrusal dönüşümdür. $x_0, x_1, \ldots, x_7$ giriş değerleri ve $y_0, y_1, \ldots, y_7$ çıkış değerleri olmak üzere Camellia şifreleme algoritmasında kullanılan *P* doğrusal dönüşümü ifade (3.6)'da olduğu gibi yazılabilir.

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} \rightarrow \begin{array}{c} y_0 = x_0 \oplus x_2 \oplus x_3 \oplus x_5 \oplus x_6 \oplus x_7 \\ y_1 = x_0 \oplus x_1 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_7 \\ y_2 = x_0 \oplus x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_7 \\ y_3 = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6 \\ y_4 = x_0 \oplus x_1 \oplus x_5 \oplus x_6 \oplus x_7 \\ y_5 = x_1 \oplus x_2 \oplus x_4 \oplus x_6 \oplus x_7 \\ y_6 = x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_7 \\ y_7 = x_0 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6 \end{array} \tag{3.6}$$

Örnek 3.3: 6353E08C0960E104 64-bitlik değerinin yayılım katmanından sonra 8A50BDB7B55F0167 değerine dönüştüğü aşağıdaki gibi hesaplanabilir.

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 63 \\ 53 \\ E0 \\ 8C \\ 09 \\ 60 \\ E1 \\ 04 \end{bmatrix} = \begin{bmatrix} 8A \\ 50 \\ BD \\ B7 \\ B5 \\ 5F \\ 01 \\ 67 \end{bmatrix}$$

3.5.ARIA Şifresinde Kullanılan Doğrusal Dönüşüm

ARIA blok şifresi Güney Koreli araştırmacılar tarafından 2004'de tasarlanmıştır ve Kore Teknoloji Ajansı tarafından standart şifreleme tekniği olarak seçilmiştir [9]. ARIA, verileri 128 bit bloklarla 128, 192 ve 256-bit anahtar seçenekleri ile şifreleyen bir şifreleme algoritmasıdır. Döngü sayıları anahtar uzunluğuna göre sırasıyla 10, 12 ya da 14'tür.

ARIA şifresi yayılım katmanı olarak 16×16 boyutunda giriş değerlerini byte değerler olarak alan involutif ve MDBL kod olarak tasarlanmış bir ikili matris kullanır. ARIA matrisinde kullanılan bu involutif ikili matrisin, $(A-I)$ matrisinin rank değeri 7 olduğundan bu dönüşüm $2^{8(16-7)} = 2^{72}$ adet sabit noktaya içerir. Dallanma sayısı 8 olan bu ikili matris (3.7)'de verilmiştir.

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{pmatrix} \quad (3.7)$$

ARIA' nın yayılım katmanı (x_0, x_1,…, x_{15}) girişlerini 16 byte'lık (y_0, y_1,…, y_{15}) çıkışlarına haritalar. Bu haritalama işlemi cebirsel olarak (3.8)'deki gibi ifade edilebilir.

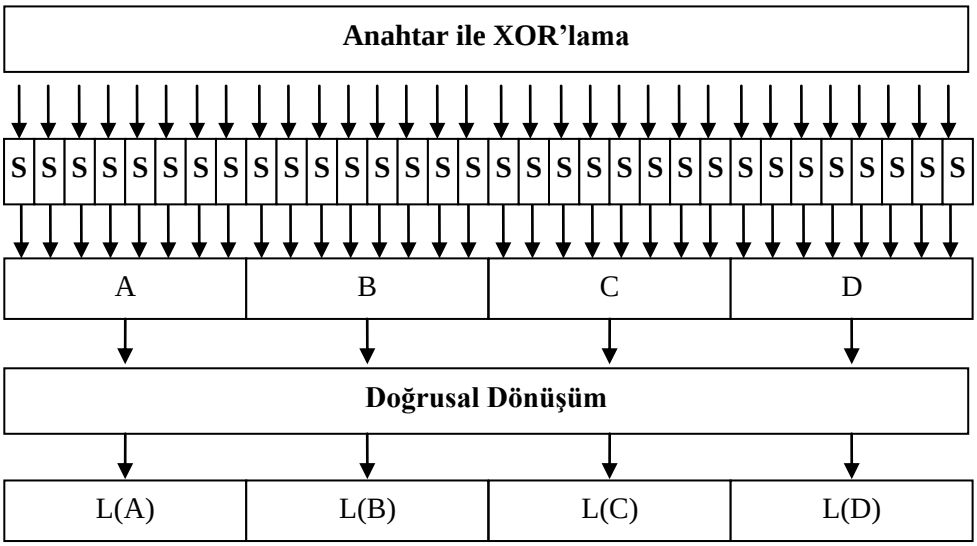
$$\begin{aligned}
y_0 &= x_3 \oplus x_4 \oplus x_6 \oplus x_8 \oplus x_9 \oplus x_{13} \oplus x_{14} \\
y_1 &= x_2 \oplus x_5 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{12} \oplus x_{15} \\
y_2 &= x_1 \oplus x_4 \oplus x_6 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus x_{15} \\
y_3 &= x_0 \oplus x_5 \oplus x_7 \oplus x_{10} \oplus x_{11} \oplus x_{13} \oplus x_{14} \\
y_4 &= x_0 \oplus x_2 \oplus x_5 \oplus x_8 \oplus x_{11} \oplus x_{14} \oplus x_{15} \\
y_5 &= x_1 \oplus x_3 \oplus x_4 \oplus x_9 \oplus x_{10} \oplus x_{14} \oplus x_{15} \\
y_6 &= x_0 \oplus x_2 \oplus x_7 \oplus x_9 \oplus x_{10} \oplus x_{12} \oplus x_{13} \\
y_7 &= x_1 \oplus x_3 \oplus x_6 \oplus x_8 \oplus x_{11} \oplus x_{12} \oplus x_{13} \\
y_8 &= x_0 \oplus x_1 \oplus x_4 \oplus x_7 \oplus x_{10} \oplus x_{13} \oplus x_{15} \\
y_9 &= x_0 \oplus x_1 \oplus x_5 \oplus x_6 \oplus x_{11} \oplus x_{12} \oplus x_{14} \\
y_{10} &= x_2 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_8 \oplus x_{13} \oplus x_{15} \\
y_{11} &= x_2 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_9 \oplus x_{12} \oplus x_{14} \\
y_{12} &= x_1 \oplus x_2 \oplus x_6 \oplus x_7 \oplus x_9 \oplus x_{11} \oplus x_{12} \\
y_{13} &= x_0 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_8 \oplus x_{10} \oplus x_{13} \\
y_{14} &= x_0 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_9 \oplus x_{11} \oplus x_{14} \\
y_{15} &= x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_8 \oplus x_{10} \oplus x_{15}
\end{aligned} \quad (3.8)$$

Bölüm 8'de ARIA şifreleme algoritması detaylı bir şekilde anlatılmıştır.

3.6.SERPENT Şifresinde Kullanılan Doğrusal Dönüşüm

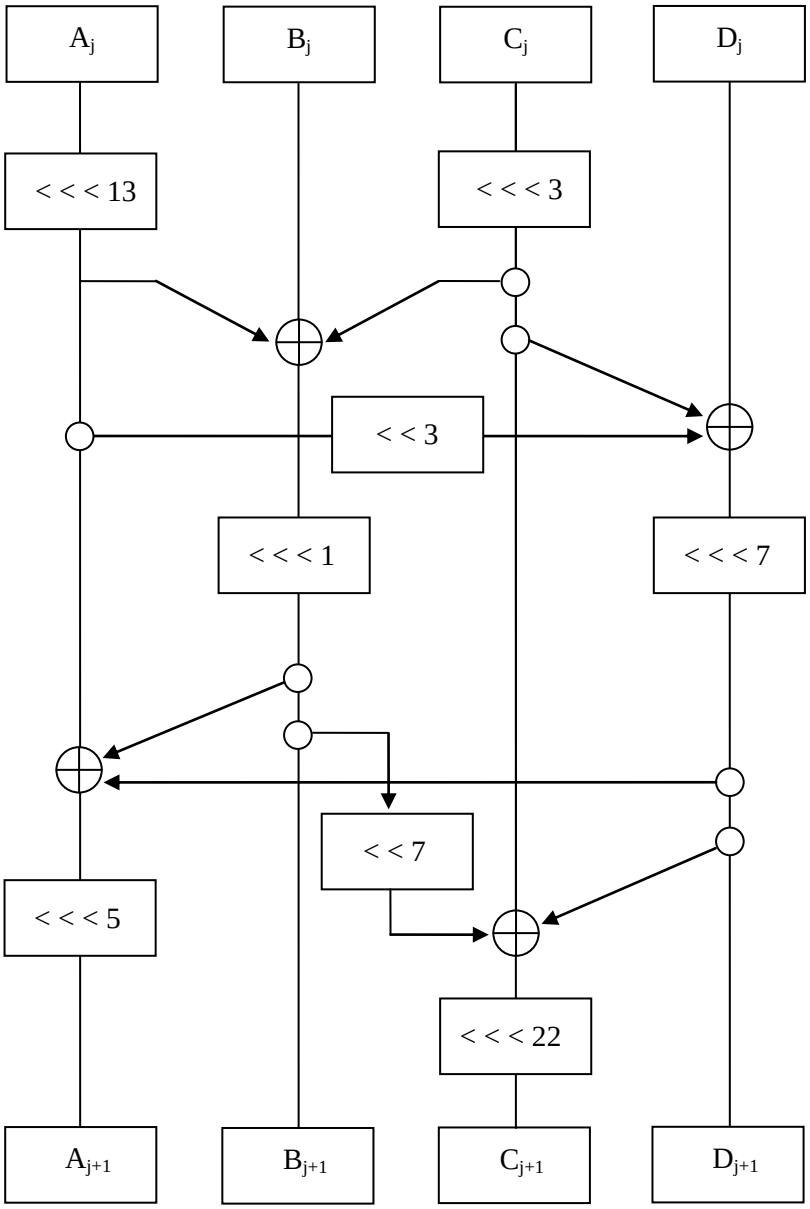
SERPENT, AES olarak kabul edilen Rijndael algoritması ile aynı yarışmada yer alan bir şifreleme algoritmasıdır [6]. 128-bit veri bloklarını, 128, 192, 256-bit anahtar seçenekleri ile 32 döngüde şifreleyen algoritma Ross Anderson, Eli Biham ve Lars Knudsen tarafından geliştirilmiştir.

Şifrenin tasarımcıları, günümüzdeki saldırılar için 16 döngünün yeterli olduğunu fakat gelecekte geliştirilebilecek saldırılar için döngü sayısının 32'ye çıkarılması gerektiğini belirtmişlerdir. SPN mimarisi ile inşa edilen şifreleme algoritmasının blok diyagramı Şekil 3.5'te verilmiştir.



Şekil 3.5. SERPENT Blok Şifresindeki Tek Döngülük SPN Mimarisi

Şekil 3.5'te verilen mimariden anlaşılacağı gibi S-kutularından geçen 128-bit veri 32-bit boyutunda 4 bölüme ayrılarak doğrusal dönüşüm aşamasına gelir. Serpent şifreleme algoritması içerisinde kullanılan doğrusal dönüşüm aşamasında uygulanan işlemler Şekil 3.6'da verilmiştir. Doğrusal dönüşüm katmanına gelen 128-bit 32-bit uzunluğunda 4 bloğa ayrılarak Şekil 3.7'deki gibi işlenir.



Şekil 3.6. SERPENT Blok Şifresindeki Doğrusal Dönüşüm Katmanı

Serpent doğrusal dönüşüm tasarımı, (<<<) döndürme, (<<) kaydırma olmak üzere (3.9)'da olduğu gibi gerçekleştirilebilir. Doğrusal dönüşüm ifade (3.9)'da olduğu gibi uygulanabilir.

$$
\begin{aligned}
A &= A \lll 13 \\
C &= C \lll 3 \\
B &= A \oplus B \oplus C \\
D &= D \oplus C \oplus (A \ll 3) \\
B &= B \lll 1 \\
D &= D \lll 7 \\
A &= A \oplus B \oplus D \\
C &= C \oplus D \oplus (B \ll 7) \\
A &= A \lll 5 \\
C &= C \lll 22
\end{aligned}
\tag{3.9}
$$

Serpent doğrusal dönüşüm fonksiyonunun dallanma sayısı 3 ve sabit nokta sayısı 2'dir.

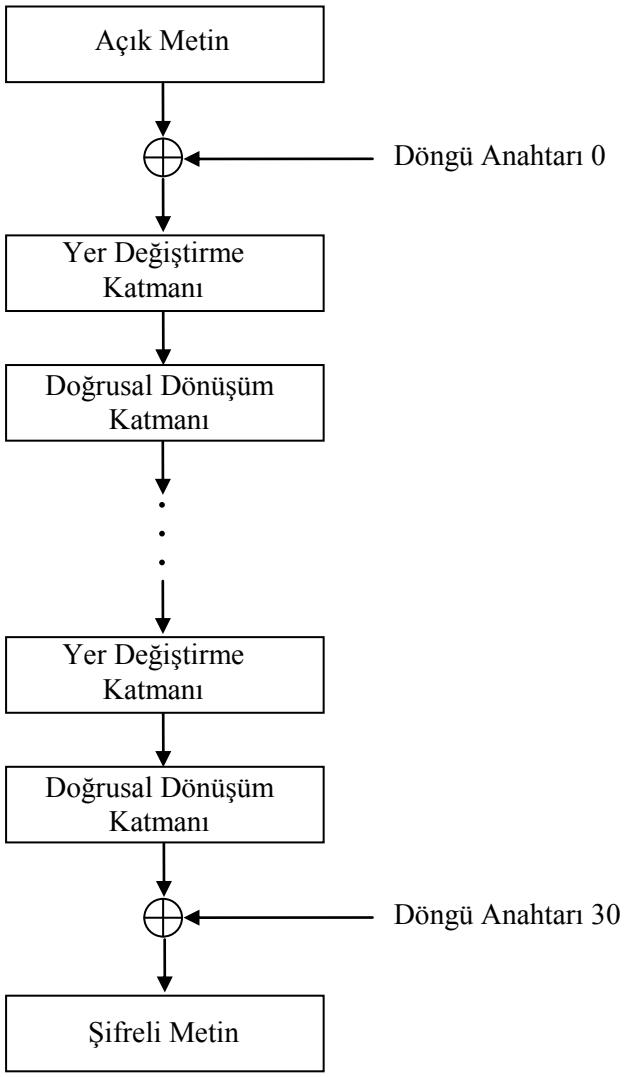
Örnek 3.4: 6353E08C0960E104CD70B751BACAD0E7 128-bitlik değerinin yayılım katmanından sonra 0F97E6543DE9AFC0859FB5B1E1849C98 değerine dönüştüğü aşağıdaki gibi gösterilebilir.

Başlangıç Değerleri	A=01100011010100111110000010001100 B=00001001011000001110000100000100 C=11001101011100001011011101010001 D=10111010110010101101000011100111
$C = C \lll 3$	C=01111100000100011000110001101010
$B = A \oplus B \oplus C$	B=00011110111101001101011111100000
$D = D \oplus C \oplus (A \ll 3)$	D=00110001110000110000100100111001
$B = B \lll 1$	B=00111101111010011010111111000000
$D = D \lll 7$	D=11100001100001001001110010011000
$A = A \oplus B \oplus D$	A=10100000011111001011111100110010
$C = C \oplus D \oplus (B \ll 7)$	C=01111110110101101100011000010110
$A = A \lll 5$	A=00001111100101111110011001010100
$C = C \lll 22$	C=10000101100111111011010110110001

Bitiş Değerleri	A=00001111100101111110011001010100 B=00111101111010011010111111000000 C=10000101100111111011010110110001 D=11100001100001001001110010011000

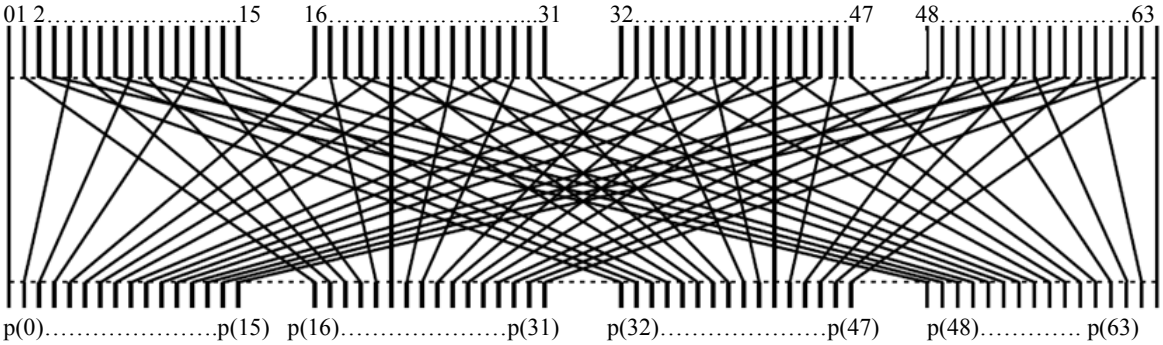
3.7.PRESENT Şifresinde Kullanılan Doğrusal Dönüşüm

Present 2007 yılında daha çok RFID gibi düşük kapasiteli donanımlar için SPN mimari ile tasarlanmış 31 döngüde şifreleme yapan bir blok şifredir [5]. 64-bit blokları, 80 veya 128-bit anahtarlar ile şifreler.



Şekil 3.7. PRESENT Blok Şifresinin Blok Diyagramı

Şifre içerisinde kullanılan doğrusal dönüşüm Player adında bit tabanlı bir permütasyon işlemidir. Bu dönüşüm Şekil 3.8'de verilmiştir. Dönüşüm tablosu Tablo 3.3'de verilmiştir. Dönüşümün dallanma sayısı 2 olmakla beraber sabit nokta sayısı 2^{24}'tür.

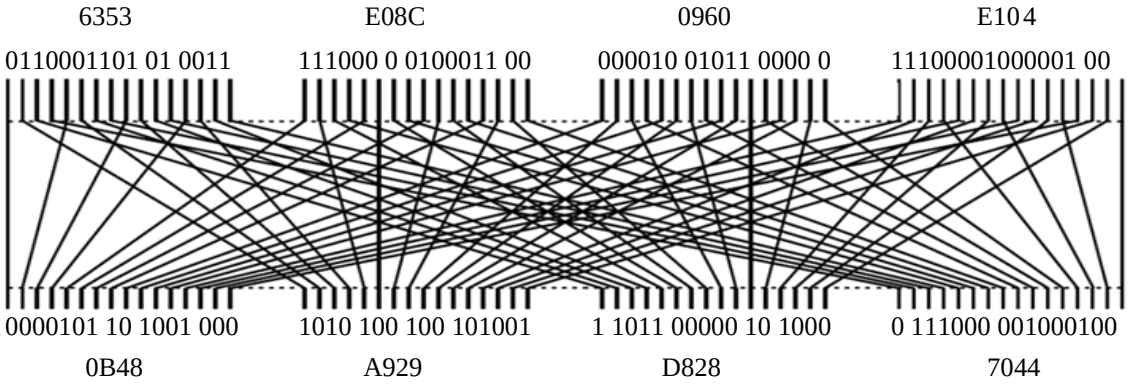


Şekil 3.8. PRESENT İçerisindeki PLayer Dönüşümü

Tablo 3.2: PRESENT PLayer Dönüşüm Tablosu

i	***P(i)***		***i***	***P(i)***		***i***	***P(i)***		***i***	***P(i)***
0	0		16	4		32	8		48	12
1	16		17	20		33	24		49	28
2	32		18	36		34	40		50	44
3	48		19	52		35	56		51	60
4	1		20	5		36	9		52	13
5	17		21	21		37	25		53	29
6	33		22	37		38	41		54	45
7	49		23	53		39	57		55	61
8	2		24	6		40	10		56	14
9	18		25	22		41	26		57	30
10	34		26	38		42	42		58	46
11	50		27	54		43	58		59	62
12	3		28	7		44	11		60	15
13	19		29	23		45	27		61	31
14	35		30	39		46	43		62	47
15	51		31	55		47	59		63	63

Örnek 3.5: 6353E08C0960E104 64-bitlik değerinin yayılım katmanından sonra 0B48A929D8287044 değerine dönüştüğü Şekil 3.9'da gösterilmiştir.



Şekil 3.9. PRESENT İçerisindeki PLayer Dönüşümü için Örnek

3.8.Sonuçlar

Tezin bu bölümünde, literatürde bulunan bazı önemli blok şifreleme algoritmalarının doğrusal dönüşümleri incelenerek özellikle dallanma sayısı ve sabit nokta sayısı gibi önemli kriptografik özellikleri irdelenmiştir. İncelenen doğrusal dönüşümlerin genel özellikleri Tablo 3.3'de verilmiştir.

Tablo 3.3: İncelenen Blok Şifrelerin Özellikleri

Şifre	**Tür**	**Dallanma Sayısı**	**Sabit Nokta Sayısı**	**Giriş-Çıkış $GF(2^m)$**
AES	4×4 MDS matris (MixColumns)	5	2^8	$GF(2^8)$
Khazad	8×8 involutif MDS matris	9	2^{32}	$GF(2^8)$
Camellia	8×8 MDBL matris	5	2^8	$GF(2^8)$
ARIA	16×16 involutif MDBL matris	8	2^{72}	$GF(2^8)$
Serpent	Bit Permütasyonu	3	2	$GF(2)$
Present	Bit Permütasyonu	2	2^{24}	$GF(2)$

Tablo 3.3 incelendiğinde aşağıdaki sonuçlar çıkartılabilir:

- Blok şifreleme algoritmalarının büyük bölümü maksimum dallanma değerine sahip MDS veya MDBL kodlarından seçilmiştir,
- Bazı şifreleme algoritmalarında involutif yapılar tercih edilmiştir. Bu şekilde şifreleme ve şifre çözme adımlarında aynı doğrusal dönüşüm kullanılmış ve bu iki işlem arasındaki uygulama farkını ortadan kaldırma amaçlıdır,
- Bazı şifreleme algoritmaları doğrusal dönüşüm katmanı için bit permütasyonlarını tercih etmiştir.
- Bazı şifreleme algoritmaları ikili doğrusal dönüşümler kullanmış ve karmaşık matris çarpımlarının yerine sadece XOR tabanlı bir dönüşüm mekanizması kullanmışlardır,
- Rastsal bir doğrusal dönüşümden beklenen sabit nokta sayısının 1 olmasına rağmen şifrelerde kullanılan doğrusal dönüşümlerin çok fazla sayıda sabit nokta içerdiği görülmektedir.

BÖLÜM 4

İkili Doğrusal Dönüşüm Tasarımı için Yeni Bir Yöntem

4.1. Giriş

Tezin bu bölümüde, elemanları $GF(2^4)$ cismine ait 2×2 ve 4×4 boyutunda matrisler kullanılarak 8×8 ve 16×16 boyutunda kriptografik özellikleri iyi ikili doğrusal dönüşümlerin tasarımı için cebirsel bir yöntem geliştirilmektedir. Dolayısıyla bu cismin tanımlanması için kullanılacak indirgenemez polinomlar x^4+x+1, x^4+x^3+1 ve $x^4+x^3+x^2+x+1$ şeklindedir. Buna ek olarak bu çalışmada bir $x \in GF(2^4)$ elemanının temsili, bu polinomun katsayılarını ifade eden hexadecimal değerler (alt indis *h* ile tanımlanmaktadır) ile gerçekleştirilmektedir. Örneğin daha önceki bölümlerde belirtildiği gibi $\alpha^3+\alpha^2$ elemanı $GF(2^4)$ sonlu cisminde C_h ile temsil edilmektedir.

$GF(2^4)$ cismi x^4+x+1 indirgenemez polinomu ile tanımlansın ve bu polinomun bir kökü α olsun. O zaman, herhangi bir $x \in GF(2^4)$ için;

$$x = x_3\alpha^3 + x_2\alpha^2 + x_1\alpha + x_0$$

şeklinde yazılabilir ve $(x_3,...,x_0) \in GF(2)$ dir. Buna ek olarak $(\alpha_3,\alpha_2,\alpha_1,\alpha_0) = (\alpha^3,\alpha^2,\alpha^1,1)$ $GF(2)$ üzerine $GF(2^4)$'ün bir polinom tabanıdır. Sonlu cisim $GF(2^4)$'ün bir elemanı 2_h ya da α'nın *x* ile çarpımı;

$$x_3\alpha^3 + x_2\alpha^2 + x_1\alpha + x_0 \rightarrow \alpha(x_3\alpha^3 + x_2\alpha^2 + x_1\alpha + x_0) \bmod p(x)$$

$$= x_3\alpha^4 + x_2\alpha^3 + x_1\alpha + x_0\alpha$$

$$= x_2\alpha^3 + x_1\alpha^2 + (x_3 + x_0)\alpha + x_3$$

şeklinde elde edilebilir ve bu da aşağıdaki 4×4 boyutunda ikili bir matris şeklinde ifade edilebilir:

$$\begin{bmatrix} x_0' \\ x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Yukarıda gösterildiği gibi $GF(2^4)$ cisminin diğer elemanları eşdeğer 4×4 boyutunda ikili matrislere dönüştürülebilir ve elemanları $GF(2^4)$ cismine ait 2×2 ve 4×4 boyutunda matrisler sırası ile 8×8 ve 16×16 boyutunda ikili matrislere dönüştürülebilir.

Benzer bir şekilde, $GF(2^4)$ cismi $x^4 + x + 1$ indirgenemez polinomu için 3_h ya da $\alpha + 1$ elemanının çarpımı aşağıdaki gibi ifade edilebilir.

$$\begin{aligned} x_3\alpha^3 + x_2\alpha^2 + x_1\alpha + x_0 &= (\alpha + 1)(x_3\alpha^3 + x_2\alpha^2 + x_1\alpha + x_0) \bmod p(x) \\ &= x_3\alpha^4 + x_2\alpha^3 + x_1\alpha^2 + x_0\alpha + x_3\alpha^3 + x_2\alpha^2 + x_1\alpha + x_0 \\ &= x_3\alpha + x_3 + x_2\alpha^3 + x_1\alpha^2 + x_0\alpha + x_3\alpha^3 + x_2\alpha^2 + x_1\alpha + x_0 \\ &= \alpha^3(x_3 + x_2) + \alpha^2(x_2 + x_1) + \alpha(x_3 + x_1 + x_0) + (x_3 + x_0) \end{aligned}$$

Dolayısı ile 3_h yada $\alpha + 1$ elemanını aşağıda verilen ikili matris gibi temsil edilebilir.

$$\begin{bmatrix} x_0' \\ x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Başka bir örnek olarak, $GF(2^4)$ cismi $x^4 + x + 1$ indirgenemez polinomu için 4_h yada α^2 elemanının çarpımı aşağıdaki gibi ifade edilebilir.

$$\begin{aligned} x_3\alpha^3 + x_2\alpha^2 + x_1\alpha + x_0 &= \alpha^2(x_3\alpha^3 + x_2\alpha^2 + x_1\alpha + x_0) \bmod p(x) \\ &= x_3\alpha^5 + x_2\alpha^4 + x_1\alpha^3 + x_0\alpha^2 \\ &= x_3\alpha^2 + x_3\alpha + x_2\alpha + x_2 + x_1\alpha^3 + x_0\alpha^2 \\ &= \alpha^3(x_1) + \alpha^2(x_3 + x_0) + \alpha(x_3 + x_2) + (x_2) \end{aligned}$$

Dolayısı ile 4_h yada α^2 elemanını aşağıda verilen ikili matris gibi temsil edilebilir.

$$\begin{bmatrix} x_0' \\ x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Tez çalışmasında, $GF(2^4)$ sonlu cisim elemanlarını temsil etmek için üretilecek ikili matrislerin elde edilmesi esnasında sonlu cisim elemanı 0 hariç diğer sonlu cisim elemanları kullanılmaktadır. Bundan dolayı, yukarıda verilen örneklerden yola çıkarak, Tablo 4.1'de 0 hariç diğer $x^4 + x + 1$ ile tanımlı $GF(2^4)$ cisminin elemanlarının 4×4 boyutunda ikili matris karşılıkları verilmektedir.

Tablo 4.1: İndirgenemez Polinom $x^4 + x + 1$ ile Tanımlı $GF(2^4)$ Cisminin Elemanlarının *x* Sonlu Cisim Elemanı ile Çarpımı Sonucu Üretilen 4×4 Büyüklüğünde İkili Doğrusal Dönüşümler

Hex. Değer	Eşdeğer 4×4 Büyüklüğünde Doğrusal Dönüşüm	Hex. Değer	Eşdeğer 4×4 Büyüklüğünde Doğrusal Dönüşüm	Hex. Değer	Eşdeğer 4×4 Büyüklüğünde Doğrusal Dönüşüm
1	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	2	$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	3	$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$
4	$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$	5	$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$	6	$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$

7	$\begin{bmatrix}1&0&1&1\\1&1&1&0\\1&1&1&1\\0&1&1&1\end{bmatrix}$	8	$\begin{bmatrix}0&1&0&0\\0&1&1&0\\0&0&1&1\\1&0&0&1\end{bmatrix}$	9	$\begin{bmatrix}1&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0\end{bmatrix}$
A	$\begin{bmatrix}0&1&0&1\\1&1&1&1\\0&1&1&1\\1&0&1&1\end{bmatrix}$	B	$\begin{bmatrix}1&1&0&1\\1&0&1&1\\0&1&0&1\\1&0&1&0\end{bmatrix}$	C	$\begin{bmatrix}0&1&1&0\\0&1&0&1\\1&0&1&0\\1&1&0&1\end{bmatrix}$
D	$\begin{bmatrix}1&1&1&0\\0&0&0&1\\1&0&0&0\\1&1&0&0\end{bmatrix}$	E	$\begin{bmatrix}0&1&1&1\\1&1&0&0\\1&1&1&0\\1&1&1&1\end{bmatrix}$	F	$\begin{bmatrix}1&1&1&1\\1&0&0&0\\1&1&0&0\\1&1&1&0\end{bmatrix}$

Tablo 4.1'de olduğu gibi x^4+x^3+1 ve $x^4+x^3+x^2+x+1$ polinomları ile oluşturulacak $GF(2^4)$ cisimlerinin 0 hariç tüm elemanlarının 4×4 boyutunda ikili matris temsilleri sırası ile Tablo 4.2 ve Tablo 4.3'te verilmiştir.

Tablo 4.2: İndirgenemez Polinom x^4+x^3+1 ile Tanımlı $GF(2^4)$ Cisminin Elemanlarının *x* Sonlu Cisim Elemanı ile Çarpımı Sonucu Üretilen 4×4 Büyüklüğünde İkili Doğrusal Dönüşümler

Hex. Değer	Eşdeğer 4×4 Büyüklüğünde Doğrusal Dönüşüm	Hex. Değer	Eşdeğer 4×4 Büyüklüğünde Doğrusal Dönüşüm	Hex. Değer	Eşdeğer 4×4 Büyüklüğünde Doğrusal Dönüşüm
1	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$	2	$\begin{bmatrix}0&0&0&1\\1&0&0&0\\0&1&0&0\\0&0&1&1\end{bmatrix}$	3	$\begin{bmatrix}1&0&0&1\\1&1&0&0\\0&1&1&0\\0&0&1&0\end{bmatrix}$

4	$\begin{bmatrix}0&0&1&1\\0&0&0&1\\1&0&0&0\\0&1&1&1\end{bmatrix}$	5	$\begin{bmatrix}1&0&1&1\\0&1&0&1\\1&0&1&0\\0&1&1&0\end{bmatrix}$	6	$\begin{bmatrix}0&0&1&0\\1&0&0&1\\1&1&0&0\\0&1&0&0\end{bmatrix}$
7	$\begin{bmatrix}1&0&1&0\\1&1&0&1\\1&1&1&0\\0&1&0&1\end{bmatrix}$	8	$\begin{bmatrix}0&1&1&1\\0&0&1&1\\0&0&0&1\\1&1&1&1\end{bmatrix}$	9	$\begin{bmatrix}1&1&1&1\\0&1&1&1\\0&0&1&1\\1&1&1&0\end{bmatrix}$
A	$\begin{bmatrix}0&1&1&0\\1&0&1&1\\0&1&0&1\\1&1&0&0\end{bmatrix}$	B	$\begin{bmatrix}1&1&1&0\\1&1&1&1\\0&1&1&1\\1&1&0&1\end{bmatrix}$	C	$\begin{bmatrix}0&1&0&0\\0&0&1&0\\1&0&0&1\\1&0&0&0\end{bmatrix}$
D	$\begin{bmatrix}1&1&0&0\\0&1&1&0\\1&0&1&1\\1&0&0&1\end{bmatrix}$	E	$\begin{bmatrix}0&1&0&1\\1&0&1&0\\1&1&0&1\\1&0&1&1\end{bmatrix}$	F	$\begin{bmatrix}1&1&0&1\\1&1&1&0\\1&1&1&1\\1&0&1&0\end{bmatrix}$

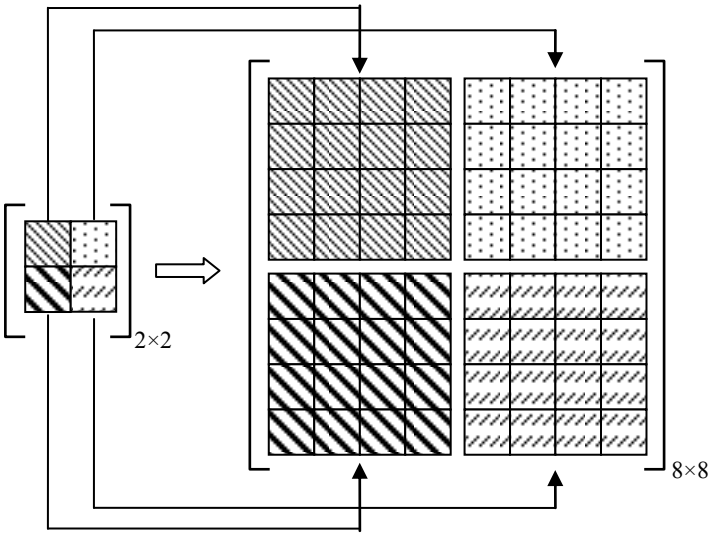
Tablo 4.3: İndirgenemez Polinom $x^4+x^3+x^2+x+1$ ile Tanımlı $GF(2^4)$ Cisminin Elemanlarının *x* Sonlu Cisim Elemanı ile Çarpımı Sonucu Üretilen 4×4 Büyüklüğünde İkili Doğrusal Dönüşümler

Hex. Değer	Eşdeğer 4×4 Büyüklüğünde Doğrusal Dönüşüm	Hex. Değer	Eşdeğer 4×4 Büyüklüğünde Doğrusal Dönüşüm	Hex. Değer	Eşdeğer 4×4 Büyüklüğünde Doğrusal Dönüşüm
1	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$	2	$\begin{bmatrix}0&0&0&1\\1&0&0&1\\0&1&0&1\\0&0&1&1\end{bmatrix}$	3	$\begin{bmatrix}1&0&0&1\\1&1&0&1\\0&1&1&1\\0&0&1&0\end{bmatrix}$

4	$\begin{bmatrix}0&0&1&1\\0&0&1&0\\1&0&1&0\\0&1&1&0\end{bmatrix}$	5	$\begin{bmatrix}1&0&1&1\\0&1&1&0\\1&0&0&0\\0&1&1&1\end{bmatrix}$	6	$\begin{bmatrix}0&0&1&0\\1&0&1&1\\1&1&1&1\\0&1&0&1\end{bmatrix}$
7	$\begin{bmatrix}1&0&1&0\\1&1&1&1\\1&1&0&1\\0&1&0&0\end{bmatrix}$	8	$\begin{bmatrix}0&1&1&0\\0&1&0&1\\0&1&0&0\\1&1&0&0\end{bmatrix}$	9	$\begin{bmatrix}1&1&1&0\\0&0&0&1\\0&1&1&0\\1&1&0&1\end{bmatrix}$
A	$\begin{bmatrix}0&1&1&1\\1&1&0&0\\0&0&0&1\\1&1&1&1\end{bmatrix}$	B	$\begin{bmatrix}1&1&1&1\\1&0&0&0\\0&0&1&1\\1&1&1&0\end{bmatrix}$	C	$\begin{bmatrix}0&1&0&1\\0&1&1&1\\1&1&1&0\\1&0&1&0\end{bmatrix}$
D	$\begin{bmatrix}1&1&0&1\\0&0&1&1\\1&1&0&0\\1&0&1&1\end{bmatrix}$	E	$\begin{bmatrix}0&1&0&0\\1&1&1&0\\1&0&1&1\\1&0&0&1\end{bmatrix}$	F	$\begin{bmatrix}1&1&0&0\\1&0&1&0\\1&0&0&1\\1&0&0&0\end{bmatrix}$

4.2. 2×2 Boyutundaki Matrislerin 8×8 Boyutundaki İkili Matrislere Dönüşümü

Tablo 4.1, Tablo 4.2 veya Tablo 4.3 kullanılarak, 2×2 boyutundaki elemanları $GF(2^4)$'ten seçilmiş olan matrisler 8×8 boyutundaki ikili matrislere Şekil 4.1'deki biçimde dönüştürülebilir.



Şekil 4.1: 2×2 Boyutundaki Matrisin 8×8 Boyutundaki Matrislere Dönüştürülmesi

Örnek 4.1: Tablo 4.1 kullanılarak elemanları x^4+x+1 indirgenemez polinomu ile tanımlı $GF(2^4)$ cisminden olan 2×2 boyutundaki $A=\begin{bmatrix} E_h & 3_h \\ 6_h & E_h \end{bmatrix}$ matrisi 8×8 büyüklüğünde bir matrise aşağıdaki gibi dönüştürülebilir.

$$A=\begin{bmatrix} E_h & 3_h \\ 6_h & E_h \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Üretilen involutif ikili matrisin dallanma sayısı 5 ve sabit nokta sayısı 2^4'tür. İkili matrisin sabit nokta sayısının 2^4 olmasının nedeni ise 2×2 boyutundaki matrisin (*A-I*) matrisinin rank değerinin 1 olması ve ikili matrisin tasarımı için $GF(2^4)$ cisim elemanlarının kullanılması olarak gösterilebilir ($2^{4(2-1)}$). *A* matrisinin MAGMA

kodları aşağıda verilmiştir.

```
n:=8;
A:=Matrix(GF(2),n,n,[
0,1,1,1,1,0,0,1,
1,1,0,0,1,1,0,1,
1,1,1,0,0,1,1,0,
1,1,1,1,0,0,1,1,
0,0,1,1,0,1,1,1,
1,0,1,0,1,1,0,0,
1,1,0,1,1,1,1,0,
0,1,1,0,1,1,1,1]);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),A);
C:=LinearCode(B);
printf("Bn :");
MinimumWeight(C);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),Transpose(A));
C:=LinearCode(B);
printf("T Bn :");
MinimumWeight(C);
I:= IdentityMatrix(GF(2),n);
2^(n-Rank(A-I));
```

Örnek 4.2: Tablo 4.2 kullanılarak elemanları $x^4 + x^3 + 1$ ile tanımlı $GF(2^4)$ cisminden seçilmiş, 2×2 boyutundaki $\begin{bmatrix} 9_h & D_h \\ A_h & 9_h \end{bmatrix}$ matrisi 8×8 büyüklüğünde bir matrise aşağıdaki gibi dönüştürülebilir. Üretilen involutif ikili matrisin dallanma sayısı 5 ve sabit nokta sayısı 2^4'tür. İkili matrisin sabit nokta sayısının 2^4 olmasının nedeni ise 2×2 boyutundaki matrisin (A-I) matrisinin rank değerinin 1 olması ve ikili matrisin tasarımı için $GF(2^4)$ cisim elemanlarının kullanılması olarak gösterilebilir ($2^{4(2-1)}$).

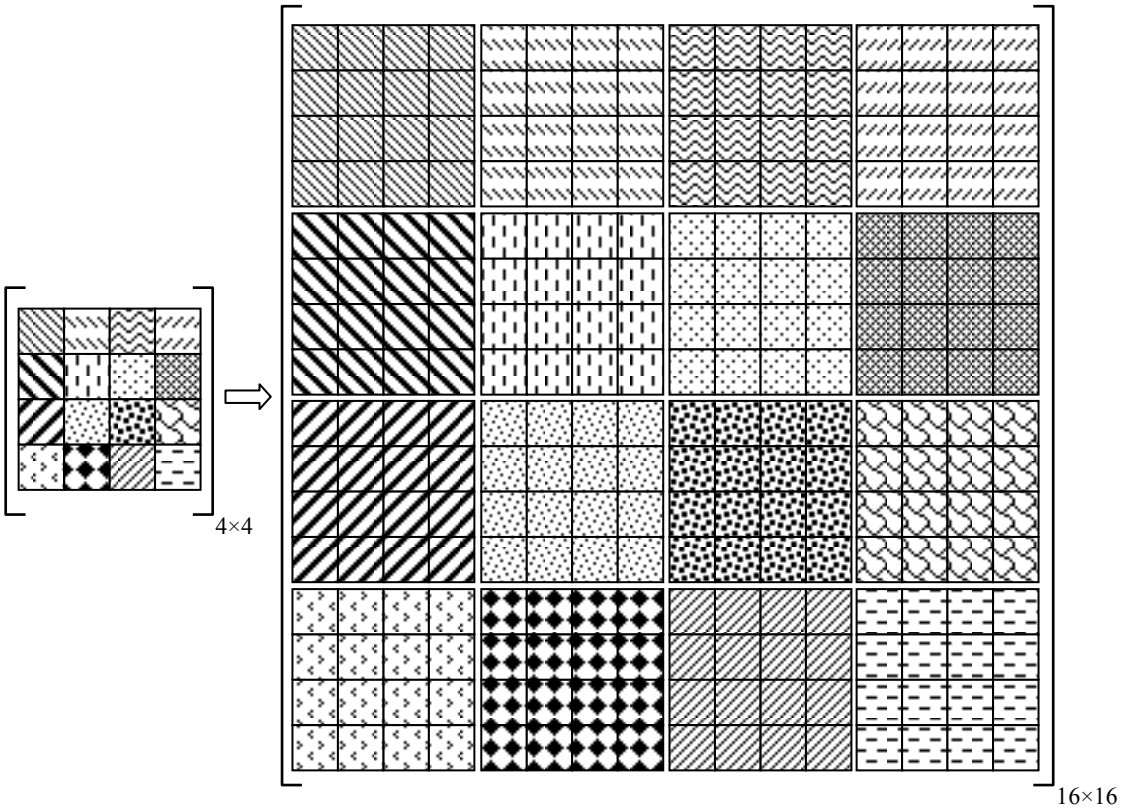
$$A = \begin{bmatrix} 9_h & D_h \\ A_h & 9_h \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Örnek 4.3: Tablo 4.3 kullanılarak, elemanları $x^4+x^3+x^2+x+1$ indirgenemez polinomu ile tanımlı $GF(2^4)$ cisminden seçilmiş, 2×2 boyutundaki $\begin{bmatrix} F_h & C_h \\ 6_h & 8_h \end{bmatrix}$ matrisi 8×8 büyüklüğünde bir matrise aşağıdaki gibi dönüştürülebilir. Üretilen involutif ikili matrisin dallanma sayısı 4 ve sabit nokta sayısı 1'dir. İkili matrisin sabit nokta sayısının 1 olmasının nedeni ise 2×2 boyutundaki matrisin (*A*-*I*) matrisinin rank değerinin 2 olması ve ikili matrisin tasarımı için $GF(2^4)$ cisim elemanlarının kullanılması olarak gösterilebilir ($2^{4(2-2)}$).

$$A=\begin{bmatrix} F_h & C_h \\ 6_h & 8_h \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

4.3. 4×4 Boyutundaki Matrislerin 16×16 Boyutundaki İkili Matrislere Dönüşümü

8×8 boyutundaki matris üretimine benzer şekilde Tablo 4.1, Tablo 4.2 veya Tablo 4.3 kullanılarak; 4×4 boyutundaki elemanları $GF(2^4)$ 'ten seçilmiş olan matrisler 16×16 boyutundaki ikili matrislere Şekil 4.2'deki gibi dönüştürülebilir.



Şekil 4.2: 4×4 Boyutundaki Matrisin 16×16 Boyutundaki Matrislere Dönüştürülmesi

Örnek 4.4: Elemanları x^4+x+1 indirgenemez polinomu ile tanımlı $GF(2^4)$ cisminden seçilmiş 4×4 boyutundaki $A=\begin{bmatrix} 1_h & 2_h & C_h & E_h \\ 2_h & 1_h & E_h & C_h \\ C_h & E_h & 1_h & 2_h \\ E_h & C_h & 2_h & 1_h \end{bmatrix}$ matrisi 16×16 büyüklüğünde bir matrise, Tablo 4.1 kullanılarak, aşağıdaki gibi dönüştürülebilir.

$$A=\begin{bmatrix} 1_h & 2_h & C_h & E_h \\ 2_h & 1_h & E_h & C_h \\ C_h & E_h & 1_h & 2_h \\ E_h & C_h & 2_h & 1_h \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Üretilen involutif ikili matrisin dallanma sayısı 8 ve sabit nokta sayısı 2^8'dir. Sabit nokta sayısının 2^8 olmasının sebebi ise 4×4 boyutundaki matrisin (*A*-*I*) matrisinin rank değerinin 2 olması ve ikili matrisin tasarımı için $GF(2^4)$ cisim elemanlarının kullanılması olarak gösterilebilir ($2^{4(4-2)}$). *A* matrisinin MAGMA kodları aşağıdaki gibidir.

```
n:=16;
A:=Matrix(GF(2),n,n,[
1,0,0,0,0,0,0,1,0,1,1,0,0,1,1,1,
0,1,0,0,1,0,0,1,0,1,0,1,1,1,0,0,
0,0,1,0,0,1,0,0,1,0,1,0,1,1,1,0,
0,0,0,1,0,0,1,0,1,1,0,1,1,1,1,1,
0,0,0,1,1,0,0,0,0,1,1,1,0,1,1,0,
1,0,0,1,0,1,0,0,1,1,0,0,0,1,0,1,
0,1,0,0,0,0,1,0,1,1,1,0,1,0,1,0,
0,0,1,0,0,0,0,1,1,1,1,1,1,1,0,1,
0,1,1,0,0,1,1,1,1,0,0,0,0,0,0,1,
0,1,0,1,1,1,0,0,0,1,0,0,1,0,0,1,
1,0,1,0,1,1,1,0,0,0,1,0,0,1,0,0,
1,1,0,1,1,1,1,1,0,0,0,1,0,0,1,0,
0,1,1,1,0,1,1,0,0,0,0,1,1,0,0,0,
1,1,0,0,0,1,0,1,1,0,0,1,0,1,0,0,
1,1,1,0,1,0,1,0,0,1,0,0,0,0,1,0,
```

```
1,1,1,1,1,1,0,1,0,0,1,0,0,0,0,1]);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),A);
C:=LinearCode(B);
printf("Bn :");
MinimumWeight(C);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),Transpose(A));
C:=LinearCode(B);
printf("T Bn :");
MinimumWeight(C);
I:= IdentityMatrix(GF(2),n);
2^(n-Rank(A-I));
```

Örnek 4.5: Elemanları $x^4 + x^3 + 1$ indirgenemez polinomu ile tanımlı $GF(2^4)$ cisminden seçilmiş, 4×4 boyutunda dairesel formdaki $A = \begin{bmatrix} 1_h & 1_h & E_h & F_h \\ F_h & 1_h & 1_h & E_h \\ E_h & F_h & 1_h & 1_h \\ 1_h & E_h & F_h & 1_h \end{bmatrix}$ matrisi 16×16 büyüklüğünde bir matrise, Tablo 4.2 kullanılarak, aşağıdaki gibi dönüştürülebilir. Üretilen involutif olmayan ikili matrisin dallanma sayısı 8 ve sabit nokta sayısı 2^4'tür. Sabit nokta sayısının 2^4 olmasının sebebi ise 4×4 boyutundaki matrisin (*A*-*I*) matrisinin rank değerinin 3 olması ve ikili matrisin tasarımı için $GF(2^4)$ cisim elemanlarının kullanılması olarak gösterilebilir ($2^{4(4-3)}$).

$$A=\begin{bmatrix} 1_h & 1_h & E_h & F_h \\ F_h & 1_h & 1_h & E_h \\ E_h & F_h & 1_h & 1_h \\ 1_h & E_h & F_h & 1_h \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Örnek 4.6: Elemanları $x^4+x^3+x^2+x+1$ ile tanımlı $GF(2^4)$ cisminden seçilmiş 4×4 boyutundaki $A=\begin{bmatrix} 1_h & 2_h & 5_h & 7_h \\ 2_h & 1_h & 7_h & 5_h \\ 5_h & 7_h & 1_h & 2_h \\ 7_h & 5_h & 2_h & 1_h \end{bmatrix}$ matrisi 16×16 büyüklüğünde bir matrise, Tablo 4.3 kullanılarak, aşağıdaki gibi dönüştürülebilir. Üretilen involutif ikili matrisin dallanma sayısı 8 ve sabit nokta sayısı 2^8'dir. Sabit nokta sayısının 2^8 olmasının sebebi ise 4×4 boyutundaki matrisin (*A*-*I*) matrisinin rank değerinin 2 olması ve ikili matrisin tasarımı için $GF(2^4)$ cisim elemanlarının kullanılması olarak gösterilebilir ($2^{4(4-2)}$).

$$A=\begin{bmatrix} 1_h & 2_h & 5_h & 7_h \\ 2_h & 1_h & 7_h & 5_h \\ 5_h & 7_h & 1_h & 2_h \\ 7_h & 5_h & 2_h & 1_h \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Tez çalışmasında kriptografik özellikleri iyi 8×8 ve 16×16 büyüklüğünde ikili matrislerin tasarımı için geliştirilen cebirsel tasarım mekanizması, 2×2 ve 4×4 büyüklüğünde, elemanları $GF(2^4)$'ten olan MDS matrisleri temel almaktadır. Bunun temel nedeni 2×2 ve 4×4 MDS matrislerin dallanma sayısı sırası ile 3 ve 5 tir. Dolayısıyla bu yüksek dallanma sayısına sahip olan matrislerin avantajını kullanarak ikili forma dönüştürülecek 8×8 ve 16×16 boyutundaki matrisler için maksimum dallanma sayısı sırası ile 5 ve 8 değerine ulaşılabilmesi amaçlanmaktadır. Bir başka deyişle tez boyunca, iyi kriptografik özelliklere sahip olan 2×2 ve 4×4 büyüklüğünde matrisler bir ***sıçrama tahtası*** gibi kullanılarak kriptografik özellikleri iyi 8×8 ve 16×16 boyutunda ikili matrisler elde edilmesi amaçlanmaktadır.

Literatürde MDS matrislerin elde edilmesi için 4 farklı temel yöntem vardır:

- Hadamard tasarım,
- Dairesel tasarım,

- Heuristik yöntemler ile düşüm maliyetli MDS matris tasarımı,
- Rastsal tasarım,
- Cauchy ve Vandermonde matrisleri (büyük boyutlu matrisler için tercih edilen yöntem).

Bu tekniklerden Hadamard yöntemi ile tasarlanan 8×8 büyüklüğünde elemanları $GF(2^8)$ 'den olan bir involutif MDS matris Khazad blok şifresinde kullanılırken dairesel yöntem ile tasarlanan 4×4 büyüklüğünde elemanları $GF(2^8)$'den olan bir MDS matris ise AES blok şifresinde kullanılmaktadır. Khazad blok şifresinde kullanılan MDS matrisin involutif olması şifreleme ve şifre çözme işlemlerinin aynı performans ile gerçekleştirilmesini sağlar. Buna ek olarak matrisin 8×8 büyüklüğünde bir MDS matrisi olması bu yayılım elemanının dallanma sayısı değerinin 9 olması anlamına gelir. Diğer yandan Hadamard tasarım yöntemiyle her satırda farklı elemanlar kullanarak 4×4 boyutunda involutif ve MDS matrisler elde edilebilirken, dairesel tasarım yöntemiyle 4×4 boyutunda matrislerin her satırında farklı elemanlar kullanılması involutif ve MDS matrislerin elde edilmesini sağlamaz.

Tanım 4.1. İki $n \times n$ ikili matris *A* ve *B,* (4.1) ifadesinde verilen denklemi sağlayacak şekilde aralarında bir satır permütasyonu ρ ve bir sütun permütasyonu γ var ise bu iki matris birbirleri ile permütasyon homomorfiktir [40].

$$\rho(\gamma(A)) = \gamma(\rho(A) = B \tag{4.1}$$

Önerme 4.1. İki *A* ve *B* matrisi birbirleri ile permütasyon homomorfik ise o zaman bu iki matris aynı dallanma sayısına sahiptir [40].

Önerme 4.1 gereği dallanma sayısı değeri herhangi bir satır ve sütun permütasyonu için aynıdır. Dolayısıyla dallanma sayısı optimal değere sahip bir ikili matristen herhangi bir satır, sütün veya bunların kombinasyonu bir permütasyon ile optimal dallanma sayısına sahip daha fazla sayıda ikili matrisler elde edilebilir.

Çalışmamızda iki özel satır ve sütun permütasyonu kullanılacaktır. Bunlar:

- $n \times n$ ikili bir matrisin dairesel olarak tüm satırlarının *l* bit ($l \in \{1, 2, ..., n-1\}$) sağa doğru döndürülmesi,

- $n \times n$ ikili bir matrisin dairesel olarak tüm sütunlarının *l* bit ($l \in \{1, 2, ..., n-1\}$) aşağıya doğru döndürülmesi.

Yukarıda verilen 1. veya 2. permütasyonlardan biri ikili doğrusal dönüşüme uygulandığında farklı sabit nokta sayısına fakat aynı dallanma sahip ikili doğrusal dönüşümler elde edilebilmektedir. Eğer ikili bir doğrusal dönüşüme 1. ve 2. permütasyonlar eşit sayıda kaydırma şeklinde uygulanır ise o zaman aynı sabit nokta ve dallanma sayısına sahip ikili doğrusal dönüşümler elde edilebilebilmektedir. Not edilmelidir ki sabit nokta sayıları ile ilgili çıkarımlar deneysel sonuçlar ile elde edilmiştir. Bu çıkarımlar Örnek 4.7 ve 4.8 ile de örneklendirilmiştir.

Örnek 4.7: Örnek 4.1'de verilen elemanları $x^4 + x + 1$ indirgenemez polinomu tabanlı $GF(2^4)$ cisminden seçilmiş 2×2 boyutundaki $A = \begin{bmatrix} E_h & 3_h \\ 6_h & E_h \end{bmatrix}$ matrisinden dönüştürülen 8×8 büyüklüğündeki matrise 1. Permütasyon uygulanarak tüm sütunlar 1 bit dairesel olarak aşağıya doğru döndürüldüğünde aşağıdaki ikili matrise ulaşılır.

$$A_{\text{1 bit sütun döndürme}} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

1 bit dairesel döndürme sonucunda elde edilen ikili matrisin dallanma sayısı 5 ve sabit nokta sayısı 1'dir.

Örnek 4.8: Örnek 4.1'de verilen elemanları $x^4 + x + 1$ indirgenemez polinomu tabanlı $GF(2^4)$ cisminden seçilmiş 2×2 boyutundaki $A = \begin{bmatrix} E_h & 3_h \\ 6_h & E_h \end{bmatrix}$ matrisinden dönüştürülen 8×8 büyüklüğündeki matrise matrise 1. Permütasyon uygulanarak tüm satırlar 1 bit dairesel olarak döndürüldüğünde ve 2. permütasyon uygulanarak tüm sütunlar 1 bit

dairesel olarak döndürüldüğünde aşağıdaki ikili matrise ulaşılır.

$$A_{\text{1 bit satır ve sütun döndürme}} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

1 bit tüm satırları dairesel döndürme ve 1 bit tüm sütunları dairesel döndürme sonucunda elde edilen involutif ikili matrisin dallanma sayısı 5 ve sabit nokta sayısı 2^4'tür.

4.4. Sonuçlar

Bu bölümde, elemanları $GF(2^4)$ cismine ait 2×2 ve 4×4 boyutunda matrisler kullanılarak sırasıyla 8×8 ve 16×16 boyutunda kriptografik özellikleri iyi ikili doğrusal dönüşümlerin tasarımı için cebirsel bir yöntem geliştirilmiştir. Geliştirilen yöntem ile hızlı ve basit bir şekilde ikili doğrusal dönüşümler üretilebilmektedir. Eğer 2×2 ve 4×4 boyutundaki matrisler kriptografik özellikleri iyi olacak şekilde dikkatli seçilir ise 8×8 ve 16×16 boyutunda kriptografik özellikleri iyi ikili matrisler elde edilebilmektedir. Diğer yandan üretilen matrislere özel satır ve sütun permütasyonları uygulanarak benzer kriptografik özelliklere sahip daha fazla sayıda matris üretilmektedir. Dolayısı ile geliştirilen yöntem kriptografik özellikleri iyi ikili matrislerin elde edilmesinde başarılıdır.

BÖLÜM 5

8×8 Boyutunda Kriptografik Özellikleri İyi İkili Doğrusal Dönüşüm Tasarımı

5.1. Giriş

Tezin bu bölümünde daha önceki bölümlerde bahsedilen kriptografik kriterler ışığında yeni tasarlanan yöntem ile 8×8 boyutunda ikili doğrusal dönüşüm tasarımı yapılacaktır.

5.2. 8×8 Boyutunda İkili Doğrusal Dönüşüm Tasarımı

Daha önceki bölümlerde geliştirilen yeni ikili doğrusal dönüşüm tasarımı ile 8×8 boyutunda matris geliştirebilmek için elemanları $GF(2^4)$ cisminden olan 2×2 boyutunda matrisler kullanılmaktadır. Elbette ki olası tüm 2×2 matrisler ile 8×8 ikili matrisler oluşturulabilir. Fakat üretilen matrislerin kriptografik özelliklerinin iyi olabilmesi için 2×2 boyutundaki matrislerin bazı kriterler doğrultusunda seçilmesi gerekmektedir.

Tez süresince elde edilen 8×8 boyutundaki ikili doğrusal dönüşümler aşağıda belirtilen kriterleri sağlayan, elemanları $GF(2^4)$ cisminden olan 2×2 boyutunda matrislerden üretilmiştir [36].

i. 2×2 Boyutundaki matris MDS olsun,
ii. 2×2 Boyutundaki matris dairesel veya Hadamard formda olsun,
iii. 2×2 Boyutundaki matristen yola çıkarak elde edilen 8×8 boyutundaki matrisin (A_{binary}) dallanma sayısı değeri 5 olsun ve ayrıca A_{binary}-I matrisinin rank değeri de 8 olsun.

Örnek 5.1: $M = circ(8_h, B_h) = \begin{bmatrix} 8_h & B_h \\ B_h & 8_h \end{bmatrix}$ 2×2 boyutunda MDS ve dairesel matris olsun. Tablo 4.1'deki (elemanları $x^4 + x + 1$ polinomu ile tanımlı $GF(2^4)$ cisminden seçilen) 8_h ve B_h sonlu cisim elemanlarına karşılık verilen 4×4 boyutundaki ikili

matrisler kullanılarak yukarıda verilen kriterleri sağlayan 8×8 boyutunda ikili matris (5.1)'de verilmektedir.

$$A=\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{5.1}$$

(5.1)'de verilen matrisin dallanma sayısı 5'tir. A_{binary}-I matrisinin rank değeri 8 olduğundan sabit nokta sayısı 1'dir. Üretilen matrisin MAGMA kodları aşağıda yer almaktadır.

```
n:=8;
A:=Matrix(GF(2),n,n,[
0,1,0,0,1,1,0,1,
0,1,1,0,1,0,1,1,
0,0,1,1,0,1,0,1,
1,0,0,1,1,0,1,0,
1,1,0,1,0,1,0,0,
1,0,1,1,0,1,1,0,
0,1,0,1,0,0,1,1,
1,0,1,0,1,0,0,1]);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),A);
C:=LinearCode(B);printf("Bn :");
MinimumWeight(C);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),Transpose(A));
C:=LinearCode(B);
printf("T Bn :");
MinimumWeight(C);
printf "Fpn:%o",2^(n-Rank(A-IdentityMatrix(GF(2),n)));
```

Örnek 5.2: Elemenları x^4+x+1 polinomu ile tanımlı $GF(2^4)$ cisminden seçilen $M=\begin{bmatrix} 6_h & 3_h \\ 3_h & F_h \end{bmatrix}$ 2×2 matrisi Tablo 4.1'deki 6_h, 3_h ve F_h sonlu cisim elemanlarına karşılık

verilen 4×4 boyutundaki ikili matrisler kullanılarak yukarıda verilen kriterleri sağlayan 8×8 boyutunda ikili matris aşağıda verilmektedir:

$$A=\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{5.2}$$

(5.2)'de verilen matrisin dallanma sayısı 5'tir. A_{binary}-I matrisinin rank değeri 8 olduğundan sabit nokta sayısı 1'dir. Üretilen matrisin MAGMA kodları aşağıda yer almaktadır.

```
n:=8;
A:=Matrix(GF(2),n,n,[
0,0,1,1,1,0,0,1,
1,0,1,0,1,1,0,1,
1,1,0,1,0,1,1,0,
0,1,1,0,0,0,1,1,
1,0,0,1,1,1,1,1,
1,1,0,1,1,0,0,0,
0,1,1,0,1,1,0,0,
0,0,1,1,1,1,1,0]);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),A);
C:=LinearCode(B);
printf("Bn :");
MinimumWeight(C);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),Transpose(A));
C:=LinearCode(B);
printf("T Bn :");
MinimumWeight(C);
printf "Fpn:%o",2^(n-Rank(A-IdentityMatrix(GF(2),n)));
```

Örnek 5.3: Elemanları x^4+x^3+1 polinomu ile tanımlı $GF(2^4)$ cisminden seçilen $M=\begin{bmatrix} F_h & D_h \\ D_h & 7_h \end{bmatrix}$ 2×2 matrisi Tablo 4.2'deki F_h, D_h ve 7_h sonlu cisim elemanlarına

karşılık verilen 4×4 boyutundaki ikili matrisler kullanılarak yukarıda verilen kriterleri sağlayan 8×8 boyutunda ikili matris aşağıda verilmektedir:

$$A=\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{5.3}$$

(5.3)'te verilen matrisin dallanma sayısı 5'tir. A_{binary}-I matrisinin rank değeri 8 olduğundan sabit nokta sayısı 1'dir. Üretilen matrisin MAGMA kodları aşağıda yer almaktadır.

```
n:=8;
A:=Matrix(GF(2),n,n,[
1,1,0,1,1,1,0,0,
1,1,1,0,0,1,1,0,
1,1,1,1,1,0,1,1,
1,0,1,0,1,0,0,1,
1,1,0,0,1,0,1,0,
0,1,1,0,1,1,0,1,
1,0,1,1,1,1,1,0,
1,0,0,1,0,1,0,1]);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),A);
C:=LinearCode(B);
printf("Bn :");
MinimumWeight(C);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),Transpose(A));
C:=LinearCode(B);
printf("T Bn :");
MinimumWeight(C);
printf "Fpn:%o",2^(n-Rank(A-IdentityMatrix(GF(2),n)));
```

5.3. 8×8 Boyutunda İnvolutif İkili Doğrusal Dönüşüm Tasarımı

8×8 boyutunda kriptografik özellikleri iyi, involutif ikili matris tasarlarken aşağıda verilen 3 kriteri sağlayan elemanları $GF(2^4)$ cisminden olan 2×2 boyutundaki matrisler aranmaktadır [36].

i. 2×2 Boyutundaki matris MDS olsun,

ii. 2×2 Boyutundaki matris involutif ve Önerme 2.3'te verildiği gibi rastsal matris olsun,

iii. 2×2 Boyutundaki matristen yola çıkarak elde edilen 8×8 boyutundaki matrisin (A_{binary}) dallanma sayısı değeri 5 olsun ve A_{binary}-I matrisinin rank değeri de 4 olsun. Bundan dolayı eğer bu 8×8 boyutundaki ikili matris 64-bit'ten 64-bit'e haritalama yapan bir doğrusal dönüşüm olarak kullanılırsa 2^{32} adet sabit nokta içerecektir.

Örnek 5.4: $M = \begin{bmatrix} E_h & 3_h \\ 6_h & E_h \end{bmatrix}$ 2×2 boyutunda (elemenları $x^4 + x + 1$ polinomu ile tanımlı $GF(2^4)$ cisminden seçilen) involutif MDS matris olsun. Tablo 4.1'deki 3_h, 6_h, E_h sonlu cisim elemanlarına karşılık verilen 4×4 boyutundaki ikili matrisler kullanılarak yukarıda verilen kriterleri sağlayan 8×8 boyutunda ikili matris (5.4)'te verilmektedir. İnvolutif olan matrisin dallanma sayısı 5'tir ve (iii) maddesinde verildiği gibi 8×8 boyutundaki ikili matris 64-bit'ten 64-bit'e haritalama yapan bir doğrusal dönüşüm olarak kullanılırsa 2^{32} ($2^{4\times 8}$) adet sabit nokta içerecektir.

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (5.4)$$

```
n:=8;
A:=Matrix(GF(2),n,n,[
0,1,1,1,1,0,0,1,
1,1,0,0,1,1,0,1,
1,1,1,0,0,1,1,0,
1,1,1,1,0,0,1,1,
0,0,1,1,0,1,1,1,
1,0,1,0,1,1,0,0,
1,1,0,1,1,1,1,0,
```

```
0,1,1,0,1,1,1,1]);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),A);
C:=LinearCode(B);
printf("Bn :");
MinimumWeight(C);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),Transpose(A));
C:=LinearCode(B);
printf("T Bn :");
MinimumWeight(C);
printf "Fpn:%o",2^(n-Rank(A-IdentityMatrix(GF(2),n)));
```

Diğer yandan, Bölüm 4'te verilen özel permütasyon 1'i veya 2'yi kullanarak dairesel olarak l bit ($l \in \{1,\ldots,7\}$) tüm satırlarının sağa doğru veya tüm sütunlarının aşağı doğru ötelenmesi ile aynı dallanma sayısına sahip ve farklı sabit nokta sayısına sahip daha fazla sayıda ikili matris elde edebiliriz. Örnek 5.4.'te verilen involutif matrisin tüm satırlarının 1 birim sağa döndürerek involutif olmayan dallanma sayısı değeri yine 5 olan ve 1 sabit nokta sayısına sahip ikili matris (5.5)'te verildiği gibi oluşturulabilir.

$$A_{(1)} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{5.5}$$

(5.5)'te verilen matrisin MAGMA kodları aşağıdaki gibidir.

```
n:=8;
A:=Matrix(GF(2),n,n,[
1,0,1,1,1,1,0,0,
1,1,1,0,0,1,1,0,
0,1,1,1,0,0,1,1,
1,1,1,1,1,0,0,1,
1,0,0,1,1,0,1,1,
0,1,0,1,0,1,1,0,
0,1,1,0,1,1,1,1,
1,0,1,1,0,1,1,1]);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),A);
```

```
C:=LinearCode(B);
printf("Bn :");
MinimumWeight(C);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),Transpose(A));
C:=LinearCode(B);
printf("T Bn :");
MinimumWeight(C);
printf "Fpn:%o",2^(n-Rank(A-IdentityMatrix(GF(2),n)));
```

Örnek 5.5: Elemanları $x^4 + x^3 + 1$ polinomu ile tanımlı $GF(2^4)$ cisminden seçilen $M = \begin{bmatrix} D_h & 7_h \\ F_h & D_h \end{bmatrix}$ 2×2 boyutunda involutif MDS matris olsun. Tablo 4.2'deki 7_h, D_h, F_h sonlu cisim elemanlarına karşılık verilen 4×4 boyutundaki ikili matrisler kullanılarak yukarıda verilen kriterleri sağlayan 8×8 boyutunda ikili matris (5.6)'da verilmektedir. İnvolutif olan matrisin dallanma sayısı 5'tir.

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{5.6}$$

(5.6)'da verilen matrisin MAGMA kodları aşağıdaki gibidir:

```
n:=8;
A:=Matrix(GF(2),n,n,[
1,1,0,0,1,0,1,0,
0,1,1,0,1,1,0,1,
1,0,1,1,1,1,1,0,
1,0,0,1,0,1,0,1,
1,1,0,1,1,1,0,0,
1,1,1,0,0,1,1,0,
1,1,1,1,1,0,1,1,
1,0,1,0,1,0,0,1]);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),A);
C:=LinearCode(B);
printf("Bn :");
MinimumWeight(C);
```

```
B:=HorizontalJoin(IdentityMatrix(GF(2),n),Transpose(A));
C:=LinearCode(B);
printf("T Bn :");
MinimumWeight(C);
printf "Fpn:%o",2^(n-Rank(A-IdentityMatrix(GF(2),n)));
```

Örnek 5.5.'te olduğu gibi özel permütasyon 1 ve 2 kullanılarak farklı ikili matrisler elde edilebilir. (5.6)'da verilen involutif matrisin tüm satırlarının 1 birim sağa döndürerek involutif olmayan dallanma sayısı değeri yine 5 olan ve 1 sabit nokta sayısına sahip ikili matris (5.7)'de verildiği gibi oluşturulabilir.

$$A=\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{5.7}$$

(5.7)'de verilen matrisin MAGMA kodları aşağıdaki gibidir.

```
n:=8;
A:=Matrix(GF(2),n,n,[
0,1,1,0,0,1,0,1,
1,0,1,1,0,1,1,0,
0,1,0,1,1,1,1,1,
1,1,0,0,1,0,1,0,
0,1,1,0,1,1,1,0,
0,1,1,1,0,0,1,1,
1,1,1,1,1,1,0,1,
1,1,0,1,0,1,0,0]);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),A);
C:=LinearCode(B);
printf("Bn :");
MinimumWeight(C);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),Transpose(A));
C:=LinearCode(B);
printf("T Bn :");
MinimumWeight(C);
printf "Fpn:%o",2^(n-Rank(A-IdentityMatrix(GF(2),n)));
```

5.4. Sonuçlar

Tez çalışmasında yukarıda anlatıldığı gibi elemanları $GF(2^4)$ cisminden olan tüm 2×2 boyutundaki matrisler taranarak 8×8 boyutunda dallanma sayısı 5 olan involutif ve involutif olmayan ikili matrisler elde edilmiştir. $GF(2^4)$ cismi 3 farklı indirgenemez polinom $(x^4+x+1,\ x^4+x^3+1$ ve $x^4+x^3+x^2+x+1)$ ile tanımlanabileceğinden, bu 3 polinom ile tanımlanan tüm cisimlerde arama yapılmıştır ve bahsedilen özelliklere sahip 2×2 boyutundaki matrisler (8×8 boyutunda dallanma sayısı 5 olan involutif ve involutif olmayan matrislere dönüştürülebilen) bölüm sonundaki Tablo 5.1, 5.2, 5.3 ve 5.4'te verilmiştir.

Tablo 5.1 ve 5.2'de verilen her matris üzerinden permütasyon 1 veya 2 ile eşleştirilebilecek olanlar çıkartıldığında 16 matris bulunmaktadır. Her bir matris 8×8 boyutundaki ikili matrislere dönüştürüldüğünde Bölüm 4'te verilen özel satır ve sütun permütasyonları uygulanarak 63 farklı matris üretilebilir. Dolayısıyla x^4+x+1 ile tanımlı $GF(2^4)$ cismi için 2×2 boyutunda matrislerden yola çıkarak bahsedilen kriterlere sahip 8×8 boyutunda ikili matris sayısı en az 1024 (16×64) olacaktır. Yine Bölüm 4'te verilen özel satır ve sütun permütasyonları uygulanarak x^4+x^3+1 ile tanımlı $GF(2^4)$ cismi için 2×2 boyutunda matrislerden yola çıkarak bahsedilen kriterlere sahip 8×8 boyutunda ikili matris sayısı en az 1024 (16×64) elde edilebilir (Tablo 5.3 ve Tablo 5.4). Fakat bu arama sırasında $x^4+x^3+x^2+x+1$ ile tanımlı $GF(2^4)$ cismi için 2×2 boyutunda matrislerden yola çıkarak bahsedilen kriterlere sahip 8×8 boyutunda ikili matrisler elde edilememiştir.

Özet olarak geliştirilen yeni yöntem ile 8×8 boyutunda dallanma sayısı 5 olan en az 2048 elde edilebilir.

Tablo 5.1: x^4+x+1 İndirgenemez Polinomuna göre 2×2 Boyutundaki İnvolutif Matrisler

$\begin{bmatrix} 3_h & 5_h \\ A_h & 3_h \end{bmatrix}$	$\begin{bmatrix} 3_h & A_h \\ 5_h & 3_h \end{bmatrix}$	$\begin{bmatrix} 3_h & 6_h \\ F_h & 3_h \end{bmatrix}$	$\begin{bmatrix} 3_h & F_h \\ 6_h & 3_h \end{bmatrix}$
$\begin{bmatrix} E_h & 2_h \\ 5_h & E_h \end{bmatrix}$	$\begin{bmatrix} E_h & 5_h \\ 2_h & E_h \end{bmatrix}$	$\begin{bmatrix} E_h & 3_h \\ 6_h & E_h \end{bmatrix}$	$\begin{bmatrix} E_h & 6_h \\ 3_h & E_h \end{bmatrix}$

Tablo 5.2: x^4+x+1 İndirgenemez Polinomuna göre 2×2 Boyutundaki İnvolutif Olmayan Matrisler

$\begin{bmatrix} 2_h & E_h \\ E_h & 5_h \end{bmatrix}$	$\begin{bmatrix} 5_h & E_h \\ E_h & 2_h \end{bmatrix}$	$\begin{bmatrix} 3_h & E_h \\ E_h & 6_h \end{bmatrix}$	$\begin{bmatrix} 5_h & 3_h \\ 3_h & A_h \end{bmatrix}$
$\begin{bmatrix} A_h & 3_h \\ 3_h & 5_h \end{bmatrix}$	$\begin{bmatrix} 6_h & E_h \\ E_h & 3_h \end{bmatrix}$	$\begin{bmatrix} 6_h & 3_h \\ 3_h & F_h \end{bmatrix}$	$\begin{bmatrix} F_h & 3_h \\ 3_h & 6_h \end{bmatrix}$
$\begin{bmatrix} 7_h & 9_h \\ 9_h & 7_h \end{bmatrix}$	$\begin{bmatrix} 9_h & 7_h \\ 7_h & 9_h \end{bmatrix}$	$\begin{bmatrix} 7_h & C_h \\ C_h & 7_h \end{bmatrix}$	$\begin{bmatrix} C_h & 7_h \\ 7_h & C_h \end{bmatrix}$
$\begin{bmatrix} 8_h & B_h \\ B_h & 8_h \end{bmatrix}$	$\begin{bmatrix} B_h & 8_h \\ 8_h & B_h \end{bmatrix}$	$\begin{bmatrix} B_h & E_h \\ E_h & B_h \end{bmatrix}$	$\begin{bmatrix} E_h & B_h \\ B_h & E_h \end{bmatrix}$

Tablo 5.3: x^4+x^3+1 İndirgenemez Polinomuna göre 2×2 Boyutundaki İnvolutif Matrisler

$\begin{bmatrix} 9_h & 7_h \\ C_h & 9_h \end{bmatrix}$	$\begin{bmatrix} 9_h & A_h \\ D_h & 9_h \end{bmatrix}$	$\begin{bmatrix} 9_h & C_h \\ 7_h & 9_h \end{bmatrix}$	$\begin{bmatrix} 9_h & D_h \\ A_h & 9_h \end{bmatrix}$
$\begin{bmatrix} D_h & 7_h \\ F_h & D_h \end{bmatrix}$	$\begin{bmatrix} D_h & 8_h \\ A_h & D_h \end{bmatrix}$	$\begin{bmatrix} D_h & A_h \\ 8_h & D_h \end{bmatrix}$	$\begin{bmatrix} D_h & F_h \\ 7_h & D_h \end{bmatrix}$

Tablo 5.4 : $x^4 + x^3 + 1$ İndirgenemez Polinomuna göre 2×2 Boyutundaki İnvolutif Olmayan Matrisler

$\begin{bmatrix} 2_h & B_h \\ B_h & 2_h \end{bmatrix}$	$\begin{bmatrix} 3_h & E_h \\ E_h & 3_h \end{bmatrix}$	$\begin{bmatrix} 5_h & B_h \\ B_h & 5_h \end{bmatrix}$	$\begin{bmatrix} 7_h & 9_h \\ 9_h & C_h \end{bmatrix}$
$\begin{bmatrix} 7_h & D_h \\ D_h & F_h \end{bmatrix}$	$\begin{bmatrix} 8_h & D_h \\ D_h & A_h \end{bmatrix}$	$\begin{bmatrix} 9_h & E_h \\ E_h & 9_h \end{bmatrix}$	$\begin{bmatrix} A_h & 9_h \\ 9_h & D_h \end{bmatrix}$
$\begin{bmatrix} A_h & D_h \\ D_h & 8_h \end{bmatrix}$	$\begin{bmatrix} B_h & 2_h \\ 2_h & B_h \end{bmatrix}$	$\begin{bmatrix} B_h & 5_h \\ 5_h & B_h \end{bmatrix}$	$\begin{bmatrix} C_h & 9_h \\ 9_h & 7_h \end{bmatrix}$
$\begin{bmatrix} D_h & 9_h \\ 9_h & A_h \end{bmatrix}$	$\begin{bmatrix} E_h & 3_h \\ 3_h & E_h \end{bmatrix}$	$\begin{bmatrix} E_h & 9_h \\ 9_h & E_h \end{bmatrix}$	$\begin{bmatrix} F_h & D_h \\ D_h & 7_h \end{bmatrix}$

BÖLÜM 6

16×16 Boyutunda Kriptografik Özellikleri İyi İkili Doğrusal Dönüşüm Tasarımı

6.1. Giriş

Tezin bu bölümünde daha önceki bölümlerde bahsedilen kriptografik kriterler ışığında yeni tasarlanan yöntem ile 16×16 boyutunda ikili doğrusal dönüşüm tasarımı yapılacaktır.

6.2. 16×16 Boyutunda İkili Doğrusal Dönüşüm Tasarımı

Daha önceki bölümlerde de bahsedildiği gibi ARIA şifreleme algoritması doğrusal dönüşüm olarak 16×16 boyutunda dallanma sayısı 8 olan ikili matris kullanmaktadır. Bu doğrusal dönüşümün, $A_{ARIA} - I$ matrisinin rank değeri 7 olduğundan dolayı, 2^{72} adet sabit noktası bulunmaktadır. Bu bölümde elemanları $GF(2^4)$'ten seçilen 4×4 boyutundaki matrisler Tablo 4.1 kullanılarak 16×16 boyutundaki ARIA blok şifresinin kullandığı doğrusal dönüşüm ile aynı dallanma sayısına sahip (dallanma sayısı 8) fakat sabit nokta sayısı 2^{64} değerine sahip (bu değer giriş vektörleri byte değerler aldığında ikili matrisin 128-bit'ten 128-bit'e dönüşüm yaptığı varsayılarak elde edilebilir) ikili matrisler elde edilecektir.

Dallanma sayısı 8 olan 16×16 boyutundaki matrisler elde edilirken seçilecek 4×4 boyutundaki matrisler için aşağıdaki kriterlerin sağlanması istenmektedir [36]:

i. 4×4 Boyutundaki matris MDS olsun,
ii. 4×4 Boyutundaki matris Hadamard formda olsun,
iii. 4×4 Boyutundaki matrisin $GF(2^4)$ cisminden seçilen elemanları ile oluşturulacak 16×16 boyutundaki ikili matrisin satır ve sütunlarının Hamming ağırlığı 7 veya 11 olsun,
iv. Üretilen A_{binary} ikili matrisinin dallanma sayısı 8 ve A_{binary}-I matrisinin rank değeri de 8 olsun.

Örnek 6.1: $M = had(1_h, 5_h, 8_h, D_h) = \begin{bmatrix} 1_h & 5_h & 8_h & D_h \\ 5_h & 1_h & D_h & 8_h \\ 8_h & D_h & 1_h & 5_h \\ D_h & 8_h & 5_h & 1_h \end{bmatrix}$ 4×4 boyutundaki matris MDS ve Hadamard formdadır. Tablo 4.1 kullanılarak üretilen 16×16 boyutundaki ikili doğrusal dönüşüm $(f_0, \ldots, f_{15}) \in GF(2)$ ve $(z_0, \ldots, z_{15}) \in GF(2)$ olmak üzere (6.1)'deki gibi yazılabilir. Verilen kriterleri sağlayan doğrusal dönüşüm incelendiğinde 16×16 boyutundaki matrisin satır ve sütunlarının Hamming ağırlığı 7 olduğu görülebilir.

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \\ z_8 \\ z_9 \\ z_{10} \\ z_{11} \\ z_{12} \\ z_{13} \\ z_{14} \\ z_{15} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \\ f_{10} \\ f_{11} \\ f_{12} \\ f_{13} \\ f_{14} \\ f_{15} \end{bmatrix} \quad (6.1)$$

(6.1)'de verilen matrisin dallanma sayısı 8'dir. Matrisin sabit nokta sayısı ise 2^8 olarak hesaplanmıştır. Üretilen matrisin MAGMA kodu aşağıdaki gibi verilebilir.

```
n:=16;
A:=Matrix(GF(2),n,n,[
1,0,0,0,1,0,1,0,0,1,0,0,1,1,1,0,
0,1,0,0,0,1,1,1,0,1,1,0,0,0,0,1,
0,0,1,0,1,0,1,1,0,0,1,1,1,0,0,0,
0,0,0,1,0,1,0,1,1,0,0,1,1,1,0,0,
1,0,1,0,1,0,0,0,1,1,1,0,0,1,0,0,
0,1,1,1,0,1,0,0,0,0,0,1,0,1,1,0,
1,0,1,1,0,0,1,0,1,0,0,0,0,0,1,1,
0,1,0,1,0,0,0,1,1,1,0,0,1,0,0,1,
```

```
0,1,0,0,1,1,1,0,1,0,0,0,1,0,1,0,
0,1,1,0,0,0,0,1,0,1,0,0,0,1,1,1,
0,0,1,1,1,0,0,0,0,0,1,0,1,0,1,1,
1,0,0,1,1,1,0,0,0,0,0,1,0,1,0,1,
1,1,1,0,0,1,0,0,1,0,1,0,1,0,0,0,
0,0,0,1,0,1,1,0,0,1,1,1,0,1,0,0,
1,0,0,0,0,0,1,1,1,0,1,1,0,0,1,0,
1,1,0,0,1,0,0,1,0,1,0,1,0,0,0,1]);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),A);
C:=LinearCode(B);
printf("Bn :");
MinimumWeight(C);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),Transpose(A));
C:=LinearCode(B);
printf("T Bn :");
MinimumWeight(C);
printf "Fpn:%o",2^(n-Rank(A-IdentityMatrix(GF(2),n)));
```

Diğer yandan (6.1) matrisi XOR işlemleri şeklinde (6.2)'de olduğu gibi yazılabilir.

$$
\begin{aligned}
z_0 &= f_0 \oplus f_4 \oplus f_6 \oplus f_9 \oplus f_{12} \oplus f_{13} \oplus f_{14} \\
z_1 &= f_1 \oplus f_5 \oplus f_6 \oplus f_7 \oplus f_9 \oplus f_{10} \oplus f_{15} \\
z_2 &= f_2 \oplus f_4 \oplus f_6 \oplus f_7 \oplus f_{10} \oplus f_{11} \oplus f_{12} \\
z_3 &= f_3 \oplus f_5 \oplus f_7 \oplus f_8 \oplus f_{11} \oplus f_{12} \oplus f_{13} \\
z_4 &= f_0 \oplus f_2 \oplus f_4 \oplus f_8 \oplus f_9 \oplus f_{10} \oplus f_{13} \\
z_5 &= f_1 \oplus f_2 \oplus f_3 \oplus f_5 \oplus f_{11} \oplus f_{13} \oplus f_{14} \\
z_6 &= f_0 \oplus f_2 \oplus f_3 \oplus f_6 \oplus f_8 \oplus f_{14} \oplus f_{15} \\
z_7 &= f_1 \oplus f_3 \oplus f_7 \oplus f_8 \oplus f_9 \oplus f_{12} \oplus f_{15} \\
z_8 &= f_1 \oplus f_4 \oplus f_5 \oplus f_6 \oplus f_8 \oplus f_{12} \oplus f_{14} \\
z_9 &= f_1 \oplus f_2 \oplus f_7 \oplus f_9 \oplus f_{13} \oplus f_{14} \oplus f_{15} \\
z_{10} &= f_2 \oplus f_3 \oplus f_4 \oplus f_{10} \oplus f_{12} \oplus f_{14} \oplus f_{15} \\
z_{11} &= f_0 \oplus f_3 \oplus f_4 \oplus f_5 \oplus f_{11} \oplus f_{13} \oplus f_{15} \\
z_{12} &= f_0 \oplus f_1 \oplus f_2 \oplus f_5 \oplus f_8 \oplus f_{10} \oplus f_{12} \\
z_{13} &= f_3 \oplus f_5 \oplus f_6 \oplus f_9 \oplus f_{10} \oplus f_{11} \oplus f_{13} \\
z_{14} &= f_0 \oplus f_6 \oplus f_7 \oplus f_8 \oplus f_{10} \oplus f_{11} \oplus f_{14} \\
z_{15} &= f_0 \oplus f_1 \oplus f_4 \oplus f_7 \oplus f_9 \oplus f_{11} \oplus f_{15}
\end{aligned}
\tag{6.2}
$$

Örnek 6.2: $M = had(B_h, E_h, 7_h, 3_h) = \begin{bmatrix} B_h & E_h & 7_h & 3_h \\ E_h & B_h & 3_h & 7_h \\ 7_h & 3_h & B_h & E_h \\ 3_h & 7_h & E_h & B_h \end{bmatrix}$ 4×4 matrisi MDS ve Hadamard formdadır. Tablo 4.1 kullanılarak üretilen 16×16 boyutundaki ikili doğrusal dönüşüm (6.3)'te verilmiştir. Verilen kriterleri sağlayan ikili matris incelendiğinde satır ve sütunlarının Hamming ağırlıklarının 11'dir.

$$M = \begin{bmatrix} 1&1&0&1&0&1&1&1&1&0&1&1&1&0&0&1 \\ 1&0&1&1&1&1&0&0&1&1&1&0&1&1&0&1 \\ 0&1&0&1&1&1&1&0&1&1&1&1&0&1&1&0 \\ 1&0&1&0&1&1&1&1&0&1&1&1&0&0&1&1 \\ 0&1&1&1&1&1&0&1&1&0&0&1&1&0&1&1 \\ 1&1&0&0&1&0&1&1&1&1&0&1&1&1&1&0 \\ 1&1&1&0&0&1&0&1&0&1&1&0&1&1&1&1 \\ 1&1&1&1&1&0&1&0&0&0&1&1&0&1&1&1 \\ 1&0&1&1&1&0&0&1&1&1&0&1&0&1&1&1 \\ 1&1&1&0&1&1&0&1&1&0&1&1&1&1&0&0 \\ 1&1&1&1&0&1&1&0&0&1&0&1&1&1&1&0 \\ 0&1&1&1&0&0&1&1&1&0&1&0&1&1&1&1 \\ 1&0&0&1&1&0&1&1&0&1&1&1&1&1&0&1 \\ 1&1&0&1&1&1&1&0&1&1&0&0&1&0&1&1 \\ 0&1&1&0&1&1&1&1&1&1&1&0&0&1&0&1 \\ 0&0&1&1&0&1&1&1&1&1&1&1&1&0&1&0 \end{bmatrix} \quad (6.3)$$

(6.3)'te verilen matrisin dallanma sayısı 8'dir. Matrisin sabit nokta sayısı ise 2^8 olarak hesaplanmıştır. Üretilen matrisin MAGMA kodu aşağıdaki gibi verilebilir.

```
n:=16;
A:=Matrix(GF(2),n,n,[
1,1,0,1,0,1,1,1,1,0,1,1,1,0,0,1,
1,0,1,1,1,1,0,0,1,1,1,0,1,1,0,1,
0,1,0,1,1,1,1,0,1,1,1,1,0,1,1,0,
1,0,1,0,1,1,1,1,0,1,1,1,0,0,1,1,
0,1,1,1,1,1,0,1,1,0,0,1,1,0,1,1,
1,1,0,0,1,0,1,1,1,1,0,1,1,1,1,0,
1,1,1,0,0,1,0,1,0,1,1,0,1,1,1,1,
1,1,1,1,1,0,1,0,0,0,1,1,0,1,1,1,
```

```
1,0,1,1,1,0,0,1,1,1,0,1,0,1,1,1,
1,1,1,0,1,1,0,1,1,0,1,1,1,1,0,0,
1,1,1,1,0,1,1,0,0,1,0,1,1,1,1,0,
0,1,1,1,0,0,1,1,1,0,1,0,1,1,1,1,
1,0,0,1,1,0,1,1,0,1,1,1,1,1,0,1,
1,1,0,1,1,1,1,0,1,1,0,0,1,0,1,1,
0,1,1,0,1,1,1,1,1,1,1,0,0,1,0,1,
0,0,1,1,0,1,1,1,1,1,1,1,1,0,1,0]);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),A);
C:=LinearCode(B);
printf("Bn :");
MinimumWeight(C);
B:=HorizontalJoin(IdentityMatrix(GF(2),n),Transpose(A));
C:=LinearCode(B);
printf("T Bn :");
MinimumWeight(C);
printf "Fpn:%o",2^(n-Rank(A-IdentityMatrix(GF(2),n)));
```

İndirgenemez polinom $x^4 + x + 1$ tabanlı $GF(2^4)$ cismi kullanılarak elde edilen tüm dallanma sayısı 8 ve *A-I* matris rank değeri 8 olan 16×16 boyutunda involutif ikili matrisler Tablo 6.1'de listelenmektedir. Bu yapılar arasında MDS matris olmayan ve örneklerde verilen Hamming ağırlığı kriterinden farklı ikili matrislerde bulunmaktadır. Diğer yandan $GF(2^4)$ cismini tanımlayabilecek diğer indirgenemez polinomlar içinde kriptografik özellikleri iyi ikili dönüşümler için arama yapılmış ve ikili matrisleri elde etme işlemi sırasında elemanları 0'dan farklı tüm 4×4 Hadamard formunda involutif matrisler önce ikili forma dönüştürülmüş ve dallanma sayısı değerleri elde edilmiştir. Dolayısıyla her indirgenemez polinom için 3164 tane 4×4 boyutunda matris toplamda ise 9492 (3×3164) matris test edilmiştir. Bu matrislerden ikili forma dönüştürülüp dallanma sayısı 8 ve sabit nokta sayısı 2^{64} (*A-I* matris rank değeri 8 olan) olanların sayısı 1824 olarak bulunmuştur. Diğer yandan bir 4×4 Hadamard formundaki $Had(a,b,c,d)$ involutif matrisin elemanlarının permütasyonu sonucu elde edilen 4×4 boyutunda 24 matristen elde edilen 16×16 boyutundaki ikili matrislerin dallanma sayısı değerinin ve sabit nokta sayısının aynı olduğu gözlenmiştir. Bu da 1824 matrisin 76 (1824/24) farklı sınıf ile listelenebilmesine olanak vermektedir. Tablo 6.1, Tablo 6.3 ve Tablo 6.5'te elde edilen 76 sınıf gösterilmiştir. Diğer bir deyişle Tablo 6.1, $x^4 + x + 1$ indirgenemez polinomu ile tanımlı $GF(2^4)$ cismi için dallanma sayısı 8 ve *A-I* matris rank değeri 8 olan 576 (24×24) tane involutif ikili matrisi temsil etmektedir.

İndirgenemez polinom x^4+x+1, x^4+x^3+1 ve $x^4+x^3+x^2+x+1$ tabanlı $GF(2^4)$ cismi ve 4×4 boyutunda dairesel matrisler kullanılarak elde edilen dallanma sayısı 8 ve *A-I* matris rank değeri 12 olan 16×16 boyutunda tüm involutif olmayan ikili matrisler sırası ile Tablo 6.2, Tablo 6.4 ve Tablo 6.6'da listelenmektedir. Bu ikili matrisler, *A-I* matris rank değeri 12 olduğu için giriş vektörleri byte (8-bit) değerler olduğunda 2^{32} sabit nokta içerecektir.

Tablo 6.1 : x^4+x+1 İndirgenemez Polinomu Kullanılarak Elde Edilen Dallanma Sayısı 8 ve *(A-I)* Matris Rank Değeri 8 Olan İnvolutif İkili Doğrusal Dönüşümlerin Sınıflandırılması

MDS Matrisler	had(1_h, 2_h, C_h, E_h)	had(3_h, 4_h, B_h,D_h)
	had(1_h, 3_h, 5_h, 6_h)	had(3_h, 5_h, 9_h, E_h)
	had(1_h, 3_h, C_h, F_h)	had(3_h, 7_h, 8_h,D_h)
	had(1_h, 3_h,D_h, E_h)	had(3_h, 7_h, A_h, F_h)
	had(1_h, 5_h, 8_h,D_h)	had(3_h, 7_h, B_h, E_h)
	had(1_h, 5_h, B_h, E_h)	had(4_h, 6_h, C_h, F_h)
	had(1_h, 7_h, B_h, C_h)	had(4_h, 7_h, C_h, E_h)
	had(2_h, 4_h, 8_h, F_h)	had(4_h, 7_h,D_h, F_h)
	had(2_h, 4_h, A_h,D_h)	had(9_h, A_h,D_h, F_h)
	had(2_h, 6_h, 8_h,D_h)	had(9_h, B_h, C_h, F_h)
	had(3_h, 4_h, A_h, C_h)	
MDS Olmayan Matrisler		had(1_h, 5_h, 9_h, C_h)
		had(5_h, 7_h, C_h, F_h)
		had(3_h, 5_h, A_h, D_h)

Tablo 6.2 : $x^4 + x + 1$ İndirgenemez Polinomuna göre Dallanma Sayısı 8 Olan İnvolutif Olmayan İkili Matrisler (Dairesel Matrisler)

MDS Matrisler	circ (1_h, 1_h, A_h, B_h)	circ (1_h, 1_h, B_h, A_h)	circ (1_h, A_h, B_h, 1_h)
	circ (1_h, B_h, A_h, 1_h)	circ (A_h, 1_h, 1_h, B_h)	circ (A_h, B_h, 1_h, 1_h)
	circ (B_h, 1_h, 1_h, A_h)	circ (B_h, A_h, 1_h, 1_h)	circ (2_h, A_h, 5_h, C_h)
	circ (2_h, C_h, 5_h, A_h)	circ (5_h, A_h, 2_h, C_h)	circ (5_h, C_h, 2_h, A_h)
	circ (A_h, 2_h, C_h, 5_h)	circ (A_h, 5_h, C_h, 2_h)	circ (C_h, 2_h, A_h, 5_h)
	circ (C_h, 5_h, A_h, 2_h)	circ (3_h, 3_h, 8_h, 9_h)	circ (3_h, 3_h, 9_h, 8_h)
	circ (3_h, 8_h, 9_h, 3_h)	circ (3_h, 9_h, 8_h, 3_h)	circ (8_h, 3_h, 3_h, 9_h)
	circ (8_h, 9_h, 3_h, 3_h)	circ (9_h, 3_h, 3_h, 8_h)	circ (9_h, 8_h, 3_h, 3_h)
	circ (3_h, B_h, 4_h, D_h)	circ (3_h, D_h, 4_h, B_h)	circ (4_h, B_h, 3_h, D_h)
	circ (4_h, D_h, 3_h, B_h)	circ (B_h, 3_h, D_h, 4_h)	circ (B_h, 4_h, D_h, 3_h)
	circ (D_h, 3_h, B_h, 4_h)	circ (D_h, 4_h, B_h, 3_h)	circ (4_h, 6_h, E_h, D_h)
	circ (4_h, D_h, E_h, 6_h)	circ (6_h, 4_h, D_h, E_h)	circ (6_h, E_h, D_h, 4_h)
	circ (D_h, 4_h, 6_h, E_h)	circ (D_h, E_h, 6_h, 4_h)	circ (E_h, 6_h, 4_h, D_h)
	circ (E_h, D_h, 4_h, 6_h)	circ (4_h, 6_h, F_h, C_h)	circ (4_h, C_h, 6_h, F_h)
	circ (4_h, C_h, F_h, 6_h)	circ (4_h, F_h, 6_h, C_h)	circ (6_h, 4_h, C_h, F_h)
	circ (6_h, C_h, 4_h, F_h)	circ (6_h, F_h, 4_h, C_h)	circ (6_h, F_h, C_h, 4_h)
	circ (C_h, 4_h, 6_h, F_h)	circ (C_h, 4_h, F_h, 6_h)	circ (C_h, 6_h, F_h, 4_h)
	circ (C_h, F_h, 6_h, 4_h)	circ (F_h, 4_h, C_h, 6_h)	circ (F_h, 6_h, 4_h, C_h)
	circ (F_h, 6_h, C_h, 4_h)	circ (F_h, C_h, 4_h, 6_h)	
MDS Olmayan Matrisler	circ (1_h, 2_h, E_h, C_h)	circ (1_h, C_h, E_h, 2_h)	circ (2_h, 1_h, C_h, E_h)
	circ (2_h, E_h, C_h, 1_h)	circ (C_h, 1_h, 2_h, E_h)	circ (C_h, E_h, 2_h, 1_h)
	circ (E_h, 2_h, 1_h, C_h)	circ (E_h, C_h, 1_h, 2_h)	circ (2_h, 4_h, D_h, A_h)
	circ (2_h, A_h, D_h, 4_h)	circ (4_h, 2_h, A_h, D_h)	circ (4_h, D_h, A_h, 2_h)
	circ (A_h, 2_h, 4_h, D_h)	circ (A_h, D_h, 4_h, 2_h)	circ (D_h, 4_h, 2_h, A_h)
	circ (D_h, A_h, 2_h, 4_h)	circ (2_h, 7_h, E_h, A_h)	circ (2_h, A_h, E_h, 7_h)
	circ (7_h, 2_h, A_h, E_h)	circ (7_h, E_h, A_h, 2_h)	circ (A_h, 2_h, 7_h, E_h)

	circ (A_h, E_h, 7_h, 2_h)	circ (E_h, 7_h, 2_h, A_h)	circ (E_h, A_h, 2_h, 7_h)
	circ (3_h, 5_h, A_h, D_h)	circ (3_h, D_h, A_h, 5_h)	circ (5_h, 3_h, D_h, A_h)
	circ (5_h, A_h, D_h, 3_h)	circ (A_h, 5_h, 3_h, D_h)	circ (A_h, D_h, 3_h, 5_h)
	circ (D_h, 3_h, 5_h, A_h)	circ (D_h, A_h, 5_h, 3_h)	circ (5_h, 7_h, C_h, F_h)
	circ (5_h, F_h, C_h, 7_h)	circ (7_h, 5_h, F_h, C_h)	circ (7_h, C_h, F_h, 5_h)
	circ (C_h, 7_h, 5_h, F_h)	circ (C_h, F_h, 5_h, 7_h)	circ (F_h, 5_h, 7_h, C_h)
	circ (F_h, C_h, 7_h, 5_h)		

Tablo 6.3: $x^4 + x^3 + 1$ İndirgenemez Polinomu Kullanılarak Elde Edilen Dallanma Sayısı 8 ve *(A-I)* Matris Rank Değeri 8 Olan İnvolutif İkili Doğrusal Dönüşümlerin Sınıflandırılması

had(1_h, 2_h, 5_h, 7_h)	had(2_h, 4_h, 8_h, F_h)	had(4_h, 6_h, D_h, E_h)
had(1_h , 3_h, 4_h, 7_h)	had(2_h, 5_h, 8_h, E_h)	had(4_h, 7_h, D_h, F_h)
had(1_h, 4_h, 9_h, D_h)	had(2_h, 7_h, 9_h, D_h)	had(5_h, 6_h, 8_h, A_h)
had(1_h, 5_h, 8_h, D_h)	had(3_h, 4_h, A_h, C_h)	had(5_h, 6_h, 9_h, B_h)
had(1_h, 5_h, 9_h, C_h)	had(3_h, 4_h, B_h, D_h)	had(5_h ,6_h, D_h, F_h)
had(1_h, 5_h, B_h, E_h)	had(3_h, 6_h, 8_h, C_h)	had(5_h, 7_h, 8_h, B_h)
had(1_h, 7_h, 9_h, E_h)	had(4_h, 6_h, 8_h, B_h)	had(8_h, B_h, D_h, F_h)
had(1_h, 7_h, A_h, D_h)	had(4_h, 6_h, C_h, F_h)	had(9_h, B_h, D_h, E_h)

Tablo 6.4: $x^4 + x^3 + 1$ İndirgenemez Polinomuna Göre Dallanma Sayısı 8 Olan İnvolutif Olmayan İkili Matrisler (Dairesel Matrisler)

circ(11_h,E_h,F_h)	circ(5_h,9_h,C_h,1_h)	circ(8_h,A_h,5_h,6_h)	circ(C_h,F_h,7_h,5_h)
circ(1_h,1_h,F_h,E_h)	circ(5_h,A_h,8_h,6_h)	circ(8_h,A_h,6_h,5_h)	circ(C_h,F_h,9_h,B_h)
circ(1_h,5_h,9_h,C_h)	circ(5_h,B_h,7_h,8_h)	circ(9_h,4_h,6_h,A_h)	circ(D_h,2_h,3_h,D_h)
circ(1_h,C_h,9_h,5_h)	circ(5_h,C_h,F_h,7_h)	circ(9_h,5_h,1_h,C_h)	circ(D_h,3_h,2_h,D_h)

circ(1_h,E_h,F_h,1_h)	circ(6_h,4_h,9_h,A_h)	circ(9_h,A_h,6_h,4_h)	circ(D_h,4_h,6_h,E_h)
circ(1_h,F_h,E_h,1_h)	circ(6_h,4_h,D_h,E_h)	circ(9_h,B_h,C_h,F_h)	circ(D_h,4_h,F_h,7_h)
circ(2_h,3_h,D_h,D_h)	circ(6_h,4_h,F_h,C_h)	circ(9_h,C_h,1_h,5_h)	circ(D_h,7_h,F_h,4_h)
circ(2_h,D_h,D_h,3_h)	circ(6_h,5_h,8_h,A_h)	circ(9_h,F_h,C_h,B_h)	circ(D_h,D_h,2_h,3_h)
circ(3_h,2_h,D_h,D_h)	circ(6_h,5_h,A_h,8_h)	circ(A_h,5_h,6_h,8_h)	circ(D_h,D_h,3_h,2_h)
circ(3_h,D_h,D_h,2_h)	circ(6_h,8_h,A_h,5_h)	circ(A_h,6_h,4_h,9_h)	circ(D_h,E_h,6_h,4_h)
circ(4_h,6_h,A_h,9_h)	circ(6_h,A_h,8_h,5_h)	circ(A_h,6_h,5_h,8_h)	circ(E_h,1_h,1_h,F_h)
circ(4_h,6_h,C_h,F_h)	circ(6_h,A_h,9_h,4_h)	circ(A_h,8_h,5_h,6_h)	circ(E_h,6_h,4_h,D_h)
circ(4_h,6_h,E_h,D_h)	circ(6_h,C_h,F_h,4_h)	circ(A_h,8_h,6_h,5_h)	circ(E_h,D_h,4_h,6_h)
circ(4_h,9_h,A_h,6_h)	circ(6_h,E_h,D_h,4_h)	circ(A_h,9_h,4_h,6_h)	circ(E_h,F_h,1_h,1_h)
circ(4_h,D_h,7_h,F_h)	circ(7_h,5_h,C_h,F_h)	circ(B_h,5_h,8_h,7_h)	circ(F_h,1_h,1_h,E_h)
circ(4_h,D_h,E_h,6_h)	circ(7_h,8_h,5_h,B_h)	circ(B_h,7_h,8_h,5_h)	circ(F_h,4_h,6_h,C_h)
circ(4_h,F_h,7_h,D_h)	circ(7_h,B_h,5_h,8_h)	circ(B_h,9_h,F_h,C_h)	circ(F_h,4_h,D_h,7_h)
circ(4_h,F_h,C_h,6_h)	circ(7_h,D_h,4_h,F_h)	circ(B_h,C_h,F_h,9_h)	circ(F_h,7_h,5_h,C_h)
circ(5_h,1_h,C_h,9_h)	circ(7_h,F_h,4_h,D_h)	circ(C_h,1_h,5_h,9_h)	circ(F_h,7_h,D_h,4_h)
circ(5_h,6_h,8_h,A_h)	circ(7_h,F_h,C_h,5_h)	circ(C_h,5_h,7_h,F_h)	circ(F_h,9_h,B_h,C_h)
circ(5_h,6_h,A_h,8_h)	circ(8_h,5_h,6_h,A_h)	circ(C_h,6_h,4_h,F_h)	circ(F_h,C_h,5_h,7_h)
circ(5_h,7_h,F_h,C_h)	circ(8_h,5_h,B_h,7_h)	circ(C_h,9_h,5_h,1_h)	circ(F_h,C_h,6_h,4_h)
circ(5_h,8_h,7_h,B_h)	circ(8_h,6_h,5_h,A_h)	circ(C_h,B_h,9_h,F_h)	circ(F_h,C_h,B_h,9_h)
circ(5_h,8_h,A_h,6_h)	circ(8_h,7_h,B_h,5_h)	circ(C_h,F_h,4_h,6_h)	circ(F_h,E_h,1_h,1_h)

Tablo 6.5: $x^4+x^3+x^2+x+1$ İndirgenemez Polinomu Kullanılarak Elde Edilen Dallanma Sayısı 8 ve *(A-I)* Matris Rank Değeri 8 Olan İnvolutif İkili Doğrusal Dönüşümlerin Sınıflandırılması

had(1_h, 2_h, 5_h, 7_h)	had(2_h, 5_h, 8_h, E_h)	had(4_h ,6_h, 8_h, B_h)
had(1_h , 3_h, 5_h, 6_h)	had(2_h, 5_h, A_h, C_h)	had(4_h, 6_h, C_h, F_h)
had(1_h, 3_h, 8_h, B_h)	had(2_h, 6_h, A_h, F_h)	had(4_h, 6_h, D_h, E_h)
had(1_h, 3_h, 9_h, A_h)	had(2_h, 7_h, 8_h, C_h)	had(1_h, 2_h, 4_h, A_h)
had(1_h, 4_h, A_h, E_h)	had(2_h, 7_h, B_h, F_h)	had(1_h ,2_h, 4_h, A_h)
had(1_h, 5, B_h, E_h)	had(3_h, 4_h, 9_h, F_h)	had(1_h, 2_h, 4_h, A_h)
had(1_h, 6_h, 9_h, F_h)	had(3_h, 5_h, 8_h, F_h)	had(1_h, 2_h, 4_h, A_h)
had(1_h, 7_h, 9_h, E_h)	had(3_h, 6_h, A_h, E_h)	had(1_h, 2_h, 4_h, A_h)
had(2_h, 4_h, 9_h, E_h)	had(3_h, 7_h, 8_h, D_h)	
had(2_h, 4_h, A_h, D_h)	had(3_h, 7_h, B_h, E_h)	

Tablo 6.6: $x^4+x^3+x^2+x+1$ İndirgenemez Polinomuna Göre Dallanma Sayısı 8 Olan İnvolutif Olmayan İkili Matrisler (Dairesel Matrisler)

circ(1_h,3_h,6_h,5_h)	circ(4_h,A_h,3_h,C_h)	circ(8_h,B_h,E_h,C_h)	circ(C_h,3_h,A_h,4_h)
circ(1_h,3_h,A_h,9_h)	circ(4_h,C_h,3_h,A_h)	circ(8_h,C_h,E_h,B_h)	circ(C_h,4_h,A_h,3_h)
circ(1_h,5_h,6_h,3_h)	circ(5_h,1_h,3_h,6_h)	circ(9_h,1_h,3_h,A_h)	circ(C_h,8_h,B_h,E_h)
circ(1_h,5_h,B_h,E_h)	circ(5_h,1_h,E_h,B_h)	circ(9_h,1_h,E_h,7_h)	circ(C_h,E_h,B_h,8_h)
circ(1_h,9_h,7_h,E_h)	circ(5_h,6_h,3_h,1_h)	circ(9_h,7_h,2_h,D_h)	circ(D_h,2_h,7_h,9_h)
circ(1_h,9_h,A_h,3_h)	circ(5_h,6_h,F_h,D_h)	circ(9_h,7_h,E_h,1_h)	circ(D_h,5_h,6_h,F_h)
circ(1_h,E_h,7_h,9_h)	circ(5_h,B_h,E_h,1_h)	circ(9_h,A_h,3_h,1_h)	circ(D_h,9_h,7_h,2_h)
circ(1_h,E_h,B_h,5_h)	circ(5_h,D_h,F_h,6_h)	circ(9_h,D_h,2_h,7_h)	circ(D_h,F_h,6_h,5_h)
circ(2_h,7_h,9_h,D_h)	circ(6_h,3_h,1_h,5_h)	circ(A_h,3_h,1_h,9_h)	circ(E_h,1_h,5_h,B_h)

$circ(2_h,D_h,9_h,7_h)$	$circ(6_h,5_h,1_h,3_h)$	$circ(A_h,3_h,C_h,4_h)$	$circ(E_h,1_h,9_h,7_h)$
$circ(3_h,1_h,5_h,6_h)$	$circ(6_h,5_h,D_h,F_h)$	$circ(A_h,4_h,C_h,3_h)$	$circ(E_h,7_h,9_h,1_h)$
$circ(3_h,1_h,9_h,A_h)$	$circ(6_h,F_h,D_h,5_h)$	$circ(A_h,9_h,1_h,3_h)$	$circ(E_h,B_h,5_h,1_h)$
$circ(3_h,6_h,5_h,1_h)$	$circ(7_h,2_h,D_h,9_h)$	$circ(B_h,5_h,1_h,E_h)$	$circ(E_h,B_h,8_h,C_h)$
$circ(3_h,A_h,4_h,C_h)$	$circ(7_h,9_h,1_h,E_h)$	$circ(B_h,8_h,C_h,E_h)$	$circ(E_h,C_h,8_h,B_h)$
$circ(3_h,A_h,9_h,1_h)$	$circ(7_h,9_h,D_h,2_h)$	$circ(B_h,E_h,1_h,5_h)$	$circ(F_h,6_h,5_h,D_h)$
$circ(3_h,C_h,4_h,A_h)$	$circ(7_h,E_h,1_h,9_h)$	$circ(B_h,E_h,C_h,8_h)$	$circ(F_h,D_h,5_h,6_h)$

6.3. Sonuçlar

Tezin bu bölümünde dallanma sayısı 8 olan 16×16 boyutunda (en düşük 2^8-involutif değil, en yüksek 2^{64} sabit nokta-involutif sayısı içeren) involutif ve involutif olmayan ikili matrislerin elde edilmesi için cebirsel bir yöntem sunulmuştur. Not edilmelidir ki verilen sabit nokta sayısı, giriş vektörleri byte (8-bit) değerler olduklarında geçerlidir. Elde edilen involutif 16×16 boyutunda ikili doğrusal dönüşümler, ARIA blok şifresinin kullandığı 16×16 boyutunda ikili doğrusal dönüşümden 2^8 kat daha az sabit nokta sayısı içermektedir. Buna ek olarak dallanma sayısı değeri 8 olan ve 2^8 sabit nokta sayısına sahip involutif olmayan yapılar da involutif ve dallanma sayısı 8 olan ikili matrisler ile özel permütasyonların kullanımı ile elde edilebilir. Ancak sabit nokta sayısı 1 olan ve dallanma sayısı 8 olan 16×16 boyutunda involutif ya da involutif olmayan ikili doğrusal dönüşümler elde edilememiştir. Bahsedilen kriptografik özelliklere sahip ikili matrisler tablolarda verilmiştir. 16×16 boyutunda dallanma sayısı 8 ve aynı sabit nokta sayısına sahip ikili matrislerin sayısının özel satır ve sütun permütasyonlarının beraber uygulanması ile arttırılabileceği gözlenmiştir. Örneğin, 16×16 boyutunda dallanma sayısı 8 ve sabit nokta sayısı 2^{64} olan involutif bir ikili matristen verilen özel satır ve sütun permütasyonlarının beraber uygulanması ile 15 tane daha aynı özellikte kriptografik dönüşüm elde edilebilmektedir. Bu da sunulan cebirsel yöntemle dallanma sayısı 8 ve sabit nokta sayısı 2^{64} olan toplamda 29184 (16×1824) tane involutif ikili matrisin üretilebileceğini göstermektedir. Bu ikili matrisler, bir blok şifre tasarımında belli bir döngü sayısı ve AES S-kutusu ile aynı kriptografik özelliklere sahip S-kutuları ile

kullanıldığında doğrusal ve diferansiyel kriptanalize karşı dayanıklıdır. Ancak bu kriptografik dönüşümlerin bir blok şifre tasarımında kullanmadan önce imkânsız diferansiyel kriptanaliz (impossible differential cryptanalysis), kesilmiş diferansiyel kriptanaliz (truncated differential cryptanalysis) gibi önemli saldırılara karşı dayanıklılıklarının test edilmesi tavsiye edilir.

BÖLÜM 7

16×16 Boyutunda Dallanma Sayısı 7 ve Sabit Nokta Sayısı 1 olan İkili Doğrusal Dönüşüm Tasarımı

7.1. Giriş

Tez çalışmasında geliştirilen cebirsel yöntem ile dallanma sayısı 5 ve 8 olan sırasıyla 8×8 ve 16×16 boyutunda involutif ve involutif olmayan ikili matrisler elde edilmiştir. Üretilen bu matrislere giriş vektörü byte değerlerden oluşursa bu ikili matrislerin sahip olduğu sabit nokta sayıları aşağıda verilebilir:

- 8×8 boyutunda involutif matrisler için 2^{32} adet sabit nokta $(rank(A-I)=4)$,
- 8×8 boyutunda involutif olmayan matrisler için 1 adet sabit nokta $(rank(A-I)=8)$,
- 16×16 boyutunda involutif matrisler için 2^{64} adet sabit nokta $(rank(A-I)=8)$,
- 16×16 boyutunda involutif olmayan matrisler için 2^{8} adet sabit nokta $(rank(A-I)=15)$.

ARIA blok şifresi dallanma sayısı 8 olan ve 2^{72} sabit nokta sayısına sahip 16×16 boyutunda involutif ikili bir doğrusal dönüşümü yayılım elemanı olarak kullanmaktadır. Tezde geliştirilen involutif ikili doğrusal dönüşümlerin tümü dallanma sayısı 8 olan ve 2^{64} sabit nokta sayısı içeren yapılardır. Bu doğrusal dönüşümler 4×4 boyutunda involutif MDS matrislerin 16×16 boyutunda ikili forma dönüştürülmesi ile elde edilmektedir. Buna ek olarak involutif olan yapıların elde edilmesi için Hadamard tasarım, involutif olmayan yapıların elde edilmesi için Hadamard ve dairesel tasarım kullanılmıştır. Bu tasarımlar sırasında verilen kriterlere göre elde edilen 3 farklı tip doğrusal dönüşüm aşağıdaki gibi verilebilir:

- (Tip 1) Hadamard tasarım ile elde edilen dallanma sayısı 8 olan ve 2^{64} sabit nokta sayısına sahip involutif yapılar (Örnek 7.1),
- (Tip 2) Hadamard tasarım ile dallanma sayısı 8 olan involutif yapılara uygulanan Bölüm 2'te verilen özel permütasyonlar kullanılarak elde edilen 2^8 sabit nokta sayısına sahip involutif olmayan yapılar (Örnek 7.2),
- (Tip 3) Dairesel tasarım ile elde edilen dallanma sayısı 8 ve sabit nokta sayısı 2^{32} olan ($A-I$ matrisinin rank değeri 12) involutif olmayan yapılar (Örnek 7.3).

Örnek 7.1: $A = Had(2_h, 4_h, 8_h, F_h) = \begin{bmatrix} 2_h & 4_h & 8_h & F_h \\ 4_h & 2_h & F_h & 8_h \\ 8_h & F_h & 2_h & 4_h \\ F_h & 8_h & 4_h & 2_h \end{bmatrix}_{4\times 4}$ matrisi indirgenemez polinom $x^4 + x + 1$ ile tanımlı elemanları $GF(2^4)$ cisminden seçilen elde edilen A matrisi Tablo 4.1 kullanılarak aşağıda verilen 16×16 boyutundaki ikili matrise dönüştürülebilir ve Tip 1'e örnek olarak verilebilir.

$$A = \begin{bmatrix}
0&0&0&1&0&0&1&0&0&1&0&0&1&1&1&1\\
1&0&0&1&0&0&1&1&0&1&1&0&1&0&0&0\\
0&1&0&0&1&0&0&1&0&0&1&1&1&1&0&0\\
0&0&1&0&0&1&0&0&1&0&0&1&1&1&1&0\\
0&0&1&0&0&0&0&1&1&1&1&1&0&1&0&0\\
0&0&1&1&1&0&0&1&1&0&0&0&0&1&1&0\\
1&0&0&1&0&1&0&0&1&1&0&0&0&0&1&1\\
0&1&0&0&0&0&1&0&1&1&1&0&1&0&0&1\\
0&1&0&0&1&1&1&1&0&0&0&1&0&0&1&0\\
0&1&1&0&1&0&0&0&1&0&0&1&0&0&1&1\\
0&0&1&1&1&1&0&0&0&1&0&0&1&0&0&1\\
1&0&0&1&1&1&1&0&0&0&1&0&0&1&0&0\\
1&1&1&1&0&1&0&0&0&0&1&0&0&0&0&1\\
1&0&0&0&0&1&1&0&0&0&1&1&1&0&0&1\\
1&1&0&0&0&0&1&1&1&0&0&1&0&1&0&0\\
1&1&1&0&1&0&0&1&0&1&0&0&0&0&1&0
\end{bmatrix}$$

Örnek 7.2: Örnek 7.1'de verilen 16×16 boyutundaki A matrisine özel satır ve sütun permütasyonları uygulanarak elde edilen B matrisi Tip 2'ye örnek olarak verilebilir.

$$B=\begin{bmatrix}
1&0&0&0&1&0&0&1&0&0&1&0&0&1&1&1\\
0&1&0&0&1&0&0&1&1&0&1&1&0&1&0&0\\
0&0&1&0&0&1&0&0&1&0&0&1&1&1&1&0\\
0&0&0&1&0&0&1&0&0&1&0&0&1&1&1&1\\
0&0&0&1&0&0&0&0&1&1&1&1&1&0&1&0\\
0&0&0&1&1&1&0&0&1&1&0&0&0&0&1&1\\
1&1&0&0&1&0&1&0&0&1&1&0&0&0&0&1\\
1&0&1&0&0&0&0&1&0&1&1&1&0&1&0&0\\
0&0&1&0&0&1&1&1&1&0&0&0&1&0&0&1\\
1&0&1&1&0&1&0&0&0&1&0&0&1&0&0&1\\
1&0&0&1&1&1&1&0&0&0&1&0&0&1&0&0\\
0&1&0&0&1&1&1&1&0&0&0&1&0&0&1&0\\
1&1&1&1&1&0&1&0&0&0&0&1&0&0&0&0\\
1&1&0&0&0&0&1&1&0&0&0&1&1&1&0&0\\
0&1&1&0&0&0&0&1&1&1&0&0&1&0&1&0\\
0&1&1&1&0&1&0&0&1&0&1&0&0&0&0&1
\end{bmatrix}$$

Örnek 7.3: $A=circ(1_h,1_h,A_h,B_h)=\begin{bmatrix}1_h&1_h&A_h&B_h\\B_h&1_h&1_h&A_h\\A_h&B_h&1_h&1_h\\1_h&A_h&B_h&1_h\end{bmatrix}_{4\times4}$ dairesel formdaki 4×4 MDS matrisinden elde edilen 16×16 boyutunda involutif olmayan A ikili matrisi Tip 3'e örnek olarak verilebilir.

$$A=\begin{bmatrix}
1&0&0&0&1&0&0&0&0&1&0&1&1&1&0&1\\
0&1&0&0&0&1&0&0&1&1&1&1&1&0&1&1\\
0&0&1&0&0&0&1&0&0&1&1&1&0&1&0&1\\
0&0&0&1&0&0&0&1&1&0&1&1&1&0&1&0\\
1&1&0&1&1&0&0&0&1&0&0&0&0&1&0&1\\
1&0&1&1&0&1&0&0&0&1&0&0&1&1&1&1\\
0&1&0&1&0&0&1&0&0&0&1&0&0&1&1&1\\
1&0&1&0&0&0&0&1&0&0&0&1&1&0&1&1\\
0&1&0&1&1&1&0&1&1&0&0&0&1&0&0&0\\
1&1&1&1&1&0&1&1&0&1&0&0&0&1&0&0\\
0&1&1&1&0&1&0&1&0&0&1&0&0&0&1&0\\
1&0&1&1&1&0&1&0&0&0&0&1&0&0&0&1\\
1&0&0&0&0&1&0&1&1&1&0&1&1&0&0&0\\
0&1&0&0&1&1&1&1&1&0&1&1&0&1&0&0\\
0&0&1&0&0&1&1&1&0&1&0&1&0&0&1&0\\
0&0&0&1&1&0&1&1&1&0&1&0&0&0&0&1
\end{bmatrix}$$

7.2. 16×16 Boyutunda Dallanma Sayısı 7 ve Sabit Nokta Sayısı 1 Olan İkili Doğrusal Dönüşüm Tasarımı

Tezin bu bölümünde elemanları $GF(2^4)$'ten seçilen 4×4 boyutundaki matrisler Tablo 4.1, Tablo 4.2 ve Tablo 4.3 kullanılarak 16×16 boyutunda yüksek dallanma sayısına ve 1 adet sabit noktaya sahip ikili doğrusal dönüşümler tasarlanmıştır.

Çalışmamızda geliştirilen yöntem 1 sabit noktaya sahip ve maksimum dallanma sayısı olan 8 olan 16×16 boyutunda matris elde edilmesi için uygulanmıştır. Ancak bu tip doğrusal dönüşümler elde edilememiştir. Dolayısıyla dallanma sayısı 1 düşük değere sahip doğrusal dönüşümler araştırılmıştır. Sonuçta dallanma sayısı 7 ve sabit nokta sayısı 1 olan 16×16 boyutunda doğrusal dönüşümler elde edilmiştir. Bu tip yapıları ***yaklaşık MDBL kodlar*** olarak isimlendirebiliriz.

Herhangi bir 16×16 boyutundaki matrisin sabit nokta sayısının 1 olması isteniyor ise bu matris için $A-I$ matrisinin rank değeri 16 olmalıdır ve bu değer daha önce verilen (2.16) ifadesinden elde edilebilir.

Örnek 7.4 ve Örnek 7.5 için dallanma sayısı 7 olan 16×16 boyutundaki matrisler elde edilirken seçilecek 4×4 boyutundaki matrisler için aşağıdaki kriterleri sağlaması istenmektedir:

i. 4×4 Boyutundaki matris MDS olsun,

ii. 4×4 Boyutundaki matris Hadamard veya Dairesel formda olsun,

iii. 4×4 Boyutundaki matrisin $GF(2^4)$ cisminden seçilen elemanları ile oluşturulacak 16×16 boyutundaki ikili matrisin Hamming ağırlığı 120 olsun,

iv. Üretilen A_{binary} ikili matrisinin dallanma sayısı 7 ve A_{binary}-I matrisinin rank değeri de 16 olsun.

Bu kriterlerden (ii) ve (iv) tüm üretilen matrisler için, (i) ve (iii) ise aşağıda verilen örnekler için geçerlidir. Kriter (iii) örneklerde verildiği gibi 16×16 boyutundaki ikili doğrusal dönüşümlerin 8-bit işlemcilerde uygulamaları için düşünülmüştür.

Örnek 7.4: $M = had(8_h, 1_h, B_h, C_h) = \begin{bmatrix} 8_h & 1_h & B_h & C_h \\ 1_h & 8_h & C_h & B_h \\ B_h & C_h & 8_h & 1_h \\ C_h & B_h & 1_h & 8_h \end{bmatrix}$ 4×4 boyutundaki matris MDS ve Hadamard formdadır. Tablo 4.1 kullanılarak üretilen 16×16 boyutundaki ikili doğrusal dönüşüm $(f_0, \ldots, f_{15}) \in GF(2)$ ve $(z_0, \ldots, z_{15}) \in GF(2)$ olmak üzere aşağıdaki gibi yazılabilir.

$$M = \begin{bmatrix}
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1
\end{bmatrix}$$

Örnek 7.5: $M = circ(1_h, 8_h, B_h, C_h) = \begin{bmatrix} 1_h & 8_h & B_h & C_h \\ C_h & 1_h & 8_h & B_h \\ B_h & C_h & 1_h & 8_h \\ 8_h & B_h & C_h & 1_h \end{bmatrix}$ 4×4 matrisi MDS ve dairesel formdadır. Tablo 4.1 kullanılarak üretilen 16×16 boyutundaki ikili doğrusal dönüşüm aşağıda verilmiştir.

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

7.3.Sonuçlar

Tezin bu bölümünde Örnek 7.4 ve Örnek 7.5 de verilen matrislerin sahip olduğu kriterlerden i, ii ve iv kriterlerine uygun matrislerin (GF(2^4)'te tanımlı üç indirgenemez polinom için) tümü taranmıştır. Elde edilen sonuçlar Tablo 7.1'de verilmiştir.

Tablo 7.1: Üretilen 1 Sabit Noktalı Dallanma Sayısı 7 Olan 16×16 Boyutundaki Matris Sayıları

İndirgenemez Polinom	Matris Tipi	MDS Matris Sayısı	MDS olmayan Matris Sayısı
x^4+x+1	Hadamard	1584	0
x^4+x+1	Dairesel	1888	856
x^4+x^3+1	Hadamard	1584	0
x^4+x^3+1	Dairesel	1888	856

$x^4+x^3+x^2+x+1$	Hadamard	-	-
$x^4+x^3+x^2+x+1$	Dairesel	96	-
Toplam Matris Sayısı		7040	1712

Tablo 7.1'de görüldüğü gibi x^4+x+1 indirgenemez polinomu ile 1584 adet Hadamard formda dallanma sayısı 7 ve 1 sabit noktaya sahip 16×16 boyutunda ikili matris üretilmiştir. Aynı zamanda bu matrisler, yapılarına göre içerisinde 24 eleman olan 66 sınıfta kategorize edilmiştir. Bu sınıflar Tablo 7.2'de verilmiştir. Tablo 7.3'te ise had(1_h;2_h;4_h;E_h) sınıfının (örnek sınıf) içerisinde bulunan 24 eleman verilmektedir.

Tablo 7.2: 1 Sabit Noktalı Dallanma Sayısı 7 Olan 16×16 Boyutundaki Matrise Dönüştürülebilecek x^4+x+1 İndirgenemez Polinomu ile Tanımlı Olan 4×4 Boyutundaki Matris Sınıfları

1-) had(1_h,2_h,4_h,E_h)	23-) had(2_h,4_h,D_h,E_h)	45-) had(5_h,7_h,A_h,D_h)
2-) had(1_h,2_h,6_h,A_h)	24-) had(2_h,5_h,6_h,A_h)	46-) had(5_h,8_h,A_h,E_h)
3-) had(1_h,2_h,6_h,B_h)	25-) had(2_h,5_h,B_h,E_h)	47-) had(5_h,8_h,A_h,F_h)
4-) had(1_h,2_h,7_h,9_h)	26-) had(2_h,5_h,C_h,E_h)	48-) had(5_h,9_h,B_h,F_h)
5-) had(1_h,2_h,7_h,F_h)	27-) had(2_h,6_h,B_h,C_h)	49-) had(5_h,9_h,D_h,F_h)
6-) had(1_h,3_h,4_h,E_h)	28-) had(2_h,7_h,8_h,9_h)	50-) had(5_h,A_h,B_h,F_h)
7-) had(1_h,3_h,6_h,C_h)	29-) had(3_h,4_h,8_h,C_h)	51-) had(5_h,B_h,C_h,E_h)
8-) had(1_h,5_h,9_h,E_h)	30-) had(3_h,4_h,8_h,D_h)	52-) had(6_h,7_h,9_h,C_h)
9-) had(1_h,5_h,A_h,D_h)	31-) had(3_h,5_h,C_h,F_h)	53-) had(6_h,7_h,9_h,F_h)
10-) had(1_h,6_h,7_h,B_h)	32-) had(3_h,5_h,D_h,E_h)	54-) had(6_h,7_h,D_h,E_h)
11-) had(1_h,6_h,9_h,B_h)	33-) had(3_h,6_h,8_h,9_h)	55-) had(6_h,8_h,B_h,E_h)
12-) had(1_h,6_h,A_h,B_h)	34-) had(3_h,7_h,9_h,E_h)	56-) had(6_h,9_h,A_h,C_h)

13-) had(1_h,6_h,A_h,F_h)	35-) had(3_h,7_h,A_h,D_h)	57-) had(6_h,A_h,D_h,E_h)
14-) had(1_h,7_h,D_h,E_h)	36-) had(3_h,8_h,B_h,F_h)	58-) had(7_h,8_h,A_h,C_h)
15-) had(1_h,8_h,B_h,C_h)	37-) had(3_h,9_h,C_h,D_h)	59-) had(7_h,8_h,A_h,D_h)
16-) had(1_h,9_h,A_h,D_h)	38-) had(4_h,6_h,8_h,F_h)	60-) had(7_h,8_h,B_h,D_h)
17-) had(1_h,B_h,C_h,E_h)	39-) had(4_h,7_h,A_h,C_h)	61-) had(7_h,8_h,B_h,F_h)
18-) had(2_h,3_h,4_h,E_h)	40-) had(4_h,8_h,C_h,E_h)	62-) had(7_h,9_h,A_h,E_h)
19-) had(2_h,3_h,5_h,B_h)	41-) had(4_h,8_h,C_h,F_h)	63-) had(7_h,9_h,A_h,F_h)
20-) had(2_h,4_h,5_h,C_h)	42-) had(4_h,A_h,B_h,E_h)	64-) had(7_h,9_h,B_h,D_h)
21-) had(2_h,4_h,6_h,E_h)	43-) had(5_h,6_h,E_h,F_h)	65-) had(7_h,9_h,C_h,F_h)
22-) had(2_h,4_h,7_h,C_h)	44-) had(5_h,7_h,8_h,F_h)	66-) had(7_h,A_h,D_h,E_h)

Tablo 7.3: had(1_h,2_h,4_h,E_h) Sınıfının Elemanları

1-)had(1_h,2_h,4_h,E_h)	9-)had(2_h,4_h,1_h,E_h)	17-)had(4_h,E_h,1_h,2_h)
2-)had(1_h,2_h,E_h,4_h)	10-)had(2_h,4_h,E_h,1_h)	18-)had(4_h,E_h,2_h,1_h)
3-)had(1_h,4_h,2_h,E_h)	11-)had(2_h,E_h,1_h,4_h)	19-)had(E_h,1_h,2_h,4_h)
4-)had(1_h,4_h,E_h,2_h)	12-)had(2_h,E_h,4_h,1_h)	20-)had(E_h,1_h,4_h,2_h)
5-)had(1_h,E_h,2_h,4_h)	13-)had(4_h,1_h,2_h,E_h)	21-)had(E_h,2_h,1_h,4_h)
6-)had(1_h,E_h,4_h,2_h)	14-)had(4_h,1_h,E_h,2_h)	22-)had(E_h,2_h,4_h,1_h)
7-)had(2_h,1_h,4_h,E_h)	15-)had(4_h,2_h,1_h,E_h)	23-)had(E_h,4_h,1_h,2_h)
8-)had(2_h,1_h,E_h,4_h)	16-)had(4_h,2_h,E_h,1_h)	24-)had(E_h,4_h,2_h,1_h)

Tez çalışmasında x^4+x^3+1 ve $x^4+x^3+x^2+x+1$ indirgenemez polinomlarında tanımlanabilecek Hadamard formdaki matrisler için tüm uzayda araştırma yapılmıştır. Buna göre x^4+x^3+1 polinomu için 66 sınıf elde edilmiş ve bu sınıflar Tablo 7.4'te sunulmuştur. $x^4+x^3+x^2+x+1$ indirgenemez polinomu için ise

ikili matrisin dallanma sayısı ile transpozunun dallanma sayısı eşit olacak şekilde ikili matrisler elde edilememiştir (Örneğin ikili matrisin transpozu'un dallanma sayısı 7 iken kendisinin dallanma sayısı 6).

Tablo 7.4: 1 Sabit Noktalı Dallanma Sayısı 7 Olan 16×16 Boyutundaki Matrise Dönüştürülebilecek $x^4 + x^3 + 1$ İndirgenemez Polinomu ile Tanımlı Olan 4×4 Boyutundaki Matris Sınıfları

1-)had(1_h,2_h,4_h,F_h)	23-)had(2_h,5_h,8_h,B_h)	45-)had(4_h,6_h,9_h,C_h)
2-)had(1_h,2_h,7_h,9_h)	24-)had(2_h,5_h,A_h,B_h)	46-)had(4_h,7_h,9_h,D_h)
3-)had(1_h,2_h,A_h,E_h)	25-)had(2_h,5_h,A_h,F_h)	47-)had(4_h,7_h,B_h,F_h)
4-)had(1_h,2_h,B_h,C_h)	26-)had(2_h,7_h,8_h,E_h)	48-)had(4_h,9_h,A_h,B_h)
5-)had(1_h,3_h,5_h,E_h)	27-)had(2_h,8_h,A_h,B_h)	49-)had(4_h,9_h,A_h,F_h)
6-)had(1_h,4_h,7_h,F_h)	28-)had(2_h,8_h,B_h,F_h)	50-)had(4_h,9_h,B_h,F_h)
7-)had(1_h,4_h,9_h,B_h)	29-)had(2_h,9_h,B_h,D_h)	51-)had(4_h,B_h,D_h,F_h)
8-)had(1_h,5_h,9_h,E_h)	30-)had(2_h,9_h,B_h,F_h)	52-)had(5_h,6_h,7_h,C_h)
9-)had(1_h,5_h,A_h,D_h)	31-)had(3_h,4_h,6_h,D_h)	53-)had(5_h,6_h,B_h,C_h)
10-)had(1_h,6_h,9_h,C_h)	32-)had(3_h,4_h,B_h,E_h)	54-)had(5_h,6_h,B_h,F_h)
11-)had(1_h,6_h,9_h,D_h)	33-)had(3_h,4_h,B_h,F_h)	55-)had(5_h,7_h,8_h,D_h)
12-)had(1_h,8_h,A_h,F_h)	34-)had(3_h,5_h,6_h,8_h)	56-)had(5_h,7_h,9_h,C_h)
13-)had(1_h,8_h,B_h,C_h)	35-)had(3_h,5_h,6_h,9_h)	57-)had(5_h,7_h,9_h,E_h)
14-)had(1_h,A_h,B_h,E_h)	36-)had(3_h,5_h,6_h,D_h)	58-)had(5_h,A_h,E_h,C_h)
15-)had(1_h,A_h,C_h,E_h)	37-)had(3_h,5_h,B_h,F_h)	59-)had(6_h,9_h,A_h,C_h)
16-)had(1_h,A_h,C_h,F_h)	38-)had(3_h,6_h,8_h,A_h)	60-)had(6_h,9_h,C_h,D_h)
17-)had(1_h,A_h,E_h,F_h)	39-)had(3_h,7_h,8_h,B_h)	61-)had(6_h,9_h,E_h,F_h)
18-)had(2_h,3_h,A_h,D_h)	40-)had(3_h,7_h,8_h,F_h)	62-)had(7_h,8_h,9_h,A_h)
19-)had(2_h,3_h,B_h,C_h)	41-)had(3_h,7_h,9_h,F_h)	63-)had(7_h,8_h,E_h,F_h)

20-)had(2_h,4_h,5_h,D_h)	42-)had(3_h,8_h,B_h,E_h)	64-)had(7_h,9_h,C_h,E_h)
21-)had(2_h,4_h,7_h,8_h)	43-)had(3_h,8_h,D_h,E_h)	65-)had(7_h,A_h,C_h,F_h)
22-)had(2_h,4_h,B_h,E_h)	44-)had(3_h,9_h,A_h,E_h)	66-)had(7_h,C_h,D_h,E_h)

Diğer yandan tez çalışması boyunca, 4×4 boyutunda ve elemanları $GF(2^4)$ (x^4+x+1 x^4+x^3+1 ve $x^4+x^3+x^2+x+1$ indirgenemez polinomları ile tanımlı) cismine ait dairesel matrisler incelenmiştir ve i, ii ve iv kriterlerini sağlayan matrislerin sayısı sırası (yukarıda verilen indirgemez polinomlar için) ile 2744, 2744 ve 96 olarak elde edilmiştir. Buna ek olarak bahsedilen kriptografik özelliklere sahip 16×16 boyutunda matrislerin elde edilmesini sağlayan 4×4 boyutundaki matrislerin MDS matris olup olmadığı da incelenmiş ve sonuçlar Tablo 7.1'de verilmiştir.

Sonuç olarak yapılan çalışmada 12858 adet sabit nokta sayısı 1 ve dallanma sayısı 7 olan 16×16 boyutunda matris elde edilmiştir. Buna ek olarak bu matrislerin elde edilmesini sağlayan 4×4 matrislerin %80'i MDS'dir. Daha önceki bölümlerde belirtilen permütasyonların elde edilen ikili matrislere beraber uygulanması ile en az 140032 (8752×16) adet 16×16 boyutunda dallanma sayısı 7 ve sabit nokta sayısı 1 olan matris üretilebilir.

Özetle tezin bu bölümünde, 16×16 boyutunda dallanma sayısı 7 ve sabit nokta sayısı 1 olan ikili matrisler tez çalışmasında geliştirilen yeni cebirsel yöntem ile başarılı olarak elde edilmiştir. Bu tip matrislere tarafımızdan yaklaşık MDBL kod adı verilmektedir.

BÖLÜM 8

Üretilen Matrislerin ARIA Blok Şifresine Uygulanması

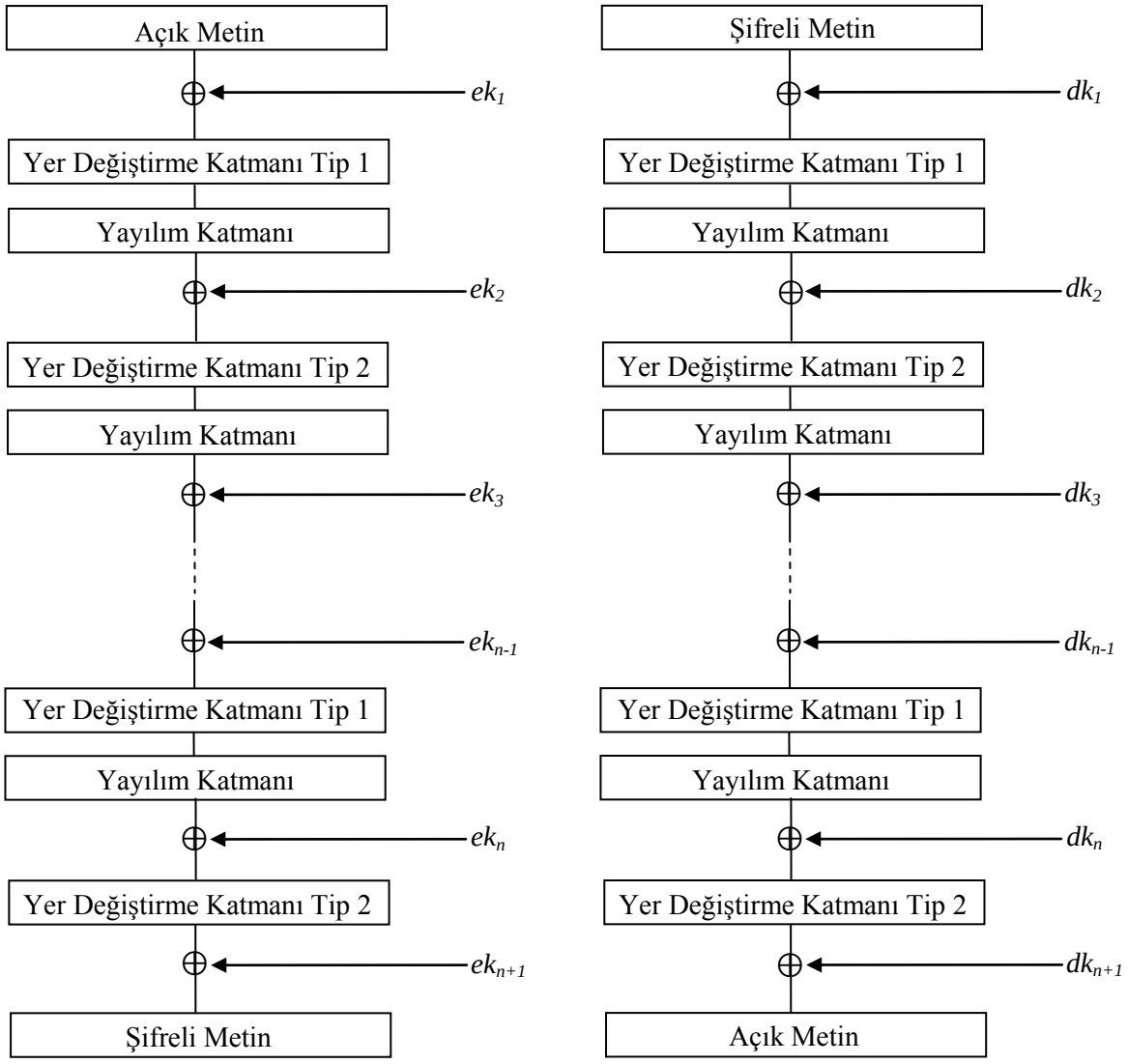
8.1. Giriş

Tezin bu bölümünde ARIA şifreleme algoritması tanıtılacaktır. Bunun yanında daha önceki bölümlerde yeni teknik ile üretilen doğrusal dönüşümlerden biri seçilerek ARIA blok şifreleme algoritmasına uygulanmaktadır.

8.2.ARIA Blok Şifreleme Algoritması

ARIA, 128-bit veri bloklarını 128/192/256 bit anahtar seçenekleri ile sırası ile 10/12/14 döngüde şifreleyen SPN (Substitution-Permutation Network) mimarisini kullanan bir blok şifreleme algoritmasıdır [9]. ARIA'nın şifreleme ve şifre çözme blok şeması Şekil 8.1'de verilmiştir.Buna göre, açık metin 128-bit veri bloklarına yani dizilere bölünerek işlenir. Anahtar seçimine göre döngü sayısı belirlenir ve her döngü aşağıdaki adımlardan oluştur:

- *Döngü Anahtarı Ekleme*: 128-bit boyutundaki dizi aynı boyuttaki döngü anahtarı ile XOR işlemine alınır,
- *Yer Değiştirme Katmanı*: Durum dizisi içerisindeki byte veriler, S-kutularından (Yer değiştirme kutuları) geçerek başka byte değerlerine haritalanır. ARIA şifreleme algoritması S_1 ve S_2 adında iki farklı S-kutusu kullanmaktadır.
- *Yayılım Katmanı*: 128-bit boyutundaki dizi içerisindeki değerler 16×16 boyutundaki ikili matris ile çarpılır.



Şekil 8.1: ARIA Şifreleme Algoritmasının Blok Diyagramı

8.2.1. Yer Değiştirme Katmanı

ARIA algoritması, şifreleme aşamasında S_1 ve S_2 adında ve şifre çözme aşamasından bu S-kutularının tersi olan S_1^{-1} ve S_2^{-1} adında iki tip S-kutusu kullanmaktadır. Bu S-kutuları Tablo 8.1 ve Tablo 8.2'de verilmiştir. Örneğin $S_1(00) = 63$, $S_1(05) = 6B$ şeklinde bir yer değiştirme işlemi gerçekleşmektedir.

Tablo 8.1: S-kutusu S_1

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Tablo 8.2: S-kutusu S_1^{-1}

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

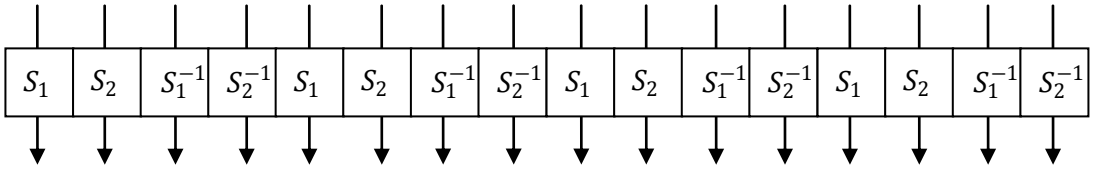
Tablo 8.3: S-kutusu S_2

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	E2	4E	54	FC	94	C2	4A	CC	62	0D	6A	46	3C	4D	8B	D1
1	5E	FA	64	CB	B4	97	BE	2B	BC	77	2E	03	D3	19	59	C1
2	1D	06	41	6B	55	F0	99	69	EA	9C	18	AE	63	DF	E7	BB
3	00	73	66	FB	96	4C	85	E4	3A	09	45	AA	0F	EE	10	EB
4	2D	7F	F4	29	AC	CF	AD	91	8D	78	C8	95	F9	2F	CE	CD
5	08	7A	88	38	5C	83	2A	28	47	DB	B8	C7	93	A4	12	53
6	FF	87	0E	31	36	21	58	48	01	8E	37	74	32	CA	E9	B1
7	B7	AB	0C	D7	C4	56	42	26	07	98	60	D9	B6	B9	11	40
8	EC	20	8C	BD	A0	C9	84	04	49	23	F1	4F	50	1F	13	DC
9	D8	C0	9E	57	E3	C3	7B	65	3B	02	8F	3E	E8	25	92	E5
A	15	DD	FD	17	A9	BF	D4	9A	7E	C5	39	67	FE	76	9D	43
B	A7	E1	D0	F5	68	F2	1B	34	70	05	A3	8A	D5	79	86	A8
C	30	C6	51	4B	1E	A6	27	F6	35	D2	6E	24	16	82	5F	DA
D	E6	75	A2	EF	2C	B2	1C	9F	5D	6F	80	0A	72	44	9B	6C
E	90	0B	5B	33	7D	5A	52	F3	61	A1	F7	B0	D6	3F	7C	6D
F	ED	14	E0	A5	3D	22	B3	F8	89	DE	71	1A	AF	BA	B5	81

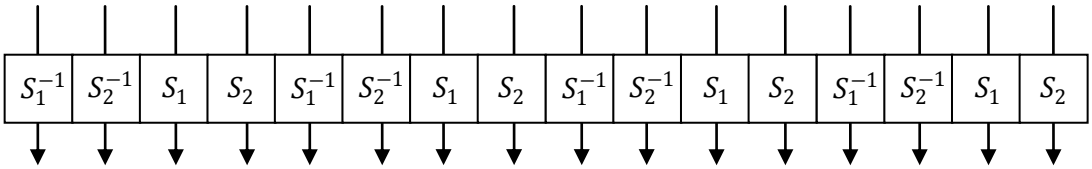
Tablo 8.4: S-kutusu S_2^{-1}

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	30	68	99	1B	87	B9	21	78	50	39	DB	E1	72	09	62	3C
1	3E	7E	5E	8E	F1	A0	CC	A3	2A	1D	FB	B6	D6	20	C4	8D
2	81	65	F5	89	CB	9D	77	C6	57	43	56	17	D4	40	1A	4D
3	C0	63	6C	E3	B7	C8	64	6A	53	AA	38	98	0C	F4	9B	ED
4	7F	22	76	AF	DD	3A	0B	58	67	88	06	C3	35	0D	01	8B
5	8C	C2	E6	5F	02	24	75	93	66	1E	E5	E2	54	D8	10	CE
6	7A	E8	08	2C	12	97	32	AB	B4	27	0A	23	DF	EF	CA	D9
7	B8	FA	DC	31	6B	D1	AD	19	49	BD	51	96	EE	E4	A8	41
8	DA	FF	CD	55	86	36	BE	61	52	F8	BB	0E	82	48	69	9A
9	E0	47	9E	5C	04	4B	34	15	79	26	A7	DE	29	AE	92	D7
A	84	E9	D2	BA	5D	F3	C5	B0	BF	A4	3B	71	44	46	2B	FC
B	EB	6F	D5	F6	14	FE	7C	70	5A	7D	FD	2F	18	83	16	A5
C	91	1F	05	95	74	A9	C1	5B	4A	85	6D	13	07	4F	4E	45
D	B2	0F	C9	1C	A6	BC	EC	73	90	7B	CF	59	8F	A1	F9	2D
E	F2	B1	00	94	37	9F	D0	2E	9C	6E	28	3F	80	F0	3D	D3
F	25	8A	B5	E7	42	B3	C7	EA	F7	4C	11	33	03	A2	AC	60

ARIA şifreleme algoritması Tablo 8.1, Tablo 8.2, Tablo 8.3 ve Tablo 8.4'te verilen S-kutularını kullanarak 2 tip yer değiştirme katmanı kullanmaktadır. Bu katmanlar Şekil 8.2'de verilmiştir.



Tip 1



Tip 2

Şekil 8.2: ARIA'da Kullanılan Yer Değiştirme Katmanları

Şekil 8.1 incelendiğinde görüleceği gibi önce tek numaralı döngülerde Tip1, çift numaralı döngülerde ise Tip 2 kullanılmaktadır.

8.2.2. Yayılım Katmanı

ARIA şifresi yayılım katmanı olarak 16×16 boyutunda giriş değerlerini byte değerler olarak alan involutif ve MDBL kod olarak tasarlanmış bir ikili matris kullanır. ARIA matrisinde kullanılan bu involutif ikili matrisin, $(A-I)$ matrisinin rank değeri 7 olduğundan bu dönüşüm $2^{8(16-7)} = 2^{72}$ adet sabit noktaya içerir. Dallanma sayısı 8 olan bu ikili matris (8.1)'de verilmiştir.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{bmatrix} \quad (8.1)$$

ARIA'nın yayılım katmanı (x_0, x_1,..., x_{15}) girişlerini 16 byte'lık (y_0, y_1,..., y_{15}) çıkışlarına haritalar. Bu haritalama işlemi cebirsel olarak aşağıdaki gibi ifade edilebilir.

$$y_0 = x_3 \oplus x_4 \oplus x_6 \oplus x_8 \oplus x_9 \oplus x_{13} \oplus x_{14} \qquad y_8 = x_0 \oplus x_1 \oplus x_4 \oplus x_7 \oplus x_{10} \oplus x_{13} \oplus x_{15}$$
$$y_1 = x_2 \oplus x_5 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{12} \oplus x_{15} \qquad y_9 = x_0 \oplus x_1 \oplus x_5 \oplus x_6 \oplus x_{11} \oplus x_{12} \oplus x_{14}$$
$$y_2 = x_1 \oplus x_4 \oplus x_6 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus x_{15} \qquad y_{10} = x_2 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_8 \oplus x_{13} \oplus x_{15}$$
$$y_3 = x_0 \oplus x_5 \oplus x_7 \oplus x_{10} \oplus x_{11} \oplus x_{13} \oplus x_{14} \qquad y_{11} = x_2 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_9 \oplus x_{12} \oplus x_{14}$$
$$y_4 = x_0 \oplus x_2 \oplus x_5 \oplus x_8 \oplus x_{11} \oplus x_{14} \oplus x_{15} \qquad y_{12} = x_1 \oplus x_2 \oplus x_6 \oplus x_7 \oplus x_9 \oplus x_{11} \oplus x_{12}$$
$$y_5 = x_1 \oplus x_3 \oplus x_4 \oplus x_9 \oplus x_{10} \oplus x_{14} \oplus x_{15} \qquad y_{13} = x_0 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_8 \oplus x_{10} \oplus x_{13}$$
$$y_6 = x_0 \oplus x_2 \oplus x_7 \oplus x_9 \oplus x_{10} \oplus x_{12} \oplus x_{13} \qquad y_{14} = x_0 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_9 \oplus x_{11} \oplus x_{14}$$
$$y_7 = x_1 \oplus x_3 \oplus x_6 \oplus x_8 \oplus x_{11} \oplus x_{12} \oplus x_{13} \qquad y_{15} = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_8 \oplus x_{10} \oplus x_{15}$$

8.2.3. Örnek Şifreleme İşlemleri

Seçilen anahtar: **000102030405060708090A0B0C0D0E0F**

Üretilen şifreleme anahtarları:

Döngü	Şifreleme Anahtarları
1	D415A75C794B85C5E0D2A0B3CB793BF6
2	369C65E4B11777AB713A3E1E6601B8F4
3	0368D4F13D14497B6529AD7AC809E7D0
4	C644552B549A263FB8D0B50906229EEC
5	5F9C434951F2D2EF342787B1A781794C
6	AFEA2C0CE71DB6DE42A47461F4323C54
7	324286DB44BA4DB6C44AC306F2A84B2C
8	7F9FA93574D842B9101A58063771EB7B
9	AAB9C57731FCD213AD5677458FCFE6D4
10	2F4423BB06465ABADA5694A19EB88459
11	9F8772808F5D580D810EF8DDAC13ABEB
12	8684946A155BE77EF810744847E35FAD
	0F0AA16DAEE61BD7DFEE5A599970FB35

Üretilen Şifre çözme anahtarları:

Döngü	Şifre Çözme Anahtarları
1	0F0AA16DAEE61BD7DFEE5A599970FB35
2	CCB3A0230B6DAC1D53EEF49D961AA57F
3	60EA3252AC3EA9BC9AC78E79DF20B5B5
4	5794EADAECE652F8A2CCBF68EE82A730
5	468A335E49EC1DB45D112AAF2109E5BF
6	938EBBDA880C6BB87FA01C97E68811A9
7	BFDA5018AB33D14CC538EA5C81BD1011
8	B5A90E77D5B94BB56E47AF759FCFA05E
9	21A6C28C5E1175A4378CD34DD3195A83
10	8D726063CA2CEDDC92AFB45DD7DB643E
11	27EFD355EB17E90E5963C46515016F8D
12	D000E81367819B077B0A657F6740E8E4
	D415A75C794B85C5E0D2A0B3CB793BF6

Şifrelenecek olan veri: **74656C656B6F6DFC6E696B6173796F6E**

Şifreleme anahtarı: **000102030405060708090A0B0C0D0E0F**

Şifreleme adımları:

Döngü	Döngü Çıkışları
0	27203390B8834E8BF692EC8BA38A684B
1	BD5293F2A74E4CEEB5592FE0DD796F1A
2	721649CB2DBABB51D30C2987671E325F
3	7653F7FEA9EC36EA41A52F8C8936757A
4	3B62FB6BB6ED3C68B8CC8C7242043492
5	B2CE880704881AA6DC6C4AD03FFBB57B
6	5C95B3BFA222CAC1F37CE247ABC15424
7	E384914CE8FDA62BE8577338EB52DDC8
8	2B55509E9DA28D5070BF7C4763413BE0
9	5E67757DB59ED0DD830C6D27158C2BD2
10	491EC7E2FDDBB87038497E909EC06E13
Şifreli Metin	50AD4C24663CD45CC0F03D047CF93CB3

Şifreli Metin: **50AD4C24663CD45CC0F03D047CF93CB3**

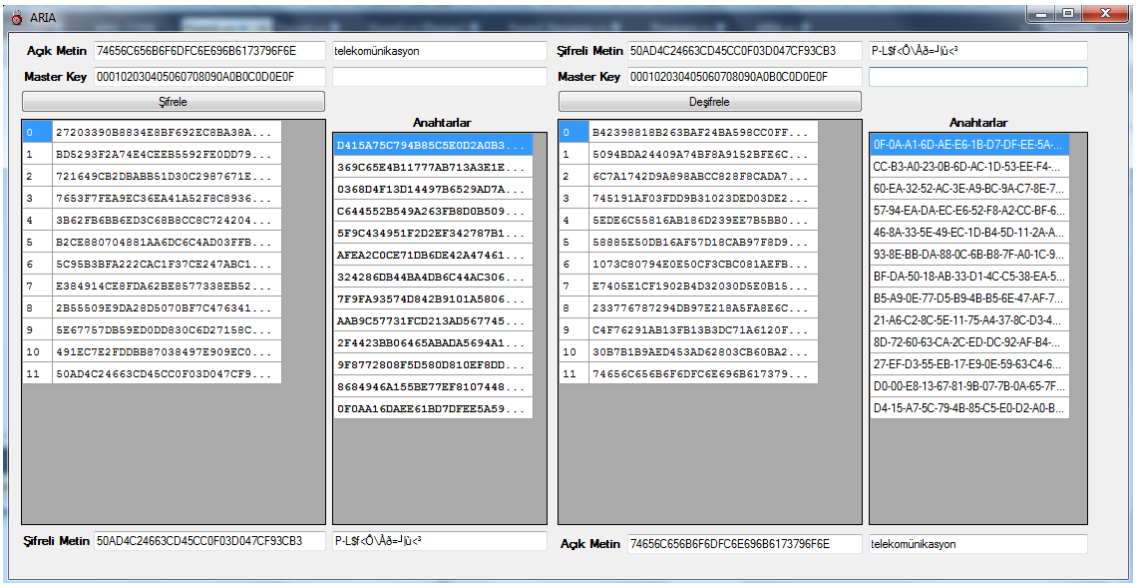
Şifre çözme Adımları:

Döngü	Döngü Çıkışları
0	B42398818B263BAF24BA598CC0FF9FD5
1	5094BDA24409A74BF8A9152BFE6CBDB0
2	6C7A1742D9A898ABCC828F8CADA7EEE6
3	745191AF03FDD9B31023DED03DE2EDEC
4	5EDE6C55816AB186D239EE7B5BB01D3A
5	58885E50DB16AF57D18CAB97F8D92036
6	1073C80794E0E50CF3CBC081AEFB213A
7	E7405E1CF1902B4D32030D5E0B15CB76
8	233776787294DB97E218A5FA8E6CF353
9	C4F76291AB13FB13B3DC71A6120F1F25
10	30B7B1B9AED453AD62803CB60BA2159D
Açık Metin	74656C656B6F6DFC6E696B6173796F6E

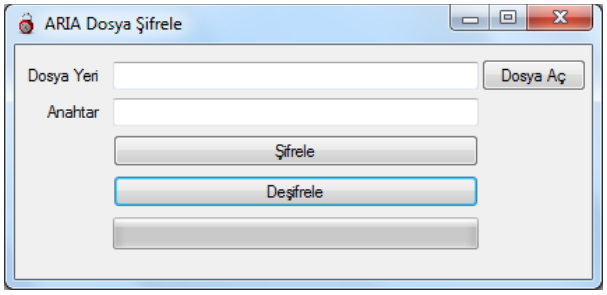
Açık metin: **74656C656B6F6DFC6E696B6173796F6E**

ARIA şifreleme algoritmanın şifreleme adımlarını gerçekleyebilmek için Şekil 8.3'teki yazılım geliştirilmiştir. Bunun yanında dosya şifrelemek için bir ara yüz daha hazırlanmıştır. Bu ara yüzde dosya yeri ve şifreleme anahtarı belirlenerek tüm dosyanın

şifrelenmesi ve deşifrelenmesi sağlanmaktadır. Dosya Şifrele yazılımı Şekil 8.4'deki gibidir.



Şekil 8.3: Programın arayüzü



Şekil 8.4: Dosya Şifreleme

8.3.Tez Çalışmasında Üretilen Doğrusal Dönüşümlerin ARIA Blok Şifresine Uygulanması

Tez çalışması esnasında ARIA benzeri şifreleme algoritmalarında kullanılabilecek birçok doğrusal dönüşüm üretilmiştir. Tezin bu bölümünde üretilen ikili doğrusal dönüşüm yapılarından bir tanesi ARIA şifresindeki doğrusal dönüşüm yapısı yerine kullanılmıştır.

4×4 boyutundaki matrisin elemanları x^4+x+1 indirgenemez polinomu ile tanımlanan $GF(2^4)$ cisminden seçilmiştir. Buna göre $M = had(1_h, 5_h, 8_h, D_h)$ Hadamard biçimindeki matris ikili biçime Tablo 4.1 kullanılarak dönüştürülebilir. ARIA doğrusal dönüşümünde kullanılan 16×16 boyutundaki doğrusal dönüşüm yerine kullanılacak yapı (8.2)'de verilmiştir.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \end{bmatrix} = \begin{bmatrix} 1&0&0&0&1&0&1&0&0&1&0&0&1&1&1&0 \\ 0&1&0&0&0&1&1&1&0&1&1&0&0&0&0&1 \\ 0&0&1&0&1&0&1&1&0&0&1&1&1&0&0&0 \\ 0&0&0&1&0&1&0&1&1&0&0&1&1&1&0&0 \\ 1&0&1&0&1&0&0&0&1&1&1&0&0&1&0&0 \\ 0&1&1&1&0&1&0&0&0&0&0&1&0&1&1&0 \\ 1&0&1&1&0&0&1&0&1&0&0&0&0&0&1&1 \\ 0&1&0&1&0&0&0&1&1&1&0&0&1&0&0&1 \\ 0&1&0&0&1&1&1&0&1&0&0&0&1&0&1&0 \\ 0&1&1&0&0&0&0&1&0&1&0&0&0&1&1&1 \\ 0&0&1&1&1&0&0&0&0&0&1&0&1&0&1&1 \\ 1&0&0&1&1&1&0&0&0&0&0&1&0&1&0&1 \\ 1&1&1&0&0&1&0&0&1&0&1&0&1&0&0&0 \\ 0&0&0&1&0&1&1&0&0&1&1&1&0&1&0&0 \\ 1&0&0&0&0&0&1&1&1&0&1&1&0&0&1&0 \\ 1&1&0&0&1&0&0&1&0&1&0&1&0&0&0&1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{bmatrix} \quad (8.2)$$

8.3.1. Örnek şifreleme İşlemleri

Seçilen anahtar: **000102030405060708090A0B0C0D0E0F**

Üretilen şifreleme anahtarları:

Döngü	Şifreleme Anahtarları
1	576E7FE339FF0195C6C896F0D2381675
2	5CBB21E798A4BB389127C2EE6FE62486
3	5619E953872B706D3C9C8972A334077B
4	E7EE94461D6555CFE67207CDF4285CA7
5	87A474F4DA06D9A77125E612C3B0ED5E
6	10DAD47CDFEC352034999E3BF7B87832

7	DB6F5BDAE5FD5F45E07F0B3A7288504F
8	5E3568583D2731494EB8EAC2C57B2930
9	9CFADC361C7F516875E937F10B9FC0CE
10	DC4743E4A36BC5B629E9559A0EE6CD82
11	996B7D4FF30E906D52F08F2C38493B7F
12	A72E55075CA4948EA6B2C68CB5E99DE6
	1E4A390577EA7ED369E057B6FB8DF9E7

Üretilen şifre çözme anahtarları:

Döngü	Şifre Çözme Anahtarları
1	1E4A390577EA7ED369E057B6FB8DF9E7
2	1C02EC5B9520BCE0CCD506DBAD26CC83
3	4098E82373092DAC5786323164CBAECF
4	766152624F3AD859A0F2100EC18A847F
5	6CAC343168C15CFD7636C4698CA8D6C4
6	7D08C0046CB20827BCCE3B9545A56955
7	B4B3E74279F1FA52EA9902E48BBE401E
8	54164CF09065F3EE674554EEAF21ECA5
9	3F41E7452B6D1C3EFE51CC53054A9137
10	5DC32D1BD5E07DA08D14C79BEDE60DC3
11	AD412BCC25C9DC5F5D496B949B5B8B30
12	F5FF798D77F5123BE6C151095E154608
	576E7FE339FF0195C6C896F0D2381675

Şifrelenecek olan veri: **74656C656B6F6DFC6E696B6173796F6E**

Şifreleme anahtarı: **000102030405060708090A0B0C0D0E0F**

Şifreleme adımları:

Döngü	Döngü Çıkışları
0	A14B4989E5357944795836CEEB1AB26B
1	8A3028814240CF83433BE0BD3E6B48C2
2	D1224D1A02785B03A45AE7F6DB21AE23
3	ED8C468EA39BF6CD7663EB6F6F3DADA3
4	AB77D5D191B6A9EA1123F499CB9FE6BF
5	CBF3E869CF99FC136F7BC7FF8C89787F
6	D1A99F9AEA6A5BF76C22EAC8B0E5DD4F
7	C924D789D45A23B23F0D979AF762D55C
8	521E7365DCE6BE74D4BC954628E8B080
9	36ECF42B549D691DE1A67DA10CC780C0

10	93092BD38716C654ED2CFB9EA959775E
Şifreli Metin	368CCA29E83F7E53A57270D23F667E97

Şifreli Metin: **368CCA29E83F7E53A57270D23F667E97**

Şifre Çözme Adımları:

Döngü	Döngü Çıkışları
0	65061E49C977D558A1FF0293B5353226
1	A686EC0318410CFD76A2884347A9519B
2	8AFACCC7A7CAC696768F572D7130ABE1
3	1F855C6065CC5E7BE291A703B33A6984
4	B7E08BF289847952CF5A1C3CFEEB8795
5	4AF59B34CF14243C28640019E6784B01
6	56FCEDA12640BDE4A2E8A70A3E3E9E07
7	1B46D2D6E04DB72808062DF94B731897
8	DB5FF0D27394899D5F8E23DEB3B59FBE
9	8C646E25368EF9E99508D489D4581B0F
10	D3B9FB33772DAA1C241C704E6C6AE9BE
Açık Metin	74656C656B6F6DFC6E696B6173796F6E

Açık metin: **74656C656B6F6DFC6E696B6173796F6E**

BÖLÜM 9

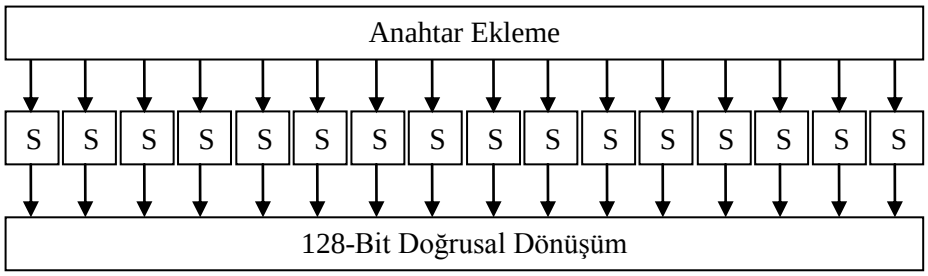
Yeni Bir Blok Şifre Uygulaması: TRAKYA

9.1. Giriş

Tezin bu bölümünde, tez süresince elde edilen elemanlar kullanılarak yeni bir şifre uygulaması gerçekleştirilecektir. Tasarlanan şifre mimarisinde ARIA ve AES şifrelerinden esinlenilmiştir.

9.2. Blok Şifre TRAKYA

TRAKYA blok şifre uygulaması 128-bit blokları 128-bit anahtar ile şifreleyen 10 döngülük bir algoritma olarak tasarlanmıştır. Her döngü anahtar ekleme, yer değiştirme katmanı ve yayılım katmanından oluşmaktadır. Bir döngünün yapısı Şekil 9.1'de verilmiştir.



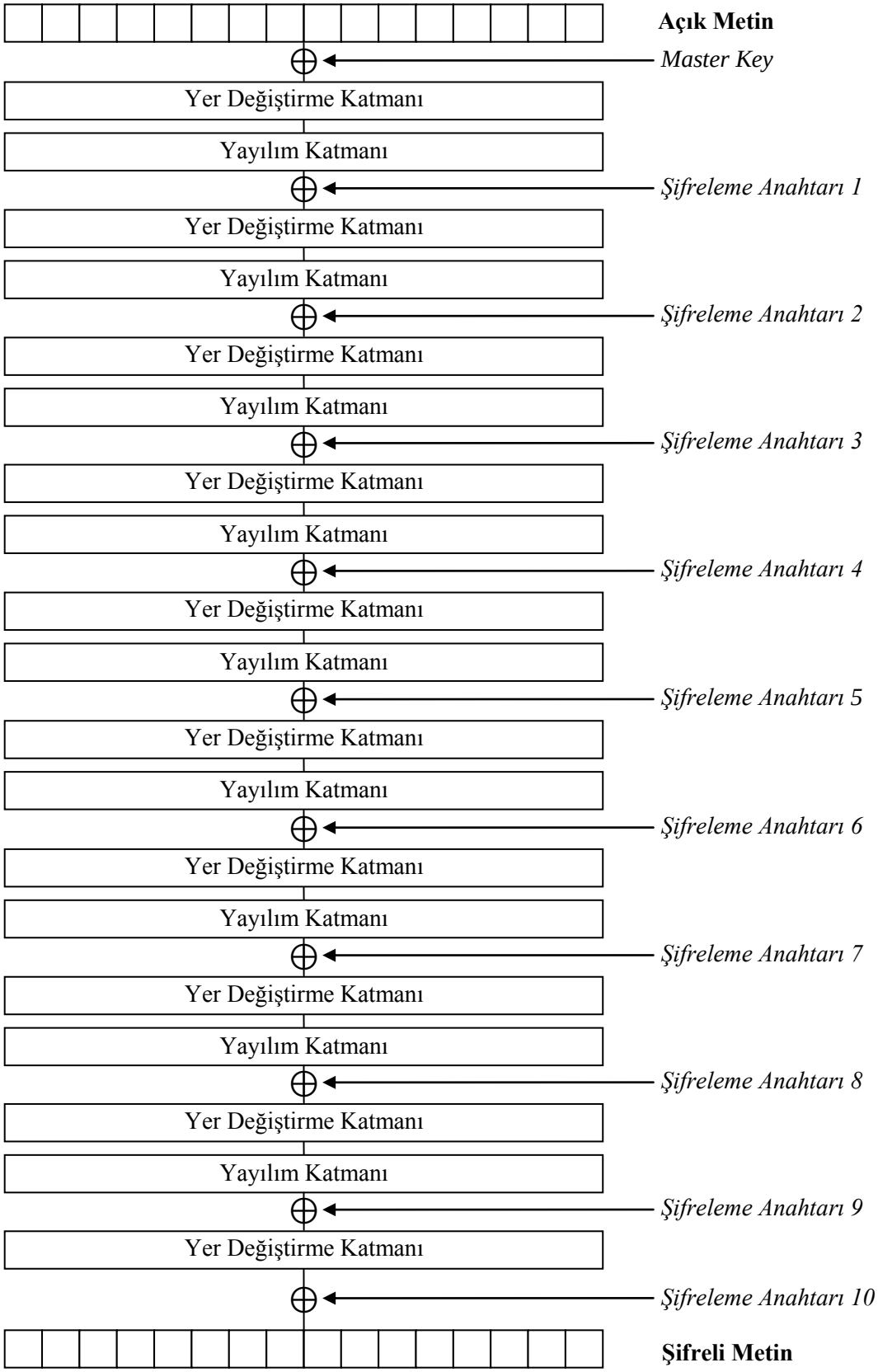
Şekil 9.1: TRAKYA Algoritmasının Tek Döngülük SPN Mimarisi

Şifreleme ve şifre çözme adımlar Şekil 9.2 ve 9.3'de verilmiştir. Şifrede gerçekleşen işlemlerin XOR tabanlı olmasına önem verilerek şifrenin hızı arttırılmıştır. Şifre için yeni bir anahtar planlama evresi tanımlanmıştır. Anahtar planlama evresi Şekil 9.7'da gösterilmiştir. Şifrede kullanılan tüm elemanlar tez süresince geliştirilen elemanlarıdır. SPN mimarisi üzerine geliştirilen şifrede bir döngü içerisinde aşağıdaki adımlar mevcuttur:

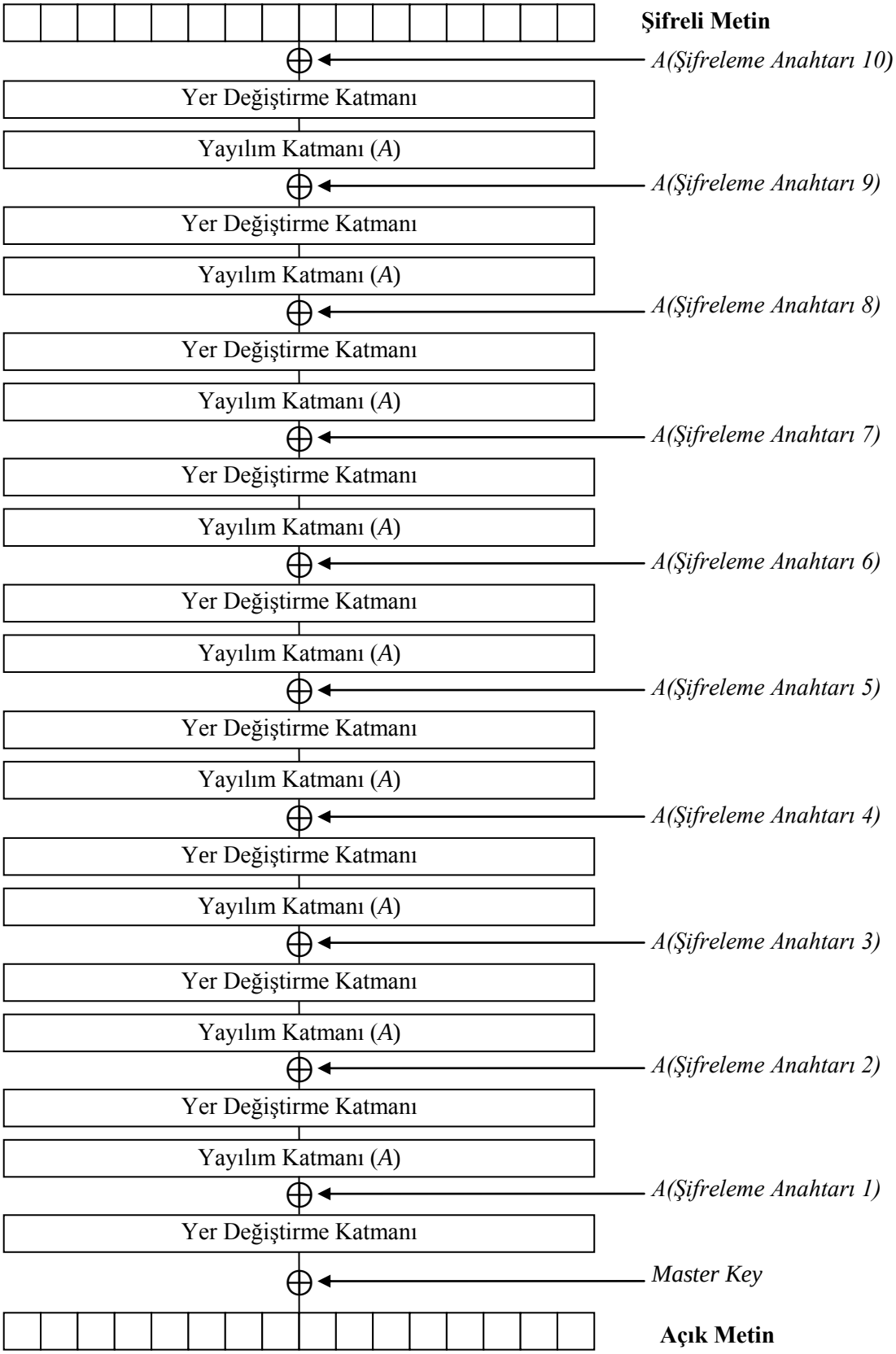
- *Yer Değiştirme Katmanı*: Dizi içerisindeki byte veriler, S-kutusundan geçerek başka byte değerlerine haritalanır. Bu işlem şifreye karışıklığın (confusion) verilmesi için kullanılmaktadır. Şifre içerisinde sadece bir adet S-kutusu

kullanılmıştır. Kullanılan S-kutusu, 2008 yılında tamamlanan yüksek lisans çalışmasında üretilen cebirsel olarak güçlendirilmiş bir S-kutusudur [41].

- *Yayılım Katmanı*: Şifreye yayılımın (diffusion) verilebilmesi için kullanılan yayılım katmanında, 16×16 boyutunda involutif ikili matris kullanılmıştır. Kullanılan ikili doğrusal dönüşüm Bölüm 4'te geliştirilen yeni yöntem ile tasarlanmıştır.
- *Döngü Anahtarı Ekleme*: Şifrenin anahtar evresinden gelen 128-bit boyutundaki döngü anahtarı, giriş bitleri ile XOR işlemine alınır.



Şekil 9.2: Geliştirilen Şifrenin Şifreleme Blok Diyagramı



Şekil 9.3: Geliştirilen Şifrenin Şifre Çözme Blok Diyagramı

9.2.1. Yer Değiştirme Katmanı

Tasarlanan şifrede kullanılan S-kutusu Tablo 9.1'de verilmiştir. $x \to x^{254}$ üs haritalaması ile oluşturulmuş olan S-kutusu 2008 yılında tamamlanan yüksek lisans tezinden alınmıştır [41].

Tablo 9.1: S-kutusu

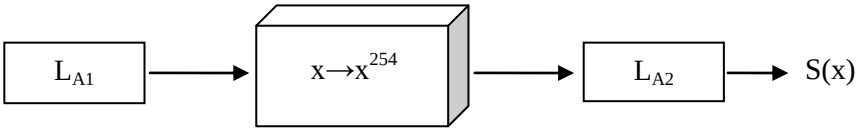
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	C3	18	27	80	15	34	FD	F7	2B	FE	6B	77	F0	CA	D4	72
1	1A	1B	E3	D6	CF	6A	D1	B1	21	10	9D	40	85	D0	F9	9F
2	66	48	C1	57	8A	E8	78	B4	E9	CE	D9	98	68	8C	99	BB
3	0A	49	95	AC	08	6C	C8	4E	14	DE	2A	4F	17	CD	A7	19
4	89	E6	B0	0F	28	1E	E1	94	74	BD	1C	2E	F6	3E	61	9E
5	13	97	64	3D	0B	EE	60	88	F4	7A	8D	6D	24	32	C2	79
6	C9	59	9C	AF	AB	01	63	C5	E5	D8	36	26	05	C7	07	75
7	AA	4D	50	7F	F3	B6	51	F5	BE	4C	20	ED	5A	83	52	84
8	E7	A9	AE	56	91	62	3A	06	C4	73	44	0C	22	DC	B8	5E
9	BA	C6	8B	DD	86	B9	B5	03	41	16	42	A1	69	11	87	55
A	53	5B	58	CB	29	B3	2C	6E	45	A8	33	EF	92	8F	DA	FF
B	B7	CC	31	A5	EB	E2	23	96	AD	C0	47	82	F2	7B	67	D7
C	A3	38	D2	BC	3C	02	FB	43	3B	2F	A0	09	FC	00	39	4A
D	7C	6F	76	30	A4	A2	7D	FA	12	B2	9A	04	3F	93	F1	71
E	81	90	DB	46	5D	7E	EC	5F	D3	E4	5C	E0	D5	37	EA	65
F	F8	8E	DF	9B	54	2D	0D	BF	35	1D	0E	70	A6	25	1F	4B

S-kutusu cebirsel olarak güçlendirilmiştir. Bunun için (9.1) ve (9.2) ifadelerindeki doğrusal dönüşümler kullanılmıştır. Doğrusal dönüşümlerin sabitleri sırası ile $L_{A_{S1}} = 33_h$ ve $L_{A_{S1}} = 63_h$ şeklindedir. Bu değerler hexadecimal notasyonda gösterilmiştir.

$$L_{A1}(x) = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} . \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad (9.1)$$

$$L_{A2}(x) = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad (9.2)$$

S-kutusu, Şekil 9.4'de verilen mimari ile oluşturulmuştur. L_{A1} doğrusal dönüşümünden sonra haritalama işlemi uygulanmıştır. Daha sonrasında ise L_{A2} doğrusal dönüşümü uygulanarak S-kutusu yaratılmıştır.



Şekil 9.4: S-kutusu Tasarımı

AES S-kutusunun cebirsel açılımında 9 terim bulunmaktadır [42]. Bu terimler ifade (9.3)'te verilmiştir.

$$S(x) = 63 + 05x^{224} + 09x^{253} + f9x^{251} + 25x^{247} + f4x^{239} + 01x^{223} + b5x^{191} + 8fx^{127} \quad (9.3)$$

Şekil 9.3'te verilen mimari ile üretilen S-kutusunun terim sayısı 255'tir [43]. AES algoritmasında kullanılan S-kutusundan cebirsel olarak daha gelişkindir. Tablo 9.1'de verilen S-kutusunun cebirsel olarak açılımı sonucunda elde edilen 255 terim ifade (9.4)'te gösterilmiştir.

$$
\begin{aligned}
S(x) = {} & 1Cx^{254} + 1Ex^{253} + 16x^{252} + 98x^{251} + 07x^{250} + 58x^{249} + 86x^{248} \\
& + E2x^{247} + B5x^{246} + 11x^{245} + 06x^{244} + 8Ex^{243} + BAx^{242} + 9Ex^{241} \\
& + 3Fx^{240} + A4x^{239} + 22x^{238} + 3Cx^{237} + E4x^{236} + 1Ax^{235} + 9Ax^{234} \\
& + 18x^{233} + DDx^{232} + 99x^{231} + 82x^{230} + 4Cx^{229} + 98x^{228} + DEx^{227} \\
& + 25x^{226} + F8x^{225} + 75x^{224} + BBx^{223} + 81x^{222} + FDx^{221} + D0x^{220} \\
& + C9x^{219} + 04x^{218} + 74x^{217} + F6x^{216} + B2x^{215} + 39x^{214} + 49x^{213} + 0Ax^{212} \\
& + F9x^{211} + 49x^{210} + 3Bx^{209} + 6Cx^{208} + A7x^{207} + 66x^{206} + E3x^{205} + 90x^{204} \\
& + 42x^{203} + B7x^{202} + 5Dx^{201} + 4Fx^{200} + 8Dx^{199} + DBx^{198} + 38x^{197} \\
& + 9Ax^{196} + 68x^{195} + E5x^{194} + 82x^{193} + 50x^{192} + 73x^{191} + BDx^{190} + 06x^{189} \\
& + A7x^{188} + F3x^{187} + 1Dx^{186} + 28x^{185} + 46x^{184} + 8Cx^{183} + 04x^{182} \\
& + CFx^{181} + 8Cx^{180} + C8x^{179} + 6Ex^{178} + 59x^{177} + 32x^{176} + 51x^{175} + DFx^{174} \\
& + A8x^{173} + 91x^{172} + A5x^{171} + E7x^{170} + 63x^{169} + D5x^{168} + A0x^{167} \\
& + 1Bx^{166} + 96x^{165} + D3x^{164} + 85x^{163} + 58x^{162} + AFx^{161} + C9x^{160} \\
& + 88x^{159} + 5Ex^{158} + 2Fx^{157} + A6x^{156} + 9Ax^{155} + 27x^{154} + 84x^{153} \\
& + 59x^{152} + 91x^{151} + C0x^{150} + 83x^{149} + 2Bx^{148} + 1Bx^{147} + BCx^{146} \\
& + 19x^{145} + 30x^{144} + 93x^{143} + 96x^{142} + 52x^{141} + 2Ex^{140} + 11x^{139} + 3Ex^{138} \\
& + 28x^{137} + E3x^{136} + E0x^{135} + 95x^{134} + 2Cx^{133} + 0Fx^{132} + 26x^{131} + 99x^{130} \\
& + FBx^{129} + 63x^{128} + 7Ex^{127} + 88x^{126} + 14x^{125} + A3x^{124} + DDx^{123} + 94x^{122} \\
& + 20x^{121} + B4x^{120} + 70x^{119} + 7Ex^{118} + B1x^{117} + F6x^{116} + 0Dx^{115} + 92x^{114} \\
& + 1Fx^{113} + 0Bx^{112} + 62x^{111} + 0Dx^{110} + 3Ex^{109} + 16x^{108} + D6x^{107} + F8x^{106} \\
& + E7x^{105} + 47x^{104} + 30x^{103} + 42x^{102} + CBx^{101} + 26x^{100} + 05x^{99} + 3Bx^{98} \\
& + 26x^{97} + 8Cx^{96} + A8x^{95} + 75x^{94} + A1x^{93} + 09x^{92} + D9x^{91} + 6Ax^{90} \\
& + D1x^{89} + 5Ax^{88} + 45x^{87} + 29x^{86} + D1x^{85} + C8x^{84} + 5Ex^{83} + 97x^{82} \\
& + 28x^{81} + 79x^{80} + 59x^{79} + C3x^{78} + 48x^{77} + 6Fx^{76} + E8x^{75} + 79x^{74} \\
& + 3Bx^{73} + DEx^{72} + A5x^{71} + B5x^{70} + EBx^{69} + 9Cx^{68} + C3x^{67} + DEx^{66} \\
& + 0Dx^{65} + 23x^{64} + F9x^{63} + 8Ax^{62} + F5x^{61} + 5Dx^{60} + B1x^{59} + 7Cx^{58} \\
& + 46x^{57} + 5Ax^{56} + F9x^{55} + 10x^{54} + EEx^{53} + 55x^{52} + 9Dx^{51} + 8Fx^{50} \\
& + C8x^{49} + E6x^{48} + 9Dx^{47} + C2x^{46} + FEx^{45} + 59x^{44} + 3Bx^{43} + 1Fx^{42} \\
& + 1Fx^{41} + BCx^{40} + 02x^{39} + 20x^{38} + E6x^{37} + E6x^{36} + 8Bx^{35} + 7Cx^{34} \\
& + B9x^{33} + 81x^{32} + 56x^{31} + 95x^{30} + 09x^{29} + 02x^{28} + 4Dx^{27} + 6Dx^{26} \\
& + 34x^{25} + 5Ax^{24} + 1Dx^{23} + 02x^{22} + 3Ex^{21} + FBx^{20} + 41x^{19} + 51x^{18} \\
& + E6x^{17} + EFx^{16} + 5Dx^{15} + C7x^{14} + B1x^{13} + 78x^{12} + BFx^{11} + FCx^{10} \\
& + D2x^{9} + 51x^{8} + FAx^{7} + BCx^{6} + A5x^{5} + F6x^{4} + 15x^{3} + 87x^{2} + E7x + C3.
\end{aligned}
\tag{9.4}
$$

S-kutusunun DDT dağılım 256×256 boyutlu bir tablo olacaktır. Bu tablonun bütününü incelemek çok zordur. Fakat Fark Dağılım Tablosunun (DDT) 01 numaralı satırı için dağılımı Tablo 9.2'deki gibidir. Bu dağılımda 129 adet 0, 126 adet 2 ve 1 adet 4 vardır. 4 uniform dağılım göstermektedir.

Tablo 9.2: Şifrede kullanılan S-kutusunun "01" giriş farkı için DDT dağılımı

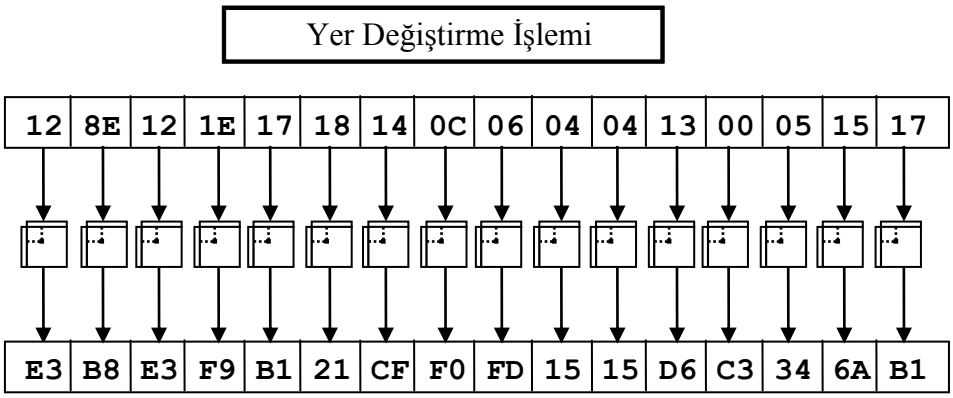
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	0	2	0	0	0	0	2	0	2	2	2	0	0	0	0	0
1	2	2	0	2	2	0	2	0	0	0	0	0	2	2	0	0
2	0	2	2	2	0	2	0	2	2	0	0	0	0	0	2	2
3	0	2	2	2	0	2	2	2	0	2	2	0	2	2	2	2
4	0	2	2	2	2	2	2	0	2	0	0	0	0	0	2	0
5	0	0	0	0	2	2	2	2	0	2	0	0	0	0	0	0
6	2	0	2	0	2	2	2	0	0	0	0	0	0	2	2	2
7	0	0	2	2	0	2	2	0	2	2	0	2	2	0	2	0
8	2	0	0	2	2	0	2	2	0	2	0	0	0	0	2	2
9	2	0	0	2	2	0	2	0	0	0	2	2	0	2	2	0
A	2	0	0	0	2	2	4	2	0	2	2	0	2	0	0	0
B	2	0	2	2	0	2	2	2	2	0	0	2	2	0	2	2
C	0	0	2	0	0	2	0	0	2	2	2	0	2	2	0	0
D	0	0	2	0	0	2	2	0	0	2	2	2	2	2	0	0
E	2	0	2	2	2	2	2	2	2	0	0	0	0	2	0	0
F	0	0	2	2	0	0	0	0	2	0	0	0	2	0	2	2

Tablo 9.3: S-kutusunun "01" giriş maskesi için LAT dağılımı

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	0	-8	8	0	16	-4	-16	-12	10	-10	-2	2	-14	-6	6	-2
1	0	-12	-8	4	4	-4	12	4	6	-2	2	10	-14	6	6	-14
2	-8	-8	0	-16	-4	16	-12	-16	2	-2	6	-6	6	-2	-14	-6
3	-12	0	4	-8	-12	4	12	12	-6	2	-2	-10	-6	-2	-2	-6
4	2	-2	-14	-10	-10	14	-2	6	-12	-12	8	8	-8	12	12	8
5	10	10	2	2	2	-2	-6	14	-8	12	-4	8	0	-8	12	-12
6	-2	-6	6	-6	-10	-2	-2	6	-8	0	-12	-4	-8	4	12	0
7	2	-14	-6	-6	6	2	6	10	-8	-12	12	0	-12	4	8	8
8	-2	10	2	14	-6	2	-2	-2	4	-4	-4	-4	0	-4	0	-4
9	2	2	6	-2	10	6	6	2	4	0	4	0	-4	12	-4	-12
A	-2	10	-6	6	-10	14	-6	10	4	-12	4	12	4	8	4	-8
B	-10	-10	-6	-14	-14	-2	-10	2	-8	-4	8	-4	-4	4	4	4
C	12	-12	-8	8	12	-8	-8	-12	-6	-2	2	-10	2	2	-14	10
D	-8	12	12	0	4	12	0	0	-14	-14	-6	2	6	2	-2	-6
E	-8	8	-12	12	-12	8	8	12	-2	2	6	10	-6	-6	-14	-6
F	8	4	4	0	8	-8	4	12	-6	10	-6	2	10	-10	2	-2

Kullanılan S-kutusunun LAT dağılım DDT'ye benzer şekilde 256×256'lık bir tablo olacaktır. $2^8\times2^8$ boyutunda LAT tablosu değerleri yerine $\mathrm{LAT}(1,\Gamma_b)$ değerlerinin elde edilmesi yeterli olacaktır. Diğer satırlarda aynı dağılımı vermektedir. Buna göre Doğrusal Yaklaşım Tablosunun (LAT) 01 numaralı satırı için dağılımı Tablo 9.3'deki gibidir. Bu tabloda 0 sayısı 17 tane, $|2|$ sayısı 48 tane, $|4|$ sayısı 36 tane, $|6|$ sayısı 40 tane, $|8|$ sayısı 34 tane, $|10|$ sayısı 24 tane, $|12|$ sayısı 36 tane, $|14|$ sayısı 16 tane, $|16|$ sayısı 5 tane şeklindedir. Dolayısı ile S-kutusunun $|\mathrm{N}_{\mathrm{Lmaks}}|=16$'dır ve aynı zamanda %93 doğrusal olmama oranındadır. Dolayısı ile şifrede kullanılan S-kutusu Lineer ve Diferansiyel saldırılara göre dayanıklıdır ve şifreye karmaşıklığı iyi derecede vermektedir.

Örnek 9.1: **128E121E1718140C0604041300051517** 128-bitlik değerinin yer değiştirme işleminden sonra **E3B8E3F9B121CFF0FD1515D6C3346AB1** değerine haritalandığı Şekil 9.5'deki gibi gösterilebilir.



Şekil 9.5: Yer Değiştirme İşlemi

9.2.2. Yayılım Katmanı

Şifrede kullanılan yayılım katmanı, Bölüm 4'te geliştirilen yeni yöntem ile tasarlanmıştır. 16×16 boyutunda ikili matris (9.5) ifadesinde verilen 4×4 boyutundaki matristen üretilmiştir [36].

$$M = had(1_h, 5_h, 8_h, D_h) = \begin{bmatrix} 1_h & 5_h & 8_h & D_h \\ 5_h & 1_h & D_h & 8_h \\ 8_h & D_h & 1_h & 5_h \\ D_h & 8_h & 5_h & 1_h \end{bmatrix} \qquad (9.5)$$

(9.5) ifadesinde verilen 4×4 boyutundaki matris MDS ve Hadamard formdadır. Matrisin elemanları indirgenemez polinom $x^4 + x + 1$ ile tanımlı $GF(2^4)$ cisminin elemanlarının olarak düşünüldüğünde Tablo 4.1 kullanılarak 4×4 büyüklüğünde ikili ifadeler şeklinde yazılabilir. Bu durumda şifrede kullanılan 16×16 boyutundaki ikili yayılım elemanına ulaşılır.

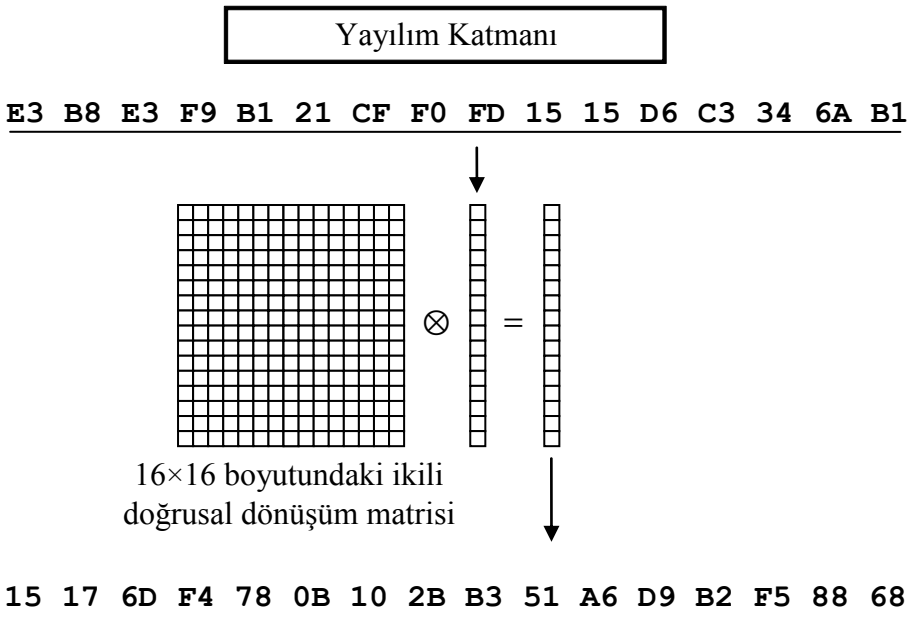
Doğrusal dönüşüm incelendiğinde 16×16 boyutundaki matrisin satır ve sütunlarının Hamming ağırlığının 7 olduğu görülebilir. Şifrede kullanılan involutif yayılım elemanın dallanma sayısı 8 ve sabit nokta sayısı 2^{64}'tür. Kullanılan doğrusal dönüşüm ifade (9.6)'da verilmiştir.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{bmatrix} \qquad (9.6)$$

İfade (9.6)'da verilen doğrusal dönüşüm XOR tabanlı olarak (9.7)'daki gibi yazılabilir.

$$
\begin{aligned}
y_0 &= x_0 \oplus x_4 \oplus x_6 \oplus x_9 \oplus x_{12} \oplus x_{13} \oplus x_{14} & y_8 &= x_1 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_8 \oplus x_{12} \oplus x_{14} \\
y_1 &= x_1 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_9 \oplus x_{10} \oplus x_{15} & y_9 &= x_1 \oplus x_2 \oplus x_7 \oplus x_9 \oplus x_{13} \oplus x_{14} \oplus x_{15} \\
y_2 &= x_2 \oplus x_4 \oplus x_6 \oplus x_7 \oplus x_{10} \oplus x_{11} \oplus x_{12} & y_{10} &= x_2 \oplus x_3 \oplus x_4 \oplus x_{10} \oplus x_{12} \oplus x_{14} \oplus x_{15} \\
y_3 &= x_3 \oplus x_5 \oplus x_7 \oplus x_8 \oplus x_{11} \oplus x_{12} \oplus x_{13} & y_{11} &= x_0 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_{11} \oplus x_{13} \oplus x_{15} \\
y_4 &= x_0 \oplus x_2 \oplus x_4 \oplus x_8 \oplus x_9 \oplus x_{10} \oplus x_{13} & y_{12} &= x_0 \oplus x_1 \oplus x_2 \oplus x_5 \oplus x_8 \oplus x_{10} \oplus x_{12} \\
y_5 &= x_1 \oplus x_2 \oplus x_3 \oplus x_5 \oplus x_{11} \oplus x_{13} \oplus x_{14} & y_{13} &= x_3 \oplus x_5 \oplus x_6 \oplus x_9 \oplus x_{10} \oplus x_{11} \oplus x_{13} \\
y_6 &= x_0 \oplus x_2 \oplus x_3 \oplus x_6 \oplus x_8 \oplus x_{14} \oplus x_{15} & y_{14} &= x_0 \oplus x_6 \oplus x_7 \oplus x_8 \oplus x_{10} \oplus x_{11} \oplus x_{14} \\
y_7 &= x_1 \oplus x_3 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{12} \oplus x_{15} & y_{15} &= x_0 \oplus x_1 \oplus x_4 \oplus x_7 \oplus x_9 \oplus x_{11} \oplus x_{15}
\end{aligned}
\tag{9.8}
$$

Örnek 9.1: **E3B8E3F9B121CFF0FD1515D6C3346AB1** 128-bitlik değerinin yayılım katmanından sonra **15176DF4780B102BB351A6D9B2F58868** değerine dönüştüğü Şekil 9.6'daki gibi gösterilebilir.



Şekil 9.6: Doğrusal Dönüşüm İşlemi

9.2.3. Anahtar Planlama

Tasarlanan şifrenin anahtar planlama evresi Şekil 9.6'daki gibidir. Anahtarlar üretilirken kullanılan döngü sabit değerleri (round constants) aşağıdaki gibidir:

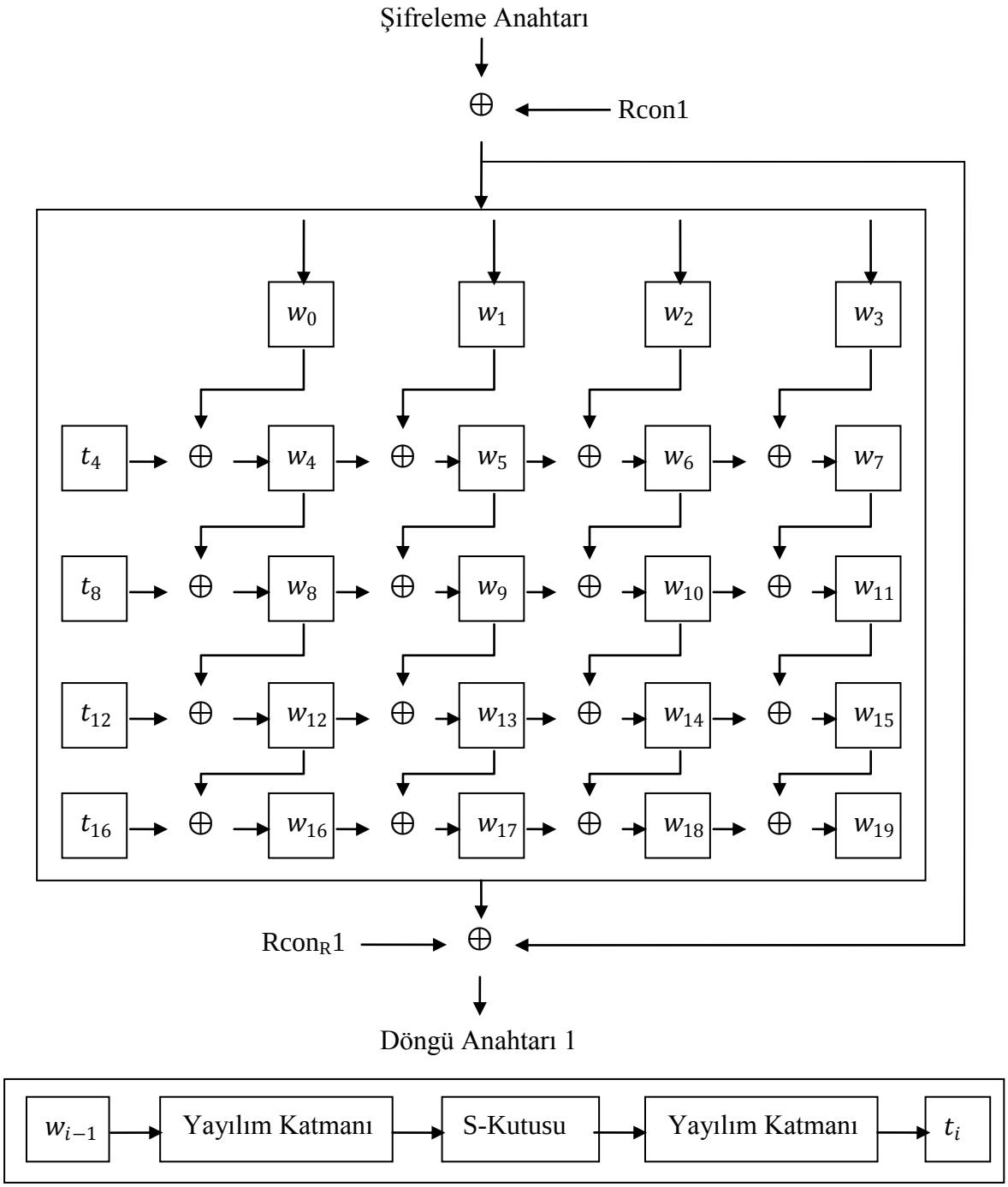
RCon1 :9CC35F046A78F579FC42EC142FCDB104
RCon2 :57EF4269CABDDEFC18046066798A03F2
RCon3 :68C9F104BDEFCC69204567983643567 5
RCon4 :F54830446978C9B5E679646209AB9CED
RCon5 :DF570249AEFCD51269334589ACDE0F14
RCon6 :9AF5667ED340E69FAB2C4D05328B8791
RCon7 :0F882DCE6524778A29F57E9A5326C5E0
RCon8 :3AB0E25DEFD5B128FF20C8219ECC4AB1
RCon9 :DBE0908C4E5077493292371D2126E970
RCon10 :187A8F4936E38867B5CB27009B152FCD

Şifre içerisinde *t* değerlerinin üretimi aşamasında kullanılan S-kutusu, şifrenin yayılım katmanında kullanılan S-kutusudur. Bunun yanında anahtar planlama evresinde kullanılan yayılım katmanı 4×4 boyutunda ikili bir doğrusal dönüşümdür. Bu yayılım elemanı ifade (9.8)'de verilmiştir.

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \tag{9.8}$$

Anahtar planlama aşamasında kullanılan 4×4 boyutundaki ikili doğrusal dönüşümün dallanma sayısı 4 ve sabit nokta sayısı 2'dir. Şifredeki tüm yapıların XOR tabanlı işlem yapabilmesi için anahtar planlama evresinde de ikili doğrusal dönüşüm tercih edilmiştir. (9.8)'de verilen doğrusal dönüşüm aşağıdaki şekilde de yazılabilir:

$$\begin{aligned} temp &= x_0 \oplus x_1 \oplus x_2 \oplus x_3 \\ y_0 &= temp \oplus x_0 \\ y_1 &= temp \oplus x_2 \\ y_2 &= temp \oplus x_3 \\ y_3 &= temp \oplus x_1 \end{aligned}$$



Şekil 9.7: Şifrenin anahtar planlama evresi

Şifreleme anahtarları (ek_i) Şekil 9.7'de verilen diyagram kullanılarak her Rcon değeri için 10 adet olarak üretilir. Şifre çözme anahtarları (dk_i) ise ifade (9.9)'da olduğu gibi şifrede kullanılan 16×16 boyutundaki ikili doğrusal dönüşüm ile oluşturulmuş yayılım katmanı (A) üzerinden geçirilmesi ile üretilir.

$$dk_1 = ek_{n+1},\ dk_1 = A(ek_n),\ dk_2 = A(ek_{n-1}),\ \ldots,\ dk_n = A(ek_1). \qquad (9.9)$$

Örnek 3: **5452414B5941554E4956455253495459** şifreleme anahtarı ile aşağıdaki şekilde döngü anahtarları üretilebilir.

Master Key	5452414B5941554E4956455253495459
Rcon	9CC35F046A78F579FC42EC142FCDB104
XOR	C8911E4F3339A037B514A9467C84E55D
	w0:C8911E4F w1:3339A037 w2:B514A946 w3:7C84E55D
t4	5F9874FB
	w4:97096AB4 w5:A430CA83 w6:112463C5 w7:6DA08698
t8	BFF6A736
	w5:28FFCD82 w6:8CCF0701 w7:9DEB64C4 w8:F04BE25C
t12	44130836
	w6:6CECC5B4 w7:E023C2B5 w8:7DC8A671 w9:8D83442D
t16	1656171D
	w7:7ABAD2A9 w8:9A99101C w9:E751B66D w10:6AD2F240
	7ABAD2A99A99101CE751B66D6AD2F240
Rev.Rcon	04B1CD2F14EC42FC79F5786A045FC39C
XOR	B69A01C9BD4CF2D72BB067411209D481
1.Döngü Anahtarı	B69A01C9BD4CF2D72BB067411209D481
Rcon	57EF4269CABDDEFC18046066798A03F2
XOR	03BD032293FC8BB2515225342AC357AB
	w0:03BD0322 w1:93FC8BB2 w2:51522534 w3:2AC357AB
t4	AAD4CE03
	w4:A969CD21 w5:3A954693 w6:6BC763A7 w7:4104340C
t8	BCE6E716
	w5:158F2A37 w6:2F1A6CA4 w7:44DD0F03 w8:05D93B0F
t12	264E5821
	w6:33C17216 w7:1CDB1EB2 w8:580611B1 w9:5DDF2ABE
t16	7D167C44
	w7:4ED70E52 w8:520C10E0 w9:0A0A0151 w10:57D52BEF
	4ED70E52520C10E00A0A015157D52BEF
Rev.Rcon	F2038A7966600418FCDEBDCA6942EF57
XOR	BF698709A7909F4AA78699AF14549313
2. Döngü Anahtarı	BF698709A7909F4AA78699AF14549313
Rcon	68C9F104BDEFCC692045679836435675
XOR	3C9BB04FE4AE9927691322CA650A022C
	w0:3C9BB04F w1:E4AE9927 w2:691322CA w3:650A022C
t4	E6AB4263
	w4:DA30F22C w5:3E9E6B0B w6:578D49C1 w7:32874BED
t8	6D3AA3D1

	w5:B70A51FD w6:89943AF6 w7:DE197337 w8:EC9E38DA
t12	8DCB0392
	w6:3AC1526F w7:B3556899 w8:6D4C1BAE w9:81D22374
t16	63AB7CB5
	w7:596A2EDA w8:EA3F4643 w9:87735DED w10:06A17E99
	596A2EDAEA3F464387735DED06A17E99
Rev.Rcon	755643369867452069CCEFBD04F1C968
XOR	10A7DDA396F69A4487AC909A675AB5DD
3. Döngü Anahtarı	10A7DDA396F69A4487AC909A675AB5DD
Rcon	F54830446978C9B5E679646209AB9CED
XOR	A11A710F30399CFBAF2F21305AE2C8B4
	w0:A11A710F w1:30399CFB w2:AF2F2130 w3:5AE2C8B4
t4	220FDD55
	w4:8315AC5A w5:B32C30A1 w6:1C031191 w7:46E1D925
t8	BB392CC5
	w5:382C809F w6:8B00B03E w7:9703A1AF w8:D1E2788A
t12	B98DF463
	w6:81A174FC w7:0AA1C4C2 w8:9DA2656D w9:4C401DE7
t16	D884BC7F
	w7:5925C883 w8:53840C41 w9:CE26692C w10:826674CB
	5925C88353840C41CE26692C826674CB
Rev.Rcon	ED9CAB09626479E6B5C97869443048F5
XOR	15A3128501D9E95CD4C030759CB4F48A
4. Döngü Anahtarı	15A3128501D9E95CD4C030759CB4F48A
Rcon	DF570249AEFCD51269334589ACDE0F14
XOR	8B054302F7BD805C206500DBFF975B4D
	w0:8B054302 w1:F7BD805C w2:206500DB w3:FF975B4D
t4	A0A5EDE1
	w4:2BA0AEE3 w5:DC1D2EBF w6:FC782E64 w7:03EF7529
t8	6DDEB1CA
	w5:467E1F29 w6:9A633196 w7:661B1FF2 w8:65F46ADB
t12	C8A672CA
	w6:8ED86DE3 w7:14BB5C75 w8:72A04387 w9:1754295C
t16	354BE1E2
	w7:BB938C01 w8:AF28D074 w9:DD8893F3 w10:CADCBAAF
	BB938C01AF28D074DD8893F3CADCBAAF
Rev.Rcon	140FDEAC8945336912D5FCAE490257DF
XOR	249911AFD1D06341EF386F867C49B63D
5. Döngü Anahtarı	249911AFD1D06341EF386F867C49B63D
Rcon	9AF5667ED340E69FAB2C4D05328B8791
XOR	CEA727358A01B3D1E27A085761C2D3C8
	w0:CEA72735 w1:8A01B3D1 w2:E27A0857 w3:61C2D3C8
t4	ACB43E38
	w4:6213190D w5:E812AADC w6:0A68A28B w7:6BAA7143

t8	6395588C
	w5:01864181 w6:E994EB5D w7:E3FC49D6 w8:88563895
t12	2AB6B274
	w6:2B30F3F5 w7:C2A418A8 w8:2158517E w9:A90E69EB
t16	574CED83
	w7:7C7C1E76 w8:BED806DE w9:9F8057A0 w10:368E3E4B
	7C7C1E76BED806DE9F8057A0368E3E4B
Rev.Rcon	91878B32054D2CAB9FE640D37E66F59A
XOR	235CB271319499A4E21C1F24292A1819
6. Döngü An	235CB271319499A4E21C1F24292A1819
Rcon	0F882DCE6524778A29F57E9A5326C5E0
XOR	5BDA6C853C6522C460A33BC8006F91B9
	w0:5BDA6C85 w1:3C6522C4 w2:60A33BC8 w3:006F91B9
t4	8B00F662
	w4:D0DA9AE7 w5:ECBFB823 w6:8C1C83EB w7:8C731252
t8	44DF1467
	w5:94058E80 w6:78BA36A3 w7:F4A6B548 w8:78D5A71A
t12	FF71188C
	w6:6B74960C w7:13CEA0AF w8:E76815E7 w9:9FBDB2FD
t16	B7D21419
	w7:DCA68215 w8:CF6822BA w9:2800375D w10:B7BD85A0
	DCA68215CF6822BA2800375DB7BD85A0
Rev.Rcon	E0C526539A7EF5298A772465CE2D880F
XOR	67B9C8C36973F557C2D428F079FF9C16
7. Döngü Anahtarı	67B9C8C36973F557C2D428F079FF9C16
Rcon	3AB0E25DEFD5B128FF20C8219ECC4AB1
XOR	6EE2A316B694E466B6768D73CD851EE8
	w0:6EE2A316 w1:B694E466 w2:B6768D73 w3:CD851EE8
t4	7C634C50
	w4:1281EF46 w5:A4150B20 w6:12638653 w7:DFE698BB
t8	530AF7FF
	w5:418B18B9 w6:E59E1399 w7:F7FD95CA w8:281B0D71
t12	1C7ED269
	w6:5DF5CAD0 w7:B86BD949 w8:4F964C83 w9:678D41F2
t16	6A2269EC
	w7:37D7A33C w8:8FBC7A75 w9:C02A36F6 w10:A7A77704
	37D7A33C8FBC7A75C02A36F6A7A77704
Rev.Rcon	B14ACC9E21C820FF28B1D5EF5DE2B03A
XOR	E87FCCB418E0BEEC5EED6E6A37C0D9D6
8. Döngü Anahtarı	E87FCCB418E0BEEC5EED6E6A37C0D9D6
Rcon	DBE0908C4E5077493292371D2126E970
XOR	8FB2D1C7171122077BC4724F726FBD29
	w0:8FB2D1C7 w1:17112207 w2:7BC4724F w3:726FBD29
t4	B7942BCF

	w4:3826FA08 w5:2F37D80F w6:54F3AA40 w7:269C1769
t8	4B1F64A0
	w5:73399EA8 w6:5C0E46A7 w7:08FDECE7 w8:2E61FB8E
t12	BE9A1C49
	w6:CDA382E1 w7:91ADC446 w8:995028A1 w9:B731D32F
t16	688646C0
	w7:A525C421 w8:34880067 w9:ADD828C6 w10:1AE9FBE9
	A525C42134880067ADD828C61AE9FBE9
Rev.Rcon	70E926211D3792324977504E8C90E0DB
XOR	5A7E33C73EAEB0529F6B0AC7E416A61B
9. Döngü Anahtarı	5A7E33C73EAEB0529F6B0AC7E416A61B
Rcon	187A8F4936E38867B5CB27009B152FCD
XOR	4C28CE026FA2DD29FC9D6252C85C7B94
	w0:4C28CE02 w1:6FA2DD29 w2:FC9D6252 w3:C85C7B94
t4	12D20374
	w4:5EFACD76 w5:3158105F w6:CDC5720D w7:05990999
t8	347373FE
	w5:6A89BE88 w6:5BD1AED7 w7:9614DCDA w8:938DD543
t12	0F467B7D
	w6:65CFC5F5 w7:3E1E6B22 w8:A80AB7F8 w9:3B8762BB
t16	DDEEC4E8
	w7:B821011D w8:863F6A3F w9:2E35DDC7 w10:15B2BF7C
	B821011D863F6A3F2E35DDC715B2BF7C
Rev.Rcon	CD2F159B0027CBB56788E336498F7A18
XOR	3926DA84E9BA7CA3B5205CA39461BEF0
10. Döngü Anahtarı	3926DA84E9BA7CA3B5205CA39461BEF0

9.2.4. Örnek Şifreleme İşlemleri

Açık Metin: **5452414B5941554E4956455253495459**

Şifreleme Anahtarı: **46DC53554E5941424F524141534C414E**

Şifreleme için üretilen döngü anahtarları:

Döngü	Şifreleme Anahtarları
1	F769690F1E7D3F1A2699488F055839C1
2	C61AA30C545DAA072A1253D7D4095B4A
3	3E529EF1700B790D3E48842D5960007B
4	9E57B504EB4E6593F32156FC425C424D
5	D73302D88FC80144F97DDDAA9E8FBB0F
6	770F945B92E62550ABC0E86C00603F31
7	552CBE1479BE44E05B8F6AC14690B09F

8	6FCDAAF582C4CCB03319AE3435DD6209
9	6D70521A09D8EE6BE2D356C4BCC1877F
10	757C2FD72068FAD993903955656BE813

Şifre çözme için üretilen döngü anahtarları:

Döngü	Şifre Çözme Anahtarları
1	757C2FD72068FAD993903955656BE813
2	6D70521A09D8EE6BE2D356C4BCC1877F
3	6FCDAAF582C4CCB03319AE3435DD6209
4	552CBE1479BE44E05B8F6AC14690B09F
5	770F945B92E62550ABC0E86C00603F31
6	D73302D88FC80144F97DDDAA9E8FBB0F
7	9E57B504EB4E6593F32156FC425C424D
8	3E529EF1700B790D3E48842D5960007B
9	C61AA30C545DAA072A1253D7D4095B4A
10	F769690F1E7D3F1A2699488F055839C1

Şifreleme Adımları:

Döngü	İşlem	Veri
	Açık Metin	5452414B5941554E4956455253495459
	Şifreleme Anahtarı	46DC53554E5941424F524141534C414E
	Anahtar Ekleme	128E121E1718140C0604041300051517
1	Yer Değiştirme	E3B8E3F9B121CFF0FD1515D6C3346AB1
	Yayılım Katmanı	15176DF4780B102BB351A6D9B2F58868
	1. Döngü Anahtarı	F769690F1E7D3F1A2699488F055839C1
	Anahtar Ekleme	E27E04FB66762F3195C8EE56B7ADB1A9
2	Yer Değiştirme	DB5215706351BB49B93BEA60968FCCA8
	Yayılım Katmanı	ED8898A84A45D8D738DE1EDE08A4D650
	2. Döngü Anahtarı	C61AA30C545DAA072A1253D7D4095B4A
	Anahtar Ekleme	2B923BA41E1872D012CC4D09DCAD8D1A
3	Yer Değiştirme	988B4F29F921507CE3FC3EFE3F8FDC9D
	Yayılım Katmanı	A1D965D980610C63038ADF859FEB4B09
	3. Döngü Anahtarı	3E529EF1700B790D3E48842D5960007B
	Anahtar Ekleme	9F8BFB28F06A756E3DC25BA8C68B4B72
4	Yer Değiştirme	550C70E9F836B607CDD26D45FB0C2E50
	Yayılım Katmanı	1064EAA7A3C4C9566CDB896B449F2F61
	4. Döngü Anahtarı	9E57B504EB4E6593F32156FC425C424D
	Anahtar Ekleme	8E335FA3488AACC59FFADF9706C36D2C
5	Yer Değiştirme	B8AC79CB74449202550E7103FDBCC768
	Yayılım Katmanı	D66F129A232262AB61CAE594F0DDC807
	5. Döngü Anahtarı	D73302D88FC80144F97DDDAA9E8FBB0F

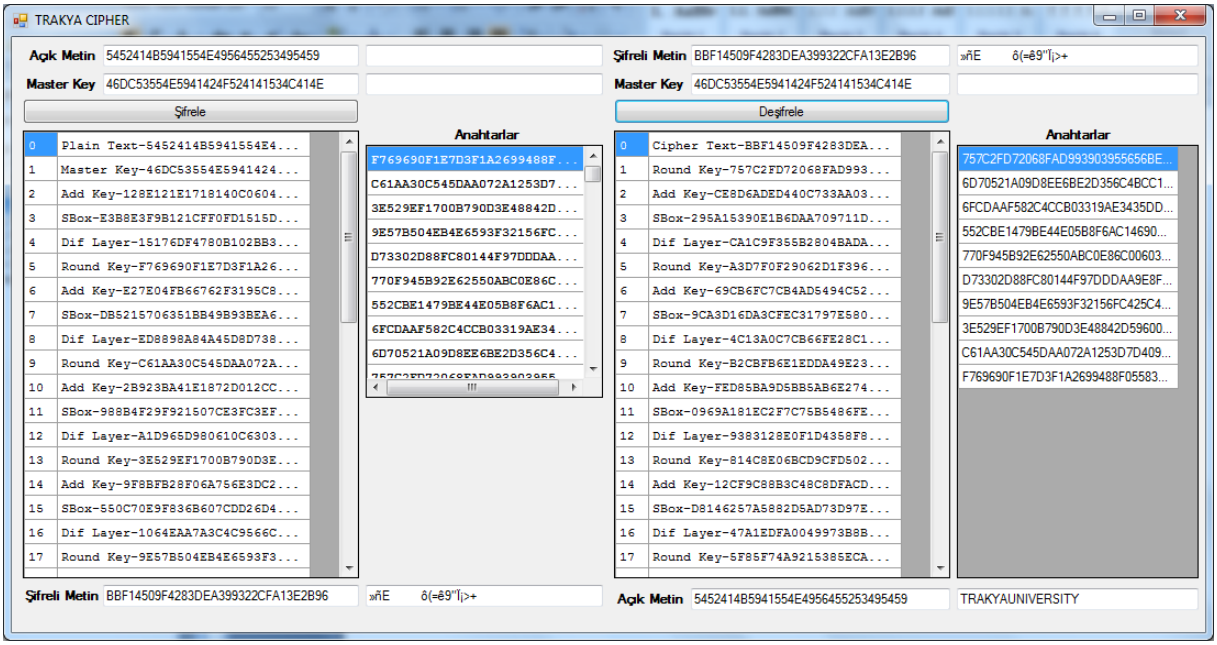
	Anahtar Ekleme	015C1042ACEA63EF98B7383E6E527308
6	Yer Değiştirme	18241AB0925CAF65419614A707647F2B
	Yayılım Katmanı	AF1BF60C376E080A7CFD7F8E28025FD1
	6. Döngü Anahtarı	770F945B92E62550ABC0E86C00603F31
	Anahtar Ekleme	D8146257A5882D5AD73D97E2286260E0
7	Yer Değiştirme	12CF9C88B3C48C8DFACD03DBE99CC981
	Yayılım Katmanı	5C451F9595913895EEC7052B9549F874
	7. Döngü Anahtarı	552CBE1479BE44E05B8F6AC14690B09F
	Anahtar Ekleme	0969A181EC2F7C75B5486FEAD3D948EB
8	Yer Değiştirme	FED85BA9D5BB5AB6E274755C30B274E0
	Yayılım Katmanı	F36E7B98210B20814A67F63761A7AD8D
	8. Döngü Anahtarı	6FCDAAF582C4CCB03319AE3435DD6209
	Anahtar Ekleme	9CA3D16DA3CFEC31797E5803547ACF84
9	Yer Değiştirme	69CB6FC7CB4AD5494C52F4800B204A91
	Yayılım Katmanı	442A472307C383C19244471E345E8763
	9. Döngü Anahtarı	6D70521A09D8EE6BE2D356C4BCC1877F
	Anahtar Ekleme	295A15390E1B6DAA709711DA889F001C
10	Yer Değiştirme	CE8D6ADED440C733AA031B9AC455C385
	10. Döngü Anahtarı	757C2FD72068FAD993903955656BE813
	Anahtar Ekleme	BBF14509F4283DEA399322CFA13E2B96
	Şifreli Metin	BBF14509F4283DEA399322CFA13E2B96

Şifre Çözme Adımları:

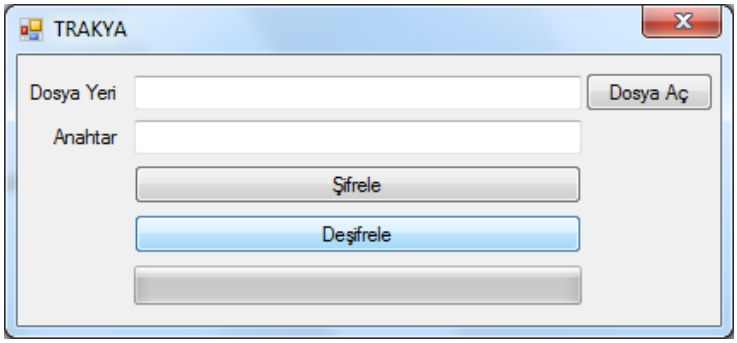
Döngü	**İşlem**	**Veri**
	Açık Metin	BBF14509F4283DEA399322CFA13E2B96
1	1. Döngü Anahtarı	757C2FD72068FAD993903955656BE813
	Döngü Anahtarı	CE8D6ADED440C733AA031B9AC455C385
	Yer Değiştirme	295A15390E1B6DAA709711DA889F001C
	Yayılım Katmanı	CA1C9F355B2804BADAF1A75C948C5586
2	2. Döngü Anahtarı	A3D7F0F29062D1F396A353DC9FAC1F17
	Anahtar Ekleme	69CB6FC7CB4AD5494C52F4800B204A91
	Yer Değiştirme	9CA3D16DA3CFEC31797E5803547ACF84
	Yayılım Katmanı	4C13A0C7CB66FE28C10C58605411AC54
3	3. Döngü Anahtarı	B2CBFB6E1EDDA49E23782D3C64A3D8B4
	Anahtar Ekleme	FED85BA9D5BB5AB6E274755C30B274E0
	Yer Değiştirme	0969A181EC2F7C75B5486FEAD3D948EB
	Yayılım Katmanı	9383128E0F1D4358F88FD393E7C678B0
4	4. Döngü Anahtarı	814C8E06BCD9CFD50242D0480E5AB131
	Anahtar Ekleme	12CF9C88B3C48C8DFACD03DBE99CC981
	Yer Değiştirme	D8146257A5882D5AD73D97E2286260E0
	Yayılım Katmanı	47A1EDFA0049973B8BF3AFC24ED86D0C
5	5. Döngü Anahtarı	5F85F74A9215385ECA65BB6549BC1227
	Anahtar Ekleme	18241AB0925CAF65419614A707647F2B
	Yer Değiştirme	015C1042ACEA63EF98B7383E6E527308

	Yayılım Katmanı	36BD58DDF8FBD3B8FC3DD3616928609F
6	6. Döngü Anahtarı	8E1121168CBF41BAA933A2629494A7F7
	Anahtar Ekleme	B8AC79CB74449202550E7103FDBCC768
	Yer Değiştirme	8E335FA3488AACC59FFADF9706C36D2C
	Yayılım Katmanı	38D93021E07C001AA9D12C972EF45D71
7	7. Döngü Anahtarı	6DD540C8184AB61D640341D2D5F87321
	Anahtar Ekleme	550C70E9F836B607CDD26D45FB0C2E50
	Yer Değiştirme	9F8BFB28F06A756E3DC25BA8C68B4B72
	Yayılım Katmanı	DE1125F4BB5A3D86D46E877C258D0192
8	8. Döngü Anahtarı	469A6ADD427B6DFA3792B9821A02DD0F
	Anahtar Ekleme	988B4F29F921507CE3FC3EFE3F8FDC9D
	Yer Değiştirme	2B923BA41E1872D012CC4D09DCAD8D1A
	Yayılım Katmanı	77B31F06303C43FEA58F873719EB52A8
9	9. Döngü Anahtarı	ACE10A76536DF8B71CB46D578F649E00
	Anahtar Ekleme	DB5215706351BB49B93BEA60968FCCA8
	Yer Değiştirme	E27E04FB66762F3195C8EE56B7ADB1A9
	Yayılım Katmanı	C89973659EBDBFF7D236D85B227F60FC
10	10. Döngü Anahtarı	2B21909C2F9C70072F23CD8DE14B0A4D
	Anahtar Ekleme	E3B8E3F9B121CFF0FD1515D6C3346AB1
	Yer Değiştirme	128E121E1718140C0604041300051517
	Şifre Çözme Anahtarı	46DC53554E5941424F524141534C414E
	Anahtar Ekleme	5452414B5941554E4956455253495459
Açık Metin		5452414B5941554E4956455253495459

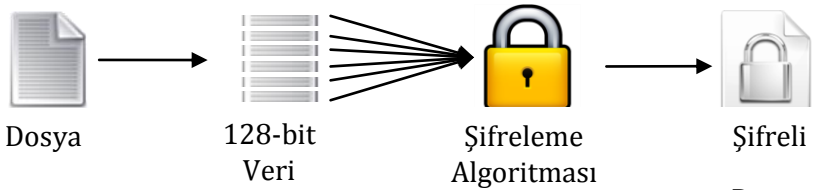
Şekil 9.8'de geliştirilen blok şifre uygulamasının test ara yüzü görülmektedir. Bunun yanında dosya şifrelemek için Şekil 9.9'da verilen yazılım geliştirilmiştir. Bu yazılımın çalışma şekli Şekil 9.10'da verilmiştir.



Şekil 9.8: Şifreleme yazılımının ara yüzü



Şekil 9.9: Dosya Şifreleme yazılımının ara yüzü



Şekil 9.10: Dosya şifreleme işlemi

9.3. Geliştirilen Blok Şifrenin Diferansiyel ve Doğrusal Kriptanalize Karşı Güvenlik Değerlendirmesi

Bölüm 2'de verilen tanım ve alt yapıdan faydalanarak geliştirilen blok şifrenin 2 döngüsü için aktif S-kutusu sayısı 8 olarak elde edilebilir ve TRAKYA blok şifresinde kullanılan S-kutusu, AES S-kutusu ile aynı kriptografik özelliklere sahiptir. Buna ek olarak cebirsel ifadesinde maksimum terim sayısı içermektedir ve dolayısıyla AES şifresinde kullanılan S-kutusuna göre cebirsel ifadesi daha karmaşıktır. TRAKYA blok şifresindeki S-kutusunun maksimum diferansiyel ve doğrusal olasılıkları 2^{-6} (Maximum differential and linear probability) olarak Bölüm 2'deki ifadeler kullanılarak elde edilebilir. Diğer yandan TRAKYA blok şifresinin 2 döngüsü için maksimum diferansiyel ve doğrusal olasılıkları $\left(2^{-6}\right)^8 = 2^{-48}$ değeri ile sınırlıdır. *2r*-döngüye sahip bir SPN tabanlı blok şifre için diferansiyel ve doğrusal karakteristiklerinin maksimum olasılıkları aşağıdaki gibi verilebilir:

$$p_D^{2r} \leq \left(2^{-6}\right)^{(r\times B_A)},\ \mathrm{q}_L^{2r} \leq \left(2^{-6}\right)^{(r\times B_A)}$$

Bu bağlamda 6 döngüden oluşan TRAKYA blok şifresinde minimum aktif S-kutusu sayısı (r = 3) 3 × 8 = 24 olarak elde edilebilir. 128-bit anahtar büyüklüğüne sahip TRAKYA blok şifresi için;

$$p_D^6 = \left(2^{-6}\right)^{24} = 2^{-144} \leq 2^{-128},\ q_L^6 = \left(2^{-6}\right)^{24} = 2^{-144} \leq 2^{-128}$$

olarak elde edilir. Dolayısıyla TRAKYA blok şifresi 6 döngüde diferansiyel ve doğrusal kriptanalize karşı dayanıklıdır. TRAKYA blok şifresinin döngü sayısı 10 olarak seçilmiştir. Elbetteki tasarlanan şifreleme algoritmasını önemli uygulamalarda kullanılmadan önce; imkânsız diferansiyel kriptanaliz, kesik diferansiyel kriptanaliz ve biclique kriptanaliz gibi önemli saldırılara karşı da gücünün test edilmesi gerekmektedir.

9.4. Sonuçlar

Tezin bu bölümünde, daha önceki bölümlerde elde edilen bir örnek doğrusal dönüşümün bir blok şifreye uygulaması gerçekleştirilmiştir. Geliştirilen örnek blok şifre uygulamasında ARIA ve AES blok şifrelerinden esinlenilmiştir. Bu uygulamada geliştirilen blok şifrenin 6 döngüde doğrusal ve diferansiyel saldırılara karşı dayanıklı olduğu gösterilmiştir. Not edilmelidir ki yeni geliştirilen blok şifre uygulaması 10 döngüdür. Ancak blok şifrenin diğer önemli saldırılar da göz önüne alınarak incelenmesi sonucu pratik kullanımı mümkün hale gelebilir.

BÖLÜM 10

SONUÇLAR

Bu tezde 8×8 ve 16×16 boyutunda kriptografik özellikleri iyi ikili doğrusal dönüşümlerin tasarımı için cebirsel bir yöntem geliştirilmiştir. Tasarlanan ikili doğrusal dönüşümler yüksek dallanma sayısı, minimum sabit nokta sayısına sahip olma ve involutif olma gibi kriptografik özelliklere sahiptir. Bunun yanında uygulama özelliklerinin iyi olması, bu doğrusal dönüşümlerin bir blok şifre geliştirmede yazılım veya donanım uygulamalarında kullanılabileceğini de göstermektedir. Özellikle tasarımı yapılan ikili doğrusal dönüşümlerin 8-bit, 32-bit ve 64-bit işlemciler için uygun uygulama özellikleri bulunmaktadır. Bunun nedeni olarak tasarlanan ikili doğrusal dönüşümlerin $GF(2^4)$ cisim elemanlarına sahip 2×2 ve 4×4 matrisler tabanlı olması, herhangi bir $GF(2^4)$ cisim elemanının 4×4 boyutunda ikili bir matrise karşılık gelmesi ve kullanılan 2×2 ve 4×4 matrislerin sonlu cisim Hadamard veya dairesel formda olması gösterilebilir. Tezde geliştirilen ikili doğrusal dönüşümler ve kriptografik özellikleri aşağıdaki gibidir:

a) 8×8 boyutunda dallanma sayısı 5 ve sabit nokta sayısı 1 (involutif değil),
b) 8×8 boyutunda dallanma sayısı 5 ve sabit nokta sayısı 2^4 (involutif),
c) 16×16 boyutunda dallanma sayısı 8 ve sabit nokta sayısı 2 (involutif değil),
d) 16×16 boyutunda dallanma sayısı 8 ve sabit nokta sayısı 2^8 (involutif),
e) 16×16 boyutunda dallanma sayısı 7 ve sabit nokta sayısı 1 (involutif değil).

Not edilmelidir ki bu doğrusal dönüşümler her giriş değeri byte değer olacak şekilde 64-bit'ten 64 bit'e ya da 128-bit'ten 128 bit'e haritalama yapacak şekilde kullanılırlarsa doğrusal dönüşümün içereceği sabit nokta sayısı değişecektir. Örneğin (d) seçeneğinde verilen doğrusal dönüşüm tipi bir blok şifrede kullanılırsa 128-bit'ten 128 bit'e haritalama yapacak doğrusal dönüşüm $2^{8\times 8} = 2^{64}$ sabit nokta sayısı içerecektir. Bu doğrusal dönüşümün içerdiği sabit nokta sayısı ARIA blok şifresinde kullanılan aynı dallanma sayısına sahip involutif doğrusal dönüşümünün içerdiği sabit nokta sayısından 2^8 kat daha azdır.

Geliştirilen yöntemle elde edilen ikili doğrusal dönüşümler aynı zamanda MDBL koddur. Çünkü maksimum dallanma sayısı değerine sahiptirler. Ancak (e) seçeneğinde gösterildiği gibi dallanma sayısı 7 olan 16×16 boyutunda tasarlanan ikili matrisler MDBL kod değildir ancak rastsal bir doğrusal dönüşümün sahip olması gereken sabit nokta sayısına sahiptirler (bu değer 1'dir). Bu yüzden bu tür kodlara tarafımızdan yaklaşık MDBL kodlar ismi verilmiştir.

Tez çalışması esnasında yöntemimiz 16×16 boyutunda dallanma sayısı 8 ve 1 sabit noktaya sahip ikili doğrusal dönüşümlerin tasarımında başarısız olmuştur. Dolayısıyla deneysel sonuçlardan yola çıkarak 16×16 boyutunda dallanma sayısı 8 ve 1 sabit noktaya sahip ikili doğrusal dönüşümlerin olmadığı kanaatini taşımaktayız. Yine tez çalışması esnasında 4×4 ve 5×5 boyutunda (her iki boyut için maksimum dallanma sayısı değer 4'tür) tüm ikili matrisler taranarak maksimum dallanma sayısına ve 1 sabit nokta sayısına sahip ikili matrisler aranmıştır. 5×5 boyutundaki ikili matrisler için bu arama başarılı olurken, 4×4 boyutunda ikili matrisler için bu arama başarılı olmamıştır. Yani 4×4 boyutunda dallanma sayısı değeri 4 ve sabit nokta sayısı 2 olan matrisler bulunabilmiştir. Dolayısıyla bu bulgularda bir önceki kanaatimizi güçlendirmektedir. Sonuç olarak geliştirdiğimiz yöntem ve deneysel sonuçlar bir ikili doğrusal dönüşüm için maksimum dallanma sayısı, bu dönüşüm içerdiği bir rastsal doğrusal dönüşümde olması gereken sabit nokta sayısı ve ikili matrisin boyutu arasında bir ilişki olduğu kanaatine varmamıza neden olmuştur.

Geliştirilen yöntem ile elde edilen ikili doğrusal dönüşümlerin büyük oranı MDS matrislerden elde edilmiştir. Literatürde ilk defa MDS matrisler ile MDBL kodlar arasında bir ilişkinin gözlenmesi açısından tezde sunulan bulgular önemlidir. Diğer yandan geliştirilen yöntemin 2'nin üssü olan boyutlar için geliştirilen maksimum dallanma sayısına sahip ikili doğrusal dönüşümlerin tasarlanmasında başarılı olduğu deneysel sonuçlar ile gözlenmiştir. Ancak bu yöntem farklı genişletilmiş cisimlerde kullanılarak yüksek dallanma sayısına sahip ikili matrislerin elde edilmesine olanak verebilir. Örneğin 2×2 boyutunda elemanları $GF(2^5)$ cismine ait matrisler 10×10 boyutunda yüksek dallanma sayısına sahip ikili matrislerin elde edilmesi için kullanılabilir. Tez çalışması esmasında bu tür matrislerin elde edilmesi içinde bir çalışma yapılmış ve dallanma sayısı değeri 5 olan ikili matrisler elde edilmiştir. Ancak bu boyuttaki matrisler için maksimum dallanma sayısı değeri 6 olan ikili matrisler elde

edilememiştir. Yine de gelecekte hafif (light weight) kriptografik yapıların tasarımında kullanılabilecek doğrusal dönüşümlerin geliştirilmesinde bu çalışmanın bir yol gösterici olması mümkün gözükmektedir.

Daha büyük boyutta (örneğin 32×32 boyutunda) ikili matrislerin elde edilmesinde elemanları $GF(2^4)$ cismine ait 8×8 boyutunda sonlu cisim Hadamard veya dairesel matrisler kullanılabilir. Bu matrislerin yüksek dallanma sayısına sahip olanları ile bilinen ve alt sınırı (lower bound) 12 olan yüksek dallanma sayısına sahip ikili matrislerin tasarlanabileceği yine tez çalışması esnasında gözlenmiştir. Not edilmelidir ki elemanları $GF(2^4)$ cismine ait 8×8 boyutunda matrislerin maksimum dallanma sayısı değeri 8'dir. Tez çalışması esnasında gözlenen bir diğer durum ise bu bahsedilen ikili matrislerin elde edilmesinde dallanma sayısı 7 veya 8 olan elemanları $GF(2^4)$ cismine ait 8×8 boyutunda matrislerden faydalanılabileceğidir.

Tez çalışması esnasında polinom taban (veya standard taban) kullanılarak elde edilen 4×4 boyutundaki ikili matrisler temel alınarak çeşitli boyutlarda kriptografik özellikleri iyi ikili matrisler üretilmiştir. Bunun yanında belirlenen kriterlere göre üretilen ikili matrislerin sayısının arttırılması için 4×4 ikili matrislerin normal taban gösterimlerinden de faydalınabilinir.

Tezin 9. Bölümünde tez esnasında geliştirilen cebirsel yöntem kullanılarak tasarlanan 16×16 boyutunda involutif ve dallanma sayısı 8 olan bir ikili doğrusal dönüşüm kullanılarak 128-bit blok ve anahtar uzunluğuna sahip bir blok şifre geliştirilmiştir ve bu blok şifreye TRAKYA ismi verilmiştir. TRAKYA blok şifresi, tablo okumaları dışında tamamen XOR işlem tabanlı bir şifredir. AES ve ARIA blok şifrelerinden esinlenilerek geliştirilmiştir. Ayrıca 10 döngüden oluşmaktadır ve 6 döngüde diferansiyel ve doğrusal saldırılara karşı dayanıklı olduğu da gösterilmiştir. TRAKYA blok şifresinin gerçek uygulamalarda kullanılabilmesi için literatürde bulunan bazı diğer saldırılara karşı direncinin incelenmesi ve buna göre döngü sayısının tekrar gözden geçirilmesi gerekebilir.

KAYNAKLAR

1. A. Forouzan, ***Cryptography and Network Security***, International Edition, 2008.
2. D. Khan, ***The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet***, Scribner, 1996.
3. S. Singh, ***The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography***, Anchor, 2000.
4. FIPS 46-3, ***Data Encryption Standard, Federal Information Processing Standard (FIPS)***, Publication 46-3, National Bureau of Standards, U.S. Department of Commerce, Washington D.C, 1999.
5. A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, C. Vikkelsoe. ***PRESENT: An ultra-lightweight block cipher***. In Cryptographic Hardware and Embedded Systems-CHES 2007, 450-466, Springer Berlin Heidelberg, 2007.
6. A. Ross, E. Biham, and L. Knudsen. ***Serpent: A proposal for the advanced encryption Standard***, NIST AES Proposal (1998).
7. FIPS 197, ***Advanced Encryption Standard, Federal Information Processing Standard (FIPS)***, Publication 197, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., 2001.
8. K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, T. Tokita, ***Camellia: A 128-bit block cipher suitable for multiple platforms-design and analysis,*** SAC 2000. Lecture Notes in Computer Science, Vol. 2012, pp. 39–56. Springer, Heidelberg, 2000
9. D. Kwon, J. Kim, S. Park, S.H. Sung, Y. Sohn, J.H. Song, Y. Yeom, E-J. Yoon, S, Lee, J. Lee, S. Chee, D. Han, and J. Hong, ***New block cipher: ARIA***, Proceedings of International Conference on Information Security and Cryptology, Lecture Notes in Computer Science, Vol. 2971, pp. 432-445, Springer-Verlag, 2004.
10. P. S. L. M. Barreto and V. Rijmen, ***The Khazad legacy-level block cipher***, First open NESSIE Workshop, Leuven, 2000.
11. B. Schneier, ***Applied Cryptography - Protocols, Algorithms, and Source code in C***, John Wiley & Sons, Inc., 2nd edition, 1996.

12. C. De Cannière and B. Preneel, ***The Stream Cipher Trivium***, eSTREAM, the ECRYPT Stream Project., 2005.
13. H. Wu, ***The Stream Cipher HC-256, eSTREAM***, the ECRYPT Stream Project., 2005.
14. N. Koblitz, ***Elliptic curve cryptosystems***, Mathematics of Computation, 48 (177), 203–209, 1987.
15. H. Dobbertin, A. Bosselaers, B. Preneel, ***RIPEMD-160: A strengthened version of RIPEMD***, Fast Software Encryption, Lecture Notes in Computer Science, Vol. 1039, ed. D. Gollmann. Springer-Verlag, Berlin, 71–82, 1996.
16. C.E. Shannon, ***Communication Theory of Secrecy Systems***, Bell System Technical Journal, No. 30, pp. 50-64, 1949.
17. L. Keliher, ***Linear Cryptanalysis of Substitution-Permutation Networks***, Doktora Tezi, 2003.
18. B. Örencik, ***Matematiksel Kriptanaliz***, http://web.itu.edu.tr/~orencik/AgGuvenligi2007Sunumlari/karakoc_matkrip_rapor.pdf, Son erişim tarihi 07 Mayıs 2013.
19. M. T. Sakallı, ***Modern Şifreleme Yöntemlerinin Gücünün İncelenmesi***, Doktora tezi, 2006.
20. M. Matsui, ***Linear Cryptanalysis Method for DES Cipher***, Advances in Cryptology- EUROCRYPT' 93, Springer-Verlag, pp. 386-397, 1994.
21. E. Biham and A. Shamir, ***Differential Cryptanalysis of DES-like Cryptosystems***, Journal of Cryptology, Vol 4, No 1 pp. 3-72, 1991.
22. R.C.-W. Phan, ***Impossible Differential Cryptanalysis of 7-round Advanced Encryption Standard (AES)***, Information Processing Letters, Vol. 91, Issue 1, pp. 33-38, 2004.
23. L. R. Knudsen, ***Truncated and higher order differentials***, Fast Software Encryption—FSE '95, 2nd Internat. Workshop (B. Preneel, ed.), Leuven, 1995, Lecture,Notes in Comput. Sci., Vol. 1008, pp. 196–211,Springer-Verlag, Berlin, 1995.
24. M. T. Sakallı, E. Buluş, A. Şahin, F. Büyüksaraçoğlu, ***Bir Blok Şifreleme Algoritmasına karşı Square Saldırısı***, Ağ ve Bilgi Güvenliği Ulusal Sempozyumu, ABG-2005, Istanbul-Türkiye, 2005.

25. T. Jakobsen, L. Knudsen, ***The interpolation attack on block ciphers***, In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 28–40. Springer, Heidelberg,1997.
26. D. Wagner, ***The boomerang attack***, In Lars Ramkilde Knudsen, editor, Proceedings of Fast Software Encryption – FSE'99, number 1636 in Lecture Notes in Computer Science, 156–170. Springer-Verlag, 1999.
27. E. Biham, O. Dunkelman, N. Keller, ***The rectangle attack – rectangling the Serpent***, in Proceedings of Eurocrypt'01 (B. Pfitzmann, ed.), no. 2045 in LNCS, pp. 340–357, Springer-Verlag, 2001.
28. J. Kelsey, B. Schneier, D. Wagner, and C. Hall, ***Side channel cryptanalysis of product ciphers***, Proc. of ESORICS'98, LNCS, vol. 1485, eds. Quisquater, Deswarte, Meadows, and Gollmann. Springer-Verlag, Louvain la Neuve, Belgium, 97–110, 1998.
29. R. Lidl, H. Niederreiter, ***Introduction to finite fields and their applications***, Revised edition, Cambridge University Press, 1994.
30. D. R.Stinson, ***Cryptography, Theory and Practice***, 2nd ed., Chapman & Hall/CRC Press, 2002.
31. S.Ling, C. Xing, ***Coding Theory: A First Course***, Cambridge University Press, 2004.
32. K. Chun, S. Kim, S. Lee, S. H. Sung, S.Yoon, ***Differential and Linear cryptanalysis for 2-round SPNs***, Information Processing Letters, Elsevier, 2002.
33. D. Kwon, S.H. Sung, J. H. Song, S. Park, ***Design of Block Ciphers and Coding Theory***, Trends in Mathematics, Vol. 8, n. 1, pp. 13-20, 2005.
34. Jr J. Nakahara, E. Abrahão, ***A new Involutory MDS Matrix for the AES***, International Journal of Network Security, Vol. 9, n. 2, pp. 109-116, 2009.
35. J. Daemen, V. Rijmen, ***The Design of Rijndael: AES (The Advanced Encryption Standard)***, Springer, Berlin (2002).
36. B. Aslan, M. T. Sakallı, ***Algebraic construction of cryptographically good binary linear transformations,*** Security and Communication Networks. doi: 10.1002/sec.556, 2012
37. A.M. Youssef, S. Mister, S.E. Tavares, ***On the design of linear transformations for substitution permutation encryption Networks***, In Proceedings of Selected Areas in Cryptography (SAC'97). Ottawa, Ontario, Canada, ; 40-48, 1997

38. M. R. Z'aba, ***Analysis of Linear Relationships in Block Ciphers***, Queensland University of Technology, Australia, Doktora Tezi, 2010.
39. W. Bosma, J. Cannon, and C. Playoust, ***The Magma Algebra System I: The User Language***, J. Symbolic Comput, 24(3-4):235-265, 1997.
40. B.W. Koo, H.S. Jang, J.H. Song, ***Constructing and Cryptanalysis of a 16x16 Binary Matrix as a Diffusion Layer***, in: Proceedings of WISA 2003, in: LNCS, vol: 2908, 489-503, 2003.
41. B. Aslan, ***Boole Fonksiyonları ve S-Kutularının Kriptografik Özelliklerinin İncelenmesi ve Ters Haritalama Tabanlı Cebirsel Açıdan Güçlendirilmiş Bir S-kutusu Önerisi***, Trakya Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, 2008.
42. A. M. Youssef, S.E. Tavares, G. Gong, ***On Some probabilistic approximations for AES-like s-boxes***, Discrete Mathematics, 306(16):2016-2020, Elsevier, 2006.
43. B. Aslan, M. T. Sakallı, E.Buluş, ***Classifying 8-bit to 8-bit S-boxes based on Power Mappings from the point of DDT and LAT Distributions***, International Workshop on the Arithmetic of Finite Fields", WAIFI 2008, Lecture Notes in Computer Science, Vol. 5130, , Springer-Verlag, pp.123-133, 2008.

TEZ SIRASINDA YAPILAN ÇALIŞMALAR

1- Aslan, B. and Sakallı, M. T. (2012), Algebraic construction of cryptographically good binary linear transformations, Security and Communication Networks, doi: 10.1002/sec.556.

2- Füsun Yavuzer Aslan, M. Tolga Sakallı, Bora Aslan, Selma Bulut, A New Involutory 4×4 MDS Matrix for the AES-like Block Ciphers, The International Review on Computers and Software (IRECOS), January 2011 Vol 6, N 1, Italy.

3- M. Tolga Sakalli, Bora Aslan, Ercan Bulus, Andac Sahin Mesut, Fatma Büyüksaraçoglu, Osman Karaahmetoglu, On the Algebraic Expression of the AES S-Box Like S-Boxes, NDT (1) 2010: 213-227.

4- Muharrem Tolga Sakallı, Bora Aslan, On the Algebraic Construction of Cryptographically Good 32×32 Binary Linear Transformations, International Conference on Applied and Computational Mathematics, METU, Ankara, 2012.

5- M. Tolga Sakalli and B. Aslan, Algebraic Construction of 16×16 Binary Matrices of Branch Number 7 with One Fixed Point, ACA 2012 18th International Conference on Applications of Computer Algebra, Sofia, Bulgaria, 2012.

6- Füsun Yavuzer ASLAN, M. Tolga SAKALLI, Bora ASLAN, Önemli Blok Şifrelerde Kullanılan Doğrusal Dönüşümlerin İncelenmesi, Akademik Bilişim 2012, Uşak, 2012.